

.REM.

## IDENTIFICATION

PRODUCT ID: AC-T818A-MC  
PRODUCT TITLE: CNTSCAO TSV05 CTRL LT3  
DECO/DEPO: 1.0  
DEPARTMENT: ISS/DIAGNOSTIC SERVICES  
DATE: APRIL 09, 1984

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) 1984 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 SBC-11/21+ RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSV05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A SBC-11/21+ 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

SBC-11/21+ PROCESSOR AND MEMORY  
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY  
(26K USEABLE I.E. 4K FOR I/O PAGE)  
TSV05 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)  
CONSOLE TERMINAL  
PDP-11 DIAGNOSTIC SUPERVISOR (HSAAA.SYS VERSION 34 OR LATER)  
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

### 1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. XXDP+ USER'S MANUAL
2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE
3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL
4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL

### 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

FUNCTIONAL SBC-11/21+ 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 TSO5 TRANSPORT IS A KNOWN GOOD REEL  
 OF TAPE.  
 CNTSAA AND CNTSBA HAVE SUCESSFULLY 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.

### 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 SBC-11/21+ DIAGNOSTIC SUPERVISOR COMPATIBLE  
 PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE  
 XXDP+ USERS MANUAL. THE USER ENTRY IS IN QUOTES.

#### BOOT THE DIAGNOSTIC MEDIA

```
.R NTSC??
DIAG. RUN-TIME SERVICES REV D. APR 79
CNTSC-A-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 "DDDD".

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:DDDD	EXECUTE DDDDD PASSES (DDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDD = 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 MACROS)
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 6 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 * 176000, 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) 176000 ? <ENTER THE ADDRESS OF THE  
1SBA/TSOB 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 11/21+ 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.

0 UNITS (D) ? 8<CR>

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

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

UNIT 3  
CSR ADDRESS (0) ? 160000<CR>  
SUB-DEVICE # (0) ? 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 "IXR" 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  
 CNTSC 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> IEOF <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:  
 CATMIS<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.

CNTSC HRD ERR 001 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.

CNTSC HRD FRR 00121 ON UNIT 00 TST 001 SUB 002 PC: 023306  
MOT BIT XSI2) NOT SET DURING REWIND (EXTENDED FEATURES MODE)  
EXPD: 0J0312 RFCV: 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 (SBC-11/21\*)

DR>STA/FLA:PNT:HCE

UNITS (D) ? 1

UNIT 0

DEVICE ADDRESS (0) 176000 ? <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 INITIALIZE #4 TEST  
TST: 002 OFF-LINE REJECT AND REWIND TEST  
TST: 003 BASIC WRITE DATA TEST  
TST: 004 BASIC READ DATA TEST  
TST: 005 SPACE RECORDS TEST  
TST: 006 REREADS TEST  
TST: 007 WRITE DATA RETRY TEST  
TST: 008 WRITE TAPE MARK 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 FALCON PROCESSOR.

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).

THE TIMES REQUIRED TO RUN TESTS 1 THROUGH 8 IN ONE COMMAND:

Q.V. 7 MINUTES  
DEFAULT 34 MINUTES

MORE EXHAUSTIVE CHECKS ARE AVAILABLE BY ALLOWING THE DIAGNOSTIC PROGRAMS TO RUN FOR MORE THAN ONE PASS. THE SECOND PASS OF THE PROGRAM IS MORE COMPREHENSIVE THAN THE FIRST PASS. ALL ITERATIONS AFTER THE FIRST PASS ARE THE SAME, HOWEVER, THEY ARE SUBSTANTIALLY LONGER.

#### 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) 176000 ? <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: INITIALIZE #4 TEST

13.1

THIS TEST VERIFIES THAT WRITING INTO THE TSSR RETURNS THE CONTROLLER TO ITS INITIALIZED STATE FROM VARIOUS CONDITIONS (I.E. LOOPBACK ENABLED, FORCING WRONG PARITY, INVERTING SENSE OF EXTENDED FEATURES SWITCH, ETC.)

TEST 2: OFF-LINE AND REJECT REWIND

THIS TEST VERIFIES BASIC TAPE-MOTION COMMAND DECODING AND BASIC OPERATION OF THE REWIND POSITIONING COMMAND. IT DOES NOT NECESSARILY DEMONSTRATE THAT THE TRANSPORT CAN BE REWOUND FROM AN ARBITRARY POSITION ON THE TAPE. SUBSEQUENT TESTS IMPLICITLY CHECK THE OPERATION OF THE REWIND COMMAND SINCE THEY MUST TYPICALLY REWIND THE TAPE IN THE NORMAL COURSE OF THEIR TEST SEQUENCES.

TEST 3: BASIC WRITE DATA

THIS TEST VERIFIES THAT THE WRITE DATA (NEXT) COMMAND OPERATES PROPERLY, UP TO THE POINT OF CHECKING THAT THE DATA WAS ACTUALLY WRITTEN ONTO THE TAPE CORRECTLY. CHECKING IN THIS TEST IS LIMITED TO VERIFYING THAT THE COMMAND TERMINATED CORRECTLY WITH THE CORRECT REGISTER, MESSAGE BUFFER AND RAM CONTENTS.

\*\*\*\*\*  
CAUTION  
THE LSI BUS DRIVERS FOR ALL AVAILABLE ADDRESS LINES(16-21) ARE ONLY CHECKED WHEN RUNNING ON A 11/21+ SYSTEM WITH MORE THAN 128K WORDS OF MEMORY!  
\*\*\*\*\*

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

THIS TEST VERIFIES THAT THE READ FORWARD AND READ REVERSE COMMANDS OPERATE PROPERLY. VARIOUS COMBINATIONS OF ODD AND EVEN DATA BUFFER BOUNDARIES, RECORD SIZES (UP TO 64K BYTES IF MEMORY SPACE IS AVAILIABLE), AND BYTE-SWAP CONTROL ARE USED. THIS TEST OF COURSE, FURTHER VERIFIES THE WRITE DATA COMMAND BY ACTUALLY READING AND VERIFYING WRITTEN DATA. ALSO TESTED ARE PROPER TERMINATIONS ON EXCEPTIONAL OR ERROR CONDITIONS: RECORD LENGTH LONG, RECORD LENGTH SHORT, READ REVERSE AT BOT, ILLEGAL DATA BUFFER ADDRESSES, ILLEGAL CODES IN THE MODE FIELD OF THE BASIC READ COMMAND, AND DATA BUFFERS IN NON-EXISTANT MEMORY.

\*\*\*\*\*  
CAUTION  
THE LSI BUS DRIVERS FOR ALL AVAILABLE ADDRESS LINES(16-21) ARE ONLY CHECKED WHEN RUNNING ON A 11/21+ SYSTEM WITH MORE THAN 128K WORDS OF MEMORY!  
\*\*\*\*\*

TEST 5: SPACE RECORDS

THIS TEST VERIFIES THAT THE SPACE RECORDS FORWARD AND SPACE RECORDS REVERSE POSITION COMMANDS OPERATE PROPERLY WHEN SPACING

OVER NORMAL DATA RECORDS. OPERATION WHEN SPACING OVER TAPE MARKS IS VERIFIED IN A SUBSEQUENT TEST. THE BASIC WRITE DATA TEST SHOULD HAVE BEEN RUN SUCCESSFULLY FOR THIS TEST TO PRODUCE MEANINGFUL RESULTS. THIS TEST CONSISTS OF A SERIES OF SUBTESTS. IN EACH OF THE SUBTESTS, THE TAPE IS ENTIRELY WRITTEN WITH RECORDS OF VARYING SIZES AND DATA CONTENT; THE FIRST 4 BYTES OF EACH RECORD INDICATE THAT RECORD'S RELATIVE POSITION ON TAPE. AFTER EACH SPACING OPERATION, THE TAPE POSITION IS VERIFIED BY READING THE NEXT OR PREVIOUS RECORD AND COMPARING THE POSITION DATA WITH THE EXPECTED RESULT.

TEST 6: REREADS

THIS TEST VERIFIES THAT THE REREAD PREVIOUS AND REREAD NEXT COMMANDS OPERATE PROPERLY. VARIOUS COMBINATIONS OF ODD AND EVEN DATA BUFFER BOUNDARIES, RECORD SIZES (UP TO 64K BYTES IF MEMORY SPACE IS AVAILIABLE), AND BYTE-SWAP (SWP) AND OPPOSITE (OPP) CONTRL ARE USED. ALSO TESTED ARE PROPER TERMINATIONS ON EXCEPTIONAL OR ERROR CONDITIONS: RECORD LENGTH LONG, RECORD LENGTH SHORT, READ REVERSE AT BOT, ILLEGAL DATA BUFFER ADDRESSES, AND DATA BUFFERS IN NONEXISTENT MEMORY.

\*\*\*\*\*  
 CAUTION  
 THE LSI BUS DRIVERS FOR ALL AVAILABLE ADDRESS LINES(16-21) ARE ONLY CHECKED WHEN RUNNING ON A 11/21. SYSTEM WITH MORE THAN 128K WORDS OF MEMORY.  
 \*\*\*\*\*

TEST 7: WRITE DATA RETRY

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

TEST 8: WRITE/READ TAPE MARK

THIS TEST VERIFIES THAT THE WRITE TAPE MARK COMMAND OPERATES PROPERLY. IT IS VERIFIED THAT THE TAPE MARK IS WRITTEN ONTO TAPE BY CHECKING THAT THE READ AND SPACE RECORDS COMMANDS DETECT THE TAPE MARK. IN ADDITION, SINCE WRITE TAPE MARK IS THE FIRST SUBCOMMAND UNDER THE FORMAT COMMAND BEING TESTED, IT IS VERIFIED THAT THE CLEAR VOLUME CHECK (CVC) BIT OPERATES PROPERLY AND THAT FORMAT COMMANDS WITH ILLEGAL MODE CODES ARE REJECTED.

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

CVTSCAO => CNTSCAO

JAKI BERG

9-APR-1984

CHANGES WERE MADE TO CVTSCAO TO PRODUCE CNTSCAO FOR THE

FALCON-PLUS PROJECT (SRC-11/21+). CHANGES, MARKED BY  
";JB REV A-0", ARE:  
- SET THE ODT BREAK VECTOR (LOCATION 140) TO THE  
STARTING ADDRESS OF FALCON'S ODT ROM (170000-OCIAL).  
- LOWER THE GENERAL INTERRUPT PRIORITY FROM 7 TO 6.  
- CHANGE DEFAULT CSR ADDRESS FROM 172540 TO 176000.

```

2          .TITLE  TSV2 - PROGRAM HEADER
3          .SBTTL  PROGRAM HEADER
4
10         .MCALL  SVC
11 000000   SVC                ; INITIALIZE SUPERVISOR MACROS
12         .FNABLE LC
13         .NLIST  BEX,CND
19 000000   .ENABL  ABS,AMA
20         .=2000
21 002000   BGNMOD  TSV2
    002000
22
23         ;**
24         ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
25         ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
26         ;--
27
28
29 002000   POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
30 002000   HEADER  CNTSC,A,0,655.,0
    002000   L$NAME::          ;DIAGNOSTIC NAME
    002000       103         .ASCII /C/
    002001       116         .ASCII /N/
    002002       124         .ASCII /Y/
    002003       123         .ASCII /S/
    002004       103         .ASCII /C/
    002005       000         .BYTE  0
    002006       000         .BYTE  0
    002007       000         .BYTE  0
    002010   L$REV::          ;REVISION LEVEL
    002010       101         .ASCII /A/
    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::        ;POINTER TO H.W. QUES.
    002016   112370         .WORD  L$HARD
    002020   L$SPCP::        ;POINTER TO S.W. QUES.
    002020   112522         .WORD  L$SOFT
    002022   L$HPTP::        ;PTR. TO DEF. H.W. PTABLE
    002022   002146         .WORD  L$HW
    002024   L$SPTP::        ;PTR. TO S.W. PTABLE
    002024   002156         .WORD  L$SW
    002026   L$LADP::        ;DIAG. END ADDRESS
    002026   113004         .WORD  L$LAST
    002030   L$STA::         ;RESERVED FOR APT STATS
    002030   000000         .WORD  0
    002032   L$CO::         .WORD  0
    002032   000000         .WORD  0
    002034   L$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

```

PROGRAM HEADER

002042		L\$PRIO::			;DIAGNOSTIC RUN PRIORITY
002042	000000		.WORD	0	
002044		L\$ENVI::			;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000		.WORD	0	
002046		L\$EXP1::			;EXPANSION WORD
002046	000000		.WORD	0	
002050		L\$MREV::			;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	003372		.WORD	L\$DVTYP	
002062		L\$REPP::			;PTR. TO REPORT CODE
002062	023002		.WORD	L\$RPT	
002064		L\$EXP4::			
002064	000000		.WORD	0	
002066		L\$EXP5::			
002066	000000		.WORD	0	
002070		L\$AUT::			;PTR. TO ADD UNIT CODE
002070	022470		.WORD	L\$AU	
002072		L\$DUT::			;PTR. TO DROP UNIT CODE
002072	022566		.WORD	L\$DU	
002074		L\$LUN::			;LUN FOR EXERCISERS TO FILE
002074	000000		.WORD	0	
002076		L\$DESP::			;PTR. TO DIAG. DESCRIPTION
002076	003400		.WORD	L\$DESC	
002100		L\$LOAD::			;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		EMT	E\$LOAD	
002102		L\$ETP::			;PTR. TO ERRtbl
002102	000000		.WORD	0	
002104		L\$ICP::			;PTR. TO INIT CODE
002104	021646		.WORD	L\$INIT	
002106		L\$CCP::			;PTR. TO CLEAN-UP CODE
002106	022754		.WORD	L\$CLEAN	
002110		L\$ACP::			;PTR. TO AUTO CODE
002110	022674		.WORD	L\$AUTO	
002112		L\$PRT::			;PTR. TO PROTECT TABLE
002112	021636		.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

32  
33  
34  
35  
36  
37  
38  
39  
40

.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 8  
.WORD 8  
L\$DISPATCH:;  
.WORD T1  
.WORD T2  
.WORD T3  
.WORD T4  
.WORD T5  
.WORD T6  
.WORD T7  
.WORD T8

002122  
002122 000010  
002124  
002124 023564  
002126 024702  
002130 027362  
002132 034352  
002134 046466  
002136 055404  
002140 074726  
002142 104744

H2

DEFAULT HARDWARE P-TABLE

```

42          .SBTTL  DEFAULT HARDWARE P-TABLE
43
44          ;**
45          ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
46          ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
47          ; IS IDENTICAL TO THE STRUCTURE OF THE RUN TIME P-TABLE.
48          ;**
49 002144          BGNHW  DFPTBL          ;DEFAULT HARD-P-TABLE
          002144  000003          .WORD  L10000-L$HW/?
          002146
          002146
50
51 002146  176000          .WORD  176000          ; 1ST (OF 2) REGISTERS.
52 002150  000224          .WORD  224           ; INTERRUPT VECTOR
53 002152  000200          .WORD  PRI04         ; INTERRUPT PRIORITY.
54 002154
          002154          ENDPHW
          L10000:

```



SOFTWARE P-TABLE

```

56          .SBTTL  SOFTWARE P-TABLE
57
58          ;**
59          ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
60          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
61          ;**
62 002154          BGNSW  SFPTBL
002154 000004      .WORD  L10001-L$SW/2
002156          L$SW::
002156          SFPTBL::
63
64 002156 000000      TRANSTST::      .WORD  0          ; ENABLE TEST OF TRANSPRT(S) IF =1
65 002160 000000      NOITS::          .WORD  0          ; INHIBIT ITERATION OPTION.
66                                     ; ... 0 = ITERATE.
67                                     ; ...NZ = INHIBIT ITERATE.
68 002162 000017      LERRMAX::          .WORD  15.         ; LOCAL (PER TEST) ERROR LIMIT
69 002164 000310      GERRMAX::          .WORD  200.        ; GLOBAL (PER UNIT) ERROR LIMIT
70 002166          ENDSW
002166          L10001:
71
72 002166          ENDMOD

```

SOFTWARE P-TABLE

7  
8  
13  
19  
20 002166  
002166  
21  
22  
23  
24  
25  
26  
27  
28  
32 002166

.TITLE TSV3 - GLOBAL AREAS  
.SBTTL GLOBAL EQUATES SECTION

BGNMOD TSV3  
TSV3::

.SBTTL GLOBAL EQUATES SECTION

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

EQUALS ; GET STANDARD EQUATES.

; BIT DEFINITIONS

100000	BIT15--	100000
040000	BIT14--	40000
020000	BIT13--	20000
010000	BIT12--	10000
004000	BIT11--	4000
002000	BIT10--	2000
001000	BIT09--	1000
000400	BIT08--	400
000200	BIT07--	200
000100	BIT06--	100
000040	BIT05--	40
000020	BIT04--	20
000010	BIT03--	10
000004	BIT02--	4
000002	BIT01--	2
000001	BIT00--	1

BIT9-- BIT09  
BIT8-- BIT08  
BIT7-- BIT07  
BIT6-- BIT06  
BIT5-- BIT05  
BIT4-- BIT04  
BIT3-- BIT03  
BIT2-- BIT02  
BIT1-- BIT01  
BIT0-- BIT00

; EVENT FLAG DEFINITIONS  
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START--	32.	; BIT POSITION IN SECOND STATUS WORD
000037	EF.RESTART--	31.	; (100000) START COMMAND WAS ISSUED
000036	EF.CONTINUE**	30.	; (040000) RESTART COMMAND WAS ISSUED
000035	EF.NEW--	29.	; (020000) CONTINUE COMMAND WAS ISSUED
000034	EF.PWR**	28.	; (010000) A NEW PASS HAS BEEN STARTED
			; (004000) A POWER-FAIL/POWER-UP OCCURRED

GLOBAL EQUATES SECTION

```

; PRIORITY LEVEL DEFINITIONS
;
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 TDU== 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

```

33  
34 002166

```

KT11
.SBTTL MEMORY MANAGEMENT DEFINITIONS ;DEFINE MEMORY MANAGEMENT REGISTERS
;*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

```

MEMORY MANAGEMENT DEFINITIONS

```

.ENDC
;*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

```

## MEMORY MANAGEMENT DEFINITIONS

SEQ 0025

```
SDPAR3= 172266
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
.IF NB
;*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
.IF NB
;*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

```

39          .SBITL TSV05 REGISTER AND PACKET DEFINITIONS
40
41          ;
42          ; SOME GENERAL EQUATES.
43          ;
44
45          000004      ERRVEC==      4          ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.
46          000060      TTIVEC==     60          ; INTERRUPT VECTOR FOR CONSOLE INPUT
47          177560      TTICSR==    177560       ; BUS ADDRESS OF CONSOLE INPUT
48          177562      TTIBFR==    177562       ; CONSOLE INPUT DATA BUFFER
49          177520      BDVPCR==    177520       ; BDV11 PAGE CONTROL REGISTER
50
51          ;+
52          ;BIT DEFINITIONS FOR TSSR REGISTER
53          ;-
54
55          100000      SC=      BIT15          ;SPECIAL CONDITION
56          040000      BIE=     BIT14          ;BUS INTERFACE ERROR
57          020000      SCE=     BIT13          ;SANITY CHECK ERROR
58          010000      RMR=     BIT12          ;MODIFICATION REFUSED
59          004000      NXM=     BIT11          ;NONEXISTANT MEMORY ERROR
60          002000      NBA=     BIT10          ;NEED BUFFER ADDRESS
61          001400      HIADDR= BIT9:BIT8      ;EXTENDED ADDRESS BITS
62          000200      SSR=     BIT7          ;SUB SYSTEM READY
63          000100      OFL=     BIT6          ;OFF LINE BIT
64          000060      FATERR= BIT4:BIT5      ;FATAL TERMINATION ERROR CODES
65          000016      TERCLS= BIT3:BIT2:BIT1 ;TERMINATION CODES
66
67          ;+
68          ;
69          ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
70          ;(XST0)
71          ;
72          ;-
73
74          100000      XSOTMK= BIT15          ;TAPE MARK DETECTED
75          040000      XSORLS= BIT14          ;RECORD LENGTH SHORT
76          020000      XSOLET= BIT13          ;LOGICAL END OF TAPE
77          010000      XSORLL= BIT12          ;RECORD LENGTH LONG
78          004000      XSOWLE= BIT11          ;WRITE LOCK ERROR
79          002000      XSONEF= BIT10          ;NON EXECUTABLE FUNCTION
80          001000      XSOILC= BIT9          ;ILLEGAL COMMAND
81          000400      XSOILA= BIT8          ;ILLEGAL ADDRESS
82          000200      XSOMOT= BIT7          ;TAPE IN MOTION
83          000100      XSOONL= BIT6          ;TRANSPORT ON LINE
84          000040      XSOIE=  BIT5          ;INTERRUPT ENABLE
85          000020      XSOVCK= BIT4          ;VOLUME CHECK BIT
86          000010      XSOPED= BIT3          ;PHASE ENCODED DRIVE
87          000004      XSOWLK= BIT2          ;WRITE LOCKED
88          000002      XS0BOT= BIT1          ;BEGINNING OF TAPE
89          000001      XS0EOT= BIT0          ;END OF TAPE
90
91          ;+
92          ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
93          ;(XST1)
94          ;
95          ;-
96          100000      X1.DLT = BIT15          ;DATA LATE

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

96      040000      X1.SPARE = BIT14      ;NOT USED
97      020000      X1.COR  = BIT13      ;CORRECTABLE DATA ERROR
98      017375      X1.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT0 ;ALWAYS 0
99      000400      X1.RBP  = BIT8      ;READ BUS PARITY ERROR
100     000002      X1.UNC  = BIT1      ;UNCORRECTABLE DATA OR HARD ERROR
101
102
103      ;
103      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
104      ;(XST2)
105      ;
106      100000      X2.OPM  = BIT15      ;OPERATION IN PROGRESS (TAPE MOVING)
107      040000      X2.RCE  = BIT14      ;RAM CHECKSUM ERROR
108      035400      X2.SPARE = BIT13+BIT12+BIT11+BIT9+BIT8      ;NOT USED BY TSV05 (ALWAYS=0)
109      002000      X2.WCF  = BIT10     ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
110      000200      X2.EXTF = BIT7      ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
111      000100      X2.BUFE = BIT6      ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
112      00C077      X2.REV  = 000077    ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
113      000007      X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
114
115
116      ;
116      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
117      ;(XST3)
118      ;
119      177400      X3.MDE  = 177400     ;MICRO-DIAGNOSTIC ERROR CODE
120      000200      X3.SPARE = BIT7      ;NOT USED BY TSV05
121      000100      X3.OPI  = BIT6      ;OPERATION INCOMPLETE
122      000040      X3.REV  = BIT5      ;REVERSE
123      000020      X3.TRF  = BIT4      ;TRANSPORT RESPONSE FAILURE
124      000010      X3.DCK  = BIT3      ;DENSITY CHECK
125      000006      X3.MBZ  = BIT2+BIT1  ;NOT USED ALWAYS 0
126      000001      X3.RIB  = BIT0      ;REVERSE INTO BOT
127
128
129      ;
129      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
130      ;(XST4)
131      ;
132      100000      X4.HSP  = BIT15     ;HIGH SPEED
133      040000      X4.RCE  = BIT14     ;RETRY COUNT EXCEEDED
134      020000      X4.TSM  = BIT13     ;TRANSPORT SPECIAL MODE
135      017400      X4.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT8      ;NOT USED ALWAYS 0
136      000377      X4.WRC  = 000377    ;WRITE RETRY COUNT FIELD
137
138
139      ;
140      ;TSSR TERMINATION CODES (BIT 0-2)
141      ;
142      ;
143
144      000006      TSREJ = 3+2          ;COMMAND REJECTED
145      000006      UNREC = 6          ;UNRECOVERABLE ERROR
146
147
148      ;
149      ;DEVICE REGISTER OFFSETS
150      ;
151      ;
152

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

153      000000      TSBA** 0
154      000000      TSDB** 0          ;TSDB/TSBA REGISTER
155      000001      TSBAH** 1
156      000001      TSDBH** 1          ;TSDB/TSBA REGISTER HIGH BYTE
157      000002      TSSR** 2          ;TSSR REGISTER
158      000003      TSSRH** 3          ;TSSR REGISTER HIGH BYTE
159
160      ;*
161      ; TSDB ADDRESS BIT DEFINITIONS
162      ;-
163      000003      A1716 = BIT1+BIT0    ;ADDRESS BITS 17:16 ARE IN 1:0
164
165      ;*
166      ; COMMAND DEFINITIONS
167      ;-
168      000017      P.GETSTAT = 17      ;GET STATUS
169      000013      P.INIT = 13         ;INITIALIZE
170      000012      P.CONTROL = 12      ;CONTROL COMMANDS
171      000011      P.FORMAT = 11       ;FORMAT
172      000010      P.POSITION = 10     ;POSITION
173      000006      P.WRISUB = 6        ;SUBSYSTEM WRITE
174      000005      P.WRITE = 5        ;WRITE
175      000004      P.WRCHAR = 4       ;WRITE CHARACTERISTICS
176      000001      P.READ = 1         ;READ
177
178      ;*
179      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
180      ;-
181      100000      P.ACK = BIT15       ;BUFFER AVAIL FOR CONTROLLER
182      040000      P.CVC = BIT14      ;CLEAR VOLUME CHECK
183      020000      P.OPP = BIT13      ;REVERSE SEQUENCE OF DATA BITS
184      010000      P.SWB = BIT12      ;SWAP BYTES IN MEMORY
185      007400      P.MODE = BIT11!BIT10!BIT9!BIT8 ;EXTENDED COMMAND MODE FIELD
186      000200      P.IE = BIT7        ;INTERRUPT ENABLE
187      000140      P.FMT = BIT6!BITS  ;PACKET HEADER TYPE (ALWAYS=0)
188      000037      P.CMD = 37         ;MAJOR COMMAND FIELD
189
190      ;*
191      ; CONTROL COMMAND MODE CODES
192      ;-
192      000000      PC.RELEASE = 0*256. ;RELEASE BUFFER
193      000400      PC.REWIND = 1*256.  ;REWIND
194      001000      PC.NOOP = 2*256.    ;NO-OP
195      002000      PC.IEREW = 4*256.   ;REWIND IMMEDIATE INTERRUPT
196      002400      PC.ERASE = 5*256.   ;SECURITY ERASE
197
198      ;*
199      ; CONTROLLER RAM DEFINITIONS
200      ;-
201      000167      RMCHBEG = 167       ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
202      000200      RMCHEND = 200       ;CHARACTERISTICS IO DATA END RAM ADDRESS
203      000201      RMPKTBEG = 201      ;COMMAND PACKET BEGIN RAM ADDRESS
204      000210      RMPKTEND = 210      ;COMMAND PACKET END RAM ADDRESS
205      000215      RMMSGBEG = 215      ;MESSAGE BUFFER BEGIN RAM ADDRESS
206      000234      RMMSGEND = 234      ;MESSAGE BUFFER END RAM ADDRESS
207
208      ;*
209      ; REGISTER DEFINITIONS IN THE MESSAGE BUFFER

```



D5

TSV05 REGISTER AND PACKET DEFINITIONS

```

210 ;
211 ;
212 ;
213 000006 XST0** 6 ;EXTENDED STATUS REGISTER 0 (WORD 4)
214 000010 XST1** 8. ;EXTENDED STATUS REGISTER 1 (WORD 5)
215 000012 XST2** 10. ;EXTENDED STATUS REGISTER 2 (WORD 6)
216 000014 XST3** 12. ;EXTENDED STATUS REGISTER 3 (WORD 7)
217 000016 XST4** 14. ;EXTENDED STATUS REGISTER 4 (WORD 8)
218 ;
219 ;
220 ;
221 ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
222 ;
223 ;
224 ;
225 000002 PKLOW = 2 ;LOW ORDER CHARACTERISTIC DATA POINTER
226 000004 PKHI = 4 ;HIGH ORDER CHARACTERISTIC DATA POINTER
227 000006 PKBCNT = 6 ;NUMBER OF BYTES IN DATA PACKET
228 ;
229 000010 EXBCNT=10 ;NUMBER OF BYTES IN EXTENDED DATA PACKET
230 ;
231 ;
232 ;DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
233 ;
234 000000 BSELO = 0 ;BYTE 0
235 000001 BSEL1 = 1 ;BYTE 1
236 000002 SEL2 = 2 ;WORD 2
237 000004 SELDATA = 4 ;WORD 3
238 ;
239 ;
240 ;BSELO SELECT CODES FOR WRITE SUBSYSTEM COMMAND
241 ;
242 000000 PW.NOP = 0 ;NO-OP
243 000001 PW.RDRAM = 1 ;READ RAM
244 000002 PW.WTRAM = 2 ;WRITE RAM
245 000003 PW.RFIFO = 3 ;READ FIFO
246 000004 PW.WFIFO = 4 ;WRITE FIFO
247 000005 PW.RDSTAT = 5 ;READ STATUS
248 000006 PW.WCTL = 6 ;WRITE TAPE CONTROL
249 000007 PW.WFMT = 7 ;WRITE TAPE FORMAT
250 000010 PW.WMISC = 10 ;WRITE MISCELLANEOUS
251 000011 PW.WNPR = 11 ;WRITE NPR CONTROL
252 000020 PW.D22 = 20 ;DO MICROTEST 22
253 000021 PW.D11 = 21 ;DO MICROTEST 11
254 000022 PW.D13 = 22 ;DO MICROTEST 13
255 000023 PW.NO1311 = 23 ;DISABLE MICROTEST 11 AND 13
256 000024 PW.RDXT = 24 ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
257 ;
258 ;
259 ;BSEL1 CODES FOR WRITE TAPE CONTROL
260 ;
261 000200 WC.IFAD = BIT7 ;IFAD - FORMATTER ADDRESS
262 000100 WC.IOTAD = BIT6 ;ITADO - TRANSPORT ADDRESS BIT 0
263 000040 WC.I1TAD = BIT5 ;I1TAD1 - TRANSPORT ADDRESS BIT 1
264 000020 WC.ISRESV = BIT4 ;IRESV5 - RESERVED #5
265 000010 WC.IREW = BIT3 ;IREW - REWIND
266 000004 WC.IRWU = BIT2 ;IRWU - REWIND AND UNLOAD

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

267      000002      WC.IFEN      = BIT1      ;IFEN - FORMATTER ENABLE
268      000001      WC.IGO       = BIT0      ;GO
269
270      ;+
271      ;BSEL1 CODES FOR WRITE FORMAT
272      ;-
273      000200      WF.IHISP     = BIT7      ;IHISP - HIGH SPEED
274      000100      WF.IWRT     = BIT6      ;IWRT - WRITE
275      000040      WF.IREV     = BIT5      ;IREV - REVERSE
276      000020      WF.IWFM     = BIT4      ;IWFM - WRITE FILE MARK
277      000010      WF.IEDIT    = BIT3      ;IEDIT - EDIT
278      000004      WF.IERASE    = BIT2      ;IERASE - ERASE
279      000002      WF.I3RESV    = BIT1      ;IRESV3 - RESERVED #3
280      000001      WF.I4RESV    = BIT0      ;IRESV4 - RESERVED #4
281
282      ;+
283      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
284      ;-
285      000200      MS.EXT      = BIT7      ;INVERT SENSE OF EXTENDED FEATURES SWITCH
286      000020      MS.RSFIFO    = BIT4      ;RESET FIFO AND INPUT PARITY ERRCLR
287      000010      MS.RSTAPE    = BIT3      ;RESET TAPE STATUS IN 2 FLIP-FLOPS
288      000006      MS.ATTN     = BIT2:BIT1 ;ATTENTION TRIGGER FIELD
289      000001      MS.RSD      = BIT0      ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
290
291      ;+
292      ;MS.ATTN SUBCODES
293      ;-
294      000000      MSA.NOP     = 0*2      ;NO-OP (NOTHING TRIGGERED)
295      000002      MSA.VOL     = 1*2      ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
296      000004      MSA.NRAM    = 2*2      ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
297      000006      MSA.FRAME    = 3*2      ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
298
299      ;+
300      ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
301      ;-
302      000200      NP.IR       = BIT7      ;INTERRUPT REQUEST (0-1 TRANSITION)
303      000100      NP.OUT      = BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
304      000040      NP.LOOP     = BIT5      ;ENABLE TRANSPORT LOOPBACK
305      000020      NP.WRP      = BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
306
307      ;+
308      ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
309      ;-
310      000200      S2.DIM      = BIT7      ;WORD #9 BYTE 2 DATA IN MISS
311      000100      S2.ILW     = BIT6      ;ILW H
312      000040      S2.OUTRDY   = BIT5      ;OUT RDY H
313      000020      S2.INRDY   = BIT4      ;IN RDY H
314      000010      S2.ATIMR    = BIT3      ;TIMER A FLAG H
315      000004      S2.BTIMR    = BIT2      ;TIMER B FLAG H
316      000003      S2.UNDEF    = BIT1:BIT0 ;(UNDEFINED)
317      100000      S1.PARIN    = BIT15     ;WORD #8 BYTE 1 PARIN H
318      040000      S1.I2RESV   = BIT14     ;IRESV2
319      020000      S1.I1RESV   = BIT13     ;IRESV1
320      010000      S1.IEOT     = BIT12     ;IEOT L
321      004000      S1.IIDENT   = BIT11     ;IIDENT H
322      002000      S1.ICER     = BIT10     ;ICER H
323      001000      S1.IFMK     = BIT9      ;IFMK H
324      000400      S1.IHER     = BIT8      ;IHER H
325      000200      S0.ISPEED   = BIT7      ;WORD #8 BYTE 0 ISPEED H

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

324      000100      SO.IRDY      = BIT6      ;      IRDY L
325      000040      SO.IONL      = BIT5      ;      IONL L
326      000020      SO.ILDP      = BIT4      ;      ILDP L
327      000010      SO.IDBY      = BIT3      ;      IDBY L
328      000004      SO.IRWD      = BIT2      ;      IRWD L
329      000002      SO.IFBY      = BIT1      ;      IFBY L
330      000001      SO.IFPT      = BIT0      ;      IFPT L
331
332      .SBTTL      SPECIAL MACROS AND OPDEFS.
333
334      ;+
335      ;SAVE GENERAL REGS 1 TO 5
336      ;-
337
338      .MACRO      SAVREG
339      JSR      R5,REGSAV
340      .ENDM
341
342      ;+
343      ; MACRO TO FORCE AN ERROR
344      ;-
345      .MACRO      FORCERROR      TAG,NOTSSR
346      .NLIST
347      .IIF NDF LISTALL, .NLIST
348      .LIST
349      .IF B NOTSSR
350      MOV      TSSR(R5),R1      ;READ TSSR
351      .ENDC
352      MOV      FORCER,FORCER      ;IS FORCER SET? (LEAVE C BIT ALONE)
353      BNE      TAG      ;BR IF YES
354      .NLIST
355      .IIF NDF LISTALL, .LIST
356      .LIST
357      .ENDM
358
359      ;+
360      ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
361      ; WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
362      ; SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
363      ; FORCER TO 17777
364      ; TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
365      ;-
366      .MACRO      FORCEEXIT      TAG
367      .NLIST
368      .IIF NDF LISTALL, .NLIST
369      .LIST
370      MOV      FORCER,FORCER      ;IS FORCER NEGATIVE?
371      BMI      TAG      ;BR IF YES
372      .NLIST
373      .IIF NDF LISTALL, .LIST
374      .LIST
375      .ENDM
376
377      ;+
378      ; MACRO TO INCREMENT ERROR COUNTS
379      ;-
380      .MACRO      NEXT.ERRNO
381      .NLIST
382      ;;;.IIF NDF LISTALL, .NLIST

