

The main body of the document is a large grid of approximately 15 columns and 15 rows of small, illegible technical diagrams or data tables. Each cell in the grid contains a small, complex diagram or table, likely representing a control system layout or data points. The text within these cells is too small to be read, but the overall structure is a dense, organized array of technical information.

TSV05

TSV05 CONTROL PART 4  
CVTSDD0

AH-T100D-MC  
2 OF 2 JAN 1986  
COPYRIGHT © 1982-85

**digital**  
MADE IN USA

This block contains a vertical column of 15 small, illegible tables or diagrams on the left side of the page. Each entry appears to be a small table with multiple columns and rows of text, but the text is too small to read. The diagrams are arranged in a regular grid pattern.

.REM\_  
IDENTIFICATION

PRODUCT ID: AC-T099D-MC  
PRODUCT TITLE: CVTSDDO TSV05 CTRL PART 4  
DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PGG  
DATE: AUGUST 23, 1985

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.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1982,1985 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL  
DEC

PDP  
DECUS

UNIBUS  
DECTAPE

MASSBUS

## TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	COMMANDS
2.2	SWITCHES
2.3	FLAGS
2.4	HARDWARE QUESTIONS
2.5	SOFTWARE QUESTIONS
2.6	EXTENDED P-TABLE DIALOGUE
2.7	QUICK STARTUP PROCEDURE
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES
7.0	MAINTENANCE HISTORY

## 1.0 GENERAL INFORMATION

### 1.1 PROGRAM ABSTRACT

THIS IS A PDP-11/23 RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSV05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A PDP-11/23 SYSTEM (Q-BUS). THE PROGRAM PROVIDES ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS THAT AID IN THE REPAIR OF THE DEVICE. THIS DIAGNOSTIC CONSIST OF EIGHT TEST WHICH ARE EXECUTED IN SEQUENCE.

THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP+, ACT, APT, SLIDE AND PAPER TAPE. FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP+ USER'S MANUAL. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

### 1.2 SYSTEM REQUIREMENTS

PDP-11/23 PROCESSOR AND MEMORY  
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY  
(28K USEABLE AND 4K RESERVED FOR I/O PAGE)  
TSV05 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)  
CONSOLE TERMINAL  
PDP-11 DIAGNOSTIC SUPERVISOR (HSAASYS VERSION 34 OR LATER)  
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

### 1.3 RELATED DOCUMENTS AND STANDARDS

#### DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. CHQUS XXDP+ USERS MANUAL; DOCUMENT NUMBER AC-F348E-MC  
DATE: 14 JULY 1980.
2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE; DOCUMENT NUMBER EK-TSV05-UG-001  
DATE: AUGUST 1982
3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSV05-TM-001  
DATE: AUGUST 1982
4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSV05-IN-001  
DATE: AUGUST 1982

#### 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

FUNCTIONAL PDP-11/23 CENTRAL PROCESSOR AND MEMORY  
FUNCTIONAL CONSOLE TERMINAL  
FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR  
FUNCTIONAL DIAGNOSTIC LOADER/MONITOR (XXDP+)

#### 1.5 ASSUMPTIONS

ALL HARDWARE EXCEPT THE HARDWARE UNDER TEST IS ASSUMED TO WORK PROPERLY OR FALSE ERRORS CAN BE REPORTED.  
THE TAPE BEING USED ON THE TS05 TRANSPORT IS A KNOWN GOOD REEL OF TAPE.  
CVTSAA, CVTSBA AND CVTSCA HAVE SUCCESSFULLY RUN.

#### 2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES. FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL (CHQUS).

#### 2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES (SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
-----	-----
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER +C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SECTION 4.0)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".

## 2.1.1 OPERATOR COMMANDS

THE TSV05 DIAGNOSTIC IS A PDP-11/23 DIAGNOSTIC SUPERVISOR COMPATIBLE PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE CHQUS XXDP+ USERS MANUAL, DOCUMENT NUMBER AC-F348E-MC. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC XXDP MEDIA

```
.R VTSD??
DIAG. RUN-TIME SERVICES REV D. APR 79
CVTSD-B-0
****TSV05 LOGIC DIAGNOSTIC****
UNIT IS TSV05
>DR
```

## 2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH. IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

```
START/TESTS:1-5/PASS:1000/EOP:100
```

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY, FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

### 2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST

\*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1



SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A "BELL" ON ERROR, YOU MAY USE THE FOLLOWING STRING:

/FLAGS:LOE:IER:BOE

#### 2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?" YOU MUST ANSWER "Y" AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 14 OF THE XXDP+ USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A "Y", THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL).

AFTER INITIAL STARTING OF THE PROGRAM (START COMMAND TO THE DIAGNOSTIC SUPERVISOR), THE PROGRAM WILL ISSUE THE "CHANGE HW?" QUESTION TO ASK IF THE HARDWARE PARAMETERS ARE TO BE CHANGED (BY THE OPERATOR).

ON A "N" (NO) RESPONSE TO THE "CHANGE HW?" QUESTION, THE DIAGNOSTIC WILL RUN USING THE DEFAULT VALUES FOR ALL QUESTIONS. THE DEFAULT ADDRESS AND VECTOR ARE:

TSBA/TSDB = 172520, VECTOR = 224

ON A "Y" (YES) RESPONSE TO THE QUESTION, THE FOLLOWING QUESTIONS WILL THEN BE ASKED TO ALLOW THE OPERATOR TO SELECT THE UNITS TO BE TESTED. A VALUE, IF PRESENT, LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN IF ONLY A CARRIAGE RETURN IS TYPED AS A RESPONSE. A "(D)" IN A QUESTION INDICATES THAT A DECIMAL NUMBER IS REQUIRED AS A RESPONSE. AN "(O)" INDICATES AN OCTAL NUMBER IS BEING SOLICITED. AN "(L)" INDICATES THAT A LOGICAL RESPONSE IS TO BE MADE: "Y" FOR YES, "N" FOR NO.

# UNITS (D) ? <ENTER THE NUMBER OF M7196 CONTROLLERS  
PRESENT TO BE TESTED>

UNIT 0

DEVICE ADDRESS (O) 172520 ? <ENTER THE ADDRESS OF THE  
TSBA/TSDB REGISTER>

VECTOR (O) 224 ? <ENTER ADDRESS OF INTERRUPT  
VECTOR>

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE "# UNITS?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING AS FOLLOWS:  
UP TO 4 TSV05 CONTROLLERS PER 11/23 AND UP TO 2 DRIVES PER CONTROLLER

## 2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". THE SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

THE FOLLOWING QUESTIONS ARE ASKED ON A START, RESTART, OR CONTINUE. THEY ALLOW FLEXIBILITY IN THE WAY THE PROGRAM BEHAVES.

CHANGE SW (L) ? <TYPE Y TO CAUSE THE FOLLOWING  
QUESTIONS TO BE ASKED>

INHIBIT ITERATIONS (L) N ? <TYPE "Y" TO PREVENT MULTIPLE  
ITERATIONS OF CERTAIN TESTS.  
THIS CAUSES EACH TEST PASS TO  
RUN AS QUICKLY AS POSSIBLE.  
ONLY QUICK-RUNNING LOGIC  
TESTS USE MULTIPLE  
ITERATIONS.>

## 2.6 EXTENDED P-TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES IN A TABLE THAT DESCRIBES THE DEVICES UNDER TEST. THE SIMPLEST WAY TO BUILD THIS TABLE IS TO ANSWER ALL QUESTIONS FOR EACH UNIT TO BE TESTED. IF YOU HAVE A MULTIPLEXED DEVICE SUCH AS A MASS STORAGE CONTROLLER WITH SEVERAL DRIVES OR A COMMUNICATION DEVICE WITH SEVERAL LINES, THIS BECOMES TEDIOUS SINCE MOST OF THE ANSWERS ARE REPETITIOUS.

TO ILLUSTRATE A MORE EFFICIENT METHOD, SUPPOSE YOU ARE TESTING A DEVICE, THE XY11. SUPPOSE THIS DEVICE CONSISTS OF A CONTROL MODULE WITH EIGHT UNITS (SUB-DEVICES) ATTACHED TO IT. THESE UNITS ARE DESCRIBED BY THE OCTAL NUMBERS 0 THROUGH 7. THERE IS ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS CALLED THE Q-FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

# UNITS (D) ? 8<CR>

UNIT 1  
CSR ADDRESS (O) ? 160000<CR>  
SUB-DEVICE # (O) ? 0<CR>  
Q-FACTOR (O) 0 ? 1<CR>

UNIT 2  
CSR ADDRESS (O) ? 160000<CR>  
SUB-DEVICE # (O) ? 1<CR>  
Q-FACTOR (O) 1 ? 0<CR>

UNIT 3  
CSR ADDRESS (O) ? 160000<CR>  
SUB-DEVICE # (O) ? 2<CR>

```
Q-FACTOR (0) 0 ? <CR>
UNIT 4
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 3<CR>
Q-FACTOR (0) 0 ? <CR>
UNIT 5
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 4<CR>
Q-FACTOR (0) 0 ? <CR>
UNIT 6
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 5<CR>
Q-FACTOR (0) 0 ? <CR>
UNIT 7
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 6<CR>
Q-FACTOR (0) 0 ? 1<CR>
UNIT 8
CSR ADDRESS (0) 160000<CR>
SUB-DEVICE # (0) ? 7<CR>
Q-FACTOR (0) 1 ? <CR>
```

NOTICE THAT THE DEFAULT VALUE FOR THE Q-FACTOR CHANGES WHEN A NON-DEFAULT RESPONSE IS GIVEN. BE CAREFUL WHEN SPECIFYING MULTIPLE UNITS!

AS YOU CAN SEE FROM THE ABOVE EXAMPLE, THE HARDWARE PARAMETERS DO NOT VARY SIGNIFICANTLY FROM UNIT TO UNIT. THE PROCEDURE SHOWN IS NOT VERY EFFICIENT.

THE RUNTIME SERVICES CAN TAKE MULTIPLE UNIT SPECIFICATIONS HOWEVER. LET'S BUILD THE SAME TABLE USING THE MULTIPLE SPECIFICATION FEATURE.

```
# UNITS (0) ? 8<CR>
UNIT 1
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 0,1<CR>
Q-FACTOR (0) 0 ? 1,0<CR>
UNIT 3
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 2-5<CR>
Q-FACTOR (0) 0 ? 0<CR>
UNIT 7
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 6,7<CR>
Q-FACTOR (0) 0 ? 1<CR>
```

AS YOU CAN SEE IN THE ABOVE DIALOGUE, THE RUNTIME SERVICES WILL

BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. IN THE FIRST PASS, TWO ENTRIES ARE BUILT SINCE TWO SUB-DEVICES AND Q-FACTORS WERE SPECIFIED. THE SERVICES ASSUME THAT THE CSR ADDRESS IS 160000 FOR BOTH SINCE IT WAS SPECIFIED ONLY ONCE. IN THE SECOND PASS, FOUR ENTRIES WERE BUILT. THIS IS BECAUSE FOUR SUB-DEVICES WERE SPECIFIED. THE "-" CONSTRUCT TELLS THE RUNTIME SERVICES TO INCREMENT THE DATA FROM THE FIRST NUMBER TO THE SECOND. IN THIS CASE, SUB-DEVICES 2, 3, 4 AND 5 WERE SPECIFIED. (IF THE SUB-DEVICE WERE SPECIFIED BY ADDRESSES, THE INCREMENT WOULD BE BY 2 SINCE ADDRESSES MUST BE ON AN EVEN BOUNDARY.) THE CSR ADDRESSES AND Q-FACTORS FOR THE FOUR ENTRIES ARE ASSUMED TO BE 160000 AND 0 RESPECTIVELY SINCE THEY WERE ONLY SPECIFIED ONCE. THE LAST TWO UNITS ARE SPECIFIED IN THE THIRD PASS.

THE WHOLE PROCESS COULD HAVE BEEN ACCOMPLISHED IN ONE PASS AS SHOWN BELOW.

♦ UNITS (0) ? 8<CR>

UNIT 1  
CSR ADDRESS (0) ? 160000<CR>  
SUB-DEVICE # (0) ? 0-7<CR>  
Q-FACTOR (0) 0 ? 0.1.0....1.1<CR>

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING A NULL FIELD) TELL THE RUNTIME SERVICES TO REPEAT THE LAST REPLY.

## 2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK) QUESTIONS
3. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM
4. TYPE "START"
5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. THESE DEFAULTS ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

## 3.0 ERROR INFORMATION

### 3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY A DIAGNOSTIC: GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES ARE ALWAYS PRINTED UNLESS THE "IER" FLAG IS SET (SECTION 2.3). THE GENERAL ERROR MESSAGE IS OF THE FORM:

NAME TYPE NUMBER ON UNIT NUMBER TST NUMBER PC:XXXXXX  
ERROR MESSAGE

.WHERE; NAME = DIAGNOSTIC NAME  
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)  
NUMBER = ERROR NUMBER  
UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)  
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED  
PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS THE "IER" OR "IBR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBR" OR "IXE" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

## 3.2 SPECIFIC ERROR MESSAGES

BELOW ARE SAMPLE ERROR MESSAGES. EACH ERROR MESSAGE REPRESENTS DIFFERENT TYPES OF ERRORS DETECTED BY THIS DIAGNOSTIC.

## ERROR MESSAGE EXAMPLE 1

THIS ERROR IS INDICATIVE OF AN INCORRECT REGISTER OR STATUS WORD RETURNED TO THE DIAGNOSTIC. THE FIRST PART DEFINES THE TEST FUNCTION AND UNIT THAT FAILED. THE SECOND PART PROVIDES THE REGISTER BITS AND THEIR MNEMONICS FOR THE INCORRECT REGISTER OR STATUS WORDS. THE THIRD PART IS THE EXPECTED AND RECEIVED DATA.

TST: 016 FIFO EXERCISER TEST  
CVTSD HRD ERR 01610 ON UNIT 00 TST 016 SUB 002 PC: 040624  
FIFO STATUS (IN WORD 9) INCORRECT AFTER WRITE FIFO

TAPE BUS SIGNALS IN WORD #8: - DESIGNATOR <BIT #>  
PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>  
IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>  
IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>

TAPE BUS SIGNALS IN WORD #9:  
DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>

MESSAGE BUFFER ADDRESS = 047352

MESSAGE BUFFER CONTENTS:

WORD #0	EXPD: 100020	RECV: 100020	XOR: 000000
WORD #1	EXPD: 000012	RECV: 000012	XOR: 000000
WORD #2	EXPD: 000000	RECV: 000000	XOR: 000000
WORD #3	EXPD: 000010	RECV: 000010	XOR: 000000
WORD #4	EXPD: 000000	RECV: 000000	XOR: 000000
WORD #5	EXPD: 000000	RECV: 000000	XOR: 000000
WORD #6	EXPD: 000000	RECV: 000000	XOR: 000000
WORD #7	EXPD: 000000	RECV: 000000	XOR: 000000
WORD #8	EXPD: 070217	RECV: 070217	XOR: 000000
WORD #9	EXPD: 000074	RECV: 000034	XOR: 000040

## ERROR MESSAGE EXAMPLE 2

THIS ERROR SHOWS A FATAL FUNCTION ERROR FROM THE TAPE DRIVE. IN THIS INSTANCE A UNRECOVERABLE ERROR OCCURED WHICH INDICATES THAT THE CONTROLLER MAY BE DEFECTIVE.

CVTSD HRD ERR 00159 ON UNIT 00 TST 001 SUB 005 PC: 026202  
TSSR NOT CORRECT AFTER SPACE RECORDS COMMAND

TSSR = 100214

TSSR BITS SET: SC,SSR

TERMINATION CLASS CODE = UNRECOVERABLE ERROR

PACKET ADDRESS = 026420

PACKET WORD # = 140010

PACKET WORD # = 000010

PACKET WORD # = 000000

PACKET WORD # = 000024

ERROR MESSAGE EXAMPLE 3

THIS ERROR SHOWS THAT THE MOTION BIT DID NOT GET SET WHILE DOING A REWIND  
WITH EXTENDED FEATURES MODE ENABLED.

CVTSD WRD ERR 00121 ON UNIT 00 TST 001 SUB 002 PC: 023306  
MOT BIT (XST0) NOT SET DURING REWIND (EXTENDED FEATURES MODE)  
EXPD: 000312 RECV: 000112 XOR: 000200

## 4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH THE TOTAL NUMBER OF ERRORS REPORTED SINCE THE DIAGNOSTIC WAS STARTED. THE "EOP" SWITCH CAN BE USED TO CONTROL HOW OFTEN THE END OF PASS MESSAGE IS PRINTED. SECTION 2.2 DESCRIBES SWITCHES.

## SUCCESSFUL RUN EXAMPLE (PDP-11/23)

```
DR>STA/FLA:PNT:HOE
UNITS (D) ? 1
UNIT 0
DEVICE ADDRESS (0) 172520 ? <CR>
VECTOR (0) 224 ? <CR>
CHANGE SW (L) ? N<CR>
```

THE ABOVE COMMAND WILL START THE DIAGNOSTIC. THE COMMAND HAS TWO SWITCHES ON WHICH ARE "PRINT EACH TEST NBR AS EXECUTED" AND "HALT ON ERROR".

```
TST: 001 SKIP TAPE MARKS TEST
TST: 002 NO-OP AND INITIALIZE TEST
TST: 003 ERASE AND OPERATION INCOMPLETE TEST
TST: 004 DATA PARITY TEST
TST: 005 TEST OF OPERATIONS AT EOT TEST
TST: 006 EXTENDED-MODE FUNCTIONS TEST
TST: 007 RECORD BUFFERING TEST
TST: 008 FUNCTION TIMING TEST
```

## 0 ERRORS

NOTE: THE DIAGNOSTIC WILL RUN CONTINUOUSLY UNLESS A PASS NUMBER LIMIT HAS BEEN SPECIFIED WITH THE "/PASS:" SWITCH.

## PROGRAM RUN TIMES

THE AVERAGE RUN TIMES OF THE PROGRAM ARE LISTED BELOW. THESE FIGURES ARE TO BE USED AS A GUIDE. THE TIMING WAS DONE ON A PDP-11/23 PROCESSOR WITH A LA34 CONSOLE.

THE PROGRAM RUNS IN TWO MODES; NO ITERATIONS AND DEFAULT MODE. IN THE NO ITERATIONS MODE, EACH TEST IS RUN ONCE, WITH NO ITERATIONS. IN THE DEFAULT MODE EACH TEST IS REPEATED BY THE NUMBER OF TIMES INDICATED BY THE ITERATION COUNT. NO ITERATIONS MODE IS SELECTED BY ANSWERING THE INHIBIT ITERATIONS QUESTION WITH A "Y" (YES).

TEST NUMBER	N/I SECS.	NUMBER ITER	DEF SECS.
1	1	2	1
2	1	1	0



3	1	1	0
4	1	1	0
5	1	1	0
6	1	1	0
7	1	1	0
8	1	1	0

THE TIMES REQUIRED TO RUN TESTS 1 THROUGH 37 IN ONE COMMAND:

Q.V. 15 SECONDS  
DEFAULT 16 SECONDS

#### 5.0 DEVICE INFORMATION TABLES

WHENEVER THE PROGRAM IS STARTED, VIA THE STA(RT) COMMAND, THE SUPERVISOR REQUESTS THE FOLLOWING P-TABLES PARAMETER CHANGES:

CHANGE HW (L) ?

# UNITS (D) ? <ENTER THE NUMBER OF M7196 CONTROLLERS  
PRESENT TO BE TESTED>

UNIT 0

DEVICE ADDRESS (O) 172520 ? <ENTER THE ADDRESS OF THE  
TSBA/TSDB REGISTER>

VECTOR (O) 224 ? <ENTER ADDRESS OF INTERRUPT  
VECTOR>

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE "# UNITS?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING.

IN ADDITION, ON A START, RESTART OR CONTINUE THE SUPERVISOR REQUESTS CHANGES TO THE SOFTWARE OPERATING PARAMETERS, AS FOLLOWS:

CHANGE SW (L) ?

INHIBIT ITERATIONS (L) N ?

## 6.0 TEST SUMMARIES

### TEST 1: WRITE TAPE MARK RETRY

THIS TEST VERIFIES PROPER OPERATION OF THE WRITE TAPE MARK RETRY COMMAND (SPACE REVERSE, ERASE, WRITE TAPE MARK).

### TEST 2: SKIP TAPE MARKS

THIS TEST VERIFIES PROPER OPERATION OF THE SKIP TAPE MARKS FORWARD AND SKIP TAPE MARKS REVERSE COMMANDS. PROPER OPERATION UNDER CONTROL OF ALL COMBINATIONS OF THE ENABLE SKIP TAPE MARKS STOP (ESS) AND ENABLE TAPE MARKS STOP OFF BOT (ENB) BITS SPECIFIED BY THE WRITE CHARACTERISTICS COMMAND.

### TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

THIS TEST VERIFIES PROPER OPERATION OF THE NO-OP ("CLEAN TAPE") AND INITIALIZE COMMAND (SPACE REVERSE, ERASE, WRITE DATA)

### TEST 4: ERASE AND OPERATION INCOMPLETE

VERIFIES THAT AN ERASE COMMAND ISSUED WHEN THE TAPE IS POSITIONED AT BOT OPERATES PROPERLY AND ACTUALLY ERASES TAPE.

### TEST 5: DATA PARITY TEST

This test verifies that the data parity circuitry in both the controller and the transport is operating properly by forcing data records with wrong parity to be written onto tape and checking the results obtained when the data is read.

### TEST 6: OPERATIONS AT EOT

THIS TEST VERIFIES PROPER OPERATION OF THE WRITE DATA RETRY COMMAND (SPACE REVERSE, ERASE, WRITE DATA)

### TEST 7: EXTENDED MODE FEATURES

THIS TEST VERIFIES THE OPERATION OF COMMANDS ONLY AVAILABLE WHEN THE CONTROLLER IS IN THE EXTENDED FEATURES MODE. THESE COMMANDS ARE:

REWIND WITH IMMEDIATE INTERRUPT

IF THE CONTROLLER IS NOT ALREADY IN EXTENDED FEATURES MODE, IT IS PLACED THERE VIA A WRITE SUBSYSTEM MEMORY COMMAND.

**TEST 8: RECORD BUFFERING**

THIS TEST VERIFIES THAT RECORD BUFFERING, USED FOR WRITE DATA AND READ NEXT COMMANDS, OPERATES PROPERLY AND IS PROPERLY CONTROLLED BY THE EXTENDED CHARACTERISTICS DATA WORD. IF THE M7196 CONTROLLER MODULE IS NOT ALREADY IN EXTENDED FEATURES MODE (AS CONTROLLED BY THE DIP SWITCH ON THE MODULE), IT IS PLACED INTO THAT MODE BY INVERTING THE SENSE OF THE SWITCH USING THE WRITE SUBSYSTEM MEMORY COMMAND. NOTE THAT RECORD BUFFERING HAS BEEN ENABLED IN PREVIOUS TESTS OF READ AND WRITE AND SO HAS BEEN PARTIALLY TESTED ALREADY. THIS TEST VERIFIES THAT BUFFERING IS ACTUALLY OPERATING.

**TEST 9: FUNCTION TIMING**

THIS TEST VERIFIES THAT THE TAPE TRANSPORT SEEMS TO BE WRITING RECORDS, GAPS, AND EXTENDED GAPS OF THE PROPER LENGTH. BOTH LOW AND HIGH SPEED MODES ARE TESTED. IT IS ALSO VERIFIED THAT A SKIP TAPE MARKS COMMAND WITH A COUNT OF 5 OR MORE, OPERATE THE TAPE IN HIGH-SPEED MODE. THIS TEST CAN ONLY BE RUN IF A REAL-TIME CLOCK IS AVAILIABLE ON THE SYSTEM. THE TEST OPERATES BY TIMING VARIOUS TAPE-MOTION OPERATIONS, USING A NUMBER OF DIFFERENT TEST RECORD LENGTHS.

**7.0 MAINTENANCE HISTORY**

REVISION A - MARCH 1982

REVISION B - APRIL 1983

- FIXED TWO PROBLEMS, ONE IN TEST 1 AND THE OTHER IN TEST 8.  
REF. DOYLE TO GRASKY "TSV05 CVTSDA DIAGNOSTIC PATCH"; 23-DEC-82.

REVISION C - JUNE 1984

MINOR CHANGES FOR "ORION" CPU  
ELIMINATED CPU ID MESSAGE.

REVISION D - JUNE 1985

CHANGES MADE TO ALLOW DIAGNOSTICS TO WORK WITH  
XXDP+ V2.1 (DRSXM) EXTENDED MONITOR.

```

811 .TITLE TSV2 - PROGRAM HEADER
812 .SBTTL PROGRAM HEADER
813
819 .MCALL SVC
820 000000 SVC ; INITIALIZE SUPERVISOR MACROS
821 .ENABLE LC
822 .NLIST BEX,CND
828 000000 .ENABL ABS,AMA
829 002000 002000 .=2000
830 002000 002000 BGNMOD TSV2
831 TSV2::
832 ;**
833 ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
834 ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
835 ;--
836
837 002000 POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
838 002000 HEADER CVTSD,D,0,655,0
002000 L$NAME:: ;DIAGNOSTIC NAME
002000 103 .ASCII /C/
002001 126 .ASCII /V/
002002 124 .ASCII /T/
002003 123 .ASCII /S/
002004 104 .ASCII /D/
002005 000 .BYTE 0
002006 000 .BYTE 0
002007 000 .BYTE 0
002010 L$REV:: ;REVISION LEVEL
002010 104 .ASCII /D/
002011 L$DEPO:: ;0
002011 060 .ASCII /0/
002012 L$UNIT:: ;NUMBER OF UNITS
002012 000000 .WORD 0
002014 L$TIML:: ;LONGEST TEST TIME
002014 001217 .WORD 655.
002016 L$HPCP:: ;PTR. TO H.W. QUES.
002016 105446 .WORD L$HARD
002020 L$SPCP:: ;PTR. TO S.W. QUES.
002020 105600 .WORD L$SOFT
002022 L$HPTP:: ;PTR. TO DEF. H.W. PTABLE
002022 002150 .WORD L$HW
002024 L$SPTP:: ;PTR. TO S.W. PTABLE
002024 002160 .WORD L$SW
002026 L$LADP:: ;DIAG. END ADDRESS
002026 106404 .WORD L$LAST
002030 L$STA:: ;RESERVED FOR APT STATS
002030 000000 .WORD 0
002032 L$CO::
002032 000000 .WORD 0
002034 L$DTYP:: ;DIAGNOSTIC TYPE
002034 000000 .WORD 0
002036 L$APT:: ;APT EXPANSION
002036 000000 .WORD 0
002040 L$DTP:: ;PTR. TO DISPATCH TABLE
002040 002124 .WORD L$DISPATCH
002042 L$PRIO:: ;DIAGNOSTIC RUN PRIORITY

```

## PROGRAM HEADER

002042	000000	L\$ENVI::	.WORD	0	
002044					;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000	L\$EXP1::	.WORD	0	
002046					;EXPANSION WORD
002046	000000	L\$MREV::	.WORD	0	
002050					;SVC REV AND EDIT *
002050	003		.BYTE	C\$REVISION	
002051	003		.BYTE	C\$EDIT	
002052		L\$EF::			;DIAG. EVENT FLAGS
002052	000000		.WORD	0	
002054	000000		.WORD	0	
002056		L\$SPC::			
002056	000000		.WORD	0	
002060		L\$DEVP::			; POINTER TO DEVICE TYPE LIST
002060	003374		.WORD	L\$DVTYP	
002062		L\$REPP::			;PTR. TO REPORT CODE
002062	022650		.WORD	L\$RPT	
002064		L\$EXP4::			
002064	C00000		.WORD	0	
002066		L\$EXP5::			
002066	000000		.WORD	0	
002070		L\$AUT::			;PTR. TO ADD UNIT CODE
002070	022336		.WORD	L\$AU	
002072		L\$DUT::			;PTR. TO DROP UNIT CODE
002072	022434		.WORD	L\$DU	
002074		L\$LUN::			;LUN FOR EXERCISERS TO FILL
002074	000000		.WORD	0	
002076		L\$DESP::			;POINTER TO DIAG. DESCRIPTION
002076	003402		.WORD	L\$DESC	
002100		L\$LOAD::			;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		EMT	E\$LOAD	
002102		L\$ETP::			;POINTER TO ERR TBL
002102	000000		.WORD	0	
002104		L\$ICP::			;PTR. TO INIT CODE
002104	021542		.WORD	L\$INIT	
002106		L\$CCP::			;PTR. TO CLEAN-UP CODE
002106	022622		.WORD	L\$CLEAN	
002110		L\$ACP::			;PTR. TO AUTO CODE
002110	022542		.WORD	L\$AUTO	
002112		L\$PRT::			;PTR. TO PROTECT TABLE
002112	021532		.WORD	L\$PROT	
002114		L\$TEST::			;TEST NUMBER
002114	000000		.WORD	0	
002116		L\$DLY::			;DELAY COUNT
002116	000000		.WORD	0	
002120		L\$HIME::			;PTR. TO HIGH MEM
002120	000000		.WORD	0	

DISPATCH TABLE

840  
841  
842  
843  
844  
845  
846  
847 002122  
002122 000011  
002124  
002124 023432  
002126 032234  
002130 041332  
002132 046670  
002134 052746  
002136 055742  
002140 063314  
002142 073244  
002144 101020  
848

.SBTTL DISPATCH TABLE

;++  
; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.  
; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.  
;--

DISPATCH 9  
.WORD 9  
L\$DISPATCH::  
.WORD T1  
.WORD T2  
.WORD T3  
.WORD T4  
.WORD T5  
.WORD T6  
.WORD T7  
.WORD T8  
.WORD T9

DEFAULT HARDWARE P-TABLE

```

850                                     .SBTTL  DEFAULT HARDWARE P-TABLE
851
852                                     ;++
853                                     ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
854                                     ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
855                                     ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
856                                     ;--
857 002146      BGNHW      DFPTBL      ;DEFAULT HARD-P-TABLE
      002146      000003      .WORD      L10000-L$HW/2
      002150
      002150      L$HW::
      DFPTBL::
858
859 002150      172520      .WORD      172520      ; 1ST (OF 2) REGISTERS.
860 002152      000224      .WORD      224        ; INTERRUPT VECTOR
861 002154      000200      .WORD      PRI04      ; INTERRUPT PRIORITY.
862 002156
      002156      ENDHW
      L10000:

```

SOFTWARE P-TABLE

```

864
865
866
867
868
869
870 002156
      002156 000004
      002160
      002160
871
872 002160 000000
873 002162 000000
874
875
876 002164 000017
877 002166 000310
878 002170
      002170
879
880 002170

      .SBTTL SOFTWARE P-TABLE

      ;**
      ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
      ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
      ;--
      BGNSW SFPTBL
      .WORD L10001-L$SW/2

L$SW::
SFPTBL::

TRANSTST:: .WORD 0 ; ENABLE TEST OF TRANSPORT(S) IF =1
NOITS:: .WORD 0 ; INHIBIT ITERATION OPTION.
; ... 0 = ITERATE.
; ... NZ = INHIBIT ITERATE.
LERRMAX:: .WORD 15. ; LOCAL (PER TEST) ERROR LIMIT
GERRMAX:: .WORD 200. ; GLOBAL (PER UNIT) ERROR LIMIT

      ENDSW
L10001:

      ENDMOD

```





## GLOBAL EQUATES SECTION

```

000340      PRI07== 340
000300      PRI06== 300
000240      PRI05== 240
000200      PRI04== 200
000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0

```

## ; OPERATOR FLAG BITS

```

000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      HOE== 100000

```

916  
917 002170

```

          KT11
          .SBTTL MEMORY MANAGEMENT DEFINITIONS
          ;*KT11 VECTOR ADDRESS
000250      MMVEC= 250
          ;*KT11 STATUS REGISTER ADDRESSES
177572      SR0= 177572
177574      SR1= 177574
177576      SR2= 177576
172516      SR3= 172516
          .IF NB
          ;*USER "I" PAGE DESCRIPTOR REGISTERS
UIPDR0= 177600
UIPDR1= 177602
UIPDR2= 177604
UIPDR3= 177606
UIPDR4= 177610
UIPDR5= 177612
UIPDR6= 177614
UIPDR7= 177616
          .IF NB
          ;*USER "D" PAGE DESCRIPTOR REGISTERS
UDPDR0= 177620
UDPDR1= 177622
UDPDR2= 177624
UDPDR3= 177626
UDPDR4= 177630
UDPDR5= 177632
UDPDR6= 177634
UDPDR7= 177636
          .ENDC
;DEFINE MEMORY MANAGEMENT REGISTERS

```

## MEMORY MANAGEMENT DEFINITIONS

```
;*USER "I" PAGE ADDRESS REGISTERS
UIPAR0= 177640
UIPAR1= 177642
UIPAR2= 177644
UIPAR3= 177646
UIPAR4= 177650
UIPAR5= 177652
UIPAR6= 177654
UIPAR7= 177656
  .IF NB
;*USER "D" PAGE ADDRESS REGISTERS
UDPAR0= 177660
UDPAR1= 177662
UDPAR2= 177664
UDPAR3= 177666
UDPAR4= 177670
UDPAR5= 177672
UDPAR6= 177674
UDPAR7= 177676
  .ENDC
  .ENDC
  .IF NB
;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
SIPDR0= 172200
SIPDR1= 172202
SIPDR2= 172204
SIPDR3= 172206
SIPDR4= 172210
SIPDR5= 172212
SIPDR6= 172214
SIPDR7= 172216
  .IF NB
;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
SDPDR0= 172220
SDPDR1= 172222
SDPDR2= 172224
SDPDR3= 172226
SDPDR4= 172230
SDPDR5= 172232
SDPDR6= 172234
SDPDR7= 172236
  .ENDC
;*SUPERVISOR "I" PAGE ADDRESS REGISTERS
SIPAR0= 172240
SIPAR1= 172242
SIPAR2= 172244
SIPAR3= 172246
SIPAR4= 172250
SIPAR5= 172252
SIPAR6= 172254
SIPAR7= 172256
  .IF NB
;*SUPERVISOR "D" PAGE ADDRESS REGISTERS
SDPAR0= 172260
SDPAR1= 172262
SDPAR2= 172264
SDPAR3= 172266
```

MEMORY MANAGEMENT DEFINITIONS

```
SDPAR4= 172270
SDPAR5= 172272
SDPAR6= 172274
SDPAR7= 172276
.ENDC
.ENDC
; *KERNEL "I" PAGE DESCRIPTOR REGISTERS
172300 KIPDR0= 172300
172302 KIPDR1= 172302
172304 KIPDR2= 172304
172306 KIPDR3= 172306
172310 KIPDR4= 172310
172312 KIPDR5= 172312
172314 KIPDR6= 172314
172316 KIPDR7= 172316
; *KERNEL "D" PAGE
DESCRIPTOR REGISTERS
KDPDR0= 172320
KDPDR1= 172322
KDPDR2= 172324
KDPDR3= 172326
KDPDR4= 172330
KDPDR5= 172332
KDPDR6= 172334
KDPDR7= 172336
.ENDC
; *KERNEL "I" PAGE ADDRESS REGISTERS
172340 KIPAR0= 172340
172342 KIPAR1= 172342
172344 KIPAR2= 172344
172346 KIPAR3= 172346
172350 KIPAR4= 172350
172352 KIPAR5= 172352
172354 KIPAR6= 172354
172356 KIPAR7= 172356
; *KERNEL "D" PAGE ADDRESS REGISTERS
KDPAR0= 172360
KDPAR1= 172362
KDPAR2= 172364
KDPAR3= 172366
KDPAR4= 172370
KDPAR5= 172372
KDPAR6= 172374
KDPAR7= 172376
.ENDC
```

TSV05 REGISTER AND PACKET DEFINITIONS

.SBTTL TSV05 REGISTER AND PACKET DEFINITIONS

```

922
923
924
925
926
927
928      000004      ERRVEC==      4      ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.
929      000060      TTIVEC==     60      ; INTERRUPT VECTOR FOR CONSOLE INPUT
930      177560      TTICSR==    177560    ; BUS ADDRESS OF CONSOLE INPUT
931      177562      TTIBFR==    177562    ; CONSOLE INPUT DATA BUFFER
932      177520      BDVPCR==    177520    ; BDV11 PAGE CONTROL REGISTER
933
934
935      ;+
936      ;BIT DEFINITIONS FOR TSSR REGISTER
937      ;-
938      100000      SC=      BIT15      ;SPECIAL CONDITION
939      040000      BIE=      BIT14      ;BUS INTERFACE ERROR
940      C20000      SCE=      BIT13      ;SANITY CHECK ERROR
941      010000      RMR=      BIT12      ;MODIFICATION REFUSED
942      004000      NXM=      BIT11      ;NONEXISTANT MEMORY ERROR
943      002000      NBA=      BIT10      ;NEED BUFFER ADDRESS
944      001400      HIADDR= BIT9!BIT8    ;EXTENDED ADDRESS BITS
945      000200      SSR=      BIT7      ;SUB SYSTEM READY
946      000100      OFL=      BIT6      ;OFF LINE BIT
947      000060      FATERR= BIT4!BIT5    ;FATAL TERMINATION ERROR CODES
948      000016      TERCLS= BIT3!BIT2!BIT1 ;TERMINATION CODES
949
950
951      ;+
952      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
953      ;(XSTO)
954      ;
955      ;-
956
957      100000      XSOTMK= BIT15      ;TAPE MARK DETECTED
958      040000      XSORLS= BIT14      ;RECORD LENGTH SHORT
959      020000      XSOLET= BIT13      ;LOGICAL END OF TAPE
960      010000      XSORLL= BIT12      ;RECORD LENGTH LONG
961      004000      XSOWLE= BIT11      ;WRITE LOCK ERROR
962      002000      XSONEF= BIT10      ;NON EXECUTABLE FUNCTION
963      001000      XSOILC= BIT9      ;ILLEGAL COMMAND
964      000400      XSOILA= BIT8      ;ILLEGAL ADDRESS
965      000200      XSOMOT= BIT7      ;TAPE IN MOTION
966      000100      XSOONL= BIT6      ;TRANSPORT ON LINE
967      000040      XSOIE=  BIT5      ;INTERRUPT ENABLE
968      000020      XSOVCK= BIT4      ;VOLUME CHECK BIT
969      000010      XSOPED= BIT3      ;PHASE ENCODED DRIVE
970      000004      XSOWLK= BIT2      ;WRITE LOCKED
971      000002      XSOBOT= BIT1      ;BEGINNING OF TAPE
972      000001      XSOEOT= BIT0      ;END OF TAPE

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

974
975      ;+
976      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
977      ;(XST1)
978      ;-
978      100000 X1.DLT = BIT15          ;DATA LATE
979      040000 X1.SPARE= BIT14          ;NOT USED
980      020000 X1.COR = BIT13          ;CORRECTABLE DATA ERROR
981      017375 X1.MBZ = BIT12+BIT11+BIT10+BIT9+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT0 ;ALWAYS 0
982      000400 X1.RBP = BIT8          ;READ BUS PARITY ERROR
983      000002 X1.UNC = BIT1          ;UNCORRECTABLE DATA OR HARD ERROR
984
985      ;+
986      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
987      ;(XST2)
988      ;-
989      100000 X2.OPM = BIT15          ;OPERATION IN PROGRESS (TAPE MOVING)
990      040000 X2.RCE = BIT14          ;RAM CHECKSUM ERROR
991      035400 X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8 ;NOT USED BY TSV05 (ALWAYS=0)
992      020000 X2.WCF = BIT10          ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
993      000200 X2.EXTF = BIT7          ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
994      000100 X2.BUFE = BIT6          ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
995      000077 X2.REV = 000077        ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
996      000007 X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
997
998      ;+
999      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
1000     ;(XST3)
1001     ;-
1002     177400 X3.MDE = 177400        ;MICRO-DIAGNOSTIC ERROR CODE
1003     000200 X3.SPARE= BIT7          ;NOT USED BY TSV05
1004     000100 X3.OPI = BIT6          ;OPERATION INCOMPLETE
1005     000040 X3.REV = BIT5          ;REVERSE
1006     000020 X3.TRF = BIT4          ;TRANSPORT RESPONSE FAILURE
1007     000010 X3.DCK = BIT3          ;DENSITY CHECK
1008     000006 X3.MBZ =BIT2+BIT1      ;NOT USED ALWAYS 0
1009     000001 X3.RIB = BIT0          ;REVERSE INTO BOT
1010
1011     ;+
1012     ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
1013     ;(XST4)
1014     ;-
1015     100000 X4.HSP = BIT15          ;HIGH SPEED
1016     040000 X4.RCE = BIT14          ;RETRY COUNT EXCEEDED
1017     020000 X4.TSM = BIT13          ;TRANSPORT SPECIAL MODE
1018     017400 X4.MBZ = BIT12+BIT11+BIT10+BIT9+BIT8 ;NOT USED ALWAYS 0
1019     000377 X4.WRC = 000377        ;WRITE RETRY COUNT FIELD
1020
1021     ;+
1022     ;
1023     ;TSSR TERMINATION CODES (BIT 0-2)
1024     ;
1025     ;-
1026
1027     000006 TSREJ= 3+2              ;COMMAND REJECTED
1028     000006 UNREC= 6                ;UNRECOVERABLE ERROR

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1030      ;+
1031      ;
1032      ;DEVICE REGISTER OFFSETS
1033      ;
1034      ;-
1035
1036      000000      TSBA== 0
1037      000000      TSDB== 0      ;TSDB/TSBA REGISTER
1038      000001      TSBAH== 1
1039      000001      TSDBH== 1      ;TSDB/TSBA REGISTER HIGH BYTE
1040      000002      TSSR== 2      ;TSSR REGISTER
1041      000003      TSSRH== 3      ;TSSR REGISTER HIGH BYTE
1042
1043      ;+
1044      ; TSDB ADDRESS BIT DEFINITIONS
1045      ;-
1046      000003      A1716 = BIT1:BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
1047
1048      ;+
1049      ; COMMAND DEFINITIONS
1050      ;-
1051      000017      P.GETSTAT      = 17      ;GET STATUS
1052      000013      P.INIT        = 13      ;INITIALIZE
1053      000012      P.CONTROL     = 12      ;CONTROL COMMANDS
1054      000011      P.FORMAT      = 11      ;FORMAT
1055      000010      P.POSITION    = 10      ;POSITION
1056      000006      P.WRTSUB      = 6       ;SUBSYSTEM WRITE
1057      000005      P.WRITE       = 5       ;WRITE
1058      000004      P.WRTCHAR    = 4       ;WRITE CHARACTERISTICS
1059      000001      P.READ        = 1       ;READ
1060
1061      ;+
1062      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
1063      ;-
1064      100000      P.ACK        = BIT15      ;BUFFER AVAIL FOR CONTROLLER
1065      040000      P.CVC        = BIT14      ;CLEAR VOLUME CHECK
1066      020000      P.OPP        = BIT13      ;REVERSE SEQUENCE OF DATA BITS
1067      010000      P.SWB        = BIT12      ;SWAP BYTES IN MEMORY
1068      007400      P.MODE       = BIT11:BIT10:BIT9:BIT8 ;EXTENDED COMMAND MODE FIELD
1069      000200      P.IE         = BIT7       ;INTERRUPT ENABLE
1070      000140      P.FMT= BIT6:BIT5      ;PACKET HEADER TYPE (ALWAYS=0)
1071      000037      P.CMD        = 37       ;MAJOR COMMAND FIELD
1072
1073      ;+
1074      ; CONTROL COMMAND MODE CODES
1075      ;-
1075      000000      PC.RELEASE    = 0*256.      ;RELEASE BUFFER
1076      000400      PC.REWIND    = 1*256.      ;REWIND
1077      001000      PC.NOOP      = 2*256.      ;NO-OP
1078      002000      PC.IEREW     = 4*256.      ;REWIND IMMEDIATE INTERRUPT
1079      002400      PC.ERASE     = 5*256.      ;SECURITY ERASE
    
```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1081
1082      ;*
1083      ; CONTROLLER RAM DEFINITIONS
1084      ;-
1085      000167      RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
1086      000200      RMCHEND = 200      ;CHARACTERISTICS IO DATA END RAM ADDRESS
1087      000201      RMPKTBE= 201      ;COMMAND PACKET BEGIN RAM ADDRESS
1088      000210      RMPKTEND= 210      ;COMMAND PACKET END RAM ADDRESS
1089      000215      RMMSGBE= 215      ;MESSAGE BUFFER BEGIN RAM ADDRESS
1090      000234      RMMSGEND= 234      ;MESSAGE BUFFER END RAM ADDRESS
1091      ;*
1092      ; REGISTER DEFINITIONS IN THE MESSAGE BUFFER
1093      ;-
1094
1095
1096      000006      XST0== 6      ;EXTENDED STATUS REGISTER 0 (WORD 4)
1097      000010      XST1== 8.      ;EXTENDED STATUS REGISTER 1 (WORD 5)
1098      000012      XST2== 10.      ;EXTENDED STATUS REGISTER 2 (WORD 6)
1099      C00014      XST3== 12.      ;EXTENDED STATUS REGISTER 3 (WORD 7)
1100      000016      XST4== 14.      ;EXTENDED STATUS REGISTER 4 (WORD 8)
1101
1102      ;*
1103      ; OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
1104      ;-
1105
1106
1107
1108      000002      PKLOW = 2      ;LOW ORDER CHARACTERISTIC DATA POINTER
1109      000004      PKHI = 4      ;HIGH ORDER CHARACTERISTIC DATA POINTER
1110      000006      PKBCNT = 6      ;NUMBER OF BYTES IN DATA PACKET
1111
1112      000010      EXBCNT=10      ;NUMBER OF BYTES IN EXTENDED DATA PACKET
1113
1114      ;*
1115      ; DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
1116      ;-
1117      000000      BSELO = 0      ;BYTE 0
1118      000001      BSEL1 = 1      ;BYTE 1
1119      000002      SEL2 = 2      ;WORD 2
1120      000004      SELDATA = 4      ;WORD 3

```



TSV05 REGISTER AND PACKET DEFINITIONS

```

1122
1123      ;*
1124      ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
1125      ;-
1126      000000      PW.NOP          = 0          ;NO-OP
1127      000001      PW.RDRAM        = 1          ;READ RAM
1128      000002      PW.WTRAM        = 2          ;WRITE RAM
1129      000003      PW.RFIFO        = 3          ;READ FIFO
1130      000004      PW.WFIFO        = 4          ;WRITE FIFO
1131      000005      PW.RDSTAT       = 5          ;READ STATUS
1132      000006      PW.WCTL         = 6          ;WRITE TAPE CONTROL
1133      000007      PW.WFMT         = 7          ;WRITE TAPE FORMAT
1134      000010      PW.WMISC        = 10         ;WRITE MISCELLANEOUS
1135      000011      PW.WNPR         = 11         ;WRITE NPR CONTROL
1136      000020      PW.D22          = 20         ;DO MICROTEST 22
1137      000021      PW.D11          = 21         ;DO MICROTEST 11
1138      000022      PW.D13          = 22         ;DO MICROTEST 13
1139      000023      PW.NO1311      = 23         ;DISABLE MICROTEST 11 AND 13
1140      000024      PW.RDXT         = 24         ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
1141
1142      ;*
1143      ;BSEL1 CODES FOR WRITE TAPE CONTROL
1144      ;-
1145      000200      WC.IFAD          = BIT7       ;IFAD - FORMATTER ADDRESS
1146      000100      WC.IOTAD        = BIT6       ;ITADO - TRANSPORT ADDRESS BIT 0
1147      000040      WC.I1TAD        = BIT5       ;ITAD1 - TRANSPORT ADDRESS BIT 1
1148      000020      WC.I5RESV       = BIT4       ;IRESV5 - RESERVED #5
1149      000010      WC.IREW         = BIT3       ;IREW - REWIND
1150      000004      WC.IRWU         = BIT2       ;IRWU - REWIND AND UNLOAD
1151      000002      WC.IFEN         = BIT1       ;IFEN - FORMATTER ENABLE
1152      000001      WC.IGO          = BIT0       ;GO
1153
1154      ;*
1155      ;BSEL1 CODES FOR WRITE FORMAT
1156      ;-
1157      000200      WF.IHISP         = BIT7       ;IHISP - HIGH SPEED
1158      000100      WF.IWRT         = BIT6       ;IWRT - WRITE
1159      000040      WF.IREV         = BIT5       ;IREV - REVERSE
1160      000020      WF.IWFM         = BIT4       ;IWFM - WRITE FILE MARK
1161      000010      WF.IEDIT        = BIT3       ;IEDIT - EDIT
1162      000004      WF.IERASE       = BIT2       ;IERASE - ERASE
1163      000002      WF.I3RESV       = BIT1       ;IRESV3 - RESERVED #3
1164      000001      WF.I4RESV       = BIT0       ;IRESV4 - RESERVED #4
1165
1166      ;*
1167      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
1168      ;-
1169      000200      MS.EXT           = BIT7       ;INVERT SENSE OF EXTENDED FEATURES SWITCH
1170      000020      MS.RSFIFO       = BIT4       ;RESET FIFO AND INPUT PARITY ERRORR
1171      000010      MS.RSTAPE       = BIT3       ;RESET TAPE STATUS IN 2 FLIP-FLOPS
1172      000006      MS.ATTN         = BIT2:BIT1  ;ATTENTION TRIGGER FIELD
1173      000001      MS.RSD          = BIT0       ;RESET TIMER A,B THEN DELAY TIMES IN SEL2

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1174
1175      ;+
1176      ; MS.ATTN SUBCODES
1177      ;-
1177      000000      MSA.NOP = 0*2      ;NO-OP (NOTHING TRIGGERED)
1178      000002      MSA.VOL = 1*2      ;SIMULATE ON-LINE/OFF-LINE TRANSITION
1179      000004      MSA.NRAM= 2*2      ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
1180      000006      MSA.FRAME= 3*2     ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
1181
1182      ;+
1183      ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
1184      ;-
1184      000200      NP.IR      = BIT7      ;INTERRUPT REQUEST (0-1 TRANSITION)
1185      000100      NP.OUT     = BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
1186      000040      NP.LOOP   = BIT5      ;ENABLE TRANSPORT LOOPBACK
1187      000020      NP.WRP    = BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
1188
1189      ;+
1190      ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
1191      ;-
1192      000200      S2.DIM      = BIT7      ;WORD #9 BYTE 2 DATA IN MISS
1193      000100      S2.ILW     = BIT6      ; ILW H
1194      000040      S2.OUTRDY  = BIT5      ; OUT RDY H
1195      000020      S2.INRDY   = BIT4      ; IN RDY H
1196      000010      S2.ATIMR   = BIT3      ; TIMER A FLAG H
1197      000004      S2.BTIMR   = BIT2      ; TIMER B FLAG H
1198      000003      S2.UNDEF   = BIT1+BIT0 ;(UNDEFINED)
1199      100000      S1.PARIN   = BIT15     ;WORD #8 BYTE 1 PARIN H
1200      040000      S1.I2RESV  = BIT14     ; IRESV2
1201      020000      S1.I1RESV  = BIT13     ; IRESV1
1202      010000      S1.IEOT    = BIT12     ; IEOT L
1203      004000      S1.IIDENT  = BIT11     ; IIDENT H
1204      002000      S1.ICER    = BIT10     ; ICER H
1205      001000      S1.IFMK    = BIT9      ; IFMK H
1206      000400      S1.IHER    = BIT8      ; IHER H
1207      000200      S0.ISPEED  = BIT7      ;WORD #8 BYTE 0 ISPEED H
1208      000100      S0.IRDY   = BIT6      ; IRDY L
1209      000040      S0.IONL    = BIT5      ; IONL L
1210      000020      S0.ILDP    = BIT4      ; ILDP L
1211      000010      S0.IDBY    = BIT3      ; IDBY L
1212      000004      S0.IRWD    = BIT2      ; IRWD L
1213      000002      S0.IFBY    = BIT1      ; IFBY L
1214      000001      S0.IFPT    = BIT0      ; IFPT L
    
```

## SPECIAL MACROS AND OPDEFS.

```

1216             .SBTTL SPECIAL MACROS AND OPDEFS.
1217
1218             ;+
1219             ;SAVE GENERAL REGS 1 TO 5
1220             ;-
1221
1222             .MACRO SAVREG
1223             JSR   RS,REGSAV
1224             .ENDM
1225
1226             ;+
1227             ; MACRO TO FORCE AN ERROR
1228             ;-
1229             .MACRO FORCERROR TAG,NOTSSR
1230             .NLIST
1231             .IIF NDF LISTALL, .NLIST
1232             .LIST
1233             .IF B NOTSSR
1234             MOV   TSSR(R5),R1 ;READ TSSR
1235             .ENDC
1236             MOV   FORCER,FORCER ;IS FORCER SET? (LEAVE C BIT ALONE)
1237             BNE  TAG           ;BR IF YES
1238             .NLIST
1239             .IIF NDF LISTALL, .LIST
1240             .LIST
1241             .ENDM
1242
1243             ;+
1244             ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
1245             ; WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
1246             ; SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
1247             ; FORCER TO 17777
1248             ; TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
1249             ;-
1250             .MACRO FORCEEXIT TAG
1251             .NLIST
1252             .IIF NDF LISTALL, .NLIST
1253             .LIST
1254             MOV   FORCER,FORCER ;IS FORCER NEGATIVE?
1255             BMI  TAG           ;BR IF YES
1256             .NLIST
1257             .IIF NDF LISTALL, .LIST
1258             .LIST
1259             .ENDM
1260             ;+
1261             ; MACRO TO INCREMENT ERROR COUNTS
1262             ;-
1263             .MACRO NEXT.ERRNO
1264             .NLIST
1265             ;;;.IIF NDF LISTALL, .NLIST
1266             ERRNO=ERRNO+1
1267             ;;;.IIF NDF LISTALL, .LIST
1268             .LIST
1269             .ENDM

```

SPECIAL MACROS AND OPDEFS.

```

1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292

```

```

;+
;MACRO TO PERFORM XOR
;-

      .MACRO XOR A,B
      MOV A,-(SP)
      BIC B,(SP)
      BIC A,B
      BIS (SP)+,B
      .ENDM

EN=0 ; INITIALIZE ERROR NUMBER
.SBTTL FORCER - FORCE ERROR FLAG

;
; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
;
FORCER:: 0 ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
; - BY THE MACRO "IFERROR"). AN ERROR NEED NOT -
; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.

```

000000

002170 000000

## GLOBAL DATA SECTION

## .SBTTL GLOBAL DATA SECTION

```

1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305 002172 000000
1306 002174 000000
1307 002176 000000
1308 002200 000000
1309 002202 000224
1310 002204 000200
1311 002206 000000
1312 002210 000000
1313 002212 000000
1314 002214 000000
1315 002216 000000
1316 002220 000000
1317 002222 000000
1318 002224 000000
1319 002226 000000
1320 002230 000000
1321 002232 000000
1322 002234
1323 002274 000000
1324 002276 000000
1325 002300 000000
1326 002302 000000
1327 002304 000000
1328 002306 000000
1329 002310 000000
1330 002312 000000
1331 002314
1332 002460
1333 002624

```

```

;***
;THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
;IN MORE THAN ONE TEST.
;--

;
;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
;
EPRTSW::      .WORD  0      ;PRINT SWITCH
UNITN::       .WORD  0      ;UNIT # UNDER TEST.
QVP::         .WORD  0      ;QUICK VERIFY FLAG.
CSRADDR::     .WORD  0      ;ADDRESS OF CSR FOR CURRENT DEVICE
IVEC::        .WORD  224    ;INTERRUPT VECTOR
IPRI::        .WORD  PRI04  ;INTERRUPT PRIORITY.
TSTCNT::      .WORD  0      ;NUMBER OF TESTS RUN IN THIS PASS
LOOPCNT::     .WORD  0      ;REMAINING ITERATION COUNT FOR TEST
DEVCNT::      .WORD  0      ;NUMBER OF DEVICE UNDER TEST
FATFLG::      .WORD  0      ;SET IF FATAL ERROR IS DETECTED IN TEST
INTRECV::     .WORD  0      ;SET IF TAPE INTERRUPT WAS RECEIVED
EXTFEA::      .WORD  0      ;EXTENDED FEATURES SOFTWARE SW 0-OFF;1-ON
BENBSW::      .WORD  0      ;BUFFER ENABLE SWITCH SW 0-OFF;1-ON
EXPD::        .WORD  0      ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
RECV::        .WORD  0      ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
ERRHI::       .WORD  0      ;HIGH ADDRESS MEMORY ERROR
ERRLO::       .WORD  0      ;LOW ADDRESS MEMORY ERROR
RAMDATA::     .BLKW  16.    ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
RAMSIZ::      .WORD  0      ;RAM DATA SIZE FOR PRAMPKT ROUTINE
RCVHIADD::    .WORD  0      ;RECEIVED BUFFER HIGH ADDRESS
RCVLOADD::    .WORD  0      ;RECEIVED BUFFER LOW ADDRESS
COUNT::      .WORD  0      ;TEST COUNT PATTERN
DATA::        .WORD  0      ;TEST DATA
TSTFLAG::     .WORD  0      ;TEST FLAG WORD
TSTPTR::      .WORD  0      ;TSTBLK POINTER
PRMNO::       .WORD  0      ;PRINT ROUTINE TEMP
EXPMSG::      .BLKB  100.   ;EXPECTED MESSAGE BUFFER DATA
RECMSG::      .BLKB  100.   ;RECEIVED MESSAGE BUFFER DATA
TMPBFR::      .BLKB  80.    ;TEMPORARY STORAGE FOR PRINT

```

TSTBLK - TEST DATA TABLE

1335  
 1336  
 1337  
 1338  
 1339  
 1340  
 1341  
 1342  
 1343  
 1344  
 1345  
 1346  
 1347  
 1348  
 1349  
 1350  
 1351 002744  
 1352 002744 000000  
 1353 002746 177777  
 1354 002750 000001  
 1355 002752 000002  
 1356 002754 000004  
 1357 002756 000010  
 1358 002760 000020  
 1359 002762 000040  
 1360 002764 000100  
 1361 002766 000200  
 1362 002770 000400  
 1363 002772 001000  
 1364 002774 002000  
 1365 002776 004000  
 1366 003000 010000  
 1367 003002 020000  
 1368 003004 040000  
 1369 003006 100000  
 1370 003010 177776  
 1371 003012 177775  
 1372 003014 177773  
 1373 003016 177767  
 1374 003020 177757  
 1375 003022 177737  
 1376 003024 177677  
 1377 003026 177577  
 1378 003030 177377  
 1379 003032 176777  
 1380 003034 175777  
 1381 003036 173777  
 1382 003040 167777  
 1383 003042 157777  
 1384 003044 137777  
 1385 003046 077777  
 1386 003050 125252  
 1387 003052 052525  
 1388 003054

.SBTTL TSTBLK - TEST DATA TABLE

```

;+
; THIS TABLE CONTAINS TEST DATA USED IN SEVERAL TESTS
; IN SEQUENCE THE DATA IS:
;
;     ALL ZEROS
;     ALL ONES
;     WALKING ONES
;     WALKING ZEROS
;     ALTERNATING ONES AND ZEROS
;
;-
    
```

```

TSTBLK::
        .WORD 0 ;ALL ZEROS
        .WORD 177777 ;ALL ONES
        .WORD BIT0 ;DATA FOR WALKING ONES
        .WORD BIT1
        .WORD BIT2
        .WORD BIT3
        .WORD BIT4
        .WORD BIT5
        .WORD BIT6
        .WORD BIT7
        .WORD BIT8
        .WORD BIT9
        .WORD BIT10
        .WORD BIT11
        .WORD BIT12
        .WORD BIT13
        .WORD BIT14
        .WORD BIT15
        .WORD †CBIT0 ;DATA FOR WALKING ZEROS
        .WORD †CBIT1
        .WORD †CBIT2
        .WORD †CBIT3
        .WORD †CBIT4
        .WORD †CBIT5
        .WORD †CBIT6
        .WORD †CBIT7
        .WORD †CBIT8
        .WORD †CBIT9
        .WORD †CBIT10
        .WORD †CBIT11
        .WORD †CBIT12
        .WORD †CBIT13
        .WORD †CBIT14
        .WORD †CBIT15
        .WORD 125252 ;ALTERNATING ONES, ZEROS
        .WORD 052525 ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
        .WORD 003054
TBLEND==.
    
```

GLOBAL ENVIRONMENT STORAGE

```

1390          .SBTTL GLOBAL ENVIRONMENT STORAGE
1391          ;
1392          ;STORAGE FOR DEVICE REGISTERS
1393          ;
1394 003054 000000 100000 000000 DUMMY: 0,100000,0,0 ;DUMMY DEVICE REGISTERS...
1395 003064 000000 000000 000000      0,0,0,0,0,0,0,0,0 ;...FOR MULTI-UNIT CHECKOUT.
1396          ;
1397          ;
1398 003104 000000 DUFLG::      .WORD 0 ;"DROPPED UNIT" FLAG.
1399          ;INHIBITS CODE IN "CLEAN-UP".
1400 003106 000000 NODEV::      .WORD 0 ;FLAG TO SAY NO DEVICE.
1401          ;
1402 003110 000000 TEMP1::      .WORD 0 ;SOME TEMP LOCATIONS.
1403 003112 000000 TEMP2::      .WORD 0
1404 003114 000000 XXCOMM::     .WORD 0 ;XXDP+ COMM BLOCK POINTER.
1405 003116 000000 FREE::      .WORD 0 ;1ST FREE MEMORY ADDRESS...
1406 003120 000000 FRESIZ::     .WORD 0 ;...AND SIZE (IN WORDS).
1407 003122 000000 FREEHI: .WORD 0 ;LAST WORD IN FREE SPACE
1408 003124 000000 KTFLG::      .WORD 0 ;KT11, MEM AVAIL FLAG -
1409          ;- .WORD 0 = <24K OR NO KT -
1410          ;- NZ = >24K AND KT.
1411 003126 000000 KTENABLE::   .WORD 0 ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
1412 003130 000000 NXMFLG::    .WORD 0 ;SET IF WE CAN TEST CLEARED OTHERWISE
1413 003132 000000 NXMLO::     .WORD 0 ;NXM LO ADDRESS BITS
1414 003134 000000 NXMHI::     .WORD 0 ;NXM HI ADDRESS BITS FOR DAL'S 16-21
1415 003136 000000 T23A::      .WORD 0 ;11/23A FLAG
1416 003140 000000 T23B::      .WORD 0 ;11/23B FLAG
1417 003142 000000 T3BFLG::    .WORD 0 ;TEST 3B FLAG +0
1418 003144 002000 PST32W::    .WORD 2000 ;32W BLOCK ADDRESS FOR 32K START
1419 003146 000000 SIFLAG::    .WORD 0
1420 003150 000000 BADDAT::    .WORD 0 ;ACTUAL DATA
1421 003152 000000 GDDAT::     .WORD 0 ;EXPECTED DATA
1422 003154 000000 LOOPFL::    .WORD 0
1423 003156 000000 CTAB::      .WORD 0 ;CONFIGURATION TABLES.
1424 003156 000000 CTABM::     .WORD 0 ;CONFIG WORK.
1425 003160 000000          .WORD 0
1426 003162 000000          .WORD 0
1427 003164 000000          .WORD 0
1428 003166 177777          .WORD 0
1429 003170          .WORD -1 ;END OF MEM TABLE.
1430          ;
1431          ;CRABE::
1432          ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
1433          ;
1434          ;      0      =      UNIT NOT TESTED
1435          ;      100000 =      UNIT ONLINE, NO ERRORS
1436          ;      10XXXX =      UNIT ONLINE, ENCOUNTERED XXXX ERRORS
1437          ;      160000 =      UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
1438          ;      160001 =      UNIT DROPPED, NOT IDLE AT START
1439          ;      14XXXX =      UNIT DROPPED, ENCOUNTERED XXXX ERRORS
1440          ;
1441          ;
1442 003170 000000 ERTABL:      .BLKW 64.
1443          ;
1444 003370 000000 ERTABE:      .WORD 0
1445          ;
1446 003372 000000 SKIPT: .WORD 0 ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST
    
```

GLOBAL TEXT MESSAGES

1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1475  
1476  
1477  
1478  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515

003374			
003374	124	123	126
003374			
003402			
003402	052	052	052
003402			
003476	003536	003541	003545
003516	003577	003603	003607
003536	123	103	000
003541	102	111	105
003545	123	103	105
003551	122	115	122
003555	116	130	115
003561	116	102	101
003565	102	111	124
003572	102	111	124
003577	123	123	122
003603	117	106	114
003607	102	111	124
003614	102	111	124
003621	102	111	124
003626	102	111	124
003633	102	111	124
003640	102	111	124
003646	124	123	123
003701	124	123	123
003734	040	040	116
003773	045	101	040
004014	045	101	040
004054	045	101	040
004113	045	116	045
004117	040	040	125
004146	040	040	111
004211	045	116	045
004215	040	040	116
004252	040	040	111
004274	045	101	040

```

.SBTTL GLOBAL TEXT MESSAGES
;
; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
; MORE THAN ONE TEST.
;
; NAMES OF DEVICES SUPPORTED
;
; DEVTYP <TSV05>
L#DVTYP::
    .ASCIZ /TSV05/
    .EVEN

;
; TEST DESCRIPTION
;
; DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - CHECK TRANSPORT IF ERROR ****>
L#DESC::
    .ASCIZ /**** TSV05 LOGIC DIAGNOSTIC - CHECK TRANSPORT IF ERROR ****/
    .EVEN

;
; BIT TO ASCII CONVERSION FOR TSSR REGISTER
;
; TSSRBIT::
    .WORD 1#,2#,3#,4#,5#,6#,7#,8#
    .WORD 9#,10#,11#,12#,13#,14#,15#,16#
1#: .ASCIZ 'SC'
2#: .ASCIZ 'BIE'
3#: .ASCIZ 'SCE'
4#: .ASCIZ 'RMR'
5#: .ASCIZ 'NXM'
6#: .ASCIZ 'NBA'
7#: .ASCIZ 'BIT9'
8#: .ASCIZ 'BIT8'
9#: .ASCIZ 'SSR'
10#: .ASCIZ 'OFL'
11#: .ASCIZ 'BITS'
12#: .ASCIZ 'BIT4'
13#: .ASCIZ 'BITS'
14#: .ASCIZ 'BIT2'
15#: .ASCIZ 'BIT1'
16#: .ASCIZ 'BIT0'
    .EVEN
SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
SFHERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
NXR: .ASCIZ / NON-EXISTANT DEVICE REGISTER/
NXRX: .ASCIZ /#A ADDRESS: #06/
TSSX: .ASCII /#A TSBA,TSSR EXP'D: #06#A,#06#N/
    .ASCIZ /#A TSBA,TSSR REC'D: #06#A,#06/
FUSI: .ASCII /#N#A/
USI: .ASCIZ / UNEXPECTED INTERRUPT/
NSI: .ASCIZ / INTERRUPT EXPECTED. NOT RECEIVED/
FNOINTR: .ASCII /#N#A/
NOINTR: .ASCIZ / NO INTERRUPT WAS GENERATED/
IFault: .ASCIZ / INTERRUPT FAULT/
INTX: .ASCIZ /#A CPU PC: #06#A TSBA: #06/
    
```



GLOBAL TEXT MESSAGES

```

1516 004331 040 040 042 NOINIT: .ASCIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
1517 004403 040 040 042 NSINIT: .ASCIZ / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
1518 004453 040 040 042 BRINIT: .ASCIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
1519 004523 000 NUL: .ASCIZ //
1520 004524 045 116 000 NULCR: .ASCIZ /#N/
1521 004527 045 101 040 EXPGOT: .ASCIZ /#A EXP'D: #06#A, REC'D: #06/
1522 004563 045 116 045 EXPGT2: .ASCIZ /#N#A EXP'D: #06#A, #06#N#A REC'D: #0#A, #06/
1523 004637 045 101 040 DUAD12: .ASCIZ /#A REG(W) WRITTEN TO: #06#A REG(R) READ; EXP'D: #06#A, REC'D: #06/
1524 004741 122 101 115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
1525 005007 040 040 103 SCHE: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
1526 005052 127 122 111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
1527 005107 124 123 123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
1528 005202 124 123 123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
1529 005274 106 101 124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
1530 005366 105 122 122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
1531 005454 045 116 045 NOMEM: .ASCIZ '#N#A ***** NO NXH ADDRESS--CANNOT TEST NXH TIMEOUT. *****N'
1532 005550 045 116 045 M8186: .ASCIZ '#N#A ***** 11/23A SYSTEM *****N'
1533 005641 045 116 045 M8189: .ASCIZ '#N#A ***** 11/23B SYSTEM *****N'

```

.EVEN  
.SBTTL GLOBAL ERROR REPORT SECTION

```

; **
; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
; CALLS THAT ARE USED IN MORE THAN ONE TEST.
; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
; --

```

```

NXRERR: BGNMSG NXRERR ;NON-EXISTANT DEVICE REGISTER.
;
PRINTX #NXRX,NODEV ;NODEV = NEXM ADDRESS.
MOV NODEV,-(SP)
MOV #NXRX,-(SP)
MOV #2,-(SP)
MOV SP,RO
TRAP C#PNTX
ADD #6,SP
JSR PC,EXTEND ; PRINT EXTENSION IF REQUIRED.
ENDMSG

```

L10002: TRAP C#MSG

```

; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
; TO ANY OF THE ABOVE ERROR SIGNATURES.

```

```

EXTEND: TST (PC)+
EXTA: 0 ; 0 = NO EXTENSION.
BEQ 1#
JSR PC,BEXTA ; APPEND EXTENSION TEXT.
1#: PRINTX #NULCR ; PRINT A BLANK LINE
MOV #NULCR,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C#PNTX
ADD #4,SP
RTS PC

```

```

1542 005732
005732
1543 005732 013746 003106
005736 012746 003773
005742 012746 000002
005746 010600
005750 104415
005752 062706 000006
1544 005756 004737 005764
1545 005762
005762
005762 104423
1546
1547
1548
1549
1550 005764 005727
1551 005766 000000
1552 005770 001402
1553 005772 004777 177770
1554 005776
005776 012746 004524
006002 012746 000001
006006 010600
006010 104415
006012 062706 000004
1555 006016 000207

```

PRITSSR - PRINT TSSR CONTENTS

```

1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575 006020
1576 006020
1577 006024 010104
1578 006026
      006026 010446
      006030 012746 006473
      006034 012746 000002
      006040 010600
      006042 104414
      006044 062706 000006
1579 006050 010400
1580 006052 004737 016124
1581 006056 103410
1582 006060
      006060 012746 006713
      006064 012746 000001
      006070 010600
      006072 104415
      006074 062706 000004
1583 006100 010403
1584 006102 042703 001476
1585 006106 001434
1586 006110 012702 002624
1587 006114 012701 003476
1588 006120 005703
1589 006122 001413
1590 006124 000241
1591 006126 006103
1592 006130 103006
1593 006132 011100
1594 006134 112022
1595 006136 001376
1596 006140 112762 000054 177777
1597 006146 005721
1598 006150 000763
1599 006152 105042
1600 006154
      006154 012746 002624
      006160 012746 006664
    