```

SPECIAL MACROS AND OPDEFS.

```

381             ERRNO=ERRNO+1
382             ;;;.IIF NDF LISTALL, .LIST
383             .LIST
384             .ENDM
385
386             ;+
387             ;MACRO TO PERFORM XOR
388             ;-
389
390             .MACRO XOR      A,B
391             MOV     A, -(SP)
392             BIC     B,(SP)
393             BIC     A,B
394             BIS     (SP)+,B
395             .ENDM
396
397             00C000             EN=0             ; INITIALIZE ERROR NUMBER
398             .SBTTL FORCER - FORCE ERROR FLAG
399
400
401             ;
402             ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
403             ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
404             ;
405             002166 000000 FORCER::          0             ; FORCE TYPE ALL HARD ERROR. (THE ONES CALLED
406             ; - BY THE MACRO "IFERROR"). AN ERROR NEED NOT -
407             ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
408             .SBTTL GLOBAL DATA SECTION
409
410             ;++
411             ;THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
412             ;IN MORE THAN ONE TEST.
413             ;--
414
415             ;
416             ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
417             ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
418             ;
419             002170 000000 EPRTSW::          .WORD 0             ;PRINT SWITCH
420             002172 000000 UNITN::          .WORD 0             ;UNIT # UNDER TEST.
421             002174 000000 QVP::           .WORD 0             ;QUICK VERIFY FLAG.
422             002176 000000 CSRADDR::       .WORD 0             ;ADDRESS OF CSR FOR CURRENT DEVICE
423             002200 000224 IVEC::          .WORD 224            ;INTERRUPT VECTOR
424             002202 000200 IPRI::          .WORD PRI04         ;INTERRUPT PRIORITY.
425             002204 000000 TSTCNT::        .WORD 0             ;NUMBER OF TESTS RUN IN THIS PASS
426             002206 000000 LOOPCNT::      .WORD 0             ;REMAINING ITERATION COUNT FOR TEST
427             002210 000000 DEVCNT::       .WORD 0             ;NUMBER OF DEVICE UNDER TEST
428             002212 000000 FATFLG::       .WORD 0             ;SET IF FATAL ERROR IS DETECTED IN TEST
429             002214 000000 INTRECV::      .WORD 0             ;SET IF TAPE INTERRUPT WAS RECEIVED
430             002216 000000 EXTFEA::       .WORD 0             ;EXTENDED FEATURES SOFTWARE SW 0=OFF;1=ON
431             002220 000000 BENBSW::        .WORD 0             ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
432             002222 000000 EXPD::          .WORD 0             ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
433             002224 000000 RECV::         .WORD 0             ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
434             002226 000000 ERRHI::         .WORD 0             ;HIGH ADDRESS MEMORY ERROR
435             002230 000000 ERRLO::         .WORD 0             ;LOW ADDRESS MEMORY ERROR
436             002232 RAMDATA::             .BLKW 16.           ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
437             002272 000000 RAMSIZ::       .WORD 0             ;RAM DATA SIZE FOR PRAMPKT ROUTINE

```

## GLOBAL DATA SECTION

```

438 002274 000000 RCVHIADD:: .WORD 0 ;RECEIVED BUFFER HIGH ADDRESS
439 002276 000000 RCVLOADD:: .WORD 0 ;RECEIVED BUFFER LOW ADDRESS
440 002300 000000 COUNT:: .WORD 0 ;TEST COUNT PATTERN
441 002302 000000 DATA:: .WORD 0 ;TEST DATA
442 002304 000000 TSTFLAG:: .WORD 0 ;TEST FLAG WORD
443 002306 000000 TSTPTR:: .WORD 0 ;TSTBLK POINTER
444 002310 000000 PRMNO:: .WORD 0 ;PRINT ROUTINE TEMP
445 002312 EXPMSG:: .BLKB 100. ;EXPECTED MESSAGE BUFFER DATA
446 002456 RECMSG:: .BLKB 100. ;RECEIVED MESSAGE BUFFER DATA
447 002622 TMPBFR:: .BLKB 80. ;TEMPORARY STORAGE FOR PRINT
448 .SBTTL TSTBLK - TEST DATA TABLE
449
450 ;*
451 ;
452 ;THIS TABLE CONTAINS TEST DATA USED IN SEVERAL TESTS
453 ;
454 ;IN SEQUENCE THE DATA IS:
455 ;
456 ; ALL ZEROS
457 ; ALL ONES
458 ; WALKING ONES
459 ; WALKING ZEROS
460 ; ALTERNATING ONES AND ZEROS
461 ;
462 ;-
463
464 002742 TSTBLK::
465 002742 000000 .WORD 0 ;ALL ZEROS
466 002744 177777 .WORD 177777 ;ALL ONES
467 002746 000001 .WORD BIT0 ;DATA FOR WALKING ONES
468 002750 000002 .WORD BIT1
469 002752 000004 .WORD BIT2
470 002754 000010 .WORD BIT3
471 002756 000020 .WORD BIT4
472 002760 000040 .WORD BIT5
473 002762 000100 .WORD BIT6
474 002764 000200 .WORD BIT7
475 002766 000400 .WORD BIT8
476 002770 001000 .WORD BIT9
477 002772 002000 .WORD BIT10
478 002774 004000 .WORD BIT11
479 002776 010000 .WORD BIT12
480 003000 020000 .WORD BIT13
481 003002 040000 .WORD BIT14
482 003004 100000 .WORD BIT15
483 003006 177776 .WORD †CBIT0 ;DATA FOR WALKING ZEROS
484 003010 177775 .WORD †CBIT1
485 003012 177773 .WORD †CBIT2
486 003014 177767 .WORD †CBIT3
487 003016 177757 .WORD †CBIT4
488 003020 177737 .WORD †CBIT5
489 003022 177677 .WORD †CBIT6
490 003024 177577 .WORD †CBIT7
491 003026 177377 .WORD †CBIT8
492 003030 176777 .WORD †CBIT9
493 003032 175777 .WORD †CBIT10
494 003034 173777 .WORD †CBIT11

```

TSTBLK TEST DATA TABLE

```

495 003036 167777 .WORD †CBIT12
496 003040 157777 .WORD †CBIT13
497 003042 137777 .WORD †CBIT14
498 003044 077777 .WORD †CBIT15
499 003046 125252 .WORD 125252 ;ALTERNATING ONES, ZEROS
500 003050 052525 .WORD 052525 ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
501 003052
502
503
504
505
506 003052 000000 100000 000000 DUMMY: 0,100000,0,0 ;DUMMY DEVICE REGISTERS...
507 003062 000000 000000 000000 0,0,0,0,0,0,0,0,0 ;...FOR MULTI-UNIT CHECKOUT.
508
509
510 003102 000000 DUFLG:: .WORD 0 ;"DROPPED UNIT" FLAG.
511 ;INHIBITS CODE IN "CLEAN-UP".
512 003104 000000 NCDEV:: .WORD 0 ;FLAG TO SAY NO DEVICE.
513
514 003106 000000 TEMP1:: .WORD 0 ;SOME TEMP LOCATIONS.
515 003110 000000 TEMP2:: .WORD 0
516 003112 000000 XXCOMM:: .WORD 0 ;XXDP* COMM BLOCK POINTER.
517 003114 000000 FREE:: .WORD 0 ;1ST FREE MEMORY ADDRESS...
518 003116 000000 FRESIZ:: .WORD 0 ;...AND SIZE (IN WORDS).
519 003120 000000 FREEHI: .WORD 0 ;LAST WORD IN FREE SPACE
520 003122 000000 KTFLG:: .WORD 0 ;KT11, MEM AVAIL FLAG -
521 ;- .WORD 0 = <24K OR NO KT -
522 ;- NZ = >24K AND KT.
523 003124 000000 KTENABLE:: .WORD 0 ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
524 003126 000000 NXMFLG:: .WORD 0 ;SET IF WE CAN TEST CLEARED OTHERWISE
525 003130 000000 NXMLU:: .WORD 0 ;NXM LO ADDRESS BITS
526 003132 000000 NXMHI:: .WORD 0 ;NXM HI ADDRESS BITS FOR DAL'S 13-21
527 003134 000000 T23A:: .WORD 0 ;11/23A FLAG
528 003136 000000 T23B:: .WORD 0 ;11/23B FLAG
529 003140 000000 T3BFLG:: .WORD 0 ;TEST 3B FLAG †0
530 003142 002000 PST32W:: .WORD 2000 ;32W BLOCK ADDRESS FOR 32K START
531 003144 000000 SIFLAG:: .WORD 0
532 003146 000000 BADDAT:: .WORD 0 ;ACTUAL DATA
533 003150 000000 GDDAT:: .WORD 0 ;EXPECTED DATA
534 003152 000000 LOOPFL:: .WORD 0
535 003154
536 003154 000000 CTAB:: .WORD 0 ;CONFIGURATION TABLES.
537 003156 000000 CTABM:: .WORD 0 ;CONFIG WORK.
538 003160 000000 .WORD 0
539 003162 000000 .WORD 0
540 003164 177777 .WORD 0
541 003166 .WORD -1 ;END OF MEM TABLE.
542
543 ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
544 ;
545 ; 0 = UNIT NOT TESTED
546 ; 100000 = UNIT ONLINE, NO ERRORS
547 ; 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
548 ; 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
549 ; 160001 = UNIT DROPPED, NOT IDLE AT START
550 ; 14XXXX = UNIT DROPPED, ENCOUNTERED XXXX ERRORS
551 003166 ERTABL: .BLKW 64.

```

03

GLOBAL ENVIRONMENT STORAGE

552 003366 000000  
553  
554 003370 000000

ERTABE: .WORD 0

SKIPT: .WORD 0

;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST

GLOBAL TEXT MESSAGES

```

556 .SBTTL GLOBAL TEXT MESSAGES
557 ;
558 ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
559 ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
560 ; MORE THAN ONE TEST.
561 ;
562 ;
563 ;
564 ; NAMES OF DEVICES SUPPORTED
565 ;
566 ;
567 003372          DEVTYP <TSV05>
003372          L$DVTYP::
003372          124      123      126          .ASCIZ *TSV05*
                    .EVEN

568 ;
583 ;
584 ; TEST DESCRIPTION
585 ;
586 003400          DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - CHK CABLES-TRANSPORT IF ERR ****>
003400          L$DESC::
003400          052      052      052          .ASCIZ /**** TSV05 LOGIC DIAGNOSTIC - CHK CABLES-TRANSPORT IF ERR ****/
                    .EVEN

594 ;
595 ;
596 ; BIT TO ASCII CONVERSION FOR TSSR REGISTER
597 ;
598 ;
599 003500 003540 003543 003547 TSSRBIT::          .WORD 1$,2$,3$,4$,5$,6$,7$,8$
600 003520 003601 003605 003611          .WORD 9$,10$,11$,12$,13$,14$,15$,16$
601 003540          123      103      000      1$:          .ASCIZ 'SC'
602 003543          102      111      105      2$:          .ASCIZ 'BIE'
603 003547          123      103      105      3$:          .ASCIZ 'SCE'
604 003553          122      115      122      4$:          .ASCIZ 'RMR'
605 003557          116      130      115      5$:          .ASCIZ 'NXM'
606 003563          116      102      101      6$:          .ASCIZ 'NBA'
607 003567          102      111      124      7$:          .ASCIZ 'BIT9'
608 003574          102      111      124      8$:          .ASCIZ 'BIT8'
609 003601          123      123      122      9$:          .ASCIZ 'SSR'
610 003605          117      106      114     10$:          .ASCIZ 'OFL'
611 003611          102      111      124     11$:          .ASCIZ 'BIT5'
612 003616          102      111      124     12$:          .ASCIZ 'BIT4'
613 003623          102      111      124     13$:          .ASCIZ 'BIT3'
614 003630          102      111      124     14$:          .ASCIZ 'BIT2'
615 003635          102      111      124     15$:          .ASCIZ 'BIT1'
616 003642          102      111      124     16$:          .ASCIZ 'BIT0'
617          .EVEN
618 003650          124      123      123     SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
619 003703          124      123      123     SFHERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
620 003736          040      040      116     NXR:      .ASCIZ / NON-EXISTANT DEVICE REGISTER/
621 003775          045      101      040     NXR:      .ASCIZ /#A ADDRESS, #06/
622 004016          045      101      040     TSSX:     .ASCIZ /#A TSBA,TSSR EXP'D: #06#A,#06#N/
623 004056          045      101      040     TSSX:     .ASCIZ /#A TSBA,TSSR REC'D: #06#A,#06/
624 004115          045      116      045     FUSI:     .ASCIZ /#N#A/
625 004121          040      040      125     USI:      .ASCIZ / UNEXPECTED INTERRUPT/
626 004150          040      040      111     NSI:      .ASCIZ / INTERRUPT EXPECTED, NOT RECEIVED/
627 004213          045      116      045     FNOINTR: .ASCIZ /#N#A/

```



## GLOBAL TEXT MESSAGES

```

628 004217      040      040      116 NOINTR: .ASCIZ / NO INTERRUPT WAS GENERATED/
629 004254      040      040      111 IFAULT: .ASCIZ / INTERRUPT FAULT/
630 004276      045      101      040 INTX: .ASCIZ /*A CPU PC: #06#A TSBA: #06/
631 004333      040      040      042 NOINIT: .ASCIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
632 004405      040      040      042 NSINIT: .ASCIZ / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
633 004455      040      040      042 BRINIT: .ASCIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
634
635 004525      000
636 004526      045      116      000 NULCR: .ASCIZ /*N/
637 004531      045      101      040 EXPGOT: .ASCIZ /*A EXP'D: #06#A, REC'D: #06/
638 004565      045      116      045 EXPGT2: .ASCIZ /*N#A EXP'D: #06#A, #06#N#A REC'D: #0#A, #06/
639 004641      045      101      040 DUAD12: .ASCIZ /*A REG(W) WRITTEN TO: #06#A REG(R) READ; EXP'D: #06#A, REC'D: #06/
640 004743      122      101      115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
641 005011      040      040      103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
642 005054      127      122      111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
643 005111      124      123      123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
644 005204      124      123      123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
645 005276      106      101      124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
646 005370      105      122      122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
647 005456      045      116      045 NOMEM: .ASCIZ '#N#A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****#N'
648 005552      045      116      045 M8186: .ASCIZ '#N#A ***** 11/23A SYSTEM *****#N'
649 005643      045      116      045 M8189: .ASCIZ '#N#A ***** 11/23B SYSTEM *****#N'
650
651
652
653
654
655
656
657
658
659 005734
005734
660 005734
005734 013746 003104
005740 012746 003775
005744 012746 000002
005750 010600
005752 104415
005754 062706 000006
661 005760 004737 005766
662 005764
005764
005764 104423
663
664
665
666
667
668 005766 005727
669 005770 000000
670 005772 001402
671 005774 004777 177770
672 006000
006000 012746 004526
006004 012746 000001
006010 010600

```

```

.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,R0
TRAP C$PNTX
ADD #6,SP
JSR PC,EXTEND ; PRINT EXTENSION IF REQUIRED.
FNDMSG
L10002: TRAP C$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,EXTA ; APPEND EXTENSION TEXT.
1$: PRINTX #NULCR ; PRINT A BLANK LINE
MOV #NULCR,-(SP)
MOV #1,-(SP)
MOV SP,R0

```

GLOBAL ERROR REPORT SECTION

006012	104415	
006014	062706	000004
673 006020	000207	

TRAP	C\$PNTX
ADD	#4,SP
RTS	PC

PRITSSR PRINT TSSR CONTENTS

```

675          .SBTTL PRITSSR - PRINT TSSR CONTENTS
676
677          ;+
678          ;
679          ;ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
680          ;THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
681          ;BY A MESSAGE PRINTING ROUTINE
682          ;
683          ;INPUTS:
684          ;
685          ;       R1       CONTENTS OF TSSR
686          ;
687          ;SUBORDINATE ROUTINES:
688          ;
689          ;       CHKAMB  CHECK FOR AMBIGUOUS CONTENTS
690          ;
691          ;-
692
693 PRITSSR:
694     006022     SAVREG                ;SAVE GENERAL REGISTERS
695     006026     010104     MOV      R1,R4          ;SAVE THE TSSR CONTENTS
696     006030     010104     PRINTB   #TSSRFOR,R4    ;PRINT THE CONTENTS OF TSSR
697     006032     010446     MOV      R4,-(SP)
698     006036     012746     006505     MOV      #TSSRFOR,-(SP)
699     006042     010600     000002     MOV      #2,-(SP)
700     006044     104414     MOV      SP,R0
701     006046     062706     000006     TRAP    C#PNTB
702     006052     010400     MOV      #6,SP
703     006054     004737     016134     MOV      R4,R0          ;GET TSSR BACK FOR CHKAMB
704     006060     103410     JSR     PC,CHKAMB      ;ARE CONTENTS AMBIGUOUS ?
705     006062     012746     006725     BCS     5$             ;BRANCH IF NOT
706     006066     012746     000001     PRINTX  #AMBTSSR      ;SHOW CONTENTS ARE AMBIGUOUS
707     006072     010600     MOV      #AMBTSSR,-(SP)
708     006074     104415     MOV      #1,-(SP)
709     006076     062706     000004     MOV      SP,R0
710     006102     010403     5$:   TRAP    C#PNTX
711     006104     042703     001476     ADD     #4,SP
712     006110     001434     MOV      R4,R3          ;CONTENTS OF TSSR
713     006112     012702     002622     BIC     #HIADDR!FATERR!TERCLS,R3 ;CLEAR ALL MULTIPLE BIT FIELDS
714     006116     012701     003500     BEQ     20$           ;NO BITS ARE SET
715     006122     005703     10$:   MOV     #TMPBFR,R2    ;TEMPORARY ASCII BUFFER
716     006124     001413     MOV     #TSSRBIT,R1    ;ASCII EQUIVALENT OF BITS
717     006126     000241     TST     R3             ;REMAINING BITS TO CONVERT
718     006130     006103     BEQ     15$           ;BRANCH WHEN ALL ARE DONE
719     006132     103006     CLC                    ;CLEAR CARRY FOR SHIFT
720     006134     011100     ROL     R3             ;SHIFT NEXT BIT TO CARRY
721     006136     112022     JCC     13$           ;BRANCH IF BIT NOT SET
722     006140     001376     11$:   MOV     (R1),R0      ;POINTER TO BIT DEFINITION
723     006142     112762     000054     MOVB   (R0)+,(R2)+    ;MOVE ASCII TO BUFFER
724     006144     005721     13$:   BNE     11$           ;MOVE ALL BITS
725     006146     105042     15$:   MOVB   #' ,-(R2)     ;INSERT A COMMA TO TERMINATE
726     006150     000763     JST     (R1)+         ;POINT TO NEXT DESCRIPTION
727     006152     006156     BR     10$           ;GET THE REMAINING BITS
728     006154     012746     002622     CLR    -(R2)         ;TERMINATE THE LINE
729     006156     012746     006676     PRINTX #TSSDEF,#TMPBFR ;PRINT THE BIT DEFINITIONS
730     006162     012746     MOV     #TMPBFR,-(SP)
731     006162     012746     MOV     #TSSDEF,-(SP)

```

PRITSSR PRINT TSSR CONTENTS

```

006166 012746 000002      MOV      #2, (SP)
006172 010600      MOV      SP,R0
006174 104415      TRAP    C#PNTX
006176 062706 000006      ADD      #6,SP
719
720 006202 010403      20$:    MOV      R4,R3          ;GET THE TSSR CONTENTS
721 006204 042703 177761      BIC      #CFATERCLS,R3 ;CLEAR ALL BUT TERMINATION
722 006210 016303 006766      MOV      TCOCOD(R3),R3 ;GET THE TERMINATION CODE MEANING
723 006214      PRINTX  #TCOASC,R3      ;PRINT THE TERMINATION CODE
      006214 010346      MOV      R3, -(SP)
      006216 012746 006566      MOV      #TCOASC, -(SP)
      006222 012746 000002      MOV      #2, -(SP)
      006226 010600      MOV      SP,R0
      006230 104415      TRAP    C#PNTX
      006232 062706 000006      ADD      #6,SP
724 006236 010403      MOV      R4,R3          ;TSSR CONTENTS AGAIN
725 006240 042703 177717      BIC      #CFATERR,R3   ;CLEAR ALL BUT FATAL TERMINATION
726 006244 001416      BEQ     25$            ;DON'T PRINT IF ZERO
727 006246 006203      ASR     R3
728 006250 006203      ASR     R3
729 006252 006203      ASR     R3
730 006254 016303 007326      MOV      TSFCOD(R3),R3 ;ALINE TERMINATION CODE FOR INDEX
731 006260      PRINTX  #TFCASC,R3    ;GET THE FATAL TERMINATION CODE
      006260 010346      MOV      R3, -(SP)
      006262 012746 006627      MOV      #TFCASC, -(SP)
      006266 012746 000002      MOV      #2, -(SP)
      006272 010600      MOV      SP,R0
      006274 104415      TRAP    C#PNTX
      006276 062706 000006      ADD      #6,SP
732 006302 042704 176377      25$:    BIC      #CHIADDR,R4   ;CLEAR ALL BUT EXTENDED ADDRESS
733 006306 001411      BEQ     30$            ;DON'T PRINT IF ZERO
734 006310      PRINTX  #TEXASC,R4    ;PRINT THE EXTENDED ADDRESS BITS
      006310 010446      MOV      R4, -(SP)
      006312 012746 006525      MOV      #TEXASC, -(SP)
      006316 012746 000002      MOV      #2, -(SP)
      006322 010600      MOV      SP,R0
      006324 104415      TRAP    C#PNTX
      006326 062706 000006      ADD      #6,SP
735 006332 013703 002170      30$:    MOV      EPRTSW,R3      ;PRINT MESSAGE BUFFER ADDRESS
736 006336      PRINTX  R3            ;PRINT PROPER MESSAGE
      006336 010346      MOV      R3, -(SP)
      006340 012746 000001      MOV      #1, -(SP)
      006344 010600      MOV      SP,R0
      006346 104415      TRAP    C#PNTX
      006350 062706 000004      ADD      #4,SP
737 006354 000207      RTS     PC              ;RETURN TO CALLER
738
749 006356      045      116      045  EPRT1:  .ASCIZ  '###A *****CHECK CABLES BETWEEN M7196 AND TRANSPORT*****'
750 006446      045      116      045  EPRT2:  .ASCIZ  '###A *****CHECK TRANSPORT*****'
756 006505      045      116      045  TSSRFOR: .ASCIZ  '###A TSSR = #06'
757 006525      045      116      045  TEXASC: .ASCIZ  '###A Extended Address Bits = #06'
758 006566      045      116      045  TCOASC: .ASCIZ  '###A Termination Class Code = #T'
759 006627      045      116      045  TFCASC: .ASCIZ  '###A Fatal Termination Class Code = #T'
760 006676      045      116      045  TSSDEF: .ASCIZ  '###A TSSR Bits Set; #T'
761 006725      045      116      045  AMBTSSR: .ASCIZ  '###A TSSR Contents Are Ambiguous'
762
763 006766 007006 007031 007057 TCOCOD: .EVEN
      .WORD  1$,2$,3$,4$,5$,6$,7$,8$

```

C4

SEQ 0041

PRITSSR PRINT TSSR CONTENTS

764	007006	116	157	162	1\$:	.ASCIZ	'Normal Termination'
765	007031	124	145	162	2\$:	.ASCIZ	'Termination Condition'
766	007057	124	14	160	3\$:	.ASCIZ	'Tape Status Alert'
767	007101	106	155	156	4\$:	.ASCIZ	'Function Reject'
768	007121	122	145	143	5\$:	.ASCIZ	'Recoverable Error - Tape Position One Record Down'
769	007203	122	145	143	6\$:	.ASCIZ	'Recoverable Error - Tape Was Not Moved'
770	007252	125	156	162	7\$:	.ASCIZ	'Unrecoverable Error'
771	007276	106	141	164	8\$:	.ASCIZ	'Fatal Controller Error'
772						.EVEN	
773							
774	007326	007336	007372	007403	TSFCOD:	.WORD	1\$,2\$,3\$,4\$
775	007336	111	156	164	1\$:	.ASCIZ	'Internal Diagnostic Failure'
776	007372	122	145	163	2\$:	.ASCIZ	'Reserved'
777	007403	102	165	163	3\$:	.ASCIZ	'Bus Interface or Sanity Check Error'
778	007447	122	145	163	4\$:	.ASCIZ	'Reserved'
779						.EVEN	
780						.SBTTL	PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
781							
782							
783							;;
784							; THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
785							; THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
786							;
787							; INPUT:
788							;
789							; R0 NUMBER OF WORDS IN PACKET
790							; R3 HIGH ORDER COMMAND PACKET ADDRESS
791							; R4 ADDRESS OF COMMAND PACKET
792							;
793							; NOTE: R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
794							;-
795	007460						PRIPKT:;
796	007460						SAVREG ;SAVE THE REGISTERS
797	007464	010005					MOV R0,R5 ;SAVE NO. OF WORDS IN PACKET
798	007466	005737	003124				TST KTENABLE ;ABOVE 28K UNDER TEST?
799	007472	001001					BNE 10\$ ;BR IF YES
800	007474	005003					CLR R3 ;SET HIGH ORDER ADDRESS TO 0
801	007476	010301			10\$:		MOV R3,R1 ;COPY HIGH ORDER ADDRESS
802	007500	010400					MOV R4,R0 ;GET LOWER ADDRESS
803	007502	006100					ROL R0 ;SHIFT BIT 15 INTO C BIT
804	007504	006101					ROL R1 ;AND INTO HIGH ORDER.
805	007506						PRINTB #PKTADD,R1,R4 ;PRINT PACKET ADDRESS
	007506	010446					MOV R4,-(SP)
	007510	010146					MOV R1,-(SP)
	007512	012746	007644				MOV #PKTADD,-(SP)
	007516	012746	000003				MOV #3,-(SP)
	007522	010600					MOV SP,R0
	007524	104414					TRAP C#PNTB
	007526	062706	000010				ADD #10,SP
806	007532	010300			15\$:		MOV R3,R0 ;GET HIGH ORDER ADDRESS
807	007534	001404					BEQ 20\$ ;BR IF NOT ABOVE 28K.
808	007536	010401					MOV R4,R1 ;GET LOW ORDER ADDRESS
809	007540	004737	017406				JSR PC,SETMAP ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
810	007544	010004					MOV R0,R4 ;GET RETURNED PAR6 ADDRESS BIAS
811	007546	005001			20\$:		CLR R1 ;SAVE WORD NUMBER
812	007550	012402			25\$:		MOV (R4),R2 ;GET PACKET CONTENTS
813	007552						PRINTB #PKTFRM,R1,R2 ;PRINT THE DATA

104

PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

```

007552 010246      MOV     R2,-(SP)
007554 010146      MOV     R1,-(SP)
007556 012746 007606  MOV     @PKTFRM,-(SP)
007562 012746 000003  MOV     @3,-(SP)
007566 010600      MOV     SP,R0
007570 104414      TRAP   C$PNTB
007572 062706 000010  ADD     @10,SP
814 007576 005201      INC     R1                ;NEXT WORD NUMBER
815 007600 020105      CMP     R1,R5            ;DONE ALL PACKET WORDS?
816 007602 002762      BLT    25$              ;LOOP TILL ALL DONE
817 007604 000207      RTS     PC                ;RETURN
818
819 007606      045      116      045  PKTFRM: .ASCIZ  '##A Packet Word #D1#A = #06'
820 007644      045      116      045  PKTADD: .ASCIZ  '##A Packet Address = #01#05'
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840 007702
841 007702
842 007706 010203
843 007710
844 007720 012700 177400
845 007724 040001
846 007726 040002
847 007730 040003
848 007732
007732 010346
007734 010146
007736 010246
007740 012746 007704
007744 012746 000004
007750 010600
007752 104414
007754 062706 000012
849 007760 010300
850 007762 000207
851
852 007764      045      116      045  XORBFOR: .ASCIZ  '##A EXPD: #03#A RECV: #03#A XOR: #03'
853
854
855

```

```

;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
;INPUTS:
;      R1      RECEIVED DATA
;      R2      EXPECTED DATA
;OUTPUT:
;      R0      XOR OF EXPECTED/RECEIVED DATA
;--
PRIBXOR:
    SA,REG                ;SAVE THE REGISTERS
    MOV     R2,R3          ;EXPECTED DATA
    XOR     R1,R3          ;FORM THE EXCLUSIVE OR
    MOV     @C<377>,R0     ;BYTE MASK
    BIC     R0,R1          ;SAVE LOW BYTE RECV
    BIC     R0,R2          ;SAVE LOW BYTE EXPD
    BIC     R0,R3          ;SAVE LOW BYTE XOR
    PRINTB @XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
    MOV     R3,-(SP)
    MOV     R1,-(SP)
    MOV     R2,-(SP)
    MOV     @XORBFOR,-(SP)
    MOV     @4,-(SP)
    MOV     SP,R0
    TRAP   C$PNTB
    ADD     @12,SP
    MOV     R3,R0          ;R0 HAS XOR ON RETURN
    RTS     PC              ;RETURN TO CALLER

```

```

;SBITL PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

```

F4

PRIXOR PRINT EXPD, RECV AND XOR

```

856 ;*
857 ;
858 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
859 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
860 ;
861 ;INPUTS:
862 ;
863 ;      R1      RECEIVED DATA
864 ;      R2      EXPECTED DATA
865 ;
866 ;OUTPUT:
867 ;
868 ;      R0      XOR OF EXPECTED/RECEIVED DATA
869 ;
870 ;-
871 ;
872 010032 PRIXOR::
873 010032      SAVREG                ;SAVE THE REGISTERS
874 010036 010203      MOV          R2,R3      ;EXPECTED DATA
875 010040      XOR          R1,R3      ;FORM THE EXCLUSIVE OR
876 010050      PRINTB     @XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
      010050 010346      MOV          R3,-(SP)
      010052 010146      MOV          R1,-(SP)
      010054 010246      MOV          R2,-(SP)
      010056 012746 010102      MOV          @XORFOR,-(SP)
      010062 012746 000004      MOV          @4,-(SP)
      010066 010600      MOV          SP,R0
      010070 104414      TRAP         C#PNTB
      010072 062706 000012      ADD          @12,SP
877 010076 010300      MOV          R3,R0      ;R0 HAS XOR ON RETURN
878 010100 000207      RTS           PC        ;RETURN TO CALLER
879 ;
880 010102 045 116 045 XORFOR: .ASCIZ '##A EXPD: #06#A RECV: #06#A XOR: #06'
881 ;
882 ;.EVEN
883 ;.SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
884 ;
885 ;*
886 ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
887 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
888 ;
889 ;INPUTS:
890 ;
891 ;      R0      OCTAL VALUE TO CONVERT
892 ;      R1      TABLE OF POINTERS TO ASCII EQUIVALENT
893 ;
894 ;-
895 ;
896 010150 PRIEQU:
897 010150      SAVREG                ;SAVE THE REGISTERS
898 010154 000207      RTS           PC        ;RETURN TO CALLER
899 ;
900 ;.SBTTL PRIRAM - PRINT RAM ADDRESS
901 ;
902 ;*
903 ;PRINT CONTROLLER RAM ADDRESS.
904 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.

```

PRIRAM PRINT RAM ADDRESS

```

905
906
907
908
909
910
911 010156
912 010156
913 010162
    010162 010446
    010164 012746 010206
    010170 012746 000002
    010174 010600
    010176 104414
    010200 062706 000006
914 010204 000207
915
916 010206 045 116 045 RAMFOR: .ASCIZ 'N/A CONTROLLER RAM ADDRESS = #06'
917 .EVEN
918
919 .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
920
921
922 ;PRINT MEMORY ADDRESS
923 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
924
925 ; IMPLICIT INPUTS
926
927 ; ERRHI - HIGH ORDER ADDRESS
928 ; ERRLO - LOW ORDER ADDRESS
929
930
931 010250
932 010250
933 010254 013700 002226
934 010260 013700 002230
935 010264 010102
936 010266 006101
937 010270 006100
938 010272
    010272 010246
    010274 010046
    010276 012746 010520
    010302 012746 000003
    010306 010600
    010310 104414
    010312 032706 000010
939 010316 000207
940
941 010320 045 116 045 PRIA0: .ASCIZ 'N/A MEMORY ERROR ADDRESS = #01#05'
942 .EVEN
943
944 .SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
945
946
947 ;PRINT MEMORY ADDRESS
948 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.

```

```

;
; INPUTS:
;
; R4 RAM ADDRESS
;
PRIRAM:
    SAVREG                ;SAVE R1-R5 UNTIL NEXT RETURN
    PRINTB #RAMFOR,R4    ;PRINT RAM ADDRESS IN ERROR
    MOV R4,-(SP)
    MOV #RAMFOR,-(SP)
    MOV #2,-(SP)
    MOV SP,R0
    TRAP C#PNTB
    ADD #6,SP
    RTS PC                ;RETURN

RAMFOR: .ASCIZ 'N/A CONTROLLER RAM ADDRESS = #06'
.EVEN

.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

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.

```



PRITADD - PRINT MEMORY TEST ADDRESS

```

949
950 ; IMPLICIT INPUTS
951 ;
952 ; ERRHI - HIGH ORDER ADDRESS
953 ; ERRLO - LOW ORDER ADDRESS
954 ;
955 ;
956 010364 PRITADD: SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
957 010364 MOV ERRHI,R2 ;GET HIGH ADDRESS
958 010370 013702 00222A MOV ERRLO,R1 ;GET LOW ADDRESS
959 010374 013701 002230 ;MOV R1,R2 ;COPY LOW ADDRESS
960 ;ROL R1 ;SHIFT BIT 15 TO C BIT
961 ;ROL R0 ;SHIFT INTO HIGH ORDER
962 ;
963 010400 PRINTB @PRIT0,R1 ;PRINT MEMORY ADDRESS LOW IN ERROR
    010400 010145 MOV R1,-(SP)
    010402 012746 010445 MOV @PRIT0,-(SP)
    010406 012746 000002 MOV @2,-(SP)
    010412 010600 MOV SP,R0
    010414 104414 TRAP C:PNTB
    010416 062706 000006 ADD @6,SP
964 010422 PRINTB @PRIT1,R2 ;PRINT MEMORY ADDRESS HIGH IN ERROR
    010422 010246 MOV R2,-(SP)
    010424 012746 010511 MOV @PRIT1,-(SP)
    010430 012746 000002 MOV @2,-(SP)
    010434 010600 MOV SP,R0
    010436 104414 TRAP C:PNTB
    01044 062706 000006 ADD @6,SP
965 010444 000207 RTS PC ;RETURN
966
967 010446 045 116 045 PRIT0: .ASCIZ 'MMA MEMORY TEST ADDRESS LOW = #06'
968 010511 045 116 045 PRIT1: .ASCIZ 'MMA MEMORY TEST ADDRESS HIGH = #06'
969 .EVEN
970 .SBTTL SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND
971
972 ;+
973 ;
974 ;ROUTINE TO ISSUE A SPACE RECORDS
975 ;COMMAND (FORWARD OR REVERSE)
976 ;
977 ;INPUT:
978 ;
979 ; R3 NUMBER OF RECORDS TO BE SPACED OVER
980 ; BIT15 CONTROLS DIRECTION
981 ; BIT15 = 0 IS FORWARD
982 ; BIT15 = 1 IS REVERSE
983 ; R5 FIRST DEVICE UNIBUS ADDRESS
984 ;
985 ; REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
986 ;
987 ;OUTPUT:
988 ;
989 ; CARRY SET - SPACE RECORDS COMMAND OK
990 ; CLR - SPACE RECORDS FAILED
991 ;
992 ;
993 ; R0 THE CONTENTS OF R4 IS MOVED TO R0

```

SPACE SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```

994 ;
995 ;
996 ;IMPLICIT OUTPUT;
997 ;
998 ; TAPE HAS BEEN MOVED
999 ;
1000 ;SIDE EFFECTS:
1001 ;
1002 ;
1003 ;-
1004 ;
1005 010556 SPACE::
1006 010556 SAVREG ;SAVE THE GENERAL REGISTERS
1007 010562 012737 000264 010750 MOV #500.,SDELAY ;SET UP DELAY
1008 010570 012737 140010 010740 MOV #140010,80$ ;SET UP COMMAND, SPACE FORWARD
1009 010576 005703 TST R3 ;CHECK FOR DIRECTION
1010 010600 100403 BMI 5$ ;BR, IF REVERSE INDICATED
1011 010602 010337 010742 MOV R3,90$ ;LOAD UP NUMBER OF RECORDS TO SPACE
1012 010606 000407 BR 10$ ;GO DO COMMAND
1013 010610 042703 170000 5$: BIC #BIT15,R3 ;CLEAR DIRECTION BIT
1014 010614 010337 010742 MOV R3,90$ ;LOAD UP NUMBER OF RECORDS TO SPACE
1015 010620 052737 000400 010740 BIS #BIT8,80$ ;SET REVERSE BIT IN COMMAND PACKET
1016 010626 012704 010730 10$: MOV #80$,R4 ;SET UP R4 WITH PACKET ADDRESS
1017 010632 010465 000000 MOV R4,TSDB(R5) ;SEND OUT COMMAND
1018 010636 004737 016340 15$: JSR PC,WAITF ;WAIT FOR SSR
1019 010642 103420 BCS 20$ ;BR, IF SSR IS SET AND OK
1020 010644 DELAY 250 ;DELAY ABOUT .25 SECONDS
    010644 012727 000250 MOV #250,(PC)+
    010650 000000 .WORD 0
    010652 013727 002116 MOV L$DLY,(PC)+
    010656 000000 .WORD 0
    010660 005367 177772 DEC -6(PC)
    010664 001375 BNE .-4
    010666 005367 177756 DEC -22(PC)
    010672 001367 BNE .-20
1021 010674 005337 010750 DEC SDELAY ;BUMP DELAY COUNTER DOWN
1022 010700 001356 BNE 15$ ;BR, IF MORE DELAY
1023 010702 000411 BR 60$ ;BR IF TROUBLE CARRY = CLEAR
1024 010704 016501 000002 20$: MOV TSSR(R5),R1 ;READ TSSR
1025 010710 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
1026 010714 020201 25$: CMP R2,R1 ;ARE THEY OK
1027 010716 001401 BEQ 40$ ;BR, IF EQUAL = OK
1028 010720 000402 BR 60$ ;TROUBLE EXIT
1029 010722 000261 40$: SEC ;SET CARRY NO TROUBLE
1030 010724 000401 BR 70$ ;EXIT
1031 010726 000241 60$: CLC ;CARRY CLEAR = ERROR
1032 010730 70$:
1033 010730 010400 MOV R4,R0 ;PASS PACKET ADDRESS
1034 010732 000207 RTS PC ;RETURN

```

SPACE SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```

1036 ;
1037 ;
1038 ;
1039 ;PACKET FOR SPACE COMMAND
1040 ;
1041 ;
1042 ;      010740      .<.*10>&177770
1043 ;
1044 ;
1045 ;COMMAND WORD
1046 010740 000000 80$: .WORD
1047 ;NUMBER OF RECORDS TO BE SPACED OVER WORD
1048 010742 000000 90$: .WORD
1049 010744 000000 .WORD
1050 010746 000000 .WORD
1051 010750 000000 SDELAY: .WORD 0 ;DELAY COUNTER
1052 .EVEN
1053 .SBTTL WRTCHR - WRITE CHARACTERISTICS COMMAND
1054
1055 ;*
1056 ;
1057 ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1058 ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1059 ;
1060 ;INPUT:
1061 ;
1062 ;      R4      ADDRESS OF PACKET FROM TEST
1063 ;      R5      FIRST DEVICE UNIBUS ADDRESS
1064 ;      REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1065 ;
1066 ;OUTPUT:
1067 ;
1068 ;      R0      TSSR CONTENTS
1069 ;      CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
1070 ;             CLR - WRITE CHARACTERISTICS FAILED
1071 ;
1072 ;IMPLICIT OUTPUT:
1073 ;
1074 ;      MESSAGE BUFFER - OTHER BUFFERS ALL SET UP
1075 ;      SOFTWARE SWITCHES SET AS FOLLOWS:
1076 ;             EXTFEA = EXTENDED FEATURES PRESENT
1077 ;             BENBSW = BUFFER ENABLE SWITCH ON OR OFF
1078 ;
1079 ;
1080 ;SIDE EFFECTS:
1081 ;
1082 ;
1083 ;-
1084 ;
1085 010752 WRTCHR::
1086 010752 ;SAVREG ;SAVE THE GENERAL REGISTERS
1087 010756 005037 002200 CLR BENBSW ;CLEAR BUFFER ENABLE SWITCH
1088 010762 005037 002210 CLR EXTFEA ;CLEAR EXTENDED FEATURES SW SWITCH
1089 010766 010465 000000 10$: MOV R4,TSDB(R5) ;SEND OUT COMMAND
1090 010772 004737 016420 JSR PC,CHKTSSR ;WAIT FOR SSR
1091 010776 1034 . BCS 20$ ;BR, IF SSR IS SET AND OK
1092 011000 000430 BR 60$ ;BR IF TROUBLE CARRY = CLEAR
1093 011002 016501 000000 20$: MOV TSSR(R5),R1 ;READ TSSR
1094 011006 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED

```

WRTECHR WRITE CHARACTERISTICS COMMAND

```

1095 011012 032701 000100 BIT 00FL,R1 ;WAS OFF LINE SET IN TSSR
1096 011016 001402 BEQ 25$ ;BR, IF NO OFL SET
1097 011020 052702 000100 BIS 00FL,R2 ;MAKE THEM LOOK ALIKE
1098 011024 020201 25$: CMP R2,R1 ;ARE THEY OK
1099 011026 001401 BEQ 40$ ;BR, IF EQUAL = OK
1100 011030 000401 BR 60$ ;TROUBLE EXIT
1101 011032 062704 000010 40$: ADD 08.,R4 ;POINT TO WRT CHARA DATA PACKET
1102 011036 011403 MOV (R4),R3 ;GET ADDRESS OF MESSAGE BUFFER
1103 011040 032763 000200 000012 BIT 0X2.EXTF,XST2(R3) ;EXTENDED FEATURES BIT SET?
1104 011046 001402 BEQ 45$ ;BR IF NO
1105 011050 005237 002216 INC EXTFEA ;SET EXTENDED FEATURES SW SWITCH
1106 011054 45$:
1107 011054 032764 000100 000012 BIT 0X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
1108 011062 001401 BEQ 50$ ;BR, IF SWITCH NOT SET
1109 011064 005237 002220 INC BENBSW ;SET SOFTWARE SWITCH FOR ENABLED
1110 011070 50$:
1111 011070 000201 SEC ;SET CARRY NO TROUBLE
1112 011072 000401 BR 70$ ;EXIT
1113 011074 000241 60$: CLC ;CARRY CLEAR = ERROR
1114 011076 016500 000002 70$: MOV TSSR(R5),R0 ;RETURN TSSR CONTENTS
1115 011102 000207 RTS PC ;RETURN
1116 .SBTTL REWIND - POSITION TAPE (REWIND) COMMAND
1117
1118 ;+
1119 ;
1120 ; THIS ROUTINE WILL REWIND THE SELECTED TAPE.
1121 ;
1122 ; CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
1123 ; TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
1124 ; SSR TO SET IN THE TSSR
1125 ;
1126 ;
1127 ; CALLING SEQUENCE:
1128 ;
1129 ; DO A SOFT INIT
1130 ; DO A WRITE CHARACTERISTICS
1131 ; JSR PC,REWIND
1132 ;
1133 ; INPUT:
1134 ;
1135 ; R5 FIRST DEVICE UNIBUS ADDRESS
1136 ;
1137 ;
1138 ; OUTPUT
1139 ;
1140 ; R0 THE CONTENTS OF R4 IS PASSED TO R0
1141 ;
1142 ;
1143 ;-
1144 REWIND::
1145 011104 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1146 011110 012704 011200 MOV 0RWPACK,R4 ;GET PACKET ADDRESS
1147 011114 010465 000000 MOV R4,TSDB(R5) ;SEND PACKET ADDRESS TO EXECUTE
1148 011120 012703 000550 MOV 0360.,R3 ;ENOUGH TIME FOR 2400' REEL TO REWIND
1149 011124 004737 016340 10$: JSR PC,WAITF ;WAIT FOR SSR TO SET
1150 011130 103417 BCS 20$ ;LEAVE WHEN SSR IS SET
1151 011132 DELAY 250. ;WAIT FOR .25 SECONDS

```

REWIND - POSITION TAPE (REWIND) COMMAND

```

011132 012727 000372      MOV      #250.,(PC)+
011136 000000              .WORD    0
011140 013727 002116      MOV      L$DLY,(PC)+
011144 000000              .WORD    0
011146 005367 177772      DEC      -6(PC)
011152 001375              BNE      .-4
011154 005367 177756      DEC      -22(PC)
011160 001367              BNE      .-20
1152 011162 005303      DEC      R3                ;BUMP COUNTER DOWN
1153 011164 001357      BNE      10$              ;KEEP GOING
1154 011166 000241      CLC                ;CLEAR CARRY TO SET ERROR
1155 011170 010400      20$: MOV      R4,R0        ;PASS THE PACKET ADDRESS
1156 011172 000207      RTS      PC            ;RETURN
1157
1159                011200
1161 011200      RWPACK: .=<.+10>&177770
1162 011200 102010      .WORD    102010          ;POSITION COMMAND (REWIND)
1163 011202 000000      .WORD    0              ;NOT USED
1164                .SBTTL  CKRAM - COMPARE RAM TO I/O PACKET
1165
1166                ;*
1167                ;
1168                ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
1169                ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
1170                ;
1171                ;INPUT:
1172                ;
1173                ;      R4      ADDRESS OF THE COMMAND PACKET
1174                ;      R5      FIRST DEVICE UNIBUS ADDRESS
1175                ;
1176                ;OUTPUT:
1177                ;
1178                ;      CARRY  SET - RAM MATCHES PACKET
1179                ;           CLR - RAM DOES NOT MATCH PACKET
1180                ;
1181                ;IMPLICIT OUTPUT:
1182                ;
1183                ;      THE TABLE RAMDATA IS FILLED WITH THE
1184                ;      DATA HELD IN RAM.
1185                ;      RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
1186                ;
1187                ;SIDE EFFECTS:
1188                ;
1189                ;      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1190                ;
1191                ;-
1192
1193 011204      CKRAM::
1194 011204      SAVREG                ;SAVE THE GENERAL REGISTERS
1195 011210 012701 002232      MOV      #RAMDATA,R1    ;ADDRESS TO SAVE THE RAM DATA
1196 011214 012702 000201      MOV      #RMPKTBEG,R2  ;BYTE ADDRESS OF FIRST RAM DATA
1197 011220 005003              CLR      R3              ;CLEAR THE ERROR FLAG
1198 011222 004737 016426      JSR      PC,CHKTSSR     ;WAIT FOR SSR
1199 011226 112765 000000 000000      MOVB    #0,TSDB(R5)    ;SET MAINTENANCE MODE
1200 011234 004737 016426      10$: JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
1201 011240 010265 000000      MOV      R2,TSDB(R5)   ;SELECT NEXT RAM ADDRESS
1202 011244 004737 016426      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET

```





CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

```

1317 011530 020427 000014      CMP      R4,#14      ;DONE FIRST 7 WORDS?
1318 011534 003764            BLE      15$         ;BR IF NO
1319 011536 032765 000200 000012  BIT      #X2.EXTF,XST2(R5); IS EXTENDED FEATURES SET IN EXPD?
1320 011544 001403            BEQ      50$         ;BR IF NO
1321 011546 020427 000016      CMP      R4,#16      ;DONE EXTENDED FEATURES WORD?
1322 011552 003755            BLE      15$         ;BR IF NO
1323 011554 005703            50$:     TST      R3          ;ANY ERRORS SEEN?
1324 011556 001402            BEQ      55$         ;BR IF NO
1325 011560 000241            CLC                      ;SET FAILURE
1326 011562 000401            BR       60$         ;
1327 011564 000261            55$:     SEC                      ;SET SUCCESS
1328 011566 000207            60$:     RTS      PC          ;RETURN
1329
1330      .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
1331
1332      ;
1333      ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
1334      ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1335      ;ERROR PRINT ROUTINES.
1336      ;
1337      ;INPUT:
1338      ;
1339      ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1340      ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1341      ;      R2      EXPD MESSAGE BUFFER ADDRESS
1342      ;      R3      NUMBER OF BYTES TO COMPARE
1343      ;
1344      ;OUTPUT:
1345      ;
1346      ;      CARRY   SET - MESSAGE BUFFERS MATCH
1347      ;              CLR - MESSAGE BUFFERS DON'T MATCH
1348      ;
1349      ;IMPLICIT OUTPUT:
1350      ;
1351      ;      EXPMSG   BUFFER IS SET TO EXPD DATA
1352      ;      RECMMSG  BUFFER IS SET TO RECV DATA
1353      ;      RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1354      ;      RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
1355      ;
1356      CKMSG2:
1357      SAVREG                      ;SAVE R1-R5 UNTIL NEXT RETURN
1358      CMP      R3,#RECMMSG-EXPMSG; @@ IS COUNT ABOVE MAX ALLOWED?
1359      BLE      5$          ;@@ BR IF NO
1360      MOV      #RECMMSG-EXPMSG,R3;@@
1361      PRINTF   #DEBUGMSG      ;@@
1362      MOV      #DEBUGMSG,-(SP)
1363      MOV      #1,-(SP)
1364      MOV      SP,R0
1365      TRAP    C:PNTF
1366      ADD     #4,SP
1367      5$:     MOV     R0,RCVHIADD      ;SAVE RECV HIGH ADDRESS
1368      MOV     R1,RCVLOADD      ;SAVE RECV LOW ADDRESS
1369      TST     KTNABLE          ;TESTING ABOVE 28K?
1370      BEQ     10$          ;BR IF NO
1371      JSR     PC,SETMAP        ;RETURN ADDRESS BIASED TO PAR6 IN RC
1372      MOV     R0,R1          ;GET RETURNED ADDRESS BIASED TO PAR6
1373      10$:    CLR     R4          ;WORD IN BUFFER

```



137

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

1369 011654 005005          CLR      R5          ;CLEAR ERROR SEEN FLAG
1370 011656 111264 002312 15%:  MOVB    (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1371 011662 111164 002456          MOVB    (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
1372 011666 122221          CMPB    (R2), (R1)    ;EXPD EQUAL RECV?
1373 011670 001401          BEQ     25%         ;BR IF YES
1374 011672 005205          INC     R5          ;SET ERROR SEEN FLAG
1375 011674 062704 000001 25%:  ADD     @1,R4        ;POINT TO NEXT BYTE
1376 011700 020403          CMB    R4,R3        ;DONE ALL BYTES?
1377 011702 002001          BGE    50%         ;BR IF YES
1378 011704 000764          BR     15%         ;DO NEXT BYTE
1379 011706 005705          50%:  TST     R5          ;ANY ERRORS SEEN?
1380 011710 001402          BEQ    55%         ;BR IF NO
1381 011712 000241          CLC                    ;SET FAILURE
1382 011714 000401          BR     60%         ;
1383 011716 000261          55%:  SEC                    ;SET SUCCESS
1384 011720 000207          60%:  RTS     PC          ;RETURN
1385
1386 011722      120      122      117  DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-' ;@@D
1387 012012      045      116      045  FERCM: .ASCII /NMA ***/
1388 012023      040      040      124  ERCM:  .ASCIZ / TSSR ERROR CODE REC'D = /
1389 012056      056      056      056  SIMSG: .ASCIZ /... AFTER DOING SOFT INIT/
1390 012111      124      105      123  TINERR: .ASCIZ /TEST: .../
1391
1392
1393
1394
1395          ;*
1396          ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
1397          ;INPUT:
1398          ;
1399          ;      R1      CONTENTS OF TSSR AT ERROR
1400          ;
1401          ;SIDE EFFECTS:
1402          ;
1403          ;      EXECUTES DROP UNIT TO CEASE TESTING
1404          ;
1405          ;-
1406
1407 012124          BGNMSG  SFIMSG
1408 012124          SFIMSG:;
1409 012130 004737 006022          JSR    PC,PRITSSR    ;PRINT CONTENTS OF TSSR REGISTER
1410 012134 004737 017272          JSR    PC,CKDROP    ;DROP UNIT, IF ALLOWED
1411          ENDMSG
1412          L10003:
1413          TRAP   CMSG
1414
1415          ;*
1416          ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1417          ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET,
1418          ;
1419          ;INPUTS:
1420          ;
1421          ;      R1      TSSR CONTENTS
1422          ;      R4      ADDRESS OF COMMAND PACKET

```

CKMSG2 COMPARE EXPD RECV MESSAGE BUFFERS

```

1423 012136          BGNMSG  PKTSSR
      012136          PKTSSR::
1424 012136 004737 006022      JSR    PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1425 012142 012700 000G04      MOV    #4,R0          ;NO. OF WORDS IN PACKET
1426 012146 004737 007460      JSR    PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1427 012152          ENDMMSG
      012152          L10004:
      012152 104423      TRAP   C$MSG

1428
1429
1430          ;*
1431          ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1432          ;TSSR AND A GET STATUS COMMAND PACKET.
1433          ;
1434          ;INPUTS:
1435          ;
1436          ;      R1      TSSR CONTENTS
1437          ;      R4      ADDRESS OF COMMAND PACKET
1438          ;
1439          ;-

1440 012154          BGNMSG  PKTGETS
      012154          PKTGETS::
1441 012154 004737 006022      JSR    PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1442 012160 012700 000002      MOV    #2,R0          ;NO. OF WORDS IN GET STATUS PACKET
1443 012164 004737 007460      JSR    PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1444 012170          ENDMMSG
      012170          L10005:
      012170 104423      TRAP   C$MSG

1445
1446
1447          ;*
1448          ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1449          ;
1450          ;INPUTS:
1451          ;
1452          ;      R1      TSSR CONTENTS
1453          ;      R4      ADDRESS OF COMMAND PACKET
1454          ;
1455          ;-

1455 012172          BGNMSG  SFFMSG
      012172          SFFMSG::
1456 012172 004737 006022      JSR    PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1457 012176          ENDMMSG
      012176          L10006:
      012176 104423      TRAP   C$MSG

1458
1459          .SBTTL  PKTMES - PRINT TSSR AND MESSAGE BUFFER
1460
1461          ;*
1462          ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
1463          ;BUFFER FOR ERROR REPORTS
1464          ;
1465          ;INPUTS:
1466          ;
1467          ;      R1      CONTENTS OF TSSR
1468          ;      R2      LOW ORDER MESSAGE BUFFER
1469          ;      R3      HIGH ORDER MESSAGE BUFFER ADDRESS
1470          ;      NOTE: R3 IS IGNORED IF KIENABLE FLAG IS CLEAR

```

D5

PKTMES PRINT TSSR AND MESSAGE BUFFER

```

1471
1472 012200
      012200
1473 012200 004737 006022
1474 012204 010200
1475 012206 010301
1476 012210 004737 014332
1477 012214
      012214
      012214 104423
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490 012216
      012216
1491 012216 004737 010364
1492 012222 016501 000002
1493 012226 004737 006022
1494 012232
      012232
      012232 104423
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508 012234
      012234
1509 012234 012700 000007
1510 012240 005737 002216
1511 012244 001402
1512 012246 012700 000010
1513 012252 004737 014642
1514 012256
      012256
      012256 104423
1515
1516
1517
1518
;
;
; BGNMSG PKTMES
PKTMES::
; JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR
; MOV R2,R0 ;LOW ORDER ADDRESS
; MOV R3,R1 ;HIGH ORDER ADDRESS
; JSR PC,PRMESS ;PRINT THE MESSAGE BUFFER
; ENDMSG
L10007:
; TRAP C#MSG
; .SBTTL ADDSSR - PRINT TEST ADDRESS AND TSSR
;
; PRINT ROUTINE TO PRINT THE CONTENTS OF
; TSSR AND A MEMORY TEST ADDRESS
;
; INPUTS:
;
; R5 FIRST DEVICE UNIBUS ADDRESS
; ERRHI HIGH ORDER MEMORY TEST ADDRESS
; ERRLO LOW ORDER MEMORY TEST ADDRESS
;
;
; BGNMSG ADDSSR
ADDSSR::
; JSR PC,PRITADD ;PRINT MEMORY TEST ADDRESS
; MOV TSSR(R5),R1 ;GET CURRENT TSSR
; JSR PC,PRITSSR ;PRINT THE CONTENTS OF TSSR REGISTER
; ENDMSG
L10010:
; TRAP C#MSG
; .SBTTL MSGEXP - PRINT WRITE CHAR. EXPD-RCV MESSAGE BUFFERS
;
; PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
;
; IMPLICIT INPUTS:
;
; EXPMSG - EXPECTED MESSAGE BUFFER
; RECMG - RECEIVED MESSAGE BUFFER
; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
;
;
; BGNMSG MSGEXP
MSGEXP::
; MOV #7,R0 ;ASSUME NO EXT FEATURES
; TST EXTFEA ;EXT FEATURES SET?
; BEQ 5$ ;BR IF NO
; MOV #8.,R0 ;EXT FEATURE BUFFER IS 8 WORDS
; JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
; ENDMSG
L10011:
; TRAP C#MSG
; .SBTTL FIFEXP - PRINT FIFO EXP/RCV DATA
;
; PRINT ROUTINE TO PRINT FIFO EXP/RCV DATA

```

F5

FIFEXP - PRINT FIFO EXP/RECV DATA

```

1519
1520
1521
1522
1523
1524
1525
1526
1527 012260
1528 012260
1529 012302
1530 012322
1531 012324
1532 012330
1533 012332
1534 012401
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549 012440
1550 012440
1551 012444
1552 012446
1553 012450
1554 012466
1555 012470

;
; R1 - BYTE COUNT
;
; IMPLICIT INPUTS:
;
; EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
; RECMSG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
;
;
; BGNMSG FIFEXP
FIFEXP:
; PRINTX #FIF1MSG,R1 ;PRINT BYTES TRANSFERRED
; MOV R1,-(SP)
; MOV #FIF1MSG,-(SP)
; MOV #2,-(SP)
; MOV SP,RO
; TRAP C#PNTX
; ADD #6,SP
; PRINTX #FIF2MSG ;PRINT HEADER MSG
; MOV #FIF2MSG,-(SP)
; MOV #1,-(SP)
; MOV SP,RO
; TRAP C#PNTX
; ADD #4,SP
; MOV R1,RO ;GET BYTE COUNT
; JSR PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
; ENDMMSG
L10012:
; TRAP C#MSG
; .ASCIZ 'NUMBER OF BYTES TRANSFERRED = #D2'
; .ASCIZ 'NUMBER OF FIFO DATA BYTES IN ERROR:'
; .EVEN
; .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
;
;
; PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
;
;
; IMPLICIT INPUTS:
;
; EXPMSG - EXPECTED MESSAGE BUFFER
; RECMSG - RECEIVED MESSAGE BUFFER
; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
;
;
; BGNMSG MSGSTAT
MSGSTAT:
; MOV #STATCOD,R1 ;ASCII ADDRESS TABLE
10$:
; MOV (R1)+,RO ;DONE ALL MSG LINES?
; BEQ 20$ ;BR IF YES
; PRINTX RO ;PRINT STATUS BIT NAMES
; MOV RO,-(SP)
; MOV #1,-(SP)
; MOV SP,RO
; TRAP C#PNTX
; ADD #4,SP
; BR 10$
; DO ANOTHER MSG LINE
20$:
; MOV #10,,RO ;NUMBER OF WORDS IN A READ STATUS BUFFER

```

MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS

```

1556 012474 004737 014642 JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
1557 012500 ENDMSG
      012500 L10013:
      012500 104423 TRAP C$MSG
1558
1559 012502 012520 012562 012653 STATCOD: .WORD 1$,2$,3$,4$,5$,6$,0
1560 012520 045 116 045 1$: .ASCIZ 'NNA Tape Bus Signals in Word #8:'
1561 012562 045 116 045 2$: .ASCIZ 'NNA PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
1562 012653 045 116 045 3$: .ASCIZ 'NNA IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
1563 012744 045 116 045 4$: .ASCIZ 'NNA IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
1564 013035 045 116 045 5$: .ASCIZ 'NNA Tape Bus Signals in Word #9:'
1565 013077 045 116 045 6$: .ASCIZ 'NNA DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
1566 .EVEN
1567
1568 .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
1569 ;*
1570 ;
1571 ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1572 ;
1573 ;IMPLICIT INPUTS:
1574 ;
1575 ; EXPMSG - EXPECTED MESSAGE BUFFER
1576 ; RECVMSG - RECEIVED MESSAGE BUFFER
1577 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1578 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1579 ;-
1580 013154 BGNMSG MSGLOOP
      013154 MSGLOOP:
1581 013154 012701 013216 MOV #LOOPCOD,R1 ;ASCII ADDRESS TABLE
1582 013160 012100 10$: MOV (R1)+,R0 ;DONE ALL MSG LINES?
1583 013162 001410 BEQ 20$ ;BR IF YES
1584 013164 PRINTX R0 ;PRINT STATUS BIT NAMES
      013164 010046 MOV RO,-(SP)
      013166 012746 000001 MOV #1,-(SP)
      013172 010600 MOV SP,R0
      013174 104415 TRAP C$PRINTX
      013176 062706 000004 ADD #4,SP
1585 013202 000766 BR 10$ ;DO ANOTHER MSG LINE
1586 013204 012700 000012 20$: MOV #10,R0 ;NUMBER OF WORDS IN A READ STATUS BUFFER
1587 013210 004737 014642 JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
1588 013214 ENDMSG
      013214 L10014:
      013214 104423 TRAP C$MSG
1589
1590 013216 013236 013311 013410 LOOPCOD: .WORD 1$,2$,3$,4$,5$,6$,7$,0
1591 013236 045 116 045 1$: .ASCIZ 'NNA Tape Bus Loopback Signals in Word #8:'
1592 013311 045 116 045 2$: .ASCIZ 'NNA PARERR<15> IRESV2<14> IRESV1<13>'
1593 013410 045 116 045 3$: .ASCIZ 'NNA IHISP->IFOT <12> IWRT->IIDENT<11> IREV ->ICER <10>'
1594 013507 045 116 045 4$: .ASCIZ 'NNA IWFM ->IFMK<09> IEDIT->IHER <08> IFAD ->ISPEED<07>'
1595 013606 045 116 045 5$: .ASCIZ 'NNA ITADO->IRDY<06> ITAD1->IONL <05> IERASE->ILDP <04>'
1596 013705 045 116 045 6$: .ASCIZ 'NNA IREW ->IDBY<03> IRWU ->IRWD <02> IFEN ->IFBY <01>'
1597 014004 045 116 045 7$: .ASCIZ 'NNA IGO ->IFPT<00>'
1598 .EVEN
1599 .SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
1600 ;*
1601 ;
1602 ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV

```

65

MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```

1603      ;
1604      ;
1605      ;IMPLICIT INPUTS:
1606      ;
1607      ;     EXPMSG - EXPECTED MESSAGE BUFFER
1608      ;     RECMMSG - RECEIVED MESSAGE BUFFER
1609      ;     RCVHIADD - RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1610      ;     RCVLOADD - RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1611      ;
1612      ;-
1612 014032      BGNMSG  MSGSUB
1613 014032      MSGSUB:  MOV     #10.,R0          ;SIZE OF WRITE SUBSYSTEM BUFFER
1614 014036 012700 000012      JSR     PC,PRMSGEXP      ;PRINT EXPD/RCV MESSAGE BUFFERS
1615 014042      ENDMSG
1615 014042      L10015:   TRAP    C$MSG
1615 014042 104423
1616      ;
1617      ;     .SBTTL  MEMADD - PRINT MEMORY ADDRESS DATA ERROR
1618      ;
1619      ;+
1620      ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
1621      ;
1622      ;IMPLICIT INPUTS:
1623      ;
1624      ;     ERRHI  - MEMORY ERROR HIGH ORDER ADDRESS
1625      ;     ERRLO  - MEMORY ERROR LOW ORDER ADDRESS
1626      ;     EXP    - EXPECTED DATA
1627      ;     RECV   - RECEIVED DATA
1628      ;
1629      ;-
1629 014044      BGNMSG  MEMADD
1630 014044      MEMADD:  JSR     PC,PRIADD      ;PRINT MEMORY ADDRESS IN ERROR
1631 014050 004737 010250      MOV     EXPD,R1          ;GET EXPD DATA
1632 014054 013701 J02222      MOV     RECV,R2         ;GET RECEIVED DATA
1633 014060 004737 010032      JSR     PC,PRIOR       ;PRINT EXPD/RCV
1634 014064      ENDMSG
1634 014064      L10016:   TRAP    C$MSG
1634 014064 104423      .SBTTL  PRAMPKT - PRINT RAM AND PACKET DATA
1635      ;
1636      ;+
1637      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1638      ;WHEN THE RAM DATA DOES NOT MATCH.
1639      ;
1640      ;
1641      ;INPUTS:
1642      ;
1643      ;     R4      POINTER TO COMMAND PACKET
1644      ;
1645      ;IMPLICIT INPUTS:
1646      ;
1647      ;     RAMDATA  DATA AS READ FROM THE RAM
1648      ;     RAMSIZ  NUMBER OF BYTES IN PACKET
1649      ;               IF RAMSIZ=0 THEN DEFAULT TO 8.
1650      ;
1651      ;IMPLICIT OUTPUTS:
1652      ;
1653      ;     RAMSIZ  SET TO 0

```

PRAMPKT - PRINT RAM AND PACKET DATA

```

1654
1655
1656 014066
1657 014066
1658 014072 012701 002232
1659 014076 005002
1660 014100 122124
1661 014102 001005
1662 014104
1663 014114 000436
1664 014116 116105 177777
1665 014122 116403 177777
1666 014126
1667 014136 042703 177400
1668 014142 116137 177777 002224
1669 014150 116437 177777 002222
1670 014156
    014156 010146
    014160 013746 002222
    014164 013746 002224
    014170 010246
    014172 012746 014246
    014176 012746 000005
    014202 010600
    014204 104414
    014206 062706 000014
1671 014212 005202
1672 014214 005737 002272
1673 014220 001404
1674 014222 020237 002272
1675 014226 003724
1676 014230 000403
1677 014232 020227 000010
1678 014236 002720
1679 014240 005037 002272
1680 014244 000207
1681
1682 014246 045 116 045 RAMASC: .ASCIZ 'N#A BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03'
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701 014332

```

```

PRAMPKT:
    SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
    MOV     #RAMDATA,R1                    ;DATA FROM THE RAM
    CLR     R2                              ;INIT BYTE NUMBER
5$:    CMPB  (R1)+,(R4)+                    ;COMPARE EXPECTED, RECEIVED
    BNE     7$                              ;BR IF NO MATCH
    FORCERROR 7$,NOTSSR
    BR      10$
7$:    MOVB  -1(R1),R5                       ;GET RECV RAM DATA
    MOVB  -1(R4),R3                          ;GET EXPD PACKET DATA
    XOR   R5,R3                              ;XOR EXPD/RECV
    BIC   #177400,R3                         ;LOW BYTE ONLY
    MOVB  -1(R1),RECV                        ;GET RECEIVED RAM DATA
    MOVB  -1(R4),EXPD                        ;GET EXPECTED RAM DATA
    PRINTB #RAMASC,R2,RECV,EXPD,R3
    MOV   R3,-(SP)
    MOV   EXPD,-(SP)
    MOV   RECV,-(SP)
    MOV   R2,-(SP)
    MOV   #RAMASC,-(SP)
    MOV   #5,-(SP)
    MOV   SP,R0
    TRAP  C#PNTB
    ADD   #14,SP
10$:   INC  R2                                ;UPDATE BYTE COUNT
    TST  RAMSIZ                              ;DEFAULT TO 8.?
    BEQ  15$                                  ;BR IF YES
    CMP  R2,RAMSIZ                          ;DONE ALL BYTES?
    BLE  5$                                  ;BR IF NO
    BR   25$
15$:   CMP  R2,#8.                            ;DONE DEFAULT NUMBER OF BYTES?
20$:   BLT  5$                                  ;BR IF NO
25$:   CLR  RAMSIZ                            ;SET DEFAULT RAMSIZ
    RTS   PC                                  ;RETURN

```

```

;+
;
;THIS ROUTINE PRINTS THE CONTENTS OF
;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE
;TSV-05.
;
;INPUT:
;
;    R0    LOW ORDER ADDRESS OF MESSAGE BUFFER
;    R1    HIGH ORDER ADDRESS OF MESSAGE BUFFER
;    NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
;
;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
;
;
PRMESS:

```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

1702 014332          SAVREG          ;SAVE THE REGISTERS
1703 014336 010005  MOV      R0,R5    ;SAVE LOW ORDER ADDRESS
1704 014340 005737 003124  TST      KTENABLE  ;ADDRESS ABOVE 28K?
1705 014344 001001          BNE      10$      ;BR IF YES
1706 014346 005001          CLR      R1        ;SET HIGH ORDER ADDRESS TO 0
1707 014350 010103          10$: MOV     R1,R3      ;SAVE HIGH ORDER ADDRESS
1708 014352 006100          ROL      RC        ;SHIFT BIT15 TO C BIT
1709 014354 006101          ROL      R1        ;SHIFT TO HIGH ORDER FOR PRINTOUT
1710 014356          PRINTX  @PROASC,R1,R5 ;PRINT MESSAGE BUFFER ADDRESS
      014356 010546  MOV      R5,-(SP)
      014360 010146  MOV      R1,-(SP)
      014362 012746 014510  MOV     @PROASC,-(SP)
      014366 012746 000003  MOV     @3,-(SP)
      014372 010600  MOV     SP,R0
      014374 104415  TRAP    C$PNTX
      014376 062706 000010  ADD     @10,SP
1711 014402          PRINTX  @PRIASC          ;PRINT HEADER FOR CONTENTS
      014402 012746 014555  MOV     @PRIASC,-(SP)
      014406 012746 000001  MOV     @1,-(SP)
      014412 010600  MOV     SP,R0
      014414 104415  TRAP    C$PNTX
      014416 062706 000004  ADD     @4,SP
1712 014422 005004          CLR      R4        ;NUMBER OF THE NEXT WORD
1713 014424 010501          MOV     R5,R1      ;COPY LOW ORDER ADDRESS
1714 014426 010300          MOV     R3,R0      ;COPY HIGH ORDER ADDRESS
1715 014430 001403          BEQ     20$      ;BR IF NOT ABOVE 28K
1716 014432 004737 017406  JSR     PC,SETMAP  ;SETUP PAR ADDRESS IN R0
1717 014436 010005          MOV     R0,R5      ;GET PAR FORMAT ADDRESS ABOVE 28K
1718 014440          20$: PRINTX  @PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
      014440 012546  MOV     (R5)+,-(SP)
      014442 010446  MOV     R4,-(SP)
      014444 012746 014613  MOV     @PRASC,-(SP)
      014450 012746 000003  MOV     @3,-(SP)
      014454 010600  MOV     SP,R0
      014456 104415  TRAP    C$PNTX
      014460 062706 000010  ADD     @10,SP
1719 014464 005204          INC     R4        ;NUMBER OF THE NEXT
1720 014466 020427 000007  CMP     R4,@7      ;DONE ALL YET ?
1721 014472 003005          BGT     50$      ;BRANCH IF ALL DONE
1722 014474 002761          BLT     20$      ;PRINT FIRST 7 WORDS
1723 014476 032763 000200 000012  BIT     @X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
1724 014504 001355          BNE     20$      ;PRINT EXTENDED STATUS WORD
1725 014506 000207          50$: RTS      PC        ;RETURN
1726
1727 014510          045      116      045  PROASC: .ASCIZ  '##A Message Buffer Address = #01#05'
1728 014555          045      116      045  PRIASC: .ASCIZ  '##A Message Buffer Contents:'
1729 014613          045      116      045  PRASC:  .ASCIZ  '##A Word D1#A: #0'
1730          .EVEN
1731          .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
1732          ;
1733          ;
1734          ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
1735          ;
1736          ;      RO      - NUMBER OF WORDS IN BUFFER
1737          ;
1738          ;IMPLICIT INPUTS:
1739          ;

```



JB

PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS

```

1740 ; EXPMSG - EXPECTED MESSAGE BUFFER
1741 ; RECMMSG - RECEIVED MESSAGE BUFFER
1742 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1743 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1744 ;
1745 014642 PRMSGEXP::
1746 014642 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1747 014646 010005 MOV R0,R5 ;SAVE NUMBER OF WORDS
1748 014650 013700 002276 MOV RCVLOADD,R0 ;GET RECV LOW ADDRESS
1749 014654 010004 MOV R0,R4 ;COPY LOW ADDRESS
1750 014656 013701 002274 MOV RCVHIADD,R1 ;GET RECV HIGH ADDRESS
1751 014662 006100 ROL R0 ;SHIFT BIT15 TO C BIT
1752 014664 006101 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
1753 014666 PRINTX #PRMSG0,R1,R4 ;PRINT MESSAGE BUFFER ADDRESS
      014666 010446 MOV R4,-(SP)
      014670 010146 MOV R1,-(SP)
      014672 012746 015022 MOV #PRMSG0,-(SP)
      014676 012746 000003 MOV #3,-(SP)
      014702 010600 MOV SP,R0
      014704 104415 TRAP C#PNTX
      014706 062706 000010 ADD #10,SP
1754 014712 PRINTX #PRMSG1 ;PRINT HEADER FOR CONTENTS
      014712 012746 015067 MOV #PRMSG1,-(SP)
      014716 012746 000001 MOV #1,-(SP)
      014722 010600 MOV SP,R0
      014724 104415 TRAP C#PNTX
      014726 062706 000004 ADD #4,SP
1755 014732 005004 CLR R4 ;NUMBER OF THE CURRENT WORD
1756 014734 012701 002312 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
1757 014740 012702 002456 MOV #RECMMSG,R2 ;GET RECV BUFFER ADDRESS
1758 014744 011100 20$: MOV (R1),R0 ;GET EXPD
1759 014746 011203 MOV (R2),R3 ;GET RECV
1760 014750 XOR R0,R3 ;XOR EXPD/RECV
1761 014760 PRINTX #PRMSG2,R4,(R1)+,(R2)+,R3
      014760 010346 MOV R3,-(SP)
      014762 012246 MOV (R2)+,-(SP)
      014764 012146 MOV (R1)+,-(SP)
      014766 010446 MOV R4,-(SP)
      014770 012746 015125 MOV #PRMSG2,-(SP)
      014774 012746 000005 MOV #5,-(SP)
      015000 010600 MOV SP,R0
      015002 104415 TRAP C#PNTX
      015004 062706 000014 ADD #14,SP
1762 015010 INC R4 ;NUMBER OF THE NEXT
1763 015012 020405 CMP R4,R5 ;DONE ALL YET?
1764 015014 002001 BGE 50$ ;BR IF YES
1765 015016 000752 BR 20$ ;DO ANOTHER
1766 015020 000207 50$: RTS PC ;RETURN
1767
1768 015022 045 116 045 PRMSG0: .ASCIIZ '###A Message Buffer Address = #01#05'
1769 015067 045 116 045 PRMSG1: .ASCIIZ '###A Message Buffer Contents.'
1770 015125 045 116 045 PRMSG2: .ASCIIZ '###A WORD #D2#A EXPD: #06#A RECV: #06#A XOR: #06#
1771 .EVEN
1772 .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
1773 ;
1774 ;
1775 ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS

```

15

PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

1776 ; ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
1777 ;
1778 ; R0 - NUMBER OF BYTES IN BUFFER
1779 ;
1780 ; IMPLICIT INPUTS:
1781 ;
1782 ; EXPMSG - EXPECTED MESSAGE BUFFER
1783 ; RECMMSG - RECEIVED MESSAGE BUFFER
1784 ;
1785 PRBYTEXP::
1786 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1787 MOV R0,R5 ;SAVE NUMBER OF BYTES
1788 CLR PRMNO ;INIT ERROR COUNT
1789 CLR R4 ;NUMBER OF THE CURRENT BYTE
1790 MOV @EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
1791 MOV @RECMMSG,R2 ;GET RECV BUFFER ADDRESS
1792 20$: MOVB (R1),R0 ;GET EXPD BYTE
1793 BIC @C<377>,R0 ;CLEAR UPPER BYTE
1794 MOVB R0,PRBEXP ;SAVE FOR ERROR REPORT
1795 MOVB (R2),R3 ;GET RECV BYTE
1796 BIC @C<377>,R3 ;CLEAR UPPER BYTE
1797 MOVB R3,PRBREC ;FOR ERROR REPORT
1798 XOR R0,R3 ;XOR EXPD/RECV
1799 CMPB (R1)+,(R2)+ ;EXPD = RECV?
1800 BEQ 30$ ;BR IF YES
1801 INC PRMNO ;UPDATE ERROR COUNT
1802 1802 015302 023727 002310 000010 CMP PRMNO,@8. ;PRINTED 8?
1803 BHI 30$ ;BR IF YES
1804 27$: PRINTX @PRBMSG,R4,PRBEXP,PRBREC,R3
1805 015312 010346 MOV R3,-(SP)
1806 015314 013746 015562 MOV PRBREC,-(SP)
1807 015320 013746 015560 MOV PRBEXP,-(SP)
1808 015324 010446 MOV R4,-(SP)
1809 015326 012746 015426 MOV @PRBMSG,-(SP)
1810 015332 012746 000005 MOV @5,-(SP)
1811 015336 010600 MOV SP,R0
1812 015340 104415 TRAP C$PNTX
1813 015342 062706 000014 ADD @14,SP
1814 1805 015346 FORCEXIT 50$ ;000
1815 1806 015356 000404 BR 35$ ;000
1816 1807 015360 30$: FORCERROR 27$,NOT$SR ;000
1817 1808 015360 35$: ;000
1818 1809 015370 005204 INC R4 ;NUMBER OF THE NEXT
1819 1810 015372 020405 CMP R4,R5 ;DONE ALL YET?
1820 1811 015374 002001 BGE 50$ ;BR IF YES
1821 1812 015376 000717 BR 20$ ;DO ANOTHER
1822 1813 015400 50$: PRINTX @PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
1823 015400 013746 002310 MOV PRMNO,-(SP)
1824 015404 012746 015513 MOV @PRBTOT,-(SP)
1825 015410 012746 000002 MOV @2,-(SP)
1826 015414 010600 MOV SP,R0
1827 015416 104415 TRAP C$PNTX
1828 015420 062706 000006 ADD @6,SP
1829 1815 015424 000207 RTS PC ;RETURN
1830 1816
1831 1817 015426 045 116 045 PRBMSG: .ASCIZ 'N%A BYTE @D2%A EXPD: @03%A RECV: @03%A XOR: @03'

```

PRBYTEXP PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

1818 015513    045    116    045 PRBTOT: .ASCIZ  'N#A NUMBER OF BYTES IN ERROR = #D2'
1819                                     .EVEN
1820 015560  000000 PRBEXP: .WORD  0          ;EXPD
1821 015562  000000 PRBREC: .WORD  0          ;RECV
1822                                     .SBTTL  EXPREC - PRINT EXPD/RECV WORD DATA
1823                                     ;+
1824                                     ;
1825                                     ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1826                                     ;
1827                                     ;INPUTS:
1828                                     ;
1829                                     ;      R1      RECEIVED DATA
1830                                     ;      R2      EXPECTED DATA
1831                                     ;
1832                                     ;-
1833
1834 015564          BGNMSG  EXPREC
1835 015564  004737  010032 EXPREC:; JSR      PC,PRIXOR          ;PRINT THE DATA
1836 015570          ENDMSG
1837 015570  104423 L10017: TRAP    C$MSG
1838                                     .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
1839                                     ;+
1840                                     ;
1841                                     ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
1842                                     ;
1843                                     ;INPUTS:
1844                                     ;
1845                                     ;      R1      RECEIVED DATA BYTE
1846                                     ;      R2      EXPECTED DATA BYTE
1847                                     ;
1848                                     ;-
1849
1850 015572          BGNMSG  EXPBREC
1851 015572  004737  007702 EXPBREC:; JSR      PC,PRIBXOR        ;PRINT THE DATA
1852 015576          ENDMSG
1853 015576  104423 L10020: TRAP    C$MSG
1854                                     .SBTTL  RAMERR - PRINT RAM AND PACKET DATA
1855                                     ;
1856                                     ;
1857                                     ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1858                                     ;
1859                                     ;INPUTS:
1860                                     ;
1861                                     ;      R4      POINTER TO COMMAND PACKET
1862                                     ;
1863                                     ;IMPLICIT INPUTS:
1864                                     ;
1865                                     ;      RAMDATA  DATA AS READ FROM THE RAM
1866                                     ;      RAMSIZ   NUMBER OF BYTES IN PACKET
1867                                     ;              IF RAMSIZ=0 THEN DEFAULT TO 8.
1868

```

RAMERR PRINT RAM AND PACKET DATA

```

1869
1870 ;IMPLICIT OUTPUTS:
1871 ;
1872 ; RAMSIZ SET TO 0
1873 ;
1874 ;
1875 015600 BGNMSG RAMERR
015600
1876 015600 004737 014066 RAMERR::
1877 015604 JSR PC,PRAMPKT ;PRINT RAM/PACKET DATA
015604 ENDMMSG
L10021:
015604 104423 TRAP C$MSG
;
; .SBTTL RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
;
; *
;
; PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
;
; INPUTS:
;
; R4 POINTER TO COMMAND PACKET
;
; IMPLICIT INPUTS:
;
; RAMDATA DATA AS READ FROM THE RAM
; RAMSIZ NUMBER OF BYTES IN PACKET
; IF RAMSIZ=0 THEN DEFAULT TO 8.
; ERRHI HIGH ORDER TEST ADDRESS
; ERRLO LOW ORDER TEST ADDRESS
;
; IMPLICIT OUTPUTS:
;
; RAMSIZ SET TO 0
;
;
1901 015606 BGNMSG RAMTADD
015606
1902 015606 004737 010364 RAMTADD::
1903 015612 004737 014066 JSR PC,PRITADD ;PRINT TEST ADDRESS
1904 015616 JSR PC,PRAMPKT ;PRINT RAM/PACKET DATA
015616 ENDMMSG
L10022:
015616 104423 TRAP C$MSG
;
; .SBTTL RAMEXP - PRINT RAM EXPD/RECV DATA
;
; *
;
; PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
;
; INPUTS:
;
; R1 RECEIVED DATA
; R2 EXPECTED DATA
; R4 CONTROLLER RAM ADDRESS
;
;
1918 015620 BGNMSG RAMEXP
015620 RAMEXP::

```

RAMEXP PRINT RAM EXPD/RECV DATA

```

1919 015620 042701 177400      BIC    #+C<377>,R1      ;SAVE EXPD RAM DATA BYTE
1920 015624 042702 177400      BIC    #+C<377>,R2      ;SAVE EXPD RAM DATA BYTE
1921 015630 004737 010156      JSR    PC,PRIRAM        ;PRINT THE RAM ADDRESS
1922 015634 004737 010032      JSR    PC,PRIXOR        ;PRINT THE DATA
1923 015640      ENDMSG
      015540
      015640 104423
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937 015642      BGNMSG  TIMEXP
      015642
1938 015642      TIMEXP:  PRINTX  #TIMSGO      ;PRINT HEADER
      015642 012746 015670      MOV    #TIMSGO,-(SP)
      015646 012746 000001      MOV    #1,-(SP)
      015652 010600      MOV    SP,R0
      015654 104415      TRAP  C$PNIX
      015656 062706 000004      ADU    #4,SP
1939 015662 004737 010032      JSR    PC,PRIXOR        ;PRINT THE DATA
1940 015666      ENDMSG
      015666
      015666 104423
1941
1942 015670      L10024:  TRAP    C$MSG
      045      116      045  TIMSGO: .ASCIZ  'NNA TIMER A STATUS IS IN BIT 3NNA TIMER B STATUS IS IN BIT 2'
1943      .EVEN
1944      .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957 015770      BGNMSG  BADSSR
      015770
1958 015770 010246      BADSSR:  MOV    R2,-(SP)      ;SAVE DATA TRANSFERRED
1959 015772 042702 177400      BIC    #177400,R2      ;GET JUST ONE BYTE
1960 015776      PRINTB  #XFERASC,R2
      015776 010246      MOV    R2,-(SP)
      016000 012746 016030      MOV    #XFERASC,-(SP)
      016004 012746 000002      MOV    #2,-(SP)
      016010 010600      MOV    SP,R0

```

136

BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

1961	016012	104414				TRAP	C#PNTB	
1962	016014	062706	000006			ADD	#6,SP	
1963	016020	012602				MOV	(SP),R2	;RESTORE R2
1964	016022	004737	006022			JSR	PC,PRITSSR	;DECODE TSSR CONTENTS
1965	016026					ENDMSG		
1966	016026				L10025:			
1967	016026	104423				TRAP	C#MSG	
1968	016030	045	116	045	XFERASC:	,ASCIIZ	'#N#A Data Transferred = #03'	
1969						,SBTTL	GLOBAL SUBROUTINES SECTION	
1970								
1971								
1972								
1973								
1974								
1975								
1976								
1977								
1978								
1979								
1980								
1981								
1982								
1983								
1984								
1985								
1986								
1987								
1988								
1989								
1990								
1991								
1992								
1993								
1994								
1995								
1996								
1997								
1998								
1999	016064							
2000	016064							
2001	016070	012765	000000	000002				
2002	016076	004737	016340					
2003	016102	016500	000002					
2004	016106	010004						
2005	016110	042704	176277					
2006	016114	052704	002200					
2007	016120	020400						
2008	016122	001402						
2009	016124	000241						
2010	016126	000401						
2011	016130	000261						
2012	016132	000207						
2013								

```

; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
; THAT ARE USED IN MORE THAN ONE TEST.
;--
.SBTTL SOFINIT - SOFT INITIALIZE OF CONTROLLER
;+
; ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
; BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
; THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
; DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
; INPUTS:
;     R5      ADDRESS OF FIRST REGISTER
; OUTPUTS:
;     R1      CONTENTS OF TSSR, IF ERROR
;     CARRY   SET IF INIT WAS OKAY
;             CLEAR IF FATAL ERROR
; CALLING SEQUENCE:
;     MOV     #ADDRESS,R5
;     JSR    PC,SOFINIT
;     BCS    CONTINUE
;     ERDF   ;REPORT FATAL ERROR
;--
SOFINIT:
    SAVREG      ; SAVE THE REGISTERS
    MOV     #0,TSSR(R5) ; DO THE INIT.
    JSR    PC,WAITF ; WAIT FOR SSR
    MOV     TSSR(R5),R0 ;GET THE TSSR REGISTER
    MOV     R0,R4 ;TSSR CONTENTS
    DIC     #C<HIADDR!OFL>,R4
    BIS     #SSR!NBA,R4 ;R4 HAS EXPECTED CONTENTS
    CMP     R4,R0 ;ONLY EXPECTED BITS SET ?
    BEQ    5f ;BRANCH IF OKAY
    CLC     ;CLEAR THE CARRY FOR ERROR
    BR     10f ;GO TO EXIT
5f: SEC ;SET THE CARRY BIT
10f: RTS PC ;RETURN TO CALLER
.SBTTL CHKAMB - CHECK TSSR FOR AMBIGUITY

```

CHKAMB CHECK TSSR FOR AMBIGUITY

2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033 016134  
2034 016134  
2035 016140 010004  
2036 016142 032700 100000  
2037 016146 001004  
2038 016150 032700 174077  
2039 016154 001023  
2040 016156 000424  
2041 016160 032700 000200  
2042 016164 001011  
2043 016166 032700 000040  
2044 016172 001414  
2045 016174 042704 177761  
2046 016200 020427 000016  
2047 016204 001007  
2048 016206 000410  
2049 016210 032700 000040  
2050 016214 001405  
2051 016216 032700 000006  
2052 016222 001002  
2053 016224 000241  
2054 016226 000401  
2055 016230 000261  
2056 016232 000207  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066 000200  
2067 000001  
2068  
2069  
2070 016234 000

```

;
; THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
; FOR AMBIGUITY
; INPUT:
;     R0      CONTENTS OF TSSR
; OUTPUT:
;     R0      CONTENTS OF TSSR
;     CARRY   SET - NO AMBIGUITY
;             CLR - AMBIGUOUS CONTENTS
;
CHKAMB:
    SAVREG                ;SAVE THE GENERAL REGISTERS
    MOV     R0,R4         ;CONTENTS OF TSSR
    BIT    05C,R0        ;IS BIT 15 SET ?
    BNE    5$            ;BRANCH IF YES
    BIT    0+C<NBA!OFL!SSR!HIADDR>,R0 ;ANY OTHER BITS SET ?
    BNE    40$           ;MUST BE AN ERROR
    BR     45$           ;RETURN WITH SUCCESS
5$:    BIT    05SR,R0    ;IS READY BIT SET ?
    BNE    10$         ;BRANCH IF READY BIT IS SET.
    BIT    0BITS,R0    ;IS FATAL ERROR BIT SET ?
    BEQ    40$         ;ERROR IF NOT
    BIC    0+CTERCL5,R4 ;CLEAR ALL BUT TERMINATION CODE
    CMP    R4,016      ;ALL THREE BITS MUST BE SET
    BNE    40$         ;ERROR IF NOT SET
    BR     45$         ;OK IF ALL ARE SET
10$:   BIT    0BITS,R0 ;IS FATAL ERROR BIT SET ?
    BEQ    45$         ;ERROR IF BIT IS SET WITH SSR
    BIT    0BIT2!BIT1,R0 ;IS THIS A FUNCTION REJECT
    BNE    45$         ;BR, IF TSSR IS OK
40$:   CLC                ;AMBIGUOUS CONTENTS
    BR     50$
45$:   SEC                ;SHOW SUCCESS - NO AMBIGUITY
50$:   RTS     PC         ;RETURN TO CALLER
    .SBTTL ENAINI,DSBINT - ENABLE/DISABLE INTERRUPTS
;
; DEFAULT DISPLAY INTERRUPT HANDLERS.
; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
;
; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
;
;     IOKCKIN=BIT7    ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
;     IOKSTP=BIT0     ; EXPECT "STOP" INTERRUPT.
;
; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
INTMASK: .BYTE 0
    