```

```

.SBTTL PRITSSR - PRINT TSSR CONTENTS
;
; ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
; THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
; BY A MESSAGE PRINTING ROUTINE
;
; INPUTS:
;
; R1 CONTENTS OF TSSR
;
; SUBORDINATE ROUTINES:
;
; CHKAMB CHECK FOR AMBIGUOUS CONTENTS
;
;-
PRITSSR:
  SAVREG                                ;SAVE GENERAL REGISTERS
  MOV R1,R4                             ;SAVE THE TSSR CONTENTS
  PRINTB @TSSRFOR,R4                   ;PRINT THE CONTENTS OF TSSR
  MOV R4,-(SP)
  MOV @TSSRFOR,-(SP)
  MOV @2,-(SP)
  MOV SP,R0
  TRAP C#PNTB
  ADD #6,SP
  MOV R4,R0                             ;GET TSSR BACK FOR CHKAMB
  JSR PC,CHKAMB                         ;ARE CONTENTS AMBIGUOUS ?
  BCS 5#                                ;BRANCH IF NOT
  PRINTX @AMBTSSR                       ;SHOW CONTENTS ARE AMBIGUOUS
  MOV @AMBTSSR,-(SP)
  MOV #1,-(SP)
  MOV SP,R0
  TRAP C#PNTX
  ADD #4,SP
5#: MOV R4,R3                             ;CONTENTS OF TSSR
  BIC #HIADDR!FATERR!TERCLS,R3         ;CLEAR ALL MULTIPLE BIT FIELDS
  BEQ 20#                                ;NO BITS ARE SET
  MOV @TMPBFR,R2                        ;TEMPORARY ASCII BUFFER
  MOV @TSSRBIT,R1                       ;ASCII EQUIVALENT OF BITS
10#: TST R3                              ;REMAINING BITS TO CONVERT
  BEQ 15#                                ;BRANCH WHEN ALL ARE DONE
  CLC                                    ;CLEAR CARRY FOR SHIFT
  ROL R3                                ;SHIFT NEXT BIT TO CARRY
  BCC 13#                                ;BRANCH IF BIT NOT SET
  MOV (R1),R0                           ;POINTER TO BIT DEFINITION
11#: MOVB (R0)+,(R2)+                   ;MOVE ASCII TO BUFFER
  BNE 11#                                ;MOVE ALL BITS
  MOVB #' ,,-1(R2)                      ;INSERT A COMMA TO TERMINATE
13#: TST (R1)+                          ;POINT TO NEXT DESCRIPTION
  BR 10#                                 ;GET THE REMAINING BITS
15#: CLRB -(R2)                         ;TERMINATE THE LINE
  PRINTX @TSSDEF,@TMPBFR                ;PRINT THE BIT DEFINITIONS
  MOV @TMPBFR,-(SP)
  MOV @TSSDEF,-(SP)
    
```

PRITSSR - PRINT TSSR CONTENTS

006164	012746	000002		MOV	#2,-(SP)	
006170	010600			MOV	SP,R0	
006172	104415			TRAP	C#PNTX	
006174	062706	000006		ADD	#6,SP	
1601						
1602	006200	010403	20#:	MOV	R4,R3	;GET THE TSSR CONTENTS
1603	006202	042703		BIC	#+CTERCLS,R3	;CLEAR ALL BUT TERMINATION
1604	006206	016303		MOV	TCOCOD(R3),R3	;GET THE TERMINATION CODE MEANING
1605	006212			PRINTX	#TCOASC,R3	;PRINT THE TERMINATION CODE
	006212	010346		MOV	R3,-(SP)	
	006214	012746		MOV	#TCOASC,-(SP)	
	006220	012746		MOV	#2,-(SP)	
	006224	010600		MOV	SP,R0	
	006226	104415		TRAP	C#PNTX	
	006230	062706		ADD	#6,SP	
1606	006234	010403		MOV	R4,R3	;TSSR CONTENTS AGAIN
1607	006236	042703	177717	BIC	#+CFATERR,R3	;CLEAR ALL BUT FATAL TERMINATION
1608	006242	001416		BEQ	25#	;DON'T PRINT IF ZERO
1609	006244	C06203		ASR	R3	
1610	006246	006203		ASR	R3	
1611	006250	006203		ASR	R3	;ALINE TERMINATION CODE FOR INDEX
1612	006252	016303	007314	MOV	TSFCOD(R3),R3	;GET THE FATAL TERMINATION CODE
1613	006256			PRINTX	#TFCASC,R3	;PRINT THE FATAL TERMINATION CODE
	006256	010346		MOV	R3,-(SP)	
	006260	012746	006615	MOV	#TFCASC,-(SP)	
	006264	012746	000002	MOV	#2,-(SP)	
	006270	010600		MOV	SP,R0	
	006272	104415		TRAP	C#PNTX	
	006274	062706	000006	ADD	#6,SP	
1614	006300	042704	176377	25#:	BIC	#+CHIADDR,R4
1615	006304	001411		BEQ	30#	;CLEAR ALL BUT EXTENDED ADDRESS
1616	006306			PRINTX	#TEXASC,R4	;DON'T PRINT IF ZERO
	006306	010446				;PRINT THE EXTENDED ADDRESS BITS
	006310	012746	006513	MOV	R4,-(SP)	
	006314	012746	000002	MOV	#TEXASC,-(SP)	
	006320	010600		MOV	#2,-(SP)	
	006322	104415		MOV	SP,R0	
	006324	062706	000006	TRAP	C#PNTX	
1617	006330	013703	002172	30#:	ADD	#6,SP
1618	006334			MOV	EPRTSW,R3	;PRINT MEASGE BUFFER ADDRESS
	006334	010346		PRINTX	R3	;PRINT PROPER MESSAGE
	006336	012746	000001	MOV	R3,-(SP)	
	006342	010600		MOV	#1,-(SP)	
	006344	104415		MOV	SP,R0	
	006346	062706	000004	TRAP	C#PNTX	
1619	006352	000207		ADD	#4,SP	
				RTS	PC	;RETURN TO CALLER

## PRITSSR - PRINT TSSR CONTENTS

1635	006354	045	116	045	EPRT1:	.ASCIZ	'#N#A *****CHECK TRANSPORT*****'
1636	006413	045	116	045	EPRT2:	.ASCIZ	'#N#A *****CHECK PARITY SWITCH IN TRANSPORT*****'
1638	006473	045	116	045	TSSRFOR:	.ASCIZ	'#N#A TSSR = #06'
1639	006513	045	116	045	TEXASC:	.ASCIZ	'#N#A Extended Address Bits = #06'
1640	006354	045	116	045	TCOASC:	.ASCIZ	'#N#A Termination Class Code = #T'
1641	006615	045	116	045	TFCASC:	.ASCIZ	'#N#A Fatal Termination Class Code = #T'
1642	006664	045	116	045	TSSDEF:	.ASCIZ	'#N#A TSSR Bits Set: #T'
1643	006713	045	116	045	AMBTSSR:	.ASCIZ	'#N#A TSSR Contents Are Ambiguous'
1644						.EVEN	
1645	006754	006774	007017	007045	TCOCOD:	.WORD	1#,2#,3#,4#,5#,6#,7#,8#
1646	006774	116	157	162	1#:	.ASCIZ	'Normal Termination'
1647	007017	124	145	162	2#:	.ASCIZ	'Termination Condition'
1648	007045	124	141	160	3#:	.ASCIZ	'Tape Status Alert'
1649	007067	106	165	156	4#:	.ASCIZ	'Function Reject'
1650	007107	122	145	143	5#:	.ASCIZ	'Recoverable Error - Tape Position One Record Down'
1651	007171	122	145	143	6#:	.ASCIZ	'Recoverable Error - Tape Was Not Moved'
1652	007240	125	156	162	7#:	.ASCIZ	'Unrecoverable Error'
1653	007264	106	141	164	8#:	.ASCIZ	'Fatal Controller Error'
1654						.EVEN	
1655							
1656	007314	007324	007360	007371	TSFCOD:	.WORD	1#,2#,3#,4#
1657	007324	111	156	164	1#:	.ASCIZ	'Internal Diagnostic Failure'
1658	007360	122	145	163	2#:	.ASCIZ	'Reserved'
1659	007371	102	165	163	3#:	.ASCIZ	'Bus Interface or Sanity Check Error'
1660	007435	122	145	163	4#:	.ASCIZ	'Reserved'
1661						.EVEN	

PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

.SBTTL PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

```

1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678 007446
1679 007446
1680 007452 010005
1681 007454 C05737 003126
1682 007460 001001
1683 007462 005003
1684 007464 010301
1685 007466 010400
1686 007470 006100
1687 007472 006101
1688 007474
    007474 010446
    007476 010146
    007500 012746 007632
    007504 012746 000003
    007510 010600
    007512 104414
    007514 062706 000010
1689 007520 010300
1690 007522 001404
1691 007524 010401
1692 007526 004737 017376
1693 007532 010004
1694 007534 005001
1695 007536 012402
1696 007540
    007540 010246
    007542 010146
    007544 012746 007574
    007550 012746 000003
    007554 010600
    007556 104414
    007560 062706 000010
1697 007564 005201
1698 007566 020105
1699 007570 002762
1700 007572 000207
1701
1702 007574 045 116 045 PKTFRM: .ASCIZ '#N#A Packet Word #D1#A = #06'
1703 007632 045 116 045 PKTADD: .ASCIZ '#N#A Packet Address = #01#05'
1704 .EVEN
    
```

```

;*
;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
;
;INPUT:
;
;      R0      NUMBER OF WORDS IN PACKET
;      R3      HIGH ORDER COMMAND PACKET ADDRESS
;      R4      ADDRESS OF COMMAND PACKET
;
;      NOTE:   R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
;-
    
```

```

PRIPKT::
    SAVREG                ;SAVE THE REGISTERS
    MOV R0,R5             ;SAVE NO. OF WORDS IN PACKET
    TST KTENABLE         ;ABOVE 28K UNDER TEST?
    BNE 10#              ;BR IF YES
    CLR R3               ;SET HIGH ORDER ADDRESS TO 0
10#: MOV R3,R1           ;COPY HIGH ORDER ADDRESS
    MOV R4,R0           ;GET LOWER ADDRESS
    ROL R0              ;SHIFT BIT 15 INTO C BIT
    ROL R1              ;AND INTO HIGH ORDER.
    PRINTB #PKTADD,R1,R4 ;PRINT PACKET ADDRESS
    MOV R4,-(SP)
    MOV R1,-(SP)
    MOV #PKTADD,-(SP)
    MOV #3,-(SP)
    MOV SP,R0
    TRAP C#PNTB
    ADD #10,SP
15#: MOV R3,R0         ;GET HIGH ORDER ADDRESS
    BEQ 20#           ;BR IF NOT ABOVE 28K.
    MOV R4,R1         ;GET LOW ORDER ADDRESS
    JSR PC,SETMAP    ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
    MOV R0,R4        ;GET RETURNED PAR6 ADDRESS BIAS
20#: CLR R1          ;SAVE WORD NUMBER
25#: MOV (R4)+,R2    ;GET PACKET CONTENTS
    PRINTB #PKTFRM,R1,R2 ;PRINT THE DATA
    MOV R2,-(SP)
    MOV R1,-(SP)
    MOV #PKTFRM,-(SP)
    MOV #3,-(SP)
    MOV SP,R0
    TRAP C#PNTB
    ADD #10,SP
    INC R1            ;NEXT WORD NUMBER
    CMP R1,R5        ;DONE ALL PACKET WORDS?
    BLT 25#         ;LOOP TILL ALL DONE
    RTS PC          ;RETURN
    
```

PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

```

1706                                     .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR BYTE
1707
1708                                     ;+
1709                                     ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
1710                                     ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
1711                                     ;
1712                                     ;INPUTS:
1713                                     ;
1714                                     ;       R1      RECEIVED DATA
1715                                     ;       R2      EXPECTED DATA
1716                                     ;
1717                                     ;OUTPUT:
1718                                     ;
1719                                     ;       R0      XOR OF EXPECTED/RECEIVED DATA
1720                                     ;-
1721 007670 PRIBXOR::
1722 007670         SAVREG                ;SAVE THE REGISTERS
1723 007674 010203     MOV      R2,R3      ;EXPECTED DATA
1724 007676         XOR      R1,R3      ;FORM THE EXCLUSIVE OR
1725 007706 012700 177400     MOV      #+C<377>,R0 ;BYTE MASK
1726 007712 040001     BIC      R0,R1      ;SAVE LOW BYTE RECV
1727 007714 040002     BIC      R0,R2      ;SAVE LOW BYTE EXPD
1728 007716 040003     BIC      R0,R3      ;SAVE LOW BYTE XOR
1729 007720         PRINTB   #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
1730         007720 010346     MOV      R3,-(SP)
1731         007722 010146     MOV      R1,-(SP)
1732         007724 010246     MOV      R2,-(SP)
1733         007726 012746 007752     MOV      #XORBFOR,-(SP)
1734         007732 012746 000004     MOV      #4,-(SP)
1735         007736 010600     MOV      SP,R0
1736         007740 104414     TRAP    C#PNTB
1737         007742 062706 000012     ADD      #12,SP
1738         007746 010300     MOV      R3,R0      ;R0 HAS XOR ON RETURN
1739         007750 000207     RTS       PC        ;RETURN TO CALLER
1740
1741 007752 045 116 045 XORBFOR: .ASCIZ '#N#A EXPD: #03#A RECV: #03#A XOR: #03'
1742         .EVEN
1743         .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR
1744
1745                                     ;+
1746                                     ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
1747                                     ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
1748                                     ;
1749                                     ;INPUTS:
1750                                     ;
1751                                     ;       R1      RECEIVED DATA
1752                                     ;       R2      EXPECTED DATA
1753                                     ;
1754                                     ;OUTPUT:
1755                                     ;
1756                                     ;       R0      XOR OF EXPECTED/RECEIVED DATA
1757                                     ;-
1758 010020 PRIBXOR::
1759 010020         SAVREG                ;SAVE THE REGISTERS
1760 010024 010203     MOV      R2,R3      ;EXPECTED DATA
1761 010026         XOR      R1,R3      ;FORM THE EXCLUSIVE OR
1762 010036         PRINTB   #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE

```

PRIXOR - PRINT EXPD, RECV AND XOR

010036	010346				MOV	R3,-(SP)	
010040	010146				MOV	R1,-(SP)	
010042	010246				MOV	R2,-(SP)	
010044	012746	010070			MOV	#XORFOR,-(SP)	
010050	012746	000004			MOV	#4,-(SP)	
010054	010600				MOV	SP,R0	
010056	104414				TRAP	C#PNTB	
010060	062706	000012			ADD	#12,SP	
1755	010064	010300			MOV	R3,R0	;R0 HAS XOR ON RETURN
1756	010066	000207			RTS	PC	;RETURN TO CALLER
1757							
1758	010070	045	116	045	XORFOR:	.ASCIZ	'#N#A EXPD: #06#A RECV: #06#A XOR: #06'
1759						.EVEN	

PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT

```

1761                                     .SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
1762
1763                                     ;+
1764                                     ;
1765                                     ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
1766                                     ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
1767                                     ;
1768                                     ;INPUTS:
1769                                     ;
1770                                     ;      RO      OCTAL VALUE TO CONVERT
1771                                     ;      R1      TABLE OF POINTERS TO ASCII EQUIVALENT
1772                                     ;
1773                                     ;-
1774
1775 010136                               PRIEQU:                               ;SAVE THE REGISTERS
1776 010136                               SAVREG                               ;RETURN TO CALLER
1777 010142 000207                       RTS      PC
1778
1779                                     .SBTTL PRIRAM - PRINT RAM ADDRESS
1780
1781                                     ;+
1782                                     ;
1783                                     ;PRINT CONTROLLER RAM ADDRESS.
1784                                     ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
1785                                     ;
1786                                     ;INPUTS:
1787                                     ;
1788                                     ;      R4      RAM ADDRESS
1789                                     ;-
1790
1791 PRIRAM:                               SAVREG                               ;SAVE R1-R5 UNTIL NEXT RETURN
1792 010150 010446                       PRINTB  @RAMFOR,R4                ;PRINT RAM ADDRESS IN ERROR
1793 010152 012746 010174                 MOV      R4,-(SP)
1794 010156 012746 000002                 MOV      @RAMFOR,-(SP)
1795 010162 010600                       MOV      @2,-(SP)
1796 010164 104414                       MOV      SP,R0
1797 010166 062706 000006                 TRAP    C#PNTB
1798 010172 000207                       ADD      #6,SP
1799                                     RTS      PC                               ;RETURN
1800
1801 010174 045 116 045 RAMFOR: .ASCIZ 'N#A CONTROLLER RAM ADDRESS = #06'
1802                                     .EVEN

```



PRIADD - PRINT MEMORY ERROR ADDRESS

```

1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810 010236
1811 010236
1812 010242 013700 002230
1813 010246 013701 002232
1814 010252 010102
1815 010254 006101
1816 010256 C06100
1817 010260
      010260 010246
      010262 010046
      010264 012746 010306
      010270 012746 000003
      010274 010600
      010276 104414
1818 010300 062706 000010
1819 010304 000207
1820 010306 045 116 045
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835 010352
1836 010352
1837 010356 013702 002230
1838 010362 013701 002232
1839
1840
1841
1842 010366
      010366 010146
      010370 012746 010434
      010374 012746 000002
      010400 010600
      010402 104414
    
```

```

.SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
;*
;PRINT MEMORY ADDRESS
;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
;
; IMPLICIT INPUTS
;
; ERRHI - HIGH ORDER ADDRESS
; ERRLO - LOW ORDER ADDRESS
;
;--
PRIADD:
  SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
  MOV ERRHI,R0    ;GET HIGH ADDRESS
  MOV ERRLO,R1    ;GET LOW ADDRESS
  MOV R1,R2       ;COPY LOW ADDRESS
  ROL R1          ;SHIFT BIT 15 TO C BIT
  ROL R0          ;SHIFT INTO HIGH ORDER
  PRINTB #PRIA0,R0,R2 ;PRINT MEMORY ADDRESS IN ERROR
  MOV R2,-(SP)
  MOV R0,-(SP)
  MOV #PRIA0,-(SP)
  MOV #3,-(SP)
  MOV SP,R0
  TRAP C#PNTB
  ADD #10,SP
  RTS PC          ;RETURN
    
```

```

045 PRIA0: .ASCIZ '#N#A MEMORY ERROR ADDRESS = #01#05'
          .EVEN
    
```

```

.SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
;*
;PRINT MEMORY ADDRESS
;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
;
; IMPLICIT INPUTS
;
; ERRHI - HIGH ORDER ADDRESS
; ERRLO - LOW ORDER ADDRESS
;
;--
PRITADD:
  SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
  MOV ERRHI,R2    ;GET HIGH ADDRESS
  MOV ERRLO,R1    ;GET LOW ADDRESS
  ;MOV R1,R2      ;COPY LOW ADDRESS
  ;ROL R1         ;SHIFT BIT 15 TO C BIT
  ;ROL R0         ;SHIFT INTO HIGH ORDER
  PRINTB #PRIT0,R1 ;PRINT MEMORY ADDRESS LOW IN ERROR
  MOV R1,-(SP)
  MOV #PRIT0,-(SP)
  MOV #2,-(SP)
  MOV SP,R0
  TRAP C#PNTB
    
```

PRITADD - PRINT MEMORY TEST ADDRESS

```

1843 010404 062706 000006      ADD    #6,SP
      010410      PRINTB  #PRIT1,R2      ;PRINT MEMORY ADDRESS HIGH IN ERROR
      010410 010246      MOV    R2,-(SP)
      010412 012746 010477      MOV    #PRIT1,-(SP)
      010416 012746 000002      MOV    #2,-(SP)
      010422 010600      MOV    SP,R0
      010424 104414      TRAP  C#PNTB
1844 010426 062706 000006      ADD    #6,SP
      010432 000207      RTS    PC      ;RETURN
1845
1846 010434      045      116      045 PRIT0: .ASCIZ  '#N#A MEMORY TEST ADDRESS LOW = #06'
1847 010477      045      116      045 PRIT1: .ASCIZ  '#N#A MEMORY TEST ADDRESS HIGH = #06'
1848                                     .EVEN
    
```

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

.SBTTL SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900

```

;+
;
;ROUTINE TO ISSUE A SPACE RECORDS
;COMMAND (FORWARD OR REVERSE)
;
;INPUT:
;
;   R3      NUMBER OF RECORDS TO BE SPACED OVER
;           BIT15 CONTROLS DIRECTION
;           BIT15 = 0 IS FORWARD
;           BIT15 = 1 IS REVERSE
;   R5      FIRST DEVICE UNIBUS ADDRESS
;
;   REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
;
;OUTPUT:
;
;   CARRY   SET - SPACE RECORDS COMMAND OK
;           CLR - SPACE RECORDS FAILED
;
;   R0      THE CONTENTS OF R4 IS MOVED TO R0
;
;IMPLICIT OUTPUT:
;
;   TAPE HAS BEEN MOVED
;
;SIDE EFFECTS:
;
;-
SPACE::
    SAVREG
    MOV     #500.,SDELAY
    MOV     #140010,80#
    TST    R3
    BMI    5#
    MOV     R3,90#
    BR     10#
    BIC    #BIT15,R3
    MOV     R3,90#
    BIS    #BIT8,80#
    MOV     #80#,R4
    MOV     R4,TSDB(R5)
    JSR    PC,WAITF
    BCS    20#
    DELAY  250
    MOV     #250,(PC)+
    .WORD  0
    MOV     L#DLY,(PC)+
    .WORD  0
    DEC    -6(PC)
    BNE    .-4
;SAVE THE GENERAL REGISTERS
;SET UP DELAY
;SET UP COMMAND, SPACE FORWARD
;CHECK FOR DIRECTION
;BR, IF REVERSE INDICATED
;LOAD UP NUMBER OF RECORDS TO SPACE
;GO DO COMMAND
;CLEAR DIRECTION BIT
;LOAD UP NUMBER OF RECORDS TO SPACE
;SET REVERSE BIT IN COMMAND PACKET
;SET UP R4 WITH PACKET ADDRESS
;SEND OUT COMMAND
;WAIT FOR SSR
;BR, IF SSR IS SET AND OK
;DELAY ABOUT .25 SECONDS

```

010544				
010544				
010550	012737	000764	010740	
010556	012737	140010	010730	
010564	005703			
010566	100403			
010570	010337	010732		
010574	000407			
010576	042703	100000		5#:
010602	010337	010732		
010606	052737	000400	010730	
010614	012704	010730		10#:
010620	010465	0J0000		
010624	004737	016330		15#:
010630	103420			
010632				
010632	012727	000250		
010636	000000			
010640	013727	002116		
010644	000000			
010646	005367	177772		
010652	001375			

## SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```

010654 005367 177756          DEC      -22(PC)
010660 001367                BNE      .-20
1901 010662 005337 010740     DEC      SDELAY          ;BUMP DELAY COUNTER DOWN
1902 010666 001356                BNE      15#             ;BR, IF MORE DELAY
1903 010670 000411                BR       60#             ;BR IF TROUBLE CARRY = CLEAR
1904 010672 016501 000002     20#:    MOV      TSSR(R5),R1 ;READ TSSR
1905 010676 012702 000200     MOV      #SSR,R2        ;SET UP EXPECTED
1906 010702 020201     25#:    CMP      R2,R1        ;ARE THEY OK
1907 010704 001401                BEQ      40#             ;BR, IF EQUAL = OK
1908 010706 000402                BR       60#             ;TROUBLE EXIT
1909 010710 000261     40#:    SEC                       ;SET CARRY NO TROUBLE
1910 010712 000401                BR       70#             ;EXIT
1911 010714 000241     60#:    CLC                       ;CARRY CLEAR = ERROR
1912 010716                70#:
1913 010716 010400          MOV      R4,R0          ;PASS PACKET ADDRESS
1914 010720 000207          RTS      PC             ;RETURN
1915
1916
1917
1918          ;PACKET FOR SPACE COMMAND
1919
1921          010730          ;
1922          ;
1923          ;
1924          ;COMMAND WORD
1925 010730 000000     80#:    .WORD
1926          ;NUMBER OF RECORDS TO BE SPACED OVER WORD
1927 010732 000000     90#:    .WORD
1928 010734 000000          .WORD
1929 010736 000000          .WORD
1930 010740 000000     SDELAY: .WORD      0          ;DELAY COUNTER
1931          .EVEN
1932          .SBTTL  WRTCHR - WRITE CHARACTERISTICS COMMAND

```

WRTCHR - WRITE CHARACTERISTICS COMMAND

```

1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957 010742
1958 010742
1959 010746 005037 002222
1960 010752 005037 002220
1961 010756 010465 000000
1962 010762 004737 016416
1963 010766 103401
1964 010770 000435
1965 010772 016501 000002
1966 010776 012702 000200
1967 011002 032701 000100
1968 011006 001402
1969 011010 052702 000100
1970 011014 020201
1971 011016 001401
1972 011020 000421
1973 011022 062704 000010
1974 011026 011403
1975 011030 032763 000200 000012
1976 011036 001402
1977 011040 005237 002220
1978 011044
1979 011044 032763 000100 000012
1980 011052 001402
1981 011054 005237 002222
1982 011060
1983 011060 000261
1984 011062 000401
1985 011064 000241
1986 011066 016500 000002
1987 011072 000207

```

```

;
;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
;
;INPUT:
;   R4      ADDRESS OF PACKET FROM TEST
;   R5      FIRST DEVICE UNIBUS ADDRESS
;           REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
;
;OUTPUT:
;   R0      TSSR CONTENTS
;   CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
;           CLR - WRITE CHARACTERISTICS FAILED
;
;IMPLICIT OUTPUT:
;
;   MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
;   SOFTWARE SWITCHES SET AS FOLLOWS:
;   EXTFEA = EXTENDED FEATURES PRESENT
;   BENBSW = BUFFER ENABLE SWITCH ON OR OFF
;
;SIDE EFFECTS:
;-
WRTCHR::
;SAVE THE GENERAL REGISTERS
;CLEAR BUFFER ENABLE SWITCH
;CLEAR EXTENDED FEATURES SW SWITCH
;SEND OUT COMMAND
;WAIT FOR SSR
;BR, IF SSR IS SET AND OK
;BR IF TROUBLE CARRY = CLEAR
;READ TSSR
;SET UP EXPECTED
;WAS OFF LINE SET IN TSSR
;BR, IF NO OFL SET
;MAKE THEM LOOK ALIKE
;ARE THEY OK
;BR, IF EQUAL = OK
;TROUBLE EXIT
;POINT TO WRT CHARA DATA PACKET
;GET ADDRESS OF MESSAGE BUFFER
;EXTENDED FEATURES BIT SET?
;BR IF NO
;SET EXTENDED FEATURES SW SWITCH
;BUFFER ENABLE SWITCH SET
;BR, IF SWITCH NOT SET
;SET SOFTWARE SWITCH FOR ENABLED
;SET CARRY NO TROUBLE
;EXIT
;CARRY CLEAR = ERROR
;RETURN TSSR CONTENTS
;RETURN

```

```

WRTCHR::
    SAVREG
    CLR     BENBSW
    CLR     EXTFEA
10$:  MOV    R4,TSDB(R5)
    JSR    PC,CHKTSSR
    BCS    20$
    BR     60$
20$:  MOV    TSSR(R5),R1
    MOV    #SSR,R2
    BIT    #OFL,R1
    BEQ    25$
    BIS    #OFL,R2
25$:  CMP    R2,R1
    BEQ    40$
    BR     60$
40$:  ADD    #8,R4
    MOV    (R4),R3
    BIT    #X2.EXTF,XST2(R3)
    BEQ    45$
    INC    EXTFEA
45$:  BIT    #X2.BUFE,XST2(R3)
    BEQ    50$
    INC    BENBSW
50$:  SEC
    BR     70$
60$:  CLC
70$:  MOV    TSSR(R5),R0
    RTS    PC

```

REWIND - POSITION TAPE (REWIND) COMMAND

1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016 011074  
2017 011074  
2018 011100 012704 011170  
2019 011104 010465 000000  
2020 011110 012703 000550  
2021 011114 004737 016330  
2022 011120 103417  
2023 011122  
011122 012727 000372  
011126 000000  
011130 013727 002116  
011134 000000  
011136 005367 177772  
011142 001375  
011144 005367 177756  
011150 001367  
2024 011152 005303  
2025 011154 001357  
2026 011156 000241  
2027 011160 010400  
2028 011162 000207  
2029  
2031 011170  
2033 011170  
2034 011170 102010  
2035 011172 000000

```

.SBTTL REWIND - POSITION TAPE (REWIND) COMMAND
;
; THIS ROUTINE WILL REWIND THE SELECTED TAPE.
;
; CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
; TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
; SSR TO SET IN THE TSSR
;
; CALLING SEQUENCE:
;
; DO A SOFT INIT
; DO A WRITE CHARACTERISTICS
; JSR PC,REWIND
;
; INPUT:
;
; R5 FIRST DEVICE UNIBUS ADDRESS
;
; OUTPUT
;
; R0 THE CONTENTS OF R4 IS PASSED TO R0
;
;
; -
REWIND::
; SAVREG
; MOV @RMPACK,R4 ;SAVE R1-R5 UNTIL NEXT RETURN
; MOV R4,TSDB(R5) ;GET PACKET ADDRESS
; MOV @360.,R3 ;SEND PACKET ADDRESS TO EXECUTE
100: ; JSR PC,WAITF ;ENOUGH TIME FOR 2400' REEL TO REWIND
; BCS 200 ;WAIT FOR SSR TO SET
; DELAY 250. ;LEAVE WHEN SSR IS SET
; MOV @250..(PC), ;WAIT FOR .25 SECONDS
; .WORD 0
; MOV L@DLY,(PC),
; .WORD 0
; DEC -6(PC)
; BNE -4
; DEC -22(PC)
; BNE -20
; DEC R3 ;BUMP COUNTER DOWN
; BNE 100 ;KEEP GOING
; CLC ;CLEAR CARRY TO SET ERROR
200: ; MOV R4,R0 ;PASS THE PACKET ADDRESS
; RTS PC ;RETURN
;
; RMPACK: .=<..10>&177770
; .WORD 102010 ;POSTION COMMAND (REWIND)
; .WORD 0 ;NOT USED
    
```

CKRAM - COMPARE RAM TO I/O PACKET

```

2037          .SBTTL  CKRAM  - COMPARE RAM TO I/O PACKET
2038          ;+
2039          ;
2040          ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
2041          ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
2042          ;
2043          ;INPUT:
2044          ;
2045          ;       R4      ADDRESS OF THE COMMAND PACKET
2046          ;       R5      FIRST DEVICE UNIBUS ADDRESS
2047          ;
2048          ;OUTPUT:
2049          ;
2050          ;       CARRY   SET - RAM MATCHES PACKET
2051          ;             CLR - RAM DOES NOT MATCH PACKET
2052          ;
2053          ;IMPLICIT OUTPUT:
2054          ;
2055          ;       THE TABLE RAMDATA IS FILLED WITH THE
2056          ;       DATA HELD IN RAM.
2057          ;       RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
2058          ;
2059          ;SIDE EFFECTS:
2060          ;
2061          ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
2062          ;
2063          ;-
2064
2065          CKRAM::
2066          SAVREG          ;SAVE THE GENERAL REGISTERS
2067          MOV             #RAMDATA,R1      ;ADDRESS TO SAVE THE RAM DATA
2068          MOV             #RMPKTBEGR,R2   ;BYTE ADDRESS OF FIRST RAM DATA
2069          CLR             R3              ;CLEAR THE ERROR FLAG
2070          JSR             PC,CHKTSSR      ;WAIT FOR SSR
2071          MOVB            #0,TSDB(R5)     ;SET MAINTENANCE MODE
2072          JSR             PC,CHKTSSR      ;WAIT FOR SSR TO SET
2073          MOV             R2,TSDB(R5)     ;SELECT NEXT RAM ADDRESS
2074          JSR             PC,CHKTSSR      ;WAIT FOR SSR TO SET
2075          MOVB            TSBA(R5),(R1)   ;READ THE RAM DATA
2076          CMPB            (R1)+,(R4)+    ;COMPARE TO EXPECTED
2077          BEQ             20$            ;BRANCH IF OK
2078          INC             R3              ;SET ERROR FLAG
2079          INC             R2              ;ADDRESS OF NEXT RAM LOCATION
2080          CMP             R2,#RMPKTEND   ;REACHED END YET ?
2081          BLE             10$            ;BRANCH TILL ALL READ
2082          TST             R3              ;WAS AN ERROR FOUND ?
2083          BEQ             30$            ;BRANCH IF NOT
2084          CLC              ;CLEAR CARRY TO SHOW ERROR
2085          BR              50$            ;AND EXIT
2086          SEC              ;SHOW GOOD COMPARE
2087          MOV             #8.,RAMSIZ     ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
2088          RTS             PC            ;RETURN

```

## CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

```

2090          .SBTTL  CKRAM2  - COMPARE RAM TO I/O CHARACTERISTICS DATA
2091          ;*
2092          ;
2093          ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
2094          ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
2095          ;
2096          ;INPUT:
2097          ;
2098          ;       R4      ADDRESS OF THE CHARACTERISTICS DATA
2099          ;       R5      FIRST DEVICE UNIBUS ADDRESS
2100          ;
2101          ;OUTPUT:
2102          ;
2103          ;       CARRY   SET - RAM MATCHES PACKET
2104          ;              CLR - RAM DOES NOT MATCH PACKET
2105          ;
2106          ;IMPLICIT OUTPUT:
2107          ;
2108          ;       THE TABLE RAMDATA IS FILLED WITH THE
2109          ;       DATA HELD IN RAM.
2110          ;       RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
2111          ;
2112          ;SIDE EFFECTS:
2113          ;
2114          ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
2115          ;-
2116 011304  CKRAM2::
2117 011304          SAVREG          ;SAVE THE GENERAL REGISTERS
2118 011310 012701 002234          MOV      @RAMDATA,R1      ;ADDRESS TO SAVE THE RAM DATA
2119 011314 012702 000167          MOV      @RMCHBEG,R2     ;BYTE ADDRESS OF FIRST RAM DATA
2120 011320 005003          CLR      R3              ;CLEAR THE ERROR FLAG
2121 011322 004737 016416          JSR      PC,CHKTSSR      ;WAIT FOR SSR
2122 011326 112765 000000 000000  MOVB   #0,TSDB(R5)     ;SET MAIN. INANCE MODE
2123 011334 004737 016416 10$:   JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
2124 011340 010265 000000          MOV      R2,TSDB(R5)   ;SELECT NEXT RAM ADDRESS
2125 011344 004737 016416          JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
2126 011350 116511 000000          MOVB   TSBA(R5),(R1)   ;READ THE RAM DATA
2127 011354 122124          CMPB   (R1),.(R4)     ;COMPARE TO EXPECTED
2128 011356 001401          BEQ      20$          ;BRANCH IF OK
2129 011360 005203          INC      R3              ;SET ERROR FLAG
2130 011362 005202 20$:   INC      R2              ;ADDRESS OF NEXT RAM LOCATION
2131 011364 012737 000010 002274  MOV     #8.,RAMSIZ     ;ASSUME EXTFEA NOT SET
2132 011372 005737 002220          TST     EXTFEA        ;IS THE SOFTWARE EXTENDED FEATURES SET
2133 011376 001407          BEQ      25$          ;BR, IF NOT SET
2134 011400 012737 000012 002274  MOV     #10.,RAMSIZ   ;SET RAMSIZ FOR EXTEND FEATURES
2135 011406 020227 000200          CMP     R2,@RMCHEND   ;AT END OF EXTENDED BUFFER
2136 011412 003750          BLE     10$          ;BR, IF NOT AT END YET
2137 011414 000403          BR      27$          ;AT END BRANCH
2138 011416 020227 000176 25$:   CMP     R2,@RMCHEND-2 ;REACHED END YET ?
2139 011422 003744          BLE     10$          ;BRANCH TILL ALL READ
2140 011424 005703 27$:   TST     R3              ;WAS AN ERROR FOUND ?
2141 011426 001402          BEQ     30$          ;BRANCH IF NOT
2142 011430 000241          CLC                    ;CLEAR CARRY TO SHOW ERROR
2143 011432 000401          BR      50$          ;AND EXIT
2144 011434 000261 30$:   SEC                    ;SHOW GOOD COMPARE
2145 011436 000207 50$:   RTS      PC          ;RETURN

```



CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

```

2147          .SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS
2148          ;*
2149          ;
2150          ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
2151          ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
2152          ;ERROR PRINT ROUTINES.
2153          ;
2154          ;INPUT:
2155          ;
2156          ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
2157          ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
2158          ;      R2      EXPD MESSAGE BUFFER ADDRESS
2159          ;OUTPUT:
2160          ;
2161          ;      CARRY   SET - MESSAGE BUFFERS MATCH
2162          ;              CLR -MESSAGE BUFFERS DON'T MATCH
2163          ;
2164          ;IMPLICIT OUTPUT:
2165          ;
2166          ;      EXPMSG   BUFFER IS SET TO EXPD DATA
2167          ;      RECVMSG  BUFFER IS SET TO RECV DATA
2168          ;      RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
2169          ;      RCVLOAD  SET TO LOW ORDER ADDRESS OF RECV
2170          ;
2171          ;-
2172          CKMSG::
2173          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
2174          MOV            R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
2175          MOV            R1,RCVLOAD  ;SAVE RECV LOW ADDRESS
2176          TST           KTNABLE     ;TESTING ABOVE 28K?
2177          BEQ           10$         ;BR IF NO
2178          JSR           PC,SETMAP   ;RETURN ADDRESS BIASED TO PAR6 IN R0
2179          MOV            R0,R1      ;GET RETURNED ADDRESS BIASED TO PAR6
2180          10$:          CLR            R4      ;WORD IN BUFFER
2181          CLR            R3         ;CLEAR ERROR SEEN FLAG
2182          MOV            R2,R5      ;GET EXPD BUFFER ADDRESS
2183          15$:          MOV            (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
2184          MOV            (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
2185          CMP            (R2),.(R1) ;EXPD EQUAL RECV?
2186          BEQ           25$         ;BR IF YES
2187          INC            R3         ;SET ERROR SEEN FLAG
2188          25$:          ADD            #2,R4   ;POINT TO NEXT WORD ADDRESS
2189          CMP            R4,#14     ;DONE FIRST 7 WORDS?
2190          BLE           15$         ;BR IF NO
2191          000012:      BIT            #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
2192          BEQ           50$         ;BR IF NO
2193          CMP            R4,#16     ;DONE EXTENDED FEATURES WORD?
2194          BLE           15$         ;BR IF NO
2195          50$:          TST            R3      ;ANY ERRORS SEEN?
2196          BEQ           55$         ;BR IF NO
2197          CLC            ;SET FAILURE
2198          BR            60$         ;
2199          55$:          SEC            ;SET SUCCESS
2200          60$:          RTS           PC      ;RETURN
    
```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2202 .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
2203
2204 ;*
2205 ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
2206 ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
2207 ;ERROR PRINT ROUTINES.
2208
2209 ;INPUT:
2210 ; R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
2211 ; R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
2212 ; R2 EXPD MESSAGE BUFFER ADDRESS
2213 ; R3 NUMBER OF BYTES TO COMPARE
2214
2215 ;OUTPUT:
2216 ; CARRY SET - MESSAGE BUFFERS MATCH
2217 ; CLR - MESSAGE BUFFERS DON'T MATCH
2218
2219 ;IMPLICIT OUTPUT:
2220 ; EXPMSG BUFFER IS SET TO EXPD DATA
2221 ; RECVMSG BUFFER IS SET TO RECV DATA
2222 ; RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
2223 ; RCVLOAD SET TO LOW ORDER ADDRESS OF RECV
2224
2224 011560 CKMSG2::
2225 011560 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2226 011564 020327 000144 CMP R3,#RECVMSG-EXPMSG;@@D IS COUNT ABOVE MAX ALLOWED?
2227 011570 003412 BLE 5# ;@@D BR IF NO
2228 011572 012703 000144 MOV #RECVMSG-EXPMSG,R3;@@D
2229 011576 PRINTF #DEBUGMSG ;@@D
011576 012746 011712 MOV #DEBUGMSG,-(SP)
011602 012746 000001 MOV #1,-(SP)
011606 010600 MOV SP,R0
011610 104417 TRAP C#PNTF
011612 062706 000004 ADD #4,SP
2230 011616 010037 002276 5# : MOV R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
2231 011622 010137 002300 MOV R1,RCVLOAD ;SAVE RECV LOW ADDRESS
2232 011626 005737 003126 TST KTENABLE ;TESTING ABOVE 28K?
2233 011632 001403 BEQ 10# ;BR IF NO
2234 011634 004737 017376 JSR PC,SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN R0
2235 011640 010001 MOV R0,R1 ;GET RETURNED ADDRESS BIASED TO PAR6
2236 011642 005004 10# : CLR R4 ;WORD IN BUFFER
2237 011644 005005 CLR R5 ;CLEAR ERROR SEEN FLAG
2238 011646 111264 002314 15# : MOVB (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
2239 011652 111164 002460 MOVB (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
2240 011656 122221 CMPB (R2)+,(R1)+ ;EXPD EQUAL RECV?
2241 011660 001401 BEQ 25# ;BR IF YES
2242 011662 005205 INC R5 ;SET ERROR SEEN FLAG
2243 011664 062704 000001 25# : ADD #1,R4 ;POINT TO NEXT BYTE
2244 011670 020403 CMP R4,R3 ;DONE ALL BYTES?
2245 011672 002001 BGE 50# ;BR IF YES
2246 011674 000764 BR 15# ;DO NEXT BYTE
2247 011676 005705 50# : TST R5 ;ANY ERRORS SEEN?
2248 011700 001402 BEQ 55# ;BR IF NO
2249 011702 000241 CLC ;SET FAILURE
2250 011704 000401 BR 60# ;
2251 011706 000261 55# : SEC ;SET SUCCESS
2252 011710 000207 60# : RTS PC ;RETURN

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2254 011712      120      122      117  DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';800
2255 012002      045      116      045  FERCM:  .ASCII /N#A ***/
2256 012013      040      040      124  ERCH:   .ASCIZ / TSSR ERROR CODE REC'D = /
2257 012046      056      056      056  SIMSG:  .ASCIZ /... AFTER DOING SOFT INIT/
2258 012101      124      105      123  TINERR: .ASCIZ /TEST: .../
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274 012114      104423
2275 012114      004737   006020
2276 012120      004737   017262
2277 012124
2278 012124      104423
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290 012126
2291 012126      004737   006020
2292 012132      012700   000004
2293 012136      004737   007446
2294 012142
2295 012142      104423

```

```

;+
;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
;INPUT:
;      R1      CONTENTS OF TSSR AT ERROR
;SIDE EFFECTS:
;      EXECUTES DROP UNIT TO CEASE TESTING
;-

      BGNMSG   SFIMSG
SFIMSG: JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
        JSR     PC,CKDROP     ;DROP UNIT, IF ALLOWED
        ENDMSG

L10003: TRAP    C#MSG

;+
;PRINT ROUTINE TO PRINT THE CONTENTS OF
;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
;INPUTS:
;      R1      TSSR CONTENTS
;      R4      ADDRESS OF COMMAND PACKET
;-

      BGNMSG   PKTSSR
PKTSSR: JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
        MOV     #4,R0          ;NO. OF WORDS IN PACKET
        JSR     PC,PRIPKT     ;PRINT THE CONTENTS OF COMMAND PACKET
        ENDMSG

L10004: TRAP    C#MSG

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2296
2297 ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2298 ;TSSR AND A GET STATUS COMMAND PACKET.
2299
2300 ;INPUTS:
2301 ;
2302 ; R1 TSSR CONTENTS
2303 ; R4 ADDRESS OF COMMAND PACKET
2304 ;-
2305 012144 BGNMSG PKTGETS
012144
2306 012144 004737 006020 PKTGETS::
2307 012150 012700 000002 JSR PC,PRITSSR ;PRINT THE CONTENTS OF TSSR REGISTER
2308 012154 004737 007446 MOV #2,R0 ;NO. OF WORDS IN GET STATUS PACKET
2309 012160 JSR PC,PRIPKT ;PRINT THE CONTENTS OF COMMAND PACKET
012160 ENDMSG
012160 104423 L10005:
TRAP C#MSG

2310 ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
2311 ;
2312 ;INPUTS:
2313 ; R1 TSSR CONTENTS
2314 ; R4 ADDRESS OF COMMAND PACKET
2315 ;-
2316 012162 BGNMSG SFFMSG
2317 012162 SFFMSG::
012162 JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR REGISTER
2318 012162 004737 006020 ENDMSG
2319 012166 L10006:
012166 TRAP C#MSG
012166 .SBTTL PKTMES - PRINT TSSR AND MESSAGE BUFFER

2320 ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
2321 ;BUFFER FOR ERROR REPORTS
2322 ;
2323 ;INPUTS:
2324 ;
2325 ; R1 CONTENTS OF TSSR
2326 ; R2 LOW ORDER MESSAGE BUFFER
2327 ; R3 HIGH ORDER MESSAGE BUFFER ADDRESS
2328 ; NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
2329 ;-
2330 012170 BGNMSG PKTMES
2331 012170 PKTMES::
2332 012170 004737 006020 JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR
2333 012174 010200 MOV R2,R0 ;LOW ORDER ADDRESS
2334 012176 010301 MOV R3,R1 ;HIGH ORDER ADDRESS
2335 012200 004737 014322 JSR PC,PRMESS ;PRINT THE MESSAGE BUFFER
2336 012204 ENDMSG
2337 012204 L10007:
012204 TRAP C#MSG
012204 104423
    
```

ADDSSR - PRINT TEST ADDRESS AND TSSR

```

2339          .SBTTL  ADDSSR  - PRINT TEST ADDRESS AND TSSR
2340          ;+
2341          ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2342          ;TSSR AND A MEMORY TEST ADDRESS
2343          ;
2344          ;INPUTS:
2345          ;
2346          ;      R5      FIRST DEVICE UNIBUS ADDRESS
2347          ;      ERRHI   HIGH ORDER MEMORY TEST ADDRESS
2348          ;      ERRLO   LOW ORDER MEMORY TEST ADDRESS
2349          ;-
2350
2351          BGNMSG  ADDSSR
2352          ADDSSR::
2353          JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
2354          MOV     TSSR(R5),R1    ;GET CURRENT TSSR
2355          JSR     PC,PRITSSR     ;PRINT THE CONTENTS OF TSSR REGISTER
2356          ENDMSG
2357          L10010:
2358          TRAP    C#MSG
2359
2360          .SBTTL  MSGEXP  - PRINT WRITE CHAR. EXPD-RECV MESSAGE BUFFERS
2361          ;+
2362          ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
2363          ;
2364          ;IMPLICIT INPUTS:
2365          ;
2366          ;      EXPMSG  - EXPECTED MESSAGE BUFFER
2367          ;      RECMG   - RECEIVED MESSAGE BUFFER
2368          ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2369          ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2370          ;-
2371          BGNMSG  MSGEXP
2372          MSGEXP::
2373          MOV     #7,R0           ;ASSUME NO EXT FEATURES
2374          TST     EXTFEA         ;EXT FEATURES SET?
2375          BEQ     5$             ;BR IF NO
2376          MOV     #8.,R0        ;EXT FEATURE BUFFER IS 8 WORDS
2377          JSR     PC,PRMSGEXP    ;PRINT EXPD/RECV MESSAGE BUFFERS
2378          ENDMSG
2379          L10011:
2380          TRAP    C#MSG

```

FIFEXP - PRINT FIFO EXP/RCV DATA

```

2377 .SBTTL FIFEXP - PRINT FIFO EXP/RCV DATA
2378 ;*
2379 ;PRINT ROUTINE TO PRINT FIFO EXP/RCV DATA
2380 ;
2381 ; R1 - BYTE COUNT
2382 ;
2383 ;IMPLICIT INPUTS:
2384 ;
2385 ; EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
2386 ; RECVMSG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
2387 ;
2388 ;-
2389 012250 BGNMSG FIFEXP
2390 FIFEXP::
2390 012250 PRINTX #FIF1MSG,R1 ;PRINT BYTES TRANSFERRED
2391 012250 MOV R1,-(SP)
2392 012252 010146 012746 012322 MOV #FIF1MSG,-(SP)
2393 012256 012746 000002 MOV #2,-(SP)
2394 012262 C10600 MOV SP,R0
2395 012264 104415 TRAP C#PNTX
2396 012266 062706 000006 ADD #6,SP
2397 012272 PRINTX #FIF2MSG ;PRINT HEADER MSG
2398 012272 012746 012371 MOV #FIF2MSG,-(SP)
2399 012276 012746 000001 MOV #1,-(SP)
2400 012302 010600 MOV SP,R0
2401 012304 104415 TRAP C#PNTX
2402 012306 062706 000004 ADD #4,SP
2403 012312 010100 MOV R1,R0 ;GET BYTE COUNT
2404 012314 004737 015202 JSR PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
2405 012320 ENDMSG
2406 012320 L10012:
2407 012320 104423 TRAP C#MSG
2408 012322 045 116 045 FIF1MSG: .ASCIZ '#N#A NUMBER OF BYTES TRANSFERRED = #D2'
2409 012371 045 116 045 FIF2MSG: .ASCIZ '#N#A FIFO DATA BYTES IN ERROR:'
2410 .EVEN

```

MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS

```

2399          .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
2400          ;*
2401          ;
2402          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
2403          ;
2404          ;
2405          ;IMPLICIT INPUTS:
2406          ;
2407          ;   EXPMSG - EXPECTED MESSAGE BUFFER
2408          ;   RECHMSG - RECEIVED MESSAGE BUFFER
2409          ;   RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2410          ;   RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2411          ;-
2412          BGNMSG MSGSTAT
MSGSTAT:
2413          MOV     #STATCOD,R1      ;ASCII ADDRESS TABLE
2414          MOV     (R1)+,R0         ;DONE ALL MSG LINES?
2415          BEQ     20$,             ;BR IF YES
2416          PRINTX R0                ;PRINT STATUS BIT NAMES
          MOV     R0,-(SP)
          MOV     #1,-(SP)
          MOV     SP,R0
          TRAP   C#PNTX
          ADD     #4,SP
2417          BR     10$,             ;DO ANOTHER MSG LINE
2418          MOV     #10,R0          ;NUMBER OF WORDS IN A READ STATUS BUFFER
2419          JSR     PC,PRMSGEXP     ;PRINT EXPD/RCV MESSAGE BUFFERS
2420          ENDMMSG
L10013:
          TRAP   C#MSG
2421
2422          .WORD 1$,2$,3$,4$,5$,6$,0
2423          1$: .ASCIZ 'N/A Tape Bus Signals in Word #8:'
2424          2$: .ASCIZ 'N/A PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
2425          3$: .ASCIZ 'N/A IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
2426          4$: .ASCIZ 'N/A IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
2427          5$: .ASCIZ 'N/A Tape Bus Signals in Word #9:'
2428          6$: .ASCIZ 'N/A DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
2429          .EVEN
2430

```

MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

```

2432          .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
2433
2434          ;
2435          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
2436          ;
2437          ;IMPLICIT INPUTS:
2438          ;
2439          ;   EXPMSG - EXPECTED MESSAGE BUFFER
2440          ;   RECMMSG - RECEIVED MESSAGE BUFFER
2441          ;   RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2442          ;   RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2443          ;
2444          BGNMSG MSGLOOP
2445          MSGLOOP:
2446          10:  MOV     #LOOPCOD,R1      ;ASCII ADDRESS TABLE
2447          BEQ     (R1)+,R0           ;DONE ALL MSG LINES?
2448          PRINTX R0                  ;BR IF YES
2449          MOV     R0,-(SP)           ;PRINT STATUS BIT NAMES
2450          MOV     #1,-(SP)
2451          MOV     SP,R0
2452          TRAP   C#PNTX
2453          ADD     #4,SP
2454          BR     10:                 ;DO ANOTHER MSG LINE
2455          20:  MOV     #10,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
2456          JSR   PC,PRMSGEXP         ;PRINT EXPD/RECV MESSAGE BUFFERS
2457          ENDMMSG
2458          L10014:
2459          TRAP   C#MSG
2460
2461          LOOPCOD: .WORD 1#,2#,3#,4#,5#,6#,7#,0
2462          1#: .ASCIZ 'N#A Tape Bus Loopback Signals in Word #8:'
2463          2#: .ASCIZ 'N#A PARERR<15> IRESV2<14> IRESV1<13>'
2464          3#: .ASCIZ 'N#A IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
2465          4#: .ASCIZ 'N#A IWM =>IFMK<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
2466          5#: .ASCIZ 'N#A ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDOP <04>'
2467          6#: .ASCIZ 'N#A IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
2468          7#: .ASCIZ 'N#A IGO =>IFPT<00>'
2469          .EVEN

```



MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```

2464 .SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
2465 ;*
2466 ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
2467 ;
2468 ;IMPLICIT INPUTS:
2469 ;
2470 ;
2471 ;
2472 ;   EXPMSG - EXPECTED MESSAGE BUFFER
2473 ;   RECMG  - RECEIVED MESSAGE BUFFER
2474 ;   RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2475 ;   RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2476 ;-
2477 014022 BGNMSG MSGSUB
      014022 MSGSUB::
2478 014022 012700 000012      MOV     #10,,R0      ;SIZE OF WRITE SUBSYSTEM BUFFER
2479 014026 004737 014632      JSR     PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
2480 014032      ENDMSG
      014032 L10015:
      014032 104423      TRAP   C#MSG

2481 .SBTTL MEMADD - PRINT MEMORY ADDRESS DATA ERROR
2482 ;*
2483 ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
2484 ;
2485 ;IMPLICIT INPUTS:
2486 ;
2487 ;
2488 ;
2489 ;   ERRHI  - MEMORY ERROR HIGH ORDER ADDRESS
2490 ;   ERRLO  - MEMORY ERROR LOW ORDER ADDRESS
2491 ;   EXP    - EXPECTED DATA
2492 ;   RECV   - RECEIVED DATA
2493 ;-
2494 014034 BGNMSG MEMADD
      014034 MEMADD::
2495 014034 004737 010236      JSR     PC,PRIADD   ;PRINT MEMORY ADDRESS IN ERROR
2496 014040 013701 002224      MOV     EXPD,R1    ;GET EXPD DATA
2497 014044 013702 002226      MOV     RECV,R2   ;GET RECEIVED DATA
2498 014050 004737 010020      JSR     PC,PRIXOR  ;PRINT EXPD/RCV
2499 014054      ENDMSG
      014054 L10016:
      014054 104423      TRAP   C#MSG

```

PRAMPKT - PRINT RAM AND PACKET DATA

```

2501                                     .SBTTL PRAMPKT - PRINT RAM AND PACKET DATA
2502                                     ;*
2503                                     ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
2504                                     ;WHEN THE RAM DATA DOES NOT MATCH.
2505                                     ;
2506                                     ;INPUTS:
2507                                     ;
2508                                     ;       R4      POINTER TO COMMAND PACKET
2509                                     ;IMPLICIT INPUTS:
2510                                     ;       RAMDATA  DATA AS READ FROM THE RAM
2511                                     ;       RAMSIZ   NUMBER OF BYTES IN PACKET
2512                                     ;                                     IF RAMSIZ=0 THEN DEFAULT TO 8.
2513                                     ;
2514                                     ;IMPLICIT OUTPUTS:
2515                                     ;       RAMSIZ  SET TO 0
2516                                     ;-
2517 014056 PRAMPKT:
2518 014056 SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
2519 014062 C12701 002234 MOV #RAMDATA,R1          ;DATA FROM THE RAM
2520 014066 005002 CLR R2                                ;INIT BYTE NUMBER
2521 014070 122124 5#: CMPB (R1)+,(R4)+        ;COMPARE EXPECTED, RECEIVED
2522 014072 001005 BNE 7#                          ;BR IF NO MATCH
2523 014074 000436 FORCERROR 7#,NOTSSR
2524 014104 000436 BR 10#                          ;BND
2525 014106 116105 177777 7#: MOVB -1(R1),R5      ;GET RECV RAM DATA
2526 014112 116403 177777 MOVB -1(R4),R3      ;GET EXPD PACKET DATA
2527 014116 XOR R5,R3                          ;XOR EXPD/RECV
2528 014126 042703 177400 BIC #177400,R3      ;LOW BYTE ONLY
2529 014132 116137 177777 002226 MOVB -1(R1),RECV    ;GET RECEIVED RAM DATA
2530 014140 116437 177777 002224 MOVB -1(R4),EXPD    ;GET EXPECTED RAM DATA
2531 014146 PRINTB #RAMASC,R2,RECV,EXPD,R5
2532 014146 010346 MOV R3,-(SP)
2533 014150 013746 002224 MOV EXPD,-(SP)
2534 014154 013746 002226 MOV RECV,-(SP)
2535 014160 010246 MOV R2,-(SP)
2536 014162 012746 014236 MOV #RAMASC,-(SP)
2537 014166 012746 000005 MOV #5,-(SP)
2538 014172 010600 MOV SP,R0
2539 014174 104414 TRAP C#PNTB
2540 014176 062706 000014 ADD #14,SP
2541 014202 005202 10#: INC R2                    ;UPDATE BYTE COUNT
2542 014204 005737 002274 TST RAMSIZ            ;DEFAULT TO 8.?
2543 014210 001404 BEQ 15#                      ;BR IF YES
2544 014212 020237 002274 CMP R2,RAMSIZ        ;DONE ALL BYTES?
2545 014216 003724 BLE 5#                          ;BR IF NO
2546 014220 000403 BR 25#                          ;
2547 014222 020227 000010 15#: CMP R2,#8.        ;DONE DEFAULT NUMBER OF BYTES?
2548 014226 002720 20#: BLT 5#                      ;BR IF NO
2549 014230 005037 002274 25#: CLR RAMSIZ        ;SET DEFAULT RAMSIZ
2550 014234 000207 RTS PC                          ;RETURN
2551 014236 045 116 045 RAMASC: .ASCIZ '##A BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03#
2552                                     .EVEN

```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

2546 .SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
2547
2548 ; THIS ROUTINE PRINTS THE CONTENTS OF
2549 ; THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE TSV-05.
2550
2551 ; INPUT:
2552 ; R0 LOW ORDER ADDRESS OF MESSAGE BUFFER
2553 ; R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER
2554 ; NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
2555 ; THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
2556
2557 PRMESS: SAVREG ;SAVE THE REGISTERS
2558 MOV RO,R5 ;SAVE LOW ORDER ADDRESS
2559 TST KTENABLE ;ADDRESS ABOVE 28K?
2560 BNE 100 ;BR IF YES
2561 CLR R1 ;SET HIGH ORDER ADDRESS TO 0
2562 100: MOV R1,R3 ;SAVE HIGH ORDER ADDRESS
2563 ROL RO ;SHIFT BIT15 TO C BIT
2564 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
2565 PRINTX @PROASC,R1,R5 ;PRINT MESSAGE BUFFER ADDRESS
2566 MOV R5,-(SP)
2567 MOV R1,-(SP)
2568 MOV @PROASC,-(SP)
2569 MOV @3,-(SP)
2570 MOV SP,RO
2571 TRAP C:PNTX
2572 ADD @10,SP
2573 PRINTX @PRIASC ;PRINT HEADER FOR CONTENTS
2574 MOV @PRIASC,-(SP)
2575 MOV @1,-(SP)
2576 MOV SP,RO
2577 TRAP C:PNTX
2578 ADD @4,SP
2579 CLR R4 ;NUMBER OF THE NEXT WORD
2580 MOV R5,R1 ;COPY LOW ORDER ADDRESS
2581 MOV R3,RO ;COPY HIGH ORDER ADDRESS
2582 BEQ 200 ;BR IF NOT ABOVE 28K
2583 JSR PC,SETHAP ;SETUP PAR ADDRESS IN RO
2584 MOV RO,R5 ;GET PAR FORMAT ADDRESS ABOVE 28K
2585 200: PRINTX @PRASC,R4,(R5) ;PRINT THE CONTENTS OF MEMORY BUFFER
2586 MOV (R5),-(SP)
2587 MOV R4,-(SP)
2588 MOV @PRASC,-(SP)
2589 MOV @3,-(SP)
2590 MOV SP,RO
2591 TRAP C:PNTX
2592 ADD @10,SP
2593 INC R4 ;NUMBER OF THE NEXT
2594 CMP R4,@7 ;DONE ALL YET ?
2595 BGT 500 ;BRANCH IF ALL DONE
2596 BLT 200 ;PRINT FIRST 7 WORDS
2597 BIT @X2.EXTF,XST2(R3) ;EXTENDED FEATUTES ON ?
2598 BNE 200 ;PRINT EXTENDED STATUS WORD
2599 500: RTS PC ;RETURN
2600 045 116 045 PROASC: .ASCIZ '###A Message Buffer Address = #01#05'
2601 045 116 045 PRIASC: .ASCIZ '###A Message Buffer Contents:'
2602 045 116 045 PRASC: .ASCIZ '###A Word#D1#A: #0'
    
```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

2585 .EVEN
2586 .SBTTL PRMSGEXP - PRINT EXPD/RCV MESSAGE BUFFERS
2587
2588 ;+
2589 ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
2590 ; RO - NUMBER OF WORDS IN BUFFER
2591 ;IMPLICIT INPUTS:
2592 ; EXPMSG - EXPECTED MESSAGE BUFFER
2593 ; RECMMSG - RECEIVED MESSAGE BUFFER
2594 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2595 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2596 PRMSGEXP::
2597 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2598 MOV RO,R5 ;SAVE NUMBER OF WORDS
2599 MOV RCVLOADD,RO ;GET RECV LOW ADDRESS
2600 MOV RO,R4 ;COPY LOW ADDRESS
2601 MOV RCVHIADD,R1 ;GET RECV HIGH ADDRESS
2602 ROL RO ;SHIFT BIT15 TO C BIT
2603 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
2604 PRINTX #PRMSG0,R1,R4 ;PRINT MESSAGE BUFFER ADDRESS
      014632
      014632 010005
      014636 013700 002300
      014644 010004
      014646 013701 002276
      014652 006100
      014654 C06101
      014656
      014656 010446
      014660 010146
      014662 012746 015012
      014666 012746 000003
      014672 010600
      014674 104415
      014676 062706 000010
2605 PRINTX #PRMSG1 ;PRINT HEADER FOR CONTENTS
      014702 012746 015057
      014706 012746 000001
      014712 010600
      014714 104415
      014716 062706 000004
2606 CLR R4 ;NUMBER OF THE CURRENT WORD
2607 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
2608 MOV #RECMMSG,R2 ;GET RECV BUFFER ADDRESS
2609 MOV (R1),R0 ;GET EXPD
2610 MOV (R2),R3 ;GET RECV
2611 XOR R0,R3 ;XOR EXPD/RCV
2612 PRINTX #PRMSG2,R4,(R1)+,(R2)+,R3
      014750 010346
      014752 012246
      014754 012146
      014756 010446
      014760 012746 015115
      014764 012746 000005
      014770 010600
      014772 104415
      014774 062706 000014
2613 INC R4 ;NUMBER OF THE NEXT
2614 CMP R4,R5 ;DONE ALL YET?
2615 BGE 50$ ;BR IF YES
2616 BR 20$ ;DO ANOTHER
2617 RTS PC ;RETURN
2618 045 116 045 PRMSG0: .ASCIZ '#N#A Message Buffer Address = #01#05'
2619 045 116 045 PRMSG1: .ASCIZ '#N#A Message Buffer Contents:'
2620 045 116 045 PRMSG2: .ASCIZ '#N#A WORD #D2#A EXPD: #06#A RECV: #06#A XOR: #06#A'

```

PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS

```

2622 .EVEN
2623 .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
2624 ;*
2625 ;
2626 ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
2627 ; ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
2628 ;
2629 ; R0 - NUMBER OF BYTES IN BUFFER
2630 ;
2631 ;IMPLICIT INPUTS:
2632 ;
2633 ; EXPMSG - EXPECTED MESSAGE BUFFER
2634 ; RECMMSG - RECEIVED MESSAGE BUFFER
2635 ;-
2636 015202 PRBYTEXP::
2637 015202 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2638 015206 010005 MOV R0,R5 ;SAVE NUMBER OF BYTES
2639 015210 005037 002312 CLR PRMNO ;INIT ERROR COUNT
2640 015214 C05004 CLR R4 ;NUMBER OF THE CURRENT BYTE
2641 015216 012701 002314 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
2642 015222 012702 002460 MOV #RECMMSG,R2 ;GET RECV BUFFER ADDRESS
2643 015226 111100 20#: MOV (R1),R0 ;GET EXPD BYTE
2644 015230 042700 177400 BIC #C<377>,R0 ;CLEAR UPPER BYTE
2645 015234 110037 015550 MOV R0,PRBEXP ;SAVE FOR ERROR REPORT
2646 015240 111203 MOV (R2),R3 ;GET RECV BYTE
2647 015242 042703 177400 BIC #C<377>,R3 ;CLEAR UPPER BYTE
2648 015246 110337 015552 MOV R3,PRBREC ;FOR ERROR REPORT
2649 015252 XOR R0,R3 ;XOR EXPD/RECV
2650 015262 122122 CMPB (R1)+,(R2)+ ;EXPD = RECV?
2651 015264 001431 BEQ 30# ;BR IF YES
2652 015266 005237 002312 INC PRMNO ;UPDATE ERROR COUNT
2653 015272 023727 002312 000010 CMP PRMNO,#8 ;PRINTED 8?
2654 015300 101023 BHI 30# ;BR IF YES
2655 015302 27#: PRINTX #PRBMSG,R4,PRBEXP,PRBREC,R3
015302 010346 MOV R3,-(SP)
015304 013746 015552 MOV PRBREC,-(SP)
015310 013746 015550 MOV PRBEXP,-(SP)
015314 010446 MOV R4,-(SP)
015316 012746 015416 MOV #PRBMSG,-(SP)
015322 012746 000005 MOV #5,-(SP)
015326 010600 MOV SP,R0
015330 104415 TRAP C#PNTX
015332 062706 000014 ADD #14,SP
2656 015336 FORCEEXIT 50# ;@@D
2657 015346 000404 BR 35# ;@@D
2658 015350 30#:
2659 015350 FORCERROR 27#,NOTSSR ;@@D
2660 015360 35#: ;@@D
2661 015360 005204 INC R4 ;NUMBER OF THE NEXT
2662 015362 020405 CMP R4,R5 ;DONE ALL YET?
2663 015364 002001 BGE 50# ;BR IF YES
2664 015366 000717 BR 20# ;DO ANOTHER
2665 015370 50#: PRINTX #PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
015370 013746 002312 MOV PRMNO,-(SP)
015374 012746 015503 MOV #PRBTOT,-(SP)
015400 012746 000002 MOV #2,-(SP)
015404 010600 MOV SP,R0
    
```

PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

015406 104415
015410 062706 000006
2666 015414 000207 TRAP C#PNTX
2667 ADD #6,SP
2668 015416 045 116 045 PRBMSG: .ASCIZ 'N#A BYTE #D2#A EXPD: #03#A RECV: #03#A XOR: #03#
2669 015503 045 116 045 PRBTOT: .ASCIZ 'N#A NUMBER OF BYTES IN ERROR = #D2'
2670 .EVEN
2671 015550 000000 PRBEXP: .WORD 0 ;EXPD
2672 015552 000000 PRBREC: .WORD 0 ;RECV
2673 .SBTTL EXPREC - PRINT EXPD/RECV WORD DATA
2674 ;+
2675 ;
2676 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2677 ;
2678 ;INPUTS:
2679 ;
2680 ; R1 RECEIVED DATA
2681 ; R2 EXPECTED DATA
2682 ;
2683 ;-
2684
2685 015554 BGNMSG EXPREC
015554 EXPREC:: JSR PC,PRIXOR ;PRINT THE DATA
2686 015554 004737 010020 ENDMSG
2687 015560 L10017:
015560 104423 TRAP C#MSG
015560 .SBTTL EXPBREC - PRINT EXPD/RECV BYTE DATA
2688 ;+
2689 ;
2690 ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
2691 ;
2692 ;INPUTS:
2693 ;
2694 ; R1 RECEIVED DATA BYTE
2695 ; R2 EXPECTED DATA BYTE
2696 ;
2697 ;-
2698
2699
2700
2701 015562 BGNMSG EXPBREC
015562 EXPBREC:: JSR PC,PRIBXOR ;PRINT THE DATA
2702 015562 004737 007670 ENDMSG
2703 015566 L10020:
015566 104423 TRAP C#MSG
2704 .SBTTL RAMERR - PRINT RAM AND PACKET DATA
2705 ;+
2706 ;
2707 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
2708 ;
2709 ;INPUTS:
2710 ;
2711 ; R4 POINTER TO COMMAND PACKET
2712 ;
2713 ;
2714 ;

```

RAMERR - PRINT RAM AND PACKET DATA

```

2715 ;IMPLICIT INPUTS:
2716 ;
2717 ;   RAMDATA      DATA AS READ FROM THE RAM
2718 ;   RAMSIZ      NUMBER OF BYTES IN PACKET
2719 ;               IF RAMSIZ=0 THEN DEFAULT TO 8.
2720 ;
2721 ;IMPLICIT OUTPUTS:
2722 ;
2723 ;   RAMSIZ SET TO 0
2724 ;-
2725 ;
2726 015570   BGNMSG  RAMERR
      015570
2727 015570 004737 014056  RAMERR:: JSR   PC,PRAMPKT      ;PRINT RAM/PACKET DATA
2728 015574   ENDMSG
      015574 104423  L10021: TRAP  C#MSG
2729 ;               .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
2730 ;
2731 ;+
2732 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
2733 ;
2734 ;INPUTS:
2735 ;
2736 ;   R4          POINTER TO COMMAND PACKET
2737 ;
2738 ;IMPLICIT INPUTS:
2739 ;
2740 ;   RAMDATA      DATA AS READ FROM THE RAM
2741 ;   RAMSIZ      NUMBER OF BYTES IN PACKET
2742 ;               IF RAMSIZ=0 THEN DEFAULT TO 8.
2743 ;   ERRHI       HIGH ORDER TEST ADDRESS
2744 ;   ERRLO       LOW ORDER TEST ADDRESS
2745 ;
2746 ;IMPLICIT OUTPUTS:
2747 ;
2748 ;   RAMSIZ SET TO 0
2749 ;-
2750 ;
2751 ;
2752 015576   BGNMSG  RAMTADD
      015576
2753 015576 004737 010352  RAMTADD:: JSR   PC,PRITADD      ;PRINT TEST ADDRESS
2754 015602 004737 014056  JSR   PC,PRAMPKT      ;PRINT RAM/PACKET DATA
2755 015606   ENDMSG
      015606 104423  L10022: TRAP  C#MSG
2756 ;               .SBTTL  RAMEXP - PRINT RAM EXPD/RECV DATA
2757 ;
2758 ;+
2759 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2760 ;
2761 ;INPUTS:
2762 ;
2763 ;   R1          RECEIVED DATA
2764 ;   R2          EXPECTED DATA
2765 ;

```

RAMEXP - PRINT RAM EXPD/RECV DATA

```

2766          ;      R4      CONTROLLER RAM ADDRESS
2767          ; -
2768
2769 015610      BGNMSG  RAMEXP
015610
2770 015610 042701 177400  RAMEXP:: BIC    #+C<377>,R1      ;SAVE EXPD RAM DATA BYTE
2771 015614 042702 177400      BIC    #+C<377>,R2      ;SAVE EXPD RAM DATA BYTE
2772 015620 004737 010144      JSR    PC,PRIRAM      ;PRINT THE RAM ADDRESS
2773 015624 004737 010020      JSR    PC,PRIXOR      ;PRINT THE DATA
2774 015630      ENDMSG
015630
015630 104423
L10023:
2775          TRAP    C#MSG
2776          .SBTTL  TIMEXP - PRINT TIMER A,B AND EXP/REC
2777          ;+
2778          ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2779          ;AND TIMER A,B HEADER MESSAGE
2780          ;
2781          ;INPUTS:
2782          ;
2783          ;      R1      RECEIVED DATA
2784          ;      R2      EXPECTED DATA
2785          ;
2786          ; -
2787
2788          BGNMSG  TIMEXP
015632
015632
2789 015632      TIMEXP:: PRINTX  #TIMSGO          ;PRINT HEADER
015632 012746 015660      MOV    #TIMSGO, -(SP)
015636 012746 000001      MOV    #1, -(SP)
015642 010600      MOV    SP,R0
015644 104415      TRAP    C#PNTX
015646 062706 000004      ADD    #4,SP
2790 015652 004737 010020      JSR    PC,PRIXOR      ;PRINT THE DATA
2791 015656      ENDMSG
015656
015656 104423
L10024:
2792          TRAP    C#MSG
2793 015660      045      116      045 TIMSGO: .ASCIZ  '#N#A TIMER A STATUS IS IN BIT 3#N#A TIMER B STATUS IS IN BIT 2'
2794          .EVEN
2795          .SBTTL  BADSSR - PRINT TSSR ERRORS C#N DATA TRANSFERS
2796          ;+
2797          ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
2798          ;
2799          ;INPUTS:
2800          ;
2801          ;      R1      CONTENTS OF TSSR
2802          ;      R2      DATA WRITTEN (8 BITS)
2803          ;
2804          ; -
2805
2806          BGNMSG  BADSSR
2807
2808          BADSSR::
015760
015760
2809 015760 010246      MOV    R2, -(SP)          ;SAVE DATA TRANSFERRED
2810 015762 042702 177400  BIC    #177400,R2      ;GET JUST ONE BYTE

```



BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

2811	015766				PRINTB	0XFERASC,R2	
	015766	010246			MOV	R2,-(SP)	
	015770	012746	016020		MOV	0XFERASC,-(SP)	
	015774	012746	000002		MOV	02,-(SP)	
	016000	010600			MOV	SP,R0	
	016002	104414			TRAP	C#PNTB	
	016004	062706	000006		ADD	06,SP	
2812	016010	012602			MOV	(SP)+,R2	;RESTORE R2
2813	016012	004737	006020		JSR	PC,PRITSSR	;DECODE TSSR CONTENTS
2814	016016				ENDMSG		
	016016			L10025:			
	016016	104423			TRAP	C#MSG	
2815	016020	045	116	045	XFERASC:	.ASCIZ	'#N#A Data Transferred = #03'

GLOBAL SUBROUTINES SECTION

```

2817      .SBTTL GLOBAL SUBROUTINES SECTION
2818
2819      ;*+
2820      ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2821      ; THAT ARE USED IN MORE THAN ONE TEST.
2822      ;--
2823      .SBTTL SOFINIT - SOFT INITIALIZE OF CONTROLLER
2824
2825      ;*
2826      ;
2827      ;ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
2828      ;BY WRITING INTO THE TSSR REGISTER, AFTER THE INIT,
2829      ;THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
2830      ;DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
2831      ;
2832      ;INPUTS:
2833      ;
2834      ;      R5      ADDRESS OF FIRST REGISTER
2835      ;
2836      ;OUTPUTS:
2837      ;
2838      ;      R0      CONTENTS OF TSSR, IF ERROR
2839      ;      CARRY   SET IF INIT WAS OKAY
2840      ;              CLEAR IF FATAL ERROR
2841      ;
2842      ;CALLING SEQUENCE:
2843      ;
2844      ;      MOV     #ADDRESS,R5
2845      ;      JSR     PC,SOFINIT
2846      ;      BCS    CONTINUE
2847      ;      ERRDF          ;REPORT FATAL ERROR
2848      ;
2849      ;-
2850
2851 016054 SOFINIT::
2852 016054      SAVREG          ; SAVE THE REGISTERS
2853 016060      MOV     #0,TSSR(R5) ; DO THE INIT.
2854 016066      JSR     PC,WAITF   ; WAIT FOR SSR
2855 016072      MOV     TSSR(R5),R0 ;GET THE TSSR REGISTER
2856 016076      MOV     R0,R4      ;TSSR CONTENTS
2857 016100      BIC     #C<HIADDR!OFL>,R4
2858 016104      BIS     #SSR!NBA,R4 ;R4 HAS EXPECTED CONTENTS
2859 016110      CMP     R4,R0      ;ONLY EXPECTED BITS SET ?
2860 016112      BEQ     5$        ;BRANCH IF OKAY
2861 016114      CLC          ;CLEAR THE CARRY FOR ERROR
2862 016116      BR     10$       ;GO TO EXIT
2863 016120      5$: SEC        ;SET THE CARRY BIT
2864 016122      10$: RTS     PC   ;RETURN TO CALLER

```

CHKAMB - CHECK TSSR FOR AMBIGUITY

```

2866 .SBTTL CHKAMB - CHECK TSSR FOR AMBIGUITY
2867
2868
2869
2870 ; THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
2871 ; FOR AMBIGUITY
2872
2873 ; INPUT:
2874
2875 ;     RO     CONTENTS OF TSSR
2876
2877 ; OUTPUT:
2878
2879 ;     RO     CONTENTS OF TSSR
2880
2881 ;     CARRY  SET - NO AMBIGUITY
2882 ;           CLR - AMBIGUOUS CONTENTS
2883
2884 ; -
2885
2886 CHKAMB:
2887     SAVREG                ; SAVE THE GENERAL REGISTERS
2888     MOV     RO,R4         ; CONTENTS OF TSSR
2889     BIT     @SC,RO        ; IS BIT 15 SET ?
2890     BNE    5$            ; BRANCH IF YES
2891     BIT     @+C<NBA!OFL!SSR!HIADDR>,RO ; ANY OTHER BITS SET ?
2892     BNE    40$          ; MUST BE AN ERROR
2893     BR     45$          ; RETURN WITH SUCCESS
2894     BIT     @SSR,RO      ; IS READY BIT SET ?
2895     BNE    10$         ; BRANCH IF READY BIT IS SET.
2896     BIT     @BIT5,RO    ; IS FATAL ERROR BIT SET ?
2897     BEQ    40$         ; ERROR IF NOT
2898     BIC    @+CTERCLS,R4 ; CLEAR ALL BUT TERMINATION CODE
2899     CMP    R4,@16      ; ALL THREE BITS MUST BE SET
2900     BNE    40$         ; ERROR IF NOT SET
2901     BR     45$         ; OK IF ALL ARE SET
2902     BIT     @BIT5,RO    ; IS FATAL ERROR BIT SET ?
2903     BEQ    45$         ; ERROR IF BIT IS SET WITH SSR
2904     BIT     @BIT2!BIT1,RO ; IS THIS A FUNCTION REJECT
2905     BNE    45$         ; BR, IF TSSR IS OK
2906     CLC    40$        ; AMBIGUOUS CONTENTS
2907     BR     50$
2908     SEC    45$
2909     RTS   50$         ; SHOW SUCCESS - NO AMBIGUITY
                        ; RETURN TO CALLER
2886 016124
2887 016124
2888 016130 010004
2889 016132 032700 100000
2890 016136 001004
2891 016140 032700 174077
2892 016144 001023
2893 016146 000424
2894 016150 032700 000200
2895 016154 001011
2896 016156 032700 000040
2897 016162 001414
2898 016164 042704 177761
2899 016170 020427 000016
2900 016174 001007
2901 016176 000410
2902 016200 032700 000040
2903 016204 001405
2904 016206 032700 000006
2905 016212 001002
2906 016214 000241
2907 016216 000401
2908 016220 000261
2909 016222 000207

```

## ENAIN,DSBINT - ENABLE/DISABLE INTERRUPTS

```

2911      .SBTTL ENAIN,DSBINT - ENABLE/DISABLE INTERRUPTS
2912      ;
2913      ; DEFAULT DISPLAY INTERRUPT HANDLERS.
2914      ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2915      ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
2916      ;
2917      ;
2918      ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2919      ;
2920      IOKCKIN=BIT7      ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
2921      IOKSTP=BIT0      ; EXPECT "STOP" INTERRUPT.
2922      ;
2923      ; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
2924      INTMASK: .BYTE 0
2925      ; INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2926      INTFLAG: .BYTE 0
2927      ;
2928      ; SAVED INTERRUPT VECTOR:
2929      INTVEC: .WORD 0
2930      ; SAVE CPU PC
2931      INTCPC: .WORD 0
2932      ;
2933      ; SUBROUTINE TO ENABLE INTERRUPTS:
2934      ENAIN: MOV     RO, -(SP)      ; SAVE RO
2935             MOV     IVEC, RO      ; GET POINTER TO VECTORS
2936             MOV     @INTR, (RO)+  ; SET UP INTERRUPT VECTOR
2937             MOV     @PRI07, (RO)+
2938             MOV     (SP)+, RO      ; RESTORE RO
2939             MOV     (SP), -(SP)
2940             MOV     @0,2(SP)      ; SET CPU TO LEVEL 0
2941             RTI
2942      ;
2943      ; SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2944      DSBINT: MOV     (SP), -(SP)
2945             MOV     @PRI07,2(SP)
2946             RTI
2947      .SBTTL INTR - INTERRUPT HANDLERS
2948      ;
2949      BGNSRV INTR      ; DEFINE INTERRUPT ENTRY
2950      INTR:: MOV     @1, INTRECV    ; SET FLAG TO SHOW INTERRUPT RECEIVED
2951             CLRB   INTFLAG      ; CLEAR FLAG TO SAY WE GOT INTERRUPT
2952             BITB   @IOKSTP, INTMASK ; EXPECTING STOP INTERRUPT?
2953             BNE    1$           ; BR IF YES
2954             BISB   @IOKSTP, INTFLAG ; NO. SET THE ERROR FLAG.
2955      ;
2956      ; SAVE REGISTERS, MSG BUFFER, ETC.
2957      1$: ENDSRV
2958      L10026: RTI

```

WAITF - WAIT FOR SUBSYSTEM READY

```

2960 .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
2961 ;
2962 ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2963 ;
2964 ; INPUTS:
2965 ;
2966 ; R5 ADDRESS OF FIRST DEVICE REGISTER
2967 ;
2968 ; OUTPUTS:
2969 ;
2970 ; R0 CONTENTS OF LAST TSSR READ
2971 ; CARRY SET - READY BIT SET
2972 ; CLR - TIMEOUT WAITING FOR READY
2973 ;
2974 016330 000401
2975 016332
016332 104422
2976 016334 012746 011000
2977 016340 C16500 000002
2978 016344 105700
2979
2980 016346 100420
2981 016350
016350 012727 000001
016354 000000
016356 013727 002116
016362 000000
016364 005367 177772
016370 001375
016372 005367 177756
016376 001367
2982 016400 005316
2983 016402 001356
2984 016404 000241
2985 016406 000401
2986 016410 000261
2987 016412 005326
2988 016414 000207

; WAITF:: BR 1# ;NOP WHEN SUPER FIXED
; BREAK ; DO A SUPVSR BREAK FIRST.
; TRAP C#BRK
; 1#: MOV #11000,-(SP) ;25-APRIL-83 REV B - 1100 MSEC TIMER
; 2#: MOV TSSR(R5),R0 ;READ THE TSSR REGISTER
; TSTB R0 ;TEST FOR READY BIT SET

; BMI 3# ; EXIT ON STOP FLAG.
; DELAY 1 ; WAIT 100 USEC
; MOV #1,(PC)+
; .WORD 0
; MOV L#DLY,(PC)+
; .WORD 0
; DEC -6(PC)
; BNE -.4
; DEC -22(PC)
; BNE -.20
; DEC (SP) ;REDUCE DELAY COUNT
; BNE 2# ;RETRY UNTIL TIMER EXPIRES
; CLC ; C = 0, CONTROLLER STILL RUNNING...
; BR 4# ;...OR HUNG-UP AFTER 300 MSEC.
; 3#: SEC ; C = 1, CONTROLLER IS STOPPED.
; 4#: DEC (SP);
; RTS PC ;RESTORE STACK WITHOUT CHANGING CARRY BIT
    
```

CHKTSSR - CHECK TSSR FOR READY

```

2990 .SBTTL  CHK TSSR - CHECK TSSR FOR READY
2991 ;*
2992 ;THIS ROUTINE WAITS FOR READY IN THE TSSR
2993 ;AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
2994 ;
2995 ;INPUT:
2996 ;      R5      ADDRESS OF CSR REGISTERS
2997 ;
2998 ;OUTPUT:
2999 ;      R0      CONTENTS OF TSSR
3000 ;      CARRY   SET - OKAY
3001 ;             CLR - NOT READY AMBIGUOUS, OR SC SET
3002 ;-
3003 CHK TSSR:
3004 JSR  PC, WAITF      ;WAIT FOR READY
3005 BCC  20$           ;BRANCH IF TIME OUT
3006 JSR  PC, CHKAMB   ;TSSR AMBIGUOUS?
3007 BCC  10$           ;BR IF YES
3008 BIT  #SC, R0      ;SPECIAL CONDITION SET?
3009 BEQ  15$           ;BR IF NO
3010 BIT  #<SCE!BIE!RMR!NXM>, R0 ;ANY ERROR BITS SET?
3011 BEQ  15$           ;BR IF NO
3012 10$: CLC          ;SET FAILURE
3013 BR   20$           ;
3014 15$: SEC          ;SET SUCCESS
3015 20$: RTS  PC      ;RETURN TO CALLER
3016 .SBTTL  NXNM - CHECK FOR NONEXISTENT MEMORY
3017 ;*
3018 ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
3019 ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
3020 ; "C" = 0, ALL ADDRESSES OK.
3021 ;
3022 ;CALL:  MOV  ADR1, R1
3023 ;       MOV  ADR2, R2
3024 ;       JSR  PC, NXNM
3025 ;       RETURN      ;TEST "C" AND PROCEED.
3026 NXNM:  MOV  #2$, #4      ; SET BUSERR VECTOR.
3027 MOV  #PRIO4, #6
3028 CLR  R3              ;FLAG.
3029 1$:  TST  (R1)        ;TEST THE ADDRESS(ES).
3030 ;IF ANY TRAP, CONTINUE AT 2$.
3031 CMP  R1, R2         ;OTHERWISE, CONTINUE HERE.
3032 BEQ  3$            ;BR IF FINISHED (NO NEXM'S).
3033 ADD  #2, R1        ;SET NEXT ADDRESS...
3034 BR   1$           ;...AND CONTINUE.
3035 2$:  COM  R3          ;GOT ONE, SET FLAG...
3036 MOV  #3$, (SP)
3037 RTI                ;...AND DISMISS INTERRUPT...
3038 3$:  CLRVEC #4      ;...AND GIVE BACK THE VECTOR.
3039 MOV  #4, R0
3040 TRAP C#CVEC
3041 TST  R3              ;DID WE CATCH ONE ??
3042 BEQ  .+4           ;NO, "C" = 0, SKIP NEXT.
3043 SEC                ;YES, "C" = 1, (R1) = NEXM ADDR.
3044 RTS  PC

```

TSTLOOP - CHECK ITERATION COUNT

```

3044
3045
3046
3047
3048
3049
3050
3051
3052 016536
3053 016536 005737 002162
3054 016542 001006
3055 016544 005737 002176
3056 016550 100403
3057 016552 005337 002210
3058 016556 001002
3059 016560 000241
3060 016562 000401
3061 016564 000261
3062 016566 C00207
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090 016570
3091 016570 010046
3092 016572 005037 003146
3093 016576 005037 017036
3094 016602 005037 005766
3095 016606 105037 016224
3096 016612 013700 002174
3097 016616 006300
3098 016620 005737 003106
3099 016624 001430
3100 016626 10001C
    
```

```

.SBTTL TSTLOOP - CHECK ITERATION COUNT
;*
; SUBROUTINE TO EXECUTE TEST ITERATIONS.
; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
;
; CALL: LOOPTO ARG
;
TSTLOOP::
    TST     NOITS      ; ITERATIONS INHIBITED?
    BNE     1#        ; YES.
    TST     QVP        ; NO.
    BMI     1#        ; LOOPS DISALLOWED IN QUICK PASS.
    DEC     LOOPCNT   ; BUMP LOOP COUNTER.
    BNE     2#
1#: CLC          ; LOOP DISALLOWED, OR DONE.
    BR     3#
2#: SEC          ; LOOP ENABLED.
3#: RTS     PC

.SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
;*
; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
; IN THE CURRENT RUN SEQUENCE.
; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
;
; INPUT:
;
;     R0     POINTER TO TEST ID ASCIZ STRING
;
; OUTPUT:
;
;     R5     ADDRESS OF FIRST DEVICE REGISTER
;
; IMPLICIT OUTPUTS:
;
;     TSTCNT UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
;
; SIDE EFFECTS:
;
;     INTERRUPT LEVEL IS RASIED TO LEVEL OF
;     THE DEVICE UNDER TEST
;
;-
TSTSETUP::
    MOV     R0, -(SP) ; SAVE THE TEST ID MESSAGE
    CLR     SIFLAG    ; CLEAR "SOFT INIT" FLAG
    CLR     ERRK      ; CLEAR LOCAL ERROR COUNTER.
    CLR     EXTA      ; CLEAR ERROR EXTENSION FLAG.
    CLR     INTMASK   ; CLEAR INTERRUPT MASK (CHECK ERROR)
    MOV     UNITN, R0 ; GET THE UNIT NUMBER,
    ASL     R0        ; ... AND MAKE IT A WORD OFFSET.
    TST     NODEV     ; DID STARTUP FIND THE DEVICE?
    BEQ     4#        ; BR IF YES
    BPL     3#        ; BR IF NOT IDLE
    
```

TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```

3101 016630 052760 160000 003170      BIS      @160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
3102 016636      ERRDF    1,NXR,NXRERR ; NO DEVICE HERE -- PRINT IT
      016636 104455      TRAP    C@ERDF
      016640 000001      .WORD   1
      016642 003734      .WORD   NXR
      016644 005732      .WORD   NXRERR
3103 016646 000407      BR      2$
3104 016650 052760 160001 003170 3$:  BIS      @160001,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
3105 016656      ERRDF    2,NOINIT ; DEVICE NOT IDLE
      016656 104455      TRAP    C@ERDF
      016660 000002      .WORD   2
      016662 004331      .WORD   NOINIT
      016664 000000      .WORD   0
3106 016666 012737 177777 003104 2$:  MOV      @-1,DUFLG ; DROP THE UNIT
3107 016674      DODU     UNITN
      016674 015700 002174      MOV      UNITN,R0
      016700 104451      TRAP    C@DODU
3108 016702      DOCLN   ; ABORT THE PASS
      016702 104444      TRAP    C@DOCLN
3109 016704 000423      BR      5$
3110
3111 016706      RFLAGS   R0 ; GET THE OPERATOR FLAGS.
      016706 104421      TRAP    C@RFLA
3112 016710 032700 001000      BIT      @PNT,R0 ; PRINT THE TEST NUMBERS?
3113 016714 001412      BEQ     1$ ; BR IF NO
3114 016716 011600      MOV      (SP),R0 ; GET THE ID MESSAGE
3115 016720      PRINTF   @TNAM,R0 ; DISPLAY THE TEST ID
      016720 010046      MOV      R0,-(SP)
      016722 012746 016764      MOV      @TNAM,-(SP)
      016726 012746 000002      MOV      @2,-(SP)
      016732 010600      MOV      SP,R0
      016734 104417      TRAP    C@PRINTF
      016736 062706 000006      ADD     @6,SP
3116 016742 005237 002206      INC     TSTCNT ; BUMP TEST COUNTER.
3117 016746      SETPRI   IPRI ; PRIORITY THAT OF DEVICE
      016746 015700 002204      MOV      IPRI,R0
      016752 104441      TRAP    C@SPRI
3118 016754 005726 5$:  TST      (SP)+ ; FIX UP THE STACK
3119 016756 015705 002200      MOV      CSRADDR,R5 ; ADDRESS OF TSV REGISTERS ON UNIBUS
3120 016762 000207      RTS     PC
3121 016764 045 123 045 TNAM: .ASCIZ 'TSV Test'
3122
3123
3124
3125 ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
3126 ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
3127
3128 TSTEND: RFLAGS   R0
      017000      TRAP    C@RFLA
3129 017002 030027 020000      BIT      R0,@IER
3130 017006 001412      BEQ     1$ ; BR IF "IER" NOT SET.
3131 017010      PRINTF   @ESUM,ERRK ; PRINT ERROR COUNT.
      017010 015746 017036      MOV      ERRK,-(SP)
      017014 012746 017040      MOV      @ESUM,-(SP)
      017020 012746 000002      MOV      @2,-(SP)
      017024 010600      MOV      SP,R0
      017026 104417      TRAP    C@PRINTF
    
```



TSTEND - PRINT ERRORS RECEIVED

```
3132 017030 062706 000006      ADD  #6,SP
3133 017034 000207      RTS   PC
3134 017036 000000      ERRK: 0 ; LOCAL ERROR COUNT.
3135 017040      045    101   040  ESUM: .ASCIZ /%A %D%A ERRORS/
3136 017057      105    122   122  EMAXDU: .ASCIZ /ERROR LIMIT REACHED -- DROPPING UNIT/
3137                                     .EVEN
```

INCERK - INCREMENT LOCAL ERROR COUNT

```

3139                                     .SBTTL  INCERK  - INCREMENT LOCAL ERROR COUNT
3140                                     ;*
3141                                     ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
3142                                     ;-
3143 017124 005237 017036 INCERK: INC  ERRK          ; INCREMENT LOCAL ERROR COUNT
3144 017130 010046      MOV  RO,-(SP)      ; SAVE RO
3145 017132 013700      MOV  UNITN,RO     ; GET UNIT NUMBER,
3146 017136 006300      ASL  RO              ; ... AND MAKE IT A WORD OFFSET.
3147 017140 062700      ADD  @ERTABL,RO    ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
3148 017144 005210      INC  (RO)          ; INCREMENT THE DEVICE ERROR COUNT
3149 017146 032710      BIT  @7777,(RO)     ; DID WE OVERFLOW THE FIELD?
3150 017152 001001      BNE  1#              ; BR IF NO.
3151 017154 005310      DEC  (RO)          ; YES -- BACK IT UP TO 7777.
3152 017156 012600      1#: MOV  (SP)+,RO    ; RESTORE RO
3153 017160 000207      RTS   PC              ; RETURN TO CALLER.
3154
3155 017162 010046      CKEMAX: MOV  RO,-(SP)      ; SAVE RO
3156 017164 013700      MOV  UNITN,RO     ; GET UNIT NUMBER
3157 017170 006300      ASL  RO              ; ... AND MAKE IT A WORD OFFSET
3158 017172 016000      MOV  ERTABL(RO),RO  ; GET ERROR TABLE ENTRY
3159 017176 042700      BIC  @170000,RO    ; EXTRACT ERROR COUNT FIELD
3160 017202 020037      CMP  RO,GERRMAX    ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
3161 017206 103004      BHIS 1#              ; BR IF YES
3162 017210 023737      CMP  ERRK,LERRMAX  ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
3163 017216 103417      BLO  2#              ; BR IF NO
3164 017220      1#: RFLAGS RO          ; GET OPERATOR FLAGS
3165 017222 032700      TRAP C#RFLA
3166 017226 001013      BIT  @IDU,RO      ; IS DROPPING INHIBITED?
3167 017230 012737      BNE  2#              ; BR IF YES.
3168 017236 177777      MOV  @-1,DUFLG    ; NO -- DROP THE UNIT
3169 017246 000000      ERRDF 4,EMAXDU
3170 017254 104451      TRAP C#ERDF
3171 017256 012600      .WORD 4
3172 017260 000207      .WORD EMAXDU
3173      .WORD 0
3174      DODU UNITN
3175      MOV  UNITN,RO
3176      TRAP C#DODU
3177      DOCLN
3178      TRAP C#DCLN
3179      2#: MOV  (SP)+,RO    ; RESTORE RO
3180      RTS   PC              ; RETURN TO CALLER

```

CKDROP - CHECK IF UNIT SHOULD BE DROPPED

```

3174          .SBTTL CKDROP - CHECK IF UNIT SHOULD BE DROPPED
3175
3176          ; CHECK IF UNIT SHOULD BE DROPPED
3177          ;
3178 017262 010046          CKDROP: MOV     RO,-(SP)
3179 017264          FORCERROR 1$,NOTSSR
3180 017274          RFLAGS RO
3181 017276 104421          TRAP   C#RFLA
3182 017302 032700 000040  BIT     #IDU,RO
3183 017304 001010          BNE    1$
3184 017306 011600          MOV     (SP),RO
3185 017314 012737 177777 003104  MOV     #-1,DUFLG
3186 017314 013700 002174          DODU   UNITN
3187 017324 104451          MOV     UNITN,RO
3188 017326 C00207          TRAP   C#DODU
3189
3190          DOCLN          ;ABORT THE PASS
3191          TRAP   C#DCLN
3192          MOV     (SP)+,RO
3193          RTS     PC
3194
3195          .SBTTL CONFIG - DETERMINE CONFIGURATION OF SYSTEM
3196          ;
3197          ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
3198          ;
3199          CONFIG:
3200          JSR     PC,SOFINIT
3201          RTS     PC
3202          .SBTTL KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
3203          ;
3204          ; SUBROUTINE - ENABLE MEM MGT.
3205          ;
3206          KTON: TST     KTFLG          ; GOT KT?
3207          BEQ     1$                ; NO.
3208          MOV     #1,SRO            ; YES. ENABLE KT11.
3209          RTS     PC
3210
3211          ; SUBROUTINE - DISABLE MEM MGT.
3212          ;
3213          KTOFF: TST     KTFLG          ; GOT KT11?
3214          BEQ     1$                ; NO.
3215          NOP
3216          MOV     #0,SRO            ; DISABLE KT.
3217          RTS     PC

```

SETMAP - SETUP PAR6 MAPPING

```

3217 .SBTTL SETMAP - SETUP PAR6 MAPPING
3218
3219 ;*
3220 ; THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
3221 ; AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
3222 ; IS RETURNED BIASED TO PAR6.
3223
3224 ; INPUTS:
3225
3226 ; R0 HIGH ORDER ADDRESS BITS
3227 ; R1 LOW ORDER ADDRESS BITS
3228
3229 ; OUTPUTS:
3230
3231 ; R0 OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
3232 ; CARRY SET IF SUCCESS
3233 ; CLR IF ERROR
3234
3235 ;-
3236 SETMAP: SAVREG ;SAVE R1-R4 UNTIL NEXT RETURN
3237 017376 TST KTF LG ;SYSTEM HAVE ABOVE 28K?
3238 017402 005737 003124 BEQ 10$ ;BR IF NO
3239 017406 001433 MOV R1,R2 ;SAVE LOW ORDER BITS
3240 017410 010102 .REPT 6
3241 000006 ASR R0 ;CONVERT WORD ADDRESS TO 32W BLOCKS
3242 ROR R1 ;MAKE IT DOUBLE PRECISION
3243 .ENDR
3244 BIC #177,R1 ;ALINE FOR LOWER 4K BOUNDARY
3245 017442 042701 000177 CMP R1,#6000 ;HIGHER THAN EXISTING MEMORY?
3246 017446 020127 006000 BHIS 10$ ;BR IF YES
3247 017452 103011 MOV R1,#KIPAR5 ;SETUP MAPPING REGISTER PAR5
3248 017454 010137 172352 BIC #160000,R2 ;SETUP DISPLACEMENT IN PAGE
3249 017460 042702 160000 ADD #120000,R2 ;ADD IN PAR5 BIAS
3250 017464 062702 120000 MOV R2,R0 ;RETURN IN R0
3251 017470 010200 SEC ;SET SUCCESS
3252 017472 000261 BR 15$
3253 017474 000401 10$: CLC ;SET FAILURE
3254 017476 000241 15$: RTS PC ;RETURN
3255 017500 000207 .SBTTL FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
3256
3257 ;*
3258 ; FILL MEMORY WITH A BACKGROUND PATTERN
3259
3260 ; INPUTS:
3261
3262 ; R0 = BACKGROUND PATTERN
3263 ; FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
3264 ; KTF LG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
3265
3266 ; OUTPUTS:
3267
3268 ; NONE
3269
3270 ;-
3271 FILLMEM: SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
3272 017502 JSR PC,KTOFF ;DISABLE KT.
3273 017506 004737 017354

```

FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN

3274	017512	010003				MOV	R0,R3	;COPY TEST PATTERN
3275	017514	013701	003116			MOV	FREE,R1	;GET FIRST FREE LOCATION
3276	017520	013702	003120			MOV	FRESIZ,R2	;SIZE OF FREE SPACE BELOW 28K.
3277	017524	010321		10#:		MOV	R3,(R1)+	;STORE A BACKGROUND WORD
3278	017526	005302				DEC	R2	;DONE ALL MEMORY IN FREE SPACE?
3279	017530	003375				BGT	10#	;BR IF NO
3280	017532	005737	003124			TST	KTFLG	; GOT KT?
3281	017536	001477				BEQ	55#	; NO. GET OUT.
3282	017540	004737	017336			JSR	PC,KTON	; YES. ENABLE KT.
3283	017544	005000				CLR	R0	;HIGH ORDER ADDRESS START
3284	017546	013701	003144			MOV	PST32W,R1	;GET >28K START ADDRESS (IN 32W BLOCKS)
3285		000006				.REPT	6	
3286						CLC		;CLEAR C BIT
3287						ROL	R1	;CONVERT BLOCKS TO WORDS
3288						ROL	R0	;MAKE IT DOUBLE PRECISION
3289						.ENDR		
3290	017616	004737	017376			JSR	PC,SETMAP	;SETUP PAR6 MAPPING REGISTER
3291	017622	010320		30#:		MOV	R3,(R0)+	;STORE TEST PATTERN IN >28K ADDRESS
3292	017624	C20027	140000			MOV	R0,#140000	;END OF PAR5 MAPPING AREA?
3293	017630	103774				BLO	30#	;BR IF NO
3294	017632	162700	020000			SUB	#20000,R0	;BACKUP INTO PAR5 MAPPING BEGIN
3295	017636	062737	000200	172352		ADD	#200,#KIPAR5	;POINT TO NEXT 4K BLOCK >28K.
3296	017644	023727	172352	006000		CMP	#KIPAR5,#6000	;END OF MEMORY?
3297	017652	001427				BEQ	50#	;BR IF YES
3298	017654	005737	003136			TST	T23A	;11/23A?
3299	017660	001407				BEQ	35#	;NO KEEP GOING
3300	017662	013704	177572			MOV	SRO,R4	;GET SRO CONTENTS
3301	017666	042704	177761			BIC	#177761,R4	;CLEAR ALL BUT PAGE NUMBER
3302	017672	022704	000016			CMP	#16,R4	;SEE IF PAGE 7
3303	017676	001415				BEQ	50#	;EXIT IF THERE
3304	017700	005737	003140		35#:	TST	T23B	;11/23B?
3305	017704	001410				BEQ	45#	;NO KEEP GOING
3306	017706	023727	172352	007600		CMP	#KIPAR5,#7600	;REACHED 18 BITS?
3307	017714	103001				BHIS	40#	;YES
3308	017716	000403				BR	45#	;NO KEEP GOING
3309	017720	012737	000020	172516	40#:	MOV	#20,SR3	;SET 22 BIT RELOCATION
3310	017726	000137	017622		45#:	JMP	30#	;KEEP GOING ON ETC.
3311	017732	004737	017354		50#:	JSR	PC,KTOFF	; DISABLE KT.
3312	017736	000207			55#:	RTS	PC	

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```

3314 .SBTTL CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
3315 ;
3316 ; COMPARE MEMORY WITH A BACKGROUND PATTERN
3317 ;
3318 ; INPUTS:
3319 ;
3320 ; RO = BACKGROUND PATTERN
3321 ; FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
3322 ; KTF LG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
3323 ;
3324 ; OUTPUTS:
3325 ;
3326 ; CARRY - SET IF NO ERROR
3327 ; CARRY - CLR IF ERROR
3328 ;
3329 ; IMPLICIT OUTPUTS:
3330 ;
3331 ; ERRHI - ERROR HIGH ADDRESS
3332 ; ERRLO - ERROR LOW ADDRESS
3333 ; EXPD - EXPECTED DATA
3334 ; RECV - RECEIVED DATA
3335 ;
3336 ; CMPMEM:
3336 017740 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
3337 017740 MOV RO,R3 ;COPY TEST PATTERN
3338 017744 010003 JSR PC,KTOFF ;DISABLE KT.
3339 017746 004737 017354 MOV FREE,R1 ;GET FIRST FREE LOCATION
3340 017752 013701 003116 MOV FRESIZ,R2 ;SIZE OF FREE SPACE BELOW 28K.
3341 017756 013702 003120 MOV R3,(R1) ;FREE SPACE LOCATION EQUAL TO EXPD?
3342 017762 020311 10#: CMP R3,(R1)
3343 017764 001411 BEQ 15# ;BR IF YES
3344 017766 010137 002232 MOV R1,ERRLO ;SAVE ADDRESS IN ERROR
3345 017772 005037 002230 CLR ERRHI ;NO HIGH ADDRESS
3346 017776 010337 002224 MOV R3,EXPD ;SAVE EXPD FOR ERROR REPORT
3347 020002 011137 002226 MOV (R1),RECV ;SAVE RECV FOR ERROR REPORT
3348 020006 000474 BR 50# ;
3349 020010 005721 15#: TST (R1)+ ;POINT TO NEXT ADDRESS
3350 020012 005302 DEC R2 ;DONE ALL MEMORY IN FREE SPACE?
3351 020014 003362 BGT 10# ;BR IF NO
3352 020016 005737 003124 TST KTF LG ; GOT KT?
3353 020022 001472 BEQ 55# ; NO. GET OUT.
3354 020024 004737 017336 JSR PC,KTON ; YES. ENABLE KT.
3355 020030 005000 CLR RO ;HIGH ORDER ADDRESS START
3356 020032 013701 003144 MOV PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
3357 000006 .REPT 6
3358 ROL R1 ;CONVERT BLOCKS TO WORDS
3359 ROL RO ;MAKE IT DOUBLE PRECISION
3360 .ENDR
3361 020066 042701 000177 BIC #177,R1 ;ALINE 4K BOUNDARY
3362 020072 010046 MOV RO,-(SP) ;SAVE HIGH ORDER
3363 020074 010146 MOV R1,-(SP) ;SAVE LOW ORDER
3364 020076 004737 017376 JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
3365 020102 010004 MOV RO,R4 ;COPY ADDRESS BIASED TO PAR6
3366 020104 012601 MOV (SP)+,R1 ;RESTORE LOW ORDER IN NON PAR6 FORMAT
3367 020106 012600 MOV (SP)+,RO ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
3368 020110 020314 30#: CMP R3,(R4) ;ABOVE 28K LOCATION EQUAL EXPD?
3369 020112 001411 BEQ 32# ;BR IF YES
3370 020114 010037 002230 MOV RO,ERRHI ;SAVE HIGH ORDER IN ERROR
    
```

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```

3371 020120 010137 002232      MOV      R1,ERRLO      ;SAVE LOW ORDER IN ERROR
3372 020124 010337 002224      MOV      R3,EXPD      ;SAVE EXPD FOR ERROR REPORT
3373 020130 011437 002226      MOV      (R4),RECV    ;SAVE RECV FOR ERROR REPORT
3374 020134 000421              BR       50$          ;
3375 020136 062701 000002      32$:    ADD      #2,R1      ;UPDATE NON PAR6 ADDRESS
3376 020142 005500              ADC      R0           ;MAKE IT DOUBLE PRECISION ADD
3377 020144 062704 000002      ADD      #2,R4        ;UPDATE PAR FORMAT ADDRESS
3378 020150 020427 140000      CMP      R4,#140000   ;END OF PAR5 MAPPING AREA?
3379 020154 103755              BLO     30$          ;BR IF NO
3380 020156 162704 020000      SUB      #20000,R4    ;BACKUP INTO PAR6 MAPPING BEGIN
3381 020162 062737 000200      ADD      #200,#KIPAR5 ;POINT TO NEXT 4K BLOCK >28K.
3382 020170 023737 172352      CMP      #KIPAR5,KTFLG ;END OF MEMORY?
3383 020176 101744              BLOS    30$          ;BR IF NO
3384 020200 004737 017354      50$:    JSR      PC,KTOFF    ;TURN OFF MEMORY MAPPING
3385 020204 000241              CLC                    ;SET FAILURE
3386 020206 000403              BR       60$          ;
3387 020210 004737 017354      55$:    JSR      PC,KTOFF    ;TURN OFF MEMORY MAPPING
3388 020214 000261              SEC                    ;SET SUCCESS
3389 020216 000207      60$:    RTS      PC
3390              .SBTTL  REGSAV - SAVE R1-R5 ON STACK
3391              ;+
3392              ;
3393              ;ROUTINE TO
3394              ;SAVE R1 THROUGH R5 ON THE STACK
3395              ;
3396              ;CALLING SEQUENCE:
3397              ;
3398              ;      JSR      R5,REGSAV
3399              ;
3400              ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
3401              ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
3402              ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
3403              ;REGISTERS.
3404              ;
3405              ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
3406              ;CALLED VIA A JSR PC INSTRUCTION
3407              ;
3408              ;-
3409
3410 020220              REGSAV:
3411 020220 010446      MOV      R4,-(SP)
3412 020222 010346      MOV      R3,-(SP)
3413 020224 010246      MOV      R2,-(SP)
3414 020226 010146      MOV      R1,-(SP)
3415 020230 010546      MOV      R5,-(SP)
3416 020232 016605 000012      MOV      10,(SP),R5
3417 020236 004736      JSR      PC,@(SP)+
3418 020240 012601      MOV      (SP)+,R1
3419 020242 012602      MOV      (SP)+,R2
3420 020244 012603      MOV      (SP)+,R3
3421 020246 012604      MOV      (SP)+,R4
3422 020250 012605      MOV      (SP)+,R5
3423 020252 000207      RTS      PC

```

GETPAT - GET 8 BIT PATTERN FROM OPERATOR

```

3425          .SBTTL  GETPAT  - GET 8 BIT PATTERN FROM OPERATOR
3426          ;+
3427          ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
3428          ;
3429          ;INPUTS:      NONE.
3430          ;
3431          ;OUTPUTS:
3432          ;          RO      OCTAL NUMBER FROM THE OPERATOR
3433          ;
3434          ;CALLING SEQUENCE:
3435          ;          JSR      PC,GETPAT
3436          ;-
3437 020254    GETPAT::
3438 020254          SAVREG          ;SAVE THE GENERAL REGISTERS
3439 020260          1$:          GMANID  DATASC,PATDAT,0,377,0,377,NO
          020260          104443    TRAP      C$GMAN
          020262          000406    BR        10000$
          020264          020310    .WORD   PATDAT
          020266          C00022    .WORD   T$CODE
          020270          020312    .WORD   DATASC
          020272          000377    .WORD   377
          020274          000000    .WORD   T$LOLIM
          020276          000377    .WORD   T$HILIM
          020300          10000$:
3440 020300          BNCOMPLETE      1$          ;RETRY IF ERROR
          020300          103367    BCC      1$
3441 020302          013700    020310    MOV     PATDAT,RO          ;DATA PATTERN FROM OPERATOR
3442 020306          000207    RTS      PC          ;RETURN TO CALLER
3443
3444          ;+
3445          ;LOCAL DATA AREA
3446          ;-
3447
3448 020310          000000          PATDAT: .WORD   0          ;TEMPORARY STORAGE FOR DATA
3449 020312          105          116          124    DATASC: .ASCIZ  'ENTER DATA PATTERN'
3450

```



GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

```

3452 .SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
3453 ;*
3454 ;ROUTINE TO ISSUE A MENU AND GET THE OPERATOR'S RESPONSE.
3455 ;
3456 ;INPUTS:
3457 ; R0 ADDRESS OF ASCIZ STRING OF MENU
3458 ; R1 MAXIMUM ALLOWABLE OPERATOR RESPONSE
3459 ;
3460 ;OUTPUTS:
3461 ; R0 NUMBER OF THE OPERATOR'S SELECTION
3462 ;-
3463 020336 GETSEL:: SAVREG ;SAVE GENERAL REGISTERS
3464 020336 MOV R0,R2 ;SAVE THE MENU ADDRESS
3465 020342 010002 MOV R2,R3 ;START OF MENU STRING
3466 020344 010203 TST (R3) ;END OF ASCII ?
3467 020346 005713 BEQ 3# ;BRANCH IF ALL LINES DISPLAYED
3468 020350 001412 PRINTF #SELASC,(R3)+ ;DISPLAY THE MENU
3469 020352 MOV (R3)+,-(SP)
020352 C12346 MOV #SELASC,-(SP)
020354 012746 020522 MOV #2,-(SP)
020360 012746 000002 MOV SP,R0
020364 010600 TRAP C#PNTF
020366 104417 ADD #6,SP
020370 062706 000006 BR 2#
3470 020374 000764 3# : GMANID MENASC,MENRES,D,-1,0,-1,NO
3471 020376 TRAP C#GMAN
020376 104443 BR 10001#
020400 000406 .WORD MENRES
020402 020556 .WORD T#CODE
020404 000042 .WORD MENASC
020406 020527 .WORD -1
020410 177777 .WORD T#LOLIM
020412 000000 .WORD T#HILIM
020414 177777 10001# : BNCOMPLETE 1# ;RETRY IF ERROR
020416 BR 1#
3472 020416 103352 BCC 1#
3473 020420 013700 020556 MOV MENRES,R0 ;GET THE OPERATOR'S REPLY
3474 020424 020001 CMP R0,R1 ;COMPARE TO MAXIMUM ALLOWED
3475 020426 101411 BLOS 5# ;BRANCH IF OK
3476 020430 PRINTF #MENERR ;DISPLAY ERROR MESSAGE
020430 012746 020454 MOV #MENERR,-(SP)
020434 012746 000001 MOV #1,-(SP)
020440 010600 MOV SP,R0
020442 104417 TRAP C#PNTF
020444 062706 000004 ADD #4,SP
3477 020450 000735 BR 1# ;RETRY
3478 020452 000207 5# : RTS PC ;RETURN TO CALLER
3479 020454 045 116 045 MFENERR: .ASCIZ '#N#A *** Menu Selection Too Large ***'
3480 020522 045 116 045 SELASC: .ASCIZ '#N#T'
3481 020527 105 156 164 MENASC: .ASCIZ 'Enter Menu Selection: '
3482 .EVEN
3483 020556 000000 MENRES: .WORD 0
    
```

CHKMAN - CHECK MANUAL INTERVENTION LEGALITY

```

3485          .SBTTL  CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
3486          ;*
3487          ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
3488          ;
3489          ;INPUT:
3490          ;
3491          ;      NONE.
3492          ;
3493          ;OUTPUT:
3494          ;
3495          ;      CARRY  0      MANUAL INTERVENTION NOT ALLOWED
3496          ;              1      MANUAL INTERVENTION IS OK
3497          ;
3498          ;SIDE EFFECTS:
3499          ;
3500          ;      A MESSAGE IS DISPLAYED WARNING THAT TEST IS
3501          ;      NOT EXECUTED IF MANUAL INTERVENTION IS NOT
3502          ;      ALLOWED.
3503          ;
3504          ;-
3505          ;-
3506          ;-
3507 020560      CHKMAN:  SAVREG          ;SAVE THE REGISTERS
3508 020560      MANUAL          ;SEE IF MANUAL INTERVENTION OK
3509 020564      TRAP  C#MANI
3510 020566      BCOMPLETE 1#          ;BRANCH IF ALLOWED
3511 020570      PRINTF #NOMAN          ;PRINT THE WARNING MESSAGE
3512 020570      MOV #NOMAN, -(SP)
3513 020574      MOV #1, -(SP)
3514 020600      MOV SP, R0
3515 020602      TRAP C#PNTF
3516 020604      ADD #4, SP
3517 020610      CLC          ;CLEAR CARRY FOR ERROR
3518 020612      RTS  PC          ;RETURN
3519 020614      .ASCIZ 'N#A *** Manual Intervention not Allowed - Test Aborted ***'
3520 045          .even

```

ENVIRN - SETUP FREE DIAGNOSTIC SPACE

```

3518          .SBTTL  ENVIRN - SETUP FREE DIAGNOSTIC SPACE
3519          ;
3520          ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
3521          ;
3522 020710     ENVIRN: MEMORY  R0
                020710     104431     TRAP      C#MEM
3523 020712     010037     003116     MOV      R0,FREE      ; GET 1ST FREE ADDRESS...
3524 020716     062737     000002     003116     ADD      #2,FREE
3525 020724     011037     003120     MOV      (R0),FRESIZ  ; ...AND WORD COUNT.
3526 020730     162737     000004     003120     SUB      #4,FRESIZ
3527 020736     013702     002012     MOV      L#UNIT,R2  ; GET NUMBER OF UNITS
3528 020742     162737     000007     003120     10#: SUB      #7,FRESIZ  ; TAKE AWAY 7 WORDS PER UNIT
3529 020750     005302     DEC      R2
3530 020752     001373     BNE     10#
3531 020754     013700     003116     MOV      FREE,R0    ;GET FIRST FREE ADDRESS
3532 020760     063700     003120     ADD      FRESIZ,R0  ;POINT TO LAST FREE ADDRESS
3533 020764     162700     000002     SUB      #2,R0      ;BACKUP 1 WORD
3534 020770     010037     003122     MOV      R0,FREEHI  ;STORE LAST FREE ADDRESS
3535 020774     C00240     NOP
3536 020776     012701     177520     MOV      #BDVPCR,R1 ;GET BDV11 PCR ADDRESS
3537 021002     010102     MOV      R1,R2      ;COPY TO R2
3538 021004     062702     000002     ADD      #2,R2      ;SET THE RANGE
3539 021010     004737     016456     JSR      PC,XNXM    ;SEE IF WE HAVE ONE
3540 021014     103001     BCC     15#         ;OK TO SET FLAGS
3541 021016     000423     BR      40#        ;RETURN WITH FLAGS CLEAR
3542 021020     013701     177520     15#: MOV      BDVPCR,R1 ;SAVE PCR CONTENTS
3543 021024     062701     000001     ADD      #1,R1      ;ADD ONE TO IT
3544 021030     012702     177520     MOV      #BDVPCR,R2 ;GET BDV11 PCR ADDRESS
3545 021034     005212     INC      (R2)       ;TRY TO WRITE TO IT
3546 021036     013703     177520     MOV      BDVPCR,R3 ;GET RESULTS
3547 021042     020103     CMP      R1,R3      ;DID IT CHANGE?
3548 021044     001006     BNE     20#        ;NO, MUST BE 11/238
3549 021046     005237     003136     INC      T23A       ;SET THE FLAG
3550 021052     042737     170000     002120     BIC      #170000,L#HIME ;SUPERVISOR COULD BE WRONG
3551          ;
3552          ; PRINTF #M8186 ;TELL THE SYSTEM TYPE
3553 021060     000402     BR      40#        ;RETURN
3554 021062     005237     003140     20#: INC      T23B   ;SET THE FLAG
3555          ;
3556          ; PRINTF #M8189 ;TELL THE SYSTEM TYPE
3557 021066     40#:
3558 021066     000207     RTS      PC        ;RETURN

```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

.SBTTL KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3560
3561
3562
3563
3564
3565
3566
3567 021070
3568 021070 005037 003124
3569 021074 005037 003126
3570 021100 023727 002120 001577
3571 021106 101454
3572 021110 013700 000004
3573 021114 012737 021226 000004
3574 021122 005737 177572
3575 021126 000240
3576 021130 013737 002120 003124
3577 021136 022737 007777 003124
3578 021144 100404
3579 021146 042737 003777 003124
3580 021154 000403
3581 021156 042737 000177 003124 44:
3582 021164 010037 000004 54:
3583 021170 005000
3584 021172 012701 172340
3585 021176 012761 077406 177740 14:
3586 021204 010021
3587 021206 062700 000200
3588 021212 020027 002000
3589 021216 001367
3590 021220 012741 177600
3591 021224 000405
3592
3593 021226 012716 021234 24:
3594 021232 000002
3595
3596 021234 010037 000004 64:
3597
3598 021240 000207 94:

```

```

;
; ROUTINE TO INIT KT-11
;
;
KTINIT:
CLR      KTFLG      ; INIT >28K MEMORY FLAG
CLR      KTENABLE  ; INIT TEST >28K FLAG
CMP      L#HIME,#1577 ; GOT ENOUGH MEMORY (>28K)?
BLOS    94         ; NO.
MOV      @#ERRVEC,R0 ; SAVE OLD ERR VEC PTR.
MOV      @24,@#ERRVEC ; SET ERR VEC PTR.
TST     @#SRO      ; GOT KT11?
NOP     ; (TRAP IF NO).
MOV      L#HIME,KTFLG ; YES. SET KT FLAG.
CMP      @7777,KTFLG ; GOT 22 BIT MACHINE?
BMI     44         ; NO
BIC     @3777,KTFLG ; ALIGN ON BOUNDARY
BR      54
BIC     @177,KTFLG ;
MOV      R0,@#ERRVEC ; RESTORE OLD ERR VEC PTR.
CLR     R0         ; R0 = AR DATA.
MOV     @#KIPAR,R1 ; R1 = KI REGS PTR.
MOV     @77406,-40(R1) ; SET DESCRIPTOR REG.
MOV     R0,(R1)+ ; SET KIPAR REG.
ADD     @200,R0   ; BUMP AR DATA BY "4K".
CMP     R0,@2000 ; AT "I/O"?
BNE    14         ; NO.
MOV     @177600,-(R1) ; YES. SET KTPAR7 FOR I/O.
BR     94
MOV     @64,(SP) ; SET UP RETURN
RTI    ; RTI TO NEXT LOCATION
MOV     R0,@#ERRVEC ; RESTORE OLD ERR VEC PTR.
RTS    PC

```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3600          ;*
3601          ;   SUBROUTINE TO SET EXTENDED FEATURES SWITCH
3602          ;
3603          ;   Requires that SOFINIT and WRTCHR have been done previous to call.
3604          ;
3605          ;
3606          ; INPUTS:
3607          ; R5          CURRENT UNIT NUMBER
3608          ;
3609          ; OUTPUTS:
3610          ; The Extended Features Switch is set.
3611          ;
3612          ;-
3613
3614          ;   COMMAND PACKET.
3615          ;
3616          ;   =          <.+3>&177774          ;MUST BE ON MOD 4 BOUNDRY.
3617
3618          ;
3619          ; CNDPKT:: 0          ;1ST WORD IS TS05 COMMAND.
3620          ;          0          ;2ND WORD IS THE BUFFER LOW ADDRESS.
3621          ;          0          ;3RD WORD IS THE BUFFER HIGH ADDRESS.
3622          ;          0          ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
3623
3624          ;   WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
3625          ;
3626          ; WSMBK:: 0          ;1ST WORD:: SEL 0
3627          ;          0          ;2ND WORD:: SEL 2
3628          ;          0          ;3RD WORD:: SEL 4
3629          ;          .EVEN
3630
3631          ;*
3632          ;   SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
3633          ;
3634          ;
3635          ; INPUTS:
3636          ;
3637          ; OUTPUTS:
3638          ; The NXMFLG is set if we can test.
3639          ; The NXMLO and NXMHI addresses are setup.
3640          ;-
3641          ;
3642          ; MEMCK::
3643          ;
3644          ; SAVREG
3645          ; CLR          NXMFLG          ;SAVE THE REGISTERS
3646          ; CLR          NXMLO          ;CLEAR THE FLAG
3647          ; CLR          NXMHI          ;CLEAR THE TEST ADDRESS LO
3648          ; TST          T23B          ;CLEAR THE TEST ADDRESS HI
3649          ; BEQ          1#          ;IS IT A 11/23B?
3650          ; CMP          L#IME.#7777          ;NO
3651          ; BLO          2#          ; GREATER THAN 128K
3652          ; JSR          PC,NXMTST          ; NO
3653          ; BR          13#          ;SETUP THE ADDRESS
3654          ; TST          T23A          ;SET THE FLAG AND EXIT
3655          ; BEQ          4#          ;IS IT A 11/23A?
3656          ; CMP          L#IME.#5777          ;NO
3657          ; BHI          14#          ;GREATER THAN 96K
3658          ; CMP          L#IME.#3777          ;YES,23A/23B WITH 128K MEMORY
3659          ; BLO          4#          ;GREATER THAN 64K BUT LESS THAN 92K?
3660          ;          ;NO, CHECK 24K

```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3657 021354 004737 021436      JSR    PC,NXMTST      ;SETUP THE ADDRESS
3658 021360 000411              BR     13#           ;SET THE FLAG AND EXIT
3659 021362 023727 002120 001577 4# :  CMP    L#HIME,#1577  ;GREATER THAN 24K BUT LESS THAN 64K?
3660 021370 103410              BLO   14#           ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
3661 021372 004737 021436      JSR    PC,NXMTST      ;SETUP THE ADDRESS
3662 021376 062737 000077 003134  ADD    #77,NXMHI     ;FOOL THE 11/02 & 11/03
3663 021404 005237 003130      13# :  INC    NXMFLG      ;SET THE FLAG
3664 021410 000411              BR     15#           ;EXIT
3665 021412 000410      14# :  BR     15#           ;NOP FOR PRINTOUT
3666 021414              PRINTF #NOMEM        ;TELL THEM & EXIT ***NO PRINT*****
      021414 012746 005454      MOV    #NOMEM,-(SP)
      021420 012746 000001      MOV    #1,-(SP)
      021424 010600      MOV    SP,R0
      021426 104417      TRAP  C#PNTF
      021430 062706 000004      ADD    #4,SP
3667 021434 000207      15# :  RTS     PC          ;RETURN
3668
3669
3670
3671      ;*
3672      ; SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
3673      ;
3674      ; OUTPUTS: NXMLO,NXMHI      ;SETUP WITH NXM ADDRESS
3675      ;
3676      ;-
3676 021436 013701 002120  NXMTST: MOV    L#HIME,R1      ;GET TOP OF MEMORY
3677 021442 062701 000200      ADD    #200,R1      ;MAKE IT I/O BLOCK OR OTHER NXM
3678 021446 042701 000177      BIC    #177,R1
3679 021452 010102              MOV    R1,R2      ;RESAVE RESULTS
3680              000006      .REPT 6
3681              ASL    R1      ;PUT IN PLACE FOR XFER
3682              .ENDR
3683 021470 010137 003132      MOV    R1,NXMLO     ;SAVE TEST ADDRESS LOW
3684              000012      .REPT 10
3685              ASR    R2      ;PUT IN PLACE FOR XFER
3686              .ENDR
3687 021520 042702 177700      BIC    #177700,R2   ;DON'T WANT ILA!
3688 021524 010237 003134      MOV    R2,NXMHI     ;SAVE TEST ADDRESS HIGH
3689 021530 000207      RTS     PC          ;RETURN
3690
3691 021532      ENDMOD
3700      .TITLE  TSV4 - MISCELLANEOUS SECTIONS
3701
3702 021532      BGNMOD  TSV4
      021532
3703
3709
3710
3711
3712      .SBTTL  PROTECTION TABLE
3713 021532      BGNPROT
      021532
3714 021532 177777 177777 177777  L#PROT: .WORD  -1, -1, -1, -1      ;NO DEVICE PROTECTION REQUIRED.
3715 021542      ENDPROT

```

INITIALIZE SECTION

```

3717          .SBTTL INITIALIZE SECTION
3718
3719          ;**
3720          ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3721          ;AT THE BEGINNING OF EACH PASS.
3722
3723          ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
3724          ;IF "CONTINUE", NOTHING IS REQUIRED.
3725
3726          ;
3727          ;--
3728          ;*
3729          ;INSERT TEMPORARY JUMP TO ODT
3730          ;-
3730 021542          BGNINIT
3731 021542          L$INIT::
3732 021546 005037 002220          40$: CLR     EXTFEA
3733 021552 012737 006354 002172          CLR     NXMFLG
3734 021560 005037 003146          MOV     @EPRT1,EPRTSW          ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
3735 021564 005037 003126          CLR     SIFLAG              ;CLEAR "SOFT INIT" FLAG
3736 021570 005037 002274          CLR     KTENABLE           ;CLEAR TEST ABOVE 28K FLAG
3737 021574          CLR     RAMSIZ              ;CLEAR RAM SIZE FOR RAMERR ROUTINE
3738 021574 012700 000036          READEF @EF.CONTINUE
3739 021600 104447          MOV     @EF.CONTINUE,RO
3740 021602          TRAP     C$REFG
3741 021602          BNCOMPLETE 1$
3742 021604 023737 002174 002012          BCC     1$
3743 021612 103070          CMP     UNITN,L$UNIT          ;UNIT IN RANGE?
3744 021614 005737 003104          BHIS   4$                   ;BR IF NO.
3745 021620 100472          TST     DUFLG                ;DROPPED UNIT?
3746 021622 013701 002174          BMI     NXTU                ;BR IF YES
3747 021626 006301          MOV     UNITN,R1
3748 021630 005761 003170          ASL     R1
3749 021634 001516          TST     ERTABL(R1)
3750 021636 032761 040000 003170          BEQ     SETU
3751 021644 001060          BIT     @BIT14,ERTABL(R1)   ;DROPPED?
3752 021646          BNE     NXTU
3753 021646          EXIT     INIT          ;DO NOTHING IF "CONTINUE".
3754 021646 104432          TRAP     C$EXIT
3755 021650 000416          .WORD  L10030-.
3756 021652          READEF @EF.NEW
3757 021652 012700 000035          1$: MOV     @EF.NEW,RO
3758 021656 104447          TRAP     C$REFG
3759 021660          BNCOMPLETE NXTU          ;TAKE NEXT UNIT IF NOT NEW PASS.
3760 021662 103052          BCC     NXTU
3761 021662 012700 000040          READEF @EF.START
3762 021666 104447          MOV     @EF.START,RO
3763 021670          TRAP     C$REFG
3764 021670 103404          BCOMPLETE 2$
3765 021672 012700 000037          BCS     2$
3766 021676 104447          READEF @EF.RESTART
3767 021700          MOV     @EF.RESTART,RO
3768 021700 103031          TRAP     C$REFG
3769 021702          BNCOMPLETE 31$
3770 021702 104433          BCC     31$
3771          2$: BRESET
3772          TRAP     C$RESET          ;1ST PASS, BUS-INIT...
3773          ;BUS RESET.

```

INITIALIZE SECTION

```

3758 021704 005037 002206      CLR      TSTCNT      ;NUMBER OF TESTS RUN IN PASS
3759 021710 005037 002214      CLR      FATFLG     ;CLEAR FATAL ERROR COUNT
3760 021714 005037 003136      CLR      T23A      ;CLEAR 11/23A FLAG
3761 021720 005037 003140      CLR      T23B      ;CLEAR 11/23B FLAG
3762      ;      MOV      @340,-(SP) ;RETURN TO DEBUGGER
3763      ;      MOV      @20,-(SP) ;
3764      ;      JMP      0.ODT  ;ENTER THE DEBUGGER
3765 021724 005037 003372      CLR      SKIPT      ;CLEAR THE SUBTEST "SKIPPER"
3766 021730      ;
3767 021730 012737 177777 002176 20$: MOV      @-1,QVP     ;...QUICK VERIFY...
3768 021736 004737 020710      JSR      PC,ENVIRN  ;SET ENVIRONMENT.
3769 021742 004737 021070      JSR      PC,KTINIT  ;INITIALIZE KT MEMORY MANAGEMENT
3770 021746 012700 003170      MOV      @ERTABL,RO
3771 021752 005020 003370      30$: CLR      (RO)+   ;CLEAR THE ERROR TABLE
3772 021754 020027 003370      CMP      RO,@ERTABE
3773 021760 103774      BLO     30$
3774 021762 000404      BR      4$
3775 021764 005037 002176      31$: CLR      QVP
3776 021770 C00137 022040      JMP      PASRPT     ;GO REPORT THE STATUS
3777
3778 021774      ;
3779 021774 012737 177777 002174 4$: NEWPAS: MOV     @-1,UNITN ;INIT UNIT NUMBER...
3780 022002 005037 002212      CLR     DEVCNT     ;CLEAR COUNT OF DEVICES RUNNING
3781 022006      ;
3782 022006 104422      ;
3783 022010 005237 002174      ;
3784 022014 023737 002174 002012 11$: NEXTU: BREAK ;
3785 022022 103423      TRAP    C#BRK     ;
3786 022024 012737 177777 003104 10$: INC     UNITN    ;...AND SET NEXT UNIT NUMBER.
3787 022034 000401      CMP     UNITN,L#UNIT
3788 022034 104444      BLO    SETU
3789 022036 000240      MOV     @-1,DUFLG
3790 022040 023727 002012 000001 11$: BR     11$
3791 022046 101752      ;
3792 022050 005737 002212      ;
3793 022054 001747      ;
3794 022056 104421      ;
3795 022060 032700 000100      ;
3796 022064 001343      ;
3797
3798 022066      ;
3799 022066 104424      ;
3800 022070 000741      ;
3801 022072      ;
3802 022072 013700 002174      10$: DORPT ;
3803 022100 104442      TRAP    C#DRPT   ;
3804 022102 005037 003104      BR     NEWPAS    ;
3805 022106 005237 002212      ;
3806 022112 012001      ;
3807 022114 010137 002200      SETU:  GPHARD UNITN,RO ;GET UNIT N P-TABLE POINTER.
      MOV     UNITN,RO
      TRAP    C#GPHRD
      BNCOMPLETE NXTU ;BR IF UNIT NOT AVAILABLE.
      BCC    NXTU
      CLR     DUFLG ;CLEAR "DROPPED" FLAG.
      INC     DEVCNT
      MOV     (RO)+,R1 ;GET 1ST REGISTER ADDRESS.
      MOV     R1,CSRADDR ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
    
```



INITIALIZE SECTION

```

3808
3809 022120 012001      MOV      (R0),R1      ;GET VECTOR ADDRESS.
3810                   ;MOV      (R0),R2      ;GET INTERRUPT PRIORITY
3811                   ;MOV      R2,IPRI    ;SET INTERRUPT PRIORITY.
3812 022122 010137 002202  MOV      R1,IVEC     ;SET INTERRUPT VECTOR POINTER...
3813 022126 012721 016276  MOV      @INTR,(R1)+ ;...VECTOR...
3814 022132 013721 002204  MOV      IPRI,(R1)+  ;...AND PRIORITY.
3815
3816 022136             1$:
3817                   ; TST      QVP      ;1ST PASS ??
3818                   ; BEQ      5$      ;NO. SKIP THE PASS 1 STUFF.
3819
3820
3821                   ;
3822                   ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
3823                   ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
3824 022136 013701 002174      MOV      UNITN,R1
3825 022142 006301             ASL      R1
3826 022144 C52761 100000 003170  BIS      @BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
3827 022152 005037 005766      CLR      EXTA        ;CLEAR ERROR EXTENSION FLAG.
3828 022156 023727 002012 000001  CMP      L$UNIT,#1    ;ARE WE TESTING MULTIPLE UNITS?
3829 022164 101416             BLOS    10$         ;BR IF NO.
3830 022166             RFLAGS  RO        ;YES -- GET OPERATOR FLAGS.
3831 022170 032700 001000      TRAP    C$RFLA
3832 022174 001412             BIT      @PNT,RO      ;SHOULD WE PRINT UNIT #?
3833 022176             BEQ      10$         ;BR IF NOT.
3834 022176 013746 002174      PRINTF @PUNIT,UNITN ;PRINT THE UNIT #
3835 022202 012746 022270      MOV      UNITN,-(SP)
3836 022206 012746 000002      MOV      @PUNIT,-(SP)
3837 022212 010600             MOV      #2,-(SP)
3838 022214 104417             MOV      SP,RO
3839 022216 062706 000006      TRAP    C$PNTF
3840 022222             ADD      @6,SP
3841 022222             10$:
3842 022226 005037 003106      CLR      NODEV
3843 022232 013701 002200      MOV      CSRADDR,R1 ;ADDRESS OF FIRST REGISTER
3844 022234 010102             MOV      R1,R2      ;START OF REGISTERS
3845 022238 062702 000002      ADD      @TSSR,R2  ;ADDRESS OF TSSR REGISTER
3846 022240 004737 016456      JSR      PC,XNXM   ;TEST BOTH CONTROLLER REGISTERS...
3847 022244 103005             BCC     2$         ;...AND BR IF ALL OK.
3848 022246 010137 003106      MOV      R1,NODEV  ;FLAG DEVICE AS NON-EXISTENT
3849 022252 012737 177777 003104  MOV      #-1,DUFLG ;DROP THIS UNIT.
3850 022260             2$:
3851 022260             ;
3852 022260             ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
3853 022260             5$:
3854 022260 012700 000000      SETPRI @PRI00      ;ENABLE INTERRUPTS.
3855 022264 104441             MOV      @PRI00,RO
3856 022266             TRAP    C$SPRI
3857 022266             ENDINIT
3858 022266 104411             L10030:
3859 022266             TRAP    C$INIT
3860 022270 045 116 045 PUNIT: .ASCIZ /#N#N#A***** TESTING UNIT #D2#A *****/
3861 022270 045 116 045 PUNIT: .EVEN

```

ADD AND DROP UNITS SECTIONS

```

3853          .SBTTL  ADD AND DROP UNITS SECTIONS
3854
3855          ;**
3856          ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3857          ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
3858          ; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
3859          ;--
3860          BGNAU
022336      L$AU::
022336      MOV      R0,R1          ; GET UNIT TO BE ADDED (R0)
3861      022336 010001          ASL      R1          ; MAKE IT A WORD INDEX
3862      022340 006301          BIS      #100000,ERTABL(R1) ; SET THE "ACTIVE" BIT
3863      022342 052761 100000 003170 BIC      #40000,ERTABL(R1) ; CLEAR THE "DROPPED" BIT
3864      022350 042761 040000 003170 PRINTF  #1$,R0
3865      022356          MOV      R0,-(SP)
022356      010046          MOV      #1,-(SP)
022360      012746 022404          MOV      #2,-(SP)
022364      012746 000002          MOV      SP,R0
022370      010600          TRAP    C$PNTF
022372      104417          ADD      #6,SP
022374      062706 000006          EXIT    AU
3866      022400          .WORD   J$JMP
022400      000167          .WORD   L10031-2-.
022402      000026          .ASCIZ  /#N#A UNIT #D#A ADDED/
3867      022404          045      116      045  1$: .EVEN
3868
3869
3870      022432          ENDAU          ; UNUSED.
022432      104452          L10031: TRAP    C$AU
3871
3872          ;**
3873          ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3874          ; TO BE REMOVED FROM THE TEST LIST.
3875          ;
3876          ; SUPVSR DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN.
3877          ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
3878          ; COMMAND, OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
3879          ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
3880          ; WHICH ARE STILL ACTIVE.
3881          ; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.
3882          BGNDU
022434      L$DU::
022434      MOV      #-1,DUFLG
3883      022434 012737 177777 003104      MOV      R0,R1
3884      022442 010001          ASL      R1
3885      022444 006301          BIS      #140000,ERTABL(R1) ; SAY DROPPED
3886      022446 052761 140000 003170      240,240,240 ; ??????????
3887      022454 000240 000240 000240      PRINTF  #1$,R0
3888      022462          MOV      R0,-(SP)
022462      010046          MOV      #1,-(SP)
022464      012746 022510          MOV      #2,-(SP)
022470      012746 000002          MOV      SP,R0
022474      010600          TRAP    C$PNTF
022476      104417          ADD      #6,SP
022500      062706 000006          EXIT    DU
3889      022504          .WORD   J$JMP
022504      000167          .WORD   L10032-2-.
022506      000030
    
```

ADD AND DROP UNITS SECTIONS

```

3890 022510    045    116    045 1$: .ASCIZ /#N#A UNIT #D#A DROPPED/
3891          .EVEN
3892 022540          ENDDU
          022540    L10032: TRAP    C#DU
          022540    104453
3893
3894          ;++
3895          ; AUTO-DROP CODE SECTION.
3896          ;--
          022542          BGNAUTO
          022542          L#AUTO::
3897 022542    013705    002200          MOV    CSRADDR,R5          ;POINT TO DEVICE REGISTER
3898 022546    012703    000550          MOV    #360.,R3          ;ENOUGH TIME FOR 2400' REEL TO REWIND
3899 022552    004737    016330          JSR    PC,WAITF          ;WAIT FOR SSR TO SET
3900 022556    103420          BCS    20$              ;LEAVE WHEN SSR IS SET
3901 022560          DELAY    250.          ;WAIT FOR .25 SECONDS
          022560    012727    000372          MOV    #250.,(PC)+
          022564    000000          .WORD    0
          022566    013727    002116          MOV    L#DLY,(PC)+
          022572    C00000          .WORD    0
          022574    005367    177772          DEC    -6(PC)
          022600    001375          BNE    -.4
          022602    005367    177756          DEC    -22(PC)
          022606    001367          BNE    -.20
3902 022610    005303          DEC    R3              ;BUMP COUNTER DOWN
3903 022612    001357          BNE    10$             ;KEEP GOING
3904 022614    004737    017262          JSR    PC,CKDROP       ;TRY AND DROP UNIT
3905 022620          20$:
3906 022620          ENDAUTO
          022620          L10033:
          022620    104461          TRAP    C#AUTO
          ; UNUSED.

```

CLEAN-UP AND REPORT CODING SECTIONS

```

3908 .SBTTL CLEAN-UP AND REPORT CODING SECTIONS
3909
3910 ;++
3911 ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
3912 ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
3913 ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
3914 ;--
3915 022622 BGNCLN
022622 L$CLEAN::
3916 022622 013705 002200 MOV CSRADDR,R5 ;POINT TO DEVICE REGISTER
3917 022626 005737 003104 TST DUFLG ;"DROPPED" FLAG IS SET ON...
3918 022632 100405 BMI 1$ ;...AND GROSS CONTROLLER FAULT...
3919 ;...DON'T TRY TO XCT CLEANUP CODE.
3920
3921 022634 012765 000000 000002 MOV #0,TSSR(R5) ;DO SOFT INIT
3922 022642 004737 016330 JSR PC,WAITF
3923 022646 1$:
3924 022646 2$: ENDCLN
022646 L10034:
022646 104412 TRAP C$CLEAN
3925 ;++
3926 ; THE REPORT CODING SECTION CONTAINS THE
3927 ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
3928 ;--
3929 022650 BGNRPT
022650 L$RPT::
3930 022650 PRINTS #DEVSUM
022650 012746 023112 MOV #DEVSUM,-(SP)
022654 012746 000001 MOV #1,-(SP)
022660 010600 MOV SP,R0
022662 104416 TRAP C$PNTS
022664 062706 000004 ADD #4,SP
3931 022670 010246 MOV R2,-(SP)
3932 022672 010346 MOV R3,-(SP)
3933 022674 010446 MOV R4,-(SP)
3934 022676 012704 003170 MOV #ERTABL,R4 ; GET START OF ERROR TABLE.
3935 022702 005003 CLR R3 ; CLEAR UNIT NUMBER
3936 022704 011402 1$: MOV (R4),R2 ; GET ERROR TABLE ENTRY & TEST IT.
3937 022706 001467 BEQ 4$ ; ZERO IF UNIT NOT RUN
3938 022710 100066 BPL 4$
3939 022712 032702 040000 BIT #BIT14,R2 ; WAS UNIT DROPPED?
3940 022716 001015 BNE 2$ ; BR IF YES
3941 022720 042702 170000 BIC #C7777,R2 ; GET ERROR COUNT FIELD
3942 022724 PRINTS #DEVONL,R3,R2 ; PRINT
022724 010246 MOV R2,-(SP)
022726 010346 MOV R3,-(SP)
022730 012746 023147 MOV #DEVONL,-(SP)
022734 012746 000003 MOV #3,-(SP)
022740 010600 MOV SP,R0
022742 104416 TRAP C$PNTS
022744 062706 000010 ADD #10,SP
3943 022750 000446 BR 4$
3944 022752 020227 160000 2$: CMP R2,#160000 ; WAS UNIT NON-EXISTENT?
3945 022756 001012 BNE 3$ ; BR IF NO
3946 022760 PRINTS #DEVNXR,R3
022760 010346 MOV R3,-(SP)
022762 012746 023217 MOV #DEVNXR,-(SP)

```

CLEAN-UP AND REPORT CODING SECTIONS