```

(D)

ENAIN,DSBINT - ENABLE/DISABLE INTERRUPTS

```

2071                                ; INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2072 016235      000                INTFLAG: .BYTE 0
2073
2074                                ; SAVED INTERRUPT VECTOR:
2075 016236      000000             INTVEC: .WORD 0
2076                                ; SAVE CPU PC
2077 016240      000000             INTCP: .WORD 0
2078
2079                                ; SUBROUTINE TO ENABLE INTERRUPTS:
2080 016242      010046             ENAIN: MOV  RO,-(SP)          ;SAVE RO
2081 016244      013700      002200      MOV  IVEC,RO          ;GET POINTER TO VECTORS
2082 016250      012720      016306      MOV  @INTR,(RO),      ;SET UP INTERRUPT VECTOR
2083 016254      012720      000300      MOV  @PRIORITY,(RO),
2084 016260      012600             MOV  (SP),RO          ;RESTORE RO
2085 016262      011646             MOV  (SP),-(SP)
2086 016264      012766      000000      000002      MOV  @0,2(SP)        ;SET CPU TO LEVEL 0
2087 016272      000002             RTI
2088
2089                                ; SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 6)
2090 016274      011646             DSBINT: MOV (SP),-(SP)
2091 016276      012766      000300      000002      MOV  @PRIORITY,2(SP)
2092 016304      000002             RTI
2093                                .SBTTL INTR - INTERRUPT HANDLERS
2094
2095 016306             BGNSRV INTR          ;DEFINE INTERRUPT ENTRY
2096 016306      012737      000001      002214      INTR:: MOV  @1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
2097 016314      105037      016235      CLR  INTFLAG        ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2098 016320      132737      000001      016234      BIT  @IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2099 016326      001003             BNE  1$             ;BR IF YES
2100 016330      152737      000001      016235      BIS  @IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2101
2102                                ;SAVE REGISTERS, MSG BUFFER, ETC.
2103 016336             1$:
2104 016336             ENDSRV
2105 016336             L10026:
2106 016336      000002             RTI
2107                                .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
2108
2109                                ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2110                                ; INPUTS:
2111                                ; R5 ADDRESS OF FIRST DEVICE REGISTER
2112                                ;
2113                                ; OUTPUTS:
2114                                ;
2115                                ; R0 CONTENTS OF LAST TSSR READ
2116                                ; CARRY SET - READY BIT SET
2117                                ; CLR - TIMEOUT WAITING FOR READY
2118
2119 016340      000401             WAITF:: BR  1$             ;NOP WHEN SUPER FIXED
2120 016342      104422             BREAK TRAP C$BRK      ; DO A SUPVSR BREAK FIRST.
2121 016344      012746      011000      1$: MOV  @11000,-(SP)    ;25-APRIL-83 REV B - 1100 MSEC TIMER
2122 016350      016500      000002      2$: MOV  TSSR(R5),R0    ;READ THE TSSR REGISTER
2123 016354      105700             TSTB R0              ;TEST FOR READY BIT SET

```



[E]

WAITF - WAIT FOR SUBSYSTEM READY

```

2124
2125 016356 100420
2126 016360
      016360 012727 000001
      016364 000000
      016366 013727 002116
      016372 000000
      016374 005367 177772
      016400 001375
      016402 005367 177756
      016406 001367
2127 016410 005316
2128 016412 001356
2129 016414 000241
2130 016416 000401
2131 016420 000261
2132 016422 005326
2133 016424 000207
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153 016426
2154 016426 004737 016340
2155 016432 103014
2156 016434 004737 016134
2157 016440 103006
2158 016442 032700 100000
2159 016446 001405
2160 016450 032700 074000
2161 016454 001402
2162 016456 000241
2163 016460 000401
2164 016462 000261
2165 016464 000207
2166
2167
2168
2169
2170
2171
2172

```

```

      BMI      3$
      DELAY   1
      MOV     01,(PC)+
      .WORD   0
      MOV     L$DLY,(PC)+
      .WORD   0
      DEC     -6(PC)
      BNE     -.4
      DEC     -22(PC)
      BNE     .-20
      DEC     (SP)
      BNE     2$
      CLC
      BR      4$
3$: SEC
4$: DEC (SP)
   RTS PC
   .SBTTL  CHKTSSR - CHECK TSSR FOR READY

;
; THIS ROUTINE WAITS FOR READY IN THE TSSR
; AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
; INPUT:
;
;   RS      ADDRESS OF CSR REGISTERS
; OUTPUT:
;
;   RO      CONTENTS OF TSSR
;   CARRY   SET - OKAY
;           CLR - NOT READY AMBIGUOUS, OR SC SET
;
;
CHKTSSR:
      JSR     PC,WAITF
      BCC    20$
      JSR     PC,CHKAMB
      BCC    10$
      BIT    0$SC,RO
      BEQ    15$
      BIT    0$<SCE!BIE!RMR!NXM>,RO
      BEQ    15$
10$: CLC
      BR     20$
15$: SEC
20$: RTS PC
   .SBTTL  XNXM - CHECK FOR NONEXISTENT MEMORY

;
; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
; "C" = 0, ALL ADDRESSES OK.
;
;CALL: MOV ADR1,R1

```

16

XNXM CHECK FOR NONEXISTENT MEMORY

```

2173      ;      MOV ADR2,R2
2174      ;      JSR PC,NXM
2175      ;      RETURN          ;TEST "C" AND PROCEED.
2176      ;
2177 016466 012737 016520 000004 XNXM: MOV    @2$,@@4      ; SET BUSERR VECTOR.
2178 016474 012737 000200 000006     MOV    @PRI04,@@6
2179 016502 005003           CLR    R3          ; FLAG.
2180 016504 005711 1$:   TST    (R1)        ; TEST THE ADDRESS(ES).
2181           ;
2182 016506 020102           CMP    R1,R2        ; IF ANY TRAP, CONTINUE AT 2$.
2183 016510 001407           BEQ    3$          ; OTHERWISE, CONTINUE HERE.
2184 016512 062701 000002     ADD    @2,R1       ; BR IF FINISHED (NO NEXM'S).
2185 016516 000772           BR     1$         ; SET NEXT ADDRESS...
2186           ; ...AND CONTINUE.
2187 016520 005103 2$:   COM    R3          ; GOT ONE, SET FLAG...
2188 016522 012716 016530     MOV    @3$, (SP)
2189 016526 000002           RTI           ; ...AND DISMISS INTERRUPT...
2190 016530           3$:   CLRVEC @4          ; ...AND GIVE BACK THE VECTOR.
      016530 012700 000004     MOV    @4,R0
      016534 104436           TRAP   C$CVEC
2191 016536 005703           TST    R3          ; DID WE CATCH ONE ??
2192 016540 001401           BEQ    .+4        ; NO, "C" = 0. SKIP NEXT.
2193 016542 000261           SEC           ; YES, "C" = 1, (R1) = NEXM ADDR.
2194 016544 000207           RTS    PC
2195
2196
2197           .SBTTL TSTLOOP - CHECK ITERATION COUNT
2198
2199 ;+
2200 ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
2201 ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
2202 ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
2203 ;
2204 ; CALL: LOOPTO ARG
2205 ;
2205 016546      TSTLOOP:;
2206 016546 005737 002160     TST    NOITS      ; ITERATIONS INHIBITED?
2207 016552 001006           BNE    1$        ; YES.
2208 016554 005737 002174     TST    QVP        ; NO.
2209 016560 100403           BMI    1$        ; LOOPS DISALLOWED IN QUICK PASS.
2210 016562 005337 002206     DEC    LOOPCNT    ; BUMP LOOP COUNTER.
2211 016566 001002           BNE    2$
2212 016570 000241 1$:   CLC          ; LOOP DISALLOWED, OR DONE.
2213 016572 000401           BR     3$
2214 016574 000261 2$:   SEC          ; LOOP ENABLED.
2215 016576 000207 3$:   RTS    PC
2216
2217           .SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
2218
2219 ;+
2220 ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
2221 ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
2222 ; IN THE CURRENT RUN SEQUENCE.
2223 ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
2224 ;
2225 ; INPUT:
2226 ;
2227 ;      RO      POINTER TO TEST ID ASCIZ STRING

```

G6

TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```

2228 ;OUTPUT:
2229 ;
2230 ; R5 ADDRESS OF FIRST DEVICE REGISTER
2231 ;
2232 ;IMPLICIT OUTPUTS:
2233 ;
2234 ; TSTCNT UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
2235 ;
2236 ;SIDE EFFECTS:
2237 ;
2238 ; INTERRUPT LEVEL IS RASIED TO LEVEL OF
2239 ; THE DEVICE UNDER TEST
2240 ;
2241 ;-
2242 ;
2243 016600 TSTSETUP::
2244 016600 010046 MOV RO,-(SP) ;SAVE THE TEST ID MESSAGE
2245 016602 005037 003144 CLR SIFLAG ; CLEAR "SOFT INIT" FLAG
2246 016606 005037 017046 CLR ERRK ; CLEAR LOCAL ERROR COUNTER.
2247 016612 005037 005770 CLR EXTA ; CLEAR ERROR EXTENSION FLAG.
2248 016616 105037 016234 CLR INTMASK ; CLEAR INTERRUPT MASK (CHECK ERROR)
2249 016622 013700 002172 MOV UNITN,RO ; GET THE UNIT NUMBER.
2250 016626 006300 ASL RO ; ... AND MAKE IT A WORD OFFSET.
2251 016630 005737 003104 TST NODEV ; DID STARTUP FIND THE DEVICE?
2252 016634 001430 BEQ 4$ ; BR IF YES
2253 016636 100010 BPL 3$ ; BR IF NOT IDLE
2254 016640 052760 160000 003166 BIS #160000,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
2255 016646 ERRDF 1,NXR,NXRERR ; NO DEVICE HERE -- PRINT IT
    016646 104455 TRAP C#ERRDF
    016650 000001 .WORD 1
    016652 003736 .WORD NXR
    016654 005734 .WORD NXRERR
2256 016656 000407 BR 2$
2257 016660 052760 160001 003166 3$: BIS #160001,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
2258 016666 ERRDF 2,NOINIT ; DEVICE NOT IDLE
    016666 104455 TRAP C#ERRDF
    016670 000002 .WORD 2
    016672 004333 .WORD NOINIT
    016674 000000 .WORD 0
2259 016676 012737 177777 003102 2$: MOV #-1,DUFLG ; DROP THE UNIT
2260 016704 DODU UNITN
    016704 013700 002172 MOV UNITN,RO
    016710 104451 TRAP C#DODU
2261 016712 DOCLN ; ABORT THE PASS
    016712 104444 TRAP C#DOCLN
2262 016714 000423 BR 5$
2263
2264 C16716 4$: RFLAGS RO ; GET THE OPERATOR FLAGS.
    016716 104421 TRAP C#RFLA
2265 016720 032700 001000 BIT #PNT,RO ; PRINT THE TEST NUMBERS?
2266 016724 001412 BEQ 1$ ; BR IF NO
2267 016726 011600 MOV (SP),RO ;GET THE ID MESSAGE
2268 016730 PRINTF #TNAM,RO ;DISPLAY THE TEST ID
    016730 010046 MOV RO,-(SP)
    016732 012746 016774 MOV #TNAM,-(SP)
    016736 012746 000002 MOV #2,-(SP)
    016742 010600 MOV SP,RO

```

TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```

016744 104417          TRAP  C$PNTF
016746 062706 000006   ADD   #6,SP
2269 016752 005237 002204 1$:   INC   TSTCNT          ; BUMP TEST COUNTER.
2270 016756          SETPRI IPRI           ; PRIORITY THAT OF DEVICE
      016756 013700 002202   MOV   IPRI,RO
      016762 104441          TRAP  C$SPRI
2271 016764 005726          TST   (SP)+          ; FIX UP THE STACK
2272 016766 013705 002176   MOV   CSRADDR,R5    ; ADDRESS OF TSV REGISTERS ON UNIBUS
2273 016772 000207          RTS   PC
2274 016774      045      123  045  TNAM:  .ASCIZ  '#S#T#A Test'
2275          .EVEN
2276          .SBTTL  TSTEND - PRINT ERRORS RECEIVED
2277          ;
2278          ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2279          ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2280          ;
2281 017010          TSTEND: RFLAGS RO
      017010 104421          TRAP  C$RFLA
2282 017012 030027 020000   BIT   RO,#IER
2283 017016 001412          BEQ   1$             ; BR IF "IER" NOT SET.
2284 017020          PRINTF #ESUM,ERRK          ; PRINT ERROR COUNT.
      017020 013746 017046   MOV   ERRK,-(SP)
      017024 012746 017050   MOV   #ESUM,-(SP)
      017030 012746 000002   MOV   #2,-(SP)
      017034 010600          MOV   SP,RO
      017036 104417          TRAP  C$PNTF
      017040 062706 000006   ADD   #6,SP
2285 017044 000207          RTS   PC
2286
2287 017046 000000          ERRK:  0             ; LOCAL ERROR COUNT.
2288 017050      045      101  040  ESUM:  .ASCIZ  '#A #D#A ERRORS/'
2289 017067      105      122  122  EMAXDU: .ASCIZ  '/ERROR LIMIT REACHED -- DROPPING UNIT/'
2290          .EVEN
2291
2292          .SBTTL  INCERK - INCREMENT LOCAL ERROR COUNT
2293          ;
2294          ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
2295          ;
2296 017134 005237 017046   INCERK: INC   ERRK          ; INCREMENT LOCAL ERROR COUNT
2297 017140 010046          MOV   RO,-(SP)        ; SAVE RO
2298 017142 013700 002172   MOV   UNITN,RO        ; GET UNIT NUMBER.
2299 017146 006300          ASL   RO              ; ... AND MAKE IT A WORD OFFSET.
2300 017150 062700 003166   ADD   #ERTABL,RO      ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
2301 017154 005210          INC   (RO)           ; INCREMENT THE DEVICE ERROR COUNT
2302 017156 032710 007777   BIT   #7777,(RO)     ; DID WE OVERFLOW THE FIELD?
2303 017162 001001          BNE   1$             ; BR IF NO.
2304 017164 005310          DEC   (RO)           ; YES -- BACK IT UP TO 7777.
2305 017166 012600          1$:  MOV   (SP)+,RO      ; RESTORE RO
2306 017170 000207          RTS   PC            ; RETURN TO CALLER.
2307
2308 017172 010046          CKEMAX: MOV  RO,-(SP)        ; SAVE RO
2309 017174 013700 002172   MOV   UNITN,RO        ; GET UNIT NUMBER
2310 017200 006300          ASL   RO              ; ... AND MAKE IT A WORD OFFSET
2311 017202 016000 003166   MOV   ERTABL(RO),RO   ; GET ERROR TABLE ENTRY
2312 017206 042700 170000   BIC   #170000,RO      ; EXTRACT ERROR COUNT FIELD
2313 017212 020037 002164   CMP   RO,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
2314 017216 103004          BHIS  1$             ; BR IF YES

```

Te

INCERK - INCREMENT LOCAL ERROR COUNT

```

2315 017220 023737 017046 002162      CMP    ERRK,LERRMAX      ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2316 017226 103417                      BLO    2$               ; BR IF NO
2317 017230                      1$:  RFLAGS  RO           ; GET OPERATOR FLAGS
      017230 104421          TRAP  C$RFLA
2318 017232 032700 000040          BIT    $IDU,RO           ; IS DROPPING INHIBITED?
2319 017236 001013          BNE    2$               ; BR IF YES.
2320 017240 012737 177777 003102      MOV    $-1,DUFLG        ; NO -- DROP THE UNIT
2321 017246          ERRDF  4,EMAXDU
      017246 104455          TRAP  C$ERDF
      017250 000004          .WORD 4
      017252 017067          .WORD EMAXDU
      017254 000000          .WORD 0
2322 017256          DODU    UNITN
      017256 013700 002172      MOV    UNITN,RO
      017262 104451          TRAP  C$DODU
2323 017264          DOCLN
      017264 104444          TRAP  C$DCLN
2324 017266 012600          2$:  MOV    (SP)+,RO      ; RESTORE RO
2325 017270 000207          RTS    PC               ; RETURN TO CALLER
2326          .SBTTL  CKDROP - CHECK IF UNIT SHOULD BE DROPPED
2327          ;+
2328          ; CHECK IF UNIT SHOULD BE DROPPED
2329          ;-
2330 017272 010046      CKDROP: MOV    RO,-(SP)
2331 017274          FORCERROR 1$,NOTSSR
2332 017304          RFLAGS  RO
      017304 104421          TRAP  C$RFLA
2333 017306 032700 000040          BIT    $IDU,RO
2334 017312 001010          BNE    1$
2335 017314 011600          MOV    (SP),RO
2336 017316 012737 177777 003102      MOV    $-1,DUFLG
2337 017324          DODU    UNITN
      017324 013700 002172      MOV    UNITN,RO
      017330 104451          TRAP  C$DODU
2338 017332          DOCLN                      ;ABORT THE PASS
      017332 104444          TRAP  C$DCLN
2339 017334 012600          1$:  MOV    (SP)+,RO
2340 017336 000207          RTS    PC
2341
2342
2343          .SBTTL  CONFIG - DETERMINE CONFIGURATION OF SYSTEM
2344          ;
2345          ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
2346          ;
2347 017340      CONFIG:
2348 017340 004737 016064          JSR    PC,SOFINIT
2349 017344 000207          RTS    PC
2350          .SBTTL  KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
2351          ;
2352          ; SUBROUTINE - ENABLE MEM MGT.
2353          ;
2354 017346 005737 003122      KTON:  TST    KTFLG      ; GOT KT?
2355 017352 001403          BEQ    1$             ; NO.
2356 017354 012737 000001 177572      MOV    $1,SRO         ; YES. ENABLE KT11.
2357 017362 000207          1$:  RTS    PC
2358
2359          ;

```

KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT

```

2360 ; SUBROUTINE - DISABLE MEM MGT.
2361 ;
2362 017364 005737 003122 KTOFF: TST KIFLG ; GOT KT11?
2363 017370 001405 BEQ 1$ ; NO.
2364 017372 000240 NOP
2365 017374 000240 NOP
2366 017376 012737 000000 177572 MOV #0,SRO ; DISABLE KT.
2367 017404 000207 1$: RTS PC
2368 ;.SBTTL SETMAP - SETUP PAR6 MAPPING
2369 ;
2370 ;
2371 ;
2372 ; THIS ROUTINE SETS UP KERNEL PAR6 IP HANDLE
2373 ; AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
2374 ; IS RETURNED BIASED TO PAR6.
2375 ;
2376 ; INPUTS:
2377 ;
2378 ; R0 HIGH ORDER ADDRESS BITS
2379 ; R1 LOW ORDER ADDRESS BITS
2380 ;
2381 ; OUTPUTS:
2382 ;
2383 ; R0 OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
2384 ; CARRY SET IF SUCCESS
2385 ; CLR IF ERROR
2386 ;
2387 017406 SETMAP:
2388 017406 SAVREG ;SAVE R1-R4 UNTIL NEXT RETURN
2389 017412 005737 003122 TST KIFLG ;SYSTEM HAVE ABOVE 28K?
2390 017416 001433 BEQ 10$ ;BR IF NO
2391 017420 010102 MOV R1,R2 ;SAVE LOW ORDER BITS
2392 000006 .REPT 6
2393 ASR R0 ;CONVERT WORD ADDRESS TO 32W BLOCKS
2394 ROR R1 ;MAKE IT DOUBLE PRECISION
2395 .ENOR
2396 017452 042701 000177 BIC #177,R1 ;ALINE FOR LOWER 4K BOUNDARY
2397 017456 020137 003122 CMP R1,KIFLG ;HICHER THAN EXISTING MEMORY?
2398 017462 103011 BHIS 10$ ;BR IF YES
2399 017464 010137 172354 MOV R1,#KIPAR6 ;SETUP MAPPING REGISTER PAR6
2400 017470 042702 160000 BIC #160000,R2 ;SETUP DISPLACEMENT IN PAGE
2401 017474 062702 140000 ADD #140000,R2 ;ADD IN PAR6 BIAS
2402 017500 010200 MOV R2,R0 ;RETURN IN R0
2403 017502 000261 SFC ;SET SUCCESS
2404 017504 000401 BR 15$ ;
2405 017506 000241 10$: CLC ;SET FAILURE
2406 017510 000207 15$: RTS PC ;RETURN
2407 ;.SBTTL FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
2408 ;
2409 ; FILL MEMORY WITH A BACKGROUND PATTERN
2410 ;
2411 ; INPUTS:
2412 ;
2413 ; R0 - BACKGROUND PATTERN
2414 ; FREE - FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2415 ; KIFLG - SET TO HIGHEST MEMORY LOCATION IF > 28K.
2416 ;

```

166

FILLMEM FILL MEMORY WITH BACKGROUND PATTERN

```

2417 ; OUTPUTS:
2418 ;
2419 ; NONE
2420 ;
2421 ;
2422 FILLMEM:
2423 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2424 JSR PC,KTOFF ;DISABLE KT.
2425 MOV RO,R3 ;COPY TEST PATTERN
2426 MOV FREE,R1 ;GET FIRST FREE LOCATION
2427 MOV FRESIZ,R2 ;SIZE OF FREE SPACE BELOW 28K.
2428 10$: MOV R3,(R1)+ ;STORE A BACKGROUND WORD
2429 DEC R2 ;DONE ALL MEMORY IN FREE SPACE?
2430 BGT 10$ ;BR IF NO
2431 TST KIFLG ; GOT KT?
2432 BEQ 55$ ; NO, GET OUT.
2433 JSR PC,KTON ; YES, ENABLE KT.
2434 CLR RO ;HIGH ORDER ADDRESS START
2435 MOV PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
2436 .REPT 6
2437 CLC ;CLEAR C BIT
2438 ROL R1 ;CONVERT BLOCKS TO WORDS
2439 ROL RO ;MAKE IT DOUBLE PRECISION
2440 .ENDR
2441 JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
2442 30$: MOV R3,(RO)+ ;STORE TEST PATTERN IN >28K ADDRESS
2443 CMP RO,#160000 ;END OF PAR6 MAPPING AREA?
2444 BLO 30$ ;BR IF NO
2445 SUB #20000,RO ;BACKUP INTO PAR6 MAPPING BEGIN
2446 ADD #200,#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2447 CMP #KIPAR6,KIFLG ;END OF MEMORY?
2448 BEQ 50$ ;BR IF YES
2449 TST T23A ;11/23A?
2450 BEQ 35$ ;NO KEEP GOING
2451 MOV SRO,R4 ;GET SRO CONTENTS
2452 BIC #177761,R4 ;CLEAR ALL B: , PAGE NUMBER
2453 CMP #16,R4 ;SEE IF PAGE 7
2454 BEQ 50$ ;EXIT IF THERE
2455 35$: TST T23B ;11/23B?
2456 BEQ 45$ ;NO KEEP GOING
2457 CMP #KIPAR6,#7600 ;REACHED 18 BITS?
2458 BHIS 40$ ;YES
2459 BR 45$ ;NO KEEP GOING
2460 40$: MOV #20,SR3 ;SET 22 BIT RELOCATION
2461 45$: JMP 30$ ;KEEP GOING ON ETC.
2462 50$: JSR PC,KTOFF ; DISABLE KT.
2463 55$: RTS PC
2464 .SBTTL CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
2465 ;
2466 ; COMPARE MEMORY WITH A BACKGROUND PATTERN
2467 ;
2468 ; INPUTS:
2469 ;
2470 ; RO = BACKGROUND PATTERN
2471 ; FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2472 ; KIFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2473 ;

```

CMPMEM COMPARE MEMORY TO BACKGROUND PATTERN

```

2474 ; OUTPUTS:
2475 ;
2476 ; CARRY - SET IF NO ERROR
2477 ; CARRY - CLR IF ERROR
2478 ;
2479 ; IMPLICIT OUTPUTS:
2480 ;
2481 ; ERRHI - ERROR HIGH ADDRESS
2482 ; ERRLO - ERROR LOW ADDRESS
2483 ; EXPD - EXPECTED DATA
2484 ; RECV - RECEIVED DATA
2485 ;
2486 017750 CMPMEM:
2487 017750 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2488 017754 010003 MOV R0,R3 ;COPY TEST PATTERN
2489 017756 004737 017364 JSR PC,KTOFF ;DISABLE KT.
2490 017762 013701 003114 MOV FREE,R1 ;GET FIRST FREE LOCATION
2491 017766 013702 003116 MOV FRESIZ,R2 ;SIZE OF FREE SPACE BELOW 28K.
2492 017772 020311 10$: CMP R3,(R1) ;FREE SPACE LOCATION EQUAL TO EXPD?
2493 017774 001411 BEQ 15$ ;BR IF YES
2494 017776 010137 502230 MOV R1,ERRLO ;SAVE ADDRESS IN ERROR
2495 020002 005037 002226 CLR ERRHI ;NO HIGH ADDRESS
2496 020006 010337 002222 MOV R3,EXPD ;SAVE EXPD FOR ERROR REPORT
2497 020012 011137 002224 MOV (R1),RECV ;SAVE RECV FOR ERROR REPORT
2498 020016 000474 BR 50$ ;
2499 020020 005721 15$: TST (R1)+ ;POINT TO NEXT ADDRESS
2500 020022 005302 DEC R2 ;DONE ALL MEMORY IN FREE SPACE?
2501 020024 003362 BGT 10$ ;BR IF NO
2502 020026 005737 003122 TST KTFLG ; GOT KT?
2503 020032 001472 BEQ 55$ ; NO. GET OUT.
2504 020034 004737 017346 JSR PC,KTON ; YES. ENABLE KT.
2505 020040 005000 CLR R0 ;HIGH ORDER ADDRESS START
2506 020042 013701 003142 MOV PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
2507 000006 .REPT 6
2508 ROL R1 ;CONVERT BLOCKS TO WORDS
2509 ROL R0 ;MAKE IT DOUBLE PRECISION
2510 .ENDR
2511 020076 042701 000177 BIC #177,R1 ;ALINE 4K BOUNDARY
2512 020102 010046 MOV R0,-(SP) ;SAVE HIGH ORDER
2513 020104 010146 MOV R1,-(SP) ;SAVE LOW ORDER
2514 020106 004737 017406 JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
2515 020112 010004 MOV R0,R4 ;COPY ADDRESS BIASED TO PAR6
2516 020114 012601 MOV (SP)+,R1 ;RESTORE LOW ORDER IN NON PAR6 FORMAT
2517 020116 012600 MOV (SP)+,R0 ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
2518 020120 020314 30$: CMP R3,(R4) ;ABOVE 28K LOCATION EQUAL EXPD?
2519 020122 001411 BEQ 32$ ;BR IF YES
2520 020124 010037 002226 MOV R0,ERRHI ;SAVE HIGH ORDER IN ERROR
2521 020130 010137 002230 MOV R1,ERRLO ;SAVE LOW ORDER IN ERROR
2522 020134 010337 002222 MOV R3,EXPD ;SAVE EXPD FOR ERROR REPORT
2523 020140 011437 002224 MOV (R4),RECV ;SAVE RECV FOR ERROR REPORT
2524 020144 000421 BR 50$ ;
2525 020146 062701 000002 32$: ADD #2,R1 ;UPDATE NON PAR6 ADDRESS
2526 020152 005500 ADC R0 ;MAKE IT DOUBLE PRECISION ADD
2527 020154 062704 000002 ADD #2,R4 ;UPDATE PAR6 FORMAT ADDRESS
2528 020160 020427 160000 CMP R4,#160000 ;END OF PAR6 MAPPING AREA?
2529 020164 103755 BLO 30$ ;BR IF NO
2530 020166 162704 020000 SUB #20000,R4 ;BACKUP INTO PAR6 MAPPING BEGIN

```



Me

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```

2531 020172 062737 000200 172354      ADD    #200,@#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2532 020200 023737 172354 003122      CMP    @#KIPAR6,KTFLG ;END OF MEMORY?
2533 020206 101744                      BLOS  30$           ;BR IF NO
2534 020210 004737 017364      50$:  JSR    PC,KTOFF   ;TURN OFF MEMORY MAPPING
2535 020214 000241                      CLC                    ;SET FAILURE
2536 020216 000403                      BR    60$             ;
2537 020220 004737 017364      55$:  JSR    PC,KTOFF   ;TURN OFF MEMORY MAPPING
2538 020224 000261                      SEC                    ;SET SUCCESS
2539 020226 000207      60$:  RTS    PC
2540                      .SBITL  REGSAV  - SAVE R1-R5 ON STACK
2541                      ;+
2542                      ;
2543                      ;ROUTINE TO
2544                      ;SAVE R1 THROUGH R5 ON THE STACK
2545                      ;
2546                      ;CALLING SEQUENCE:
2547                      ;
2548                      ;      JSR    R5,REGSAV
2549                      ;
2550                      ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
2551                      ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
2552                      ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
2553                      ;REGISTERS.
2554                      ;
2555                      ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
2556                      ;CALLED VIA A JSR PC INSTRUCTION
2557                      ;
2558                      ;-
2559
2560                      REGSAV:
2561 020230 010446      MOV    R4,-(SP)
2562 020232 010346      MOV    R3,-(SP)
2563 020234 010246      MOV    R2,-(SP)
2564 020236 010146      MOV    R1,-(SP)
2565 020240 010546      MOV    R5,-(SP)
2566 020242 016605 000012      MOV    10.(SP),R5
2567 020246 004736      JSR    PC,@(SP)+
2568 020250 012601      MOV    (SP)+,R1
2569 020252 012602      MOV    (SP)+,R2
2570 020254 012603      MOV    (SP)+,R3
2571 020256 012604      MOV    (SP)+,R4
2572 020260 012605      MOV    (SP)+,R5
2573 020262 000207      RTS    PC
2574                      .SBITL  GETPAT  - GET 8 BIT PATTERN FROM OPERATOR
2575                      ;+
2576                      ;
2577                      ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
2578                      ;
2579                      ;INPUTS:
2580                      ;
2581                      ;      NONE.
2582                      ;
2583                      ;OUTPUTS:
2584                      ;
2585                      ;      R0      OCTAL NUMBER FROM THE OPERATOR
2586                      ;
2587                      ;CALLING SEQUENCE:

```

GETPAT GET 8 BIT PATTERN FROM OPERATOR

```

2588 ;
2589 ; JSR PC,GETPAT
2590 ;
2591 ; -
2592 ;
2593 GETPAT::
2594 020264 SAVREG ;SAVE THE GENERAL REGISTERS
2595 020264 1$: GMANID DATASC,PATDAT,0,377,0,377,NO
020270 104443 TRAP C$GMAN
020272 000406 BR 10000$
020274 020320 .WORD PATDAT
020276 000022 .WORD T$CODE
020300 020322 .WORD DATASC
020302 000377 .WORD 377
020304 000000 .WORD T$LOLIM
020306 000377 .WORD T$HILIM
020310 10000$:
2596 020310 BNCOMPLETE 1$ ;RETRY IF ERROR
020310 103367 BCC 1$
2597 020312 013700 020320 MOV PATDAT,R0 ;DATA PATTERN FROM OPERATOR
2598 020316 000207 RTS PC ;RETURN TO CALLER
2599 ;
2600 ;*
2601 ;LOCAL DATA AREA
2602 ; -
2603 ;
2604 020320 000000 PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
2605 020322 105 116 124 DATASC: .ASCIZ 'ENTER DATA PATTERN'
2606 .EVEN
2607 .SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
2608 ;*
2609 ;
2610 ;ROUTINE TO ISSUE A MENU AND GET
2611 ;THE OPERATOR'S RESPONSE.
2612 ;
2613 ;INPUTS:
2614 ;
2615 ; R0 ADDRESS OF ASCIZ STRING OF MENU
2616 ; R1 MAXIMUM ALLOWABLE OPERATOR RESPONSE
2617 ;
2618 ;OUTPUTS:
2619 ;
2620 ; R0 NUMBER OF THE OPERATOR'S SELECTION
2621 ;
2622 ; -
2623 ;
2624 GETSEL::
2625 020345 SAVREG ;SAVE GENERAL REGISTERS
2626 020352 010002 MOV R0,R2 ;SAVE THE MENU ADDRESS
2627 020354 010203 1$: MOV R2,R3 ;START OF MENU STRING
2628 020356 005713 2$: TST (R3) ;END OF ASCII ?
2629 020360 001412 BEQ 3$ ;BRANCH IF ALL LINES DISPLAYED
2630 020362 PRINTF #SELASC,(R3)+ ;DISPLAY THE MENU
020362 012346 MOV (R3)+,-(SP)
020364 012746 020532 MOV #SELASC,-(SP)
020370 012746 000002 MOV #2,-(SP)
020374 010600 MOV SP,R0