```

022766 012746 000002      MOV      #2,-(SP)
022772 010600      MOV      SP,R0
022774 104416      TRAP     C#PNTS
022776 062706 000006      ADD      #6,SP
3947 023002 000431      BR       4#
3948 023004 020227 160001      3#:     CMP      R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
3949 023010 001012      BNE     30#              ; BR IF NO.
3950 023012      PRINTS  #DEVNRD,R3
023012 010346      MOV      R3,-(SP)
023014 012746 023301      MOV      #DEVNRD,-(SP)
023020 012746 000002      MOV      #2,-(SP)
023024 010600      MOV      SP,R0
023026 104416      TRAP     C#PNTS
023030 062706 000006      ADD      #6,SP
3951 023034 000414      BR       4#
3952 023036 042702 170000      30#:    BIC      #+C7777,R2
3953 023042      PRINTS  #DEVDR0,R3,R2
023042 010246      MOV      R2,-(SP)
023044 C10346      MOV      R3,-(SP)
023046 012746 023362      MOV      #DEVDR0,-(SP)
023052 012746 000003      MOV      #3,-(SP)
023056 010600      MOV      SP,R0
023060 104416      TRAP     C#PNTS
023062 062706 000010      ADD      #10,SP
3954 023066 062704 000002      4#:     ADD      #2,R4
3955 023072 005203      INC      R3
3956 023074 020427 003370      CMP      R4,#ERTABE
3957 023100 103701      BLO     1#
3958 023102 012604      MOV      (SP)+,R4
3959 023104 012603      MOV      (SP)+,R3
3960 023106 012602      MOV      (SP)+,R2
3961 023110      ENDRPT      ; UNUSED.
023110      L10035:
023110 104425      TRAP     C#RPT
3962
3963 023112      045      116      045  DEVSUM: .ASCIZ /#N#ADEVICE STATUS SUMMARY:#N/
3964 023147      045      101      040  DEVONL: .ASCIZ /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
3965 023217      045      101      040  DEVNXR: .ASCIZ /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
3966 023301      045      101      040  DEVNRD: .ASCIZ /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
3967 023362      045      101      040  DEVDR0: .ASCIZ /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
3968      .EVEN
3969
3970 023432      ENDMOD
3971

```

CLEAN-UP AND REPORT CODING SECTIONS

3975  
3976  
3977  
3984  
3985  
3991

023432  
023432

.TITLE TEST 1 - HARDWARE TEST 1-8 TESTS

TSV7B:: BGNMOD TSV7B



## TEST 1: WRITE TAPE MARK RETRY

	023566	012114										.WORD	SFIMSG
4050	023570	013737	002174	026240	20‡:	MOV	UNITN,T29DSW						
4051													
4052	023576	012704	026220			MOV	#T29PACKET,R4						
4053	023602	004737	010742			JSR	PC,WRTCHR						
4054	023606	103407				BCS	25‡						
4055	023610	005237	002214			INC	FATFLG						
4059	023614	010001				MOV	R0,R1						
4060	023616					ERRHRD	ERRNO,WRTMSG,SFIMSG						
	023616	104456											TRAP
	023620	000146											.WORD
	023622	005052											.WORD
	023624	012114											.WORD
4061	023626				25‡:	CKLOOP							
	023626	104406											TRAP
4062	023630	016501	000002			MOV	TSSR(R5),R1						
4063	023634	010102				MOV	R1,R2						
4064	023636	042702	000100			BIC	#OFL,R2						
4065	023642	020102				CMP	R1,R2						
4066	023644	001406				BEQ	26‡						
4070	023646					ERRDF	ERRNO,T29OFL,EXPREC						
	023646	104455											TRAP
	023650	000147											.WORD
	023652	026402											.WORD
	023654	015554											.WORD
4071	023656	004737	017262			JSR	PC,CKDROP						
4072	023662	004737	011074			JSR	PC,REWIND						
4073	023666	016501	000002		26‡:	MOV	TSSR(R5),R1						
4074	023672	012702	000200			MOV	#SSR,R2						
4075	023676	103407				BCS	30‡						
4076	023700	010004				MOV	R0,R4						
4077	023702	005237	002214			INC	FATFLG						
4081	023706					ERRHRD	ERRNO,T29RWN,PKTSSR						
	023706	104456											TRAP
	023710	000150											.WORD
	023712	030205											.WORD
	023714	012126											.WORD
4082	023716				30‡:	CKLOOP							
	023716	104406											TRAP
4083	023720	013701	026250			MOV	T29BFR+6,R1						
4084	023724	010102				MOV	R1,R2						
4085	023726	052702	000002			BIS	#BIT1,R2						
4086	023732	020102				CMP	R1,R2						
4087	023734	001406				BEQ	40‡						
4088	023736	005237	002214			INC	FATFLG						
4092	023742					ERRHRD	ERRNO,T29BOT,EXPREC						
	023742	104456											TRAP
	023744	000151											.WORD
	023746	027676											.WORD
	023750	015554											.WORD
4093	023752				40‡:	CKLOOP							
	023752	104406											TRAP
4094	023754	013737	003116	026342		MOV	FREE,T29RB						
4095	023762	012737	141011	026340		MOV	#141011,T29PK3						
4096	023770	012704	026340			MOV	#T29PK3,R4						
4097	023774	010465	000000			MOV	R4,TSDB(R5)						
4098	024000	004737	016330			JSR	PC,WAITF						





TEST 1: WRITE TAPE MARK RETRY

	024146	000154										.WORD	108
	024150	003646										.WORD	SFIERR
	024152	012114										.WORD	SFIMSG
4151	024154	013737	002174	026240	20:	MOV	UNITN,T29DSW			;SET UP UNIT NUMBER			
4152													
4153	024162	012704	026220			MOV	@T29PACKET,R4			;SUBROUTINE NEEDS PACKET ADDRESS			
4154	024166	004737	010742			JSR	PC,WRTCHR			;ISSUE WRITE CHARACTERISTICS			
4155	024172	103407				BCS	25:			;BR, IF COMMAND ISSUED OK			
4156	024174	005237	002214			INC	FATFLG			;ERROR COUNT			
4160	024200	010001				MOV	RO,R1			;SAVE CONTENTS OF TSSR			
4161	024202					ERRHRD	ERRNO,WRTMSG,SFIMSG			;WRITE CHARACTERISTICSC FAILED			
	024202	104456								TRAP			C#ERHRD
	024204	000155								.WORD			109
	024206	005052								.WORD			WRTMSG
	024210	012114								.WORD			SFIMSG
4162	024212				25:	CKLOOP				;LOOP IF SELECTED			
	024212	104406								TRAP			C#CLP1
4163	024214	004737	011074		26:	JSR	PC,REWIND			;CALL TAPE REWIND COMMAND			
4164	024220	C16501	000002			MOV	TSSR(R5),R1			;GET TSSR			
4165	024224	012702	000200			MOV	@SSR,R2			;SET UP EXPECTED TSSR			
4166	024230	103407				BCS	30:			;BR, IF NO PROBLEM			
4167	024232	010004				MOV	RO,R4			;PACKET ADDRESS SET UP			
4168	024234	005237	002214			INC	FATFLG			;ERROR COUNT			
4172	024240					ERRHRD	ERRNO,T29RWN,PKTSSR			;REWIND NOT ACCEPTED			
	024240	104456								TRAP			C#ERHRD
	024242	000156								.WORD			110
	024244	030205								.WORD			T29RWN
	024246	012126								.WORD			PKTSSR
4173	024250				30:	CKLOOP				;LOOP IF SELECTED			
	024250	104406								TRAP			C#CLP1
4174	024252	013701	026250			MOV	T298FR+6,R1			;PICK UP XSTO			
4175	024256	010102				MOV	R1,R2			;SET UP EXPECTED			
4176	024260	052702	000002			BIS	@BIT1,R2			;SET BOT BIT IN EXPECTED			
4177	024264	020102				CMP	R1,R2			;DOES EXP = REC'D			
4178	024266	001406				BEQ	40:			;BR, IF EQUAL (OK)			
4179	024270	005237	002214			INC	FATFLG			;ERROR COUNT			
4183	024274					ERRHRD	ERRNO,T29BOT,EXPREC			;TAPE NOT AT BOT AFTER REWIND			
	024274	104456								TRAP			C#ERHRD
	024276	000157								.WORD			111
	024300	027676								.WORD			T29BOT
	024302	015554								.WORD			EXPREC
4184	024304	012737	000001	026342	40:	MOV	@1,T29RB			;NUMBER OF RECORDS TO SPACE OVER			
4185	024312	012737	000400	026346		MOV	@256,T29SZ			;SET UP RECORD SIZE			
4186	024320	012737	140005	026340		MOV	@140005,T29PK3			;WRITE FORWARD,CVC-1,ACK COMMAND			
4187	024326	012704	026340			MOV	@T29PK3,R4			;SET UP R4 WITH PACKET ADDRESS			
4188	024332	010465	000000			MOV	R4,TSDB(R5)			;ISSUE COMMAND			
4189	024336	004737	016330			JSR	PC,WAITF			;WAIT FOR SSR TO SET			
4190	024342	016501	000002			MOV	TSSR(R5),R1			;GET TSSR CONTENTS			
4191	024346	012702	000200			MOV	@SSR,R2			;SET UP EXPECTED			
4192	024352	020102				CMP	R1,R2			;ARE THEY EQUAL			
4193	024354	001420				BEQ	75:			;BR, IF OK			
4194	024356	013703	026250			MOV	T298FR+6,R3			;PICK UP XTSD			
4195	024362	032703	000004			BIT	@4,R3			;IS UNIT WRITE-LOCKED?			
4196	024366	001405				BEQ	41:			;NO,PROCEED WITH NORMAL ERROR			
4197	024370					ERRDF	ERRNO,T29MLK,SFIMSG			;TAPE IS WRITE LOCKED			
	024370	104455								TRAP			C#ERDF
	024372	000157								.WORD			111





TEST 1: WRITE TAPE MARK RETRY

4287	025002	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND		
4288	025006	103411			BCS	30#		;BR, IF NO PROBLEM		
4289	025010	016501	000002		MOV	TSSR(R5),R1		;GET TSSR		
4290	025014	010004			MOV	R0,R4		;SAVE PACKET ADDRESS		
4291	025016	005237	002214		INC	FATFLG		;ERROR COUNT		
4295	025022				ERRHRD	ERRNO,T29RWN,PKTSSR		;REWIND NOT ACCEPTED		
	025022	104456							TRAP	C#ERHRD
	025024	000166							.WORD	118
	025026	030205							.WORD	T29RWN
	025030	012126							.WORD	PKTSSR
4296	025032			30#:	CKLOOP			;LOOP IF SELECTED		
	025032	104406							TRAP	C#CLP1
4297	025034	013701	026250		MOV	T29BFR+6,R1		;PICK UP XSTO		
4298	025040	010102			MOV	R1,R2		;SET UP EXPECTED		
4299	025042	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED		
4300	025046	020102			CMP	R1,R2		;DOES EXP = REC'D		
4301	025050	001406			BEQ	40#		;BR, IF EQUAL (OK)		
4302	025052	005237	002214		INC	FATFLG		;ERROR COUNT		
4306	025056				ERRHRD	ERRNO,T29BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND		
	025056	104456							TRAP	C#ERHRD
	025060	000167							.WORD	119
	025062	027676							.WORD	T29BOT
	025064	015554							.WORD	EXPREC
4307	025066			40#:	CKLOOP			;LOOP IF SELECTED		
	025066	104406							TRAP	C#CLP1
4308	025070	012737	140011	026340	MOV	#140011,T29PK3		;WRITE TAPE MARK,ACK,CVC-1 COMMAND		
4309	025076	012704	026340		MOV	#T29PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
4310	025102	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND		
4311	025106	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET		
4312	025112	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
4313	025116	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED		
4314	025122	020102			CMP	R1,R2		;ARE THEY EQUAL		
4315	025124	001406			BEQ	70#		;BR, IF OK		
4316	025126	005237	002214		INC	FATFLG		;ERROR COUNT		
4320	025132				ERRHRD	ERRNO,T29WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE TAPE MARK		
	025132	104456							TRAP	C#ERHRD
	025134	000170							.WORD	120
	025136	030577							.WORD	T29WDC
	025140	012126							.WORD	PKTSSR
4321	025142			70#:	CKLOOP			;LOOP IF SELECTED		
	025142	104406							TRAP	C#CLP1
4322	025144	012703	000001		MOV	#1.,R3		;NUMBER OF RECORDS TO WRITE TH		
4323	025150	012737	141011	026340	MOV	#141011,T29PK3		;WRITE TAPE MARK RETRY,ACK,CVC-1 COMMAND		
4324	025156	012704	026340		MOV	#T29PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
4325	025162	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND		
4326	025166	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET		
4327	025172	016501	000002		MOV	TSSR(R5),R1		;PICK UP TSSR		
4328	025176	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED (SSR ONLY)		
4329	025202	020102			CMP	R1,R2		;WAS STATUS GOOD		
4330	025204	001406			BEQ	165#		;BR, IF TERMINATION WAS GOOD		
4331	025206	005237	002214		INC	FATFLG		;ERROR COUNT		
4335	025212				ERRHRD	ERRNO,T29WDC,PKTSSR		;TSSR NOT CORRECT AFTER WRT TAPE M.		
	025212	104456							TRAP	C#ERHRD
	025214	000171							.WORD	121
	025216	030577							.WORD	T29WDC
	025220	012126							.WORD	PKTSSR
4336	025222			165#:	CKLOOP			;LOOP IF SELECTED		





TEST 1: WRITE TAPE MARK RETRY

4427	025614	012737	140011	026340	MOV	#140011,T29PK3	;WRITE TAPE MARK,ACK,CVC=1 COMMAND
4428	025622	012704	026340		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
4429	025626	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND
4430	025632	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
4431	025636	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS
4432	025642	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED
4433	025646	020102			CMP	R1,R2	;ARE THEY EQUAL
4434	025650	001406			BEQ	70#	;BR, IF OK
4435	025652	005237	002214		INC	FATFLG	;ERROR COUNT
4439	025656				ERRHRD	ERRNO,T29WDC,PKTSSR	;TSSR INCORRECT AFTER WRITE TAPE MARK
	025656	104456					TRAP C#ERHRD
	025660	000200					.WORD 128
	025662	030577					.WORD T29WDC
	025664	012126					.WORD PKTSSR
4440	025666				70#:	CKLOOP	;LOOP IF SELECTED
	025666	104406					TRAP C#CLP1
4441	025670	012703	000012		150#:	MOV	#10.,R3
4442	025674	012737	000001	026342		MOV	#1,T29RB
4443	025702	012737	141011	026340		MOV	#141011,T29PK3
4444	025710	012704	026340			MOV	#T29PK3,R4
4445	025714	010465	000000		155#:	MOV	R4,TSDB(R5)
4446	025720	004737	016330			JSR	PC,WAITF
4447	025724	016501	000002			MOV	TSSR(R5),R1
4448	025730	012702	000200			MOV	#SSR,R2
4449	025734	020102				CMP	R1,R2
4450	025736	001406				BEQ	165#
4451	025740	005237	002214			INC	FATFLG
4455	025744					ERRHRD	ERRNO,T29WDC,PKTSSR
	025744	104456					TRAP C#ERHRD
	025746	000201					.WORD 129
	025750	030577					.WORD T29WDC
	025752	012126					.WORD PKTSSR
4456	025754				165#:	CKLOOP	;LOOP IF SELECTED
	025754	104406					TRAP C#CLP1
4457	025756	005303				DEC	R3
4458	025760	001355				BNE	155#
4459	025762	012737	140410	026340		MOV	#140410,T29PK3
4460	025770	012737	000001	026342		MOV	#1,T29RB
4461	025776	012704	026340			MOV	#T29PK3,R4
4462	026002	010465	000000			MOV	R4,TSDB(R5)
4463	026006	004737	016330			JSR	PC,WAITF
4464	026012	016501	000002			MOV	TSSR(R5),R1
4465	026016	012702	100204			MOV	#SSR!SC!BIT2,R2
4466	026022	020102				CMP	R1,R2
4467	026024	001406				BEQ	222#
4468	026026	005237	002214			INC	FATFLG
4472	026032					ERRHRD	ERRNO,T29WDE,PKTSSR
	026032	104456					TRAP C#ERHRD
	026034	000202					.WORD 130
	026036	027462					.WORD T29WDE
	026040	012126					.WORD PKTSSR
4473	026042				222#:	CKLOOP	;LOOP IF SELECTED
	026042	104406					TRAP C#CLP1
4474	026044	012737	100410	026340		MOV	#100410,T29PK3
4475	026052	012737	000005	026342		MOV	#5,T29RB
4476	026060	012704	026340			MOV	#T29PK3,R4
4477	026064	010465	000000			MOV	R4,TSDB(R5)



TEST 1: WRITE TAPE MARK RETRY

```

4478 026070 004737 016330          JSR    PC,WAITF           ;WAIT FOR SSR TO SET
4479 026074 016501 000002          MOV    TSSR(R5),R1       ;GET TSSR CONTENTS
4480 026100 012702 100204          MOV    #SSR!SC!BIT2,R2  ;SET UP EXPECTED
4481 026104 020102                  CMP    R1,R2             ;ARE THEY EQUAL
4482 026106 001406                  BEQ    260#              ;BR, IF OK
4483 026110 005237 002214          INC    FATFLG            ;ERROR COUNT
4487 026114                  ERRHRD  ERRNO,T29RDG,PKTSSR ;TSSR INCORRECT AFTER SPACE REV CMD.
                                TRAP    C#ERHRD
                                .WORD   131
                                .WORD   T29RDG
                                .WORD   PKTSSR
                                TRAP    C#CLP1
                                .WORD   104456
                                .WORD   000203
                                .WORD   031543
                                .WORD   012126
4488 026124 260#  CKLOOP             ;LOOP IF SELECTED
                                TRAP    C#CLP1
                                .WORD   104406
4489 026126 013701 026256          MOV    T29BFR+14,R1     ;PICK UP XST3
4490 026132 010102                  MOV    R1,R2            ;SET UP EXPECTED
4491 026134 052702 000001          BIS    #BIT0,R2        ;RIB SHOULD BE SET
4492 026140 020102                  CMP    R1,R2            ;IS RIB SET
4493 026142 001406                  BEQ    270#              ;BR, IF RIB WAS SET (GOOD)
4494 026144 C05237 002214          INC    FATFLG            ;ERROR COUNT
4498 026150                  ERRHRD  ERRNO,T29RIB,EXPREC ;TMK NOT SET AFTER READ REV
                                TRAP    C#ERHRD
                                .WORD   132
                                .WORD   T29RIB
                                .WORD   EXPREC
                                TRAP    C#CLP1
                                .WORD   104456
4499 026160 270#  CKLOOP             ;LOOP IF SELECTED
                                TRAP    C#CLP1
                                .WORD   104406
4500 026162 330#  CKLOOP             ;LOOP IF SELECTED
                                TRAP    C#CLP1
                                .WORD   104406
4501 026164                  ENDSUB                  ;<<<<<<<<<<<<< END SUBTEST >>>>>>>>>
                                L10042:
                                TRAP    C#ESUB
                                .WORD   104403
4502 026166 023727 002214 000017      CMP    FATFLG,#15.      ;IS ERROR COUNT AT 25
4503 026174 103402                  BLO    999#              ;BR, IF LESS THAN 25
4504 026176 004737 017262          JSR    PC,CKDROP        ;TRY TO DROP THE UNIT
4505 026202 999#  ;
4506 ;
4507 ;
4508 ;
4509 026202 004737 016536          JSR    PC,TSTLOOP      ;DO WE NEED TO ITERATE TEST
4510 026206 103002                  BCC    163#              ;BR, IF NO LOOP REQUIRED
4511 026210 000137 023462          JMP    T29LOOP         ;EXECUTE AGAIN
4512 026214 163#  EXIT                ;ALL DONE THIS TEST
                                TRAP    C#EXIT
                                .WORD   L10036--
                                .WORD   104432
                                .WORD   004014
4513 ;
4514 ;* LOCAL STORAGE FOR THIS TEST
4515 ;-
T29PACKET:
4519 026220                  ;COMMAND PACKET FOR TEST
4520 026220 014004                  .WORD 14004             ;WRITE CHARACTERISTICS COMMAND, WITH CVC=1. ACK
4521 026222 026230                  .WORD T29DATA           ;ADDRESS OF CHARACTERISTICS BLOCK
4522 026224 000000                  .WORD 0
4523 026226 000012                  .WORD 10.              ;STARTING VALUE OF BLOCK SIZE
4524 026230 T29DATA:                    ;CHARACTERISTICS DATA BLOCK
4525 026230 026242                  .WORD T29BFR           ;ADDRESS OF MESSAGE BUFFER
4526 026232 000000                  .WORD 0
4527 026234 000024                  .WORD 20.              ;LENGTH OF MESSAGE BUFFER
4528 026236 000000                  .WORD 0

```

TEST 1: WRITE TAPE MARK RETRY

```

4529 026240 000000 T29DSW: .WORD 0 ;SELECT DRIVE 0
4530 026242 T29BFR: .BLKW 25. ;MESSAGE BUFFER
4531 ;
4532 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
4533 ;
4535 026330 .=<..10>&177770
4537 026330 T29PK2: .WORD 100006 ;WRITE SUB SYS MEM COMMAND, AND ACK
4538 026330 100006 ;ADDRESS OF SELECT BLOCK DATA
4539 026332 026350 .WORD T29BF2
4540 026334 000000 .WORD 0
4541 026336 000006 .WORD 6. ;SIZE OF DATA PACKET
4542 ;
4546 026340 T29PK3: .WORD 140005 ;WRITE TAPE MARK RETRY COMMAND, CVC=1 AND ACK
4547 026340 140005
4548 026342 T29RB: .WORD FREE ;ADDRESS OF WRITE BUFFER
4549 026342 003116 T29WB: .WORD 0
4550 026344 000000 .WORD 0
4551 026346 000000 T29SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
4552 .EVEN
4553 ;
4554 ;
4555 ;
4556 026350 T29BF2:
4557 026350 010 T29BS0: .BYTE 10 ;BSELO AREA
4558 026351 200 T29BS1: .BYTE 200 ;BSEL1 AREA
4559 026352 000000 T29S2: .WORD 0 ;SEL 2 AREA
4560 026354 000000 T29S3: .WORD 0 ;DATA AREA
4561 ;
4562 ;
4563 .EVEN
4564 ;TAPE MOTION PACKET COMMAND VALUES
4565 ;
4566 026356 140001 T29RN: .WORD 140001 ;READ DATA
4567 026360 140401 T29WDR: .WORD 140401 ;READ DATA REVERSE
4568 026362 141001 T29CON: .WORD 141001 ;READ PREVIOUS OPP=0
4569 026364 161001 .WORD 161001 ;READ PREVIOUS OPP=1
4570 026366 141401 .WORD 141401 ;WRITE TAPE MARK RETRY NEXT OPP=0
4571 026370 161401 .WORD 161401 ;WRITE TAPE MARK RETRY NEXT OPP=1
4572 026372 177777 .WORD 177777 ;END OF DATA
4573 ;
4574 ;
4575 026374 000000 T29CNT: .WORD 0 ;TAPE RECORD COUNTER STORAGE AREA
4576 ;
4577 026376 000000 T29RSZ: .WORD 0 ;RECORD STORAGE SIZE AREA
4578 026400 000000 T29DLY: .WORD ;DELAY COUNTER STORAGE AREA
4579 ;
4580 ;*
4581 ;LOCAL TEXT MESSAGES FOR TEST
4582 ;-
4583 026402 104 162 151 T29OFL: .ASCIZ 'Drive is OFFLINE'
4584 026423 124 141 160 T29WNG: .ASCIZ 'Tape Position Incorrect After WRITE TAPE MARK RETRY Previous (OPP=1)'
4585 026530 127 122 111 T29NEF: .ASCIZ 'WRITE TAPE MARK RETRY, At BOT, Failed To Set NEF (XSTO)'
4586 026620 124 123 123 T29RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
4587 026667 127 122 111 T29RRF: .ASCIZ 'WRITE TAPE MARK RETRY Previous (Space Reverse, Read Forward) Command Failed'
4588 027003 127 122 111 T29RRG: .ASCIZ 'WRITE TAPE MARK RETRY Previous (Read Forward, Space Reverse) Command Failed'
4589 027117 120 117 123 T29SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
4590 027201 122 111 102 T29LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'

```



TEST 1: WRITE TAPE MARK RETRY

```

4648 032144 012701 026330      MOV      #T29PK2,R1          ;START OF THE PACKET
4649 032150 012721 140006      MOV      #140006,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1.
4650 032154 012721 026350      MOV      #T29BF2,(R1)+     ;ADDRESS OF DATA BLOCK
4651 032160 005021              CLR      (R1)+              ;EXTENDED ADDRESS
4652 032162 012721 000006      MOV      #6,(R1)+          ;SIZE OF DATA BLOCK IN BYTES
4653 032166 005021              CLR      (R1)+
4654 032170 012701 026350      MOV      #T29BF2,R1        ;POINT TO DATA SEL AREA
4655 032174 005021              CLR      (R1)+
4656 032176 005011              CLR      (R1)
4657 032200 000207              RTS      PC                  ;RETURN
4658 032202
4659 032202
4660 032206 012701 026340      SAVREG                    ;SAVE THE REGISTERS
4661 032212 012721 000000      MOV      #T29PK3,R1        ;START OF THE PACKET
4662 032216 012721 000000      MOV      #0,(R1)+          ;WRITE SUBSYSTEM MEM. WITH ACK.
4663 032222 005021              MOV      #0,(R1)+          ;ADDRESS OF DATA BLOCK
4664 032224 012711 000000      CLR      (R1)+            ;EXTENDED ADDRESS
4665 032230 000207              MOV      #0,(R1)          ;SIZE OF DATA BLOCK IN BYTES
4666 032232 000207              RTS      PC                  ;RETURN
                                ENDTST
                                L10036: TRAP C#ETST
032232 104401

```

.SBTTL TEST 2: SKIP TAPE MARKS

```

4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685

```

;+  
; THIS TEST VERIFIES PROPER OPERATION OF THE SKIP TAPE MARKS  
; FORWARD AND SKIP TAPE MARKS REVERSE COMMANDS. PROPER OPERATION  
; UNDER CONTROL OF ALL COMBINATIONS OF THE ENABLE SKIP TAPE MARKS  
; STOP (ESS) AND ENABLE TAPE MARKS STOP OFF BOT (ENB) BITS SPECIFIED  
; BY THE WRITE CHARACTERISTICS COMMAND. THE TEST CONSISTS OF THE  
; FOLLOWING SUBTESTS (FOR EACH SUBTEST, THE TAPE IS FIRST WRITTEN  
; WITH AN APPROPRIATE SERIES OF DATA RECORDS AND/OR TAPE MARKS  
; AND/OR DOUBLE TAPE MARKS.

THE TEST CONSISTS OF THE FOLLOWING 11 SUBTESTS

```

4685 032234 032234              BGNTST
4686 032234 012737 006354 002172      MOV      #EPRT1,EPRTSW     T2::
4691 032242 012700 041131              MOV      #TST30ID,R0      ;PRIMARY ERROR MESSAGE
4692 032246 004737 016570              JSR      PC,TSTSETUP      ;ASCII MESSAGE TO IDENTIFY TEST
4693 032252 012737 000005 002210      MOV      #5,LOOPCNT       ;DO INITIAL TEST SETUP
                                ;PERFORM 5 ITERATIONS

```

```

4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705

```

;+  
; TEST 2. SUBTEST 1  
; VERIFIES THAT A SKIP TAPE MARKS FORWARD COMMAND WITH  
; A TAPE MARK COUNT OF 1 OPERATES PROPERLY. THE TAPE  
; IS FIRST REWOUND, THEN WRITTEN WITH SEVERAL "FILES";  
; EACH FILE CONSISTS OF A NUMBER OF DATA RECORDS  
; FOLLOWED BY A TAPE MARK. THE FINAL FILE IS  
; TERMINATED BY A DOUBLE TAPE MARK. EACH DATA RECORD  
; CONTAINS A FILE NUMBER AND THE RECORD NUMBER WITHIN



TEST 2: SKIP TAPE MARKS

```

4755 032416 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
4756 032420          ERRHRD  ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTIC
      032420 104456          TRAP   C:ERHRD
      032422 000312          .WORD 202
      032424 005052          .WORD WRTMSG
      032426 012114          .WORD SFMSG
4757 032430          23:   CKLOOP          ;LOOP IF SELECTED
      032430 104406          TRAP   C:CLP1
4758
4759          ;*****
4760          ;
4761          ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
4762          ;
4763          ;*****
4764
4765 032432 004737 011074    JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
4766 032436 103411          BCS    30:           ;BR, IF NO PROBLEM
4767 032440 010004          MOV    R0,R4          ;GET PACKET ADDRESS
4768 032442 C16501 000002    MOV    TSSP(R5),R1   ;GET STATUS REGISTER
4769 032446 005237 002214    INC    FATFLG        ;ERROR COUNT
4773 032452          ERRHRD  ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
      032452 104456          TRAP   C:ERHRD
      032454 000313          .WORD 203
      032456 040140          .WORD T30RWN
      032460 012126          .WORD PKTSSR
4774 032462          30:   CKLOOP          ;LOOP IF SELECTED
      032462 104406          TRAP   C:CLP1
4775
4776          ;*****
4777          ;
4778          ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
4779          ;
4780          ;*****
4781
4782 032464 013701 036430    MOV    T30BFR+6,R1   ;PICK UP XSTO
4783 032470 010102          MOV    R1,R2          ;SET UP EXPECTED
4784 032472 052702 000002    BIS    @BIT1,R2      ;SET BOT BIT IN EXPECTED
4785 032476 020102          CMP    R1,R2          ;DOES EXP = REC'D
4786 032500 001406          BEQ    40:           ;BR, IF EQUAL (OK)
4787 032502 005237 002214    INC    FATFLG        ;ERROR COUNT
4791 032506          ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      032506 104456          TRAP   C:ERHRD
      032510 000314          .WORD 204
      032512 037741          .WORD T30BOT
      032514 015554          .WORD EXPREC
4792 032516          40:   CKLOOP          ;LOOP IF SELECTED
      032516 104406          TRAP   C:CLP1
4793 032520 012737 000001 036554    MOV    @1.,T30FCN    ;SET "FILE" COUNTER AT 1 DECIMAL
4794 032526 012703 000001          MOV    @1,R3          ;ONE RECORD PER "FILE"
4795 032532 013737 003116 036522 64:   MOV    FREE,T30WB    ;SET UP PACKETS'S WRITE BUFFER
4796 032540 012737 003720 036526 65:   MOV    @2000.,T30SZ ;SET RECORD SIZE AT 2000 BYTES
4797
4798          ;*****
4799          ;
4800          ;WRITE DATA,ACK,CVC-1 COMMAND
4801          ;
4802          ;*****

```

TEST 2: SKIP TAPE MARKS

```

4803
4804 032546 012737 140005 036520      MOV      #140005,T30PK3      ;WRITE DATA,ACK,CVC-1 COMMAND
4805 032554 012704 036520              MOV      #T30PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4806 032560 013702 036554              MOV      T30FCN,R2        ;GET FILE COUNTER
4807 032564 000302                      SWAB     R2                ;MOVE TO UPPER BYTE
4808 032566 010301                      MOV      R3,R1            ;GET RECORD COUNTER
4809 032570 060201                      ADD      R2,R1            ;FILE COUNTER IN UPPER, RECORD # LOW
4810 032572 010177 150320              MOV      R1,#FREE         ;MOV TO OUT PUT BUFFER
4811 032576 010465 000000              MOV      R4,TSDB(R5)      ;ISSUE COMMAND
4812 032602 004737 016330              JSR      PC,WAITF         ;WAIT FOR SSR TO SET
4813 032606 016501 000002              MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
4814 032612 012702 000200              MOV      #SSR,R2         ;SET UP EXPECTED
4815 032616 020102                      CMP      R1,R2            ;ARE THEY EQUAL
4816 032620 001406                      BEQ      70#              ;BR, IF OK
4817 032622 005237 002214              INC      FATFLG           ;ERROR COUNT
4821 032626                      ERRHRD  ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C#ERRRD
                                .WORD   205
                                .WORD   T30WDD
                                .WORD   PKTSSR
                                TRAP    C#CLP1
4822 032636 104456                      70# :   CKLOOP           ;LOOP IF SELECTED
                                TRAP    C#CLP1
                                .WORD   205
                                .WORD   T30WDD
                                .WORD   PKTSSR
4823 032640 005203                      INC      R3                ;COUNT THE RECORD COUNTER DOWN
4824 032642 020327 000021              CMP      R3,#21          ;AT 20 YET
4825 032646 001331                      BNE     65#              ;BR, IF NOT AT 20 RECORDS WRITTEN
4826
4827 ;*****
4828 ;
4829 ;WRITE TAPE MARK,ACK,CVC-1 COMMAND
4830 ;
4831 ;*****
4832
4833 032650 012737 141011 036520      MOV      #141011,T30PK3   ;WRITE TAPE MARK,ACK,CVC-1 COMMAND
4834 032656 012704 036520              MOV      #T30PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4835 032662 010465 000000              MOV      R4,TSDB(R5)     ;ISSUE COMMAND
4836 032666 004737 016330              JSR      PC,WAITF         ;WAIT FOR SSR TO SET
4837 032672 016501 000002              MOV      TSSR(R5),R1     ;PICK UP TSSR
4838 032676 012702 000200              MOV      #SSR,R2         ;SET UP EXPECTED (SSR ONLY)
4839 032702 020102                      CMP      R1,R2            ;WAS STATUS GOOD
4840 032704 001406                      BEQ      160#            ;BR, IF TERMINATION WAS GOOD
4841 032706 005237 002214              INC      FATFLG           ;ERROR COUNT
4845 032712                      ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
                                TRAP    C#ERRRD
                                .WORD   206
                                .WORD   T30WDC
                                .WORD   PKTSSR
                                TRAP    C#CLP1
4846 032722 104406                      160# :  CKLOOP           ;LOOP IF SELECTED
                                TRAP    C#CLP1
                                .WORD   206
                                .WORD   T30WDC
                                .WORD   PKTSSR
4847 032724 005237 036554              INC      T30FCN           ;COUNT THE "FILE" COUNTER DOWN
4848 032730 023727 036554 000006              CMP      T30FCN,#6       ;WRITE 5 FILE TO TAPE
4849 032736 001273                      BNE     64#              ;BR, IF NOT AT 5 FILES WRITTEN
4850
4851 ;*****
4852 ;
4853 ;WRITE TAPE MARK,ACK,CVC-1 COMMAND
4854 ;
4855 ;*****

```

TEST 2: SKIP TAPE MARKS

```

4856
4857 032740 012737 141011 036520      MOV      #141011,T30PK3      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
4858 032746 012704 036520      MOV      #T30PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4859 032752 010465 000000      MOV      R4,TSDB(R5)       ;ISSUE COMMAND
4860 032756 004737 016330      JSR      PC,WAITF         ;WAIT FOR SSR TO SET
4861 032762 016501 000002      MOV      TSSR(R5),R1      ;PICK UP TSSR
4862 032766 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED (SSR ONLY)
4863 032772 020102      CMP      R1,R2           ;WAS STATUS GOOD
4864 032774 001406      BEQ      165$            ;BR, IF TERMINATION WAS GOOD
4865 032776 005237 002214      INC      FATFLG          ;ERROR COUNT
4869 033002      ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
                                TRAP      C#ERHRD
                                .WORD    207
                                .WORD    T30WDC
                                .WORD    PKTSSR
4870 033012      165$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C#CLP1
                                .WORD    104406
4871
4872      ;*****
4873      ;
4874      ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
4875      ;
4876      ;*****
4877
4878 033014 004737 011074      JSR      PC,REWIND        ;CALL TAPE REWIND COMMAND
4879 033020 103411      BCS     170$            ;BR, IF NO PROBLEM
4880 033022 010004      MOV      RO,R4           ;GET PACKET ADDRESS
4881 033024 016501 000002      MOV      TSSR(R5),R1      ;GET STATUS REGISTER
4882 033030 005237 002214      INC      FATFLG          ;ERROR COUNT
4886 033034      ERRHRD  ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C#ERHRD
                                .WORD    208
                                .WORD    T30RWN
                                .WORD    PKTSSR
4887 033044      170$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C#CLP1
                                .WORD    104406
4888
4889      ;*****
4890      ;
4891      ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
4892      ;
4893      ;*****
4894
4895 033046 013701 036430      MOV      T30BFR+6,R1     ;PICK UP XSTO
4896 033052 010102      MOV      R1,R2           ;SET UP EXPECTED
4897 033054 052702 000002      BIS     #BIT1,R2        ;SET BOT BIT IN EXPECTED
4898 033060 020102      CMP      R1,R2           ;DOES EXP = REC'D
4899 033062 001406      BEQ     180$            ;BR, IF EQUAL (OK)
4900 033064 005237 002214      INC      FATFLG          ;ERROR COUNT
4904 033070      ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C#ERHRD
                                .WORD    209
                                .WORD    T30BOT
                                .WORD    EXPREC
4905 033100      180$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C#CLP1
                                .WORD    104406
4906 033102 012703 036536      MOV      #T30IMV,R3      ;SET UP POINTER TO COMMAND TABLE

```



TEST 2: SKIP TAPE MARKS

```

4907 033106 013737 002174 036420      MOV      UNITN,T30DSW      ;SET UP UNIT NUMBER
4908 033114 011337 036416      182$:  MOV      (R3),T30ETM  ;GET NEXT COMMAND
4909 033120 012704 036400      MOV      @T30PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
4910
4911      ;*****
4912      ;
4913      ;ISSUE WRITE CHARACTERISTICS COMMAND
4914      ;
4915      ;*****
4916
4917 033124 004737 010742      JSR      PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
4918 033130 103407      BCS      188$            ;BR, IF COMMAND ISSUED OK
4919 033132 005237 002214      INC      FATFLG          ;ERROR COUNT
4923 033136 010001      MOV      R0,R1           ;SAVE CONTENTS OF TSSR
4924 033140      ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP      C#ERHRD
                                .WORD    210
                                .WORD    WRTMSG
                                .WORD    SFIMSG
4925 033150      188$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C#CLP1
4926
4927      ;*****
4928      ;
4929      ;SKIP TAPE MARK.ACK,CVC=1 COMMAND
4930      ;
4931      ;*****
4932
4933 033152 012737 141010 036520      MOV      @141010,T30PK3   ;SKIP TAPE MARK.ACK,CVC=1 COMMAND
4934 033160 012737 000001 036522      MOV      @1,T30RB        ;SET UP NUMBER TO SKIP
4935 033166 012704 036520      MOV      @T30PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4936 033172 010465 000000      189$:  MOV      R4,T30DB(R5) ;ISSUE COMMAND
4937 033176 012737 176750 036556      MOV      @65000.,T30DLY   ;SET UP DELAY COUNTER
4938 033204 004737 016330      190$:  JSR      PC,WAITF    ;WAIT FOR SSR TO SET
4939 033210 016501 000002      MOV      TSSR(R5),R1     ;PICK UP TSSR
4940 033214 032701 000200      BIT      @SSR,R1         ;IS SSR SET YET
4941 033220 001017      BNE     191$            ;BR, IF SSR IS SET
4942 033222      DELAY  250           ;CALL DELAY ROUTINE
                                MOV      @250,(PC)+
                                .WORD    0
                                MOV      L#DLY,(PC)+
                                .WORD    0
                                DEC      -6(PC)
                                BNE     -.4
                                DEC      -22(PC)
                                BNE     -.20
4943 033252 005337 036556      DEC      T30DLY          ;BUMP DELAY ROUTINE
4944 033256 001352      BNE     190$            ;BR, IF MORE DELAY TO GO
4945 033260 012702 000200      191$:  MOV      @SSR,R2     ;SET UP EXPECTED (SSR ONLY)
4946 033264 020102      CMP     R1,R2           ;WAS STATUS GOOD
4947 033266 001406      BEQ     192$            ;BR, IF TERMINATION WAS GOOD
4948 033270 005237 002214      INC     FATFLG          ;ERROR COUNT
4952 033274      ERRHRD  ERRNO,T30SKM,PKTSSR ;TSSR NOT CORRECT AFTER SKIP TAPE M.
                                TRAP      C#ERHRD
                                .WORD    211
                                .WORD    T30SKM
                                .WORD    PKTSSR
033274 104456
033276 000323
033300 037014
033302 012126

```

TEST 2: SKIP TAPE MARKS

```

4953 033304      192$:  CKLOOP                ;LOOP IF SELECTED                TRAP  C#CLP1
      033304  104406
4954
4955 ;*****
4956 ;
4957 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
4958 ;
4959 ;*****
4960
4961 033306  013701  036430      MOV      T30BFR+6,R1      ;PICK UP XSTO
4962 033312  010102      MOV      R1,R2           ;SET UP EXPECTED
4963 033314  052702  100000      BIS      @BIT15,R2      ;SET TMK BIT IN EXPECTED
4964 033320  020102      CMP      R1,R2          ;DOES EXP = REC'D
4965 033322  001406      BEQ      195$           ;BR, IF EQUAL (OK)
4966 033324  005237  002214      INC      FATFLG         ;ERROR COUNT
4970 033330      ERRHRD  ERRNO,T30TMK,EXPREC ;TMK NOT SET AFTER WRT TAPE MARK
      033330  104456      TRAP    C#ERHRD
      033332  000324      .WORD  212
      033334  C40414      .WORD  T30TMK
      033336  015554      .WORD  EXPREC
4971 033340      195$:  CKLOOP                ;LOOP IF SELECTED                TRAP  C#CLP1
      033340  104406
4972 033342  012700  177777      MOV      @177777,R0     ;VALUE TO WRITTEN TO MEMORY
4973 033346  004737  017502      JSR      PC,FILLMEM     ;FILL MEM WITH ALL ONES
4974 033352  013737  003116  036522      MOV      FREE,T30RB     ;STARTING READ BUFFER ADDRESS
4975
4976 ;*****
4977 ;
4978 ;READ FORWARD,ACK,CVC-1 COMMAND
4979 ;
4980 ;*****
4981
4982 033360  012737  140001  036520      MOV      @140001,T30PK3 ;READ FORWARD,ACK,CVC-1 COMMAND
4983 033366  012704  036520      MOV      @T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
4984 033372  012737  003720  036526      MOV      @2000,T30SZ    ;SET UP RECORD SIZE IN PACKET
4985 033400  010465  000000      MOV      R4,TSD8(R5)    ;ISSUE COMMAND
4986 033404  004737  016330      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
4987 033410  016501  000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
4988 033414  012702  000200      MOV      @SSR,R2        ;SET UP EXPECTED
4989 033420  020102      CMP      R1,R2          ;ARE THEY EQUAL
4990 033422  001406      BEQ      200$           ;BR, IF OK
4991 033424  005237  002214      INC      FATFLG         ;ERROR COUNT
4995 033430      ERRHRD  ERRNO,T30RDF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      033430  104456      TRAP    C#ERHRD
      033432  000325      .WORD  213
      033434  037313      .WORD  T30RDF
      033436  012126      .WORD  PKTSSR
4996 033440      200$:  CKLOOP                ;LOOP IF SELECTED                TRAP  C#CLP1
      033440  104406
4997 033442  017701  147450      MOV      @FREE,R1       ;FIRST LOC IN READ BUFFER
4998 033446  012702  177777      MOV      @177777,R2     ;EXPECTED IF NO DATA TRANS.
4999 033452  020102      CMP      R1,R2          ;DID ANY DATA GET TRANSFERRED
5000 033454  001006      BNE      220$           ;BR, IF NO DATA TRANS (GOOD)
5001 033456  005237  002214      INC      FATFLG         ;ERROR COUNT
5005 033462      ERRHRD  ERRNO,T30DTR,EXPREC ;DATA TRANSFERRED ON READ TAPE MARK
      033462  104456      TRAP    C#ERHRD
      033464  000326      .WORD  214
    
```

TEST 2: SKIP TAPE MARKS

```

033466 040770 .WORD T30DTR
033470 015554 .WORD EXPREC
5006 033472 220$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
033472 104406 ;SET UP RECORD NUMBER EXPECTED (FILE 2)
5007 033474 012702 001001 MOV #1001,R2 ;GET INFO FROM BUFFER
5008 033500 017701 147412 MOV @FREE,R1 ;ARE THEY EQUAL
5009 033504 020201 CMP R2,R1 ;BR, IF EQUAL (OK)
5010 033506 001406 BEQ 228$ ;ERROR COUNT
5011 033510 005237 002214 INC FATFLG ;RECORD POSITION WAS NOT CORRECT
5015 033514 ERRHRD ERRNO,T30PTB,EXPREC ;TRAP C#ERHRD
033514 104456 .WORD 215
033516 000327 .WORD T30PTB
033520 037142 .WORD EXPREC
033522 015554
5016 033524 228$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
033524 104406
5017
5018 ;*****
5019 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
5020 ;
5021 ;*****
5022
5023
5024 033526 004737 011074 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
5025 033532 103411 BCS 230$ ;BR, IF NO PROBLEM
5026 033534 010004 MOV R0,R4 ;SAVE PACKET ADDRESS
5027 033536 016501 000002 MOV TSSR(R5),R1 ;GET TSSR STATUS
5028 033542 005237 002214 INC FATFLG ;ERROR COUNT
5032 033546 ERRHRD ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
033546 104456 TRAP C#ERHRD
033550 000330 .WORD 216
033552 040140 .WORD T30RWN
033554 012126 .WORD PKTSSR
5033 033556 230$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
033556 104406
5034
5035 ;*****
5036 ;GET EXTENDED STATUS REGISTER ZERO (XST0) FROM MESSAGE BUFFER
5037 ;
5038 ;*****
5039
5040
5041 033560 013701 036430 MOV T30BFR+6,R1 ;PICK UP XST0
5042 033564 010102 MOV R1,R2 ;SET UP EXPECTED
5043 033566 052702 000002 BIS @BIT1,R2 ;SET BOT BIT IN EXPECTED
5044 033572 020102 CMP R1,R2 ;DOES EXP = REC'D
5045 033574 001406 BEQ 240$ ;BR, IF EQUAL (OK)
5046 033576 005237 002214 INC FATFLG ;ERROR COUNT
5050 033602 ERRHRD ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
033602 104456 TRAP C#ERHRD
033604 000331 .WORD 217
033606 037741 .WORD T30BOT
033610 015554 .WORD EXPREC
5051 033612 240$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
033612 104406
5052 033614 005723 TST (R3)+ ;POINT TO NEXT POSITION
5053 033616 011301 MOV (R3),R1 ;GET NEXT COMMAND ETC.

```



TEST 2: SKIP TAPE MARKS

```

5097 ;*****
5098 ;
5099 ;ISSUE WRITE CHARACTERISTICS COMMAND
5100 ;
5101 ;*****
5102
5103 033776 004737 010742 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
5104 034002 103407 BCS 23$ ;BR, IF COMMAND ISSUED OK
5105 034004 005237 002214 INC FATFLG ;ERROR COUNT
5109 034010 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
5110 034012 ERRHRD ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTIC FAILED
; TRAP C$ERHRD
; .WORD 219
; .WORD WRTMSG
; .WORD SFMSG
5111 034022 23$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
; 034022 104406
5112 ;*****
5113 ;
5114 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
5115 ;
5116 ;*****
5117 ;
5118 ;*****
5119 034024 004737 011074 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
5120 034030 103411 BCS 30$ ;BR, IF NO PROBLEM
5121 034032 010004 MOV R0,R4 ;GET PACKET ADDRESS
5122 034034 016501 000002 MOV TSSR(R5),R1 ;GET STATUS REGISTER
5123 034040 005237 002214 INC FATFLG ;ERROR COUNT
5127 034044 ERRHRD ERRNO,T3ORWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERHRD
; .WORD 220
; .WORD T3ORWN
; .WORD PKTSSR
5128 034054 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
; 034054 104406
5129 ;*****
5130 ;
5131 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
5132 ;
5133 ;*****
5134 ;
5135 ;*****
5136 034056 013701 036430 MOV T30BFR+6,R1 ;PICK UP XSTO
5137 034062 010102 MOV R1,R2 ;SET UP EXPECTED
5138 034064 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
5139 034070 020102 CMP R1,R2 ;DOES EXP = REC'D
5140 034072 001406 BEQ 40$ ;BR, IF EQUAL (OK)
5141 034074 005237 002214 INC FATFLG ;ERROR COUNT
5145 034100 ERRHRD ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
; TRAP C$ERHRD
; .WORD 221
; .WORD T30BOT
; .WORD EXPREC
5146 034110 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
; 034110 104406
5147 034112 012737 000001 036554 MOV #1.,T30FCN ;SET "FILE" COUNTER AT 1 DECIMAL

```

TEST 2: SKIP TAPE MARKS

```

5148 034120 012703 000001      64$:  MOV      #1,R3          ;ONE RECORD PER "FILE"
5149 034124 013737 003116      65$:  MOV      FREE,T30WB      ;SET UP PACKETS'S WRITE BUFFER
5150 034132 012737 000024      036522  MOV      #20.,T30SZ        ;SET RECORD SIZE AT 2000 BYTES
5151
5152      ;*****
5153      ;
5154      ;WRITE DATA,ACK,CVC=1 COMMAND
5155      ;
5156      ;*****
5157
5158 034140 012737 140005      036520      MOV      #140005,T30PK3      ;WRITE DATA,ACK,CVC=1 COMMAND
5159 034146 012704 036520      MOV      #T30PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
5160 034152 013702 036554      MOV      T30FCN,R2        ;GET FILE COUNTER
5161 034156 000302      SWAB     R2                ;MOVE TO UPPER BYTE
5162 034160 010301      MOV      R3,R1            ;GET RECORD COUNTER
5163 034162 060201      ADD     R2,R1            ;FILE COUNTER IN UPPER, RECORD # LOW
5164 034164 010177 146726      MOV      R1,#FREE        ;MOV TO OUT PUT BUFFER
5165 034170 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
5166 034174 C04737 016330      JSR     PC,WAITF         ;WAIT FOR SSR TO SET
5167 034200 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
5168 034204 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED
5169 034210 020102      CMP     R1,R2            ;ARE THEY EQUAL
5170 034212 001406      BEQ     70$              ;BR, IF OK
5171 034214 005237 002214      INC     FATFLG           ;ERROR COUNT
5175 034220      ERRHRD  ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      TRAP  C#ERHRD
      .WORD 222
      .WORD T30WDD
      .WORD PKTSSR
5176 034230      70$:  CKLOOP              ;LOOP IF SELECTED
      TRAP  C#CLP1
5177 034232 005203      INC     R3                ;COUNT THE RECORD COUNTER DOWN
5178 034234 020327 000021      CMP     R3,#21           ;AT 20 YET
5179 034240 001331      BNE     65$              ;BR, IF NOT AT 20 RECORDS WRITTEN
5180
5181      ;*****
5182      ;
5183      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
5184      ;
5185      ;*****
5186
5187 034242 012737 141011      036520      MOV      #141011,T30PK3    ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
5188 034250 012704 036520      MOV      #T30PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
5189 034254 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
5190 034260 004737 016330      JSR     PC,WAITF         ;WAIT FOR SSR TO SET
5191 034264 016501 000002      MOV      TSSR(R5),R1      ;PICK UP TSSR
5192 034270 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED (SSR ONLY)
5193 034274 020102      CMP     R1,R2            ;WAS STATUS GOOD
5194 034276 001406      BEQ     160$             ;BR, IF TERMINATION WAS GOOD
5195 034300 005237 002214      INC     FATFLG           ;ERROR COUNT
5199 034304      ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
      TRAP  C#ERHRD
      .WORD 223
      .WORD T30WDC
      .WORD PKTSSR
5200 034314      160$: CKLOOP              ;LOOP IF SELECTED
      TRAP  C#CLP1
034314 104406

```

TEST 2: SKIP TAPE MARKS

```

5201 034316 005237 036554          INC      T30FCN          ;COUNT THE "FILE" COUNTER DOWN
5202 034322 023727 036554 000031  CMP      T30FCN,#25.    ;WRITE 25 FILES TO TAPE
5203 034330 001273                BNE      64$           ;BR, IF NOT AT 25 FILES WRITTEN
5204
5205 ;*****
5206 ;
5207 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
5208 ;
5209 ;*****
5210
5211 034332 012737 141011 036520      MOV      #141011,T30PK3 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
5212 034340 012704 036520          MOV      #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
5213 034344 010465 000000          MOV      R4,TSD8(R5)    ;ISSUE COMMAND
5214 034350 004737 016330          JSR      PC,WAITF       ;WAIT FOR SSR TO SET
5215 034354 016501 000002          MOV      TSSR(R5),R1   ;PICK UP TSSR
5216 034360 012702 000200          MOV      #SSR,R2       ;SET UP EXPECTED (SSR ONLY)
5217 034364 020102                CMP      R1,R2          ;WAS STATUS GOOD
5218 034366 001406                BEQ      165$           ;BR, IF TERMINATION WAS GOOD
5219 034370 C05237 002214          INC      FATFLG         ;ERROR COUNT
5223 034374                ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
                                TRAP      C#ERHRD
                                .WORD    224
                                .WORD    T30WDC
                                .WORD    PKTSSR
5224 034404                165$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C#CLP1
5225
5226 ;*****
5227 ;
5228 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
5229 ;
5230 ;*****
5231
5232 034406 004737 011074          JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
5233 034412 103411                BCS      170$           ;BR, IF NO PROBLEM
5234 034414 010004                MOV      R0,R4         ;GET PACKET ADDRESS
5235 034416 016501 000002          MOV      TSSR(R5),R1   ;GET STATUS REGISTER
5236 034422 005237 002214          INC      FATFLG         ;ERROR COUNT
5240 034426                ERRHRD  ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C#ERHRD
                                .WORD    225
                                .WORD    T30RWN
                                .WORD    PKTSSR
5241 034436                170$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C#CLP1
5242
5243 ;*****
5244 ;
5245 ;GET EXTENDED STATUS REGISTER ZERO (XST0) FROM MESSAGE BUFFER
5246 ;
5247 ;*****
5248
5249 034440 013701 036430          MOV      T30BFR+6,R1   ;PICK UP XST0
5250 034444 010102                MOV      R1,R2         ;SET UP EXPECTED
5251 034446 052702 000002          BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
5252 034452 020102                CMP      R1,R2         ;DOES EXP = REC'D
5253 034454 001406                BEQ      180$           ;BR, IF EQUAL (OK)

```

TEST 2: SKIP TAPE MARKS

```

5254 034456 005237 002214          INC      FATFLG          ;ERROR COUNT
5258 034462          ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      034462 104456          TRAP      C#ERHRD
      034464 000342          .WORD    226
      034466 037741          .WORD    T30BOT
      034470 015554          .WORD    EXPREC
5259 034472          180#:  CKLOOP          ;LOOP IF SELECTED          TRAP      C#CLP1
      034472 104406
5260 034474 012737 000002 036554      MOV      #2,T30FCN      ;SET TO NUMBER OF SKIP "FILES"
5261 034502 012703 036536          MOV      #T30IMV,R3     ;SET UP POINTER TO COMMAND TABLE
5262 034506 013737 002174 036420      MOV      UNITN,T30DSW   ;SET UP UNIT NUMBER
5263 034514 011337 036416 182#:  MOV      (R3),T30ETM ;GET NEXT COMMAND
5264 034520 012704 036400      MOV      #T30PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
5265
5266          ;*****
5267          ;
5268          ;ISSUE WRITE CHARACTERISTICS COMMAND
5269          ;
5270          ;*****
5271
5272 034524 004737 010742          JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
5273 034530 103407          BCS     188#           ;BR, IF COMMAND ISSUED OK
5274 034532 005237 002214          INC      FATFLG          ;ERROR COUNT
5278 034536 010001          MOV      R0,R1         ;SAVE CONTENTS OF TSSR
5279 034540          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      034540 104456          TRAP      C#ERHRD
      034542 000343          .WORD    227
      034544 005052          .WORD    WRTMSG
      034546 012114          .WORD    SFIMSG
5280 034550          188#:  CKLOOP          ;LOOP IF SELECTED          TRAP      C#CLP1
      034550 104406
5281
5282          ;*****
5283          ;
5284          ;SKIP TAPE MARK,ACK,CVC=1 COMMAND
5285          ;
5286          ;*****
5287
5288 034552 012737 141010 036520      MOV      #141010,T30PK3 ;SKIP TAPE MARK,ACK,CVC=1 COMMAND
5289 034560 013737 036554 036522      MOV      T30FCN,T30RB   ;SET UP NUMBER TO SKIP
5290 034566 012704 036520          MOV      #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
5291 034572 010465 000000 189#:  MOV      R4,TSDB(R5)  ;ISSUE COMMAND
5292 034576 012737 176750 036556      MOV      #65000,T30DLY  ;SET UP DELAY COUNTER
5293 034604 004737 016330 190#:  JSR      PC,WAITF      ;WAIT FOR SSR TO SET
5294 034610 016501 000002          MOV      TSSR(R5),R1    ;PICK UP TSSR
5295 034614 032701 000200          BIT      #SSR,R1        ;IS SSR SET YET
5296 034620 001017          BNE     191#           ;BR, IF SSR IS SET
5297 034622          DELAY  250           ;CALL DELAY ROUTINE
      034622 012727 000250          MOV      #250,(PC)+
      034626 000000          .WORD    0
      034630 013727 002116          MOV      L#DLY,(PC)+
      034634 000000          .WORD    0
      034636 005367 177772          DEC      -6(PC)
      034642 001375          BNE     -4
      034644 005367 177756          DEC      -22(PC)
      034650 001367          BNE     -20
5298 034652 005337 036556          DEC      T30DLY        ;BUMP DELAY ROUTINE

```



TEST 2: SKIP TAPE MARKS

```

5299 034656 001352
5300 034660 012702 000200
5301 034664 020102
5302 034666 001406
5303 034670 005237 002214
5307 034674
    034674 104456
    034676 000344
    034700 037014
    034702 012126
5308 034704
    034704 104406
5309
5310
5311
5312
5313
5314
5315
5316 034706 013701 036430
5317 034712 010102
5318 034714 052702 100000
5319 034720 020102
5320 034722 001406
5321 034724 005237 002214
5325 034730
    034730 104456
    034732 000345
    034734 040414
    034736 015554
5326 034740
    034740 104406
5327 034742 012700 177777
5328 034746 004737 017502
5329 034752 013737 003116 036522
5330
5331
5332
5333
5334
5335
5336
5337 034760 012737 140001 036520
5338 034766 012704 036520
5339 034772 012737 000024 036526
5340 035000 010465 000000
5341 035004 004737 016330
5342 035010 016501 000002
5343 035014 012702 000200
5344 035020 020102
5345 035022 001406
5346 035024 005237 002214
5350 035030
    035030 104456
    035032 000346
    035034 037313
    035036 012126

    191$: BNE 190$ ;BR, IF MORE DELAY TO GO
          MOV #SSR,R2 ;SET UP EXPECTED (SSR ONLY)
          CMP R1,R2 ;WAS STATUS GOOD
          BEQ 192$ ;BR, IF TERMINATION WAS GOOD
          INC FATFLG ;ERROR COUNT
          ERRHRD ERRNO,T30SKM,PKTSSR ;TSSR NOT CORRECT AFTER SKIP TAPE M.
                                     TRAP C#ERHRD
                                     .WORD 228
                                     .WORD T30SKM
                                     .WORD PKTSSR

    192$: CKLOOP ;LOOP IF SELECT'D
                                     TRAP C#CLP1

;*****
;
;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
;
;*****

          MOV T30BFR+6,R1 ;PICK UP XSTO
          MOV R1,R2 ;SET UP EXPECTED
          BIS #BIT15,R2 ;SET TMK BIT IN EXPECTED
          CMP R1,R2 ;DOES EXP = REC'D
          BEQ 195$ ;BR, IF EQUAL (OK)
          INC FATFLG ;ERROR COUNT
          ERRHRD ERRNO,T30TMK,EXPREC ;TMK NOT SET AFTER WRT TAPE MARK
                                     TRAP C#ERHRD
                                     .WORD 229
                                     .WORD T30TMK
                                     .WORD EXPREC

    195$: CKLOOP ;LOOP IF SELECTED
                                     TRAP C#CLP1

          MOV #177777,R0 ;VALUE TO WRITTEN TO MEMORY
          JSR PC,FILLMEM ;FILL MEM WITH ALL ONES
          MOV FREE,T30RB ;STARTING READ BUFFER ADDRESS

;*****
;
;READ FORWARD,ACK,CVC=1 COMMAND
;
;*****

          MOV #140001,T30PK3 ;READ FORWARD,ACK,CVC=1 COMMAND
          MOV #T30PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
          MOV #20.,T30SZ ;SET UP RECORD SIZE IN PACKET
          MOV R4,TSD8(R5) ;ISSUE COMMAND
          JSR PC,WAITF ;WAIT FOR SSR TO SET
          MOV TSSR(R5),R1 ;GET TSSR CONTENTS
          MOV #SSR,R2 ;SET UP EXPECTED
          CMP R1,R2 ;ARE THEY EQUAL
          BEQ 200$ ;BR, IF OK
          INC FATFLG ;ERROR COUNT
          ERRHRD ERRNO,T30RDF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                     TRAP C#ERHRD
                                     .WORD 230
                                     .WORD T30RDF
                                     .WORD PKTSSR

```

TEST 2: SKIP TAPE MARKS

```

5351 035040      104406      200:  CKLOOP                ;LOOP IF SELECTED
      035040      146050      MOV      @FREE,R1          ;FIRST LOC IN READ BUFFER
5352 035042      177777      MOV      @177777,R2       ;EXPECTED IF NO DATA TRANS.
5353 035046      020102      CMP      R1,R2            ;DID ANY DATA GET TRANSFERRED
5354 035052      001006      BNE     220:              ;BR, IF NO DATA TRANS (GOOD)
5355 035054      005237      002214      INC     FATFLG           ;ERROR COUNT
5356 035056      005237      002214      ERRHRD ERRNO,T30DTR,EXPREC ;DATA TRANSFERRED ON READ TAPE MARK
      035062      104456      TRAP    C#ERHRD
      035064      000347      .WORD  231
      035066      040770      .WORD  T30DTR
      035070      015554      .WORD  EXPREC

5361 035072      104406      220:  CKLOOP                ;LOOP IF SELECTED
      035072      036554      TRAP    C#CLP1
5362 035074      013702      MOV     T30FCN,R2        ;GET NUMBER OF SKIPS
5363 035100      005202      INC     R2               ;SET TO CORRECT FILE VALUE
5364 035102      000302      SWAB   R2               ;SWAP BYTE HALVES
5365 035104      052702      000001      BIS    @BIT0,R2         ;SET FOR RECORD #1
5366 035110      C17701      146002      MOV     @FREE,R1        ;GET INFO FROM BUFFER
5367 035114      020701      CMP     R2,R1            ;ARE THEY EQUAL
5368 035116      001406      BEQ    228:              ;BR, IF EQUAL (OK)
5369 035120      005237      002214      INC     FATFLG           ;ERROR COUNT
5373 035124      005237      002214      ERRHRD ERRNO,T30PTB,EXPREC ;RECORD POSITION WAS NOT CORRECT
      035124      104456      TRAP    C#ERHRD
      035126      000350      .WORD  232
      035130      037142      .WORD  T30PTB
      035132      015554      .WORD  EXPREC

5374 035134      104406      228:  CKLOOP                ;LOOP IF SELECTED
      035134      TRAP    C#CLP1

5375
5376      ;*****
5377      ;
5378      ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
5379      ;
5380      ;*****
5381
5382 035136      004737      011074      JSR    PC,REWIND        ;CALL TAPE REWIND COMMAND
5383 035142      103411      BCS    230:              ;BR, IF NO PROBLEM
5384 035144      010004      MOV     R0,R4           ;SAVE PACKET ADDRESS
5385 035146      016501      000002      MOV     TSSR(R5),R1     ;GET TSSR STATUS
5386 035152      005237      002214      INC     FATFLG           ;ERROR COUNT
5390 035156      005237      002214      ERRHRD ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
      035156      104456      TRAP    C#ERHRD
      035160      000351      .WORD  233
      035162      040140      .WORD  T30RWN
      035164      012126      .WORD  PKTSSR

5391 035166      104406      230:  CKLOOP                ;LOOP IF SELECTED
      035166      TRAP    C#CLP1

5392
5393      ;*****
5394      ;
5395      ;GET EXTENDED STATUS REGISTER ZERO (XST0) FROM MESSAGE BUFFER
5396      ;
5397      ;*****
5398
5399 035170      013701      036430      MOV     T30BFR+6,R1     ;PICK UP XST0
5400 035174      010102      MOV     R1,R2           ;SET UP EXPECTED

```

TEST 2: SKIP TAPE MARKS

```

5401 035176 052702 000002      BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
5402 035202 020102                CMP      R1,R2        ;DOES EXP = REC'D
5403 035204 001406                BEQ      240#         ;BR, IF EQUAL (OK)
5404 035206 005237 002214      INC      FATFLG      ;ERROR COUNT
5408 035212                                ERRHRD   ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      035212 104456                                TRAP    C#ERHRD
      035214 000352                                .WORD  234
      035216 037741                                .WORD  T30BOT
      035220 015554                                .WORD  EXPREC
5409 035222                                240#:   CKLOOP      ;LOOP IF SELECTED
      035222 104406                                TRAP    C#CLP1
5410 035224 005723                TST      (R3)+        ;POINT TO NEXT POSITION
5411 035226 011301                MOV      (R3),R1     ;GET NEXT COMMAND ETC.
5412 035230 020127 177777      CMP      R1,#177777  ;END OF TABLE MARKER
5413 035234 001410                BEQ      330#         ;BR, IF AT END OF TABLE
5414 035236 013701 036554      MOV      T30FCN,R1  ;GET NUMBER OF SKIPS
5415 035242 000241                CLC                          ;CLEAR THE CARRY BIT
5416 035244 006101                ROL      R1           ;PUSH OVER ONE POSITION
5417 035246 C10137 036554      MOV      R1,T30FCN  ;PUT BACK IN COUNTER
5418 035252 000137 034514      JMP      182#        ;JUMP TO MORE COMMANDS TO DO
5419 035256                                330#:   CKLOOP      ;LOOP IF SELECTED
      035256 104406                                TRAP    C#CLP1
5420 035260                                ENDSUB                    ;<<<<<<<<<<<< END SUBTEST >>>>>>>>>>>>>>>>>>
      035260 104403                                L10045:
5421 035262 023727 002214 000017  CMP      FATFLG,#15. ;IS ERROR COUNT AT 25
5422 035270 103402                                BLO      999#         ;BR, IF LESS THAN 25
5423 035272 004737 017262      JSR      PC,CKDROP   ;TRY TO DROP THE UNIT
5424 035276                                999#:
5425                                ;*
5426                                ;
5427                                ;TEST 2, SUBTEST 3
5428                                ;
5429                                ;
5430                                ;VERIFIES THAT A SKIP TAPE MARKS REVERSE COMMAND
5431                                ;ISSUED WHILE THE TAPE IS POSITIONED AT BOT CAUSES
5432                                ;FUNCTION REJECT TERMINATION, WITH THE NON-EXECUTABLE
5433                                ;FUNCTION (NEF) ERROR BIT SET.
5434                                ;
5435                                ;
5436                                ;
5437                                ;
5438                                ;
5439                                ;-
5440 035276                                BGNSUB                    ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>>>>>>>
      035276 104402                                T2.3:
5441 035300 004737 041152      JSR      PC,T30REST   ;SET COMMAND PACKET
5442 035304 005037 036554      CLR      T30FCN      ;CLEAR FILE COUNTER
5443 035310 004737 041244      JSR      PC,T30RT2    ;SET UP OTHER COMMAND PACKET
5444 035314 004737 041306      JSR      PC,T30RT3    ;SET UP OTHER COMMAND PACKET
5445 035320 012737 176750 036556  MOV      #65000.,T30DLY ;SET UP DELAY COUNTER
5446 035326 004737 016054 10#:   JSR      PC,SOFINIT   ;DO INITIALIZE ON CONTROLLER
5447 035332 103426                BCS      20#         ;BR IF INIT WAS OK
5448 035334                                DELAY   250           ;DELAY ROUTINE CALL
      035334 012727 000250                                MOV      #250.(PC)+
      035340 000000                                .WORD  0

```

## TEST 2: SKIP TAPE MARKS

```

035342 013727 002116          MOV      L#DLY,(PC)+
035346 000000                   .WORD    0
035350 005367 177772          DEC      -6(PC)
035354 001375                   BNE      -.4
035356 005367 177756          DEC      -22(PC)
035362 001367                   BNE      .-20
5449 035364 005337 036556      DEC      T30DLY          ;BUMP COUNTER
5450 035370 001356            BNE      10#             ;BR, IF MORE COUNTING TO DO
5451 035372 005237 002214      INC      FATFLG         ;ERROR COUNT
5453 035376 010001            MOV      R0,R1          ;CONTENTS OF TSSR REGISTER
5456 035400                   ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
035400 104455                   TRAP    C#ERRDF
035402 000353                   .WORD   235
035404 003646                   .WORD   SFIERR
035406 012114                   .WORD   SFIMSG
5457 035410
5458 035410 013737 002174 036420 20#:  MOV      UNITN,T30DSW      ;SET UP UNIT NUMBER
5459 035416 012704 036400      MOV      @T30PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
5460
5461 ;*****
5462 ;
5463 ;ISSUE WRITE CHARACTERISTICS COMMAND
5464 ;
5465 ;*****
5466
5467 035422 004737 010742      JSR      PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
5468 035426 103407            BCS      23#             ;BR, IF COMMAND ISSUED OK
5469 035430 005237 002214      INC      FATFLG         ;ERROR COUNT
5473 035434 010001            MOV      R0,R1          ;SAVE CONTENTS OF TSSR
5474 035436                   ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
035436 104456                   TRAP    C#ERRRD
035440 000354                   .WORD   236
035442 005052                   .WORD   WRTMSG
035444 012114                   .WORD   SFIMSG
5475 035446                   23#:  CKLOOP             ;LOOP IF SELECTED
035446 104406                   TRAP    C#CLP1
5476
5477 ;*****
5478 ;
5479 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
5480 ;
5481 ;*****
5482
5483 035450 004737 011074      JSR      PC,REWIND       ;CALL TAPE REWIND COMMAND
5484 035454 103411            BCS      30#             ;BR, IF NO PROBLEM
5485 035456 010004            MOV      R0,R4          ;GET PACKET ADDRESS
5486 035460 016501 000002      MOV      TSSR(R5),R1    ;GET STATUS REGISTER
5487 035464 005237 002214      INC      FATFLG         ;ERROR COUNT
5491 035470                   ERRHRD  ERRNO,T30RMN,PKTSSR ;REWIND NOT ACCEPTED
035470 104456                   TRAP    C#ERRRD
035472 000355                   .WORD   237
035474 040140                   .WORD   T30RMN
035476 012126                   .WORD   PKTSSR
5492 035500                   30#:  CKLOOP             ;LOOP IF SELECTED
035500 104406                   TRAP    C#CLP1
5493
5494 ;*****

```

TEST 2: SKIP TAPE MARKS