```

137

GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

```

020376 104417 TRAP C#PNTF
020400 062706 000006 ADD 06,SP
2631 020404 000764 BR 2#
2632 020406 3# GMANID MENASC,MENRES,D,-1,0,-1,NO
020406 104443 TRAP C#GMAN
020410 000406 BR 10001#
020412 020566 .WORD MENRES
020414 000042 .WORD T#CODE
020416 020537 .WORD MENASC
020420 177777 .WORD -1
020422 000000 .WORD T#LOLIM
020424 177777 .WORD T#HILIM
020426 10001#
2633 020426 BNCOMPLETE 1# ;RETRY IF ERROR
020426 103352 BCC 1#
2634 020430 013700 020566 MOV MENRES,RO ;GET THE OPERATOR'S REPLY
2635 020434 020001 CMP RO,R1 ;COMPARE TO MAXIMUM ALLOWED
2636 020436 101411 BLOS 5# ;BRANCH IF OK
2637 020440 PRINTF #MENERR ;DISPLAY ERROR MESSAGE
020440 012746 020464 MOV #MENERR,-(SP)
020444 012746 000001 MOV #1,-(SP)
020450 010600 MOV SP,RO
020452 104417 TRAP C#PNTF
020454 062706 000004 ADD 04,SP
2638 020460 000735 BR 1# ;RETRY
2639 020462 000207 RTS PC ;RETURN TO CALLER
2640 020464 045 116 045 MENERR: .ASCIZ 'MNA *** Menu Selection Too Large ***'
2641 020532 045 116 045 SELASC: .ASCIZ 'MNT'
2642 020537 105 156 164 MENASC: .ASCIZ 'Enter Menu Selection: '
2643 .EVEN
2644 020566 000000 MENRES: .WORD 0
2645 .SBTTL CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
2646
2647
2648 ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2649
2650 ;INPUT:
2651
2652 ; NONE.
2653
2654 ;OUTPUT:
2655
2656 ; CARRY 0 MANUAL INTERVENTION NOT ALLOWED
2657 ; 1 MANUAL INTERVENTION IS OK
2658
2659 ;SIDE EFFECTS:
2660
2661 ; A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2662 ; NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2663 ; ALLOWED.
2664
2665
2666
2667 020570 CHKMAN: SAVREG ;SAVE THE REGISTERS
2668 020570 MANUAL ;SEE IF MANUAL INTERVENTION OK
2669 020574 TRAP C#MANI
020574 104450

```

CHKMAN CHECK MANUAL INTERVENTION LEGALITY

```

2670 020576          BCOMPLETE 1$          ;BRANCH IF ALLOWED
      020576 103411  BCS 1$
2671 020600          PRINTF 0NOMAN          ;PRINT THE WARNING MESSAGE
      020600 012746 020624  MOV 0NOMAN,-(SP)
      020604 012746 000001  MOV 01,-(SP)
      020610 010600          MOV SP,R0
      020612 104417          TRAP C#PNTF
      020614 062706 000004  ADD 04,SP
2672 020620 000241  CLC          ;CLEAR CARRY FOR ERROR
2673 020622 000207  1$: RTS PC          ;RETURN
2674
2675 020624 045 116 045 NOMAN: .ASCIIZ 'NMA *** Manual Intervention not Allowed - Test Aborted ***'
2676 .even
2677 .SBTTL ENVIRN - SETUP FREE DIAGNOSTIC SPACE
2678 ;
2679 ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
2680 ;
2681 020720          ENVIRN: MEMORY R0
      020720 104431  TRAP C#MEM
2682 020722 010037 003114  MOV R0,FREE          ; GET 1ST FREE ADDRESS...
2683 020726 062737 000002 003114  ADD 02,FREE
2684 020734 011037 003116  MOV (R0),FRESIZ          ;...AND WORD COUNT.
2685 020740 162737 000004 003116  SUB 04,FRESIZ
2686 020746 013702 002012  MOV L#UNIT,R2          ; GET NUMBER OF UNITS
2687 020752 162737 000007 003116 10$: SUB 07,FRESIZ          ; TAKE AWAY 7 WORDS PER UNIT
2688 020760 005302  DEC R2
2689 020762 001373  BNE 10$
2690 020764 013700 003114  MOV FREE,R0          ;GET FIRST FREE ADDRESS
2691 020770 063700 003116  ADD FRESIZ,R0          ;POINT TO LAST FREE ADDRESS
2692 020774 162700 000002  SUB 02,R0          ;BACKUP 1 WORD
2693 021000 010037 003120  MOV R0,FREEHI          ;STORE LAST FREE ADDRESS
2694 021004 000240  NOP
2695 021006 012701 177520  MOV 0BDVPCR,R1          ;GET BDV11 PCR ADDRESS
2696 021012 010102  MOV R1,R2          ;COPY TO R2
2697 021014 062702 000002  ADD 02,R2          ;SET THE RANGE
2698 021020 004737 016466  JSR PC,XNXM          ;SEE IF WE HAVE ONE
2699 021024 103001  BCC 15$          ;OK TO SET FLAGS
2700 021026 000445  BR 40$          ;RETURN WITH FLAGS CLEAR
2701 021030 013701 177520 15$: MOV BDVPCR,R1          ;SAVE PCR CONTENTS
2702 021034 062701 000001  ADD 01,R1          ;ADD ONE TO IT
2703 021040 012702 177520  MOV 0BDVPCR,R2          ;GET BDV11 PCR ADDRESS
2704 021044 005212  INC (R2)          ;TRY TO WRITE TO IT
2705 021046 013703 177520  MOV BDVPCR,R3          ;GET RESULTS
2706 021052 020103  CMP R1,R3          ;DID IT CHANGE?
2707 021054 001017  BNE 20$          ;NO, MUST BE 11/23B
2708 021056 005237 003134  INC T23A          ;SET THE FLAG
2709 021062 042737 170000 002120  BIC 0170000,L#HIME          ;SUPERVISOR COULD BE WRONG
2710 021070 000240  NOP          ;BR 40$ FOR RELEASE
2711 021072          PRINTF 0M8186          ;TELL THE SYSTEM TYPE
      021072 012746 005552  MOV 0M8186,-(SP)
      021076 012746 000001  MOV 01,-(SP)
      021102 010600          MOV SP,R0
      021104 104417          TRAP C#PNTF
      021106 062706 000004  ADD 04,SP
2712 021112 000413  BR 40$          ;RETURN
2713 021114 005237 003136 20$: INC T23B          ;SET THE FLAG
2714 021120 000240  NOP          ;BR 40$ FOR RELEASE

```

D7

SEQ 0081

ENVIRN SETUP FREE DIAGNOSTIC SPACE

```

2715 021122          PRINTF 0M8189          ; TELL THE SYSTEM TYPE
      021122 012746 005643      MOV 0M8189, -(SP)
      021126 012746 000001      MOV 01, -(SP)
      021132 010600          MOV SP, R0
      021134 104417          TRAP C$PNTF
      021136 062706 000004      ADD 04, SP
2716 021142 000207      40$: RTS PC          ; RETURN
2717          .SBTTL KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS
2718          ;+
2719          ;
2720          ; ROUTINE TO INIT KT-11
2721          ;
2722          ;-
2723
2724 021144          KTINIT:
2725 021144 005037 003122      CLR KTF LG          ; INIT >28K MEMORY FLAG
2726 021150 005037 003124      CLR K TENABLE       ; INIT TEST >28K FLAG
2727 021154 023727 002120 001577  CMP L$HIME, 01577    ; GOT ENOUGH MEMORY (>28K)?
2728 021162 101444          BLOS 9$          ; NO.
2729 021164 013700 000004      MOV 00ERRVEC, R0    ; SAVE OLD ERR VEC PTR.
2730 021170 012737 021262 000004  MOV 02$, 00ERRVEC   ; SET ERR VEC PTR.
2731 021176 005737 177572      TST 00SRO          ; GOT KT11?
2732 021202 000240          NOP          ; (TRAP IF NO).
2733 021204 013737 002120 003122  MOV L$HIME, KTF LG  ; YES. SET KT FLAG.
2734 021212 042737 000177 003122  BIC 0177, KTF LG
2735 021220 010037 000004      MOV R0, 00ERRVEC   ; RESTORE OLD ERR VEC PTR.
2736 021224 005000          CLR R0          ; R0 = AR DATA.
2737 021226 012701 172340      MOV 0KIPAR0, R1    ; R1 = KI REGS PTR.
2738 021232 012761 077406 177740 1$: MOV 077406, -40(R1) ; SET DESCRIPTOR REG.
2739 021240 010021          MOV R0, (R1)+      ; SET KIPAR REG.
2740 021242 062700 000200      ADD 0200, R0       ; BUMP AR DATA BY "4K".
2741 021246 020027 002000      CMP R0, 02000     ; AT "I/O"?
2742 021252 001367          BNE 1$          ; NO.
2743 021254 012741 177600      MOV 0177600, -(R1) ; YES. SET KTPAR7 FOR I/O.
2744 021260 000405          BR 9$
2745
2746 021262 012716 021270      2$: MOV 06$, (SP)      ; SET UP RETURN
2747 021266 000002          RTI          ; RTI TO NEXT LOCATION
2748
2749 021270 010037 000004      6$: MOV R0, 00ERRVEC   ; RESTORE OLD ERR VEC PTR.
2750
2751 021274 000207      9$: RTS PC
2752          ;+
2753          ; SUBROUTINE TO SET EXTENDED FEATURES SWITCH
2754          ;
2755          ; Requires that SOFINIT and WRTCHR have been done previous to call.
2756          ;
2757          ;
2758          ; INPUTS:
2759          ; R5 CURRENT UNIT NUMBER
2760          ; OUTPUTS:
2761          ; The Extended Features Switch is set.
2762          ;
2763          ;-
2764
2765 021276          INVERT::
2766

```

17

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

2767 021276 005737 002216          TST      EXTFEA          ; IS SWITCH SET?
2768 021302 001020                    BNE     1$             ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
2769 021304 012737 100206 021350    MOV     #100206,CMDPKT ; WRT SUB-SYS MEM CMD
2770 021312 012737 021360 021352    MOV     #WSMBK,CMDPKT+2 ; MSG BUF ADDR
2771 021320 012737 000006 021356    MOV     #6,CMDPKT+6    ; BYTE COUNT
2772 021326 012737 100010 021360    MOV     #100010,WSMBK  ; INVERT THE SWITCH
2773 021334 012704 021350          MOV     #CMDPKT,R4     ; SET CMDPKT INTO R4
2774 021340 004737 010752          JSR     PC,WRTCHR      ; DO IT
2775 021344 000207          1$:    RTS      PC          ; RETURN
2776
2777          ;          COMMAND PACKET.
2778
2779          021350          .          =          <.,+3>&177774 ;MUST BE ON MOD 4 BOUNDRY.
2780
2781 021350 000000          CMDPKT:: 0          ;1ST WORD IS TS05 COMMAND.
2782 021352 000000                    0          ;2ND WORD IS THE BUFFER LOW ADDRESS.
2783 021354 000000                    0          ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2784 021356 000000                    0          ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2785
2786          ;          WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
2787
2788 021360 000000          WSMBK:: 0          ;1ST WORD:: SEL 0
2789 021362 000000                    0          ;2ND WORD:: SEL 2
2790 021364 000000                    0          ;3RD WORD:: SEL 4
2791          .EVEN
2792
2793          ;+
2794          ;          SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
2795          ;
2796          ;INPUTS:
2797          ;OUTPUTS:
2798          ;          The NXMFLG is set if we can test.
2799          ;          The NXMLO and NXMHI addresses are setup.
2800          ;--
2801
2802 021366          MEMCK::
2803
2804 021366          SAVREG          ;SAVE THE REGISTERS
2805 021372 005037 003126          CLR     NXMFLG        ;CLEAR THE FLAG
2806 021376 005037 003130          CLR     NXMLO        ;CLEAR THE TEST ADDRESS LO
2807 021402 005037 003132          CLR     NXMHI        ;CLEAR THE TEST ADDRESS HI
2808 021406 005737 003136          TST     T23B         ;IS IT A 11/23B?
2809 021412 001407                    BEQ     1$            ;NO
2810 021414 023727 002120 007777    CMP     L#HIME,#7777  ; GREATER THAN 128K
2811 021422 103406                    BLO    2$            ; NO
2812 021424 004737 021542          JSR     PC,NXMTST     ;SETUP THE ADDRESS
2813 021430 000427                    BR     13$           ;SET THE FLAG AND EXIT
2814 021432 005737 003134          1$:    TST     T23A         ;IS IT A 11/23A?
2815 021436 001413                    BEQ     4$            ;NO
2816 021440 023727 002120 005777    2$:    CMP     L#HIME,#5777 ;GREATER THAN 96K
2817 021446 101023                    BHI    14$           ;YES,23A/23B WITH 128K MEMORY
2818 021450 023727 002120 003777    CMP     L#HIME,#3777  ;GREATER THAN 64K BUT LESS THAN 92K?
2819 021456 103403                    BLO    4$            ;NO, CHECK 24K
2820 021460 004737 021542          JSR     PC,NXMTST     ;SETUP THE ADDRESS
2821 021464 000411                    BR     13$           ;SET THE FLAG AND EXIT
2822 021466 023727 002120 001577    4$:    CMP     L#HIME,#1577 ;GREATER THAN 24K BUT LESS THAN 64K?
2823 021474 103410                    BLO    14$           ;NO, TELL THEM AND EXIT WITH FLAG CLEAR

```

F7

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

2824 021476 004737 021542      JSR      PC,NXMTST      ;SETUP THE ADDRESS
2825 021502 062737 000077 003132  ADD      #77,NXMHI     ;FOOL THE 11/02 & 11/03
2826 021510 005237 003126      13$:    INC      NXMFLG     ;SET THE FLAG
2827 021514 000411              BR       15$           ;EXIT
2828 021516 000410              14$:    BR       15$           ;NOP FOR PRINTOUT
2829 021520              PRINTF   #NOMEM        ;TELL THEM 2 EXIT ***NO PRINT*****
      021520 012746 005456      MOV      #NOMEM,-(SP)
      021524 012746 000001      MOV      #1,-(SP)
      021530 010600      MOV      SP,R0
      021532 104417      TRAP    C$PNTF
      021534 062706 000004      ADD      #4,SP
2830 021540 000207      15$:    RTS      PC      ;RETURN
2831
2832      ;*
2833      ;      SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
2834      ;
2835      ;OUTPUTS:NXMLO,NXMHI      ;SETUP WITH NXM ADDRESS
2836      ;
2837      ;-
2838
2839 021542 013701 002120      NXMTST: MOV      L$HIME,R1      ;GET TOP OF MEMORY
2840 021546 062701 000200      ADD      #200,R1          ;MAKE IT I/O BLOCK OR OTHER NXM
2841 021552 042701 000177      BIC      #177,R1
2842 021556 010102              MOV      R1,R2          ;RESAVE RESULTS
2843              .REPT      6
2844              ASL      R1          ;PUT IN PLACE FOR XFER
2845              .ENDR
2846 021574 010137 003130      MOV      R1,NXMLO        ;SAVE TEST ADDRESS LOW
2847              .REPT      10
2848              ASR      R2          ;PUT IN PLACE FOR XFER
2849              .ENDR
2850 021624 042702 177700      BIC      #177700,R2      ;DON'T WANT ILA!
2851 021630 010237 003132      MOV      R2,NXMHI        ;SAVE TEST ADDRESS HIGH
2852 021634 000207      RTS      PC      ;RETURN
2853
2854
2855
2856 021636      ENDMOD

```

67

KTINIT SETUP KT11 MEMORY MANAGEMENT REGISTERS

7  
8  
9  
10  
16

.TITLE TSV4 - MISCELLANEOUS SECTIONS  
BGNMOD TSV4

TSV4::



H/

PROTECTION TABLE

```

18
19 021636 .SBITL PROTECTION TABLE
    021636 BGNPROT
20 021636 177777 177777 177777 L$PROT::
21 021646 .WORD -1, -1, -1, -1
22      ENDPROT

```

;NO DEVICE PROTECTION REQUIRED.

INITIALIZE SECTION

```

24          .SBTTL  INITIALIZE SECTION
25
26          ;**
27          ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
28          ;AT THE BEGINNING OF EACH PASS.
29
30          ;IF "START" OR "RESTART", SET QUICK PASS FLAG AND BUS-INIT.
31          ;IF "CONTINUE", NOTHING IS REQUIRED.
32          ;
33          ;--
34          ;+
35          ;INSERT TEMPORARY JUMP TO ODT
36          ;-
37 021646          BGNINIT
38 021646          L$INIT::
39 021646          SETVEC  0140,0170000,0340          ;ODT ROM ADDRESS          ;JB REV A-0
021646          MOV      0340,-(SP)
021652          MOV      012746 000340
021652          MOV      012746 170000
021656          MOV      012746 000140
021662          MOV      012746 000003
021666          TRAP    C$SVEC
021670          ADD      062706 000010
40
41 021674          005037 002216          40$: CLR      EXTFEA
42 021700          005037 003126          CLR      NXMFLG
43 021704          012737 006356 002170  MOV      0EPRT1,EPRTSW          ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
44 021712          005037 003144          CLR      SIFLAG          ;CLEAR "SOFT INIT" FLAG
45 021716          005037 003124          CLR      KTENABLE          ;CLEAR TEST ABOVE 28K FLAG
46 021722          005037 002272          CLR      RAMSIZ          ;CLEAR RAM SIZE FOR RAMERR ROUTINE
47 021726          READEF  0EF,CONTINUE
021726          MOV      012700 000036
021732          TRAP    C$REFG
48 021734          BNCOMPLETE 1$
021734          BCC      103023
49 021736          023737 002172 002012  CMP      UNITN,L$UNIT          ;UNIT IN RANGE?
50 021744          103070          BHS      4$          ;BR IF NO.
51 021746          005737 003102          TST      DUFLG          ;DROPPED UNIT?
52 021752          100472          BMI      NXTU          ;BR IF YES
53 021754          013701 002172          MOV      UNITN,R1
54 021760          006301          ASL      R1
55 021762          005761 003166          TST      ERTABL(R1)
56 021766          001516          BEQ      SETU
57 021770          032761 040000 003166  BIT      0BIT14,ERTABL(R1)          ;DROPPED?
58 021776          001060          BNE      NXTU
59 022000          EXIT      INIT          ;DO NOTHING IF "CONTINUE".
022000          TRAP    C$EXIT
022002          .WORD   L10030-.
60 022004          1$: READEF  0EF,NEW
022004          MOV      012700 000035
022010          TRAP    C$REFG
61 022012          BNCOMPLETE NXTU          ;TAKE NEXT UNIT IF NOT NEW PASS.
022012          BCC      103052
62 022014          READEF  0EF,START
022014          MOV      012700 000040
022020          TRAP    C$REFG
63 022022          BCOMPLETE 2$

```

INITIALIZE SECTION

```

022022 103404      BCS      2$
64 022024      REDEF    #REF,RESTART
022024 012700 000037  MOV     #REF,RESTART,RO
022030 104447      TRAP    C$REFG
65 022032      BNCOMPLETE 31$
022032 103031      BCC     31$
66 022034      2$:
67 022034      BRESET
022034 104433      TRAP    C$RESET
68 022036 005037 002204  CLR     TSTCNT
69 022042 005037 002212  CLR     FATFLG
70 022046 005037 003134  CLR     T23A
71 022052 005037 003136  CLR     T23B
72          ;      MOV     #340,-(SP)
73          ;      MOV     #20,-(SP)
74          ;      JMP     0,ODT
75 022056 005037 003370  CLR     SKIPT
76 022062      20$:
77 022062 012737 177777 002174  MOV     #-1,QVP
78 022070 004737 020720      JSR     PC,ENVIRN
79 022074 004737 021144      JSR     PC,KTINIT
80 022100 012700 003166      MOV     #ERTABL,RO
81 022104 0C5020      30$:  CLR     (RO)+
82 022106 020027 003366      CMP     RO,#ERTABE
83 022112 103774      BLO    30$
84 022114 000404      BR     4$
85 022116 005037 002174      31$:  CLR     QVP
86 022122 000137 022172      JMP     PASRPT
87
88 022126      4$:
89 022126 012737 177777 002172  NEWPAS: MOV    #-1,UNITN
90 022134 005037 002210      CLR     DEVCNT
91 022140      NXTU:  BREAK
022140 104422      TPAP   C$BRK
92 022142 005237 002172      INC     UNITN
93 022146 023737 002172 002012  CMP     UNITN,L$UNIT
94 022154 103423      BLO    SETU
95 022156 012737 177777 003102  MOV     #-1,DUFLG
96 022164 000401      BR     11$
97 022166      DOCLN
022166 104444      TRAP   C$DCLN
98 022170 000240      11$:  NOP
99 022172      PASRPT:
100 022172 023727 002012 000001  CMP     L$UNIT,#1
101 022200 101752      BLOS   NEWPAS
102 022202 005737 002210      TST    DEVCNT
103 022206 001747      BEQ    NEWPAS
104 022210      RFLAGS RO
022210 104421      TRAP   C$RFLA
105 022212 032700 000100  BIT     #ISR,RO
106 022216 001343      BNE    NEWPAS
107
108 022220      DORPT
022220 104424      TRAP   C$DRPT
109 022222 000741      BR     NEWPAS
110 022224      10$:
111

```

```

;1ST PASS, BUS-INIT...
;BUS RESET.
;NUMBER OF TESTS RUN IN PASS
;CLEAR FATAL ERROR COUNT
;CLEAR 11/23A FLAG
;CLEAR 11/23B FLAG
;RETURN TO DEBUGGER
;;@ENTER THE DEBUGGER
;CLEAR THE SUBTEST "SKIPPER"
;...QUICK VERIFY...
;SET ENVIRONMENT.
;INITIALIZE KT MEMORY MANAGEMENT
;CLEAR THE ERROR TABLE
;GO REPORT THE STATUS
;INIT UNIT NUMBER...
;CLEAR COUNT OF DEVICES RUNNING
;...AND SET NEXT UNIT NUMBER.
;ABORT, NO MORE UNITS.
;HOW MANY UNITS SELECTED?
;BR IF ONLY 1
;ARE ANY STILL RUNNING?
;BR IF NO
;SHOULD WE PRINT STATISTICS
;BR IF NO

```

INITIALIZE SECTION

```

112 022224          SETU:  GPBARD  UNITN,R0          ;GET UNIT N P-TABLE POINTER.
      022224 013700 002172  MOV      UNITN,R0
      022230 104442      TRAP    C$GPBARD
113 022232          BNCOMPLETE NXTU          ;BR IF UNIT NOT AVAILABLE.
      022232 103342      BCC     NXTU
114 022234 005037 003102  CLR     DUFLG          ;CLEAR "DROPPED" FLAG.
115 022240 005237 002210  INC     DEVCNT
116 022244 012001      MOV     (R0)+,R1          ;GET 1ST REGISTER ADDRESS.
117 022246 010137 002176  MOV     R1,CSRADDR    ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
118
119 022252 012001      MOV     (R0)+,R1          ;GET VECTOR ADDRESS.
120      ;MOV    (R0),R2          ;GET INTERRUPT PRIORITY
121      ;MOV    R2,IPRI          ;SET INTERRUPT PRIORITY.
122 022254 010137 002200  MOV     R1,IVEC       ;SET INTERRUPT VECTOR POINTER...
123 022260 012721 016306  MOV     @INTR,(R1)+   ;...VECTOR...
124 022264 013721 002202  MOV     IPRI,(R1)+   ;...AND PRIORITY.
125
126 022270          1$:
127      ; TST     QVP          ;1ST PASS ??
128      ; BEQ    5$          ;NO, SKIP THE PASS 1 STUFF.
129
130
131      ;
132      ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
133      ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
134
134 022270 013701 002172  ;
      MOV     UNITN,R1
135 022274 006301      ASL     R1
136 022276 052761 100000 003166  BIS     @BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
137 022304 005037 005770      CLR     EXTA          ;CLEAR ERROR EXTENSION FLAG.
138 022310 023727 002012 000001  CMP     L$UNIT,@1     ;ARE WE TESTING MULTIPLE UNITS?
139 022316 101416      BLOS   10$          ;BR IF NO.
140 022320          RFLAGS  RO          ;YES -- GET OPERATOR FLAGS.
      022320 104421      TRAP    C$RFLA
141 022322 032700 001000  BIT     @PNT,R0       ;SHOULD WE PRINT UNIT #?
142 022326 001412      BEQ    10$          ;BR IF NOT.
143 022330          PRINTF @PUNIT,UNITN      ;PRINT THE UNIT #
      022330 013746 002172  MOV     UNITN,-(SP)
      022334 012746 022422  MOV     @PUNIT,-(SP)
      022340 012746 000002  MOV     @2,-(SP)
      022344 010600      MOV     SP,R0
      022346 104417      TRAP    C$PNTF
      022350 062706 000006  ADD     @6,SP
144 022354          10$:
145 022354 005037 003104      CLR     NODEV
146 022360 013701 002176  MOV     CSRADDR,R1   ;ADDRESS OF FIRST REGISTER
147 022364 010102      MOV     R1,R2        ;START OF REGISTERS
148 022366 062702 000002  ADD     @TSSR,R2     ;ADDRESS OF TSSR REGISTER
149 022372 004737 016466  JSR    PC,XNXM       ;TEST BOTH CONTROLLER REGISTERS...
150 022376 103005      BCC   2$          ;...AND BR IF ALL OK.
151 022400 010137 003104  MOV     R1,NODEV     ;FLAG DEVICE AS NON-EXISTENT
152 022404 012737 177777 003102  MOV     @-1,DUFLG    ;DROP THIS UNIT.
153 022412          2$:
154
155      ;
156      ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
157
157 022412          5$:  SETPRI  @PRI00          ;ENABLE INTERRUPTS.
      022412 012700 000000  MOV     @PRI00,R0

```

INITIALIZE SECTION

```

158 022416 104441          TRAP  C$SPRI
      022420          ENDINIT
      022420          L10030:
159 022420 104411          TRAP  C$INIT
160 022422   045   116   045 PUNIT: .ASCIZ /%N%N%A***** TESTING UNIT %D2%A *****/
161          .EVEN
```

ADD AND DROP UNITS SECTIONS

```

163                                     .SBTTL  ADD AND DROP UNITS SECTIONS
164
165                                     ;**
166                                     ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
167                                     ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
168                                     ; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
169                                     ;--
170 022470                                BGNAU
170 022470                                L$AU::
171 022470 010001                          MOV     RO,R1                ; GET UNIT TO BE ADDED (RO)
172 022472 006301                          ASL     R1                  ; MAKE IT A WORD INDEX
173 022474 052761 100000 003166            BIS     @100000,ERTABL(R1) ; SET THE "ACTIVE" BIT
174 022502 042761 040000 003166            BIC     @%0000,ERTABL(R1) ; CLEAR THE "DROPPED" BIT
175 022510                                PRINTF  @1$,RO
175 022510 010046                          MOV     RO,-(SP)
175 022512 012746 022536                    MOV     @1$,-(SP)
175 022516 012746 000002                    MOV     @2,-(SP)
175 022522 010600                          MOV     SP,RO
175 022524 104417                          TRAP   C$PNTF
175 022526 062706 000006                    ADD     @6,SP
176 022532                                EXIT   AU
176 022532 000167                          .WORD  J$JMP
176 022534 000026                          .WORD  L10031-2-.
177 022536 045 116 045 1$:                .ASCIZ  /%N%A UNIT %D%A ADDED/
178                                     .EVEN
179
180 022564                                ENDAU                ; UNUSED.
180 022564                                L10031:
180 022564 104452                          TRAP   C$AU
181
182                                     ;**
183                                     ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
184                                     ; TO BE REMOVED FROM THE TEST LIST.
185                                     ;
186                                     ; SUPVSR DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN.
187                                     ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
188                                     ; COMMAND, OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
189                                     ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
190                                     ; WHICH ARE STILL ACTIVE.
191                                     ; UPON ENTRY, RO CONTAINS THE UNIT TO BE DROPPED.
192 022566                                BGNDU
192 022566                                L$DU::
193 022566 012737 177777 003102            MOV     @-1,DUFLG
194 022574 010001                          MOV     RO,R1
195 022576 006301                          ASL     R1
196 022600 052761 140000 003166            BIS     @140000,ERTABL(R1) ; SAY DROPPED
197 022606 000240 000240 000240            240,240,240           ; ??????????
198 022614                                PRINTF  @1$,RO
198 022614 010046                          MOV     RO,-(SP)
198 022616 012746 022642                    MOV     @1$,-(SP)
198 022622 012746 000002                    MOV     @2,-(SP)
198 022626 010600                          MOV     SP,RO
198 022630 104417                          TRAP   C$PNTF
198 022632 062706 000006                    ADD     @6,SP
199 022636                                EXIT   DU
199 022636 000167                          .WORD  J$JMP
199 022640 000030                          .WORD  L10032-2-.