```

5495 ;
5496 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
5497 ;
5498 ;*****
5499 ;
5500 035502 013701 036430          MOV      T30BFR+6,R1      ;PICK UP XSTO
5501 035506 010102                MOV      R1,R2           ;SET UP EXPECTED
5502 035510 052702 000002        BIS      #BIT1,R2       ;SET BOT BIT IN EXPECTED
5503 035514 020102                CMP      R1,R2           ;DOES EXP = REC'D
5504 035516 001406                BEQ     40$             ;BR, IF EQUAL (OK)
5505 035520 005237 002214        INC      FATFLG         ;ERROR COUNT
5509 035524                ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                               TRAP      C$ERHRD
                               .WORD    238
                               .WORD    T30BOT
                               .WORD    EXPREC
                               TRAP      C$CLP1
035524 104456
035526 000356
035530 037741
035532 015554
5510 035534                40$:   CKLOOP           ;LOOP IF SELECTED
035534 104406                MOV      #1,T30WB       ;SET # OF TM TO SKIP
5511 035536 C12737 000001 036522        MOV      #1,T30WB
5512 ;
5513 ;*****
5514 ;
5515 ;SKIP TAPE MARK REVERSE,ACK,CVC=1 COMMAND
5516 ;
5517 ;*****
5518 ;
5519 035544 012737 141410 036520    MOV      #141410,T30PK3 ;SKIP TAPE MARK REVERSE,ACK,CVC=1 CMD
5520 035552 012704 036520          MOV      #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
5521 035556 010465 000000          MOV      R4,TSDR(R5)    ;ISSUE COMMAND
5522 035562 004737 016330          JSR     PC,WAITF        ;WAIT FOR SSR TO SET
5523 035566 016501 000002          MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
5524 035572 012702 100206          MOV      #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
5525 035576 020102                CMP      R1,R2           ;ARE THEY EQUAL
5526 035600 001406                BEQ     70$             ;BR, IF OK
5527 035602 005237 002214        INC      FATFLG         ;ERROR COUNT
5531 035606                ERRHRD  ERRNO,T30IBT,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                               TRAP      C$ERHRD
                               .WORD    239
                               .WORD    T30IBT
                               .WORD    PKTSSR
035606 104456
035610 000357
035612 036731
035614 012126
5532 035616                70$:   CKLOOP           ;LOOP IF SELECTED
035616 104406                TRAP      C$CLP1
5533 ;
5534 ;*****
5535 ;
5536 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
5537 ;
5538 ;*****
5539 ;
5540 035620 013701 036430          MOV      T30BFR+6,R1    ;PICK UP XSTO
5541 035624 010102                MOV      R1,R2           ;SET UP EXPECTED
5542 035626 052702 002000        BIS      #BIT10,R2      ;SET NEF BIT IN EXPECTED
5543 035632 020102                CMP      R1,R2           ;DOES EXP = REC'D
5544 035634 001406                BEQ     180$            ;BR, IF EQUAL (OK)
5545 035636 005237 002214        INC      FATFLG         ;ERROR COUNT
5549 035642                ERRHRD  ERRNO,T3ONEF,EXPREC ;TAPE NOT AT NEF
035642 104456                TRAP      C$ERHRD

```

TEST 2: SKIP TAPE MARKS

```

035644 000360                                  .WORD 240
035646 040476                                  .WORD T30NEF
035650 015554                                  .WORD EXPREC
5550 035652 180#: CKLOOP                      ;LOOP IF SELECTED
035652 104406                                  TRAP C#CLP1
5551 035654                                  ENDSUB                      ;<<<<<<<<<< END SUBTEST >>>>>>>>>>>>>>
035654                                  L10046:
035654 104403                                  TRAP C#ESUB
5552 035656 023727 002214 000017             CMP     FATFLG,#15.        ;IS ERROR COUNT AT 25
5553 035664 103402                    BLO    999#               ;BR, IF LESS THAN 25
5554 035666 004737 017262                    JSR    PC,CKDROP          ;TRY TO DROP THE UNIT
5555 035672                    999#:#
5556                                ;*
5557                                ;
5558                                ;TEST 2. SUBTEST 4
5559                                ;
5560                                ;
5561                                ;VERIFIES THAT A SKIP TAPE MARKS REVERSE COMMAND
5562                                ;ISSUED WHILE THE TAPE IS POSITIONED JUST BEFORE THE
5563                                ;FIRST RECORD ON ON TAPE (BUT NOT AT BOT) CAUSES TAPE
5564                                ;STATUS ALERT TERMINATION, WITH THE REVERSE INTO BOT
5565                                ;(RIB) STATUS BIT SET.
5566                                ;
5567                                ;
5568                                ;
5569                                ;
5570                                ;
5571                                ;
5572 035672                    :- BGNSUB                      ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
035672                                  T2.4:
035672 104402                                  TRAP C#BSUB
5573 035674 004737 041152                    JSR    PC,T30REST        ;SET COMMAND PACKET
5574 035700 005037 036554                    CLR    T30FCN           ;CLEAR FILE COUNTER
5575 035704 004737 041244                    JSR    PC,T30RT2        ;SET UP OTHER COMMAND PACKET
5576 035710 004737 041306                    JSR    PC,T30RT3        ;SET UP OTHER COMMAND PACKET
5577 035714 012737 176750 036556             MOV    #65000.,T30DLY   ;SET UP DELAY COUNTER
5578 035722 004737 016054                    10#: JSR    PC,SOFINIT   ;DO INITIALIZE ON CONTROLLER
5579 035726 103426                    BCS    20#              ;BR IF INIT WAS OK
5580 035730                    DELAY 250               ;DELAY ROUTINE CALL
035730 012727 000250                                  MOV    #250.(PC)+
035734 000000                                  .WORD 0
035736 013727 002116                                  MOV    L#DLY.(PC)+
035742 000000                                  .WORD 0
035744 005367 177772                                  DEC    -6(PC)
035750 001375                                  BNE    --4
035752 005367 177756                                  DEC    -22(PC)
035756 001367                                  BNE    --20
5581 035760 005337 036556                    DEC    T30DLY           ;BUMP COUNTER
5582 035764 001356                    BNE    10#             ;BR, IF MORE COUNTING TO DO
5583 035766 005237 002214                    INC    FATFLG          ;ERROR COUNT
5587 035772 010001                    MOV    R0,R1           ;CONTENTS OF TSSR REGISTER
5588 035774                    ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
035774 104455                                  TRAP C#ERDF
035776 000361                                  .WORD 241
036000 003646                                  .WORD SFIERR
036002 012114                                  .WORD SFIMSG
5589 036004                    20#:

```

TEST 2: SKIP TAPE MARKS

```

5590 036004 013737 002174 036420      MOV      UNITN,T30DSW      ;SET UP UNIT NUMBER
5591 036012 012704 036400      MOV      @T30PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
5592
5593      ;*****
5594      ;
5595      ;ISSUE WRITE CHARACTERISTICS COMMAND
5596      ;
5597      ;*****
5598
5599 036016 004737 010742      JSR      PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
5600 036022 103407                BCS      23$             ;BR, IF COMMAND ISSUED OK
5601 036024 005237 002214      INC      FATFLG          ;ERROR COUNT
5605 036030 010001                MOV      R0,R1           ;SAVE CONTENTS OF TSSR
5606 036032                ERRHRD  ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICSC FAILED
                                TRAP      C$ERHRD
                                .WORD    242
                                .WORD    WRTMSG
                                .WORD    SFMSG
                                TRAP      C$CLP1
5607 036042 104406      23$:   CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
5608
5609      ;*****
5610      ;
5611      ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
5612      ;
5613      ;*****
5614
5615 036044 004737 011074      JSR      PC,REWIND       ;CALL TAPE REWIND COMMAND
5616 036050 103411                BCS      30$             ;BR, IF NO PROBLEM
5617 036052 010004                MOV      R0,R4           ;GET PACKET ADDRESS
5618 036054 016501 000002      MOV      TSSR(R5),R1     ;GET STATUS REGISTER
5619 036060 005237 002214      INC      FATFLG          ;ERROR COUNT
5623 036064                ERRHRD  ERRNO,T30R!N,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    243
                                .WORD    T30RMN
                                .WORD    PKTSSR
                                TRAP      C$CLP1
5624 036074 104406      30$:   CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
5625
5626      ;*****
5627      ;
5628      ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
5629      ;
5630      ;*****
5631
5632 036076 013701 036430      MOV      T30BFR+6,R1     ;PICK UP XSTO
5633 036102 010102                MOV      R1,R2           ;SET UP EXPECTED
5634 036104 052702 000002      BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
5635 036110 020102                CMP      R1,R2           ;DOES EXP = REC'D
5636 036112 001406                BEQ      40$             ;BR, IF EQUAL (OK)
5637 036114 005237 002214      INC      FATFLG          ;ERROR COUNT
5641 036120                ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    244
                                .WORD    T30BOT
                                .WORD    EXPREC
036120 104456
036122 000364
036124 037741
036126 015554

```

TEST 2: SKIP TAPE MARKS

```

5642 036130      40$:  CKLOOP      ;LOOP IF SELECTED
      036130 104406      ;                               TRAP    C$CLP1
5643 036132 013737 003116 036522      MOV    FREE,T30WB      ;SET UP GOOD WRITE BUFFER
5644 036140 012737 000400 036526      MOV    #256.,T30SZ    ;SET UP SIZE
5645
5646      ;*****
5647      ;
5648      ;WRITE DATA,ACK,CVC=1 COMMAND
5649      ;
5650      ;*****
5651
5652 036146 012737 140005 036520      MOV    #140005,T30PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
5653 036154 012704 036520      MOV    #T30PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
5654 036160 010465 000000      MOV    R4,TSDB(R5)   ;ISSUE COMMAND
5655 036164 004737 016330      JSR    PC,WAITF      ;WAIT FOR SSR TO SET
5656 036170 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
5657 036174 012702 000200      MOV    #SSR,R2      ;SET UP EXPECTED
5658 036200 020102      CMP    R1,R2        ;ARE THEY EQUAL
5659 036202 C01406      BEQ    70$          ;BR, IF OK
5660 036204 005237 002214      INC    FATFLG       ;ERROR COUNT
5664 036210      ERRHRD  ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      036210 104456      TRAP    C$ERHRD
      036212 000365      .WORD  245
      036214 037070      .WORD  T30WDD
      036216 012126      .WORD  PKTSSR
5665 036220      70$:  CKLOOP      ;LOOP IF SELECTED
      036220 104406      ;                               TRAP    C$CLP1
5666
5667      ;*****
5668      ;
5669      ;SKIP TAPE MARK REVERSE,ACK,CVC=1 COMMAND
5670      ;
5671      ;*****
5672
5673 036222 012737 000001 036522      MOV    #1,T30WB      ;# OF TM TO SKIP
5674 036230 012737 141410 036520      MOV    #141410,T30PK3 ;SKIP TAPE MARK REVERSE,ACK,CVC=1 CMD
5675 036236 012704 036520      MOV    #T30PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
5676 036242 010465 000000      MOV    R4,TSDB(R5)   ;ISSUE COMMAND
5677 036246 004737 016330      JSR    PC,WAITF      ;WAIT FOR SSR TO SET
5678 036252 016501 000002      MOV    TSSR(R5),R1   ;PICK UP TSSR
5679 036256 012702 100204      MOV    #SSR!BIT2!SC,R2 ;SET UP EXPECTED (SSR AND SC ONLY)
5680 036262 020102      CMP    R1,R2        ;WAS STATUS GOOD
5681 036264 001406      BEQ    160$         ;BR, IF TERMINATION WAS GOOD
5682 036266 005237 002214      INC    FATFLG       ;ERROR COUNT
5686 036272      ERRHRD  ERRNO,T30IBU,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
      036272 104456      TRAP    C$ERHRD
      036274 000366      .WORD  246
      036276 036560      .WORD  T30IBU
      036300 012126      .WORD  PKTSSR
5687 036302      160$: CKLOOP      ;LOOP IF SELECTED
      036302 104406      ;                               TRAP    C$CLP1
5688
5689      ;*****
5690      ;
5691      ;GET EXTENDED STATUS REGISTER ZERO (XST3) FROM MESSAGE BUFFER
5692      ;
5693      ;*****

```





TEST 2: SKIP TAPE MARKS

5752	036520	100205			T30RB:	.WORD	100205		;REREAD COMMAND, IE AND ACK
5753	036522				T30WB:	.WORD	FREE		;ADDRESS OF WRITE BUFFER
5754	036522	003116				.WORD	0		
5755	036524	000000			T30SZ:	.WORD	0		;SIZE OF BUFFER (EXTENT)
5756	036526	000000				.EVEN			
5757									
5758									
5759									
5760									
5761	036530				T30BF2:				
5762	036530	010			T30BS0:	.BYTE	10		;BSELO AREA
5763	036531	200			T30BS1:	.BYTE	200		;BSEL1 AREA
5764	036532	000000			T30S2:	.WORD	0		;SEL 2 AREA
5765	036534	000000			T30S3:	.WORD	0		;DATA AREA
5766									
5767									
5768						.EVEN			
5769									;TAPE MOTION PACKET COMMAND VALUES
5770									
5771	036536				T30IMV:				
5772	036536				T30RN:				
5773	036536	000000				.WORD	000000		;NEITHER EWB NOR ESS
5774	036540	000100				.WORD	000100		;EWB SET
5775	036542	000200				.WORD	000200		;ESS SET
5776	036544	000300				.WORD	000300		;BOTH EWB AND ESS SET
5777	036546	177777				.WORD	177777		;END OF DATA
5778									
5779									
5780	036550	000000			T30CNT:	.WORD	0		;TAPE TIMER COUNTER STORAGE AREA
5781	036552	000000			T30CNU:	.WORD	0		;TAPE TIMER COUNTER STORAGE AREA
5782	036554	000000			T30FCN:	.WORD	0		;FILE NUMBER COUNTER
5783	036556	000000			T30DLY:	.WORD	0		;DELAY COUNTER STORAGE
5784									
5785									
5786									
5787									
5788	036560	124	123	123	T30IBU:	.ASCIZ			'TSSR Incorrect After SKIP TAPE MARK REVERSE Into BOT'
5789	036645	122	111	102	T30RIB:	.ASCIZ			'RIB Bit (XST3) Failed To Set After Reverse Into BOT'
5790	036731	124	123	123	T30IBT:	.ASCIZ			'TSSR Incorrect After SKIP TAPE MARK REVERSE At BOT'
5791	037014	124	123	123	T30SKM:	.ASCIZ			'TSSR Incorrect After SKIP TAPE MARK Command'
5792	037070	124	123	123	T30WDD:	.ASCIZ			'TSSR Not Correct After WRITE DATA Command'
5793	037142	124	141	160	T30PTB:	.ASCIZ			'Tape Not Positioned On Correct Record After READ REVERSE'
5794	037233	124	141	160	T30TPB:	.ASCIZ			'Tape Not Positioned On Second File First Record'
5795	037313	124	123	123	T30RDF:	.ASCIZ			'TSSR Incorrect After READ FORWARD Into "File"'
5796	037371	124	123	123	T30RDG:	.ASCIZ			'TSSR Incorrect After SPACE Command Into TAPE MARK'
5797	037453	124	123	123	T30WDF:	.ASCIZ			'TSSR Not Correct After Illegal Mode Bits Set'
5798	037530	111	154	154	T30LOQ:	.ASCIZ			'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
5799	037611	127	122	111	T30SSR:	.ASCIZ			'WRITE MISCELLANEOUS Command Not Accepted'
5800	037662	124	123	123	T30WDE:	.ASCIZ			'TSSR Not Correct After SKIP TAPE MARKS, At BOT'
5801	037741	124	141	160	T30BOT:	.ASCIZ			'Tape Not At BOT After REWIND Command'
5802	040006	124	123	123	T30TM:	.ASCIZ			'TSSR Not Correct After SPACE FORWARD Command'
5803	040063	124	123	123	T30TM2:	.ASCIZ			'TSSR Not Correct After SPACE REVERSE Command'
5804	040140	122	145	167	T30RWN:	.ASCIZ			'Rewind (POSITION) Command Not Accepted'
5805	040207	104	162	151	T30OFL:	.ASCIZ			'Drive 7 Select Failed To Set "OFL" In TSSR'
5806	040262	124	123	123	T30WDC:	.ASCIZ			'TSSR Not Correct After WRITE TAPE MARK Command'
5807	040341	103	126	103	T30VCK:	.ASCIZ			'CVC Set, Didn't Reset VCK In Message Buffer'
5808	040414	124	115	113	T30TMK:	.ASCIZ			'TMK Not Set After WRITE TAPE MARK (RETRY) Command'

TEST 2: SKIP TAPE MARKS

```

5809 040476      123      113      111 T3ONEF: .ASCIZ 'SKIP TAPE MARKS, At BOT, Failed To Set NEF Bit'
5810 040555      124      115      113 T3ORRM: .ASCIZ 'TMK Not Set After READ REVERSE Into TAPE MARK'
5811 040633      124      115      113 T3ORRN: .ASCIZ 'TMK Not Set After SPACE REVERSE Into TAPE MARK'
5812 040712      124      115      113 T3ORRP: .ASCIZ 'TMK Not Set After READ FORWARD Into TAPE MARK'
5813 040770      116      117      040 T3ODTR: .ASCIZ 'NO Data Transferred On READ FORWARD'
5814 041034      104      141      164 T3ODTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
5815 041131      123      153      151 TST30ID: .ASCIZ 'Skip Tape Marks'

```

```

5816 .EVEN
5817 ;*
5818 ;
5819 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5820 ;WRITE SUBSYSTEM MEMORY COMMAND
5821 ;
5822 ;-
5823

```

```

5824 041152
5825 041152
5826 041156      012701    036400
5827 041162      012721    100004
5828 041166      012721    036410
5829 041172      005021
5830 041174      012721    000012
5831 041200      012721    036422
5832 041204      005021
5833 041206      012721    000024
5834 041212      005021
5835 041214      012711    000000
5836 041220      012702    000030
5837 041224      012762    177777    036422    64$:
5838 041232      005742
5839 041234      022702    000000
5840 041240      001371
5841 041242      000207
5842

```

```

T3OREST:
SAVREG
MOV #T3OPACKET,R1 ;SAVE THE REGISTERS
MOV #100004,(R1)+ ;START OF THE PACKET
MOV #T30DATA,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK.
CLR (R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
MOV #10,(R1)+ ;EXTENDED ADDRESS
MOV #T30BFR,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
CLR (R1)+ ;ADDRESS OF MESSAGE BUFFER
MOV #20,(R1)+ ;LENGTH OF MESSAGE BUFFER
CLR (R1)+
MOV #0,(R1) ;SELECT DRIVE ZERO
MOV #24,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
MOV #177777,T30BFR(R2) ;ALL ONES TO MESSAGE BUFFER
TST -(R2) ;NEXT LOCATION
CMP #0,R2 ;CHECK R2 FOR DONE
BNE 64$ ;KEEP GOING UNTIL DONE
RTS PC ;RETURN

```

```

5843 041244
5844 041244
5845 041250      012701    036510
5846 041254      012721    100006
5847 041260      012721    036530
5848 041264      005021
5849 041266      012721    000006
5850 041272      005021
5851 041274      012701    036530
5852 041300      005021
5853 041302      005011
5854 041304      000207
5855 041306

```

```

T3ORT2:
SAVREG
MOV #T3OPK2,R1 ;SAVE THE REGISTERS
MOV #100006,(R1)+ ;START OF THE PACKET
MOV #T30BF2,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK.
CLR (R1)+ ;ADDRESS OF DATA BLOCK
MOV #6,(R1)+ ;EXTENDED ADDRESS
CLR (R1)+ ;SIZE OF DATA BLOCK IN BYTES
MOV #T30BF2,R1 ;POINT TO DATA SEL AREA
CLR (R1)+
CLR (R1)
RTS PC ;RETURN

```

```

5856 041306
5857 041312      012701    036520
5858 041316      005021
5859 041320      005021
5860 041322      005021
5861 041324      005011
5862 041326      000207
5863 041330

```

```

T3ORT3:
SAVREG
MOV #T3OPK3,R1 ;SAVE REGISTERS
CLR (R1)+ ;SET UP POINTER ADDRESS
CLR (R1)+ ;COMMAND SPACE
CLR (R1)+ ;ADDRESS OF DATA BLOCK
CLR (R1)+ ;EXTENDED ADDRESS
RTS PC ;SIZE OF DATA TRANSFER BLOCK
ENDTST ;RETURN

```

L10043: TRAP C#ETST





TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

5964	041650	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND
5965	041654	004737	016330			JSR	PC,WAITF		;WAIT FOR SSR TO SET
5966	041660	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS
5967	041664	012702	000200			MOV	#SSR,R2		;SET UP EXPECTED
5968	041670	020102				CMP	R1,R2		;ARE THEY EQUAL
5969	041672	001406				BEQ	80#		;BR, IF OK
5970	041674	005237	002214			INC	FATFLG		;ERROR COUNT
5974	041700					ERRHRD	ERRNO,T31WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA
	041700	104456							TRAP C#ERHRD
	041702	000461							.WORD 305
	041704	045130							.WORD T31WDC
	041706	012126							.WORD PKTSSR
5975	041710			80#:	CKLOOP				;LOOP IF SELECTED
	041710	104406							TRAP C#CLP1
5976	041712	004737	011074			JSR	PC,REWIND		;CALL TAPE REWIND COMMAND
5977	041716	103407				BCS	230#		;BR, IF NO PROBLEM
5978	041720	010001				MOV	R0,R1		;SAVE TSSR
5979	041722	005237	002214			INC	FATFLG		;ERROR COUNT
5983	041726					ERRHRD	ERRNO,T31RWN,EXPREC		;REWIND NOT ACCEPTED
	041726	104456							TRAP C#ERHRD
	041730	000462							.WORD 306
	041732	044574							.WORD T31RWN
	041734	015554							.WORD EXPREC
5984	041736			230#:	CKLOOP				;LOOP IF SELECTED
	041736	104406							TRAP C#CLP1
5985	041740	013701	043120			MOV	T31BFR+6,R1		;PICK UP XSTO
5986	041744	010102				MOV	R1,R2		;SET UP EXPECTED
5987	041746	052702	000002			BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED
5988	041752	020102				CMP	R1,R2		;DOES EXP = REC'D
5989	041754	001406				BEQ	240#		;BR, IF EQUAL (OK)
5990	041756	005237	002214			INC	FATFLG		;ERROR COUNT
5994	041762					ERRHRD	ERRNO,T31BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND
	041762	104456							TRAP C#ERHRD
	041764	000463							.WORD 307
	041766	044245							.WORD T31BOT
	041770	015554							.WORD EXPREC
5995	041772			240#:	CKLOOP				;LOOP IF SELECTED
	041772	104406							TRAP C#CLP1
5996	041774	012737	041012	043210	265#:	MOV	#041012,T31PK3		;NO-OP,CVC=1 COMMAND
5997	042002	012704	043210			MOV	#T31PK3,R4		;SET UP R4 WITH PACKET ADDRESS
5998	042006	010337	043216			MOV	R3,T31SZ		;SET UP RECORD SIZE IN PACKET
5999	042012	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND
6000	042016	004737	016330			JSR	PC,WAITF		;WAIT FOR SSR TO SET
6001	042022	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS
6002	042026	012702	000200			MOV	#SSR,R2		;SET UP EXPECTED
6003	042032	020102				CMP	R1,R2		;ARE THEY EQUAL
6004	042034	001406				BEQ	280#		;BR, IF OK
6005	042036	005237	002214			INC	FATFLG		;ERROR COUNT
6009	042042					ERRHRD	ERRNO,T31RDF,PKTSSR		;TSSR INCORRECT AFTER READ DATA
	042042	104456							TRAP C#ERHRD
	042044	000464							.WORD 308
	042046	043443							.WORD T31RDF
	042050	012126							.WORD PKTSSR
6010	042052			280#:	CKLOOP				;LOOP IF SELECTED
	042052	104406							TRAP C#CLP1
6011	042054	013701	043120			MOV	T31BFR+6,R1		;PICK UP XSTO
6012	042060	010102				MOV	R1,R2		;SET UP EXPECTED

TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

```

6013 042062 052702 000002      BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
6014 042066 020102      CMP      R1,R2        ;DOES EXP = REC'D
6015 042070 001406      BEQ      285#         ;BR, IF EQUAL (OK)
6016 042072 005237 002214      INC      FATFLG       ;ERROR COUNT
6020 042076      ERRHRD  ERRNO,T31BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C#ERHRD
                                .WORD    309
                                .WORD    T31BOT
                                .WORD    EXPREC
        042076 104456
        042100 000465
        042102 044245
        042104 015554
6021 042106      285# : CKLOOP      ;LOOP IF SELECTED
        042106 104406      TRAP      C#CLP1
6022 042110 012737 140001 043210      MOV      #140001,T31PK3 ;READ,ACK,CVC-1 COMMAND
6023 042116 012704 043210      MOV      #T31PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
6024 042122 012737 000144 043216      MOV      #100.,T31SZ   ;SET UP RECORD SIZE IN PACKET
6025 042130 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
6026 042134 004737 016330      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
6027 042140 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
6028 042144 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
6029 042150 C20102      CMP      R1,R2        ;ARE THEY EQUAL
6030 042152 001406      BEQ      290#         ;BR, IF OK
6031 042154 005237 002214      INC      FATFLG       ;ERROR COUNT
6035 042160      ERRHRD  ERRNO,T31RDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C#ERHRD
                                .WORD    310
                                .WORD    T31RDE
                                .WORD    PKTSSR
        042160 104456
        042162 000466
        042164 043244
        042166 012126
6036 042170      290# : CKLOOP      ;LOOP IF SELECTED
        042170 104406      TRAP      C#CLP1
6037 042172 017701 140720      MOV      #FREE,R1     ;GET DATA READ
6038 042176 012702 000144      MOV      #100.,R2     ;READ EXPECTED
6039 042202 020102      CMP      R1,R2        ;DID TAPE STAY POSITIONED
6040 042204 001406      BEQ      330#         ;BR, IF EXPD = RECD
6041 042206 005237 002214      INC      FATFLG       ;ERROR COUNT
6045 042212      ERRHRD  ERRNO,T31WNG,EXPREC ;TAPE DATA NOT CORRECT
                                TRAP      C#ERHRD
                                .WORD    311
                                .WORD    T31WNG
                                .WORD    EXPREC
        042212 104456
        042214 000467
        042216 043371
        042220 015554
6046 042222      330# : ENDSUB
6047 042222      L10051: TRAP      C#ESUB
        042222 104403
6048 042224 023727 002214 000017      CMP      FATFLG,#15.  ;IS ERROR COUNT AT 25
6049 042232 103402      BLO      999#         ;BR, IF LESS THAN 25
6050 042234 004737 017262      JSR      PC,CKDROP    ;TRY TO DROP THE UNIT
6051 042240      999# :
6052      ;+
6053      ;
6054      ;TEST 3. SUBTEST 2
6055      ;
6056      ;
6057      ;
6058      ;
6059      ;
6060      ;
6061      ;
6062      ;
        VERIFIES THAT THE INITIALIZE COMMAND OPERATES AS A NO-OP,
        ASSUMING NO MICRODIAGNOSTIC ERRORS ARE PRESENT (THEY WOULD
        ALREADY HAVE BEEN DETECTED IN PREVIOUS TESTS). THE TEST
        SEQUENCE IS SIMILAR TO THAT USED IN SUBTEST 1.
    
```

TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

```

6063
6064
6065 042240          ;
        042240          ;
        042240 104402          ;
6066 042242 004737 046510      JSR    PC,T31REST          ;SET COMMAND PACKET
6067 042246 004737 046602      JSR    PC,T31RT2          ;SET UP OTHER COMMAND PACKET
6068 042252 004737 046644      JSR    PC,T31RT3          ;SET UP OTHER COMMAND PACKET
6069 042256 004737 016054      JSR    PC,SOFINIT          ;DO INITIALIZE ON CONTROLLER
6070 042262 103407              BCS    20#                ;BR IF INIT WAS OK
6071 042264 005237 002214      INC    FATFLG              ;ERROR COUNT
6073 042270 010001              MOV    R0,R1              ;CONTENTS OF TSSR REGISTER
6076 042272          ERRDF    ERRNO,SFIERR,SFIMSG          ;FATAL ERROR TSSR WAS NOT OK
        042272 104455          TRAP    C#ERDF
        042274 000470          .WORD  312
        042276 003646          .WORD  SFIERR
        042300 012114          .WORD  SFIMSG
6077 042302 013737 002174 043110 20# :  MOV    UNITN,T31DSW          ;SET UP UNIT NUMBER IN PACKET
6078 042310 C12704 043070          MOV    #T31PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
6079 042314 004737 010742      JSR    PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
6080 042320 103407              BCS    23#                ;BR, IF COMMAND ISSUED OK
6081 042322 005237 002214      INC    FATFLG              ;ERROR COUNT
6083 042326 010001              MOV    R0,R1              ;SAVE CONTENTS OF TSSR
6086 042330          ERRHRD    ERRNO,WRTMSG,SFIMSG          ;WRITE CHARACTERISTICS FAILED
        042330 104456          TRAP    C#ERHRD
        042332 000471          .WORD  313
        042334 005052          .WORD  WRTMSG
        042336 012114          .WORD  SFIMSG
6087 042340          23# :  CKLOOP          ;LOOP IF SELECTED
        042340 104406          TRAP    C#CLP1
6088 042342 004737 011074      JSR    PC,REWIND          ;CALL TAPE REWIND COMMAND
6089 042346 103407              BCS    30#                ;BR, IF NO PROBLEM
6090 042350 010004              MOV    R0,R4              ;SET UP REWIND PACKET ADDRESS
6091 042352 005237 002214      INC    FATFLG              ;ERROR COUNT
6093 042356          ERRHRD    ERRNO,T31RWN,PKTSSR          ;REWIND NOT ACCEPTED
        042356 104456          TRAP    C#ERHRD
        042360 000472          .WORD  314
        042362 044574          .WORD  T31RWN
        042364 012126          .WORD  PKTSSR
6096 042366          30# :  CKLOOP          ;LOOP IF SELECTED
        042366 104406          TRAP    C#CLP1
6097 042370 013701 043120      MOV    T31BFR+6,R1        ;PICK UP XSTO
6098 042374 010102              MOV    R1,R2              ;SET UP EXPECTED
6099 042376 052702 000002      BIS    #BIT1,R2          ;SET BOT BIT IN EXPECTED
6100 042402 020102              CMP    R1,R2              ;DOES EXP = REC'D
6101 042404 001406              BEQ    40#                ;BR, IF EQUAL (OK)
6102 042406 005237 002214      INC    FATFLG              ;ERROR COUNT
6106 042412          ERRHRD    ERRNO,T31BOT,EXPREC          ;TAPE NOT AT BOT AFTER REWIND
        042412 104456          TRAP    C#ERHRD
        042414 000473          .WORD  315
        042416 044245          .WORD  T31BOT
        042420 015554          .WORD  EXPREC
6107 042422          40# :  CKLOOP          ;LOOP IF SELECTED
        042422 104406          TRAP    C#CLP1
6108 042424 013737 003116 043212  MOV    FREE,T31WB          ;STARTING WRITE BUFFER ADDRESS
6109 042432 012737 140005 043210 65# :  MOV    #140005,T31PK3      ;WRITE DATA,CVC=1,ACK COMMAND
6110 042440 012704 043210      MOV    #T31PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
    
```



TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

6111	042444	012700	000144		MOV	#100.,R0		;SET PATTERN IN CORRECT REGISTER
6112	042450	004737	017502		JSR	PC,FILLMEM		;FILL MEMORY WITH RECORD SIZE
6113	042454	012737	000144	043216	MOV	#100.,T31SZ		;SET UP RECORD SIZE IN PACKET
6114	042462	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND
6115	042466	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET
6116	042472	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
6117	042476	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED
6118	042502	020102			CMP	R1,R2		;ARE THEY EQUAL
6119	042504	001406			BEQ	804		;BR, IF OK
6120	042506	005237	002214		INC	FATFLG		;ERROR COUNT
6124	042512				ERRHRD	ERRNO,T31WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA
	042512	104456					TRAP	C:ERHRD
	042514	000474					.WORD	316
	042516	045130					.WORD	T31WDC
	042520	012126					.WORD	PKTSSR
6125	042522			804:	CKLOOP			;LOOP IF SELECTED
	042522	104406					TRAP	C:CLP1
6126	042524	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND
6127	042530	103407			BCS	2304		;BR, IF NO PROBLEM
6128	042532	010001			MOV	R0,R1		;SAVE TSSR
6129	042534	005237	002214		INC	FATFLG		;ERROR COUNT
6133	042540				ERRHRD	ERRNO,T31RWN,EXPREC		;REWIND NOT ACCEPTED
	042540	104456					TRAP	C:ERHRD
	042542	000475					.WORD	317
	042544	044574					.WORD	T31RWN
	042546	015554					.WORD	EXPREC
6134	042550			2304:	CKLOOP			;LOOP IF SELECTED
	042550	104406					TRAP	C:CLP1
6135	042552	013701	043120		MOV	T31BFR+6,R1		;PICK UP XSTO
6136	042556	010102			MOV	R1,R2		;SET UP EXPECTED
6137	042560	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED
6138	042564	020102			CMP	R1,R2		;DOES EXP = REC'D
6139	042566	001406			BEQ	2404		;BR, IF EQUAL (OK)
6140	042570	005237	002214		INC	FATFLG		;ERROR COUNT
6144	042574				ERRHRD	ERRNO,T31BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND
	042574	104456					TRAP	C:ERHRD
	042576	000476					.WORD	318
	042600	044245					.WORD	T31BOT
	042602	015554					.WORD	EXPREC
6145	042604			2404:	CKLOOP			;LOOP IF SELECTED
	042604	104406					TRAP	C:CLP1
6146	042606	012737	041012	043210	2654:	MOV	#041012,T31PK3	;INITIALIZE,CVC=1 COMMAND
6147	042614	012704	043210		MOV	#T31PK3,R4		;SET UP R4 WITH PACKET ADDRESS
6148	042620	010337	043216		MOV	R3,T31SZ		;SET UP RECORD SIZE IN PACKET
6149	042624	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND
6150	042630	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET
6151	042634	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
6152	042640	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED
6153	042644	020102			CMP	R1,R2		;ARE THEY EQUAL
6154	042646	001406			BEQ	2804		;BR, IF OK
6155	042650	005237	002214		INC	FATFLG		;ERROR COUNT
6159	042654				ERRHRD	ERRNO,T31RDF,PKTSSR		;TSSR INCORRECT AFTER READ DATA
	042654	104456					TRAP	C:ERHRD
	042656	000477					.WORD	319
	042660	043443					.WORD	T31RDF
	042662	012126					.WORD	PKTSSR
6160	042664			2804:	CKLOOP			;LOOP IF SELECTED



TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

```

043066 003600 .WORD L10050-
6209
6210 ;*
6211 ;LOCAL STORAGE FOR THIS TEST
6215 043070 ;
6216 043070 100004 T31PACKET: ;COMMAND PACKET FOR TEST
6217 043072 043100 .WORD 100004 ;WRITE CHARACTERISTICS COMMAND, WITH . ACK
6218 043074 000000 .WORD T31DATA ;ADDRESS OF CHARACTERISTICS BLOCK
6219 043076 000012 .WORD 0 ;STARTING VALUE OF BLOCK SIZE
6220 043100 T31DATA: ;CHARACTERISTICS DATA BLOCK
6221 043100 043112 .WORD T31BFR ;ADDRESS OF MESSAGE BUFFER
6222 043102 000000 .WORD 0
6223 043104 000024 .WORD 20. ;LENGTH OF MESSAGE BUFFER
6224 043106 000000 .WORD 0
6225 043110 000000 T31DSW: .WORD 0 ;SELECT DRIVE 0
6226 043112 T31BFR: .BLKW 25. ;MESSAGE BUFFER
6227
6228 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
6229 ;
6231 043200 043200 T31PK2: .-<..10>&177770
6233 043200 T31PK2: .WORD 100006 ;WRITE SUB SYS MEM COMMAND, AND ACK
6234 043200 100006 .WORD T31BF2 ;ADDRESS OF SELECT BLOCK DATA
6235 043202 043220 .WORD 0 ;SIZE OF DATA PACKET
6236 043204 000000 .WORD 6.
6237 043206 000006
6238
6242 043210 T31PK3:
6243 043210 100005 .WORD 100005 ;REREAD COMMAND, AND ACK
6244 043212 T31RB:
6245 043212 003116 T31WB: .WORD FREE ;ADDRESS OF WRITE BUFFER
6246 043214 000000 .WORD 0
6247 043216 000000 T31SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
6248 .EVEN
6249 ;
6250 ;
6251 ;
6252 043220 T31BF2:
6253 043220 010 T31BS0: .BYTE 10 ;BSELO AREA
6254 043221 200 T31BS1: .BYTE 200 ;BSEL1 AREA
6255 043222 000000 T31S2: .WORD 0 ;SEL 2 AREA
6256 043224 000000 T31S3: .WORD 0 ;DATA AREA
6257 ;
6258 ;
6259 .EVEN
6260 ;TAPE MOTION PACKET COMMAND VALUES
6261
6262 043226 100205 T31RN: .WORD 100205 ;REREAD DATA (NEXT)
6263 043230 100605 T31WDR: .WORD 100605 ;REREAD DATA RETRY
6264 043232 102205 T31CON: .WORD 102205 ;WRITE CONTINOUS
6265 043234 177777 .WORD 177777 ;END OF DATA
6266
6267 ;
6268 043236 000000 T31CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
6269 043240 000000 T31CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
6270 043242 000000 T31DLY: .WORD 0 ;DELAY COUNTER
6271 ;*
6272 ;LOCAL TEXT MESSAGES FOR TEST

```

TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

```

6273
6274
6275 043244      124      123      123  T31RDE: .ASCIZ  'TSSR Not Correct After READ Command'
6276 043310      124      141      160  T31WNH: .ASCIZ  'Tape Position Incorrect After INITIALIZE Command'
6277 043371      124      141      160  T31WNG: .ASCIZ  'Tape Position Incorrect After NOP Command'
6278 043443      124      123      123  T31RDF: .ASCIZ  'TSSR Incorrect After READ DATA Command'
6279 043512      122      105      122  T31RRF: .ASCIZ  'REREAD Previous (Space Reverse, Read Forward) Command Failed'
6280 043607      120      117      123  T31SC: .ASCIZ   'POSITION (Space Command) Failed, TSSR Not Correct'
6281 043671      122      111      102  T31LOR: .ASCIZ  'RIB NOT SET AFTER READ REVERSE INTO BOT'
6282 043741      124      123      123  T31WDF: .ASCIZ  'TSSR Not Correct After Illegal Mode Bits Set'
6283 044016      111      154      154  T31LOQ: .ASCIZ  'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
6284 044077      122      105      122  T31SSR: .ASCIZ  'REREAD COMMAND Not Accepted'
6285 044133      124      123      123  T31WDE: .ASCIZ  'TSSR Not Correct After NO-OP ("CLEAN TAPE") AND INITIALIZE Command, At BOT'
6286 044245      124      141      160  T31BOT: .ASCIZ  'Tape Not At BOT After REWIND Command (BOT Not Set In XSTO)'
6287 044340      116      117      055  T31TIM: .ASCIZ  'NO-OP ("CLEAN TAPE") AND INITIALIZE'S Erase Tape Not Long Enough'
6288 044440      122      105      122  T31EOT: .ASCIZ  'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
6289 044517      124      123      123  T31TM: .ASCIZ   'TSSR Not Correct After REREAD COMMAND Reject'
6290 044574      122      145      167  T31RWV: .ASCIZ  'Rewind (POSITION) Command Not Accepted'
6291 044643      122      101      115  T31RNC: .ASCIZ  'RAM Error, Correct Data Pattern Not In Ram'
6292 044716      124      123      123  T31AM3: .ASCIZ  'TSSR Init. Failed After REREAD COMMAND'
6293 044765      104      162      151  T31OFL: .ASCIZ  'Drive 7 Select Failed To Set "OFL" In TSSR'
6294 045040      124      123      123  T31WDD: .ASCIZ  'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
6295 045130      124      123      123  T31WDC: .ASCIZ  'TSSR Not Correct After REREAD DATA Command'
6296 045203      103      126      103  T31VCK: .ASCIZ  'CVC Set, Didn't Reset VCK In Message Buffer'
6297 045256      124      123      102  T31BA: .ASCIZ   'TSBA Not Correct After REREAD DATA Command'
6298 045331      127      122      111  T31WSS: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
6299 045420      122      145      141  T31LON: .ASCIZ  'Reading Long Record Failed To Set RLL Bit In XSTO'
6300 045502      122      145      141  T31LOP: .ASCIZ  'Reading Long Record Failed To Set RLS Bit In XSTO'
6301 045564      122      145      163  T31PBP: .ASCIZ  'Residual Byte Count Incorrect After Short Record Read'
6302 045652      122      145      141  T31TRL: .ASCIZ  'Reading Long Record Failed To Give Tape Status Alert'
6303 045740      116      117      055  T31NEF: .ASCIZ  'NO-OP ("CLEAN TAPE") AND INITIALIZE, At First Record, Failed To Set RIB Bit
X
6304 046061      124      123      123  T31SCF: .ASCIZ  'TSSR Not Correct After SPACE RECORDS Command'
6305 046136      124      123      123  T31TSA: .ASCIZ  'TSSR Not Correct After NO-OP ("CLEAN TAPE") AND INITIALIZE, Into BOT'
6306 046243      124      123      123  T31WRF: .ASCIZ  'TSSR Not Correct After NO-OP ("CLEAN TAPE") AND INITIALIZE Command'
6307 046346      104      141      164  T31DTA: .ASCIZ  'Data Compare Error, Data Read From Tape Not Equal To Written'
6308 046443      116      117      055  TST31ID: .ASCIZ  'NO-OP ("Clean Tape") And INITIALIZE'
6309
6310
6311
6312
6313
6314
6315
6316
6317 046510
6318 046510
6319 046514      012701      043070
6320 046520      012721      100004
6321 046524      012721      043100
6322 046530      005021
6323 046532      012721      000012
6324 046536      012721      043112
6325 046542      005021
6326 046544      012721      000024
6327 046550      005021
6328 046552      012711      000000
6329 046556      012702      000030

;
;
; ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
; WRITE SUBSYSTEM MEMORY COMMAND
;
;
T31REST:
        SAVREG
        MOV     @T31PACKET,R1
        MOV     @100004,(R1)+
        MOV     @T31DATA,(R1)+
        CLR     (R1)+
        MOV     @10.,(R1)+
        MOV     @T31BFR,(R1)+
        CLR     (R1)+
        MOV     @20.,(R1)+
        CLR     (R1)+
        MOV     @0,(R1)
        MOV     @24.,R2

; SAVE THE REGISTERS
; START OF THE PACKET
; WRITE SUBSYSTEM MEM. WITH ACK,
; ADDRESS OF CHARAISTICS DATA BLOCK
; EXTENDED ADDRESS
; SIZE OF DATA BLOCK IN BYTES
; ADDRESS OF MESSAGE BUFFER
; LENGTH OF MESSAGE BUFFER
; SELECT DRIVE ZERO
; NUMBER OF LOCATIONS TO BE CLEARED
    
```

TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

```

6330 046562 012762 177777 043112 64#: MOV #177777,T31BFR(R2) ;ALL ONES TO MESSAGE BUFFER
6331 046570 005742 TST -(R2) ;NEXT LOCATION
6332 046572 022702 000000 CMP #0,R2 ;AT END OF LOOP YET
6333 046576 001371 BNE 64# ;KEEP GOING UNTIL DONE
6334 046600 000207 RTS PC ;RETURN
6335
6336 046602 T31RT2: SAVREG ;SAVE THE REGISTERS
6337 046602 MOV #T31PK2,R1 ;START OF THE PACKET
6338 046606 012701 043200 MOV #100006,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK.
6339 046612 012721 100006 MOV #T31BF2,(R1)+ ;ADDRESS OF DATA BLOCK
6340 046616 012721 043220 CLR (R1)+ ;EXTENDED ADDRESS
6341 046622 005021 MOV #6,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
6342 046624 012721 000006 CLR (R1)+
6343 046630 005021 CLR (R1)+
6344 046632 012701 043220 MOV #T31BF2,R1 ;POINT TO DATA SEL AREA
6345 046636 005021 CLR (R1)+
6346 046640 005011 CLR (R1)
6347 046642 000207 RTS PC ;RETURN
6348 046644 T31RT3: SAVREG ;SAVE REGISTERS
6349 046644 MOV #T31PK3,R1 ;SET UP POINTER ADDRESS
6350 046650 012701 043210 CLR (R1)+ ;COMMAND SPACE
6351 046654 005021 CLR (R1)+ ;ADDRESS OF DATA BLOCK
6352 046656 005021 CLR (R1)+ ;EXTENDED ADDRESS
6353 046660 005021 CLR (R1) ;SIZE OF DATA TRANSFER BLOCK
6354 046662 005011 RTS PC ;RETURN
6355 046664 000207 ENDTST
6356 046666 L10050: TRAP C#ETST
046666 104401

```

.SBTTL TEST 4: Erase And Operation Incomplete

VERIFIES THAT AN ERASE COMMAND ISSUED WHEN THE TAPE IS POSITIONED AT BOT OPERATES PROPERLY AND ACTUALLY ERASES TAPE. THE FOLLOWING TEST SEQUENCE IS PERFORMED:

1. THE TAPE IS FIRST REWOUND, SEVERAL TEST RECORDS ARE WRITTEN, AND THE TAPE IS REWOUND AGAIN.
2. AN ERASE COMMAND IS ISSUED, WHICH SHOULD ERASE A NUMBER OF THE TEST RECORDS.
3. NORMAL TERMINATION IS VERIFIED AND STATUS IS CHECKED (BOT SHOULD BE 0).
4. A READ REVERSE COMMAND IS ISSUED. IT IS VERIFIED THAT THE COMMAND TERMINATES WITH TAPE STATUS ALERT, THAT THE REVERSE INTO BOT (RIB) STATUS BIT IS SET, AND THAT NO DATA IS TRANSFERRED. THIS DEMONSTRATES THAT NO DATA WAS ENCOUNTERED IN THE AREA ERASED BY THE ERASE COMMAND.

6357  
6358  
6359  
6360  
6361  
6362  
6363  
6364  
6365  
6366  
6367  
6368  
6369  
6370  
6371  
6372  
6373  
6374  
6375  
6376  
6377  
6378  
6379  
6380  
6381  
6382  
6383  
6384

TEST 4: Erase And Operation Incomplete

```

6385 :
6386 :
6387 : THE TEST CONSISTS OF THE FOLLOWING 3 SUBTESTS
6388 :
6389 :
6390 :
6391 :
6392 046670 BGNTST
      046670
6393 046670 012737 006354 002172 MOV #EPRT1,EPRTSW ;PRIMARY ERROR MESSAGE T4::
6398 04667F 012700 052540 MOV #TST32ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
6399 046702 004737 016570 JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
6400 046706 012737 000005 002210 MOV #5,LOOPCNT ;PERFORM 5 ITERATIONS
6401 046714 005037 051410 CLR T32CNT ;CLEAR TAPE RECORD COUNTER
6402 :
6403 :
6404 : TST 4, SUBTEST 1
6405 :
6406 :
6407 : VERIFIES THAT A Erase And Operation Incomplete COMMAND ISSUED WHILE
6408 : THE TAPE IS POSITIONED AT BOT CAUSES FUNCTION REJECT
6409 : TERMINATION, WITH THE NON-EXECUTABLE FUNCTION (NEF)
6410 : ERROR BIT SET.
6411 :
6412 :
6413 :
6414 :
6415 046720 T32LOOP:
6416 :
6417 :
6418 :
6419 046720 BGNSUB ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
      046720 T4.1:
6420 046720 104402 JSR PC,T32REST ;SET COMMAND PACKET TRAP C#BSUB
6421 046726 004737 052600 JSR PC,T32RT2 ;SET UP OTHER COMMAND PACKET
6422 046732 004737 052722 JSR PC,T32RT3 ;SET UP OTHER COMMAND PACKET
6423 046736 012737 176750 051414 MOV #65000,T32DLY ;SET UP DELAY COUNTER
6424 046744 004737 016054 10#: JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
6425 046750 103426 BCS 20# ;BR IF INIT WAS OK
6426 046752 DELAY 250 ;DELAY ABOUT .25 SEC
      046752 012727 000250 MOV #250,(PC)+
      046756 000000 .WORD 0
      046760 013727 002116 MOV L#DLY,(PC)+
      046764 000000 .WORD 0
      046766 005367 177772 DEC -6(PC)
      046772 001375 BNE -4
      046774 005367 177756 DEC -22(PC)
      047000 001367 BNE -20
6427 047002 005337 051414 DEC T32DLY ;BUMP COUNTER
6428 047006 001356 BNE 10# ;BR, IF COUNTER NOT DONE
6429 047010 005237 002214 INC FATFLG ;ERROR COUNT
6433 047014 010001 MOV R0,R1 ;CONTENTS OF TSSR REGISTER
6434 047016 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      047016 104455 TRAP C#ERDF
      047020 000621 .WORD 401
      047022 003646 .WORD SFIERR

```

## TEST 4: Erase And Operation Incomplete

```

047024 012114
6435 047026 013737 002174 051250 20#: MOV UNITN,T32DSW ;SET UP DRIVE NUMBER .WORD SFIMSG
6436 047034 012704 051230 MOV #T32PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
6437 047040 004737 010742 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
6438 047044 103407 BCS 25# ;BR, IF COMMAND ISSUED OK
6439 047046 005237 002214 INC FATFLG ;ERROR COUNT
6443 047052 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
6444 047054 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
047054 104456 TRAP C#ERHRD
047056 000622 .WORD 402
047060 005052 .WORD WRTMSG
047062 012114 .WORD SFIMSG
6445 047064 25#: CKLOOP ;LOOP IF SELECTED
047064 104406 TRAP C#CLP1
6446 047066 004737 011074 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6447 047072 103411 BCS 26# ;BR, IF NO PROBLEM
6448 047074 010004 MOV RO,R4 ;SET UP REWIND PACKET ADDRESS
6449 047076 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6450 047102 005237 002214 INC FATFLG ;ERROR COUNT
6454 047106 ERRHRD ERRNO,T32RWN,PKTSSR ;REWIND NOT ACCEPTED
047106 104456 TRAP C#ERHRD
047110 000623 .WORD 403
047112 051600 .WORD T32RWN
047114 012126 .WORD PKTSSR
6455 047116 26#: CKLOOP ;LOOP IF SELECTED
047116 104406 TRAP C#CLP1
6456 047120 012703 000400 MOV #256.,R3 ;STARTING RECORD SIZE
6457 047122 013737 003116 051352 MOV FREE,T32WB ;STARTING WRITE BUFFER ADDRESS
6458 047132 012737 140005 051350 MOV #140005,T32PK3 ;WRITE DATA,CVC=1,ACK COMMAND
6459 047140 012704 051350 MOV #T32PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6460 047144 010337 051356 27#: MOV R3,T32SZ ;SET UP RECORD SIZE IN PACKET
6461 047150 010465 000000 MOV R4,T32S2 ;ISSUE COMMAND
6462 047154 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
6463 047160 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6464 047164 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6465 047170 020102 CMP R1,R2 ;ARE THEY EQUAL
6466 047172 001406 BEQ 28# ;BR, IF OK
6467 047174 005237 002214 INC FATFLG ;ERROR COUNT
6471 047200 ERRHRD ERRNO,T32WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
047200 104456 TRAP C#ERHRD
047202 000624 .WORD 404
047204 052436 .WORD T32WDC
047206 012126 .WORD PKTSSR
6472 047210 28#: CKLOOP ;LOOP IF SELECTED
047210 104406 TRAP C#CLP1
6473 047212 005723 TST (R3)+ ;BUMP RECORD COUNTER
6474 047214 020327 001002 CMP R3,#514. ;AT MAX SIZE YET
6475 047220 001351 BNE 27# ;BR, IF NOT AT END OF LOOP
6476 047222 004737 011074 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6477 047226 103411 BCS 30# ;BR, IF NO PROBLEM
6478 047230 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6479 047234 010004 MOV RO,R4 ;SET UP REWIND PACKET ADDRESS
6480 047236 005237 002214 INC FATFLG ;ERROR COUNT
6484 047242 ERRHRD ERRNO,T32RWN,PKTSSR ;REWIND NOT ACCEPTED
047242 104456 TRAP C#ERHRD
047244 000625 .WORD 405
047246 051600 .WORD T32RWN

```

TEST 4: Erase And Operation Incomplete

```

6485 047250 012126
      047252 104406          30$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
6486 047254 013701 051260      MOV T32BFR+6,R1 ;PICK UP XSTO TRAP C#CLP1
6487 047260 010102          MOV R1,R2 ;SET UP EXPECTED
6488 047262 052702 000002      BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6489 047266 020102          CMP R1,R2 ;DOES EXP = REC'D
6490 047270 001406          BEQ 40$ ;BR, IF EQUAL (OK)
6491 047272 005237 002214      INC FATFLG ;ERROR COUNT
6495 047276          ERRHRD ERRNO,T32BOE,EXPREC ;TAPE AT BOT AFTER ERASE
      047276 104456          TRAP C#ERHRD
      047300 000626          .WORD 406
      047302 052266          .WORD T32BOE
      047304 015554          .WORD EXPREC
6496 047306          40$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
      047306 104406          ;ERASE TAPE,CVC=1,ACK COMMAND
6497 047310 012737 140411 051350      MOV #140411,T32PK3 ;SET UP R4 WITH PACKET ADDRESS
6498 047316 012704 051350      MOV #T32PK3,R4 ;ISSUE COMMAND
6499 047322 C10465 000000      MOV R4,TSDB(R5) ;WAIT FOR SSR TO SET
6500 047326 004737 016330      JSR PC,WAITF ;GET TSSR CONTENTS
6501 047332 016501 000002      MOV TSSR(R5),R1 ;SET UP EXPECTED
6502 047336 012702 000200      MOV #SSR,R2 ;ARE THEY EQUAL
6503 047342 020102          CMP R1,R2 ;BR, IF OK
6504 047344 001406          BEQ 50$ ;ERROR COUNT
6505 047346 005237 002214      INC FATFLG ;TSSR INCORRECT AFTER ERASE DATA
6509 047352          ERRHRD ERRNO,T32ERA,PKTSSR TRAP C#ERHRD
      047352 104456          .WORD 407
      047354 000627          .WORD T32ERA
      047356 051716          .WORD PKTSSR
      047360 012126
6510 047362          50$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
      047362 104406          ;PICK UP XSTO
6511 047364 013701 051260      MOV T32BFR+6,R1 ;SET UP EXPECTED
6512 047370 010102          MOV R1,R2 ;SET BOT BIT IN EXPECTED
6513 047372 042702 000002      BIC #BIT1,R2 ;DOES EXP = REC'D
6514 047376 020102          CMP R1,R2 ;BR, IF EQUAL (OK)
6515 047400 001406          BEQ 55$ ;ERROR COUNT
6516 047402 005237 002214      INC FATFLG ;TAPE NOT AT BOT AFTER REWIND
6520 047406          ERRHRD ERRNO,T32BOE,EXPREC TRAP C#ERHRD
      047406 104456          .WORD 408
      047410 000630          .WORD T32BOE
      047412 052266          .WORD EXPREC
      047414 015554
6521 047416          55$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
      047416 104406          ;ADDRESS OF BUFFER
6522 047420 013737 003116 051352      MOV FREE,T32RB ;READ REVERSE,ACK,CVC=1 COMMAND
6523 047426 012737 140401 051350      MOV #140401,T32PK3 ;SET UP THE SIZE OF RECORD
6524 047434 012737 000400 051356      MOV #256,T32SZ ;SET UP R4 WITH PACKET ADDRESS
6525 047442 012704 051350      MOV #T32PK3,R4 ;ISSUE COMMAND
6526 047446 010465 000000      MOV R4,TSDB(R5) ;WAIT FOR SSR TO SET
6527 047452 004737 016330      JSR PC,WAITF ;GET TSSR CONTENTS
6528 047456 016501 000002      MOV TSSR(R5),R1 ;SET UP EXPECTED TAPE STATUS ALERT
6529 047462 012702 100204      MOV #SSR:SC:BIT2,R2 ;ARE THEY EQUAL
6530 047466 020102          CMP R1,R2 ;BR, IF OK
6531 047470 001406          BEQ 180$ ;ERROR COUNT
6532 047472 005237 002214      INC FATFLG ;TSSR INCORRECT AFTER READ DATA
6536 047476          ERRHRD ERRNO,T32TSA,PKTSSR

```







TEST 4: Erase And Operation Incomplete

6634	047756	013737	003116	051352		MOV	FREE,T32WB		:STARTING WRITE BUFFER ADDRESS
6635	047764	012737	140005	051350	65#:	MOV	#140005,T32PK3		:WRITE DATA,CVC-1,ACK COMMAND
6636	047772	012704	051350			MOV	#T32PK3,R4		:SET UP R4 WITH PACKET ADDRESS
6637	047776	010300				MOV	R3,R0		:SET PATTERN IN CORRECT REGISTER
6638	050000	004737	017502			JSR	PC,FILLMEM		:FILL MEMORY WITH RECORD SIZE
6639	050004	010337	051356			MOV	R3,T32SZ		:SET UP RECORD SIZE IN PACKET
6640	050010	010465	000000			MOV	R4,TSDB(R5)		:ISSUE COMMAND
6641	050014	004737	016330			JSR	PC,WAITF		:WAIT FOR SSR TO SET
6642	050020	016501	000002			MOV	TSSR(R5),R1		:GET TSSR CONTENTS
6643	050024	012702	000200			MOV	#SSR,R2		:SET UP EXPECTED
6644	050030	020102				CMP	R1,R2		:ARE THEY EQUAL
6645	050032	001406				BEQ	80#		:BR, IF OK
6646	050034	005237	002214			INC	FATFLG		:ERROR COUNT
6650	050040					ERRHRD	ERRNO,T32WDC,PKTSSR		:TSSR INCORRECT AFTER WRITE DATA
	050040	104456						TRAP	C#ERHRD
	050042	000637						.WORD	415
	050044	052436						.WORD	T32WDC
	050046	012126						.WORD	PKTSSR
6651	050050				80#:	CKLOOP			:LOOP IF SELECTED
	050050	104406						TRAP	C#CLP1
6652	050052	005723				TST	(R3)+		:BUMP RECORD SIZE COUNTER
6653	050054	020327	000156			CMP	R3,#110.		:AT 160 SIZE YET
6654	050060	001341				BNE	65#		:BR, IF MORE RECORDS TO WRITE
6655	050062	004737	011074			JSR	PC,REWIND		:CALL TAPE REWIND COMMAND
6656	050066	103407				BCS	230#		:BR, IF NO PROBLEM
6657	050070	010001				MOV	R0,R1		:SAVE TSSR
6658	050072	005237	002214			INC	FATFLG		:ERROR COUNT
6662	050076					ERRHRD	ERRNO,T32RWN,EXPREC		:REWIND NOT ACCEPTED
	050076	104456						TRAP	C#ERHRD
	050100	000640						.WORD	416
	050102	051600						.WORD	T32RWN
	050104	015554						.WORD	EXPREC
6663	050106				230#:	CKLOOP			:LOOP IF SELECTED
	050106	104406						TRAP	C#CLP1
6664	050110	013701	051260			MOV	T32BFR+6,R1		:PICK UP XSTO
6665	050114	010102				MOV	R1,R2		:SET UP EXPECTED
6666	050116	052702	000002			BIS	#BIT1,R2		:SET BOT BIT IN EXPECTED
6667	050122	020102				CMP	R1,R2		:DOES EXP = REC'D
6668	050124	001406				BEQ	240#		:BR, IF EQUAL (OK)
6669	050126	005237	002214			INC	FATFLG		:ERROR COUNT
6673	050132					ERRHRD	ERRNO,T32BOT,EXPREC		:TAPE NOT AT BOT AFTER REWIND
	050132	104456						TRAP	C#ERHRD
	050134	000641						.WORD	417
	050136	051416						.WORD	T32BOT
	050140	015554						.WORD	EXPREC
6674	050142				240#:	CKLOOP			:LOOP IF SELECTED
	050142	104406						TRAP	C#CLP1
6675	050144	012703	000001			MOV	#1,R3		:SET UP FOR SPACE COMMAND
6676	050150	004737	010544			JSR	PC,SPACE		:ISSUE SPACE COMMAND 1 FORWARD
6677	050154	012737	140411	051350	265#:	MOV	#140411,T32PK3		:ERASE DATA,ACK COMMAND
6678	050162	012704	051350			MOV	#T32PK3,R4		:SET UP R4 WITH PACKET ADDRESS
6679	050166	010465	000000			MOV	R4,TSDB(R5)		:ISSUE COMMAND
6680	050172	004737	016330			JSR	PC,WAITF		:WAIT FOR SSR TO SET
6681	050176	016501	000002			MOV	TSSR(R5),R1		:GET TSSR CONTENTS
6682	050202	012702	000200			MOV	#SSR,R2		:SET UP EXPECTED
6683	050206	020102				CMP	R1,R2		:ARE THEY EQUAL
6684	050210	001406				BEQ	280#		:BR, IF OK





TEST 4: Erase And Operation Incomplete

6780	050516	004737	010742		JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS		
6781	050522	103407			BCS	23:		;BR, IF COMMAND ISSUED OK		
6782	050524	005237	002214		INC	FATFLG		;ERROR COUNT		
6786	050530	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR		
6787	050532				ERRHRD	ERRNO,WRTMSG,SFMSG		;WRITE CHARACTERISTIC FAILED		
	050532	104456						TRAP	C:ERHRD	
	050534	000646						.WORD	422	
	050536	005052						.WORD	WRTMSG	
	050540	012114						.WORD	SFMSG	
6788	050542			23:	CKLOOP			;LOOP IF SELECTED		
	050542	104406						TRAP	C:CLP1	
6789	050544	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND		
6790	050550	103411			BCS	30:		;BR, IF NO PROBLEM		
6791	050552	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
6792	050556	010004			MOV	R0,R4		;GET PACKET ADDRESS		
6793	050560	005237	002214		INC	FATFLG		;ERROR COUNT		
6797	050564				ERRHRD	ERRNO,T32RMN,PKTSSR		;REWIND NOT ACCEPTED		
	050564	104456						TRAP	C:ERHRD	
	050566	000647						.WORD	423	
	050570	051600						.WORD	T32RMN	
	050572	012126						.WORD	PKTSSR	
6798	050574			30:	CKLOOP			;LOOP IF SELECTED		
	050574	104406						TRAP	C:CLP1	
6799	050576	013701	051260		MOV	T32BFR+6,R1		;PICK UP XSTO		
6800	050602	010102			MOV	R1,R2		;SET UP EXPECTED		
6801	050604	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED		
6802	050610	020102			CMP	R1,R2		;DOES EXP = REC'D		
6803	050612	001406			BEQ	40:		;BR, IF EQUAL (OK)		
6804	050614	005237	002214		INC	FATFLG		;ERROR COUNT		
6808	050620				ERRHRD	ERRNO,T32BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND		
	050620	104456						TRAP	C:ERHRD	
	050622	000650						.WORD	424	
	050624	051416						.WORD	T32BOT	
	050626	015554						.WORD	EXPREC	
6809	050630			40:	CKLOOP			;LOOP IF SELECTED		
	050630	104406						TRAP	C:CLP1	
6810	050632	012737	140411	051350	65:	MOV	#140411,T32PK3	;ERASE DATA,CVC-1,ACK COMMAND		
6811	050640	012704	051350		MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
6812	050644	010337	051356		MOV	R3,T32SZ		;SET UP RECORD SIZE IN PACKET		
6813	050650	010465	000000		MOV	R4,TSD8(R5)		;ISSUE COMMAND		
6814	050654	004737	016330		JSR	PC,WAIF		;WAIT FOR SSR TO SET		
6815	050660	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
6816	050664	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED		
6817	050670	020102			CMP	R1,R2		;ARE THEY EQUAL		
6818	050672	001757			BEQ	65:		;BR, IF OK		
6819	050674	032701	000004		BIT	#BIT2,R1		;CHECK FOR TAPE STATUS ALERT		
6820	050700	001006			BNE	80:		;BR, IF TAPE STATUS ALERT SET		
6821	050702	005237	002214		INC	FATFLG		;ERROR COUNT		
6825	050706				ERRHRD	ERRNO,T32MDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA		
	050706	104456						TRAP	C:ERHRD	
	050710	000651						.WORD	425	
	050712	052436						.WORD	T32MDC	
	050714	012126						.WORD	PKTSSR	
6826	050716			80:	CKLOOP			;LOOP IF SELECTED		
	050716	104406						TRAP	C:CLP1	
6827	050720	013701	051260		MOV	T32BFR+6,R1		;PICK UP XSTO		
6828	050724	010102			MOV	R1,R2		;SET UP EXPECTED		

TEST 4: Erase And Operation Incomplete

```

6829 050726 052702 000001      BIS      #BIT0,R2      ;SET EOT BIT IN EXPECTED
6830 050732 020102      CMP      R1,R2        ;DOES EXP = REC'D
6831 050734 001406      BEQ     240#          ;BR, IF EQUAL (OK)
6832 050736 005237 002214      INC     FATFLG        ;ERROR COUNT
6836 050742      ERRHRD  ERRNO,T32EOT,EXPREC ;TAPE NOT AT EOT AFTER ERASE COMMANDS
        050742 104456      TRAP   C#ERHRD
        050744 000652      .WORD  426
        050746 051511      .WORD  T32EOT
        050750 015554      .WORD  EXPREC
6837 050752      240# : CKLOOP      ;LOOP IF SELECTED
        050752 104406      TRAP   C#CLP1
6838 050754 012703 051360      MOV     #T32CMD,R3    ;STARTING RECORD SIZE
6839 050760 013737 003116 051352      MOV     FREE,T32RB    ;STARTING READ BUFFER ADDRESS
6840 050766 011337 051350      MOV     (R3),T32PK3   ;READ DATA,ACK COMMAND
6841 050772 012704 051350      MOV     #T32PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
6842 050776 012700 177777      MOV     #177777,R0     ;SET PATTERN IN CORRECT REGISTER
6843 051002 004737 017502      JSR     PC,FILLMEM     ;FILL MEMORY WITH ALL ONES
6844 051006 012737 000144 051356      MOV     #100.,T32SZ    ;SET UP RECORD SIZE IN PACKET
6845 051014 C10465 000000      MOV     R4,TSD8(R5)   ;ISSUE COMMAND
6846 051020 012737 000062 051414      MOV     #50.,T32DLY   ;SET UP DELAY COUNTER
6847 051026 004737 016330      JSR     PC,WAITF      ;WAIT FOR SSR TO SET
6848 051032 016501 000002      MOV     TSSR(R5),R1   ;GET TSSR CONTENTS
6849 051036 012702 100214      MOV     #SSR!SC!BIT2!BIT3,R2 ;SET UP EXPECTED
6850 051042 020102      CMP     R1,R2         ;ARE THEY EQUAL
6851 051044 001425      BEQ     280#          ;BR, IF OK
6852 051046      DELAY  250           ;DELAY FOR SSR TO BE SET
        051046 012727 000250      MOV     #250,(PC)+
        051052 000000      .WORD  0
        051054 013727 002116      MOV     L#DLY,(PC)+
        051060 000000      .WORD  0
        051062 005367 177772      DEC     -6(PC)
        051066 001375      BNE     -.4
        051070 005367 177756      DEC     -22(PC)
        051074 001367      BNE     -.20
6853 051076 005337 051414      DEC     T32DLY        ;COUNT DELAY ROUTINE DOWN
6854 051102 001351      BNE     270#          ;BR, IF DELAY HAS NOT ENDED
6855 051104 005237 002214      INC     FATFLG        ;ERROR COUNT
6859 051110      ERRHRD  ERRNO,T32ECF,PKTSSR ;TSSR INCORRECT AFTER READ DATA
        051110 104456      TRAP   C#ERHRD
        051112 000653      .WORD  427
        051114 052355      .WORD  T32ECF
        051116 012126      .WORD  PKTSSR
6860 051120      280# : CKLOOP      ;LOOP IF SELECTED
        051120 104406      TRAP   C#CLP1
6861 051122 013701 051266      MOV     T32BFR+14,R1  ;PICK UP XST3
6862 051126 010102      MOV     R1,R2         ;SET UP EXPECTED
6863 051130 052702 000100      BIS     #BIT6,R2      ;SET OPI BIT IN EXPECTED
6864 051134 020102      CMP     R1,R2         ;IS OPI BIT SET
6865 051136 001406      BEQ     290#          ;BR, IF BIT IS SET
6866 051140 005237 002214      INC     FATFLG        ;ERROR COUNT
6870 051144      ERRHRD  ERRNO,T32OPI,EXPREC ;OPI BIT NOT SET
        051144 104456      TRAP   C#ERHRD
        051146 000654      .WORD  428
        051150 052503      .WORD  T32OPI
        051152 015554      .WORD  EXPREC
6871 051154      290# : CKLOOP      ;LOOP IF SELECTED
        051154 104406      TRAP   C#CLP1
    
```





TEST 4: Erase And Operation Incomplete

```

6932
6933
6934
6935 051360
6936 051360 140410
6937 051362 141410
6938 051364 140401
6939 051366 141001
6940 051370 161401
6941 051372 161001
6942 051374 141401
6943 051376 140001
6944 051400 141410
6945 051402 141010
6946 051404 141005
6947 051406 177777
6948
6949
6950 051410 C00000
6951 051412 000000
6952 051414 000000
6953
6954
6955
6956
6957 051416 124 141 160
6958 051511 124 141 160
6959 051600 122 145 167
6960 051647 124 123 123
6961 051716 124 123 123
6962 051763 124 123 102
6963 052036 122 105 101
6964 052134 124 123 123
6965 052211 124 123 123
6966 052266 102 117 124
6967 052355 105 122 101
6968 052436 124 123 123
6969 052503 117 120 111
6970 052540 105 162 141
6971
6972
6973
6974
6975
6976
6977
6978
6979 052600
6980 052600
6981 052604 012701 051230
6982 052610 012721 100004
6983 052614 012721 051240
6984 052620 005021
6985 052622 012721 000012
6986 052626 012721 051252
6987 052632 005021
6988 052634 012721 000024

;TAPES MOTION PACKET COMMAND VALUES
T32CMD:
        .WORD 140410 ;SPACE RECORDS REVERSE
        .WORD 141410 ;SKIP TAPE MARKS REVERSE
        .WORD 140401 ;READ REVERSE
        .WORD 141001 ;REREAD PREVIOUS (OPP=0)
        .WORD 161401 ;REREAD NEXT (OPP=1)
        .WORD 161001 ;REREAD PREVIOUS (OPP=1)
        .WORD 141401 ;REREAD NEXT (OPP=0)
        .WORD 140001 ;READ NEXT
        .WORD 141410 ;SKIP TAPE MARKS REVERSE
        .WORD 141010 ;SKIP RECORDS FORWARD
        .WORD 141005 ;WRITE DATA RETRY
        .WORD 177777 ;END OF DATA

;
T32CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T32CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T32DLY: .WORD 0 ;DELAY COUNTER

;*
;LOCAL TEXT MESSAGES FOR TEST
;-
T32BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
T32EOT: .ASCIZ 'Tape Status Alert During Erase To EOT, But EOT Not Set'
T32RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
T32AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
T32ERA: .ASCIZ 'TSSR Not Correct After ERASE Command'
T32BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
T32RIB: .ASCIZ 'READ REVERSE, After ERASE From BOT, Failed To Set RIB In XST3'
T32SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
T32TSA: .ASCIZ 'TSSR Not Correct After READ REVERSE Into BOT'
T32BOE: .ASCIZ 'BOT (XST0) Still Set After Erase From Tape's BOT Marker'
T32ECF: .ASCIZ 'ERASE Failed To Clear Tape (Erase) Tape Properly'
T32WDC: .ASCIZ 'TSSR Not Correct After ERASE Command'
T32OPI: .ASCIZ 'OPI Bit (XST3) Failed To Set'
TST32ID: .ASCIZ 'Erase And Operation Incomplete'

;
;
;*
;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
;WRITE SUBSYSTEM MEMORY COMMAND
;
;-
T32REST:
        SAVREG ;SAVE THE REGISTERS
        MOV #T32PACKET,R1 ;START OF THE PACKET
        MOV #100004,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK.
        MOV #T32DATA,(R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
        CLR (R1)+ ;EXTENDED ADDRESS
        MOV #10.,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
        MOV #T32BFR,(R1)+ ;ADDRESS OF MESSAGE BUFFER
        CLR (R1)+
        MOV #20.,(R1)+ ;LENGTH OF MESSAGE BUFFER
    
```

## TEST 4: Erase And Operation Incomplete

```

6989 052640 005021          CLR      (R1)+
6990 052642 012711 000000    MOV      #0,(R1)          ;SELECT DRIVE ZERO
6991 052646 012702 000030    MOV      #24.,R2        ;NUMBER OF LOCATIONS TO BE CLEARED
6992 052652 012762 177777 051252 64#: MOV      #177777,T32BFR(R2) ;ALL ONES TO MESSAGE BUFFER
6993 052660 005742          TST      -(R2)          ;NEXT LOCATION
6994 052662 022702 000000    CMP      #0,R2          ;AT END OF LOOP YET
6995 052666 001371          BNE      64#           ;KEEP GOING UNTIL DONE
6996 052670 000207          RTS      PC             ;RETURN
6997
6998 052672          T32RT2:
6999 052672          SAVREG          ;SAVE THE REGISTERS
7000 052676 012701 051340    MOV      #T32PK2,R1     ;START OF THE PACKET
7001 052702 012721 100006    MOV      #100006,(R1)+  ;WRITE SUBSYSTEM MEM. WITH ACK.
7002 052706 005021          CLR      (R1)+         ;ADDRESS OF DATA BLOCK
7003 052710 005021          CLR      (R1)+         ;EXTENDED ADDRESS
7004 052712 012721 000006    MOV      #6.,(R1)+     ;SIZE OF DATA BLOCK IN BYTES
7005 052716 005021          CLR      (R1)+
7006 052720 000207          RTS      PC             ;RETURN
7007 052722          T32RT3:
7008 052722          SAVREG          ;SAVE REGISTERS
7009 052726 012701 051350    MOV      #T32PK3,R1     ;SET UP POINTER ADDRESS
7010 052732 005021          CLR      (R1)+         ;COMMAND SPACE
7011 052734 005021          CLR      (R1)+         ;ADDRESS OF DATA BLOCK
7012 052736 005021          CLR      (R1)+         ;EXTENDED ADDRESS
7013 052740 005011          CLR      (R1)         ;SIZE OF DATA TRANSFER BLOCK
7014 052742 000207          RTS      PC             ;RETURN
7015 052744          ENDTST
7016 052744 104401          L10053: TRAP C#ETST

```

.SBTTL TEST 5: DATA PARITY TEST

```

7016
7017
7018
7019
7020
7021
7022
7023 ;TEST 5 -- Data Parity Test
7024
7025
7026 ;This test verifies that the data parity circuitry in both the controller and the
7027 ;transport is operating properly by forcing data records with wrong parity to be
7028 ;written onto tape and checking the results obtained when the data is read. The
7029 ;following test sequence is performed:
7030
7031 ;
7032 ; 1. A Write Characteristics command is issued and the resulting status is
7033 ; examined to determine the states of the Extended Features and Buffering
7034 ; Enable switches on the controller module. If buffering is disabled, no
7035 ; further actions need be taken in this step and the program proceeds to
7036 ; the next step. If buffering is enabled, it is disabled via the Buffer
7037 ; Control field in the extended characteristics data word supplied by a
7038 ; Write Characteristics command. (The module must be in Extended mode,
7039 ; so if it is not already, a Write Subsystem Memory command is issued to
7040 ; change the logical sense of the Extended Features switch.)
7041 ;
7042 ; 2. The Write Subsystem Memory command is used to set the Force Wrong
7043 ; Parity control flip-flop.

```

TEST 5: DATA PARITY TEST

- 7044 : 3. The tape is rewound.
- 7045 :
- 7046 : 4. A Write Data command is issued to write a data record containing all
- 7047 : 0's. It is verified that this command results in Recoverable Error
- 7048 : termination (TC=4) and that the Uncorrectable Data Error (UNC) error
- 7049 : bit is set.
- 7050 :
- 7051 : 5. The previous step is repeated for each data value 2 through 377
- 7052 : (octal).
- 7053 :
- 7054 : 6. The tape is rewound.
- 7055 :
- 7056 : 7. A Read Next command is issued to read a record with faulty parity. It
- 7057 : is verified that this command results in Recoverable Error termination
- 7058 : (TC=4) and that both the Uncorrectable Data (UNC) and Read Bus Parity
- 7059 : (RBP) error bits are set. It is also verified that the data actually
- 7060 : read is correct.
- 7061 :
- 7062 : 8. A Read Reverse command with OPP=1 is issued to read, in reverse, the
- 7063 : same record with faulty parity as read in the previous step. It is
- 7064 : verified that this command results in Recoverable Error termination
- 7065 : (TC=4) and that both the Uncorrectable Data (UNC) and Read Bus Parity
- 7066 : (RBP) error bits are set. It is also verified that the data actually
- 7067 : read is correct.
- 7068 :
- 7069 : 9. Tape is spaced forward one record.
- 7070 :
- 7071 : 10. The previous three steps are executed for each test record originally
- 7072 : written.
- 7073 :
- 7074 : 11. The controller is initialized to clear the special test conditions
- 7075 : previously set up.