```

ADD AND DROP UNITS SECTIONS

```

200 022642 045 116 045 1$: .ASCIZ /%N%A UNIT %D%A DROPPED/
201 .EVEN
202 022672 ENDDU
022672 L10032: TRAP C$DU
022672 104453
203 ;**
204 ; AUTO-DROP CODE SECTION.
205 ;--
206 022674 BGNAUTO
022674 L$AUTO::
207 022674 013705 002176 MOV CSRADDR,R5 ;POINT TO DEVICE REGISTER
208 022700 012703 000550 MOV #360.,R3 ;ENOUGH TIME FOR 2400' REEL TO REWIND
209 022704 004737 016340 10$: JSR PC,WAITF ;WAIT FOR SSR TO SET
210 022710 103420 BCS 20$ ;LEAVE WHEN SSR IS SET
211 022712 DELAY 250. ;WAIT FOR .25 SECONDS
022712 012727 000372 MOV #250.,(PC)+
022716 000000 .WORD 0
022720 013727 002116 MOV L$DLY,(PC)+
022724 000000 .WORD 0
022726 005367 177772 DEC -6(PC)
022732 001375 BNE .-4
022734 005367 177756 DEC -22(PC)
022740 001367 BNE .-20
212 022742 005303 DEC R3 ;BUMP COUNTER DOWN
213 022744 001357 BNE 10$ ;KEEP GOING
214 022746 004737 017272 JSR PC,CKDROP ;TRY AND DROP UNIT
215 022752 20$: ENDAUTO ; UNUSED.
216 022752 L10033: TRAP C$AUTU
022752 104461

```





CLEAN-UP AND REPORT CODING SECTIONS

```

023120 012746 000002      MOV      #2, -(SP)
023124 010600      MOV      SP,R0
023126 104416      TRAP     C1PNTS
023130 062706 000006      ADD      #6,SP
257 023134 000431      BR       4$
258 023136 020227 160001      3$:     CMP      R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
259 023142 001012      BNE     30$           ; BR IF NO.
260 023144      PRINTS  #DEVNRD,R3
023144 010346      MOV      R3, -(SP)
023146 012746 023433      MOV      #DEVNRD, -(SP)
023152 012746 000002      MOV      #2, -(SP)
023156 010600      MOV      SP,R0
023160 104416      TRAP     C1PNTS
023162 062706 000006      ADD      #6,SP
261 023166 000414      BR       4$
262 023170 042702 170000      30$:    BIC     #C7777,R2
263 023174      PRINTS  #DEVDRD,R3,R2
023174 010246      MOV      R2, -(SP)
023176 010346      MOV      R3, -(SP)
023200 012746 023514      MOV      #DEVDRD, -(SP)
023204 012746 000003      MOV      #3, -(SP)
023210 010600      MOV      SP,R0
023212 104416      TRAP     C1PNTS
023214 062706 000010      ADD      #10,SP
264 023220 062704 000002      4$:     ADD      #2,R4
265 023224 005203      INC      R3
266 023226 020427 003366      CMP      R4,#ERTABE
267 023232 103701      BLO     1$
268 023234 012604      MOV      (SP)+,R4
269 023236 012603      MOV      (SP)+,R3
270 023240 012602      MOV      (SP)+,R2
271 023242      ENDRPT      ; UNUSED.
023242 104425      L10035:    TRAP     C1RPT
272
273
274 023244      045      116      045  DEVSUM: .ASCIZ /#N#ADEVICE STATUS SUMMARY:#N/
275 023301      045      101      040  DEVONL: .ASCIZ /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
276 023351      045      101      040  DEVNXR: .ASCIZ /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
277 023433      045      101      040  DEVNRD: .ASCIZ /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
278 023514      045      101      040  DEVDRD: .ASCIZ /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
279      .EVEN
280
281 023564      ENDMOD
282
283

```

DS

CLEAN-UP AND REPORT CODING SECTIONS

1  
2  
9  
10 023564  
    023564  
16  
24

.TITLE TSV7 - HARDWARE TESTS 1-8  
BGNMOD TSV7  
TSV7::

TEST 1: INITIALIZE #4 TEST

```

26                                     .SBTTL TEST 1: INITIALIZE #4 TEST
27                                     ;+
28                                     ;
29                                     ;THIS TEST VERIFIES THAT WRITING INTO THE TSSR RETURNS THE
30                                     ;CONTROLLER TO ITS INITIALIZED STATE FROM VARIOUS CONDITIONS
31                                     ;(I.E. LOOPBACK ENABLED, FORCING WRONG PARITY, INVERTING SENSE OF
32                                     ;EXTENDED FEATURES SWITCH, ETC.)
33                                     ;
34                                     ;-
35 023564                                BGNTST
36 023564 012737 006356 002170          MOV     #EPRT1,EPRTSW          T1::
                                     ;SET UP PRIMARY ERROR MESSAGE
37
38                                     ;
39                                     ;
40                                     ;TEST 1
41                                     ;
42                                     ;
43                                     ;-
44
49 023572 004737 016274                JSR     PC,DSBINT          ;DISABLE INTERRUPTS
50 023576 012700 024524                MOV     #TST21ID,RO      ;ASCII MESSAGE TO IDENTIFY TEST
51 023602 004737 016600                JSR     PC,TSTSETUP      ;DO INITIAL TEST SETUP
52 023606 012737 000005 002206        MOV     #5,LOOPCNT       ;PERFORM 5 ITERATIONS
53 023614                                T21LOOP:
54 023614 004737 024546                JSR     PC,T21REST      ;SET COMMAND PACKET
55 023620 004737 024636                JSR     PC,T21RT2       ;SET UP OTHER COMMAND PACKET
56
57                                     ;*****
58                                     ;
59                                     ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
60                                     ;
61                                     ;*****
62
63 023624 012737 176750 024202        11$:  MOV     #65000.,T21DLY    ;SET DELAY ROUTINE
64 023632 004737 016064                JSR     PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
65 023636 103426                        BCS     20$              ;BR IF INIT WAS OK
66 023640                                DELAY   250              ;DELAY FOR A REWIND TO FINISH
                                     MOV     #250,(PC)+
                                     .WORD   0
67 023640 012727 000250                MOV     L#DLY,(PC)+
68 023644 000000                        .WORD   0
69 023646 013727 002116                MOV     .WORD   0
70 023652 000000                        .WORD   0
71 023654 005367 177772                DEC     -6(PC)
72 023660 001375                        BNE     -4
73 023662 005367 177756                DEC     -22(PC)
74 023666 001367                        BNE     -20
75 023670 005337 024202                DEC     T21DLY          ;BUMP COUNTER DOWN
76 023674 001356                        BNE     11$             ;BR, IF MORE TIME TO GO
77 023676 005237 002212                INC     FATFLG          ;BUMP COUNT
78 023702 010001                        MOV     RO,R1           ;CONTENTS OF TSSR REGISTER
79 023704                                ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
80 023704 104455                        TRAP   C#ERDF
81 023706 000145                        .WORD  101
82 023710 003650                        .WORD  SFIERR
83 023712 012124                        .WORD  SFIMSG
84
85 023714                                20$:
86 023714 012704 024160                MOV     #T21PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS

```



TEST 1: INITIALIZE #4 TEST

```

024112 000151
024114 024503
024116 015564
130 024120 004737 017272
131 024124 000241
132 024126 106037 024301
133 024132 001316
134 024134
024134 104406
135 024136 004737 016546
136 024142 103002
137 024144 000137 023614
138 024150
024150 104432
024152 000526

```

```

40$: JSR PC,CKDROP
      CLC
      RORB T21BS1
      BNE 25$
50$: CKLOOP
      JSR PC,TSTLOOP
      BCC 63$
      JMP T21LOOP
63$: EXIT TST

```

```

;TRY AND DROP UNIT
;DON'T LET CARRY SNEAK IN
;TRY NEXT "LOWEST" BIT POSITION
;LOOP UNTIL ALL EIGHT BITS TESTED
;SCOPE LOOP
      TRAP C$CLP1
;DO WE NEED TO ITERATE TEST
;BR, IF NO LOOP REQUIRED
;EXECUTE AGAIN
;ALL DONE THIS TEST
      TRAP C$EXIT
      .WORD L10036-.
      .WORD 105
      .WORD T21OFL
      .WORD EXPREC

```

TEST 1: INITIALIZE #4 TEST

140					
141			;	LOCAL STORAGE FOR THIS TEST	
142			;		
144	024160			.*<.+10>&177770	
146	024160		T21PACKET:		;COMMAND PACKET FOR TEST
147	024160	100004	.WORD	100004	;WRITE CHARACTERISTICS COMMAND, WITH, ACK
148	024162	024170	.WORD	T21DATA	;ADDRESS OF CHARACTERISTICS BLOCK
149	024164	000000	.WORD	0	
150	024166	000012	.WORD	10.	;STARTING VALUE OF BLOCK SIZE
151	024170		T21DATA:		;CHARACTERISTICS DATA BLOCK
152	024170	024204	.WORD	T21BFR	;ADDRESS OF MESSAGE BUFFER
153	024172	000000	.WORD	0	
154	024174	000024	.WORD	20.	;LENGTH OF MESSAGE BUFFER
155	024176	000000	.WORD	0	
156	024200	000000	T21DSW:	.WORD 0	;DRIVE SELECT WORD
157	024202	000000	T21DLY:	.WORD 0	;DELAY COUNTER
158	024204		T21BFR:	.BLKW 25.	;MESSAGE BUFFER
159			;		
160			;	WRITE SUBSYSTEM MEMORY COMMAND PACKET	
161			;		
163	024270			.*<.+10>&177770	
165	024270		T21PK2:		
166	024270	100206	.WORD	100206	;WRITE SUB SYS MEM COMMAND, IE AND ACK
167	024272	024300	.WORD	T21BF2	;ADDRESS OF SELECT BLOCK DATA
168	024274	000000	.WORD	0	
169	024276	000006	.WORD	6.	;SIZE OF DATA PACKET
170					
171			.EVEN		
172	024300		T21BF2:		
173	024300	000	T21BS0:	.BYTE 0	;BSELO AREA --- "COMMAND" BYTE
174	024301	000	T21BS1:	.BYTE 0	;BSEL1 AREA
175	024302	000000	T21S2:	.WORD 0	;SEL 2 AREA
176	024304	000000	T21S3:	.WORD 0	;DATA AREA
177					
178					
179					

## TEST 1: INITIALIZE #4 TEST

```

181
182
183      ;+
184      ;LOCAL TEXT MESSAGES FOR TEST
185      ;-
186 024306      127      122      111 T21SSR: .ASCIZ 'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
187 024403      124      123      123 T21AM3: .ASCIZ 'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
188 024503      104      162      151 T21OFL: .ASCIZ 'Drive is OFFLINE'
189 024524      111      156      151 TST21ID: .ASCIZ 'Initialization #4'
190
191
192      ;+
193      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
194      ;WRITE SUBSYSTEM MEMORY COMMAND
195      ;
196      ;-
197 024546      T21REST:
198 024546      SAVREG      ;SAVE THE REGISTERS
199 024552      012701      024160      MOV      #T21PACKET,R1      ;START OF THE PACKET
200 024556      012721      100004      MOV      #100004,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,
201 024562      012721      024170      MOV      #T21DATA,(R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
202 024566      005021      CLR      (R1)+      ;EXTENDED ADDRESS
203 024570      012721      000010      MOV      #8,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
204 024574      012721      024204      MOV      #T21BFR,(R1)+      ;ADDRESS OF MESSAGE BUFFER
205 024600      005021      CLR      (R1)+
206 024602      012721      000024      MOV      #20,(R1)+      ;LENGTH OF MESSAGE BUFFER
207 024606      005021      CLR      (R1)+
208 024610      005011      CLR      (R1)
209 024612      012702      000020      MOV      #20,R2      ;NUMBER OF LOCATIONS TO BE CLEARED
210 024616      012762      177777      024204      64$: MOV      #177777,T21BFR(R2)      ;ALL ONES TO MESSAGE BUFFER
211 024624      005742      TST      -(R2)      ;NEXT LOCATION
212 024626      020227      000000      CMP      R2,#0      ;CHECK R2 FOR ZERO
213 024632      001371      BNE      64$      ;BR, IF NOT AT ZERO YET
214 024634      000207      RTS      PC      ;RETURN
215
216 024636      T21RT2:
217 024636      SAVREG      ;SAVE THE REGISTERS
218 024642      012701      024270      MOV      #T21PK2,R1      ;START OF THE PACKET
219 024646      012721      100206      MOV      #100206,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
220 024652      012721      024300      MOV      #T21BF2,(R1)+      ;ADDRESS OF DATA BLOCK
221 024656      005021      CLR      (R1)+      ;EXTENDED ADDRESS
222 024660      012721      000006      MOV      #6,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
223 024664      005021      CLR      (R1)+
224 024666      012701      024300      MOV      #T21BF2,R1      ;ADDRESS OF DATA FOR WRT SUB SYS MEM
225 024672      005021      CLR      (R1)+
226 024674      005011      CLR      (R1)
227 024676      000207      RTS      PC      ;RETURN
228 024700      ENDTST
      024700
      024700      104401
      L10036:
      TRAP      C$ETST

```





TEST 2: OFF-LINE AND REJECT REWIND

```

288
289
290 ;*****
291 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
292 ;
293 ;*****
294
295 024774 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
296 025000 103407 BCS 23$ ;BR, IF COMMAND ISSUED OK
297 025002 005237 002212 INC FATFLG ;BUMP COUNT
301 025006 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
302 025010 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
; TRAP C$ERHRD
; .WORD 202
; .WORD WRTMSG
; .WORD SFIMSG
303 025020 23$: CKLOOP ;
; TRAP C$CLP1
; .WORD 203
; .WORD T22WLK
; .WORD SFIMSG
304 025022 013701 026300 MOV T22BFR+6,R1 ;PICK UP XTSSO
305 025026 032701 000004 BIT #4,R1 ;IS UNIT WRITE-LOCKED?
306 025032 001407 BEQ 24$ ;NO,PROCEED WITH TESTING
307 025034 005237 002212 INC FATFLG ;BUMP COUNT
311 025040 ERRDF ERRNO,T22WLK,SFIMSG ;TAPE IS WRITE LOCKED
; TRAP C$ERDF
; .WORD 203
; .WORD T22WLK
; .WORD SFIMSG
312 025050 DOCLN ;
; TRAP C$DCLN
; .WORD 204
; .WORD T22SSR
; .WORD PKTSSR
313 025052 24$: CKLOOP ;
; TRAP C$CLP1
; .WORD 204
; .WORD T22SSR
; .WORD PKTSSR
314 025054 005737 002216 TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
315 025060 001041 BNE 50$ ;BR IF SWITCH IS ON
316 025062 112737 000200 026371 MOV #200,T22BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
317 025070 112737 000010 026370 MOV #10,T22BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
318 025076 012704 026360 MOV #T22PK2,R4 ;WRITE SUBSYS MEM PACKET
319 025102 010465 000000 MOV R4,TSD8(R5) ;ISSUE COMMAND
320 025106 004737 016426 JSR PC,CHKTSSR ;WAIT FOR SSR
321 025112 103407 BCS 30$ ;BR, IF NO ERROR
322 025114 010001 MOV RO,R1 ;ERROR, SAVE TSSR
323 025116 005237 002212 INC FATFLG ;BUMP COUNT
327 025122 ERRHRD ERRNO,T22SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
; TRAP C$ERHRD
; .WORD 204
; .WORD T22SSR
; .WORD PKTSSR
328 025132 30$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
; .WORD 204
; .WORD T22SSR
; .WORD PKTSSR
329 025134 012704 026250 MOV #T22PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
330
331 ;*****
332 ;
333 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
334 ;
335 ;*****
336
337 025140 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS

```

TEST 2: OFF-LINE AND REJECT REWIND

18

```

338 025144 103407          BCS      50$              ;BR, IF COMMAND ISSUED OK
339 025146 005237 002212  INC      FATFLG          ;BUMP COUNT
343 025152 010001          MOV      R0,R1           ;SAVE CONTENTS OF TSSR
344 025154          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
    025154 104456                                TRAP      C$ERHRD
    025156 000315                                .WORD    205
    025160 005054                                .WORD    WRTMSG
    025162 012124                                .WORD    SFIMSG
345 025164          50$:    CKLOOP            ;SCOPE LOOP
    025164 104406                                TRAP      C$CLP1
346 025166 016501 000002  MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
347 025172 032701 000100  BIT      #0FL,R1         ;CHECK FOR THE OFFLINE BIT SET
348 025176 001006          BNE      60$             ;BR, IF OFFLINE (GOOD)
349 025200 005237 002212  INC      FATFLG          ;BUMP COUNT
353 025204          ERRDF  ERRNO,T22OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
    025204 104455                                TRAP      C$ERDF
    025206 000316                                .WORD    206
    025210 026605                                .WORD    T22OFL
    025212 012124                                .WORD    SFIMSG
354 025214          60$:    CKLOOP            ;LOOP IF SELECTED
    025214 104406                                TRAP      C$CLP1
355 025216 012703 026376  MOV      #T22RD,R3      ;POINTER FOR COMMANDS
356 025222 011337 026360  65$:    MOV      (R3),T22PK2 ;TAPE MOTION COMMAND IN PLACE
357 025226 012704 026360  MOV      #T22PK2,R4     ;R4 = POINTER TO PACKET
358 025232 010465 000000  MOV      R4,TSDR(R5)    ;ISSUE COMMAND
359 025236 004737 016340  JSR      PC,WAITF       ;WAIT FOR SSR TO SET
360 025242 016501 000002  MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
361 025246 012702 100306  MOV      #SSR!SC!OFL!BIT1!BIT2,R2 ;SET UP EXPECTED
362 025252 020102          CMP      R1,R2          ;ARE THEY EQUAL
363 025254 001406          BEQ      80$             ;BR, IF OK ESP. FUNCTION REJECT
364 025256 005237 002212  INC      FATFLG          ;BUMP COUNT
368 025262          ERRHRD  ERRNO,T22TM,EXPREC ;TSSR INCORRECT AFTER TAPE MOTION CMD
    025262 104456                                TRAP      C$ERHRD
    025264 000317                                .WORD    207
    025266 026660                                .WORD    T22TM
    025270 015564                                .WORD    EXPREC
369 025272          80$:    CKLOOP            ;LOOP IF SELECTED
    025272 104406                                TRAP      C$CLP1
370 025274 005723          TST      (R3)+          ;POINT TO NEXT COMMAND
371 025276 022713 177777  CMP      #177777,(R3)   ;END OF THE COMMANDS YET
372 025302 001401          BEQ      90$             ;BR, IF DONE
373 025304 000746          BR       65$             ;MORE COMMAND(S) TO GO
374 025306          90$:    ENDSUB            ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>
375 025306          U10040:
    025306 104403                                TRAP      C$ESUB
376 025310 023727 002212 000017  CMP      FATFLG,#15     ;IS ERROR COUNT AT 25
377 025316 103402          BLO      999$           ;BR, IF LESS THAN 25
378 025320 004737 017272  JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
379 025324          999$:

```





{ 3 }

TEST 2: OFF-LINE AND REJECT REWIND

SEQ 0105

```

025630
025630 104403
483 025632 023727 002212 000017      CMP      FATFIG.#15.
484 025640 103402                      BLU      999$
485 025642 004737 017272                      JSR      PC,CKDROP
486 025646                                      999$:

```

```

110041:      TRAP      C$ESUB
;IS ERROR COUNT AT 25
;ER, IF LESS THAN 25
;TRY TO DROP THE UNIT

```



## TEST 2: OFF-LINE AND REJECT REWIND

```

541 025774 010001          MOV      R0,R1          ;ERROR, SAVE TSSR
542 025776 005237 002212  INC      FATFLG        ;BUMP COUNT
546 026002          ERRHRD  ERRNO,T22SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
                                TRAP      C$ERHRD
                                .WORD    216
                                .WORD    T22SSR
                                .WORD    PKTSSR
    026002 104456
    026004 000330
    026006 026410
    026010 012136
547 026012          30$:   CKLOOP          ;LOOP IF SELECTED
    026012 104406
548 026014 012704 026250  MOV      #T22PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
549
550 ;*****
551 ;
552 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
553 ;
554 ;*****
555
556 026020 004737 010752          JSR      PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
557 026024 103407          BCS     50$              ;BR, IF COMMAND ISSUED OK
558 026026 005237 002212  INC      FATFLG        ;BUMP COUNT
562 026032 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
563 026034          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP      C$ERHRD
                                .WORD    217
                                .WORD    WRTMSG
                                .WORD    SFIMSG
    026034 104456
    026036 000331
    026040 005054
    026042 012124
564 026044          50$:   CKLOOP          ;SCOPE LOOP
    026044 104406
                                TRAP      C$CLP1
565 026046 016501 000002  MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
566 026052 032701 000100  BIT      #OFL,R1        ;CHECK FOR THE OFFLINE BIT SET
567 026056 001006          BNE     60$              ;BR, IF OFFLINE (GOOD)
568 026060 005237 002212  INC      FATFLG        ;BUMP COUNT
572 026064          ERRDF  ERRNO,T22OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
                                TRAP      C$ERDF
                                .WORD    218
                                .WORD    T22OFL
                                .WORD    SFIMSG
    026064 104455
    026066 000332
    026070 026605
    026072 012124
573 026074          60$:   CKLOOP          ;LOOP IF SELECTED
    026074 104406
                                TRAP      C$CLP1
574 026076 012737 142010 026360 65$:   MOV      #142010,T22PK2 ;POSITION: COMMAND (REWIND MODE) CVC=1
575 026104 012704 026360          MOV      #T22PK2,R4    ;R4 = POINTER TO PACKET
576 026110 010465 000000          MOV      R4,TSD8(R5)  ;ISSUE COMMAND
577 026114 004737 016340          JSR      PC,WAITF      ;WAIT FOR SSR TO SET
578 026120 016501 000002  MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
579 026124 012702 100306  MOV      #SSR!OFL!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
580 026130 020102          CMP     R1,R2          ;ARE THEY EQUAL
581 026132 001406          BEQ     80$              ;BR, IF OK ESP. FUNCTION REJECT
582 026134 005237 002212  INC      FATFLG        ;BUMP COUNT
586 026140          ERRHRD  ERRNO,T22RWJ,EXPREC ;TSSR INCORRECT AFTER TAPE MOTION CMD
                                TRAP      C$ERHRD
                                .WORD    219
                                .WORD    T22RWJ
                                .WORD    EXPREC
    026140 104456
    026142 000333
    026144 026754
    026146 015564
587 026150          80$:   CKLOOP          ;LOOP IF SELECTED
    026150 104406
                                TRAP      C$CLP1
588 026152 012703 026272  MOV      #T22BFR,R3    ;POINTER TO MESSAGE BUFFER
589 026156 016301 000006  MOV      XST0(R3),R1   ;PICK UP XST0 FROM MESSAGE BUFFER

```

[9]

TEST 2: OFF-LINE AND REJECT REWIND

```

590 026162 010102          MOV     R1,R2          ;SET UP EXPECTED
591 026164 042702 000020   BIC     #BIT4,R2     ;VCK SHOULD BE CLEAR
592 026170 020102          CMP     R1,R2        ;ARE THEY EQUAL
593 026172 001406          BEQ     90$          ;BR, IF OK (GOOD)
594 026174 005237 002212   INC     FATFLG       ;BUMP COUNT
598 026200          ERRHRD  ERRNO,T22VCK,EXPREC ;VCK WASN'T CLEAR (BAD)
        026200 104456
        026202 000334          TRAP   C$ERHRD
        026204 027027          .WORD  220
        026206 015564          .WORD  T22VCK
599 026210          90$:          .WORD  EXPREC
600 026210          ENDSUB          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
        026210 104403          L10042:
601 026212 023727 002212 000017   CMP     FATFLG,#15.  ;IS ERROR COUNT AT 25
602 026220 103402          BLO     999$         ;BR, IF LESS THAN 25
603 026222 004737 017272   JSR     PC,CKDROP    ;TRY TO DROP THE UNIT
604 026226          999$:
605 026226 004737 016546   JSR     PC,TSTLOOP
606 026232 103002          BCC     163$
607 026234 000137 024732   JMP     T22LOOP
608 026240          163$:  EXIT   TST
        026240 104432          ;DO WE NEED TO ITERATE TEST
        026242 001116          ;BR, IF NO LOOP REQUIRED
                                ;EXECUTE AGAIN
                                ;ALL DONE THIS TEST
                                TRAP   C$EXIT
                                .WORD  L10037-
    
```



TEST 2: OFF-LINE AND REJECT REWIND

SEQ 0109

```

610
611
612
614      026250
616 026250 100204
617 026250 100204
618 026252 026260
619 026254 000000
620 026256 000012
621 026260
622 026260 026272
623 026262 000000
624 026264 000024
625 026266 000000
626 026270 000007
627 026272
628
629
630
632      026360
634 026360 100206
635 026360 100206
636 026362 026370
637 026364 000000
638 026366 000006
639
640
641 026370
642 026370 000
643 026371 000
644 026372 000000
645 026374 000000
646
647
648
649
650 026376 100201
651 026400 100205
652 026402 100210
653 026404 100211
654 026406 177777
655
656

```

```

;+
;LOCAL STORAGE FOR THIS TEST
;-
      .<.+10>&177770
T22PACKET:
      .WORD 100204
      .WORD T22DATA
      .WORD 0
      .WORD 10.
T22DATA:
      .WORD T22BFR
      .WORD 0
      .WORD 20.
      .WORD 0
      .WORD 7
T22BFR: .BLKW 25.
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
      .<.+10>&177770
T22PK2:
      .WORD 100206
      .WORD T22BF2
      .WORD 0
      .WORD 6.
      .EVEN
T22BF2:
T22BS0: .BYTE 0
T22BS1: .BYTE 0
T22S2: .WORD 0
T22S3: .WORD 0
;
;
      .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T22RD: .WORD 100201
T22WRT: .WORD 100205
T22POS: .WORD 100210
T22FOR: .WORD 100211
      .WORD 177777

```

```

;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
;ADDRESS OF CHARACTERISTICS BLOCK

;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER

;LENGTH OF MESSAGE BUFFER

;SELECT DRIVE 7
;MESSAGE BUFFER

;WRITE SUB SYS MEM COMMAND, IE AND ACK
;ADDRESS OF SELECT BLOCK DATA

;SIZE OF DATA PACKET

;BSELO AREA
;BSEL1 AREA
;SEL 2 AREA
;DATA AREA

;READ TAPE FORWARD
;WRITE TAPE FORWARD
;POSITION TAPE
;FORMAT TAPE
;END OF DATA

```

TEST 2: OFF-LINE AND REJECT REWIND

```

658
659
660          ;+
661          ;LOCAL TEXT MESSAGES FOR TEST
662          ;-
663 026410    127    122    111 T22SSR: .ASCIZ 'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
664 026505    124    123    123 T22AM3: .ASCIZ 'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
665 026605    104    162    151 T22OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
666 026660    124    123    123 T22TM:  .ASCIZ 'TSSR Incorrect After Tape Motion Command To Off-Line Device'
667 026754    124    123    123 T22RWJ: .ASCIZ 'TSSR Not Correct After REWIND With VCK Set'
668 027027    103    126    103 T22VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
669 027102    052    052    052 T22WLK: .ASCIZ '*****TAPE IS WRITE-LOCKED AND WILL CAUSE ERRORS*****'
670 027167    117    146    146 T22ID:  .ASCIZ 'Off-Line And Reject Rewind'
671
672          .EVEN
673
674          ;+
675          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
676          ;WRITE SUBSYSTEM MEMORY COMMAND
677          ;-
678 027222    T22REST:
679 027222          SAVREG          ;SAVE THE REGISTERS
680 027226    012701 026250          MOV          #T22PACKET,R1      ;START OF THE PACKET
681 027232    012721 100204          MOV          #100204,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
682 027236    012721 026260          MOV          #T22DATA,(R1)+    ;ADDRESS OF CHARACTERISTICS DATA BLOCK
683 027242    005021          CLR          (R1)+             ;EXTENDED ADDRESS
684 027244    012721 000012          MOV          #10.,(R1)+        ;SIZE OF DATA BLOCK IN BYTES
685 027250    012721 026272          MOV          #T22BFR,(R1)+    ;ADDRESS OF MESSAGE BUFFER
686 027254    005021          CLR          (R1)+
687 027256    012721 000024          MOV          #20.,(R1)+        ;LENGTH OF MESSAGE BUFFER
688 027262    005021          CLR          (R1)+
689 027264    012711 000007          MOV          #7,(R1)           ;SELECT DRIVE SEVEN
690 027270    012702 000020          MOV          #20,R2            ;NUMBER OF LOCATIONS TO BE CLEARED
691 027274    012762 177777 026272 64$: MOV          #177777,T22BFR(R2) ;ALL ONES TO MESSAGE BUFFER
692 027302    005742          TST          -(R2)             ;BUMP R2 DOWN
693 027304    020227 000000          CMP          R2,#0             ;IS R2 AT ZERO YET
694 027310    001371          BNE          64$               ;KEEP GOING UNTIL DONE
695 027312    000207          RTS          PC                ;RETURN
696
697 027314    T22RT2:
698 027314          SAVREG          ;SAVE THE REGISTERS
699 027320    012701 026360          MOV          #T22PK2,R1        ;START OF THE PACKET
700 027324    012721 100206          MOV          #100206,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
701 027330    012721 026370          MOV          #T22BF2,(R1)+    ;ADDRESS OF DATA BLOCK
702 027334    005021          CLR          (R1)+             ;EXTENDED ADDRESS
703 027336    012721 000006          MOV          #6.,(R1)+         ;SIZE OF DATA BLOCK IN BYTES
704 027342    005021          CLR          (R1)+
705 027344    012701 026370          MOV          #T22BF2,R1        ;POINT TO DATA SEL AREA
706 027350    005021          CLR          (R1)+
707 027352    005011          CLR          (R1)
708 027354    005011          CLR          (R1)
709 027356    000207          RTS          PC                ;LAST LOC TO BE CLEARED
710 027360          ENDTST          ;RETURN
711          027360    104401          L10037:    TRAP          C$ETST

```

```
713                .SBTTL TEST 3: BASIC WRITE DATA  
714                ;+  
715                ;  
716                ;THIS TEST VERIFIES THAT THE WRITE DATA (NEXT) COMMAND OPERATES  
717                ;PROPERLY, UP TO THE POINT OF CHECKING THAT THE DATA WAS ACTUALLY  
718                ;WRITTEN ONTO THE TAPE CORRECTLY. CHECKING IN THIS TEST IS  
719                ;LIMITED TO VERIFYING THAT THE COMMAND TERMINATED CORRECTLY WITH  
720                ;THE CORRECT REGISTER, MESSAGE BUFFER AND RAM CONTENTS.  
721                ;  
722                ;THE TEST CONSISTS OF THE FOLLOWING 7 SUBTESTS  
723                ;  
724                ;  
725                ;  
726                ;-  
727 027362                RGNTST  
728 027362                T3:;  
027362 012737 006356 002170    MOV    $EPR1,EPR1    ;SET UP PRIMARY ERROR MESSAGE  
027370 005037 003124    CLR    KTENABLE    ;TURN OFF KT11  
027374 004737 017364    JSR    PC,KTOFF    ;TURN OFF KT11  
027400 004737 016274    JSR    PC,DSBINT    ;DISABLE INTERRUPTS  
027404 012700 033766    MOV    $TST23ID,R0    ;ASCII MESSAGE TO IDENTIFY TEST  
027410 004737 016600    JSR    PC,TSTSETUP    ;DO INITIAL TEST SETUP  
027414 004737 021366    JSR    PC,MEMCK    ;CHECK FOR MEMORY  
027420 012737 000005 002206    MOV    $5,LOOPCNT    ;PERFORM 5 ITERATIONS  
740                ;+  
741                ;  
742                ;TEST 3, SUBTEST 1  
743                ;  
744                ;VERIFIES THAT A WRITE COMMAND (ANY VALID MODE CODE)  
745                ;WITH THE CLEAR VOLUME CHECK (CVC) BIT CLEAR IS  
746                ;REJECTED IF THE VOLUME CHECK (VCK) FLAG IS SET. ALL  
747                ;VALID MODE CODES ARE CHECKED (WRITE DATA, WRITE  
748                ;RETRY).  
749                ;-  
750                ;  
751 027426                T23LOOP:  
752 027426                BGNSUB    ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>  
027426                T3.1:  
027426 104402                TRAP    C$BSUB  
753 027430 004737 034002    JSR    PC,T23REST    ;SET COMMAND PACKET  
754 027434 004737 034074    JSR    PC,T23RT2    ;SET UP OTHER COMMAND PACKET  
755                ;  
756                ;*****  
757                ;  
758                ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR  
759                ;  
760                ;*****  
761                ;  
762 027440 004737 016064    JSR    PC,SOFINIT    ;DO INITIALIZE ON CONTROLLER  
763 027444 103407    BCS    20$    ;BR IF INIT WAS OK  
764 027446 005237 002212    INC    FATFLG    ;BUMP COUNT  
765 027452 010001    MOV    R0,R1    ;CONTENTS OF TSSR REGISTER  
766 027454                ERRDF    ERRNO,SFIERR,SFIMSG    ;FATAL ERROR TSSR WAS NOT OK  
027454 104455                TRAP    C$ERDF  
027456 000455                .WORD    301  
027460 003650                .WORD    SFIERR  
027462 012124                .WORD    SFIMSG
```

TEST 3: BASIC WRITE DATA

```

770 027464
771 027464 012737 000007 032510 20$: MOV #7,T23DSW ;SET DRIVE NUMBER IN PACKET
772 027472 012704 032470 MOV #T23PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
773
774 ;*****
775 ;
776 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
777 ;
778 ;*****
779
780 027476 004737 010752 JSR PC,WRTPHR ;ISSUE WRITE CHARACTERISTICS
781 027502 103407 BCS 23$ ;BR, IF COMMAND ISSUED OK
782 027504 005237 002212 INC FATFLG ;BUMP COUNT
786 027510 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
787 027512 ERRHRD ERRNO,WRTPHR,SFIMSG ;WRITE CHARACTERISTICS FAILED
027512 104456 TRAP C$ERRHD
027514 000456 .WORD 302
027516 005054 .WORD WRTPHR
027520 012124 .WORD SFIMSG
788 027522 005737 002216 23$: TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
789 027526 001044 RNE 50$ ;BR IF SWITCH IS ON
790
791 027530 112737 000200 032623 MOVW #200,T23BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
792 027536 112737 000010 032622 MOVW #10,T23BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
793 027544 012704 032600 MOV #T23PK2,R4 ;WRITE SUBSYS MEM PACKET
794 027550 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
795 027554 004737 016426 JSR PC,CHKTSSR ;WAIT FOR SSR
796 027560 103407 BCS 30$ ;BR, IF NO ERROR
797 027562 010001 MOV RO,R1 ;ERROR, SAVE TSSR
798 027564 005237 002212 INC FATFLG ;BUMP COUNT
802 027570 ERRHRD ERRNO,T23SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
027570 104456 TRAP C$ERRHD
027572 000457 .WORD 303
027574 032644 .WORD T23SSR
027576 012136 .WORD PKTSSR
803 027600 30$: CKLOOP ;LOOP IF SELECTED
027600 104406 TRAP C$CLP1
804 027602 012737 000007 032510 MOV #7,T23DSW ;SET DRIVE NUMBER IN PACKET
805 027610 012704 032470 MOV #T23PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
806
807 ;*****
808 ;
809 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
810 ;
811 ;*****
812
813 027614 004737 010752 JSR PC,WRTPHR ;ISSUE WRITE CHARACTERISTICS
814 027620 103407 BCS 50$ ;BR, IF COMMAND ISSUED OK
815 027622 005237 002212 INC FATFLG ;BUMP COUNT
819 027626 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
820 027630 ERRHRD ERRNO,WRTPHR,SFIMSG ;WRITE CHARACTERISTICS FAILED
027630 104456 TRAP C$ERRHD
027632 000460 .WORD 304
027634 005054 .WORD WRTPHR
027636 012124 .WORD SFIMSG
821 027640 50$: CKLOOP ;SCOPE LOOP
027640 104406 TRAP C$CLP1
    
```