```

7080 052746      BGNTST
      052746
7081 052746 012737 006413 002172      MOV    #EPRT2,EPRTSW      ;SECONDARY ERROR MESSAGE
7086 052754 012700 055545              MOV    #TST33ID,R0       ;ASCII MESSAGE TO IDENTIFY TEST
7087 052760 004737 016570              JSR    PC,TSTSETUP       ;DO INITIAL TEST SETUP
7088 052764 012737 000005 002210      MOV    #5,LOOPCNT       ;PERFORM 5 ITERATIONS
7089 052772 005037 054616              CLR    T33CNT           ;CLEAR TAPE RECORD COUNTER
7090
7091          ;+
7092 052776      T33LOOP:
7093 052776      BGNSUB
      052776
      052776 104402
7094 053000 005037 002216              CLR    INTRECV          ;INTERRUPT INDICATOR
7095 053004 005037 054616              CLR    T33CNT          ;TIMER FOR WRITE DATA SPACING
7096 053010 005037 054620              CLR    T33CNU         ;TIMER FOR WRITE DATA RETRY SPACING
7097 053014 004737 055562              JSR    PC,T33REST       ;SET COMMAND PACKET
7098 053020 004737 055654              JSR    PC,T33RT2        ;SET UP OTHER COMMAND PACKET
7099 053024 004737 055716              JSR    PC,T33RT3        ;SET UP OTHER COMMAND PACKET
7100 053030 012737 176750 054622      MOV    #65000.,T33DLY   ;SET UP DELAY COUNTER
7101 053036 004737 016054 10$:      JSR    PC,SOFINIT       ;DO INITIALIZE ON CONTROLLER

```

TEST 5: DATA PARITY TEST

```

7102 053042 103426          BCS      20#           ;BR IF INIT WAS OK
7103 053044          DELAY    250           ;DELAY ABOUT .25 SEC
      053044 012727 000250          MOV      #250,(PC)+
      053050 000000          .WORD    0
      053052 013727 002116          MOV      L#DLY,(PC)+
      053056 000000          .WORD    0
      053060 005367 177772          DEC      -6(PC)
      053064 001375          BNE     .-4
      053066 005367 177756          DEC     -22(PC)
      053072 001367          BNE     .-20
7104 053074 005337 054622          DEC      T33DLY          ;BUMP COUNTER
7105 053100 001356          BNE     10#           ;BR, IF COUNTER NOT DONE
7106 053102 005237 002214          INC     FATFLG          ;ERROR COUNT
7110 053106 010001          MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
7111 053110          ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      053110 104455          TRAP   C#ERDF
      053112 000765          .WORD  501
      053114 003646          .WORD  SFIERR
      053116 C12114          .WORD  SFIMSG
7112 053120 013737 002174 054470 20# : MOV     UNITN,T33DSW          ;SET UP UNIT NUMBER
7113 053126 012704 054450          MOV     #T33PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
7114 053132 004737 010742          JSR    PC,WRTCHR         ;ISSUE WRITE CHARACTERISTICS
7115 053136 103407          BCS    23#           ;BR, IF COMMAND ISSUED OK
7116 053140 005237 002214          INC     FATFLG          ;ERROR COUNT
7117 053144 010001          MOV     R0,R1           ;SAVE CONTENTS OF TSSR
7122 053146          ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      053146 104456          TRAP   C#ERHRD
      053150 000766          .WORD  502
      053152 005052          .WORD  WRTMSG
      053154 012114          .WORD  SFIMSG
7123 053156          23# : CKLOOP          ;LOOP IF SELECTED
      053156 104406          TRAP   C#CLP1
7124 053160 004737 011074          JSR    PC,REWIND         ;CALL TAPE REWIND COMMAND
7125 053164 103411          BCS    30#           ;BR, IF NO PROBLEM
7126 053166 016501 000002          MOV     TSSR(R5),R1      ;GET TSSR CONTENTS
7127 053172 010004          MOV     R0,R4           ;GET PACKET ADDRESS
7128 053174 005237 002214          INC     FATFLG          ;ERROR COUNT
7132 053200          ERRHRD ERRNO,T33RWN,PKTSSR ;REWIND NOT ACCEPTED
      053200 104456          TRAP   C#ERHRD
      053202 000767          .WORD  503
      053204 055320          .WORD  T33RWN
      053206 012126          .WORD  PKTSSR
7133 053210          30# : CKLOOP          ;LOOP IF SELECTED
      053210 104406          TRAP   C#CLP1
7134 053212 013701 054500          MOV     T33BFR+6,R1      ;PICK UP XSTO
7135 053216 010102          MOV     R1,R2           ;SET UP EXPECTED
7136 053220 052702 000002          BIS    #BIT1,R2         ;SET BOT BIT IN EXPECTED
7137 053224 020102          CMP    R1,R2           ;DOES EXP = REC'D
7138 053226 001406          BEQ    40#           ;BR, IF EQUAL (OK)
7139 053230 005237 002214          INC     FATFLG          ;ERROR COUNT
7143 053234          ERRHRD ERRNO,T33BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      053234 104456          TRAP   C#ERHRD
      053236 000770          .WORD  504
      053240 055225          .WORD  T33BOT
      053242 015554          .WORD  EXPREC
7144 053244          40# : CKLOOP          ;LOOP IF SELECTED

```

TEST 5: DATA PARITY TEST

Line	Address	Code	Hex	Hex	Label	Comments
7145	053244	104406				
7145	053246	005737	002220	42:	TST EXTFEA	;CHECK FOR EXTENDED FEATURES SW SWITCH
7146	053252	001025			BNE 55:	;BR IF SWITCH IS ON
7147	053254	112737	000200	054601	MOVB #200,T33BS1	;WRITE MISCELLANEOUS CONT/READ STATUS
7148	053262	112737	000010	054600	MOVB #10,T33BS0	;FUNC. SEL. BIT (TURN ON EXTFEA SWITCH)
7149	053270	012704	054560		MOV #T33PK2,R4	;WRITE SUBSYS MEM PACKET
7150	053274	010465	000000		MOV R4,TSDB(R5)	;ISSUE COMMAND
7151	053300	004737	016416		JSR PC,CHKTSSR	;WAIT FOR SSR
7152	053304	103407			BCS 50:	;BR, IF NO ERROR
7153	053306	010001			MOV R0,R1	;ERROR, SAVE TSSR
7154	053310	005237	002214		INC FATFLG	;ERROR COUNT
7158	053314				ERRHRD ERRNO,T33SSR,PKTSSR	;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
	053314	104456				TRAP C#ERHRD
	053316	000771				.WORD 505
	053320	055141				.WORD T33SSR
	053322	012126				.WORD PKTSSR
7159	053324			50:	CKLOOP	;LOOP IF SELECTED
	053324	104406				TRAP C#CLP1
7160	053326	005737	002222	55:	TST BENBSW	;CHECK FOR BUFFER ENABLED
7161	053332	001426			BEQ 70:	;BR, IF BUFFERING NOT ENABLED
7162	053334	013737	002174	054470	MOV UNITN,T33DSW	;SET UP UNIT NUMBER
7163	053342	042737	000020	054470	BIC #BIT4,T33DSW	;BUFFER DISABLE
7164	053350	052737	000010	054470	BIS #BIT3,T33DSW	;BUFFER DISABLE SEND 01 TO BITS 4 AND 3
7165	053356	012704	054450		MOV #T33PACKET,R4	;SUBROUTINE NEEDS PACKET ADDRESS
7166	053362	004737	010742		JSR PC,WRTCHR	;ISSUE WRITE CHARACTERISTICS
7167	053366	103407			BCS 60:	;BR, IF COMMAND ISSUED OK
7168	053370	005237	002214		INC FATFLG	;ERROR COUNT
7172	053374	010001			MOV R0,R1	;SAVE CONTENTS OF TSSR
7173	053376				ERRHRD ERRNO,WRTMSG,SFIMSG	;WRITE CHARACTERISTICSC FAILED
	053376	104456				TRAP C#ERHRD
	053400	000772				.WORD 506
	053402	005052				.WORD WRTMSG
	053404	012114				.WORD SFIMSG
7174	053406			60:	CKLOOP	;LOOP IF SELECTED
	053406	104406				TRAP C#CLP1
7175	053410			70:		
7176	053410	112737	000100	054601	MOVB #100,T33BS1	;WRITE MISCELLANEOUS CONT/READ STATUS
7177	053416	112737	000011	054600	MOVB #11,T33BS0	;FUNC. SEL. BIT (SET WRONG PARITY)
7178	053424	012704	054560		MOV #T33PK2,R4	;WRITE SUBSYS MEM PACKET
7179	053430	010465	000000		MOV R4,TSDB(R5)	;ISSUE COMMAND
7180	053434	004737	016416		JSR PC,CHKTSSR	;WAIT FOR SSR
7181	053440	103407			BCS 80:	;BR, IF NO ERROR
7182	053442	010001			MOV R0,R1	;ERROR, SAVE TSSR
7183	053444	005237	002214		INC FATFLG	;ERROR COUNT
7187	053450				ERRHRD ERRNO,T33SSR,PKTSSR	;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
	053450	104456				TRAP C#ERHRD
	053452	000773				.WORD 507
	053454	055141				.WORD T33SSR
	053456	012126				.WORD PKTSSR
7188	053460			80:	CKLOOP	;LOOP IF SELECTED
	053460	104406				TRAP C#CLP1
7189	053462	012703	000026		MOV #22.,R3	;NUMBER OF RECORDS TO BE WRITTEN
7190	053466	013737	003116	054572	MOV FREE,T33WB	;STARTING WRITE BUFFER ADDRESS
7191	053474	005037	054620		CLR T33CNU	;MAKE SURE ITS CLEAR
7192	053500	012737	140005	054570	110:	MOV #140005,T33PK3
7193	053506	012704	054570		MOV #T33PK3,R4	;WRITE DATA,ACK,CVC=1 COMMAND
7194	053512	012737	000024	054576	MOV #20.,T33SZ	;SET UP R4 WITH PACKET ADDRESS
						;SET UP RECORD SIZE IN PACKET

TEST 5: DATA PARITY TEST

```

7195 053520 013777 054620 127370      MOV      T33CNU, @FREE      ;MEMORY FILLED WITH DATA IN RECORD
7196 053526 005237 054620              INC      T33CNU            ;READY FOR NEXT RECORD
7197 053532 010465 000000      MOV      R4, TSD8(R5)     ;ISSUE COMMAND
7198 053536 004737 016330      JSR      PC, WAITF        ;WAIT FOR SSR TO SET
7199 053542 016501 000002      MOV      TSSR(R5), R1     ;GET TSSR CONTENTS
7200 053546 012702 100210      MOV      @SSR!SC!BIT3, R2 ;SET UP EXPECTED
7201 053552 020102              CMP      R1, R2           ;ARE THEY EQUAL
7202 053554 001406              BEQ      120$             ;BR, IF OK
7203 053556 005237 002214      INC      FATFLG           ;ERROR COUNT
7207 053562              ERRHRD  ERRNO, T33WPW, PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C#ERHRD
                                .WORD    508
                                .WORD    T33WPW
                                .WORD    PKTSSR
7208 053572              120$:  CKLOOP            ;LOOP IF SELECTED
                                TRAP      C#CLP1
7209 053574 013701 054502      MOV      T33BFR+10, R1    ;PICK UP XST1
7210 053600 010102              MOV      R1, R2           ;SET UP EXPECTED
7211 053602 052702 000002      BIS      @BIT1, R2        ;SET UNC BIT IN EXPECTED
7212 053606 020102              CMP      R1, R2           ;DOES EXP = REC'D
7213 053610 001406              BEQ      130$             ;BR, IF EQUAL (OK)
7214 053612 005237 002214      INC      FATFLG           ;ERROR COUNT
7218 053616              ERRHRD  ERRNO, T33UNC, EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C#ERHRD
                                .WORD    509
                                .WORD    T33UNC
                                .WORD    EXPREC
7219 053626              130$:  CKLOOP            ;LOOP IF SELECTED
                                TRAP      C#CLP1
7220 053630 005303              DEC      R3               ;DEC RECORD COUNTER
7221 053632 001322              BNE      110$             ;BR, IF MORE RECORDS TO WRITE
7222 053634 004737 011074      JSR      PC, REWIND       ;CALL TAPE REWIND COMMAND
7223 053640 103411              BCS      140$             ;BR, IF NO PROBLEM
7224 053642 016501 000002      MOV      TSSR(R5), R1     ;GET TSSR CONTENTS
7225 053646 010004              MOV      R0, R4           ;GET PACKET ADDRESS
7226 053650 005237 002214      INC      FATFLG           ;ERROR COUNT
7230 053654              ERRHRD  ERRNO, T33RWN, PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C#ERHRD
                                .WORD    510
                                .WORD    T33RWN
                                .WORD    PKTSSR
7231 053664              140$:  CKLOOP            ;LOOP IF SELECTED
                                TRAP      C#CLP1
7232 053666 013701 054500      MOV      T33BFR+6, R1     ;PICK UP XST0
7233 053672 010102              MOV      R1, R2           ;SET UP EXPECTED
7234 053674 052702 000002      BIS      @BIT1, R2        ;SET BOT BIT IN EXPECTED
7235 053700 020102              CMP      R1, R2           ;DOES EXP = REC'D
7236 053702 001406              BEQ      150$             ;BR, IF EQUAL (OK)
7237 053704 005237 002214      INC      FATFLG           ;ERROR COUNT
7241 053710              ERRHRD  ERRNO, T33BOT, EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C#ERHRD
                                .WORD    511
                                .WORD    T33BOT
                                .WORD    EXPREC
7242 053720              150$:  CKLOOP            ;LOOP IF SELECTED
7243 053722 005037 054620      CLR      T33CNU           ;CLEAR DATA VALUE IN RECORD
                                TRAP      C#CLP1
    
```

TEST 5: DATA PARITY TEST

7244	053726	012703	000024		MOV	#20.,R3		;RECORD SIZE
7245	053732	013737	003116	054572	155#:	MOV	FREE,T33RB	;STARTING WRITE BUFFER ADDRESS
7246	053740	012737	140001	054570		MOV	#140001,T33PK3	;READ DATA,CVC-1,ACK COMMAND
7247	053746	012704	054570		MOV	#T33PK3,R4		;SET UP R4 WITH PACKET ADDRESS
7248	053752	012737	000024	054576		MOV	#20.,T33SZ	;SET UP RECORD SIZE IN PACKET
7249	053760	010465	000000		MOV	R4,T5DB(R5)		;ISSUE COMMAND
7250	053764	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET
7251	053770	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
7252	053774	012702	100210		MOV	#SSR!SC!BIT3,R2		;SET UP EXPECTED
7253	054000	020102			CMP	R1,R2		;ARE THEY EQUAL
7254	054002	001406			BEQ	160#		;BR, IF OK
7255	054004	005237	002214		INC	FATFLG		;ERROR COUNT
7259	054010				ERRHRD	ERRNO,T33WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA
	054010	104456					TRAP	C#ERHRD
	054012	001000					.WORD	512
	054014	055367					.WORD	T33WDC
	054016	012126					.WORD	PKTSSR
7260	054020				160#:	CKLOOP		;LOOP IF SELECTED
	054020	104406					TRAP	C#CLP1
7261	054022	013701	054502		MOV	T33BFR+10,R1		;PICK UP XST1
7262	054026	010102			MOV	R1,R2		;SET UP EXPECTED
7263	054030	052702	000002		BIS	#BIT1,R2		;SET UNC BIT IN EXPECTED
7264	054034	020102			CMP	R1,R2		;DOES EXP = REC'D
7265	054036	001406			BEQ	170#		;BR, IF EQUAL (OK)
7266	054040	005237	002214		INC	FATFLG		;ERROR COUNT
7270	054044				ERRHRD	ERRNO,T33UND,EXPREC		;UNC BIT NOT SET AFTER READ CMD.
	054044	104456					TRAP	C#ERHRD
	054046	001001					.WORD	513
	054050	055052					.WORD	T33UND
	054052	015554					.WORD	EXPREC
7271	054054				170#:	CKLOOP		;LOOP IF SELECTED
	054054	104406					TRAP	C#CLP1
7272	054056	013701	054502		MOV	T33BFR+10,R1		;PICK UP XST1
7273	054062	010102			MOV	R1,R2		;SET UP EXPECTED
7274	054064	052702	000400		BIS	#BIT8,R2		;SET RBP BIT IN EXPECTED
7275	054070	020102			CMP	R1,R2		;DOES EXP = REC'D
7276	054072	001406			BEQ	180#		;BR, IF EQUAL (OK)
7277	054074	005237	002214		INC	FATFLG		;ERROR COUNT
7281	054100				ERRHRD	ERRNO,T33RBP,EXPREC		;READ BUS PARITY ERROR BIT NOT SET
	054100	104456					TRAP	C#ERHRD
	054102	001002					.WORD	514
	054104	054624					.WORD	T33RBP
	054106	015554					.WORD	EXPREC
7282	054110				180#:	CKLOOP		;LOOP IF SELECTED
	054110	104406					TRAP	C#CLP1
7283	054112	017701	127000		MOV	#FREE,R1		;GET DATA READ
7284	054116	013702	054620		MOV	T33CNU,R2		;GET PATTERN
7285	054122	020102			CMP	R1,R2		;ARE THEY EQUAL
7286	054124	001406			BEQ	182#		;BR, IF OK
7287	054126	005237	002214		INC	FATFLG		;ERROR COUNT
7291	054132				ERRHRD	ERRNO,T33DTA,EXPREC		;DATA NOT CORRECT
	054132	104456					TRAP	C#ERHRD
	054134	001003					.WORD	515
	054136	055450					.WORD	T33DTA
	054140	015554					.WORD	EXPREC
7292	054142				182#:	CKLOOP		;LOOP IF SELECTED
	054142	104406					TRAP	C#CLP1

## TEST 5: DATA PARITY TEST

7293	054144	013737	003116	054572		MOV	FREE,T33WB		;STARTING WRITE BUFFER ADDRESS		
7294	054152	012737	140401	054570	195‡:	MOV	#140401,T33PK3		;READ REVERSE DATA RETRY,ACK COMMAND		
7295	054160	012704	054570			MOV	#T33PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
7296	054164	012737	000024	054576		MOV	#20.,T33SZ		;SET UP RECORD SIZE IN PACKET		
7297	054172	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND		
7298	054176	004737	016330			JSR	PC,WAITF		;WAIT FOR SSR TO SET		
7299	054202	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
7300	054206	012702	100210			MOV	#SC!SSR!BIT3,R2		;SET UP EXPECTED		
7301	054212	020102				CMP	R1,R2		;ARE THEY EQUAL		
7302	054214	001406				BEQ	190‡		;BR, IF OK		
7303	054216	005237	002214			INC	FATFLG		;ERROR COUNT		
7307	054222					ERRHRD	ERRNO,T33WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA		
	054222	104456							TRAP	C#ERHRD	
	054224	001004							.WORD	516	
	054226	055367							.WORD	T33WDC	
	054230	012126							.WORD	PKTSSR	
7308	054232				190‡:	CKLOOP			;LOOP IF SELECTED		
	054232	104406							TRAP	C#CLP1	
7309	054234	C13701	054502			MOV	T33BFR+10,R1		;PICK UP XST1		
7310	054240	010102				MOV	R1,R2		;SET UP EXPECTED		
7311	054242	052702	000002			BIS	#BIT1,R2		;SET UNC BIT IN EXPECTED		
7312	054246	020102				CMP	R1,R2		;DOES EXP = REC'D		
7313	054250	001406				BEQ	200‡		;BR, IF EQUAL (OK)		
7314	054252	005237	002214			INC	FATFLG		;ERROR COUNT		
7318	054256					ERRHRD	ERRNO,T33UND,EXPREC		;TAPE NOT AT BOT AFTER REWIND		
	054256	104456							TRAP	C#ERHRD	
	054260	001005							.WORD	517	
	054262	055052							.WORD	T33UND	
	054264	015554							.WORD	EXPREC	
7319	054266				200‡:	CKLOOP			;LOOP IF SELECTED		
	054266	104406							TRAP	C#CLP1	
7320	054270	013701	054502			MOV	T33BFR+10,R1		;PICK UP XST0		
7321	054274	010102				MOV	R1,R2		;SET UP EXPECTED		
7322	054276	052702	000400			BIS	#BIT8,R2		;SET RBP BIT IN EXPECTED		
7323	054302	020102				CMP	R1,R2		;DOES EXP = REC'D		
7324	054304	001406				BEQ	210‡		;BR, IF EQUAL (OK)		
7325	054306	005237	002214			INC	FATFLG		;ERROR COUNT		
7329	054312					ERRHRD	ERRNO,T33RBP,EXPREC		;READ BUS PARITY ERROR BIT NOT SET		
	054312	104456							TRAP	C#ERHRD	
	054314	001006							.WORD	518	
	054316	054624							.WORD	T33RBP	
	054320	015554							.WORD	EXPREC	
7330	054322				210‡:	CKLOOP			;LOOP IF SELECTED		
	054322	104406							TRAP	C#CLP1	
7331	054324	017701	126566			MOV	#FREE,R1		;GET DATA READ		
7332	054330	013702	054620			MOV	T33CNU,R2		;GET PATTERN		
7333	054334	020102				CMP	R1,R2		;ARE THEY EQUAL		
7334	054336	001406				BEQ	215‡		;BR, IF OK		
7335	054340	005237	002214			INC	FATFLG		;ERROR COUNT		
7339	054344					ERRHRD	ERRNO,T33DTA,EXPREC		;DATA NOT CORRECT		
	054344	104456							TRAP	C#ERHRD	
	054346	001007							.WORD	519	
	054350	055450							.WORD	T33DTA	
	054352	015554							.WORD	EXPREC	
7340	054354				215‡:	CKLOOP			;LOOP IF SELECTED		
	054354	104406							TRAP	C#CLP1	
7341	054356	010302				MOV	R3,R2		;SAVE R3 FOR A MOMENT		





## TEST 5: DATA PARITY TEST

```

7402
7403
7404
7405 054600
7406 054600 010
7407 054601 200
7408 054602 000000
7409 054604 000000
7410
7411
7412
7413
7414
7415 054606 100205
7416 054610 100605
7417 054612 102205
7418 054614 177777
7419
7420
7421 054616 000000
7422 054620 000000
7423 054622 000000
7424
7425
7426
7427
7428 054624 122 145 141
7429 054702 124 123 123
7430 054762 125 116 103
7431 055052 125 116 103
7432 055141 127 122 111
7433 055225 124 141 160
7434 055320 122 145 167
7435 055367 124 123 123
7436 055450 104 141 164
7437 055545 104 141 164
7438
7439
7440
7441
7442
7443
7444
7445
7446 055562
7447 055562
7448 055566 012701 054450
7449 055572 012721 100004
7450 055576 012721 054460
7451 055602 005021
7452 055604 012721 000012
7453 055610 012721 054472
7454 055614 005021
7455 055616 012721 000024
7456 055622 005021
7457 055624 012711 000000
7458 055630 012702 000030

```

```

:
:
:
T33BF2:
T33BS0: .BYTE 10 ;BSELO AREA
T33BS1: .BYTE 200 ;BSEL1 AREA
T33S2: .WORD 0 ;SEL 2 AREA
T33S3: .WORD 0 ;DATA AREA
:
:
.EVEN
;TAPE MOTION PACKET COMMAND VALUES
T33RN: .WORD 100205 ;REREAD DATA (NEXT)
T33WDR: .WORD 100605 ;REREAD DATA RETRY
T33CON: .WORD 102205 ;WRITE CONTINUOUS
        .WORD 177777 ;END OF DATA
:
T33CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T33CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T33DLY: .WORD 0 ;DELAY COUNTER
:
;LOCAL TEXT MESSAGES FOR TEST
;-
T33RBP: .ASCIZ 'Read Bus Parity Bit Not Set (XST1), Should Be'
T33WPW: .ASCIZ 'TSSR Incorrect After Wrong Parity Write Command'
T33UNC: .ASCIZ 'UNC Bit (XST1) Not Set After Wrong Parity WRITE Command'
T33UND: .ASCIZ 'UNC Bit (XST1) Not Set After Wrong Parity READ Command'
T33SSR: .ASCIZ 'WRITE MISCELLANEOUS CONT/READ COMMAND Not Accepted'
T33BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
T33RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
T33WDC: .ASCIZ 'TSSR Not Correct After READ Wrong Parity Command'
T33DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
TST33ID: .ASCIZ 'Data Parity'
.EVEN
:
;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
;WRITE SUBSYSTEM MEMORY COMMAND
:
;-
T33REST:
SAVREG ;SAVE THE REGISTERS
MOV #T33PACKET,R1 ;START OF THE PACKET
MOV #100004,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK.
MOV #T33DATA,(R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
CLR (R1)+ ;EXTENDED ADDRESS
MOV #10.,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
MOV #T33FR,(R1)+ ;ADDRESS OF MESSAGE BUFFER
CLR (R1)+
MOV #20.,(R1)+ ;LENGTH OF MESSAGE BUFFER
CLR (R1)+
MOV #0,(R1) ;SELECT DRIVE ZERO
MOV #24.,R2 ;NUMBER OF LOCATIONS TO BE CLEARED

```

## TEST 5: DATA PARITY TEST

```

7459 055634 012762 177777 054472 64#:  MOV    #177777,T33BFR(R2)    ;ALL ONES TO MESSAGE BUFFER
7460 055642 005742                    TST    -(R2)                ;NEXT LOCATION
7461 055644 022702 000000             CMP    #0,R2                ;AT END OF LOOP YET
7462 055650 001371                    BNE    64#                  ;KEEP GOING UNTIL DONE
7463 055652 000207                    RTS    PC                    ;RETURN
7464
7465 055654                    T33RT2:
7466 055654                    SAVREG                       ;SAVE THE REGISTERS
7467 055660 012701 054560             MOV    #T33PK2,R1           ;START OF THE PACKET
7468 055664 012721 100006             MOV    #100006,(R1)+        ;WRITE SUBSYSTEM MEM. WITH ACK.
7469 055670 012721 054600             MOV    #T33BF2,(R1)+       ;ADDRESS OF DATA BLOCK
7470 055674 005021                    CLR    (R1)+                ;EXTENDED ADDRESS
7471 055676 012721 000006             MOV    #6,(R1)+            ;SIZE OF DATA BLOCK IN BYTES
7472 055702 005021                    CLR    (R1)+                ;POINT TO DATA SEL AREA
7473 055704 012701 054600             MOV    #T33BF2,R1
7474 055710 005021                    CLR    (R1)+
7475 055712 005011                    CLR    (R1)
7476 055714 000207                    RTS    PC                    ;RETURN
7477 055716                    T33RT3:
7478 055716                    SAVREG                       ;SAVE REGISTERS
7479 055722 012701 054570             MOV    #T33PK3,R1           ;SET UP POINTER ADDRESS
7480 055726 005021                    CLR    (R1)+                ;COMMAND SPACE
7481 055730 005021                    CLR    (R1)+                ;ADDRESS OF DATA BLOCK
7482 055732 005021                    CLR    (R1)+                ;EXTENDED ADDRESS
7483 055734 005011                    CLR    (R1)                 ;SIZE OF DATA TRANSFER BLOCK
7484 055736 000207                    RTS    PC                    ;RETURN
7485 055740                    ENDTST
7486 055740 104401                    L10057: TRAP    C#ETST
7487
7488
7489
7490
7491
7492
7493
7494
7495
7496
7497
7498 055742                    .SBTTL TEST 6: OPERATIONS AT EOT
7499 055742                    ;*
7500
7501
7502
7503
7504 055750 012737 006354 002172       ; THIS TEST VERIFIES PROPER OPERATION OF THE WRITE DATA RETRY
7505 055754 004737 016570             ; COMMAND (SPACE REVERSE, ERASE, WRITE DATA)
7506 055760 012737 000005 002210       ;
7507 055766 005037 060572             ; THE TEST CONSISTS OF THE FOLLOWING 1 SUBTEST
7508
7509
7510
7511
7512
7513
7514
7515
7516
7517
7518
7519
7520
7521
7522
7523
7524
7525
7526
7527
7528
7529
7530
7531
7532
7533
7534
7535
7536
7537
7538
7539
7540
7541
7542
7543
7544
7545
7546
7547
7548
7549
7550
7551
7552
7553
7554
7555
7556
7557
7558
7559
7560
7561
7562
7563
7564
7565
7566
7567
7568
7569
7570
7571
7572
7573
7574
7575
7576
7577
7578
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589
7590
7591
7592
7593
7594
7595
7596
7597
7598
7599
7600
7601
7602
7603
7604
7605
7606
7607
7608
7609
7610
7611
7612
7613
7614
7615
7616
7617
7618
7619
7620
7621
7622
7623
7624
7625
7626
7627
7628
7629
7630
7631
7632
7633
7634
7635
7636
7637
7638
7639
7640
7641
7642
7643
7644
7645
7646
7647
7648
7649
7650
7651
7652
7653
7654
7655
7656
7657
7658
7659
7660
7661
7662
7663
7664
7665
7666
7667
7668
7669
7670
7671
7672
7673
7674
7675
7676
7677
7678
7679
7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7690
7691
7692
7693
7694
7695
7696
7697
7698
7699
7700
7701
7702
7703
7704
7705
7706
7707
7708
7709
7710
7711
7712
7713
7714
7715
7716
7717
7718
7719
7720
7721
7722
7723
7724
7725
7726
7727
7728
7729
7730
7731
7732
7733
7734
7735
7736
7737
7738
7739
7740
7741
7742
7743
7744
7745
7746
7747
7748
7749
7750
7751
7752
7753
7754
7755
7756
7757
7758
7759
7760
7761
7762
7763
7764
7765
7766
7767
7768
7769
7770
7771
7772
7773
7774
7775
7776
7777
7778
7779
7780
7781
7782
7783
7784
7785
7786
7787
7788
7789
7790
7791
7792
7793
7794
7795
7796
7797
7798
7799
7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824
7825
7826
7827
7828
7829
7830
7831
7832
7833
7834
7835
7836
7837
7838
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849
7850
7851
7852
7853
7854
7855
7856
7857
7858
7859
7860
7861
7862
7863
7864
7865
7866
7867
7868
7869
7870
7871
7872
7873
7874
7875
7876
7877
7878
7879
7880
7881
7882
7883
7884
7885
7886
7887
7888
7889
7890
7891
7892
7893
7894
7895
7896
7897
7898
7899
7900
7901
7902
7903
7904
7905
7906
7907
7908
7909
7910
7911
7912
7913
7914
7915
7916
7917
7918
7919
7920
7921
7922
7923
7924
7925
7926
7927
7928
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7950
7951
7952
7953
7954
7955
7956
7957
7958
7959
7960
7961
7962
7963
7964
7965
7966
7967
7968
7969
7970
7971
7972
7973
7974
7975
7976
7977
7978
7979
7980
7981
7982
7983
7984
7985
7986
7987
7988
7989
7990
7991
7992
7993
7994
7995
7996
7997
7998
7999
8000

```

TEST 6: OPERATIONS AT EOT

```

7517 :
7518 :
7519 :
7520 :
7521 :
7522 :
7523 :
7524 :
7525 :
7526 :
7527 :
7528 :
7529 :
7530 :
7531 :
7532 :
7533 :
7534 :
7535 :
7536 :
7537 :
7538 :
7539 :
7540 :
7541 :
7542 :
7543 :
7544 :
7545 :
7546 :
7547 :
7548 :
7549 :
7550 :
7551 :
7552 :
7553 :
7554 :
7555 :
7556 :
7557 :
7558 :
7559 :
7560 :
7561 :
7562 :
7563 :
7564 :
7565 :
7566 :
7567 :
7568 :
7569 :
7570 :
7571 055772 T34LOOP:
7572 :
7573 :
    
```

1. THE TAPE IS REWOUND.
2. WRITE DATA COMMANDS ARE REPEATEDLY ISSUED UNTIL TAPE STATUS ALERT TERMINATION IS SEEN WITH EOT=1. ERRORS OTHER THAN OCCASIONAL CORRECTABLE OR UNCORRECTABLE DATA ERRORS CAUSE A FATAL ERROR REPORT. RECORDS WITH DATA ERRORS ARE RETRIED, SO THE TAPE ENDS UP WITH GOOD DATA.
3. ANOTHER WRITE DATA COMMAND IS ISSUED, AND IT IS CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=1.
4. A WRITE TAPE MARK COMMAND IS ISSUED, AND IT IS CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=1.
5. A SKIP TAPE MARKS REVERSE COMMAND IS ISSUED, AND IT IS CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=1 AND TMK=1.
6. A SPACE RECORDS REVERSE COMMAND, WITH A RECORD COUNT OF 1, IS ISSUED, AND IT IS CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=1 AND TMK=1.
7. A SPACE RECORDS REVERSE COMMAND, WITH A RECORD COUNT OF 1, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
8. A SPACE RECORDS FORWARD COMMAND, WITH A RECORD COUNT OF 1, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
9. A READ REVERSE COMMAND IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
10. A READ FORWARD COMMAND IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
11. A SPACE RECORDS REVERSE COMMAND, WITH A RECORD COUNT OF 3, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=0.
12. A SPACE RECORDS FORWARD COMMAND, WITH A RECORD COUNT OF 3, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
13. A SKIP FILE MARKS REVERSE COMMAND IS ISSUED, WHICH SHOULD SKIP ALL THE WAY TO BOT, AND IT IS CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=0, BOT=1, AND RIB=1.



TEST 6: OPERATIONS AT EOT

7619	056140	103407			BCS	30#			:BR, IF COMMAND ISSUED OK
7620	056142	005237	002214		INC	FATFLG			:ERROR COUNT
7624	056146	010001			MOV	RO,R1			:SAVE CONTENTS OF TSSR
7625	056150				ERRHRD	ERRNO,WRTMSG,SFIMSG			:WRITE CHARACTERISTIC FAILED
	056150	104456							TRAP C#ERHRD
	056152	001132							.WORD 602
	056154	005052							.WORD WRTMSG
	056156	012114							.WORD SFIMSG
7626	056160				30#:	CKLOOP			:LOOP IF SELECTED
	056160	104406							TRAP C#CLP1
7627	056162	004737	011074		JSR	PC,REWIND			:REWIND CALL
7628	056166	103411			BCS	35#			:BR, IF TSSR IS OK (GOOD)
7629	056170	016501	000002		MOV	TSSR(R5),R1			:GET TSSR
7630	056174	010004			MOV	RO,R4			:SET UP PACKET
7631	056176	005237	002214		INC	FATFLG			:ERROR COUNT
7635	056202				ERRHRD	ERRNO,T34RWN,PKTSSR			:TSSR IS INCORRECT AFTER REWIND
	056202	104456							TRAP C#ERHRD
	056204	001133							.WORD 603
	056206	C62237							.WORD T34RWN
	056210	012126							.WORD PKTSSR
7636	056212				35#:	CKLOOP			:LOOP IF SELECTED
	056212	104406							TRAP C#CLP1
7637	056214	012737	140005	060560	MOV	#140005,T34PK3			:WRITE DATA, ACK, CVC=1
7638	056222	012703	176750		MOV	#65000.,R3			:SET MAX NUMBER OF WRITES
7639	056226	013737	003116	060562	MOV	FREE,T34WB			:SET UP WRITE BUFFER ADDRESS
7640	056234	012737	006654	060566	MOV	#3500.,T34SZ			:SET UP BUFFER SIZE (4K BYTES)
7641	056242	012704	060560		MOV	#T34PK3,R4			:R4 = POINTER TO PACKET
7642	056246	010465	000000		40#:	MOV R4,TSDB(R5)			:ISSUE COMMAND
7643	056252	004737	016330		JSR	PC,WAITF			:WAIT FOR SSR TO SET
7644	056256	016501	000002		MOV	TSSR(R5),R1			:GET TSSR CONTENTS
7645	056262	012702	000200		MOV	#SSR,R2			:SET UP EXPECTED
7646	056266	020102			CMP	R1,R2			:ARE THEY EQUAL
7647	056270	001010			BNE	50#			:BR, IT MIGHT BE END OF TAPE
7648	056272	005303			DEC	R3			:DEC RECORD COUNTER
7649	056274	001364			BNE	40#			:BR, IF MORE TO GO
7650	056276	005237	002214		INC	FATFLG			:ERROR COUNT
7654	056302				ERRDF	ERRNO,T34ET,PKTSSR			:EOT NOT FOUND (USE SHORTER TAPE?)
	056302	104455							TRAP C#ERDF
	056304	001134							.WORD 604
	056306	062016							.WORD T34ET
	056310	012126							.WORD PKTSSR
7655	056312	032701	000004		50#:	BIT #BIT2,R1			:CHECK FOR TAPE STATUS ALERT
7656	056316	001001			BNE	60#			:BR, IF SET
7657	056320	000752			BR	40#			:KEEP GOING
7658	056322	013701	060470		60#:	MOV T34BFR+6,R1			:PICK UP XSTO
7659	056326	010102			MOV	R1,R2			:SET UP EXPECTED
7660	056330	052702	000001		BIS	#BIT0,R2			:SET THE EOT BIT ON IN EXPECTED
7661	056334	020102			CMP	R1,R2			:WAS THE BIT ON
7662	056336	001402			BEQ	80#			:BR, IF EOT WAS FOUND
7663	056340	000137	056246		JMP	40#			:KEEP LOOKING
7664	056344				80#:	CKLOOP			:LOOP IF SELECTED
	056344	104406							TRAP C#CLP1
7665	056346	012737	140005	060560	MOV	#140005,T34PK3			:WRITE DATA, ACK, CVC=1
7666	056354	013737	003116	060562	MOV	FREE,T34WB			:SET UP WRITE BUFFER ADDRESS
7667	056362	012737	006654	060566	MOV	#3500.,T34SZ			:SET UP BUFFER SIZE (4K BYTES)
7668	056370	012704	060560		MOV	#T34PK3,R4			:R4 = POINTER TO PACKET
7669	056374	010465	000000		MOV	R4,TSDB(R5)			:ISSUE COMMAND

## TEST 6: OPERATIONS AT EOT

```

7670 056400 004737 016330      JSR      PC, WAITF          ;WAIT FOR SSR TO SET
7671 056404 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
7672 056410 012702 100204      MOV      #SC!SSR!BIT2,R2  ;SET UP EXPECTED
7673 056414 020102              CMP      R1,R2             ;ARE THEY EQUAL
7674 056416 001406              BEQ      90$               ;BR. IF THEY ARE OK
7675 056420 005237 002214      INC      FATFLG           ;ERROR COUNT
7679 056424              ERRHRD  ERRNO,T34ET2,PKTSSR ;WRITE TAPE AT EOT FAILED TO SET TSA
                                TRAP      C#ERHRD
                                .WORD    605
                                .WORD    T34ET2
                                .WORD    PKTSSR
                                056424 104456
                                056426 001135
                                056430 061267
                                056432 012126
7680 056434              90$:  CKLOOP             ;LOOP IF SELECTED
                                TRAP      C#CLP1
                                056434 104406
7681 056436 013701 060470      MOV      T34BFR+6,R1      ;PICK UP XSTO
7682 056442 010102              MOV      R1,R2            ;SET UP EXPECTED
7683 056444 052702 000001      BIS      #BIT0,R2         ;SET THE EOT BIT ON IN EXPECTED
7684 056450 020102              CMP      R1,R2            ;WAS THE BIT ON
7685 056452 001406              BEQ      100$             ;BR. IF EOT WAS FOUND
7686 056454 005237 002214      INC      FATFLG           ;ERROR COUNT
7690 056460              ERRHRD  ERRNO,T34ETN,EXPREC ;EOT BIT (XSTO) NOT SET
                                TRAP      C#ERHRD
                                .WORD    606
                                .WORD    T34ETN
                                .WORD    EXPREC
                                056460 104456
                                056462 001136
                                056464 061351
                                056466 015554
7691 056470              100$: CKLOOP           ;LOOP IF SELECTED
                                TRAP      C#CLP1
                                056470 104406
7692 056472 012737 140011 060560  MOV      #140011,T34PK3   ;WRITE TAPE MARK, ACK, CVC-1 COMMAND
7693 056500 012704 060560      MOV      #T34PK3,R4       ;R4 = POINTER TO PACKET
7694 056504 010465 000000      MOV      R4,TSD(BR5)      ;ISSUE COMMAND
7695 056510 004737 016330      JSR      PC, WAITF        ;WAIT FOR SSR TO SET
7696 056514 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
7697 056520 012702 100204      MOV      #SC!SSR!BIT2,R2 ;SET UP EXPECTED
7698 056524 020102              CMP      R1,R2            ;ARE THEY EQUAL
7699 056526 001406              BEQ      110$             ;BR. IF STATUS IS GOOD (OK)
7700 056530 005237 002214      INC      FATFLG           ;ERROR COUNT
7704 056534              ERRHRD  ERRNO,T34WTH,PKTSSR ;EOT NOT FOUND (USE SHORTER TAPE?)
                                TRAP      C#ERHRD
                                .WORD    607
                                .WORD    T34WTH
                                .WORD    PKTSSR
                                056534 104456
                                056536 001137
                                056540 061200
                                056542 012126
7705 056544              110$: CKLOOP           ;LOOP IF SELECTED
                                TRAP      C#CLP1
                                056544 104406
7706 056546 013701 060470      MOV      T34BFR+6,R1     ;PICK UP XSTO
7707 056552 010102              MOV      R1,R2            ;SET UP EXPECTED
7708 056554 052702 000001      BIS      #BIT0,R2         ;SET THE EOT BIT ON IN EXPECTED
7709 056560 020102              CMP      R1,R2            ;WAS THE BIT ON
7710 056562 001406              BEQ      120$             ;BR. IF EOT WAS FOUND
7711 056564 005237 002214      INC      FATFLG           ;ERROR COUNT
7715 056570              ERRHRD  ERRNO,T34ETO,EXPREC ;EOT BIT (XSTO) NOT SET
                                TRAP      C#ERHRD
                                .WORD    608
                                .WORD    T34ETO
                                .WORD    EXPREC
                                056570 104456
                                056572 001140
                                056574 060702
                                056576 015554
7716 056600              120$: CKLOOP           ;LOOP IF SELECTED
                                TRAP      C#CLP1
                                056600 104406
7717 056602 012737 141410 060560  MOV      #141410,T34PK3   ;SKIP TAPE MARK REVERSE ACK,CVC-1 COMMAND
7718 056610 012737 000001 060562  MOV      #1,T34WB         ;SET NUMBER (1) OF TMS TO SKIP

```

TEST 6: OPERATIONS AT EOT

```

7719 056616 012704 060560      MOV      #T34PK3,R4      ;R4 = POINTER TO PACKET
7720 056622 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
7721 056626 004737 016330      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
7722 056632 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
7723 056636 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED
7724 056642 020102                CMP      R1,R2          ;ARE THEY EQUAL
7725 056644 001406                BEQ      130$           ;BR, IF STATUS IS GOOD (OK)
7726 056646 005237 002214      INC      FATFLG         ;ERROR COUNT
7730 056652                ERRHRD   ERRNO,T34STM,PKTSSR ;SKIP TAPE MARK REV. DIDN'T SET TSA
                                TRAP      C#ERHRD
                                .WORD    609
                                .WORD    T34STM
                                .WORD    PKTSSR
7731 056662                130$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C#CLP1
7732 056664 013701 060470      MOV      T34BFR+6,R1   ;PICK UP XSTO
7733 056670 010102                MOV      R1,R2          ;SET UP EXPECTED
7734 056672 052702 000001      BIS      #BIT0,R2       ;SET THE EOT BIT ON IN EXPECTED
7735 056676 020102                CMP      R1,R2          ;WAS THE BIT ON
7736 056700 001406                BEQ      140$           ;BR, IF EOT WAS FOUND
7737 056702 005237 002214      INC      FATFLG         ;ERROR COUNT
7741 056706                ERRHRD   ERRNO,T34ETN,EXPREC ;EOT BIT (XSTO) NOT SET
                                TRAP      C#ERHRD
                                .WORD    610
                                .WORD    T34ETN
                                .WORD    EXPREC
7742 056716                140$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C#CLP1
7743 056720 013701 060470      MOV      T34BFR+6,R1   ;PICK UP XSTO
7744 056724 010102                MOV      R1,R2          ;SET UP EXPECTED
7745 056726 052702 100000      BIS      #BIT15,R2      ;SET THE TMK BIT ON IN EXPECTED
7746 056732 020102                CMP      R1,R2          ;WAS THE BIT ON
7747 056734 001406                BEQ      150$           ;BR, IF TMK WAS FOUND
7748 056736 005237 002214      INC      FATFLG         ;ERROR COUNT
7752 056742                ERRHRD   ERRNO,T34TMK,EXPREC ;EOT BIT (XSTO) NOT SET
                                TRAP      C#ERHRD
                                .WORD    611
                                .WORD    T34TMK
                                .WORD    EXPREC
7753 056752                150$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C#CLP1
7754 056754 012737 140410 060560      MOV      #140410,T34PK3 ;SPACE RECORDS REVERSE, ACK, CVC-1 CMD
7755 056762 012737 000001 060562      MOV      #1,T34WB       ;SPACE ONE RECORD REVERSE
7756 056770 012704 060560      MOV      #T34PK3,R4    ;R4 = POINTER TO PACKET
7757 056774 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
7758 057000 004737 016330      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
7759 057004 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
7760 057010 012702 100204      MOV      #SC!SSR!BIT2,R2 ;SET UP EXPECTED
7761 057014 020102                CMP      R1,R2          ;ARE THEY EQUAL
7762 057016 001006                BNE      160$           ;BR, IT MIGHT BE END OF TAPE
7763 057020 005237 002214      INC      FATFLG         ;ERROR COUNT
7767 057024                ERRHRD   ERRNO,T34POS,PKTSSR ;EOT NOT FOUND (USE SHORTER TAPE?)
                                TRAP      C#ERHRD
                                .WORD    612
                                .WORD    T34POS
                                .WORD    PKTSSR
7768 057034                160$:  CKLOOP          ;LOOP IF SELECTED

```



TEST 6: OPERATIONS AT EOT

7769	057034	104406					TRAP	C#CLP1
	057036	013701	060470	MOV	T34BFR+6,R1			
	057042	010102		MOV	R1,R2			
	057044	052702	000001	BIS	#BIT0,R2			
	057050	020102		CMP	R1,R2			
	057052	001406		BEQ	163#			
	057054	005237	002214	INC	FATFLG			
	057060			ERRHRD	ERRNO,T34ETN,EXPREC			
	057060	104456					TRAP	C#ERHRD
	057062	001145					.WORD	613
	057064	061351					.WORD	T34ETN
	057066	015554					.WORD	EXPREC
7779	057070			163#:	CKLOOP			
	057070	104406					TRAP	C#CLP1
	057072	013701	060470	MOV	T34BFR+6,R1			
	057076	010102		MOV	R1,R2			
	057100	042702	100000	BIC	#BIT15,R2			
	057104	020102		CMP	R1,R2			
	057106	001406		BEQ	165#			
	057110	005237	002214	INC	FATFLG			
	057114			ERRHRD	ERRNO,T34TMK,EXPREC			
	057114	104456					TRAP	C#ERHRD
	057116	001146					.WORD	614
	057120	061663					.WORD	T34TMK
	057122	015554					.WORD	EXPREC
7790	057124			165#:	CKLOOP			
	057124	104406					TRAP	C#CLP1
	057126	012737	140410	MOV	#140410,T34PK3			
	057134	012737	000001	MOV	#1,T34WB			
	057142	012704	060560	MOV	#T34PK3,R4			
	057146	010465	000000	MOV	R4,TSDB(R5)			
	057152	004737	016330	JSR	PC,WAITF			
	057156	016501	000002	MOV	TSSR(R5),R1			
	057162	012702	000200	MOV	#SSR,R2			
	057166	020102		CMP	R1,R2			
	057170	001406		BEQ	167#			
	057172	005237	002214	INC	FATFLG			
	057176			ERRHRD	ERRNO,T34POS,PKTSSR			
	057176	104456					TRAP	C#ERHRD
	057200	001147					.WORD	615
	057202	060614					.WORD	T34POS
	057204	012126					.WORD	PKTSSR
7805	057206			167#:	CKLOOP			
	057206	104406					TRAP	C#CLP1
	057210	013701	060470	MOV	T34BFR+6,R1			
	057214	010102		MOV	R1,R2			
	057216	042702	000001	BIC	#BIT0,R2			
	057222	020102		CMP	R1,R2			
	057224	001400		BEQ	170#			
	057226			170#:	CKLOOP			
	057226	104406					TRAP	C#CLP1
	057230	012737	140010	MOV	#140010,T34PK3			
	057236	012737	000002	MOV	#2,T34WB			
	057244	012704	060560	MOV	#T34PK3,R4			
	057250	010465	000000	MOV	R4,TSDB(R5)			
	057254	004737	016330	JSR	PC,WAITF			
	057260	016501	000002	MOV	TSSR(R5),R1			

TEST 6: OPERATIONS AT EOT

```

7818 057264 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
7819 057270 020102      CMP      R1,R2      ;ARE THEY EQUAL
7820 057272 001406      BEQ      190$      ;BR, IT MIGHT BE END OF TAPE
7821 057274 005237 002214      INC      FATFLG      ;ERROR COUNT
7825 057300      ERRHRD  ERRNO,T34POS,PKTSSR ;EOT NOT FOUND (USE SHORTER TAPE?)
      057300 104456      TRAP    C$ERHRD
      057302 001150      .WORD  616
      057304 060614      .WORD  T34POS
      057306 012126      .WORD  PKTSSR
7826 057310      190$:  CKLOOP      ;LOOP IF SELECTED      TRAP    C$CLP1
      057310 104406      ;PICK UP XSTO
7827 057312 013701 060470      MOV      T34BFR+6,R1 ;SET UP EXPECTED
7828 057316 010102      MOV      R1,R2      ;SET THE EOT BIT ON IN EXPECTED
7829 057320 052702 000001      BIS      #BIT0,R2    ;WAS THE BIT ON
7830 057324 020102      CMP      R1,R2      ;BR, IF EOT WAS FOUND
7831 057326 001406      BEQ      200$      ;ERROR COUNT
7832 057330 005237 002214      INC      FATFLG      ;EOT BIT (XSTO) NOT SET
7836 057334      ERRHRD  ERRNO,T34ETS,EXPREC ;
      057334 104456      TRAP    C$ERHRD
      057336 001151      .WORD  617
      057340 061430      .WORD  T34ETS
      057342 015554      .WORD  EXPREC
7837 057344      200$:  CKLOOP      ;LOOP IF SELECTED      TRAP    C$CLP1
      057344 104406      ;READ REVERSE, ACK, CVC=1
7838 057346 012737 140401 060560      MOV      #140401,T34PK3 ;SET UP WRITE BUFFER ADDRESS
7839 057354 013737 003116 060562      MOV      FREE,T34RB   ;R4 = POINTER TO PACKET
7840 057362 012704 060560      MOV      #T34PK3,R4  ;ISSUE COMMAND
7841 057366 010465 000000      MOV      R4,TSDB(R5) ;WAIT FOR SSR TO SET
7842 057372 004737 016330      JSR      PC,WAITF    ;GET TSSR CONTENTS
7843 057376 016501 000002      MOV      TSSR(R5),R1 ;SET UP EXPECTED
7844 057402 012702 000200      MOV      #SSR,R2    ;ARE THEY EQUAL
7845 057406 020102      CMP      R1,R2      ;BR, ONLY SSR IS SET
7846 057410 001406      BEQ      205$      ;ERROR COUNT
7847 057412 005237 002214      INC      FATFLG      ;EOT NOT FOUND (USE SHORTER TAPE?)
7851 057416      ERRHRD  ERRNO,T34RRE,PKTSSR ;
      057416 104456      TRAP    C$ERHRD
      057420 001152      .WORD  618
      057422 060766      .WORD  T34RRE
      057424 012126      .WORD  PKTSSR
7852 057426      205$:  CKLOOP      ;LOOP IF SELECTED      TRAP    C$CLP1
      057426 104406      ;READ REVERSE, ACK, CVC=1
7853 057430 012737 140401 060560      MOV      #140401,T34PK3 ;SET UP WRITE BUFFER ADDRESS
7854 057436 013737 003116 060562      MOV      FREE,T34RB   ;R4 = POINTER TO PACKET
7855 057444 012704 060560      MOV      #T34PK3,R4  ;ISSUE COMMAND
7856 057450 010465 000000      MOV      R4,TSDB(R5) ;WAIT FOR SSR TO SET
7857 057454 004737 016330      JSR      PC,WAITF    ;GET TSSR CONTENTS
7858 057460 016501 000002      MOV      TSSR(R5),R1 ;SET UP EXPECTED
7859 057464 012702 000200      MOV      #SSR,R2    ;ARE THEY EQUAL
7860 057470 020102      CMP      R1,R2      ;BR, IT MIGHT BE END OF TAPE
7861 057472 001406      BEQ      210$      ;ERROR COUNT
7862 057474 005237 002214      INC      FATFLG      ;EOT NOT FOUND (USE SHORTER TAPE?)
7866 057500      ERRHRD  ERRNO,T34RRE,PKTSSR ;
      057500 104456      TRAP    C$ERHRD
      057502 001153      .WORD  619
      057504 060766      .WORD  T34RRE
      057506 012126      .WORD  PKTSSR
7867 057510      210$:  CKLOOP      ;LOOP IF SELECTED

```

## TEST 6: OPERATIONS AT EOT

7868	057510	104406								TRAP	C#CLP1
	057512	012737	140001	060560	MOV	#140001,T34PK3		;READ DATA, ACK, CVC=1			
7869	057520	013737	003116	060562	MOV	FREE,T34RB		;SET UP WRITE BUFFER ADDRESS			
7870	057526	012737	006654	060566	MOV	#3500.,T34SZ		;SET UP BUFFER SIZE (4K BYTES)			
7871	057534	012704	060560		MOV	#T34PK3,R4		;R4 = POINTER TO PACKET			
7872	057540	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND			
7873	057544	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET			
7874	057550	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS			
7875	057554	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED			
7876	057560	020102			CMP	R1,R2		;ARE THEY EQUAL			
7877	057562	001406			BEQ	230#		;BR, IT MIGHT BE END OF TAPE			
7878	057564	005237	002214		INC	FATFLG		;ERROR COUNT			
7882	057570				ERRHRD	ERRNO,T34RRE,PKTSSR		;EOT NOT FOUND (USE SHORTER TAPE?)			
	057570	104456							TRAP	C#ERHRD	
	057572	001154							.WORD	620	
	057574	060766							.WORD	T34RRE	
	057576	012126							.WORD	PKTSSR	
7883	057600				230#:	CKLOOP		;LOOP IF SELECTED			
	057600	104406							TRAP	C#CLP1	
7884	057602	012737	140001	060560	MOV	#140001,T34PK3		;READ DATA, ACK, CVC=1			
7885	057610	013737	003116	060562	MOV	FREE,T34RB		;SET UP WRITE BUFFER ADDRESS			
7886	057616	012737	006654	060566	MOV	#3500.,T34SZ		;SET UP BUFFER SIZE (4K BYTES)			
7887	057624	012704	060560		MOV	#T34PK3,R4		;R4 = POINTER TO PACKET			
7888	057630	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND			
7889	057634	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET			
7890	057640	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS			
7891	057644	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED			
7892	057650	020102			CMP	R1,R2		;ARE THEY EQUAL			
7893	057652	001406			BEQ	235#		;BR, IT MIGHT BE END OF TAPE			
7894	057654	005237	002214		INC	FATFLG		;ERROR COUNT			
7898	057660				ERRHRD	ERRNO,T34RRE,PKTSSR		;EOT NOT FOUND (USE SHORTER TAPE?)			
	057660	104456							TRAP	C#ERHRD	
	057662	001155							.WORD	621	
	057664	060766							.WORD	T34RRE	
	057666	012126							.WORD	PKTSSR	
7899	057670				235#:	CKLOOP		;LOOP IF SELECTED			
	057670	104406							TRAP	C#CLP1	
7900	057672	013701	060470		MOV	T34BFR+6,R1		;PICK UP XSTO			
7901	057676	010102			MOV	R1,R2		;SET UP EXPECTED			
7902	057700	052702	000001		BIS	#BIT0,R2		;SET THE EOT BIT ON IN EXPECTED			
7903	057704	020102			CMP	R1,R2		;WAS THE BIT ON			
7904	057706	001406			BEQ	240#		;BR, IF EOT WAS FOUND			
7905	057710	005237	002214		INC	FATFLG		;ERROR COUNT			
7909	057714				ERRHRD	ERRNO,T34ETZ,EXPREC		;EOT BIT (XSTO) NOT SET			
	057714	104456							TRAP	C#ERHRD	
	057716	001156							.WORD	622	
	057720	061522							.WORD	T34ETZ	
	057722	015554							.WORD	EXPREC	
7910	057724				240#:	CKLOOP		;LOOP IF SELECTED			
	057724	104406							TRAP	C#CLP1	
7911	057726	012737	140410	060560	MOV	#140410,T34PK3		;SPACE RECORDS REVERSE, ACK, CVC=1 CMD.			
7912	057734	012737	000005	060562	MOV	#5,T34RB		;NUMBER OF RECORDS TO SPACE			
7913	057742	012704	060560		MOV	#T34PK3,R4		;R4 = POINTER TO PACKET			
7914	057746	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND			
7915	057752	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET			
7916	057756	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS			
7917	057762	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED			

TEST 6: OPERATIONS AT EOT

7918	057766	020102			CMP	R1,R2		;ARE THEY EQUAL		
7919	057770	001406			BEQ	250¢		;BR, IT MIGHT BE END OF TAPE		
7920	057772	005237	002214		INC	FATFLG		;ERROR COUNT		
7924	057776				ERRHRD	ERRNO,T34POS,PKTSSR		;POSITION COMMAND DIDN'T WORK		
	057776	104456							TRAP	C#ERHRD
	060000	001157							.WORD	623
	060002	060614							.WORD	T34POS
	060004	012126							.WORD	PKTSSR
7925	060006			250¢:	CKLOOP			;LOOP IF SELECTED		
	060006	104406							TRAP	C#CLP1
7926	060010	013701	060470		MOV	T34BFR+6,R1		;PICK UP XSTO		
7927	060014	010102			MOV	R1,R2		;SET UP EXPECTED		
7928	060016	042702	000001		BIC	#8BIT0,R2		;CLEAR THE EOT BIT ON IN EXPECTED		
7929	060022	020102			CMP	R1,R2		;WAS THE BIT ON		
7930	060024	001406			BEQ	260¢		;BR, IF EOT WAS FOUND		
7931	060026	005237	002214		INC	FATFLG		;ERROR COUNT		
7935	060032				ERRHRD	ERRNO,T34ETC,EXPREC		;EOT BIT (XSTO) NOT CLEAR		
	060032	104456							TRAP	C#ERHRD
	060034	C01160							.WORD	624
	060036	061057							.WORD	T34ETC
	060040	015554							.WORD	EXPREC
7936	060042			260¢:	CKLOOP			;LOOP IF SELECTED		
	060042	104406							TRAP	C#CLP1
7937	060044	012737	140010	060560	MOV	#140010,T34PK3		;SPACE RECORDS FORWARD, ACK, CVC=1 CMD.		
7938	060052	012737	000005	060562	MOV	#5,T34RB		;NUMBER OF RECORDS TO SPACE		
7939	060060	012704	060560		MOV	#T34PK3,R4		;R4 = POINTER TO PACKET		
7940	060064	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND		
7941	060070	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET		
7942	060074	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
7943	060100	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED		
7944	060104	020102			CMP	R1,R2		;ARE THEY EQUAL		
7945	060106	001406			BEQ	270¢		;BR, IT MIGHT BE END OF TAPE		
7946	060110	005237	002214		INC	FATFLG		;ERROR COUNT		
7950	060114				ERRHRD	ERRNO,T34ET,PKTSSR		;TSSR NOT CORRECT		
	060114	104456							TRAP	C#ERHRD
	060116	001161							.WORD	625
	060120	062016							.WORD	T34ET
	060122	012126							.WORD	PKTSSR
7951	060124			270¢:	CKLOOP			;LOOP IF SELECTED		
	060124	104406							TRAP	C#CLP1
7952	060126	013701	060470		MOV	T34BFR+6,R1		;PICK UP XSTO		
7953	060132	010102			MOV	R1,R2		;SET UP EXPECTED		
7954	060134	052702	000001		BIS	#8BIT0,R2		;SET THE EOT BIT ON IN EXPECTED		
7955	060140	020102			CMP	R1,R2		;WAS THE BIT ON		
7956	060142	001400			BEQ	280¢		;BR, IF EOT WAS FOUND		
7957	060144			280¢:	CKLOOP			;LOOP IF SELECTED		
	060144	104406							TRAP	C#CLP1
7958	060146	012737	141410	060560	MOV	#141410,T34PK3		;SKIP FILE MARKS REVERSE,ACK,CVC=1 COMMAND		
7959	060154	012737	000003	060562	MOV	#3,T34RB		;NUMBER OF FILE MARKS		
7960	060162	012704	060560		MOV	#T34PK3,R4		;R4 = POINTER TO PACKET		
7961	060166	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND		
7962	060172	012737	176750	060574	MOV	#65000,T34DLY		;SET UP DELAY COUNTER		
7963	060200	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET		
7964	060204	016501	000002	285¢:	MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
7965	060210	032701	000200		BIT	#SSR,R1		;CHECK FOR SSR SET		
7966	060214	001017			BNE	286¢		;BR, WHEN SSR IS SET		
7967	060216				DELAY	250		;WAIT ABOUT .25 SECONDS		

TEST 6: OPERATIONS AT EOT

060216	012727	000250				MOV	#250,(PC)+
060222	000000					.WORD	0
060224	013727	002116				MOV	L#DLY,(PC)+
060230	000000					.WORD	0
060232	005367	177772				DEC	-6(PC)
060236	001375					BNE	.-4
060240	005367	177756				DEC	-22(PC)
060244	001367					BNE	.-20
7968	060246	005337	060574				
7969	060252	001352					
7970	060254	012702	000200	286#:	DEC	T34DLY	
7971	060260	020102			BNE	285#	
7972	060262	001007			MOV	#SSR,R2	
7973	060264	005303			MOV	R1,R2	
7974	060266	005237	002214		CMP	R1,R2	
7978	060272				BNE	290#	
	060272	104456			DEC	R3	
	060274	001162			INC	FATFLG	
	060276	C62016			ERRHRD	ERRNO,T34ET,PKTSSR	
	060300	012126					
7979	060302	032701	000004				
7980	060306	013701	060470	290#:	BIT	#BIT2,R1	
7981	060312	010102			MOV	T34BFR+6,R1	
7982	060314	042702	000001		MOV	R1,R2	
7983	060320	020102			BIC	#BIT0,R2	
7984	060322	001406			CMP	R1,R2	
7985	060324	005237	002214		BEQ	300#	
7989	060330				INC	FATFLG	
	060330	104456			ERRHRD	ERRNO,T34ETC,EXPREC	
	060332	001163					
	060334	061057					
	060336	015554					
7990	060340			300#:	CKLOOP		
	060340	104406					
7991	060342	013701	060470		MOV	T34BFR+6,R1	
7992	060346	010102			MOV	R1,R2	
7993	060350	052702	000002		BIS	#BIT1,R2	
7994	060354	020102			CMP	R1,R2	
7995	060356	001406			BEQ	320#	
7996	060360	005237	002214		INC	FATFLG	
8000	060364				ERRHRD	ERRNO,T34BOT,EXPREC	
	060364	104456					
	060366	001164					
	060370	061134					
	060372	015554					
8001	060374			320#:	CKLOOP		
	060374	104406					
8002	060376			600#:	ENDSUB		
8003	060376						
	060376	104403					
8004	060400	023727	002214	000017	CMP	FATFLG,#15.	
8005	060406	103402			BLO	999#	
8006	060410	004737	017262		JSR	PC,CKDROP	
8007	060414			999#:			
8008	060414	004737	016536		JSR	PC,TSTLOOP	
8009	060420	103002			BCC	163#	

## TEST 6: OPERATIONS AT EOT

```

8010 060422 000137 055772
8011 060426
      060426 104432
      060430 002662
8012
8013
8014
8016 060440 060440
8018 060440
8019 060440 100004
8020 060442 060450
8021 060444 000000
8022 060446 000010
8023 060450
8024 060450 060462
8025 060452 000000
8026 060454 000012
8027 060456 000000
8028 060460 C00000
8029 060462
8030
8031
8032
8034 060550 060550
8036 060550
8037 060550 100006
8038 060552 060576
8039 060554 000000
8040 060556 000006
8041
8045 060560
8046 060560 100005
8047 060562
8048 060562 000000
8049 060564 000000
8050 060566 000000
8051
8052
8053 060570 000000
8054 060572 000000
8055 060574 000000
8056
8057
8058 060576
8059 060576 010
8060 060577 200
8061 060600 000000
8062 060602 000000
8063
8064
8065
8066
8067
8068 060604 100005
8069 060606 100405
8070 060610 102005
8071 060612 177777

1634: JMP T34LOOP ;EXECUTE AGAIN
      EXIT TST ;ALL DONE THIS TEST
                                TRAP CEXIT
                                .WORD L10061-.

;
;LOCAL STORAGE FOR THIS TEST
;
;
T34PACKET: .=<.+10>&177770 ;COMMAND PACKET FOR TEST
           .WORD 100004 ;WRITE CHARACTERISTICS COMMAND, WITH ACK
           .WORD T34DATA ;ADDRESS OF CHARACTERISTICS BLOCK
           .WORD 0
           .WORD 8. ;STARTING VALUE OF BLOCK SIZE
T34DATA: .WORD T34BFR ;CHARACTERISTICS DATA BLOCK
         .WORD 0 ;ADDRESS OF MESSAGE BUFFER
         .WORD 10. ;LENGTH OF MESSAGE BUFFER
         .WORD 0
T34DSW: .WORD 0 ;SELECT DRIVE 0
T34BFR: .BLKW 25. ;MESSAGE BUFFER
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
T34PK2: .=<.+10>&177770
        .WORD 100006 ;WRITE SUB SYS MEM COMMAND, AND ACK
        .WORD T34BF2 ;ADDRESS OF SELECT BLOCK DATA
        .WORD 0
        .WORD 6. ;SIZE OF DATA PACKET
T34PK3: .WORD 100005 ;WRITE COMMAND, AND ACK
T34RB:
T34WB: .WORD 0 ;ADDRESS OF WRITE/READ BUFFER
        .WORD 0
T34SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
        .EVEN
;
T34RSZ: .WORD 0 ;LARGEST TAPE RECORD IN BYTES
T34CNT: .WORD 0 ;TAPE RECORD COUNTER
T34DLY: .WORD 0 ;DELAY COUNTER
;
;
T34BF2:
T34BS0: .BYTE 10 ;BSELO AREA
T34BS1: .BYTE 200 ;BSEL1 AREA
T34S2: .WORD 0 ;SEL 2 AREA
T34S3: .WORD 0 ;DATA AREA
;
;
.EVEN
;TAPE MOTION PACKET COMMAND VALUES
T34WD: .WORD 100005 ;WRITE DATA (NEXT)
T34WDR: .WORD 100405 ;WRITE DATA RETRY
T34CON: .WORD 102005 ;WRITE CONTINUOUS
        .WORD 177777 ;END OF DATA

```

TEST 6: OPERATIONS AT EOT

```

8072
8073
8074
8075
8076 060614 124 123 123 T34POS: .ASCIZ 'TSSR Incorrect After Position (SPACE RECORDS) Command'
8077 060702 127 122 111 T34ETO: .ASCIZ 'WRITE TAPE MARK At EOT Failed To Set EOT Bit (XSTO)'
8078 060766 122 105 101 T34RRE: .ASCIZ 'READ Command At EOT Didn't Give Normal Termination (TSSR)'
8079 061057 125 156 141 T34ETC: .ASCIZ 'Unable To Clear EOT Indication, (XSTO) Bit 0'
8080 061134 122 105 127 T34BOT: .ASCIZ 'REWIND Failed To Set BOT (XSTO) Bit'
8081 061200 127 122 111 T34WTH: .ASCIZ 'WRITE TAPE MARK At EOT Failed To Set Tape Status Alert'
8082 061267 127 122 111 T34ET2: .ASCIZ 'WRITE DATA At EOT Failed To Set Tape Status Alert'
8083 061351 127 122 111 T34ETN: .ASCIZ 'WRITE DATA At EOT Failed To Set EOT Bit (XSTO)'
8084 061430 123 120 101 T34ETS: .ASCIZ 'SPACE RECORDS FORWARD At EOT Failed To Set EOT Bit (XSTO)'
8085 061522 122 105 101 T34ETZ: .ASCIZ 'READ DATA At EOT Failed To Set EOT Bit (XSTO)'
8086 061600 124 123 123 T34STM: .ASCIZ 'TSSR Incorrect After SKIP TAPE MARK REVERSE At EOT'
8087 061663 120 117 123 T34TMK: .ASCIZ 'POSITION Command At EOT Onto Tape Mark Failed To Set TMK (XSTO)'
8088 061763 127 122 111 T34SSR: .ASCIZ 'WRITE Command Not Accepted'
8089 062016 105 117 124 T34ET: .ASCIZ 'EOT Not Found In 65000 3.5K Writes, (Use Shorter Tape)'
8090 062105 127 122 111 T34EOT: .ASCIZ 'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
8091 062163 124 123 123 T34TH: .ASCIZ 'TSSR Not Correct After WRITE Command Reject'
8092 062237 122 145 167 T34RMN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
8093 062306 122 101 115 T34RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
8094 062361 124 123 123 T34AM3: .ASCIZ 'TSSR Init. Failed After WRITE Command'
8095 062427 104 162 151 T34OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
8096 062502 124 123 123 T34WDD: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, SMB Bit Set'
8097 062571 124 123 123 T34WDC: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
8098 062673 103 126 103 T34VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
8099 062746 124 123 102 T34BA: .ASCIZ 'TSBA Not Correct After WRITE DATA Command'
8100 063020 127 122 111 T34WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
8101 063107 117 160 145 TST34ID: .ASCIZ 'Operations At EOT'
8102
8103
8104
8105
8106
8107
8108
8109
8110 063132
8111 063132
8112 063136 012701 060440
8113 063142 012721 100004
8114 063146 012721 060450
8115 063152 005021
8116 063154 012721 000012
8117 063160 012721 060462
8118 063164 005021
8119 063166 012721 000024
8120 063172 005021
8121 063174 012711 000000
8122 063200 012702 000030
8123 063204 012762 177777 060462 64:
8124 063212 005742
8125 063214 020227 000000
8126 063220 001371
8127 063222 000207
8128
;
;LOCAL TEXT MESSAGES FOR TEST
;-
;
;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
;WRITE SUBSYSTEM MEMORY COMMAND
;-
T34REST:
SAVREG
MOV #T34PACKET,R1 ;SAVE THE REGISTERS
MOV #100004,(R1) ;START OF THE PACKET
MOV #T34DATA,(R1) ;WRITE SUBSYSTEM MEM. WITH ACK
CLR (R1) ;ADDRESS OF CHARAISTICS DATA BLOCK
MOV #10,(R1) ;EXTENDED ADDRESS
MOV #T34BFR,(R1) ;SIZE OF DATA BLOCK IN BYTES
CLR (R1) ;ADDRESS OF MESSAGE BUFFER
MOV #20,(R1) ;LENGTH OF MESSAGE BUFFER
CLR (R1)
MOV #0,(R1) ;SELECT DRIVE ZERO
MOV #24,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
MOV #177777,T34BFR(R2) ;ALL ONES TO MESSAGE BUFFER
TST -(R2) ;BUMP DOWN TO NEXT LOCATION
CMP R2,#0 ;R2 AT ZERO YET
BNE 64: ;KEEP GOING UNTIL DONE
RTS PC ;RETURN
    
```

TEST 6: OPERATIONS AT EOT

```

8129 063224
8130 063224
8131 063230 012701 060550
8132 063234 012721 100006
8133 063240 012721 060576
8134 063244 005021
8135 063246 012721 000006
8136 063252 012701 060576
8137 063256 005021
8138 063260 005021
8139 063262 005011
8140 063264 000207
8141 063266
8142 063266
8143 063272 012701 060560
8144 063276 012721 100005
8145 063302 005021
8146 063304 005021
8147 063306 005011
8148 063310 000207
8149 063312
      063312
      063312 104401
    
```

```

T34RT2:
  SAVREG
  MOV #T34PK2,R1 ;SAVE THE REGISTERS
  MOV #100006,(R1)+ ;START OF THE PACKET
  MOV #T34BF2,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK
  CLR (R1)+ ;ADDRESS OF DATA BLOCK
  MOV #6,(R1)+ ;EXTENDED ADDRESS
  MOV #T34BF2,R1 ;SIZE OF DATA BLOCK IN BYTES
  CLR (R1)+ ;POINT TO DATA SEL AREA
  CLR (R1)+
  CLR (R1)+
  RTS PC ;RETURN
    
```

```

T34RT3:
  SAVREG
  MOV #T34PK3,R1 ;SAVE THE REGISTERS
  MOV #100005,(R1)+ ;START OF THE PACKET
  CLR (R1)+ ;WRITE TAPE. WITH ACK
  CLR (R1)+ ;ADDRESS OF DATA BLOCK
  CLR (R1)+ ;EXTENDED ADDRESS
  RTS PC ;SIZE OF DATA BLOCK
  ENDTST ;RETURN
    
```

L10061: TRAP C#ETST

.SBTTL TEST 7: EXTENDED MODE FEATURES

THIS TEST VERIFIES THE OPERATION OF COMMANDS ONLY AVAILABLE WHEN THE CONTROLLER IS IN THE EXTENDED FEATURES MODE. THESE COMMANDS ARE:

REWIND WITH IMMEDIATE INTERRUPT

IF THE CONTROLLER IS NOT ALREADY IN EXTENDED FEATURES MODE, IT IS PLACED THERE VIA A WRITE SUBSYSTEM MEMORY COMMAND.

THE TEST CONSISTS OF THE FOLLOWING 7 SUBTESTS

BGNTST

```

T7::
  MOV #EPR1,EPR1SM ;PRIMARY ERROR MESSAGE
  MOV #TST35ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
  JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
  MOV #5,LOOPCNT ;PERFORM 5 ITERATIONS
  CLR T35CNT ;CLEAR TAPE RECORD COUNTER
    
```

TEST 7. SUBTEST 1

VERIFIES THAT A REWIND WITH IMMEDIATE INTERRUPT COMMAND, ISSUED WITH THE INTERRUPT ENABLE (IE) BIT CLEAR (0), CAUSES ALMOST IMMEDIATE TERMINATION BUT NO INTERRUPT. STATUS IN THE MESSAGE

```

8150
8151
8152
8153
8154
8155
8156
8157
8158
8159
8160
8161
8162
8163
8164
8165
8166
8167
8168
8169
8170 063314 012737 006354 002172
8175 063322 012700 073033
8176 063326 004737 016570
8177 063332 012737 000005 002210
8178 063340 005037 067436
8179
8180
8181
8182
8183
8184
8185
8186
    
```





TEST 7: EXTENDED MODE FEATURES

```

8234 063542          ERRHRD  ERRNO,T35RWN,PKTSSR      ;REWIND NOT ACCEPTED
      063542 104456
      063544 001277
      063546 070544
      063550 012126
8235 063552          30$:  CKLOOP                      ;LOOP IF SELECTED
      063552 104406
8236 063554 013701 067320      MOV      T35BFR+6,R1      ;PICK UP XSTO
8237 063560 010102      MOV      R1,R2          ;SET UP EXPECTED
8238 063562 052702 000002      BIS      @BIT1,R2      ;SET BOT BIT IN EXPECTED
8239 063566 020102      CMP      R1,R2        ;DOES EXP = REC'D
8240 063570 001406      BEQ      40$          ;BR, IF EQUAL (OK)
8241 063572 005237 002214      INC      FATFLG        ;ERROR COUNT
8245 063576          ERRHRD  ERRNO,T35BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      063576 104456
      063600 001300
      063602 070240
      063604 015554
8246 063606          40$:  CKLOOP                      ;LOOP IF SELECTED
      063606 104406
8247 063610 012703 000024      MOV      @20.,R3       ;NUMBER OF RECORDS
8248 063614 012737 000400 067416  MOV      @256.,T35SZ   ;SET UP RECORD SIZE
8249 063622 013737 003116 067412  MOV      FREE,T35WB    ;ADDRESS OF WRITE BUFFER
8250
8251 ;*****
8252 ;
8253 ;WRITE DATA,ACK,CVC-1 COMMAND
8254 ;
8255 ;*****
8256
8257 063630 012737 140005 067410      MOV      @140005,T35PK3 ;WRITE DATA,ACK,CVC-1 COMMAND
8258 063636 012704 067410      MOV      @T35PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
8259 063642 010465 000000          50$:  MOV      R4,TSD8(R5) ;ISSUE COMMAND
8260 063646 004737 016330          JSR      PC,WAITF      ;WAIT FOR SSR TO SET
8261 063652 016501 000002          MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
8262 063656 012702 000200          MOV      @SSR,R2      ;SET UP EXPECTED
8263 063662 020102          CMP      R1,R2        ;ARE THEY EQUAL
8264 063664 001406          BEQ      60$          ;BR, IF OK
8265 063666 005237 002214          INC      FATFLG        ;ERROR COUNT
8269 063672          ERRHRD  ERRNO,T35WDE,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA
      063672 104456
      063674 001301
      063676 070166
      063700 012126
8270 063702          60$:  CKLOOP                      ;LOOP IF SELECTED
      063702 104406
8271 063704 005303          DEC      R3           ;BUMP RECORD COUNTER
8272 063706 001355          BNE     50$          ;BR, IF MORE RRECORDS TO COUNT
8273
8274 ;*****
8275 ;
8276 ;WAIT FOR TAPE TO STOP ALL MOTION
8277 ;
8278 ;*****
8279
8280 063710 012737 000012 067442      70$:  MOV      @10.,T35DLY ;SET UP DELAY COUNTER
8281 063716          DELAY  250      ;WAIT ABOUT .25 SEC
    
```

TEST 7: EXTENDED MODE FEATURES

```

063716 012727 000250
063722 000000
063724 013727 002116
063730 000000
063732 005367 177772
063736 001375
063740 005367 177756
063744 001367
8282 063746 005337 067442
8283 063752 001361
8284 063754 005737 002220
8285 063760 001042
8286 063762 112737 000200 067421
8287 063770 112737 000010 067420
8288 063776 012704 067400
8289 064002 010465 000000
8290 064006 004737 016416
8291 064012 103407
8292 064014 C10001
8293 064016 005237 002214
8297 064022
      064022 104456
      064024 001302
      064026 072322
      064030 012126
8298 064032
      064032 104406
8299 064034 012704 067270
8300 064040 004737 010742
8301 064044 103407
8302 064046 005237 002214
8306 064052 010001
8307 064054
      064054 104456
      064056 001303
      064060 005052
      064062 012114
8308 064064
      064064 104406
8309 064066 012737 176750 067442
8310 064074 005037 067436
8311
8312
8313
8314
8315
8316
8317
8318 064100 012737 142012 067410
8319 064106 012704 067410
8320 064112 010465 000000
8321 064116 016501 000002
8322 064122 032701 000200
8323 064126 001021
8324 064130 005237 067436
8325 064134
      064134 012727 000001

      MOV      #250,(PC)+
      .WORD   0
      MOV      L#DLY,(PC)+
      .WORD   0
      DEC      -6(PC)
      BNE     .-4
      DEC      -22(PC)
      BNE     .-20

      DEC      T35DLY      ;BUMP COUNTER DOWN
      BNE     70$         ;BR, IF MORE TO DELAY
      TST     EXTFEA      ;CHECK FOR EXTENDED FEATURES SW SWITCH
      BNE     110$        ;BR IF SWITCH IS ON
      MOVB    #200,T358S1 ;WRITE MISCELLANEOUS CONT/READ STATUS
      MOVB    #10,T358S0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
      MOV     #T35PK2,R4  ;WRITE SUBSYS MEM PACKET
      MOV     R4,TSDB(R5) ;ISSUE COMMAND
      JSR    PC,CHKTSSR   ;WAIT FOR SSR
      BCS    90$          ;BR, IF NO ERROR
      MOV     R0,R1       ;ERROR, SAVE TSSR
      INC    FATFLG      ;ERROR COUNT
      ERRHRD ERRNO,T35SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      TRAP   C#ERHRD
      .WORD  706
      .WORD  T35SSR
      .WORD  PKTSSR

90$:  CKLOOP              ;LOOP IF SELECTED
      TRAP   C#CLP1

      MOV     #T35PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
      JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
      BCS    100$          ;BR, IF COMMAND ISSUED OK
      INC    FATFLG      ;ERROR COUNT
      MOV     R0,R1       ;SAVE CONTENTS OF TSSR
      ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      TRAP   C#ERHRD
      .WORD  707
      .WORD  WRTMSG
      .WORD  SFIMSG

100$: CKLOOP              ;SCOPE LOOP
      TRAP   C#CLP1

110$: MOV     #65000.,T35DLY ;SET UP DELAY COUNTER
      CLR    T35CNT        ;DELAY COUNTER

;*****
;REWIND IMED. INTERRUPT,ACK,CVC=1,IE=0 COMMAND
;*****

      MOV     #142012,T35PK3 ;REWIND IMED. INTERRUPT,ACK,CVC=1,IE=0 COMMAND
      MOV     #T35PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
      MOV     R4,TSDB(R5)   ;ISSUE COMMAND
120$: MOV     TSSR(R5),R1   ;GET TSSR CONTENTS
      BIT     #SSR,R1       ;CHECK FOR SSR SET
      BNE     130$         ;BR, WHEN SSR IS SET
      INC    T35CNT        ;BUMP THE CYCLE COUNTER
      DELAY  1             ;DELAY TO KEEP COUNTER DOWN
      MOV     #1,(PC)+

```

TEST 7: EXTENDED MODE FEATURES

```

064140 000000
064142 013727 002116
064146 000000
064150 005367 177772
064154 001375
064156 005367 177756
064162 001367
8326 064164 005337 067442
8327 064170 001352
8328 064172 012702 000200
8329 064176 020102
8330 064200 001406
8331 064202 005237 002214
8335 064206
      064206 104456
      064210 001304
      064212 072670
      064214 012126
8336 064216
      064216 104406
8337 064220 005737 002216
8338 064224 001410
8339 064226 016501 000002
8340 064232 005237 002214
8344 064236
      064236 104456
      064240 001305
      064242 072501
      064244 012126
8345 064246
      064246 104406
8346
8347
8348
8349
8350
8351
8352
8353 064250 013701 067320
8354 064254 010102
8355 064256 052702 000200
8356 064262 020102
8357 064264 001406
8358 064266 005237 002214
8362 064272
      064272 104456
      064274 001306
      064276 072403
      064300 015554
8363 064302
      064302 104406
8364 064304 013701 067324
8365 064310 010102
8366 064312 052702 100000
8367 064316 020102
8368 064320 001406
8369 064322 005237 002214

      .WORD 0
      MOV L#DLY,(PC)+
      .WORD 0
      DEC -6(PC)
      BNE -.4
      DEC -22(PC)
      BNE .-20

130$: DEC T35DLY ;DROP DEAD TIMER BUMP DOWN
      BNE 120$ ;BR, IF MORE TIME TO GO
      MOV #SSR,R2 ;SET UP EXPECTED
      CMP R1,R2 ;ARE THEY EQUAL
      BEQ 140$ ;BR, IF OK
      INC FATFLG ;ERROR COUNT
      ERRHRD ERRNO,T35RWE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA

      TRAP C#ERHRD
      .WORD 708
      .WORD T35RWE
      .WORD PKTSSR

140$: CKLOOP ;LOOP IF SELECTED
      TRAP C#CLP1
      TST INTRECV ;CHECK FOR INTERRUPTS
      BEQ 150$ ;BR, IF NO INTERRUPTS DETECTED
      MOV TSSR(R5),R1 ;GET TSSR STATUS FOR PRINTOUT
      INC FATFLG ;ERROR COUNT
      ERRHRD ERRNO,T35INT,PKTSSR ;INTERRUPT RECEIVED (BAD)

      TRAP C#ERHRD
      .WORD 709
      .WORD T35INT
      .WORD PKTSSR

150$: CKLOOP ;LOOP IF SELECTED
      TRAP C#CLP1

;*****
;
;NOW CHECK FOR THE MOTION BITS SET
;
;*****

      MOV T35BFR+6,R1 ;PICK UP XST0
      MOV R1,R2 ;SET UP EXPECTED
      BIS #BIT7,R2 ;SET MOT BIT IN EXPECTED
      CMP R1,R2 ;DOES EXP = REC'D
      BEQ 160$ ;BR, IF EQUAL (OK)
      INC FATFLG ;ERROR COUNT
      ERRHRD ERRNO,T35MOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND

      TRAP C#ERHRD
      .WORD 710
      .WORD T35MOT
      .WORD EXPREC

160$: CKLOOP ;LOOP IF SELECTED
      TRAP C#CLP1
      MOV T35BFR+12,R1 ;PICK UP XST2
      MOV R1,R2 ;SET UP EXPECTED
      BIS #BIT15,R2 ;SET OPM BIT IN EXPECTED
      CMP R1,R2 ;DOES EXP = REC'D
      BEQ 170$ ;BR, IF EQUAL (OK)
      INC FATFLG ;ERROR COUNT

```

TEST 7: EXTENDED MODE FEATURES

```

8373 064326          ERRHRD  ERRNO,T35OPM,EXPREC      ;OPM BIT NOT SET
      064326 104456
      064330 001307
      064332 072572
      064334 015554
8374 064336          170#: CKLOOP                    ;LOOP IF SELECTED
      064336 104406
8375 064340 012737 000027 067442          MOV      #23.,T35DLY      ;SET UP DELAY COUNTER
8376 064346          175#: DELAY 250                ;START DELAY
      064346 012727 000250
      064352 000000
      064354 013727 002116
      064360 000000
      064362 005367 177772
      064366 001375
      064370 005367 177756
      064374 001367
8377 064376 005337 067442          DEC      T35DLY          ;BUMP DELAY COUNTER
8378 064402 C01361          BNE      175#           ;BR, IF MORE DELAY
8379 064404          ENDSUB
      064404
      064404 104403
8380 064406 023727 002214 000017          CMP      FATFLG,#15.    ;IS ERROR COUNT AT 25
8381 064414 103402          BLO      999#           ;BR, IF LESS THAN 25
8382 064416 004737 017262          JSR      PC,CKDROP     ;TRY TO DROP THE UNIT
8383 064422
8384
8385
8386          ;TEST 7: SUBTEST 2
8387
8388          ;
8389          ; WITH THE INTERRUPT ENABLE (IE) BIT SET (1), CAUSES ALMOST
8390          ; IMMEDIATE TERMINATION AND AN INTERRUPT. STATUS IN THE MESSAGE
8391          ; BUFFER IS CHECKED TO VERIFY THAT THE MOTION (MOT) AND OPERATION
8392          ; IN PROGRESS (OPM) STATUS BITS ARE BOTH SET.
8393          ;
8394          ;
8395          ;
8396          ;
8397          ;
8398          ;
8399          ;-
8400 064422          BGNSUB
      064422
      064422 104402
8401 064424          SETPRI #PRI00                    ;ENABLE INTERRUPTS.
      064424 012700 000000
      064430 104441
8402 064432 004737 073064          JSR      PC,T35REST    ;SET COMMAND PACKET
8403 064436 005037 002216          CLR      INTRECV      ;CLEAR INTERRUPT RECEIVED FLAG
8404 064442 004737 073156          JSR      PC,T35RT2    ;SET UP OTHER COMMAND PACKET
8405 064446 004737 073220          JSR      PC,T35RT3    ;SET UP OTHER COMMAND PACKET
8406 064452 012737 176750 067442          MOV      #65000.,T35DLY ;SET UP DELAY COUNTER
8407 064460 005037 067436          CLR      T35CNT      ;CLEAR COUNTER
8408 064464 004737 016054          JSR      PC,SOFINIT   ;DO INITIALIZE ON CONTROLLER
8409 064470 103426          BCS      20#           ;BR IF INIT WAS OK
8410 064472          DELAY 250                ;DELAY ABOUT .25 SEC

```

TEST 7: EXTENDED MODE FEATURES

064472	012727	000250								MOV	#250,(PC)+
064476	000000									.WORD	0
064500	013727	002116								MOV	L#DLY,(PC)+
064504	000000									.WORD	0
064506	005367	177772								DEC	-6(PC)
064512	001375									BNE	.-4
064514	005367	177756								DEC	-22(PC)
064520	001367									BNE	.-20
8411	064522	005337	067442			DEC	T35DLY				
8412	064526	001356				BNE	10#				
8413	064530	005237	002214			INC	FATFLG				
8417	064534	010001				MOV	R0,R1				
8418	064536					ERRDF	ERRNO,SFIERR,SFIMSG				
	064536	104455								TRAP	C#ERDF
	064540	001310								.WORD	712
	064542	003646								.WORD	SFIERR
	064544	012114								.WORD	SFIMSG
8419	064546	013737	002174	067310	20#:	MOV	UNITN,T35DSW				
8420	064554	C12704	067270			MOV	#T35PACKET,R4				
8421	064560	004737	010742			JSR	PC,WRTCHR				
8422	064564	103407				BCS	25#				
8423	064566	005237	002214			INC	FATFLG				
8427	064572	010001				MOV	R0,R1				
8428	064574					ERRHRD	ERRNO,WRTMSG,SFIMSG				
	064574	104456								TRAP	C#ERHRD
	064576	001311								.WORD	713
	064600	005052								.WORD	WRTMSG
	064602	012114								.WORD	SFIMSG
8429	064604				25#:	CKLOOP					
	064604	104406								TRAP	C#CLP1
8430	064606	004737	011074			JSR	PC,REWIND				
8431	064612	103411				BCS	30#				
8432	064614	010004				MOV	R0,R4				
8433	064616	016501	000002			MOV	TSSR(R5),R1				
8434	064622	005237	002214			INC	FATFLG				
8438	064626					ERRHRD	ERRNO,T35RWN,PKTSSR				
	064626	104456								TRAP	C#ERHRD
	064630	001312								.WORD	714
	064632	070544								.WORD	T35RWN
	064634	012126								.WORD	PKTSSR
8439	064636				30#:	CKLOOP					
	064636	104406								TRAP	C#CLP1
8440	064640	013701	067320			MOV	T35BFR+6,R1				
8441	064644	010102				MOV	R1,R2				
8442	064646	052702	000002			BIS	#BIT1,R2				
8443	064652	020102				CMP	R1,R2				
8444	064654	001406				BEQ	40#				
8445	064656	005237	002214			INC	FATFLG				
8449	064662					ERRHRD	ERRNO,T35BOT,EXPREC				
	064662	104456								TRAP	C#ERHRD
	064664	001313								.WORD	715
	064666	070240								.WORD	T35BOT
	064670	015554								.WORD	EXPREC
8450	064672				40#:	CKLOOP					
	064672	104406								TRAP	C#CLP1
8451	064674	012703	000024			MOV	#20.,R3				
8452	064700	012737	000400	067416		MOV	#256.,T35SZ				

TEST 7: EXTENDED MODE FEATURES

```

8453 064706 013737 003116 067412      MOV      FREE,T35WB      ;ADDRESS OF WRITE BUFFER
8454
8455      ;*****
8456      ;
8457      ;WRITE DATA,ACK,CVC=1 COMMAND
8458      ;
8459      ;*****
8460
8461 064714 012737 140005 067410      MOV      #140005,T35PK3  ;WRITE DATA,ACK,CVC=1 COMMAND
8462 064722 012704 067410      MOV      #T35PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
8463 064726 010465 000000      50$:    MOV      R4,TSDB(R5)   ;ISSUE COMMAND
8464 064732 004737 016330      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
8465 064736 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
8466 064742 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
8467 064746 020102      CMP      R1,R2          ;ARE THEY EQUAL
8468 064750 001406      BEQ      60$            ;BR, IF OK
8469 064752 005237 002214      INC      FATFLG         ;ERROR COUNT
8473 064756      ERRHRD  ERRNO,T35WDE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      TRAP  C#ERHRD
      .WORD 716
      .WORD T35WDE
      .WORD PKTSSR
      064756 104456
      064760 001314
      064762 070166
      064764 012126
8474 064766      60$:    CKLOOP          ;LOOP IF SELECTED
      TRAP  C#CLP1
      064766 104406
8475
8476      ;*****
8477      ;
8478      ;WAIT FOR TAPE TO STOP ALL MOTION
8479      ;
8480      ;*****
8481
8482 064770 012737 000012 067442      70$:    MOV      #10.,T35DLY ;SET UP DELAY COUNTER
8483 064776      DELAY  250           ;WAIT ABOUT .25 SEC
      MOV      #250,(PC)+
      .WORD  0
      MOV      L#DLY,(PC)+
      .WORD  0
      DEC      -6(PC)
      BNE      -.4
      DEC      -22(PC)
      BNE      -.20
      064776 012727 000250
      065002 000000
      065004 013727 002116
      065010 000000
      065012 005367 177772
      065016 001375
      065020 005367 177756
      065024 001367
8484 065026 005337 067442      DEC      T35DLY        ;BUMP COUNTER DOWN
8485 065032 001361      BNE      70$           ;BR, IF MORE TO DELAY
8486 065034 005737 002220      TST      EXTFEA        ;CHECK FOR EXTENDED FEATURES SW SWITCH
8487 065040 001042      BNE      110$          ;BR IF SWITCH IS ON
8488 065042 112737 000200 067421      MOVB     #200,T35BS1    ;WRITE MISCELLANEOUS CONT/READ STATUS
8489 065050 112737 000010 067420      MOVB     #10,T35BS0    ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
8490 065056 012704 067400      MOV      #T35PK2,R4    ;WRITE SUBSYS MEM PACKET
8491 065062 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
8492 065066 004737 016416      JSR      PC,CHKTSSR    ;WAIT FOR SSR
8493 065072 103407      BCS      90$           ;BR, IF NO ERROR
8494 065074 010001      MOV      R0,R1         ;ERROR, SAVE TSSR
8495 065076 005237 002214      INC      FATFLG         ;ERROR COUNT
8499 065102      ERRHRD  ERRNO,T35SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      TRAP  C#ERHRD
      .WORD 717
      .WORD T35SSR
      065102 104456
      065104 001315
      065106 072322
    
```

TEST 7: EXTENDED MODE FEATURES

```

8500 065110 012126          90#:  CKLOOP          ;LOOP IF SELECTED          .WORD  PKTSSR
      065112 104406          ;SUBROUTINE NEEDS PACKET ADDRESS          TRAP  C#CLP1
8501 065114 012704 067270  MOV    #T3SPACKET,R4
8502 065120 004737 010742  JSR    PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
8503 065124 103407          BCS    100#          ;BR, IF COMMAND ISSUED OK
8504 065126 005237 002214  INC    FATFLG          ;ERROR COUNT
8508 065132 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
8509 065134          ERRHRD  ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICS FAILED
      065134 104456          TRAP  C#ERHRD
      065136 001316          .WORD  718
      065140 005052          .WORD  WRTMSG
      065142 012114          .WORD  SFMSG
8510 065144          100#:  CKLOOP          ;SCOPE LOOP          TRAP  C#CLP1
      065144 104406
8511 065146 012737 176750 067442 110#:  MOV    #65000.,T35DLY ;SET UP DELAY COUNTER
8512 065154 005037 067436  CLR    T35CNT          ;DELAY COUNTER
8513
8514          ;*****
8515          ;
8516          ;REWIND IMMEDIATE,ACK,CVC=1,IE=1 COMMAND
8517          ;
8518          ;*****
8519
8520 065160 012737 142212 067410  MOV    #142212,T35PK3 ;REWIND IMMEDIATE,ACK,CVC=1,IE=1 COMMAND
8521 065166 012704 067410  MOV    #T35PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
8522 065172 010465 000000  MOV    R4,TSDB(R5)     ;ISSUE COMMAND
8523 065176 016501 000002 120#:  MOV    TSSR(R5),R1  ;GET TSSR CONTENTS
8524 065202 032701 000200  BIT    #SSR,R1         ;CHECK FOR SSR SET
8525 065206 001021          BNE    130#          ;BR, WHEN SSR IS SET
8526 065210 005237 067436  INC    T35CNT          ;BUMP THE CYCLE COUNTER
8527 065214          DELAY  1              ;DELAY TO KEEP COUNTER DOWN
      065214 012727 000001  MOV    #1,(PC)+
      065220 000000          .WORD  0
      065222 013727 002116  MOV    L#DLY,(PC)+
      065226 000000          .WORD  0
      065230 005367 177772  DEC    -6(PC)
      065234 001375          BNE    -4
      065236 005367 177756  DEC    -22(PC)
      065242 001367          BNE    -20
8528 065244 005337 067442  DEC    T35DLY          ;DROP DEAD TIMER BUMP DOWN
8529 065250 001352          BNE    120#          ;BR, IF MORE TIME TO GO
8530 065252 012702 000200 130#:  MOV    #SSR,R2      ;SET UP EXPECTED
8531 065256 020102          CMP    R1,R2          ;ARE THEY EQUAL
8532 065260 001406          BEQ    140#          ;BR, IF OK
8533 065262 005237 002214  INC    FATFLG          ;ERROR COUNT
8537 065266          ERRHRD  ERRNO,T35RWE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      065266 104456          TRAP  C#ERHRD
      065270 001317          .WORD  719
      065272 072670          .WORD  T35RWE
      065274 012126          .WORD  PKTSSR
8538 065276          140#:  CKLOOP          ;LOOP IF SELECTED          TRAP  C#CLP1
      065276 104406
8539 065300 005737 002216  TST    INTRECV          ;CHECK FOR INTERRUPTS
8540 065304 001010          BNE    150#          ;BR, IF INTERRUPTS DETECTED
8541 065306 016501 000002  MOV    TSSR(R5),R1  ;GET TSSR STATUS FOR PRINTOUT
8542 065312 005237 002214  INC    FATFLG          ;ERROR COUNT
    
```



TEST 7: EXTENDED MODE FEATURES

```

8546 065316          ERRHRD  ERRNO,T35NIN,PKTSSR      ;INTERRUPT NOT RECEIVED (BAD)
      065316 104456          TRAP                  C#ERHRD
      065320 001320          .WORD                  720
      065322 072756          .WORD                  T35NIN
      065324 012126          .WORD                  PKTSSR
8547 065326          150#:  CKLOOP                    ;LOOP IF SELECTED
      065326 104406          TRAP                  C#CLP1
8548
8549          ;*****
8550          ;
8551          ;NOW CHECK FOR THE MOTION BITS SET
8552          ;
8553          ;*****
8554
8555 065330 013701 067320      MOV          T35BFR+6,R1      ;PICK UP XST0
8556 065334 010102          MOV          R1,R2          ;SET UP EXPECTED
8557 065336 052702 000200      BIS          #BIT7,R2      ;SET MOT BIT IN EXPECTED
8558 065342 020102          CMP          R1,R2          ;DOES EXP = REC'D
8559 065344 C01406          BEQ          160#          ;BR, IF EQUAL (OK)
8560 065346 005237 002214      INC          FATFLG          ;ERROR COUNT
8564 065352          ERRHRD  ERRNO,T35MOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      065352 104456          TRAP                  C#ERHRD
      065354 001321          .WORD                  721
      065356 072403          .WORD                  T35MOT
      065360 015554          .WORD                  EXPREC
8565 065362          160#:  CKLOOP                    ;LOOP IF SELECTED
      065362 104406          TRAP                  C#CLP1
8566 065364 013701 067324      MOV          T35BFR+12,R1   ;PICK UP XST2
8567 065370 010102          MOV          R1,R2          ;SET UP EXPECTED
8568 065372 052702 100000      BIS          #BIT15,R2     ;SET OPM BIT IN EXPECTED
8569 065376 020102          CMP          R1,R2          ;DOES EXP = REC'D
8570 065400 001406          BEQ          170#          ;BR, IF EQUAL (OK)
8571 065402 005237 002214      INC          FATFLG          ;ERROR COUNT
8575 065406          ERRHRD  ERRNO,T35OPM,EXPREC      ;OPM BIT NOT SET
      065406 104456          TRAP                  C#ERHRD
      065410 001322          .WORD                  722
      065412 072572          .WORD                  T35OPM
      065414 015554          .WORD                  EXPREC
8576 065416          170#:  CKLOOP                    ;LOOP IF SELECTED
      065416 104406          TRAP                  C#CLP1
8577 065420 012737 000027 067442 175#:  MOV          #23.,T35DLY      ;SET UP DELAY COUNTER
8578 065426          DELAY          250          ;START DELAY
      065426 012727 000250          MOV          #250,(PC)+
      065432 000000          .WORD                  0
      065434 013727 002116          MOV          L#DLY,(PC)+
      065440 000000          .WORD                  0
      065442 005367 177772          DEC          -6(PC)
      065446 001375          BNE          -4
      065450 005367 177756          DEC          -22(PC)
      065454 001367          BNE          -20
8579 065456 005337 067442      DEC          T35DLY          ;BUMP DELAY COUNTER
8580 065462 001361          BNE          175#          ;BR, IF MORE DELAY
8581 065464          ENDSUB
      065464 104403          L10065:  TRAP                  C#ESUB
8582 065466 023727 002214 000017      CMP          FATFLG,#15.
8583 065474 103402          BLO          999#          ;IS ERROR COUNT AT 25
                                ;BR, IF LESS THAN 25

```



TEST 7: EXTENDED MODE FEATURES

```

065626 001325 .WORD 725
065630 070544 .WORD T35RWN
065632 012126 .WORD PKTSSR
8638 065634 104406 30#: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
065634 104406 ;PICK UP XSTO
8639 065636 013701 067320 MOV T35FR+6,R1 ;SET UP EXPECTED
8640 065642 010102 MOV R1,R2 ;SET BOT BIT IN EXPECTED
8641 065644 052702 000002 BIS #BIT1,R2 ;DOES EXP = REC'D
8642 065650 020102 CMP R1,R2 ;BR, IF EQUAL (OK)
8643 065652 001406 BEQ 40# ;ERROR COUNT
8644 065654 005237 002214 INC FATFLG ;TAPE NOT AT BOT AFTER REWIND
8648 065660 ERRHRD ERRNO,T35BOT,EXPREC ;TRAP C#ERHRD
065660 104456 TRAP C#ERHRD
065662 001326 .WORD 726
065664 070240 .WORD T35BOT
065666 015554 .WORD EXPREC
8649 065670 104406 40#: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
065670 104406 ;STARTING RECORD SIZE
8650 065672 C12703 000024 MOV #20.,R3 ;STARTING WRITE BUFFER ADDRESS
8651 065676 013737 003116 067412 MOV FREE,T35WB
8652 ;*****
8653 ;WRITE DATA,CVC=1,ACK COMMAND
8654 ;
8655 ;*****
8656
8657
8658
8659 065704 012737 140005 067410 65#: MOV #140005,T35PK3 ;WRITE DATA,CVC=1,ACK COMMAND
8660 065712 012704 067410 MOV #T35PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8661 065716 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
8662 065720 004737 017502 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
8663 065724 010337 067416 MOV R3,T35SZ ;SET UP RECORD SIZE IN PACKET
8664 065730 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
8665 065734 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
8666 065740 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8667 065744 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
8668 065750 020102 CMP R1,R2 ;ARE THEY EQUAL
8669 065752 001406 BEQ 80# ;BR, IF OK
8670 065754 005237 002214 INC FATFLG ;ERROR COUNT
8674 065760 ERRHRD ERRNO,T35WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
065760 104456 TRAP C#ERHRD
065762 001327 .WORD 727
065764 071100 .WORD T35WDC
065766 012126 .WORD PKTSSR
8675 065770 104406 80#: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
065770 104406 ;WRITE DATA
8676 ;*****
8677 ;WRITE DATA RETRY,CVC=1,ACK COMMAND
8678 ;
8679 ;*****
8680
8681
8682
8683 065772 012737 141005 067410 MOV #141005,T35PK3 ;WRITE DATA RETRY,CVC=1,ACK COMMAND
8684 066000 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
8685 066004 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
8686 066010 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS

```

TEST 7: EXTENDED MODE FEATURES

```

8687 066014 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
8688 066020 020102      CMP      R1,R2      ;ARE THEY EQUAL
8689 066022 001406      BEQ     90$          ;BR, IF OK
8690 066024 005237 002214      INC     FATFLG      ;ERROR COUNT
8694 066030      ERRHRD  ERRNO,T35WRF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA RETRY
      066030 104456      TRAP    C#ERHRD
      066032 001330      .WORD  728
      066034 072145      .WORD  T35WRF
      066036 012126      .WORD  PKTSSR
8695 066040      90$:  CKLOOP      ;LOOP IF SELECTED
      066040 104406      TRAP    C#CLP1
8696 066042 005723      TST     (R3)+      ;BUMP RECORD SIZE COUNTER
8697 066044 020327 000052      CMP     R3,#42.    ;AT 42 SIZE YET
8698 066050 001315      BNE     65$          ;BR, IF MORE RECORDS TO WRITE
8699 066052 004737 011074      JSR     PC,REWIND  ;CALL TAPE REWIND COMMAND
8700 066056 103411      BCS     230$        ;BR, IF NO PROBLEM
8701 066060 010001      MOV     R0,R1      ;SAVE TSSR
8702 066062 016501 000002      MOV     TSSR(R5),R1 ;GET TSSR CONTENTS
8703 066066 C05237 002214      INC     FATFLG      ;ERROR COUNT
8707 066072      ERRHRD  ERRNO,T35RWN,EXPREC ;REWIND NOT ACCEPTED
      066072 104456      TRAP    C#ERHRD
      066074 001331      .WORD  729
      066076 070544      .WORD  T35RWN
      066100 015554      .WORD  EXPREC
8708 066102      230$: CKLOOP      ;LOOP IF SELECTED
      066102 104406      TRAP    C#CLP1
8709 066104 013701 067320      MOV     T35BFR+6,R1 ;PICK UP XSTO
8710 066110 010102      MOV     R1,R2      ;SET UP EXPECTED
8711 066112 052702 000002      BIS     #BIT1,R2   ;SET BOT BIT IN EXPECTED
8712 066116 020102      CMP     R1,R2      ;DOES EXP = REC'D
8713 066120 001406      BEQ     240$        ;BR, IF EQUAL (OK)
8714 066122 005237 002214      INC     FATFLG      ;ERROR COUNT
8718 066126      ERRHRD  ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      066126 104456      TRAP    C#ERHRD
      066130 001332      .WORD  730
      066132 070240      .WORD  T35BOT
      066134 015554      .WORD  EXPREC
8719 066136      240$: CKLOOP      ;LOOP IF SELECTED
      066136 104406      TRAP    C#CLP1
8720 066140 012703 000024      MOV     #20.,R3    ;STARTING RECORD SIZE
8721 066144 013737 003116 067412      MOV     FREE,T35RB ;STARTING READ BUFFER ADDRESS
8722
8723      ;*****
8724      ;
8725      ;READ DATA,ACK COMMAND
8726      ;
8727      ;*****
8728
8729 066152 012737 100001 067410 265$: MOV     #100001,T35PK3 ;READ DATA,ACK COMMAND
8730 066160 012704 067410      MOV     #T35PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8731 066164 012700 177777      MOV     #177777,R0 ;SET PATTERN IN CORRECT REGISTER
8732 066170 004737 017502      JSR     PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
8733 066174 010337 067416      MOV     R3,T35SZ   ;SET UP RECORD SIZE IN PACKET
8734 066200 010465 000000      MOV     R4,T35DB(R5) ;ISSUE COMMAND
8735 066204 004737 016330      JSR     PC,WAITF   ;WAIT FOR SSR TO SET
8736 066210 016501 000002      MOV     TSSR(R5),R1 ;GET TSSR CONTENTS
8737 066214 012702 000200      MOV     #SSR,R2   ;SET UP EXPECTED
    
```





TEST 7: EXTENDED MODE FEATURES

```

066566 104456 TRAP C#ERHRD
066570 001340 .WORD 736
066572 070240 .WORD T35BOT
066574 015554 .WORD EXPREC
8833 066576 40$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
066576 104406 ;STARTING RECORD SIZE
8834 066600 012703 000024 MOV #20,R3 ;STARTING WRITE BUFFER ADDRESS
8835 066604 013737 003116 067412 MOV FREE,T35WB
8836 ;*****
8837 ;
8838 ;WRITE DATA,CVC=1,ACK COMMAND
8839 ;
8840 ;*****
8841 ;
8842 ;*****
8843 066612 012737 140005 067410 65$: MOV #140005,T35PK3 ;WRITE DATA,CVC=1,ACK COMMAND
8844 066620 012704 067410 MOV #T35PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8845 066624 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
8846 066626 C04737 017502 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
8847 066632 010337 067416 MOV R3,T35SZ ;SET UP RECORD SIZE IN PACKET
8848 066636 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
8849 066642 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
8850 066646 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8851 066652 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
8852 066656 020102 CMP R1,R2 ;ARE THEY EQUAL
8853 066660 001406 BEQ 80$ ;BR, IF OK
8854 066662 005237 002214 INC FATFLG ;ERROR COUNT
8858 066666 ERRHRD ERRNO,T35WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
066666 104456 TRAP C#ERHRD
066670 001341 .WORD 737
066672 071100 .WORD T35WDC
066674 012126 .WORD PKTSSR
8859 066676 80$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
066676 104406 ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
8860 ;*****
8861 ;
8862 ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
8863 ;
8864 ;*****
8865 ;
8866 ;*****
8867 066700 012737 111005 067410 MOV #111005,T35PK3 ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
8868 066706 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
8869 066712 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
8870 066716 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8871 066722 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
8872 066726 020102 CMP R1,R2 ;ARE THEY EQUAL
8873 066730 001406 BEQ 90$ ;BR, IF OK
8874 066732 005237 002214 INC FATFLG ;ERROR COUNT
8878 066736 ERRHRD ERRNO,T35WRF,EXPREC ;TSSR INCORRECT AFTER WRITE DATA RETRY
066736 104456 TRAP C#ERHRD
066740 001342 .WORD 738
066742 072145 .WORD T35WRF
066744 015554 .WORD EXPREC
8879 066746 90$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
066746 104406 ;BUMP RECORD SIZE COUNTER
8880 066750 005723 TST (R3)+

```

TEST 7: EXTENDED MODE FEATURES

```

8881 066752 020327 000052          CMP      R3,#42.          ;AT 42 SIZE YET
8882 066756 001315          BNE      65$             ;BR, IF MORE RECORDS TO WRITE
8883 066760 004737 011074          JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
8884 066764 103411          BCS      230$           ;BR, IF NO PROBLEM
8885 066766 016501 000002          MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
8886 066772 010004          MOV      R0,R4          ;GET PACKET ADDRESS
8887 066774 005237 002214          INC      FATFLG         ;ERROR COUNT
8891 067000          ERRHRD  ERRNO,T35RWN,PKTSSR ;REWIND NOT ACCEPTED
      067000 104456          TRAP     C$ERHRD
      067002 001343          .WORD   739
      067004 070544          .WORD   T35RWN
      067006 012126          .WORD   PKTSSR
8892 067010          230$:  CKLOOP          ;LOOP IF SELECTED          TRAP     C$CLP1
      067010 104406
8893 067012 013701 067320          MOV      T35BFR+6,R1    ;PICK UP XSTO
8894 067016 010102          MOV      R1,R2          ;SET UP EXPECTED
8895 067020 052702 000002          BIS      #BIT1,R2       ;SET BOT BIT IN EXPECTED
8896 067024 020102          CMP      R1,R2          ;DOES EXP = REC'D
8897 067026 C01406          BEQ      240$           ;BR, IF EQUAL (OK)
8898 067030 005237 002214          INC      FATFLG         ;ERROR COUNT
8902 067034          ERRHRD  ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      067034 104456          TRAP     C$ERHRD
      067036 001344          .WORD   740
      067040 070240          .WORD   T35BOT
      067042 015554          .WORD   EXPREC
8903 067044          240$:  CKLOOP          ;LOOP IF SELECTED          TRAP     C$CLP1
      067044 104406
8904 067046 012703 000024          MOV      #20.,R3        ;STARTING RECORD SIZE
8905 067052 013737 003116 067412          MOV      FREE,T35RB     ;STARTING READ BUFFER ADDRESS
8906
8907          ;*****
8908          ;
8909          ;READ DATA,ACK COMMAND
8910          ;
8911          ;*****
8912
8913 067060 012737 100001 067410 265$:  MOV      #100001,T35PK3  ;READ DATA,ACK COMMAND
8914 067066 012704 067410          MOV      #T35PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
8915 067072 010337 067416          MOV      R3,T35SZ      ;SET UP RECORD SIZE IN PACKET
8916 067076 010465 000000          MOV      R4,T5DB(R5)    ;ISSUE COMMAND
8917 067102 004737 016330          JSR      PC,WAITF       ;WAIT FOR SSR TO SET
8918 067106 016501 000002          MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
8919 067112 012702 000200          MOV      #SSR,R2       ;SET UP EXPECTED
8920 067116 020102          CMP      R1,R2          ;ARE THEY EQUAL
8921 067120 001406          BEQ      280$           ;BR, IF OK
8922 067122 005237 002214          INC      FATFLG         ;ERROR COUNT
8926 067126          ERRHRD  ERRNO,T35RDF,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      067126 104456          TRAP     C$ERHRD
      067130 001345          .WORD   741
      067132 067532          .WORD   T35RDF
      067134 012126          .WORD   PKTSSR
8927 067136          280$:  CKLOOP          ;LOOP IF SELECTED          TRAP     C$CLP1
      067136 104406
8928 067140 013702 003116          MOV      FREE,R2       ;GET BUFFER ADDRESS
8929 067144 010304          MOV      R3,R4          ;GET RECORD SIZE
8930 067146 162704 000024          SUB      #20.,R4        ;POINT BACK TO 1ST RECORD
8931 067152 060204          285$:  ADD      R2,R4     ;POINT TO 1ST LOC IN BUFFER

```







TEST 7: EXTENDED MODE FEATURES

9046	071010	124	123	123	T35WDD:	.ASCIZ	'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
9047	071100	124	123	123	T35WDC:	.ASCIZ	'TSSR Not Correct After REREAD DATA Command'
9048	071153	103	126	103	T35VCK:	.ASCIZ	'CVC Set, Didn't Reset VCK In Message Buffer'
9049	071226	124	123	102	T35BA:	.ASCIZ	'TSBA Not Correct After REREAD DATA Command'
9050	071301	127	122	111	T35WSS:	.ASCIZ	'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
9051	071370	122	145	141	T35LON:	.ASCIZ	'Reading Long Record Failed To Set RLL Bit In XST0'
9052	071452	122	145	141	T35LOP:	.ASCIZ	'Reading Long Record Failed To Set RLS Bit In XST0'
9053	071534	122	145	163	T35PBP:	.ASCIZ	'Residual Byte Count Incorrect After Short Record Read'
9054	071622	122	145	141	T35TRL:	.ASCIZ	'Reading Long Record Failed To Give Tape Status Alert'
9055	071710	127	122	111	T35NEF:	.ASCIZ	'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
9056	072006	124	123	123	T35SCF:	.ASCIZ	'TSSR Not Correct After SPACE RECORDS Command'
9057	072063	124	123	123	T35TSA:	.ASCIZ	'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
9058	072145	124	123	123	T35WRF:	.ASCIZ	'TSSR Not Correct After WRITE DATA RETRY Command'
9059	072225	104	141	164	T35DTA:	.ASCIZ	'Data Compare Error, Data Read From Tape Not Equal To Written'
9060	072322	124	123	123	T35SSR:	.ASCIZ	'TSSR Incorrect After WRITE MISCELLANEOUS Command'
9061	072403	115	117	124	T35MOT:	.ASCIZ	'MOT Bit (XST0) Not Set During Rewind (Extended Features Mode)'
9062	072501	111	156	164	T35INT:	.ASCIZ	'Interrupt Received After REWIND Command (IE Bit Not Set)'
9063	072572	117	120	115	T35OPM:	.ASCIZ	'OPM Bit (XST2) Not Set During Rewind (Extended Features Mode)'
9064	072670	124	123	123	T35RWE:	.ASCIZ	'TSSR Incorrect After Extended Features REWIND Command'
9065	072756	116	157	040	T35NIN:	.ASCIZ	'No Interrupt Detected After REWIND IMMEDIATE'
9066	073033	105	170	164	TST35ID:	.ASCIZ	'Extended Mode Functions'
9067						.EVEN	
9068						;	
9069						;	
9070						;	
9071						;	
9072						;	
9073						;	
9074						;	
9075	073064				T35REST:		
9076	073064					SAVREG	
9077	073070	012701	067270			MOV	#T35PACKET,R1 ;SAVE THE REGISTERS
9078	073074	012721	100004			MOV	#100004,(R1)+ ;START OF THE PACKET
9079	073100	012721	067300			MOV	#T35DATA,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
9080	073104	005021				CLR	(R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
9081	073106	012721	000012			MOV	#10,(R1)+ ;EXTENDED ADDRESS
9082	073112	012721	067312			MOV	#T35BFR,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
9083	073116	005021				CLR	(R1)+ ;ADDRESS OF MESSAGE BUFFER
9084	073120	012721	000024			MOV	#20,(R1)+ ;LENGTH OF MESSAGE BUFFER
9085	073124	005021				CLR	(R1)+
9086	073126	012711	000000			MOV	#0,(R1) ;SELECT DRIVE ZERO
9087	073132	012702	000030			MOV	#24,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
9088	073136	012762	177777	067312	64#:	MOV	#177777,T35BFR(R2) ;ALL ONES TO MESSAGE BUFFER
9089	073144	005742				TST	-(R2) ;NEXT LOCATION
9090	073146	022702	000000			CMP	#0,R2 ;AT END OF LOOP YET
9091	073152	001371				BNE	64# ;KEEP GOING UNTIL DONE
9092	073154	000207				RTS	PC ;RETURN
9093							
9094	073156				T35RT2:		
9095	073156					SAVREG	
9096	073162	012701	067400			MOV	#T35PK2,R1 ;SAVE THE REGISTERS
9097	073166	012721	100006			MOV	#100006,(R1)+ ;START OF THE PACKET
9098	073172	012721	067420			MOV	#T35BF2,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
9099	073176	005021				CLR	(R1)+ ;ADDRESS OF DATA BLOCK
9100	073200	012721	000006			MOV	#6,(R1)+ ;EXTENDED ADDRESS
9101	073204	005021				CLR	(R1)+ ;SIZE OF DATA BLOCK IN BYTES
9102	073206	012701	067420			MOV	#T35BF2,R1 ;POINT TO DATA SEL AREA





TEST 8: RECORD BUFFERING

9208	073374	005237	002214		INC	FATFLG							
9212	073400	010001			MOV	RO,R1							
9213	073402				ERRDF	ERRNO,SFIERR,SFIMSG							
	073402	104455											
	073404	001441											
	073406	003646											
	073410	012114											
9214	073412	013737	002174	075500	20:	MOV	UNITN,T36DSW						
9215	073420	012704	075460			MOV	#T36PACKET,R4						
9216	073424	004737	010742			JSR	PC,WRTCHR						
9217	073430	103407				BCS	25:						
9218	073432	005237	002214			INC	FATFLG						
9222	073436	010001				MOV	RO,R1						
9223	073440					ERRHRD	ERRNO,WRTMSG,SFIMSG						
	073440	104456											
	073442	001442											
	073444	005052											
	073446	012114											
9224	073450				25:	CKLOOP							
	073450	104406											
9225	073452	004737	011074			JSR	PC,REWIND						
9226	073456	103407				BCS	30:						
9227	073460	010004				MOV	RO,R4						
9228	073462	005237	002214			INC	FATFLG						
9232	073466					ERRHRD	ERRNO,T36RWN,PKTSSR						
	073466	104456											
	073470	001443											
	073472	077041											
	073474	012126											
9233	073476				30:	CKLOOP							
	073476	104406											
9234	073500	013701	075510			MOV	T36BFR+6,R1						
9235	073504	010102				MOV	R1,R2						
9236	073506	052702	000002			BIS	#BIT1,R2						
9237	073512	020102				CMP	R1,R2						
9238	073514	001406				BEQ	40:						
9239	073516	005237	002214			INC	FATFLG						
9243	073522					ERRHRD	ERRNO,T36BOT,EXPREC						
	073522	104456											
	073524	001444											
	073526	076535											
	073530	015554											
9244	073532				40:	CKLOOP							
	073532	104406											
9245	073534	013737	002174	075500		MOV	UNITN,T36DSW						
9246	073542	052737	000030	075500		BIS	#BIT3:BIT4,T36DSW						
9247	073550	012704	075460			MOV	#T36PACKET,R4						
9248	073554	004737	010742			JSR	PC,WRTCHR						
9249	073560	103407				BCS	50:						
9250	073562	005237	002214			INC	FATFLG						
9254	073566	010001				MOV	RO,R1						
9255	073570					ERRHRD	ERRNO,WRTMSG,SFIMSG						
	073570	104456											
	073572	001445											
	073574	005052											
	073576	012114											
9256	073600				50:	CKLOOP							

TEST 8: RECORD BUFFERING

9257	073600	104406							TRAP	C#CLP1
9258	073602	012737	003720	075606	MOV	#2000.,T36SZ				
9259	073610	013737	003116	075602	MOV	FREE,T36WB				
9260	073616	012737	140005	075600	MOV	#140005,T36PK3				
9261	073624	012704	075600		MOV	#T36PK3,R4				
9262	073630	010465	000000		MOV	R4,TSDB(R5)				
9263	073634	004737	016330		JSR	PC,WAITF				
9264	073640	016501	000002		MOV	TSSR(R5),R1				
9265	073644	012702	000200		MOV	#SSR,R2				
9266	073650	020102			CMP	R1,R2				
9267	073652	001406			BEQ	60#				
9271	073654	005237	002214		INC	FATFLG				
	073660				ERRHRD	ERRNO,WRTERR,PKTSSR				
	073660	104456								
	073662	001446							TRAP	C#ERHRD
	073664	005107							.WORD	806
	073666	012126							.WORD	WRTERR
									.WORD	PKTSSR
9272	073670				60#:	CKLOOP				
	073670	104406								
9273	073672	012737	000005	075632	MOV	#05.,T36DLY			TRAP	C#CLP1
9274	073700				70#:	DELAY	1			
	073700	012727	000001							
	073704	000000								
	073706	013727	002116							
	073712	000000								
	073714	005367	177772							
	073720	001375								
	073722	005367	177756							
	073726	001367								
9275	073730	005337	075632			T36DLY				
9276	073734	001361			BNE	70#				
9277	073736	012737	006642	075606	MOV	#3490.,T36SZ				
9278	073744	012737	140005	075600	MOV	#140005,T36PK3				
9279	073752	012704	075600		MOV	#T36PK3,R4				
9280	073756	005037	075626		CLR	T36CNT				
9281	073762	012737	001750	075632	MOV	#1000.,T36DLY				
9282	073770	010465	000000		MOV	R4,TSDB(R5)				
9283	073774	016501	000002		MOV	TSSR(R5),R1				
9284	074000	032701	000200	80#:	MOV	#SSR,R1				
9285	074004	001021			BIT	90#				
9286	074006	005237	075626		BNE					
9287	074012				INC	T36CNT				
	074012	012727	000001		DELAY	1				
	074016	000000								
	074020	013727	002116							
	074024	000000								
	074026	005367	177772							
	074032	001375								
	074034	005367	177756							
	074040	001367								
9288	074042	005337	075632		DEC	T36DLY				
9289	074046	001352			BNE	80#				
9290	074050	012702	000200	90#:	MOV	#SSR,R2				
9291	074054	020102			CMP	R1,R2				
9292	074056	001406			BEQ	100#				
9293	074060	005237	002214		INC	FATFLG				
9297	074064				ERRHRD	ERRNO,T36WDE,PKTSSR				

TEST 8: RECORD BUFFERING

```

074064 104456                                TRAP    C#ERHRD
074066 001447                                .WORD  807
074070 076463                                .WORD  T36WDE
074072 012126                                .WORD  PKTSSR
9298 074074 100#: CKLOOP                      ;LOOP IF SELECTED
074074 104406                                TRAP    C#CLP1
9299 074076 013737 002174 075500             MOV     UNITN,T36DSW      ;SET UP DRIVE NUMBER
9300 074104 052737 000010 075500             BIS     #BIT3,T36DSW     ;25-APR-83 REV B - TURN OFF BUFFERING
9301 074112 012704 075460                     MOV     #T36PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
9302 074116 004737 010742                     JSR     PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
9303 074122 103407                             BCS     110#             ;BR, IF COMMAND ISSUED OK
9304 074124 005237 002214                     INC     FATFLG           ;ERROR COUNT
9308 074130 010001                             MOV     R0,R1            ;SAVE CONTENTS OF TSSR
9309 074132                                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
074132 104456                                TRAP    C#ERHRD
074134 001450                                .WORD  808
074136 005052                                .WORD  WRTMSG
074140 012114                                .WORD  SFIMSG
9310 074142 110#: CKLOOP                      ;LOOP IF SELECTED
074142 104406                                TRAP    C#CLP1
9311 074144 012737 006642 075606             MOV     #3490.,T36SZ     ;SET SIZE OF TRANSFER
9312 074152 012737 140005 075600             MOV     #140005,T36PK3  ;WRITE DATA,ACK,CVC=1 COMMAND
9313 074160 012704 075600                     MOV     #T36PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
9314 074164 005037 075630                     CLR     T36CNU          ;CLEAR COUNTER
9315 074170 012737 001750 075632             MOV     #1000.,T36DLY   ;SET DROP DEAD COUNTER VALUE
9316 074176 010465 000000                     MOV     R4,TSDB(R5)     ;ISSUE COMMAND
9317 074202 016501 000002                     MOV     TSSR(R5),R1     ;GET TSSR CONTENTS
9318 074206 032701 000200                     BIT     #SSR,R1         ;CHECK FOR SSR SET
9319 074212 001021                             BNE     130#            ;BR, IF SSR IS SET
9320 074214 005237 075630                     INC     T36CNU          ;BUMP CYCLE COUNTER
9321 074220                                DELAY  1                ;CUT NUMBER OF LOOPS DOWN
074220 012727 000001                             MOV     #1.(PC)+0
074224 000000                                .WORD  0
074226 013727 002116                             MOV     L#DLY.(PC)+0
074232 000000                                .WORD  0
074234 005367 177772                             DEC     -6(PC)
074240 001375                                BNE     -.4
074242 005367 177756                             DEC     -22(PC)
074246 001367                                BNE     -.20
9322 074250 005337 075632                     DEC     T36DLY          ;BUMP DROP DEAD COUNTER
9323 074254 001352                             BNE     120#            ;BR, IF THERE IS STILL TIME
9324 074256 012702 000200                     MOV     #SSR,R2        ;SET UP EXPECTED
9325 074262 020102                             CMP     R1,R2          ;ARE THEY EQUAL
9326 074264 001406                             BEQ     140#            ;BR, IF OK
9327 074266 005237 002214                     INC     FATFLG           ;ERROR COUNT
9331 074272                                ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
074272 104456                                TRAP    C#ERHRD
074274 001451                                .WORD  809
074276 005107                                .WORD  WRTERR
074300 012126                                .WORD  PKTSSR
9332 074302 140#: CKLOOP                      ;LOOP IF SELECTED
074302 104406                                TRAP    C#CLP1
9333 074304 013701 075626                     MOV     T36CNT,R1       ;GET FIRST COUNTER
9334 074310 013702 075630                     MOV     T36CNU,R2       ;GET SECOND COUNTER
9335 074314 020102                             CMP     R1,R2          ;25-APR-83 REV B - COMPARE EM
9336 074316 003406                             BLE     300#            ;BR, IF VALUES ARE CORRECT (OK)
9337 074320 005237 002214                     INC     FATFLG           ;ERROR COUNT

```



TEST 8: RECORD BUFFERING

```

9341 074324          ERRHRD  ERRNO,T36NAS,EXPREC      ;TAPE NOT AT CORRECT SPEED
      074324      104456          TRAP              C#ERHRD
      074326      001452          .WORD              810
      074330      075634          .WORD              T36NAS
      074332      015554          .WORD              EXPREC
9342 074334          300#:  CKLOOP                      ;LOOP IF SELECTED          TRAP              C#CLP1
      074334      104406          ENDSUB
9343 074336          L10071: TRAP              C#ESUB
      074336      104403
9344 074340      023727  002214  000017  CMP          FATFLG,#15.      ;IS ERROR COUNT AT 25
9345 074346      103402          BLO          999#            ;BR, IF LESS THAN 25
9346 074350      004737  017262          JSR          PC,CKDROP      ;TRY TO DROP THE UNIT
9347 074354          999#:
9348          ;*
9349          ;
9350          ;TEST 8, SUBTEST 2
9351          ;
9352          ;
9353          ; THIS TEST VERIFIES THAT RECORD BUFFERING, USED FOR WRITE DATA
9354          ; AND READ NEXT COMMANDS, OPERATES PROPERLY AND IS PROPERLY
9355          ; CONTROLLED BY THE EXTENDED CHARACTERISTICS DATA WORD. IF THE
9356          ; M7196 CONTROLLER MODULE IS NOT ALREADY IN EXTENDED FEATURES MODE
9357          ; (AS CONTROLLED BY THE DIP SWITCH ON THE MODULE), IT IS PLACED
9358          ; INTO THAT MODE BY INVERTING THE SENSE OF THE SWITCH USING THE
9359          ; WRITE SUBSYSTEM MEMORY COMMAND. NOTE THAT RECORD BUFFERING HAS
9360          ; BEEN ENABLED IN PREVIOUS TESTS OF READ AND WRITE AND SO HAS BEEN
9361          ; PARTIALLY TESTED ALREADY. THIS TEST VERIFIES THAT BUFFERING IS
9362          ; ACTUALLY OPERATING. THE FOLLOWING SUBTESTS ARE PERFORMED:
9363          ;
9364          ; VERIFIES THAT NORMAL BUFFERING ON WRITE DATA COMMANDS OPERATES
9365          ; PROPERLY AT LOW TAPE SPEED. THE FOLLOWING SEQUENCE IS
9366          ; PERFORMED:
9367          ;
9368          ; 1. THE TAPE IS REWOUND.
9369          ;
9370          ; 2. BUFFERING IS DISABLED AND LOW TAPE SPEED IS SELECTED
9371          ; (VIA WRITE CHARACTERISTICS COMMAND).
9372          ;
9373          ; 3. AN INITIAL RECORD IS WRITTEN ONTO THE TAPE IN ORDER TO
9374          ; MOVE THE TAPE OFF BOT.
9375          ;
9376          ; 4. THE PROGRAM DELAYS FOR A TIME SUFFICIENT TO ALLOW THE
9377          ; TAPE TO REPOSITION AND COME TO REST.
9378          ;
9379          ; 5. A WRITE DATA COMMAND, WITH A BYTE COUNT LESS THAN 3.5K,
9380          ; IS ISSUED, AND THE PROGRAM COUNTS, IN A WAIT LOOP, THE
9381          ; TIME IT TAKES TO RECEIVE COMMAND TERMINATION. THIS
9382          ; SHOULD BE A RELATIVELY LONG TIME SINCE BUFFERING IS
9383          ; DISABLED.
9384          ;
9385          ; 6. BUFFERING IS ENABLED.
9386          ;
9387          ; 7. THE WRITE DATA COMMAND IS AGAIN ISSUED, WITH THE SAME
9388          ; BYTE COUNT AS THAT USED PREVIOUSLY. THE TIME TO
9389          ; COMPLETION IS AGAIN MEASURED.
9390          ;

```



TEST 8: RECORD BUFFERING

```

9435 074544 005237 002214          INC    FATFLG          ;ERROR COUNT
9439 074550          ERRHRD  ERRNO,T36RWN,PKTSSR ;REWIND NOT ACCEPTED
          074550 104456          TRAP    C#ERHRD
          074552 001455          .WORD  813
          074554 077041          .WORD  T36RWN
          074556 012126          .WORD  PKTSSR
9440 074560          30$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C#CLP1
          074560 104406          ;PICK UP XSTO
9441 074562 013701 075510          MOV    T36BFR+6,R1      ;SET UP EXPECTED
9442 074566 010102          MOV    R1,R2            ;SET BOT BIT IN EXPECTED
9443 074570 052702 000002          BIS    #BIT1,R2        ;DOES EXP = REC'D
9444 074574 020102          CMP    R1,R2            ;BR, IF EQUAL (OK)
9445 074576 001406          BEQ    40$              ;ERROR COUNT
9446 074600 005237 002214          INC    FATFLG          ;TAPE NOT AT BOT AFTER REWIND
9450 074604          ERRHRD  ERRNO,T36BOT,EXPREC ;
          074604 104456          TRAP    C#ERHRD
          074606 001456          .WORD  814
          074610 076535          .WORD  T36BOT
          074612 C15554          .WORD  EXPREC
9451 074614          40$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C#CLP1
          074614 104406          ;SET UP DRIVE NUMBER
9452 074616 013737 002174 075500          MOV    UNITN,T36DSW     ;25-APR-83 REV B - TURN ON THE BUFFERING
9453 074624 052737 000030 075500          BIS    #BIT3!BIT4,T36DSW ;SUBROUTINE NEEDS PACKET ADDRESS
9454 074632 012704 075460          MOV    #T36PACKET,R4   ;ISSUE WRITE CHARACTERISTICS
9455 074636 004737 010742          JSR    PC,WRTCHR        ;BR, IF COMMAND ISSUED OK
9456 074642 103407          BCS    50$              ;ERROR COUNT
9457 074644 005237 002214          INC    FATFLG          ;SAVE CONTENTS OF TSSR
9461 074650 010001          MOV    R0,R1            ;WRITE CHARACTERISTIC FAILED
9462 074652          ERRHRD  ERRNO,WRTMSG,SFIMSG ;
          074652 104456          TRAP    C#ERHRD
          074654 001457          .WORD  815
          074656 005052          .WORD  WRTMSG
          074660 012114          .WORD  SFIMSG
9463 074662          50$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C#CLP1
          074662 104406          ;SET UP RECORD SIZE
9464 074664 012737 003720 075606          MOV    #2000.,T36SZ     ;ADDRESS OF WRITE BUFFER
9465 074672 013737 003116 075602          MOV    FREE,T36WB      ;WRITE DATA,ACK,CVC=1 COMMAND
9466 074700 012737 140005 075600          MOV    #140005,T36PK3  ;SET UP R4 WITH PACKET ADDRESS
9467 074706 012704 075600          MOV    #T36PK3,R4      ;ISSUE COMMAND
9468 074712 010465 000000          MOV    R4,TSDB(R5)     ;WAIT FOR SSR TO SET
9469 074716 004737 016330          JSR    PC,WAITF        ;GET TSSR CONTENTS
9470 074722 016501 000002          MOV    TSSR(R5),R1     ;SET UP EXPECTED
9471 074726 012702 000200          MOV    #SSR,R2         ;ARE THEY EQUAL
9472 074732 020102          CMP    R1,R2            ;BR, IF OK
9473 074734 001406          BEQ    60$              ;ERROR COUNT
9474 074736 005237 002214          INC    FATFLG          ;TSSR INCORRECT AFTER READ DATA
9478 074742          ERRHRD  ERRNO,WRTERR,PKTSSR ;
          074742 104456          TRAP    C#ERHRD
          074744 001460          .WORD  816
          074746 005107          .WORD  WRTERR
          074750 012126          .WORD  PKTSSR
9479 074752          60$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C#CLP1
          074752 104406          ;25-APR-83 REV B - DELAY FOR TAPE TO STOP
9480 074754 012737 000005 075632          MOV    #05.,T36DLY     ;25-APR-83 REV B - DELAY ROUTINE CALL
9481 074762          70$:  DELAY    1
          074762 012727 000001          MOV    #1,(PC)+
          074766 000000          .WORD  0
    
```

TEST 8: RECORD BUFFERING

074770	013727	002116				MOV	L#DLY,(PC)+
074774	000000					.WORD	0
074776	005367	177772				DEC	-6(PC)
075002	001375					BNE	.-4
075004	005367	177756				DEC	-22(PC)
075010	001367					BNE	.-20
9482	075012	005337	075632		DEC	T36DLY	;BUMP COUNTER DOWN
9483	075016	001361			BNE	70#	;BR, IF MORE DELAY TO GO
9484	075020	012737	006642	075606	MOV	#3490.,T36SZ	;SET SIZE OF TRANSFER
9485	075026	012737	140005	075600	MOV	#140005,T36PK3	;WRITE DATA,ACK,CVC=1 COMMAND
9486	075034	012704	075600		MOV	#T36PK3,R4	;SET UP R4 WITH PACKET ADDRESS
9487	075040	005037	075626		CLR	T36CNT	;CLEAR COUNTER
9488	075044	012737	001750	075632	MOV	#1000.,T36DLY	;SET DROP DEAD COUNTER VALUE
9489	075052	010465	000000		MOV	R4,T36DLY	;ISSUE COMMAND
9490	075056	016501	000002		MOV	T36CNT,R1	;GET T36CNT CONTENTS
9491	075062	032701	000200	80#:	BIT	#SSR,R1	;CHECK FOR SSR SET
9492	075066	001021			BNE	90#	;BR, IF SSR IS SET
9493	075070	005237	075626		INC	T36CNT	;BUMP CYCLE COUNTER
9494	075074				DELAY	1	;CUT NUMBER OF LOOPS DOWN
	075074	012727	000001			MOV	#1,(PC)+
	075100	000000				.WORD	0
	075102	013727	002116			MOV	L#DLY,(PC)+
	075106	000000				.WORD	0
	075110	005367	177772			DEC	-6(PC)
	075114	001375				BNE	.-4
	075116	005367	177756			DEC	-22(PC)
	075122	001367				BNE	.-20
9495	075124	005337	075632		DEC	T36DLY	;BUMP DROP DEAD COUNTER
9496	075130	001352			BNE	80#	;BR, IF THERE IS STILL TIME
9497	075132	012702	000200	90#:	MOV	#SSR,R2	;SET UP EXPECTED
9498	075136	020102			CMP	R1,R2	;ARE THEY EQUAL
9499	075140	001406			BEQ	100#	;BR, IF OK
9500	075142	005237	002214		INC	FATFLG	;ERROR COUNT
9504	075146				ERRHRD	ERRNO,T36WDE,PKTSSR	;TSSR INCORRECT AFTER READ DATA
	075146	104456				TRAP	C#ERHRD
	075150	001461				.WORD	817
	075152	076463				.WORD	T36WDE
	075154	012126				.WORD	PKTSSR
9505	075156			100#:	CKLOOP		;LOOP IF SELECTED
	075156	104406				TRAP	C#CLP1
9506	075160	013737	002174	075500	MOV	UNITN,T36DSW	;SET UP DRIVE NUMBER
9507	075166	052737	000010	075500	BIS	#BIT3,T36DSW	;25-APR-83 REV B - TURN OFF BUFFERING
9508	075174	012704	075460		MOV	#T36PACKET,R4	;SUBROUTINE NEEDS PACKET ADDRESS
9509	075200	004737	010742		JSR	PC,WRTCHR	;ISSUE WRITE CHARACTERISTICS
9510	075204	103407			BCS	110#	;BR, IF COMMAND ISSUED OK
9511	075206	005237	002214		INC	FATFLG	;ERROR COUNT
9515	075212	010001			MOV	R0,R1	;SAVE CONTENTS OF TSSR
9516	075214				ERRHRD	ERRNO,WRTMSG,SFIMSG	;WRITE CHARACTERISTICS FAILED
	075214	104456				TRAP	C#ERHRD
	075216	001462				.WORD	818
	075220	005052				.WORD	WRTMSG
	075222	012114				.WORD	SFIMSG
9517	075224			110#:	CKLOOP		;LOOP IF SELECTED
	075224	104406				TRAP	C#CLP1
9518	075226	012737	006642	075606	MOV	#3490.,T36SZ	;SET SIZE OF TRANSFER
9519	075234	012737	140005	075600	MOV	#140005,T36PK3	;WRITE DATA,ACK,CVC=1 COMMAND
9520	075242	012704	075600		MOV	#T36PK3,R4	;SET UP R4 WITH PACKET ADDRESS

TEST 8: RECORD BUFFERING

```

9521 075246 005037 075630          CLR      T36CNU           ;CLEAR COUNTER
9522 075252 012737 001750 075632  MOV      #1000.,T36DLY   ;SET DROP DEAD COUNTER VALUE
9523 075260 010465 000000          MOV      R4,TSDB(R5)     ;ISSUE COMMAND
9524 075264 016501 000002          MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
9525 075270 032701 000200 120$:  BIT      #SSR,R1         ;CHECK FOR SSR SET
9526 075274 001021          BNE     130$             ;BR, IF SSR IS SET
9527 075276 005237 075630          INC      T36CNU         ;BUMP CYCLE COUNTER
9528 075302          DELAY  1                 ;CUT NUMBER OF LOOPS DOWN
                                MOV      #1.(PC)+
                                .WORD   0
                                MOV      L#DLY,(PC)+
                                .WORD   0
                                DEC      -6(PC)
                                BNE     .-4
                                DEC      -22(PC)
                                BNE     .-20
                                075302 012727 000001
                                075306 000000
                                075310 013727 002116
                                075314 000000
                                075316 005367 177772
                                075322 001375
                                075324 005367 177756
                                075330 001367
9529 075332 005337 075632          DEC      T36DLY         ;BUMP DROP DEAD COUNTER
9530 075336 001352          BNE     120$            ;BR, IF THERE IS STILL TIME
9531 075340 C12702 000200 130$:  MOV      #SSR,R2        ;SET UP EXPECTED
9532 075344 020102          CMP      R1,R2         ;ARE THEY EQUAL
9533 075346 001406          BEQ     140$            ;BR, IF OK
9534 075350 005237 002214          INC      FATFLG         ;ERROR COUNT
9538 075354          ERRHRD  ERRNO,WRterr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP   C#ERHRD
                                .WORD  819
                                .WORD  WRterr
                                .WORD  PKTSSR
                                075354 104456
                                075356 001463
                                075360 005107
                                075362 012126
9539 075364          140$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP   C#CLP1
                                075364 104406
9540 075366 013701 075626          MOV      T36CNT,R1     ;GET FIRST COUNTER
9541 075372 013702 075630          MOV      T36CNU,R2    ;GET SECOND COUNTER
9542 075376 020102          CMP      R1,R2         ;25-APR-83 REV B - COMPARE EM
9543 075400 003406          BLE     300$            ;BR, IF VALUES ARE CORRECT (OK)
9544 075402 005237 002214          INC      FATFLG         ;ERROR COUNT
9548 075406          ERRHRD  ERRNO,T36NAS,EXPREC ;TAPE NOT AT CORRECT SPEED
                                TRAP   C#ERHRD
                                .WORD  820
                                .WORD  T36NAS
                                .WORD  EXPREC
                                075406 104456
                                075410 001464
                                075412 075634
                                075414 015554
9549 075416          300$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP   C#CLP1
                                075416 104406
9550 075420          ENDSUB
                                L10072:
                                TRAP   C#ESUB
                                075420 104403
9551 075422 023727 002214 000017  CMP      FATFLG,#15.    ;IS ERROR COUNT AT 25
9552 075430 103402          BLO     999$           ;BR, IF LESS THAN 25
9553 075432 004737 017262          JSR     PC,CKDROP     ;TRY TO DROP THE UNIT
9554 075436          999$:
9555          ;
9556          ;
9557          ;
9558 075436 004737 016536          JSR     PC,TSTLOOP    ;DO WE NEED TO ITERATE TEST
9559 075442 103002          BCC     163$           ;BR, IF NO LOOP REQUIRED
9560 075444 000137 073300          JMP     T36LOOP       ;EXECUTE AGAIN
9561 075450          163$:
9562 075450          EXIT   TST         ;ALL DONE THIS TEST
                                TRAP   C#EXIT
                                075450 104432
    
```

TEST 8: RECORD BUFFERING

```

075452 003344 .WORD L10070-.
9563
9564 ;*
9565 ;LOCAL STORAGE FOR THIS TEST
9566 ;-
9567 075460 075460 .=<.10>&177770
9569 075460 T36PACKET: ;COMMAND PACKET FOR TEST
9570 075460 100004 .WORD 100004 ;WRITE CHARACTERISTICS COMMAND, WITH . ACK
9571 075462 075470 .WORD T36DATA ;ADDRESS OF CHARACTERISTICS BLOCK
9572 075464 000000 .WORD 0
9573 075466 000012 .WORD 10. ;STARTING VALUE OF BLOCK SIZE
9574 075470 T36DATA: ;CHARACTERISTICS DATA BLOCK
9575 075470 075502 .WORD T36BFR ;ADDRESS OF MESSAGE BUFFER
9576 075472 000000 .WORD 0
9577 075474 000024 .WORD 20. ;LENGTH OF MESSAGE BUFFER
9578 075476 000000 .WORD 0
9579 075500 000000 T36DSW: .WORD 0 ;SELECT DRIVE 0
9580 075502 T36BFR: .BLKW 25. ;MESSAGE BUFFER
9581 ;
9582 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
9583 ;
9585 075570 075570 .=<.10>&177770
9587 075570 T36PK2: ;WRITE SUB SYS MEM COMMAND, AND ACK
9588 075570 100006 .WORD 100006 ;ADDRESS OF SELECT BLOCK DATA
9589 075572 075610 .WORD T36BF2
9590 075574 000000 .WORD 0 ;SIZE OF DATA PACKET
9591 075576 000006 .WORD 6.
9592
9596 075600 T36PK3: ;REREAD COMMAND, AND ACK
9597 075600 100005 .WORD 100005
9598 075602 T36RB: ;ADDRESS OF WRITE BUFFER
9599 075602 003116 T36WB: .WORD FREE
9600 075604 000000 .WORD 0
9601 075606 000000 T36SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
9602 .EVEN
9603 ;
9604 ;
9605 ;
9606 075610 T36BF2:
9607 075610 010 T36BS0: .BYTE 10 ;BSELO AREA
9608 075611 200 T36BS1: .BYTE 200 ;BSEL1 AREA
9609 075612 000000 T36S2: .WORD 0 ;SEL 2 AREA
9610 075614 000000 T36S3: .WORD 0 ;DATA AREA
9611 ;
9612 ;
9613 .EVEN
9614 ;TAPE MOTION PACKET COMMAND VALUES
9615
9616 075616 100205 T36RN: .WORD 100205 ;REREAD DATA (NEXT)
9617 075620 100605 T36WDR: .WORD 100605 ;REREAD DATA RETRY
9618 075622 102205 T36CON: .WORD 102205 ;WRITE CONTINOUS
9619 075624 177777 .WORD 177777 ;END OF DATA
9620
9621 ;
9622 075626 000000 T36CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
9623 075630 000000 T36CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
9624 075632 000000 T36DLY: .WORD 0 ;DELAY COUNTER
9625 ;*
    
```

TEST 8: RECORD BUFFERING

```

9626          ;LOCAL TEXT MESSAGES FOR TEST
9627          ;-
9628
9629 075634    111    155    160  T36NAS: .ASCIZ  'Improper Tape Controller Buffering Speed'
9630 075705    124    141    160  T36MNG: .ASCIZ  'Tape Position Incorrect After REREAD Previous (OPP=1)'
9631 075773    124    123    123  T36RDF: .ASCIZ  'TSSR Incorrect After READ DATA Command'
9632 076042    122    105    122  T36RRF: .ASCIZ  'REREAD Previous (Space Reverse, Read Forward) Command Failed'
9633 076137    120    117    123  T36SC: .ASCIZ   'POSITION (Space Command) Failed, TSSR Not Correct'
9634 076221    122    111    102  T36LOR: .ASCIZ  'RIB NOT SET AFTER READ REVERSE INTO BOT'
9635 076271    124    123    123  T36WDF: .ASCIZ  'TSSR Not Correct After Illegal Mode Bits Set'
9636 076346    111    154    154  T36LOQ: .ASCIZ  'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
9637 076427    122    105    122  T36SSR: .ASCIZ  'REREAD COMMAND Not Accepted'
9638 076463    124    123    123  T36WDE: .ASCIZ  'TSSR Not Correct After WRITE DATA Command'
9639 076535    124    141    160  T36BOT: .ASCIZ  'Tape Not At BOT After REWIND Command (BOT Not Set In XSTO)'
9640 076630    127    122    111  T36TIM: .ASCIZ  'WRITE DATA RETRY'S Erase Tape Not Long Enough'
9641 076705    122    105    122  T36EOT: .ASCIZ  'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
9642 076764    124    123    123  T36TM: .ASCIZ   'TSSR Not Correct After REREAD COMMAND Reject'
9643 077041    122    145    167  T36RWN: .ASCIZ  'Rewind (POSITION) Command Not Accepted'
9644 077110    122    101    115  T36RNC: .ASCIZ  'RAM Error, Correct Data Pattern Not In Ram'
9645 077163    124    123    123  T36AM3: .ASCIZ  'TSSR Init. Failed After REREAD COMMAND'
9646 077232    104    162    151  T36OFL: .ASCIZ  'Drive 7 Select Failed To Set "OFL" In TSSR'
9647 077305    124    123    123  T36WDD: .ASCIZ  'TSSR Not Correct After REREAD DATA Command, SMB Bit Set'
9648 077375    124    123    123  T36WDC: .ASCIZ  'TSSR Not Correct After REREAD DATA Command'
9649 077450    103    126    103  T36VCK: .ASCIZ  'CVC Set, Didn't Reset VCK In Message Buffer'
9650 077523    124    123    102  T36BA: .ASCIZ   'TSBA Not Correct After REREAD DATA Command'
9651 077576    127    122    111  T36WSS: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
9652 077665    122    145    141  T36LON: .ASCIZ  'Reading Long Record Failed To Set RLL Bit In XSTO'
9653 077747    122    145    141  T36LOP: .ASCIZ  'Reading Long Record Failed To Set RLS Bit In XSTO'
9654 100031    122    145    163  T36PBP: .ASCIZ  'Residual Byte Count Incorrect After Short Record Read'
9655 100117    122    145    141  T36TRL: .ASCIZ  'Reading Long Record Failed To Give Tape Status Alert'
9656 100205    127    122    111  T36NEF: .ASCIZ  'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
9657 100303    124    123    123  T36SCF: .ASCIZ  'TSSR Not Correct After SPACE RECORDS Command'
9658 100360    124    123    123  T36TSA: .ASCIZ  'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
9659 100442    124    123    123  T36WRF: .ASCIZ  'TSSR Not Correct After WRITE DATA RETRY Command'
9660 100522    104    141    164  T36DTA: .ASCIZ  'Data Compare Error, Data Read From Tape Not Equal To Written'
9661 100617    122    145    143  TST36ID: .ASCIZ  'Record Buffering'
9662          .EVEN
9663          ;+
9664          ;
9665          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
9666          ;WRITE SUBSYSTEM MEMORY COMMAND
9667          ;
9668          ;-
9669
9670 100640
9671 100640
9672 100644    012701    075460
9673 100650    012721    100004
9674 100654    012721    075470
9675 100660    005021
9676 100662    012721    000012
9677 100666    012721    075502
9678 100672    005021
9679 100674    012721    000024
9680 100700    005021
9681 100702    012711    000000
9682 100706    012702    000030

          T36REST:
          SAVREG
          MOV    #T36PACKET,R1          ;SAVE THE REGISTERS
          MOV    #100004,(R1)+        ;START OF THE PACKET
          MOV    #T36DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,
          CLR    (R1)+                ;ADDRESS OF CHARAISTICS DATA BLOCK
          MOV    #10.,(R1)+          ;EXTENDED ADDRESS
          MOV    #T36BFR,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
          CLR    (R1)+                ;ADDRESS OF MESSAGE BUFFER
          MOV    #20.,(R1)+          ;LENGTH OF MESSAGE BUFFER
          CLR    (R1)+
          MOV    #0,(R1)              ;SELECT DRIVE ZERO
          MOV    #24.,R2              ;NUMBER OF LOCATIONS TO BE CLEARED
    
```

TEST 8: RECORD BUFFERING

```

9683 100712 012762 177777 075502 644:  MOV    #177777,T36BFR(R2)    ;ALL ONES TO MESSAGE BUFFER
9684 100720 005742                    TST    -(R2)                ;NEXT LOCATION
9685 100722 022702 000000            CMP    #0,R2                ;AT END OF LOOP YET
9686 100726 001371                    BNE    644                  ;KEEP GOING UNTIL DONE
9687 100730 000207                    RTS    PC                    ;RETURN
9688
9689 100732                    T36RT2:
9690 100732                    SAVREG                      ;SAVE THE REGISTERS
9691 100736 012701 075570            MOV    #T36PK2,R1           ;START OF THE PACKET
9692 100742 012721 100006            MOV    #100006,(R1)+        ;WRITE SUBSYSTEM MEM. WITH ACK.
9693 100746 012721 075610            MOV    #T36BF2,(R1)+       ;ADDRESS OF DATA BLOCK
9694 100752 005021                    CLR    (R1)+                ;EXTENDED ADDRESS
9695 100754 012721 000006            MOV    #6,(R1)+            ;SIZE OF DATA BLOCK IN BYTES
9696 100760 005021                    CLR    (R1)+
9697 100762 012701 075610            MOV    #T36BF2,R1          ;POINT TO DATA SEL AREA
9698 100766 005021                    CLR    (R1)+
9699 100770 005011                    CLR    (R1)
9700 100772 000207                    RTS    PC                    ;RETURN
9701 100774                    T36RT3:
9702 100774                    SAVREG                      ;SAVE REGISTERS
9703 101000 012701 075600            MOV    #T36PK3,R1           ;SET UP POINTER ADDRESS
9704 101004 005021                    CLR    (R1)+                ;COMMAND SPACE
9705 101006 005021                    CLR    (R1)+                ;ADDRESS OF DATA BLOCK
9706 101010 005021                    CLR    (R1)+                ;EXTENDED ADDRESS
9707 101012 005011                    CLR    (R1)                ;SIZE OF DATA TRANSFER BLOCK
9708 101014 000207                    RTS    PC                    ;RETURN
9709 101016                    ENDTST
101016 104401                    L10070: TRAP    C#ETST

```

9710 .SBTTL TEST 9: FUNCTION TIMING

```

9711 ;*
9712 ;
9713 ;THIS TEST VERIFIES THAT THE TAPE TRANSPORT SEEMS TO BE WRITING
9714 ;RECORDS, GAPS, AND EXTENDED GAPS OF THE PROPER LENGTH. BOTH LOW
9715 ;AND HIGH SPEED MODES ARE TESTED. IT IS ALSO VERIFIED THAT A
9716 ;SPACE RECORDS COMMAND WITH A RECORD COUNT OF 80 OR MORE, AND A
9717 ;SKIP TAPE MARKS COMMAND WITH A COUNT OF 2 OF MORE, OPERATE THE
9718 ;TAPE IN HIGH-SPEED MODE. THIS TEST CAN ONLY BE RUN IF A
9719 ;REAL-TIME CLOCK IS AVAILIABLE ON THE SYSTEM. THE TEST OPERATES BY
9720 ;TIMING VARIOUS TAPE-MOTION OPERATIONS, USING A NUMBER OF
9721 ;DIFFERENT TEST RECORD LENGTHS.
9722 ;
9723 ;
9724 ;-

```

```

9725 101020                    BGNTST
9726 101020 012737 006354 002172            MOV    #EPRT1,EPRTSW       ;PRIMARY ERROR MESSAGE
9727 101026 004737 017354                    JSR    PC,KTOFF             ;TURN KT OFF
9728 101032 012700 105243                    MOV    #TST37ID,R0         ;ASCII MESSAGE TO IDENTIFY TEST
9729 101036 004737 016570                    JSR    PC,TSTSETUP         ;DO INITIAL TEST SETUP
9730 101042 012737 000005 002210            MOV    #5,LOOPCNT         ;PERFORM 5 ITERATIONS
9731 101050 005037 102306                    CLR    T37CNT              ;CLEAR TAPE RECORD COUNTER

```

```

9736 ;*
9737 ;
9738 ;TEST 9, SUBTEST 1
9739 ;
9740 ;

```





TEST 9: FUNCTION TIMING

```

101262 103465 .WORD T37RWN
101264 012126 .WORD PKTSSR
9786 101266 30$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
101266 104406
9787 101270 013701 102170 MOV T37BFR+6,R1 ;PICK UP XSTO
9788 101274 010102 MOV R1,R2 ;SET UP EXPECTED
9789 101276 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
9790 101302 020102 CMP R1,R2 ;DOES EXP = REC'D
9791 101304 001406 BEQ 40$ ;BR, IF EQUAL (OK)
9792 101306 005237 002214 INC FATFLG ;ERROR COUNT
9796 101312 ERRHRD ERRNO,T37BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
101312 104456 TRAP C#ERHRD
101314 001610 .WORD 904
101316 103161 .WORD T37BOT
101320 015554 .WORD EXPREC
9797 101322 40$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
101322 104406 ;NUMBER OF RECORDS TO BE WRITTEN
9798 101324 012703 000144 MOV #100.,R3 ;STARTING WRITE BUFFER ADDRESS
9799 101330 C13737 003116 102262 MOV FREE,T37WB ;WRITE DATA,ACK,CVC=1 COMMAND
9800 101336 012737 140005 102260 65$: MOV #140005,T37PK3 ;SET UP R4 WITH PACKET ADDRESS
9801 101344 012704 102260 MOV #T37PK3,R4 ;SET UP RECORD SIZE IN PACKET
9802 101350 012737 001130 102266 MOV #600.,T37SZ ;ISSUE COMMAND
9803 101356 010465 000000 MOV R4,T37DB(R5) ;WAIT FOR SSR TO SET
9804 101362 004737 016330 JSR PC,WAITF ;GET TSSR CONTENTS
9805 101366 016501 000002 MOV TSSR(R5),R1 ;SET UP EXPECTED
9806 101372 012702 000200 MOV #SSR,R2 ;ARE THEY EQUAL
9807 101376 020102 CMP R1,R2 ;BR, IF OK
9808 101400 001406 BEQ 70$ ;ERROR COUNT
9809 101402 005237 002214 INC FATFLG ;TSSR INCORRECT AFTER WRITE DATA
9813 101406 ERRHRD ERRNO,T37WDC,PKTSSR TRAP C#ERHRD
101406 104456 .WORD 905
101410 001611 .WORD T37WDC
101412 104021 .WORD PKTSSR
101414 012126
9814 101416 70$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
101416 104406 ;DEC RECORD COUNTER
9815 101420 005303 DEC R3 ;BR, IF MORE RECORDS TO WRITE
9816 101422 001345 BNE 65$ ;CALL TAPE REWIND COMMAND
9817 101424 004737 011074 JSR PC,REWIND ;BR, IF NO PROBLEM
9818 101430 103411 BCS 130$ ;GET TSSR CONTENTS
9819 101432 016501 000002 MOV TSSR(R5),R1 ;GET PACKET ADDRESS
9820 101436 010004 MOV R0,R4 ;ERROR COUNT
9821 101440 005237 002214 INC FATFLG ;REWIND NOT ACCEPTED
9825 101444 ERRHRD ERRNO,T37RWN,PKTSSR TRAP C#ERHRD
101444 104456 .WORD 906
101446 001612 .WORD T37RWN
101450 103465 .WORD PKTSSR
101452 012126
9826 101454 130$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
101454 104406 ;PICK UP XSTO
9827 101456 013701 102170 MOV T37BFR+6,R1 ;SET UP EXPECTED
9828 101462 010102 MOV R1,R2 ;SET BOT BIT IN EXPECTED
9829 101464 052702 000002 BIS #BIT1,R2 ;DOES EXP = REC'D
9830 101470 020102 CMP R1,R2 ;BR, IF EQUAL (OK)
9831 101472 001406 BEQ 140$ ;ERROR COUNT
9832 101474 005237 002214 INC FATFLG ;TAPE NOT AT BOT AFTER REWIND
9836 101500 ERRHRD ERRNO,T37BOT,EXPREC

```



## TEST 9: FUNCTION TIMING

```

9876 101716      104406      175$: CKLOOP                ;LOOP IF SELECTED                TRAP      C#CLP1
      101716      104406
9877 101720      012704      102260      MOV      #T37PK3,R4            ;SET UP PACKET ADDRESS
9878 101724      012737      000121      102262      MOV      #81.,T37RB           ;SET UP RECORDS TO SPACE OVER
9879 101732      012737      140010      102260      MOV      #140010,T37PK3       ;ACK,CVC=1,SPACE FORWARD COMMAND
9880 101740      010465      000000      250$:  MOV      R4,TSDB(R5)       ;ISSUE COMMAND
9881 101744      005237      102310      252$:  INC      T37CNU              ;BUMP TIMER
9882 101750      012727      000001      DELAY    1                    ;DELAY ABOUT 100US
      101754      000000
      101756      013727      002116      MOV      #1.(PC),
      101762      000000
      101764      005367      177772      .WORD   0
      101770      001375
      101772      005367      177756      MOV      L#DLY,(PC),
      101776      001367
      .WORD   0
      9883 102000      016501      000002      DEC     -6(PC)
      9884 102004      032701      000200      BNE     -.4
      9885 102010      001755
      9886 102012      012702      000200      DEC     -22(PC)
      9887 102016      020201
      9888 102020      001406
      9889 102022      005237      002214      BNE     .-20
      9893 102026      104456
      102026      001617
      102030      104727
      102032      012126
      9894 102036      104406      260$: CKLOOP                ;LOOP IF SELECTED                TRAP      C#ERHRD
      102036      104406
      102040      013701      102306      MOV      T37CNT,R1            ;GET TSSR
      9895 102044      013702      102310      MOV      T37CNU,R2            ;CHECK FOR TSSR'S SSR SET
      9896 102044      042701      000077      BIC     #000077,R1            ;KEEP COUNTING UNTIL SET
      9897 102050      042702      000077      BIC     #000077,R2            ;SET UP EXPECTED
      9898 102054      020102      000077      CMP     R1,R2                  ;SET UP EXPECTED
      9899 102060      003406      002214      BLE     300$                   ;WAS EVERYTHING OK
      9900 102062      005237      002214      INC     FATFLG                  ;BR, IF ALL IS WELL
      9901 102064      005237      002214      ERRHRD ERRNO,T37SCF,PKTSSR    ;BR, IF ALL IS WELL
      9905 102070      104456
      102070      001620
      102072      103254
      102074      015554
      9906 102100      104406      300$: CKLOOP                ;LOOP IF SELECTED                TRAP      C#ERHRD
      102100      104406
      102102      104403
      102102      023727      002214      000017      MOV      FATFLG,#15.          ;SPACE FORWARD DIDN'T WORK OUT
      9908 102104      103402
      9909 102112      004737      017262      JSR     PC,CKDROP              ;SPACE FORWARD DIDN'T WORK OUT
      9910 102114
      9911 102120
      9912
      9913
      9914
      9915 102120      004737      016536      JSR     PC,TSTLOOP            ;SPACE FORWARD DIDN'T WORK OUT
      9916 102124      103002
      9917 102126      000137      101054      BCC     163$
      JMP     T37LOOP              ;SPACE FORWARD DIDN'T WORK OUT

```

TEST 9: FUNCTION TIMING

```

9918 102132
9919 102132
      102132 104432
      102134 003306
9920
9921
9922
9924      102140
9926 102140
9927 102140 100004
9928 102142 102150
9929 102144 000000
9930 102146 000012
9931 102150
9932 102150 102162
9933 102152 000000
9934 102154 000024
9935 102156 000000
9936 102160 000000
9937 102162
9938
9939
9940
9942      102250
9944 102250
9945 102250 100006
9946 102252 102270
9947 102254 000000
9948 102256 000006
9949
9953 102260
9954 102260 100005
9955 102262
9956 102262 003116
9957 102264 000000
9958 102266 000000
9959
9960
9961
9962
9963 102270
9964 102270      010
9965 102271      200
9966 102272 000000
9967 102274 000000
9968
9969
9970
9971
9972
9973 102276 100205
9974 102300 100605
9975 102302 102205
9976 102304 177777
9977
9978
9979 102306 000000

1634:  EXIT   TST                ;ALL DONE THIS TEST
                                           TRAP   CEXIT
                                           .WORD  L10073-.

;*
;LOCAL STORAGE FOR THIS TEST
;-
;= < . * 10 > & 177770
T37PACKET:
      .WORD 100004                ;COMMAND PACKET FOR TEST
      .WORD T37DATA              ;WRITE CHARACTERISTICS COMMAND, WITH . ACK
      .WORD 0                    ;ADDRESS OF CHARACTERISTICS BLOCK
      .WORD 10.                 ;STARTING VALUE OF BLOCK SIZE
T37DATA:
      .WORD T37BFR              ;CHARACTERISTICS DATA BLOCK
      .WORD 0                   ;ADDRESS OF MESSAGE BUFFER
      .WORD 20.                ;LENGTH OF MESSAGE BUFFER
T37DSW: .WORD 0                 ;SELECT DRIVE 0
T37BFR: .BLKW 25.              ;MESSAGE BUFFER

;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
;= < . * 10 > & 177770
T37PK2:
      .WORD 100006              ;WRITE SUB SYS MEM COMMAND, AND ACK
      .WORD T37BF2             ;ADDRESS OF SELECT BLOCK DATA
      .WORD 0
      .WORD 6.                 ;SIZE OF DATA PACKET
T37PK3:
      .WORD 100005              ;REREAD COMMAND, AND ACK
T37RB:
T37WB: .WORD FREE              ;ADDRESS OF WRITE BUFFER
      .WORD 0
T37SZ: .WORD 0                 ;SIZE OF BUFFER (EXTENT)
      .EVEN
;
;
;T37BF2:
T37BS0: .BYTE 10                ;BSELO AREA
T37BS1: .BYTE 200              ;BSEL1 AREA
T37S2: .WORD 0                 ;SEL 2 AREA
T37S3: .WORD 0                 ;DATA AREA
;
;
      .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T37RN: .WORD 100205            ;REREAD DATA (NEXT)
T37WR: .WORD 100605            ;REREAD DATA RETRY
T37CON: .WORD 102205           ;WRITE CONTINOUS
      .WORD 177777             ;END OF DATA
;
;T37CNT: .WORD 0              ;TAPE TIMER COUNTER STORAGE AREA

```

TEST 9: FUNCTION TIMING

9980	102310	000000				T37CNU: .WORD	0		;TAPE TIMER COUNTER STORAGE AREA
9981	102312	000000				T37DLY: .WORD	0		;DELAY COUNTER
9982						;			
9983						;			
9984						;			
9985						;			
9986	102314	124	141	160		T37WNG: .ASCIZ			'Tape Position Incorrect After REREAD Previous (OPP=1)'
9987	102402	124	123	123		T37RDF: .ASCIZ			'TSSR Incorrect After READ DATA Command'
9988	102451	122	105	122		T37RRF: .ASCIZ			'REREAD Previous (Space Reverse, Read Forward) Command Failed'
9989	102546	120	117	123		T37SC: .ASCIZ			'POSITION (Space Command) Failed, TSSR Not Correct'
9990	102630	122	111	102		T37LOR: .ASCIZ			'RIB NOT SET AFTER READ REVERSE INTO BOT'
9991	102700	124	123	123		T37WDF: .ASCIZ			'TSSR Not Correct After Illegal Mode Bits Set'
9992	102755	111	154	154		T37LOQ: .ASCIZ			'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
9993	103036	122	105	122		T37SSR: .ASCIZ			'REREAD COMMAND Not Accepted'
9994	103072	124	123	123		T37WDE: .ASCIZ			'TSSR Not Correct After WRITE DATA RETRY Command, At BOT'
9995	103161	124	141	160		T37BOT: .ASCIZ			'Tape Not At BOT After REWIND Command (BOT Not Set In XSTO)'
9996	103254	127	122	111		T37TIM: .ASCIZ			'WRITE DATA RETRY'S Erase Tape Not Long Enough'
9997	103331	122	105	122		T37EOT: .ASCIZ			'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
9998	103410	124	123	123		T37TH: .ASCIZ			'TSSR Not Correct After REREAD COMMAND Reject'
9999	103465	122	145	167		T37RMN: .ASCIZ			'Rewind (POSITION) Command Not Accepted'
10000	103534	122	101	115		T37RNC: .ASCIZ			'RAM Error, Correct Data Pattern Not In Ram'
10001	103607	124	123	123		T37AM3: .ASCIZ			'TSSR Init. Failed After REREAD COMMAND'
10002	103656	104	162	151		T37OFL: .ASCIZ			'Drive 7 Select Failed To Set "OFL" In TSSR'
10003	103731	124	123	123		T37WDD: .ASCIZ			'TSSR Not Correct After REREAD DATA Command, SMB Bit Set'
10004	104021	124	123	123		T37WDC: .ASCIZ			'TSSR Not Correct After REREAD DATA Command'
10005	104074	103	126	103		T37VCK: .ASCIZ			'CVC Set, Didn't Reset VCK In Message Buffer'
10006	104147	124	123	102		T37BA: .ASCIZ			'TSBA Not Correct After REREAD DATA Command'
10007	104222	127	122	111		T37WSS: .ASCIZ			'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
10008	104311	122	145	141		T37LON: .ASCIZ			'Reading Long Record Failed To Set RLL Bit In XSTO'
10009	104373	122	145	141		T37LOP: .ASCIZ			'Reading Long Record Failed To Set RLS Bit In XSTO'
10010	104455	122	145	163		T37P8P: .ASCIZ			'Residual Byte Count Incorrect After Short Record Read'
10011	104543	122	145	141		T37TRL: .ASCIZ			'Reading Long Record Failed To Give Tape Status Alert'
10012	104631	127	122	111		T37NEF: .ASCIZ			'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
10013	104727	124	123	123		T37SCF: .ASCIZ			'TSSR Not Correct After SPACE RECORDS Command'
10014	105004	124	123	123		T37TSA: .ASCIZ			'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
10015	105066	124	123	123		T37WRF: .ASCIZ			'TSSR Not Correct After WRITE DATA RETRY Command'
10016	105146	104	141	164		T37DTA: .ASCIZ			'Data Compare Error, Data Read From Tape Not Equal To Written'
10017	105243	106	165	156		T37ID: .ASCIZ			'Function Timing'
10018						.EVEN			
10019						;			
10020						;			
10021						;			
10022						;			
10023						;			
10024						;			
10025						;			
10026	105264					T37REST:			
10027	105264					SAVREG			;SAVE THE REGISTERS
10028	105270	012701	102140			MOV	#T37PACKET,R1		;START OF THE PACKET
10029	105274	012721	100004			MOV	#100004,(R1)+		;WRITE SUBSYSTEM MEM. WITH ACK,
10030	105300	012721	102150			MOV	#T37DATA,(R1)+		;ADDRESS OF CHARAISTICS DATA BLOCK
10031	105304	005021				CLR	(R1)+		;EXTENDED ADDRESS
10032	105306	012721	000012			MOV	#10,(R1)+		;SIZE OF DATA BLOCK IN BYTES
10033	105312	012721	102162			MOV	#T37BFR,(R1)+		;ADDRESS OF MESSAGE BUFFER
10034	105316	005021				CLR	(R1)+		
10035	105320	012721	000024			MOV	#20,(R1)+		;LENGTH OF MESSAGE BUFFER
10036	105324	005021				CLR	(R1)+		

## TEST 9: FUNCTION TIMING

```

10037 105326 012711 000000      MOV      #0,(R1)          ;SELECT DRIVE ZERO
10038 105332 012702 000030      MOV      #24.,R2         ;NUMBER OF LOCATIONS TO BE CLEARED
10039 105336 012762 177777 102162 64#: MOV      #177777,T37BFR(R2) ;ALL ONES TO MESSAGE BUFFER
10040 105344 005742              TST      -(R2)           ;NEXT LOCATION
10041 105346 022702 000000      CMP      #0,R2          ;AT END OF LOOP YET
10042 105352 001371              BNE      64#            ;KEEP GOING UNTIL DONE
10043 105354 000207              RTS      PC              ;RETURN
10044
10045 105356              T37RT2:
10046 105356              SAVREG                 ;SAVE THE REGISTERS
10047 105362 012701 102250      MOV      #T37PK2,R1     ;START OF THE PACKET
10048 105366 012721 100006      MOV      #100006,(R1)+  ;WRITE SUBSYSTEM MEM. WITH ACK.
10049 105372 012721 102270      MOV      #T37BF2,(R1)+ ;ADDRESS OF DATA BLOCK
10050 105376 005021              CLR      (R1)+          ;EXTENDED ADDRESS
10051 105400 012721 000006      MOV      #6.,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
10052 105404 005021              CLR      (R1)+
10053 105406 012701 102270      MOV      #T37BF2,R1     ;POINT TO DATA SEL AREA
10054 105412 005021              CLR      (R1)+
10055 105414 005011              CLR      (R1)
10056 105416 000207              RTS      PC              ;RETURN
10057 105420              T37RT3:
10058 105420              SAVREG                 ;SAVE REGISTERS
10059 105424 012701 102260      MOV      #T37PK3,R1     ;SET UP POINTER ADDRESS
10060 105430 005021              CLR      (R1)+          ;COMMAND SPACE
10061 105432 005021              CLR      (R1)+          ;ADDRESS OF DATA BLOCK
10062 105434 005021              CLR      (R1)+          ;EXTENDED ADDRESS
10063 105436 005011              CLR      (R1)           ;SIZE OF DATA TRANSFER BLOCK
10064 105440 000207              RTS      PC              ;RETURN
10065 105442
10066 105442 104401              L10073: TRAP      C#ETST
10066 105444              ENDMOD
10067              .TITLE  TSV6 - PARAMETER CODING
10073
10078
10084
10085 105444              BGNMOD  TSV6
10085 105444              TSV6::
10086              .SBTTL  HARDWARE PARAMETER CODING SECTION
10087
10088
10089
10090
10091
10092
10093
10094
10095
10096
10097 105444              ;++
10097 105444 000010      ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
10097 105446              ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
10097 105448              ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
10097 105450              ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
10097 105452              ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
10097 105454              ; WITH THE OPERATOR.
10097 105456              ;--
10097 105458              BGNHRD
10097 105460              .WORD  L10075-L#HARD/2
10098 105462              L#HARD::
10099 105446              GPRMA  HPM1,0,0,160010,177776,YES ;GET TSBA/TSDB REGISTER ADDRESS.
10099 105448              .WORD  T#CODE
10099 105450              .WORD  HPM1
10099 105452              .WORD  T#LOLIM

```

HARDWARE PARAMETER CODING SECTION

```

10100 105454 177776      .WORD  T#HILIM
      105456      GPRMA  HPM2,2,0,0,776,YES      ;GET VECTOR ADDRESS.
      105456 001031      .WORD  T#CODE
      105460 105522      .WORD  HPM2
      105462 000000      .WORD  T#LOLIM
      105464 000776      .WORD  T#HILIM
10101      ;GPRMD  HPM3,4,0,340,0,7,YES      ;GET INTERRUPT PRIORITY.
10102 105466      ENDHRD
      .EVEN

      105466      L10075:
10103 105466      104      105      126  HPM1:  .ASCIZ  'DEVICE ADDRESS (TSBA/TSD) '
10104 105522      111      116      124  HPM2:  .ASCIZ  'INTERRUPT VECTOR '
10105 105546      111      116      124  HPM3:  .ASCIZ  'INTERRUPT PRIORITY '
10106      .EVEN
    
```



SOFTWARE PARAMETER CODING SECTION

```

10108                                     .SBTTL SOFTWARE PARAMETER CODING SECTION
10109
10110
10111                                     ;**
10112                                     ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
10113                                     ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
10114                                     ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
10115                                     ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
10116                                     ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
10117                                     ; WITH THE OPERATOR.
10118                                     ;--
10118 105576                                BGNSFT
10118 105576 000003                          .WORD L10076-L$SOFT/2
10118 105600
10119                                     L$SOFT::
10120 105600                                GPRML SPM1,0,-1,YES ; GET TRANSPORT TEST FLAG.
10120 105600 001130                          GPRML SPM4,2,-1,YES ; GET ITERATION CONTROL.
10120 105602 105636                          .WORD T$CODE
10120 105604 177777                          .WORD SPM4
10121                                     .WORD -1
10122                                     GPRMD SPM6,4,D,7777,0,7777,YES ; GET LOCAL ERROR LIMIT
10123 105606                                GPRMD SPM7,6,D,7777,0,7777,YES ; GET GLOBAL ERROR LIMIT
10124                                     ENDSFT
10125                                     .EVEN
10125 105606                                L10076:
10125 105606 105 116 101 SPM1: .ASCIZ 'ENABLE TRANSPORT TESTS '
10126 105636 111 116 110 SPM4: .ASCIZ 'INHIBIT ITERATIONS '
10127 105666 120 105 122 SPM6: .ASCIZ 'PER TEST ERROR LIMIT '
10128 105716 120 105 122 SPM7: .ASCIZ 'PER UNIT ERROR LIMIT '
10129                                     .SBTTL PATCH AREA
10130
10131                                     ;
10132                                     ; FINALLY A GENEROUS PATCH AREA.
10133                                     ;
10134                                     ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
10135                                     ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
10136                                     ;
10137                                     PATCH::
10138 105746                                .BLKW 32.
10139
10140 105746
10141
10143 106400 106400                          .=.!377*1
10145 106400                                LASTAD ;SET LAST USED ADDRESS.
10145 106400 000000                          .EVEN
10145 106402 000000                          .WORD 0
10145 106404                                .WORD 0
10146 106404                                L$LAST::
10147 000001                                ENDMOD
10147                                     .END

```

## Symbol table

ADDSSR	012206	G	C#AU	= 000052	DEVDR0	023362	FRESIZ	003120	G	INTFLA	016225		
ADR	= 000020	G	C#AUTO	= 000061	DEVNRD	023301	FUSI	004113		INTMAS	016224		
AMBTSS	006713		C#BRK	= 000022	DEVNXR	023217	F#AU	= 000015		INTR	016276	G	
ASSEMB	= 000010		C#BSEG	= 000004	DEVONL	023147	F#AUTO	= 000020		INTREC	002216	G	
A1716	= 000003		C#BSUB	= 000002	DEVSUM	023112	F#BGN	= 000040		INTVEC	016226		
BADDAT	003150	G	C#CEFG	= 000045	DFPTBL	002150	F#CLEA	= 000007		INTX	004274		
BADSSR	015760	G	C#CLCK	= 000062	DIAGMC	= 000000	F#DU	= 000016		IOKCKI	= 000200		
BDVPCR	= 177520	G	C#CLEA	= 000012	DICED	= 000001	F#END	= 000041		IOKSTP	= 000001		
BENBSW	002222	G	C#CLOS	= 000035	DSBINT	016264	F#HARD	= 000004		IPRI	= 002204	G	
BIE	= 040000		C#CLP1	= 000006	DUAD12	004637	F#HW	= 000013		ISR	= 000100	G	
BIT0	= 000001	G	C#CVEC	= 000036	DUFLG	003104	F#INIT	= 000006		IVEC	= 002202	G	
BIT00	= 000001	G	C#DCLN	= 000044	DUMMY	003054	F#JMP	= 000050		IXE	= 004000	G	
BIT01	= 000002	G	C#DODU	= 000051	EF.CON	= 000036	F#MOD	= 000000		I#AU	= 000041		
BIT02	= 000004	G	C#DRPT	= 000024	EF.NEW	= 000035	F#MSG	= 000011		I#AUTO	= 000041		
BIT03	= 000010	G	C#DU	= 000053	EF.PWR	= 000034	F#PROT	= 000021		I#CLN	= 000041		
BIT04	= 000020	G	C#EDIT	= 000003	EF.RES	= 000037	F#PWR	= 000017		I#DU	= 000041		
BIT05	= 000040	G	C#ERDF	= 000055	EF.STA	= 000040	F#RPT	= 000012		I#HRD	= 000041		
BIT06	= 000100	G	C#ERHR	= 000056	EMAXDU	017057	F#SEG	= 000003		I#INIT	= 000041		
BIT07	= 000200	G	C#ERRO	= 000060	EN	= 000000	F#SOFT	= 000005		I#MOD	= 000041		
BIT08	= 000400	G	C#ERSF	= 000054	ENAINI	016232	F#SRV	= 000010		I#MSG	= 000041		
BIT09	= 001000	G	C#ERSO	= 000057	ENVIRN	020710	F#SUB	= 000002		I#PROT	= 000040		
BIT1	= 000002	G	C#ESCA	= 000010	EPRTSW	002172	F#SW	= 000014		I#PTAB	= 000041		
BIT10	= 002000	G	C#ESEG	= 000005	EPRT1	006354	F#TEST	= 000001		I#PWR	= 000041		
BIT11	= 004000	G	C#ESUB	= 000003	EPRT2	006413	GDDAT	003152	G	I#RPT	= 000041		
BIT12	= 010000	G	C#ETST	= 000001	ERCM	012013	GERRMA	002166	G	I#SEG	= 000041		
BIT13	= 020000	G	C#EXIT	= 000032	ERRHI	002230	GETPAT	020254	G	I#SETU	= 000041		
BIT14	= 040000	G	C#GETB	= 000026	ERRK	017036	GETSEL	020336	G	I#SFT	= 000041		
BIT15	= 100000	G	C#GETW	= 000027	ERRLO	002232	G#CNT0	= 000200		I#SRV	= 000041		
BIT2	= 000004	G	C#GMAN	= 000043	ERRNO	= 001620	G#DELM	= 000372		I#SUB	= 000041		
BIT3	= 000010	G	C#GPHR	= 000042	ERRVEC	= 000004	G#DISP	= 000003		I#TST	= 000041		
BIT4	= 000020	G	C#GPL0	= 000030	ERTABE	003370	G#EXCP	= 000400		J#JMP	= 000167		
BIT5	= 000040	G	C#GPRI	= 000040	ERTABL	003170	G#HILI	= 000002		KIPAR0	= 172340		
BIT6	= 000100	G	C#INIT	= 000011	ESUM	017040	G#LOLI	= 000001		KIPAR1	= 172342		
BIT7	= 000200	G	C#INLP	= 000020	EVL	= 000004	G#NO	= 000000		KIPAR2	= 172344		
BIT8	= 000400	G	C#MANI	= 000050	EXBCNT	= 000010	G#OFFS	= 000400		KIPAR3	= 172346		
BIT9	= 001000	G	C#MEM	= 000031	EXPBRE	015562	G#OFSI	= 000376		KIPAR4	= 172350		
BOE	= 000400	G	C#MSG	= 000023	EXPD	002224	G#PRMA	= 000001		KIPAR5	= 172352		
BRINIT	004453		C#OPEN	= 000034	XPGBT	004527	G#PRMD	= 000002		KIPAR6	= 172354		
BSEL0	= 000000		C#PNTB	= 000014	XPGBT2	004563	G#PRML	= 000000		KIPAR7	= 172356		
BSEL1	= 000001		C#PNTF	= 000017	EXPMMSG	002314	G#RADA	= 000140		KIPDR0	= 172300		
CHKAMB	016124		C#PNTS	= 000016	EXPREC	015554	G#RADB	= 000000		KIPDR1	= 172302		
CHKMAN	020560	G	C#PNTX	= 000015	EXTA	005766	G#RADD	= 000040		KIPDR2	= 172304		
CHKTSS	016416		C#QIO	= 000377	EXTEND	005764	G#RADL	= 000120		KIPDR3	= 172306		
CKDROP	017262		C#RDBU	= 000007	EXTFEA	002220	G#RADO	= 000020		KIPDR4	= 172310		
CKEMAX	017162		C#REFG	= 000047	E#END	= 002100	G#XFER	= 000004		KIPDR5	= 172312		
CKMSG	011440	G	C#RESE	= 000033	E#LOAD	= 000035	G#YES	= 000010		KIPDR6	= 172314		
CKMSG2	011560	G	C#REVI	= 000003	FATERR	= 000060	HIADDR	= 001400		KIPDR7	= 172316		
CKRAM	011174	G	C#RFLA	= 000021	FATFLG	002214	HOE	= 100000	G	KTENAB	003126	G	
CKRAM2	011304	G	C#RPT	= 000025	FERCH	012002	HPM1	= 105466		KTFLG	003124	G	
CMDPKT	021244	G	C#SEFG	= 000046	FIFEXP	012250	HPM2	= 105522		KTINIT	021070		
CMPMEM	017740		C#SPRI	= 000041	FIF1MS	012322	HPM3	= 105546		KTOFF	017354		
CONFIG	017330		C#SVEC	= 000037	FIF2MS	012371	IBE	= 010000	G	KTON	017336		
COUNT	002302	G	C#TPRI	= 000013	FILLME	017502	IDU	= 000040	G	LERRMA	002164	G	
CSRADD	002200	G	DATA	002304	G	FNOINT	004211	IER	= 020000	G	LISTAL	= 000001	
CTAB	003156	G	DATASC	020312		FORCER	002170	IFAUlT	004252		LOE	= 040000	G
CTABE	003170	G	DEBUGM	011712		FREE	003116	INCRK	017124		LOOPCN	002210	G
CTABM	003156	G	DEVCNT	002212	G	FREEHI	003122	INTCPC	016230		LOOPCO	013206	

## Symbol table

LOOPFL	003154	G	L10002	005762	L10074	102102	O#GNSW=	000001	PUNIT	022270
LOT	000010	G	L10003	012124	L10075	105466	O#POIN=	000001	PW.D11=	000021
L#ACP	002110	G	L10004	012142	L10076	105606	O#SETU=	000000	PW.D13=	000022
L#APT	002036	G	L10005	012160	MEMADD	014034	PASRPT	022040	PW.D22=	000020
L#AU	022336	G	L10006	012166	MEMCK	021262	PATCH	105746	PW.NOP=	000000
L#AUT	002070	G	L10007	012204	MENASC	020527	PATDAT	020310	PW.NO1=	000023
L#AUTO	022542	G	L10010	012222	MENERR	020454	PC.ERA=	002400	PW.RDE=	000024
L#CCP	002106	G	L10011	012246	MENRES	020556	PC.IER=	002000	PW.RDR=	000001
L#CLEA	022622	G	L10012	012320	MMVEC =	000250	PC.NOO=	001000	PW.RDS=	000005
L#CO	002032	G	L10013	012470	MSA.FR=	000006	PC.REL=	000000	PW.RFI=	000003
L#DEPO	002011	G	L10014	013204	MSA.NO=	000000	PC.REW=	000400	PW.WCT=	000006
L#DESC	003402	G	L10015	014032	MSA.NR=	000004	PKBCNT=	000006	PW.WFI=	000004
L#DESP	002076	G	L10016	014054	MSA.VO=	000002	PKHI =	000004	PW.WFM=	000007
L#DEVP	002060	G	L10017	015560	MSGEXP	012224	PKLOW =	000002	PW.WMI=	000010
L#DISP	002124	G	L10020	015566	MSGLOO	013144	PKTADD	007632	PW.WNP=	000011
L#DLY	002116	G	L10021	015574	MSGSTA	012430	PKTFRM	007574	PW.WTR=	000002
L#DTP	002040	G	L10022	015606	MSGSUB	014022	PKTGET	012144	P.ACK =	100000
L#DTYP	002034	G	L10023	015630	MS.ATT=	000006	PKTMES	012170	P.CMD =	000037
L#DU	022434	G	L10024	015656	MS.EXT=	000200	PKTRAM	004741	P.CONT=	000012
L#DUT	002072	G	L10025	016016	MS.RSD=	000001	PKTSSR	012126	P.CVC =	040000
L#DVTY	003374	G	L10026	016326	MS.RSF=	000020	PNT =	001000	P.FMT =	000140
L#EF	002052	G	L10030	022266	MS.RST=	000010	PRAMPK	014056	P.FORM=	000011
L#ENVI	002044	G	L10031	022432	M8186	005550	PRASC	014603	P.GETS=	000017
L#ETP	002102	G	L10032	022540	M8189	005641	PRBEXP	015550	P.IE =	000200
L#EXP1	002046	G	L10033	022620	NBA =	002000	PRBMSG	015416	P.INIT=	000013
L#EXP4	002064	G	L10034	022646	NEWPAS	021774	PRBREC	015552	P.MODE=	007400
L#EXP5	002066	G	L10035	023110	NODEV	003106	PRBTOT	015503	P.OPP =	020000
L#HARD	105446	G	L10036	032232	NOINIT	004331	PRBYTE	015202	P.POSI=	000010
L#HIME	002120	G	L10037	024074	NOINTR	004215	PRI =	002000	P.READ=	0000C1
L#HPCP	002016	G	L10040	024616	NOITS	002162	PRIADD	010236	P.SWB =	010000
L#HPTP	002022	G	L10041	025342	NOMAN	020614	PRIAO	010306	P.WRIT=	000005
L#HW	002150	G	L10042	026164	NOMEM	005454	PRI BXO	007670	P.WRTC=	000004
L#ICP	002104	G	L10043	041330	NP.IR =	000200	PRIEQU	010136	P.WRTS=	000006
L#INIT	021542	G	L10044	033634	NP.LOO=	000040	PRI PKT	007446	QVP	002176
L#LADP	002026	G	L10045	035260	NP.OUT=	000100	PRI RAM	010144	RAMASC	014236
L#LAST	106404	G	L10046	035654	NP.WRP=	000020	PRI TAD	010352	RAMDAT	002234
L#LOAD	002100	G	L10047	036340	NSI	004146	PRI TSS	006020	RAMERR	015570
L#LUN	002074	G	L10050	046666	NSINIT	004403	PRI TO	010434	RAMEXP	015610
L#MREV	002050	G	L10051	042222	NUL	004523	PRI T1	010477	RAMFOR	010174
L#NAME	002000	G	L10052	043034	NULCR	004524	PRI XOR	010020	RAMSIZ	002274
L#PRIO	002042	G	L10053	052744	NXM =	004000	PRI O0	= 000000	RAMTAD	015576
L#PROT	021532	G	L10054	047542	NXMFLG	003130	PRI O1	= 000040	RCVHIA	002276
L#PRT	002112	G	L10055	050352	NXMHI	003134	PRI O2	= 000100	RCVLOA	002300
L#REPP	002062	G	L10056	051166	NXMLO	003132	PRI O3	= 000140	RDERR	005202
L#REV	002010	G	L10057	055740	NXMTST	021436	PRI O4	= 000200	RECMG	002460
L#RPT	022650	G	L10060	054406	NXR	003734	PRI O5	= 000240	RECV	002226
L#SOFT	105600	G	L10061	063312	NXRERR	005732	PRI O6	= 000300	REGSAV	020220
L#SPC	002056	G	L10062	060376	NXRX	003773	PRI O7	= 000340	RETERR	005366
L#SPCP	002020	G	L10063	073242	NXTU	022006	PRMESS	014322	REWIND	011074
L#SPTP	002024	G	L10064	064404	OFL =	000100	PRMNO	002312	RMCHBE=	000167
L#STA	002030	G	L10065	065464	ONEFIL=	000000	PRMSG	014632	RMCHEN=	000200
L#SW	002160	G	L10066	066326	O#APTS=	000000	PRMSG0	015012	RMMSGB=	000215
L#TEST	002114	G	L10067	067230	O#AU =	000001	PRMSG1	015057	RMMSGG=	000234
L#TIML	002014	G	L10070	101016	O#BGNR=	000001	PRMSG2	015115	RMPKTB=	000201
L#UNIT	002012	G	L10071	074336	O#BGNS=	000001	PROASC	014500	RMPKTE=	000210
L10000	002156		L10072	075420	O#DU =	000001	PRI ASC	014545	RMR =	010000
L10001	002170		L10073	105442	O#ERRT=	000000	PST32W	003144	RWPACK	011170

## Symbol table

SC	=	100200	S2.OUT	=	000040	T#FLAG	=	000040	T29DAT	026230	T30BS0	036530	
SCE	=	020000	S2.UND	=	000003	T#GMAN	=	000000	T29DLY	026400	T30BS1	036531	
SCHERR	005274	TBLEND	=	003054	G	T#HILI	=	000776	T29DSW	026240	T30CNT	036550	
SCHE	005007	TCOASC	006554	T#LAST	=	000001	T29DTA	027743	T29E0T	030031	T30CNU	036552	
SDELAY	010740	TCOCOD	006754	T#LOLI	=	000000	T29LON	031125	T29L00	023462	T30DAT	036410	
SELASC	020522	TEMP1	003110	G	T#LSYM	=	010000	T29L0P	031207	T30DLY	036556		
SELDAT	=	000004	TEMP2	003112	G	T#LTNO	=	000011	T29L0Q	027326	T30DSW	036420	
SEL2	=	000002	TERCLS	=	000016	T#NEST	=	177777	T29LOR	027201	T30DTA	041034	
SEMAP	017376	TESTNO	=	000011	T#NSO	=	000000	T29NEF	026530	T30DTR	040770		
SETU	022072	TEXASC	006513	T#NS1	=	000005	T29NEQ	031445	T30ETH	036416			
SFFMSG	012162	TFCASC	006615	T#NS2	=	000002	T29OFL	026402	T30FCN	036554			
SFHERR	003701	TIMEXP	015632	G	T#PTNU	=	000000	T29OF7	030415	T30IBT	036731		
SFIERR	003646	TIMSGO	015660	T#SAVL	=	177777	T29PAC	026220	T30IBU	036560			
SFMSG	012114	TINERR	012101	T#SEGL	=	177777	T29PBP	031271	T30IMV	036536			
SFPTBL	002160	TMPBFR	002624	G	T#SUBN	=	000001	T29PK2	026330	T30LO0	032260		
SIFLAG	003146	TNAM	016764	T#TAGL	=	177777	T29PK3	026340	T30LOQ	037530			
SIMSG	012046	TRANST	002160	G	T#TAGN	=	010077	T29RB	026342	T3ONEF	040476		
SKIPT	003372	TSBA	=	000000	G	T#TEMP	=	000000	T29RDF	026620	T3OFL	040207	
SOFINI	016054	TSBAH	=	000001	G	T#TEST	=	000011	T29RDG	031543	T30PAC	036400	
SPACE	010544	TSDB	=	000000	G	T#TSTM	=	177777	T29RES	032046	T30PK2	036510	
SPM1	105606	TSDBH	=	000001	G	T#TSTS	=	000001	T29RIB	031624	T30PK3	036520	
SPM4	105636	TSFCOD	007314	T#T#AU	=	010031	T29RNC	030254	T29RN	026356	T30PTB	037142	
SPM6	105666	TSREJ	=	000006	T#T#AUT	=	010033	T29RRF	026667	T3ORB	036522		
SPM7	105716	TSSDEF	006664	T#T#CLE	=	010034	T29RRG	027003	T29RRN	031724	T3ORDF	037313	
SRO	=	177572	TSSR	=	000002	G	T#T#DU	=	010032	T29RSZ	026376	T3ORDG	037371
SR1	=	177574	TSSRBI	003476	G	T#T#HAR	=	010075	T29RT2	032140	T3ORES	041152	
SR2	=	177576	TSSRFO	006473	T#T#HW	=	010000	T29RT3	032202	T3ORIB	036645		
SR3	=	172516	TSSRH	=	000003	G	T#T#INI	=	010030	T29RWN	030205	T3ORN	036536
SSR	=	000200	TSSX	004014	T#T#MSG	=	010025	T29SSR	027407	T3ORRM	040555		
STATCO	012472	TSTBLK	002744	G	T#T#PRO	=	010027	T29S2	026346	T3ORRN	040633		
SVCGBL	000000	TSTCNT	002206	G	T#T#RPT	=	010035	T29S3	026352	T3ORRP	040712		
SVCINS	=	000000	TSTEND	017000	T#T#SOF	=	010076	T29TM	030127	T3ORT2	041244		
SVCSUB	=	000001	TSTFLA	002306	G	T#T#SRV	=	010026	T29TRL	031357	T3ORT3	041306	
SVCTAG	=	000000	TSTL00	016536	G	T#T#SUB	=	010074	T29VCK	030671	T3ORWN	040140	
SVCTST	=	000001	TSTPTR	002310	G	T#T#SW	=	010001	T29WDC	026342	T3OSKM	037014	
S#LSYM	=	010000	TSTSET	016570	G	T#T#TES	=	010073	T29WDD	030577	T3OSSR	037611	
SO.IDB	=	000010	TST29I	032017	T1	023432	G	T29WDE	027462	T3OSZ	036526		
SO.IFB	=	000002	TST30I	041131	T1.1	023462	G	T29WDF	027251	T3OS2	036532		
SO.IFP	=	000001	TST31I	046443	T1.2	024112	G	T29WDR	026360	T3OS3	036534		
SO.ILD	=	000020	TST32I	052540	T1.3	024634	G	T29WLK	027544	T30TM	040006		
SO.ION	=	000040	TST33I	055545	T1.4	025360	G	T29WNG	026423	T30TMK	040414		
SO.IRD	=	000100	TST34I	063107	T2	032234	G	T29WRT	027631	T30TM2	040063		
SO.IRW	=	000004	TST35I	073033	T2.1	032260	G	T29WSS	031036	T30TPB	037233		
SO.ISP	=	000200	TST36I	100617	T2.2	033652	G	T3	041332	T3OVCK	040341		
S1.ICE	=	002000	TST37I	105243	T2.3	035276	G	T3BFLG	003142	T30WB	036522		
S1.IEQ	=	010000	TSV2	002000	G	T2.4	035672	G	T3.1	041362	T30WDC	040262	
S1.IFM	=	001000	TSV3	002170	G	T23A	003136	G	T3.2	042240	T30WDD	037070	
S1.IHE	=	000400	TSV4	021532	G	T23B	003140	G	T30WF2	036530	T30WDE	037662	
S1.IID	=	004000	TSV6	105444	G	T29AM3	030327	G	T30WDF	037453	T30WDF	037453	
S1.IIR	=	020000	TSV7B	023432	G	T29BA	030744	G	T31AM3	044716	T31AM3	044716	
S1.I2R	=	040000	TTIBFR	=	177562	G	T29BF2	026350	T31BA	045256	T31BA	045256	
S1.PAR	=	100000	TTICSR	=	177560	G	T29B0T	027676	T31BFR	043112	T31BFR	043112	
S2.ATI	=	000010	TTIVEC	=	000060	G	T29BS0	026350	T31BF2	043220	T31BF2	043220	
S2.BTI	=	000004	T#ARGC	=	000003	T#CODE	=	001130	T31B0T	044245	T31B0T	044245	
S2.DIM	=	000200	T#ERRN	=	001620	T#ERRN	=	001620	T31BS0	043220	T31BS0	043220	
S2.ILW	=	000100	T#EXCP	=	000000	T29CNT	026374	T30BF2	036530	T31BS1	043221		
S2.INR	=	000020				T29CON	026362	T30B0T	037741	T31CNT	043236		

## Symbol table

T31CNU	043240	T32DAT	051240	T34BFR	060462	T35DAT	067300	T36CNT	075626
T31CON	043232	T32DLY	051414	T34BF2	060576	T35DLY	067442	T36CNU	075630
T31DAT	043100	T32DSW	051250	T34BOT	061134	T35DSW	067310	T36CON	075622
T31DLY	043242	T32ECF	052355	T34BS0	060576	T35DTA	072225	T36DAT	075470
T31DSW	043110	T32EOT	051511	T34BS1	060577	T35EOT	070410	T36DLY	075632
T31DTA	046346	T32ERA	051716	T34CNT	060572	T35INT	072501	T36DSW	075500
T31EOT	044440	T32L00	046720	T34CON	060610	T35LON	071370	T36DTA	100522
T31LON	045420	T32OPI	052503	T34DAT	060450	T35L00	063344	T36EOT	076705
T31L00	041362	T32PAC	051230	T34DLY	060574	T35L0P	071452	T36LON	077665
T31L0P	045502	T32PK2	051340	T34DSW	060460	T35L0Q	070105	T36L00	073300
T31L0Q	044016	T32PK3	051350	T34EOT	062105	T35LOR	067760	T36L0P	077747
T31LOR	043671	T32RB	051352	T34ET	062016	T35MOT	072403	T36L0Q	076346
T31NEF	045740	T32RES	052600	T34ETC	061057	T35NEF	071710	T36LOR	076221
T31OFL	044765	T32RIB	052036	T34ETN	061351	T35NIN	072756	T36NAS	075634
T31PAC	043070	T32RT2	052672	T34ETO	060702	T35OFL	070735	T36NEF	100205
T31PBP	045564	T32RT3	052722	T34ETS	061430	T35OPM	072572	T36OFL	077232
T31PK2	043200	T32RWN	051600	T34ETZ	061522	T35PAC	067270	T36PAC	075460
T31PK3	043210	T32SCF	052134	T34ET2	061267	T35PBP	071534	T36PBP	100031
T31RB	043212	T32SZ	051356	T34L00	055772	T35PK2	067400	T36PK2	075570
T31RDE	043244	T32TSA	052211	T34OFL	062427	T35PK3	067410	T36PK3	075600
T31RDF	043443	T32WB	051352	T34PAC	060440	T35RB	067412	T36RB	075602
T31RES	046510	T32WDC	052436	T34PK2	060550	T35RDF	067532	T36RDF	075773
T31RN	043226	T33BFR	054472	T34PK3	060560	T35RES	073064	T36RES	100640
T31RNC	044643	T33BF2	054600	T34POS	060614	T35RN	067426	T36RN	075616
T31RRF	043512	T33BOT	055225	T34RB	060562	T35RNC	070613	T36RNC	077110
T31RT2	046602	T33BS0	054600	T34RES	063132	T35RRF	067601	T36RRF	076042
T31RT3	046644	T33BS1	054601	T34RNC	062306	T35RT2	073156	T36RT2	100732
T31RWN	044574	T33CNT	054616	T34RRE	060766	T35RT3	073220	T36RT3	100774
T31SC	043607	T33CNU	054620	T34RSZ	060570	T35RWE	072670	T36RWN	077041
T31SCF	046061	T33CON	054612	T34RT2	063224	T35RWN	070544	T36SC	076137
T31SSR	044077	T33DAT	054460	T34RT3	063266	T35SC	067676	T36SCF	100303
T31SZ	043216	T33DLY	054622	T34RWN	062237	T35SCF	072006	T36SSR	076427
T31S2	043222	T33DSW	054470	T34SSR	061763	T35SSR	072322	T36SZ	075606
T31S3	043224	T33DTA	055450	T34STM	061600	T35SZ	067416	T36S2	075612
T31TIH	044340	T33L00	052776	T34SZ	060566	T35S2	067422	T36S3	075614
T31TH	044517	T33PAC	054450	T34S2	060600	T35S3	067424	T36TIM	076630
T31TRL	045652	T33PK2	054560	T34S3	060602	T35TIM	070333	T36TH	076764
T31TSA	046136	T33PK3	054570	T34TM	062163	T35TM	070467	T36TRL	100117
T31VCK	045203	T33RB	054572	T34THK	061663	T35TRL	071622	T36TSA	100360
T31WB	043212	T33RBP	054624	T34VCK	062673	T35TSA	072063	T36VCK	077450
T31WDC	045130	T33RES	055562	T34WB	060562	T35VCK	071153	T36WB	075602
T31WDD	045040	T33RN	054606	T34WD	060604	T35WB	067412	T36WDC	077375
T31WDE	044133	T33RT2	055654	T34WDC	062571	T35WDC	071100	T36WDD	077305
T31WDF	043741	T33RT3	055716	T34WDD	062502	T35WDD	071010	T36WDE	076463
T31WDR	043230	T33RWN	055320	T34WDR	060606	T35WDE	070166	T36WDF	076271
T31WNG	043371	T33SSR	055141	T34WSS	063020	T35WDF	070030	T36WDR	075620
T31WNH	043310	T33SZ	054576	T34WTH	061200	T35WDR	067430	T36WNG	075705
T31WRF	046243	T33S2	054602	T35AM3	070666	T35WNG	067444	T36WRF	100442
T31WSS	045331	T33S3	054604	T35BA	071226	T35WRF	072145	T36WSS	077576
T32AM3	051647	T33UNC	054762	T35BFR	067312	T35WSS	071301	T37AM3	103607
T32BA	051763	T33UND	055052	T35BF2	067420	T36AM3	077163	T37BA	104147
T32BFR	051252	T33WB	054572	T35BOT	070240	T36BA	077523	T37BFR	102162
T32BOE	052266	T33WDC	055367	T35BS0	067420	T36BFR	075502	T37BF2	102270
T32BOT	051416	T33WDR	054610	T35BS1	067421	T36BF2	075610	T37BOT	103161
T32CMD	051360	T33WPW	054702	T35CNT	067436	T36BOT	076535	T37BS0	102270
T32CNT	051410	T34AM3	062361	T35CNU	067440	T36BS0	075610	T37BS1	102271
T32CNU	051412	T34BA	062746	T35CON	067432	T36BS1	075611	T37CNT	102306

## Symbol table

T37CNU	102310	T37SZ	102266	T8	073244	G	WSMBK	021254	G	X#OFFS=	000400
T37CON	102302	T37S2	102272	T8.1	073300		XFERAS	016020		X#TRUE=	000020
T37DAT	102150	T37S3	102274	T8.2	074354		XNXM	016456		X1.COR=	020000
T37DLY	102312	T37TIM	103254	T9	101020	G	XORBFO	007752		X1.DLT=	100000
T37DSW	102160	T37TM	103410	T9.1	101054		XORFOR	010070		X1.MBZ=	017375
T37DTA	105146	T37TRL	104543	UAM	= 000200	G	XST0	= 000006	G	X1.RBP=	000400
T37EOT	103331	T37TSA	105004	UNITN	002174	G	XST1	= 000010	G	X1.SPA=	040000
T37LON	104311	T37VCK	104074	UNREC	= 000006		XST2	= 000012	G	X1.UNC=	000002
T37LOO	101054	T37WB	102262	USI	004117		XST3	= 000014	G	X2.BUF=	000100
T37LOP	104373	T37WDC	104021	WAITF	016330	G	XST4	= 000016	G	X2.EXT=	000200
T37LOQ	102755	T37WDD	103731	WC.IFA=	000200		XS0BOT=	000002		X2.OPH=	100000
T37LOR	102630	T37WDE	103072	WC.IFE=	000002		XS0EOT=	000001		X2.RCE=	040000
T37NEF	104631	T37WDF	102700	WC.IG0=	000001		XSOIE	= 000040		X2.REV=	000077
T37OFL	103656	T37WDR	102300	WC.IG1=	000010		XSOILA=	000400		X2.SPA=	035400
T37PAC	102140	T37WNG	102314	WC.IRE=	000010		XSOILC=	001000		X2.UNI=	000007
T37PBP	104455	T37WRF	105066	WC.IRW=	000004		XSOLET=	020000		X2.WCF=	002000
T37PK2	102250	T37WSS	104222	WC.IOT=	000100		XSOMOT=	000200		X3.DCK=	000010
T37PK3	102260	T4	046670	WC.I1T=	000040		XSONEF=	002000		X3.MBZ=	000006
T37RB	102262	T4.1	046720	WC.I5R=	000020		XSOONL=	000100		X3.MDE=	177400
T37RDF	102402	T4.2	047560	WF.IED=	000010		XSOPED=	000010		X3.OPI=	000100
T37RES	105264	T4.3	050370	WF.IER=	000004		XSORLL=	010000		X3.REV=	000040
T37RN	102276	T5	052746	WF.IHI=	000200		XSORLS=	040000		X3.RIB=	000001
T37RNC	103534	T5.1	052776	WF.IRE=	000040		XSO7MK=	100C00		X3.SPA=	000200
T37RRF	102451	T6	055742	WF.IWF=	000020		XSOVCK=	000020		X3.TRF=	000020
T37RT2	105356	T6.1	055772	WF.IWR=	000100		XSOWLE=	004000		X4.HSP=	100000
T37RT3	105420	T7	063314	WF.I3R=	000002		XSOWLK=	000004		X4.MBZ=	017400
T37RWN	103465	T7.1	063344	WF.I4R=	000001		XXCOMM	003114	G	X4.RCE=	040000
T37SC	102546	T7.2	064422	WRTCHR	010742	G	X#ALWA=	000000		X4.TSM=	020000
T37SCF	104727	T7.3	065502	WRTERR	005107		X#FALS=	000040		X4.WRC=	000377
T37SSR	103036	T7.4	066344	WRTMSG	005052						

. ABS. 106404 000 (RW,I,GBL,ABS,OVR)  
 000000 001 (RW,I,LCL,REL,CON)

Errors detected: 0

## \*\*\* Assembler statistics

Work file reads: 297  
 Work file writes: 285  
 Size of work file: 31544 Words ( 124 Pages)  
 Size of core pool: 19714 Words ( 75 Pages)  
 Operating system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:07:05.40  
 CVTSDD,CVTSDD,-SVC/ML,CVTSDD