## TEST 3: BASIC WRITE DATA

```
822 027642 016501 000002        MOV     TSSR(R5),R1          ;GET TSSR CONTENTS
823 027646 032701 000100        BIT     #OFL,R1            ;CHECK FOR THE OFFLINE BIT SET
824 027652 001006                BNE     60$                ;BR, IF OFFLINE (GOOD)
825 027654 005237 002212        INC     FATFLG             ;BUMP COUNT
829 027660                ERRDF  ERRNO,T23OFL,SFMSG    ;OFF LINE SHOULD HAVE BEEN SET (BAD)
      027660 104455                    TRAP   C$ERDF
      027662 000461                    .WORD 305
      027664 033306                    .WORD T23OFL
      027666 012124                    .WORD SFMSG
830 027670                60$:  CKLOOP                    ;LOOP IF SELECTED
      027670 104406                    TRAP   C$CLP1
831 027672 012703 032634        MOV     #T23WD,R3          ; POINTER FOR COMMANDS
832 027676 011337 032600        65$:  MOV     (R3),T23PK2    ; TAPE MOTION COMMAND IN PLACE
833 027702 012704 032600        MOV     #T23PK2,R4        ; R4 = POINTER TO PACKET
834 027706 010465 000000        MOV     R4,TSDB(R5)       ; ISSUE COMMAND
835 027712 004737 016340        JSR     PC,WAITF          ; WAIT FOR SSR TO SET
836 027716 016501 000002        MOV     TSSR(R5),R1       ; GET TSSR CONTENTS
837 027722 012702 100306        MOV     #SSR!SC!OFL!BIT1!BIT2,R2 ; SET UP EXPECTED
838 027726 020102                CMP     R1,R2             ; ARE THEY EQUAL
839 027730 001406                BEQ     80$                ; BR, IF OK ESP. FUNCTION REJECT
840 027732 005237 002212        INC     FATFLG             ; BUMP COUNT
844 027736                ERRHRD ERRNO,T23TM,EXPREC    ; TSSR INCORRECT AFTER TAPE MOTION CMD
      027736 104456                    TRAP   C$ERHRD
      027740 000462                    .WORD 306
      027742 033042                    .WORD T23TM
      027744 015564                    .WORD EXPREC
845 027746                80$:  CKLOOP                    ; LOOP IF SELECTED
      027746 104406                    TRAP   C$CLP1
846 027750 005723                TST     (R3)+              ; POINT TO NEXT COMMAND
347 027752 022713 177777        CMP     #177777,(R3)      ; END OF THE COMMANDS YET
848 027756 001401                BEQ     90$                ; BR, IF DONE
849 027760 000746                BR      65$                ; MORE COMMAND(S) TO GO
850 027762                90$:
851 027762                ENDSUB                    ; >>>>>>>>>> END SUBTEST >>>>>>>>>>
      027762                L10044:
      027762 104403                    TRAP   C$ESUB
852 027764 023727 002212 000017  CMP     FATFLG,#15        ; IS ERROR COUNT AT 25
853 027772 103402                BLO     999$                ; BR, IF LESS THAN 25
854 027774 004737 017272        JSR     PC,CKDROP         ; TRY TO DROP THE UNIT
855 030000                999$:
```



## TEST 3: BASIC WRITE DATA

```

911 030074 103407          BCS      23$          ;BR, IF COMMAND ISSUED OK
912 030076 005237 002212  INC      FATFLG      ;BUMP COUNT
916 030102 010001          MOV      R0,R1       ;SAVE CONTENTS OF TSSR
917 030104          ERR:RD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      030104 104456          TRAP      C$ERHRD
      030106 000464          .WORD    308
      030110 005054          .WORD    WRTMSG
      030112 012124          .WORD    SFIMSG
918 030114          23$:   CKLOOP          ;LOOP IF SELECTED
      030114 104406          TRAP      C$CLP1
919
920          ;*****
921          ;
922          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
923          ;
924          ;*****
925
926 030116 004737 011104          JSR      PC,REWIND    ;CALL THE TAPE REWIND
927 030122 012703 000024          MOV      #20.,R3     ;STARTING RECORD SIZE
928 030126 013737 003114 032612 65$:   MOV      FREE,T23WB  ;STARTING WRITE BUFFER ADDRESS
929
930          ;*****
931          ;
932          ;WRITE DATA,CVC=1,ACK COMMAND
933          ;
934          ;*****
935
936 030134 012737 140005 032610  MOV      #140005,T23PK3 ;WRITE DATA,CVC=1,ACK COMMAND
937 030142 012737 140005 032632  MOV      #140005,T23WRT ;SETUP FOR RETRY COMMAND
938 030150 052737 004000 032632  BIS      #4000,T23WRT  ;MAKE IT A RETRY
939 030156 012704 032610          MOV      #T23PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
940 030162 010300          MOV      R3,R0       ;SET PATTERN IN CORRECT REGISTER
941 030164 004737 017512          JSR      PC,FILLMEM   ;FILL MEMORY WITH RECORD SIZE
942 030170 010337 032616          MOV      R3,T23SZ    ;SET UP RECORD SIZE IN PACKET
943 030174 010465 000000          MOV      R4,TSDB(R5) ;ISSUE COMMAND
944 030200 004737 016340          JSR      PC,WAITF    ;WAIT FOR SSR TO SET
945 030204 016501 000002          MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
946 030210 012702 000200          MOV      #SSR,R2    ;SET UP EXPECTED
947 030214 020102          CMP      R1,R2       ;ARE THEY EQUAL
948 030216 001402          BEQ     80$          ;BR, IF OK
949 030220 004737 034162          JSR      PC,T23CHK   ;CHECK SPECIAL CONDITION
950 030224          80$:   CKLOOP          ;LOOP IF SELECTED
      030224 104406          TRAP      C$CLP1
951 030226 016501 000000          MOV      TSBA(R5),R1 ;GET TSBA CONTENTS
952 030232 012702 032512          MOV      #T23BFR,R2 ;SET UP EXPECTED
953 030236 062702 000016          ADD     #16,R2       ;SET TO END OF MESSAGE BUFFER
954 030242 005737 002216          TST     EXTFEA       ;CHECK FOR EXTENDED FEATURES SW SET
955 030246 001402          BEQ     85$          ;BR, IF IT NOT SET
956 030250 062702 000002          ADD     #2,R2        ;BUMP R2 FOR EXTRA DATA
957 030254 020102          85$:   CMP      R1,R2       ;ARE THEY EQUAL
958 030256 001406          BEQ     90$          ;BR, IF TSBA IS CORRECT
959 030260 005237 002212          INC     FATFLG      ;BUMP COUNT
963 030264          ERRHRD  ERRNO,T23BA,EXPREC ;TSBA WAS NOT CORRECT AFTER WRITE DATA
      030264 104456          TRAP      C$ERHRD
      030266 000465          .WORD    309
      030270 033625          .WORD    T23BA
      030272 015564          .WORD    EXPREC

```

## TEST 3: BASIC WRITE DATA

```

964 030274          90$:  CKLOOP          ;LOOP IF SELECTED
      030274 104406          TRAP      C$CLP1
965 030276 020327 007376      CMP      R3,#7376      ;ONLY CHECK RAM UNTIL ITS FULL.
966 030302 002114          BGE      115$          ;IT WRAPS AROUND ETC.
967 030304 004737 034074      JSR      PC,T23RT2      ;MAKE SURE PACKET AND DATA ARE CLEAN
968 030310 012737 000400 032624  MOV      #256.,T23S2      ;STARTING RAM ADDRESS
969 030316 112737 000000 032622  MOVB     #0,T23BS0      ;STOP INTERNAL TSV05 DIAGNOSTICS
970 030324 112737 000000 032623  MOVB     #0,T23BS1      ;SIZE OF RAM READ
971 030332 012704 032600      MOV      #T23PK2,R4      ;SET R4 WITH PACKET ADDRESS
972 030336 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE WRITE SUB SYS MEM COMMAND
973 030342 004737 016426      JSR      PC,CHKTSSR      ;CHECK TSSR AND WAIT FOR SSR TO SET
974 030346 103407          BCS      92$          ;BR, IF NO ERRORS IN TSSR
975 030350 010001          MOV      R0,R1          ;SAVE TSSR
976 030352 005237 002212      INC      FATFLG         ;BUMP COUNT
980 030356          ERRHRD  ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
      030356 104456          TRAP      C$ERHRD
      030360 000466          .WORD    310
      030362 033677          .WORD    T23WSS
      030364 012136          .WORD    PKTSSR
981 030366          92$:  CKLOOP          ;LOOP IF SELECTED
      030366 104406          TRAP      C$CLP1
982 030370 004737 034074      JSR      PC,T23RT2      ;MAKE SURE PACKET AND DATA ARE CLEAN
983 030374 012737 000400 032624  MOV      #256.,T23S2      ;STARTING RAM ADDRESS
984 030402 112737 000001 032622  MOVB     #1,T23BS0      ;READ RAM COMMAND FOR WRITE SUB SYS M.
985 030410 112737 000002 032623  MOVB     #2,T23BS1      ;SIZE OF RAM READ
986 030416 012704 032600      MOV      #T23PK2,R4      ;SET R4 WITH PACKET ADDRESS
987 030422 010465 000000 95$:  MOV      R4,TSDB(R5)      ;ISSUE WRITE SUB SYS MEM COMMAND
988 030426 004737 016426      JSR      PC,CHKTSSR      ;CHECK TSSR AND WAIT FOR SSR TO SET
989 030432 103407          BCS      100$         ;BR, IF NO ERRORS IN TSSR
990 030434 010001          MOV      R0,R1          ;SAVE TSSR
991 030436 005237 002212      INC      FATFLG         ;BUMP COUNT
995 030442          ERRHRD  ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
      030442 104456          TRAP      C$ERHRD
      030444 000467          .WORD    311
      030446 033677          .WORD    T23WSS
      030450 012136          .WORD    PKTSSR
996 030452          100$: CKLOOP          ;LOOP IF SELECTED
      030452 104406          TRAP      C$CLP1
997 030454 005001          CLR      R1          ;CLEAR REGISTER
998 030456 005002          CLR      R2          ;CLEAR REGISTER
999 030470 013701 032532      MOV      T23BFR+20,R1      ;PICK UP BYTE READ FROM RAM
1000 030464 010302          MOV      R3,R2          ;SET UP EXPECTED
1001 030466 020102          CMP      R1,R2          ;IS RAM DATA CORRECT
1002 030470 001406          BEQ      110$         ;BR, IF OK (EQUAL)
1003 030472 005237 002212      INC      FATFLG         ;BUMP COUNT
1007 030476          ERRHRD  ERRNO,T23RNC,EXPREC ;RNC=RAM NOT CORRECT
      030476 104456          TRAP      C$ERHRD
      030500 000470          .WORD    312
      030502 033165          .WORD    T23RNC
      030504 015564          .WORD    EXPREC
1008 030506          110$: CKLOOP          ;LOOP IF SELECTED
      030506 104406          TRAP      C$CLP1
1009 030510 005237 032624      INC      T23S2          ;BUMP RAM ADDRESS TO BE CHECKED
1010 030514 005237 032624      INC      T23S2          ;BUMP RAM ADDRESS TO BE CHECKED
1011 030520 010301          MOV      R3,R1          ;GET SIZE OF RECORD
1012 030522 062701 000400      ADD      #256.,R1        ;FIGURE OUT END RECORD ADDRESS
1013 030526 023701 032624      CMP      T23S2,R1        ;AT END OF RAM CHECK YET

```







TEST 3: BASIC WRITE DATA

```

1095
1096
1097
1098
1099
1100 030756 012737 150005 032610      MOV      #150005,T23PK3      ;WRITE DATA,CVC=1,ACK,SWB COMMAND
1101 030764 012737 150005 032632      MOV      #150005,T23WRT    ;SETUP FOR RETRY COMMAND
1102 030772 052737 004000 032632      BIS      #4000,T23WRT     ;MAKE IT A RETRY
1103 031000 012704 032610      MOV      #T23PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
1104 031004 010300      MOV      R3,R0           ;SET PATTERN IN CORRECT REGISTER
1105 031006 004737 017512      JSR      PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
1106 031012 010337 032616      MOV      R3,T23S2       ;SET UP RECORD SIZE IN PACKET
1107 031016 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
1108 031022 004737 016340      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
1109 031026 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
1110 031032 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED
1111 031036 020102      CMP      R1,R2         ;ARE THEY EQUAL
1112 031040 001402      BEQ      80$           ;BR, IF OK
1113 031042 004737 034162      JSR      PC,T23CHK      ;CHECK SPECIAL CONDITION
1114 031046      80$:      CKLOOP                ;LOOP IF SELECTED
1115 031050 016501 000000      MOV      TSBA(R5),R1    ;GET TSBA CONTENTS          TRAP      C$CLP1
1116 031054 012702 032512      MOV      #T23BFR,R2    ;SET UP EXPECTED
1117 031060 062702 000016      ADD      #16,R2        ;SET TO END OF MESSAGE BUFFER
1118 031064 005737 002216      TST      EXTFEA        ;CHECK FOR EXTENDED FEATURES SW SET
1119 031070 001402      BEQ      85$           ;BR, IF IT NOT SET
1120 031072 062702 000002      ADD      #2,R2         ;BUMP R2 FOR EXTRA DATA
1121 031076 020102      85$:      CMP      R1,R2        ;ARE THEY EQUAL
1122 031100 001406      BEQ      90$           ;BR, IF TSBA IS CORRECT
1123 031102 005237 002212      INC      FATFLG        ;BUMP COUNT
1127 031106      ERRHRD  ERRNO,T23BA,EXPREC ;TSBA WAS NOT CORRECT AFTER WRITE DATA
1128 031106 104456      TRAP      C$ERHRD
1128 031110 000474      .WORD     316
1128 031112 033625      .WORD     T23BA
1128 031114 015564      .WORD     EXPREC
1128 031116      90$:      CKLOOP                ;LOOP IF SELECTED
1129 031120 020327 007376      CMP      R3,#7376      ;ONLY CHECK RAM UNTIL ITS FULL
1130 031124 002115      BGE      115$         ;IT WRAPS AROUND ETC.
1131 031126 004737 034074      JSR      PC,T23RT2     ;MAKE SURE PACKET AND DATA ARE CLEAN
1132 031132 012737 000400 032624      MOV      #256.,T23S2   ;STARTING RAM ADDRESS
1133 031140 112737 000000 032622      MOV      #0,T23BS0    ;STOP INTERNAL TSV05 DIAGNOSTICS
1134 031146 112737 000000 032623      MOV      #0,T23BS1    ;SIZE OF RAM READ
1135 031154 012704 032600      MOV      #T23PK2,R4    ;SET R4 WITH PACKET ADDRESS
1136 031160 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE WRITE SUB SYS MEM COMMAND
1137 031164 004737 016426      JSR      PC,CHKTSSR    ;CHECK TSSR AND WAIT FOR SSR TO SET
1138 031170 103407      BCS      92$           ;BR, IF NO ERRORS IN TSSR
1139 031172 010001      MOV      R0,R1         ;SAVE TSSR
1140 031174 005237 002212      INC      FATFLG        ;BUMP COUNT
1144 031200      ERRHRD  ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
1144 031200 104456      TRAP      C$ERHRD
1144 031202 000475      .WORD     317
1144 031204 033677      .WORD     T23WSS
1144 031206 012136      .WORD     PKTSSR
1145 031210      92$:      CKLOOP                ;LOOP IF SELECTED
1146 031210 104406      TRAP      C$CLP1
1146 031212 004737 034074      JSR      PC,T23RT2     ;MAKE SURE PACKET AND DATA ARE CLEAN

```

## TEST 3: BASIC WRITE DATA

1147	031216	012737	000400	032624	MOV	#256.,T23S2	;STARTING RAM ADDRESS		
1148	031224	112737	000001	032622	MOVB	#1,T23BS0	;READ RAM COMMAND FOR WRITE SUB SYS M.		
1149	031232	112737	000002	032623	MOVB	#2,T23BS1	;SIZE OF RAM READ		
1150	031240	012704	032600		MOV	#T23PK2,R4	;SET R4 WITH PACKET ADDRESS		
1151	031244	010465	000000		MOV	R4,TSDB(R5)	;ISSUE WRITE SUB SYS MEM CMD (READ RAM)		
1152	031250	004737	016426		JSR	PC,CHKTSSR	;CHECK TSSR AND WAIT FOR SSR TO SET		
1153	031254	103407			BCS	100#	;BR, IF NO ERRORS IN TSSR		
1154	031256	010001			MOV	R0,R1	;SAVE TSSR		
1155	031260	005237	002212		INC	FATFLG	;BUMP COUNT		
1159	031264				ERRHRD	ERRNO,T23WSS,PKTSSR	;TSSR BAD AFTER WRITE SUB SYS MEM		
	031264	104456						TRAP	C\$ERHRD
	031266	000476						.WORD	318
	031270	033677						.WORD	T23WSS
	031272	012136						.WORD	PKTSSR
1160	031274				100#:	CKLOOP	;LOOP IF SELECTED		
	031274	104406						TRAP	C\$CLP1
1161	031276	005001			CLR	R1	;CLEAR REGISTERS		
1162	031300	005002			CLR	R2	;CLEAR REGISTERS		
1163	031302	013701	032532		MOV	T23BFR+20,R1	;PICK UP BYTE READ FROM RAM		
1164	031306	010302			MOV	R3,R2	;SET UP EXPECTED		
1165	031310	000302			SWAB	R2	;SWAP BYTES		
1166	031312	020102			CMP	R1,R2	;IS RAM DATA CORRECT		
1167	031314	001406			BEQ	110#	;BR, IF OK (EQUAL)		
1168	031316	005237	002212		INC	FATFLG	;BUMP COUNT		
1172	031322				ERRHRD	ERRNO,T23RNC,EXPREC	;RNC=RAM NOT CORRECT		
	031322	104456						TRAP	C\$ERHRD
	031324	000477						.WORD	319
	031326	033165						.WORD	T23RNC
	031330	015564						.WORD	EXPREC
1173	031332				110#:	CKLOOP	;LOOP IF SELECTED		
	031332	104406						TRAP	C\$CLP1
1174	031334	005237	032624		INC	T23S2	;BUMP RAM ADDRESS TO BE CHECKED		
1175	031340	005237	032624		INC	T23S2	;BUMP RAM ADDRESS TO BE CHECKED		
1176	031344	010301			MOV	R3,R1	;GET SIZE OF RECORD		
1177	031346	062701	000400		ADD	#256.,R1	;FIGURE OUT END RECORD ADDRESS		
1178	031352	023701	032624		CMP	T23S2,R1	;AT END OF RAM CHECK YET		
1179	031356	001332			BNE	95#	;BR, IF MORE TO CHECK		
1180	031360	062703	001750		ADD	#1000.,R3	;NEXT RECORD SIZE/DATA PATTERN		
1181	031364	020337	032620		CMP	R3,T23RSZ	;IS R3 OVER MAX RECORD SIZE		
1182	031370	002005			BGE	120#	;IF RECORD SIZE IS TOO BIG QUIT		
1183	031372	020327	177776		CMP	R3,#65534.	;END OF SUBTEST MAX RECORD SIZE		
1184	031376	001402			BEQ	120#	;BR, IF COMPLETED		
1185	031400	000137	030750		JMP	65#	;DO MORE RECORDS		
1186	031404				120#:				
1187	031404	004737	034074		JSR	PC,T23RT2	;CLEAN UP PACKET		
1188	031410	012737	102010	032600	MOV	#102010,T23PK2	;REWIND (POSITION) COMMAND		
1189	031416	012704	032600		MOV	#T23PK2,R4	;LOAD R4 WITH PACKET ADDRESS		
1190	031422	010465	000000		MOV	R4,TSDB(R5)	;ISSUE REWIND COMMAND		
1191	031426	004737	016426		JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET		
1192	031432	103407			BCS	130#	;BR, IF TSSR IS OK (GOOD)		
1193	031434	010001			MOV	R0,R1	;SAVE TSSR CONTENTS		
1194	031436	005237	002212		INC	FATFLG	;BUMP COUNT		
1198	031442				ERRHRD	ERRNO,T23RWN,PKTSSR	;TSSR IS INCORRECT AFTER REWIND		
	031442	104456						TRAP	C\$ERHRD
	031444	000500						.WORD	320
	031446	033116						.WORD	T23RWN
	031450	012136						.WORD	PKTSSR





TEST 3: BASIC WRITE DATA

```

1259                                     ;*****
1260                                     ;*****
1261 031570 012737 104405 032610 23$:  MOV   #104405,T23PK3           ;WRITE DATA, ACK, ILLEGAL BITS
1262 031576 013737 003114 032612      MOV   FREE,T23WB           ;SET UP WRITE BUFFER ADDRESS
1263 031604 062737 000001 032612      ADD   #1,T23WB           ;MAKE ADDRESS ODD (ILLEGAL)
1264 031612 012737 000400 032616      MOV   #256.,T23SZ       ;SET UP BUFFER SIZE
1265 031620 012704 032610              MOV   #T23PK3,R4        ;R4 = POINTER TO PACKET
1266 031624 010465 000000              MOV   R4,TSDB(R5)       ;ISSUE COMMAND
1267 031630 004737 016340              JSR   PC,WAITF          ;WAIT FOR SSR TO SET
1268 031634 016501 000002              MOV   TSSR(R5),R1      ;GET TSSR CONTENTS
1269 031640 012702 100206              MOV   #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
1270 031644 020102                    CMP   R1,R2             ;ARE THEY EQUAL
1271 031646 001406                    BEQ   80$               ;BR, IF OK ESP. FUNCTION REJECT
1272 031650 005237 002212              INC   FATFLG            ;BUMP COUNT
1276 031654                                ERRHRD ERRNO,T23TM,PKTSSR ;TSSR INCORRECT AFTER WRITE COMMAND
                                TRAP   C$ERHRD
                                .WORD  323
                                .WORD  T23TM
                                .WORD  PKTSSR
                                TRAP   C$CLP1
1277 031664 104406 80$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP   C$CLP1
1278 031666                                ENDSUB                ;>>>>>>>>>> END SUBTEST >>>>>>>>>>
                                L10047:
                                TRAP   C$ESUB
                                .WORD  10047
1279 031670 023727 002212 000017      CMP   FATFLG,#15.     ;IS ERROR COUNT AT 25
1280 031676 103402                    BLO   999$            ;BR, IF LESS THAN 25
1281 031700 004737 017272              JSR   PC,CKDROP        ;TRY TO DROP THE UNIT
1282 031704                                999$:

```











TEST 3: BASIC WRITE DATA

1480	032450	004737	016546
1481	032454	103002	
1482	032456	000137	027426
1483	032462		
1484	032462		
	032462	104432	
	032464	001664	

163\$:

JSR	PC,TSTLOOP
BCC	163\$
JMP	T23LOOP
EXIT	TST

```

;DO WE NEED TO ITERATE TEST
;BR, IF NO LOOP REQUIRED
;EXECUTE AGAIN

```

;ALL DONE THIS TEST

TRAP	C\$EXIT
.WORD	L10043-

## TEST 3: BASIC WRITE DATA

1486			;
1487			;LOCAL STORAGE FOR THIS TEST
1488			;
1490		032470	; ;=<.+10>E177770
1492	032470		T23PACKET:
1493	032470	100004	.WORD 100004 ;COMMAND PACKET FOR TEST
1494	032472	032500	.WORD T23DATA ;WRITE CHARACTERISTICS COMMAND, WITH ACK
1495	032474	000000	.WORD 0 ;ADDRESS OF CHARACTERISTICS BLOCK
1496	032476	000010	.WORD 8. ;STARTING VALUE OF BLOCK SIZE
1497	032500		T23DATA:
1498	032500	032512	.WORD T23BFR ;CHARACTERISTICS DATA BLOCK
1499	032502	000000	.WORD 0 ;ADDRESS OF MESSAGE BUFFER
1500	032504	000012	.WORD 10. ;LENGTH OF MESSAGE BUFFER
1501	032506	000000	.WORD 0
1502	032510	000000	T23DSW: .WORD 0 ;SELECT DRIVE 0
1503	032512		T23BFR: .BLKW 25. ;MESSAGE BUFFER
1504			;
1505			;WRITE SUBSYSTEM MEMORY COMMAND PACKET
1506			;
1508		032600	; ;=<.+10>E177770
1510	032600		T23PK2:
1511	032600	100006	.WORD 100006 ;WRITE SUB SYS MEM COMMAND, AND ACK
1512	032602	032622	.WORD T23BF2 ;ADDRESS OF SELECT BLOCK DATA
1513	032604	000000	.WORD 0
1514	032606	000006	.WORD 6. ;SIZE OF DATA PACKET
1515			
1519	032610		T23PK3:
1520	032610	100005	.WORD 100005 ;WRITE COMMAND, AND ACK
1521	032612	000000	T23WB: .WORD 0 ;ADDRESS OF WRITE BUFFER
1522	032614	000000	.WORD 0
1523	032616	000000	T23SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
1524			.EVEN
1525			;
1526	032620	000000	T23RSZ: .WORD 0 ;LARGEST TAPE RECORD IN BYTES
1527			;
1528			;
1529	032622		T23BF2:
1530	032622	010	T23BS0: .BYTE 10 ;BSELO AREA
1531	032623	200	T23BS1: .BYTE 200 ;BSEL1 AREA
1532	032624	000000	T23S2: .WORD 0 ;SEL 2 AREA
1533	032626	000000	T23S3: .WORD 0 ;DATA AREA
1534			;
1535			;
1536	032630	000000	T23TMP: .WORD 0 ;TEMPORARY REGISTER
1537	032632	000000	T23WRT: .WORD 0 ;RETRY COMMAND
1538			;
1539			.EVEN
1540			;TAPE MOTION PACKET COMMAND VALUES
1541			
1542	032634	100005	T23WD: .WORD 100005 ;WRITE DATA (NEXT)
1543	032636	100405	T23WDR: .WORD 100405 ;WRITE DATA RETRY
1544	032640	102005	T23CON: .WORD 102005 ;WRITE CONTINOUS
1545	032642	177777	.WORD 177777 ;END OF DATA
1546			
1547			

TEST 3: BASIC WRITE DATA

```

1549
1550
1551          ;+
1552          ;LOCAL TEXT MESSAGES FOR TEST
1553          ;-
1554 032644      127      122      111  T23SSR: .ASCIZ 'WRITE Command Not Accepted'
1555 032677      105      117      124  T23ET:  .ASCIZ 'EOT Not Found In 12000 4k Writes. (Use Shorter Tape)'
1556 032764      127      122      111  T23EOT: .ASCIZ 'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
1557 033042      124      123      123  T23TM:  .ASCIZ 'TSSR Not Correct After WRITE Command Reject'
1558 033116      122      145      167  T23RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
1559 033165      122      101      115  T23RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
1560 033240      124      123      123  T23AM3: .ASCIZ 'TSSR Init. Failed After WRITE Command'
1561 033306      104      162      151  T23OFI: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
1562 033361      124      123      123  T23WDD: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, SWB Bit Set'
1563 033450      124      123      123  T23WDC: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
1564 033552      103      126      103  T23VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
1565 033625      124      123      102  T23BA:  .ASCIZ 'TSBA Not Correct After WRITE DATA Command'
1566 033677      127      122      111  T23WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
1567 033766      102      141      163  T23ID:  .ASCIZ 'Basic Write'
1568          .EVEN
1569          ;+
1570          ;
1571          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1572          ;WRITE SUBSYSTEM MEMORY COMMAND
1573          ;
1574          ;-
1575
1576 034002
1577 034002
1578 034006      012701    032470
1579 034012      012721    100004
1580 034016      012721    032500
1581 034022      005021
1582 034024      012721    000012
1583 034030      012721    032512
1584 034034      005021
1585 034036      012721    000024
1586 034042      005021
1587 034044      012711    000000
1588 034050      012702    000030
1589 034054      012762    177777    032512    64$:
1590 034062      005712
1591 034064      020227    000000
1592 034070      001371
1593 034072      000207
1594
1595
1596 034074
1597 034074
1598 034100      012701    032600
1599 034104      012721    100006
1600 034110      012721    032622
1601 034114      005021
1602 034116      012721    000006
1603 034122      012701    032622
1604 034126      005021
1605 034130      005021

          T23REST:
          SAVREG
          MOV     #T23PACKET,R1      ;SAVE THE REGISTERS
          MOV     #100004,(R1)+     ;START OF THE PACKET
          MOV     #T23DATA,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK
          CLR     (R1)+             ;ADDRESS OF CHARAISTICS DATA BLOCK
          MOV     #10.,(R1)+       ;EXTENDED ADDRESS
          MOV     #T23BFR,(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,T23BFR(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

          T23RT2:
          SAVREG
          MOV     #T23PK2,R1      ;SAVE THE REGISTERS
          MOV     #100006,(R1)+   ;START OF THE PACKET
          MOV     #T23BF2,(R1)+  ;WRITE SUBSYSTEM MEM. WITH ACK
          CLR     (R1)+           ;ADDRESS OF DATA BLOCK
          MOV     #6.,(R1)+       ;EXTENDED ADDRESS
          MOV     #T23BF2,R1     ;SIZE OF DATA BLOCK IN BYTES
          CLR     (R1)+           ;POINT TO DATA SEL AREA
          CLR     (R1)+
    
```

TEST 3: BASIC WRITE DATA

```

1606 034132 005011          CLR      (R1)
1607 034134 000207          RTS      PC          ;RETURN
1608 034136                T23RT3:
1609 034136                SAVREG
1610 034142 012701 032610    MOV      #T23PK3,R1    ;SAVE THE REGISTERS
1611 034146 012721 100005    MOV      #100005,(R1) ;START OF THE PACKET
1612 034152 005021          CLR      (R1)         ;WRITE TAPE, WITH ACK
1613 034154 005021          CLR      (R1)         ;ADDRESS OF DATA BLOCK
1614 034156 005011          CLR      (R1)         ;EXTENDED ADDRESS
1615 034160 000207          RTS      PC          ;SIZE OF DATA BLOCK
1616                                ;RETURN
1617                                ;
1618                                ;ROUTINE TO RETRY WRITE DATA IN CASE OF BAD TAPE FOR TEST
1619                                ;3,SUBTEST 2 & 3
1620                                ;
1621                                ;INPUTS:          R1-TSSR
1622                                ;          SUBROUTINE SETS UP T23WRT FOR RETRY
1623                                ;
1624                                ;
1625 034162                T23CHK:
1626 034162                SAVREG
1627 034166 005037 032630    CLR      T23TMP        ;SAVE THE REGISTERS
1628 034172 032701 100000    BIT      #SC,R1        ;CLEAR LOCAL REGISTER
1629 034176 001452          BEQ      FATAL        ;IS SC SET IN TSSR?
1630 034200 013702 032522    MOV      T23BFR+10,R2 ;NO, YOU GOT PROBLEMS!
1631 034204 032702 000002    BIT      #X1,UNC,R2   ;YES,GET XSTAT1
1632 034210 001401          BEQ      1#           ;IS UNC SET IN XSTAT1?
1633 034212 000405          BR      RETRY        ;NO, CHECK COR
1634 034214 032702 020000    1# :    BIT      #X1,COR,R2 ;YES,DO WRITE DATA RETRY
1635 034220 001002          BNE      RETRY        ;IS COR SET IN XSTAT1 THEN?
1636 034222 000440          BR      FATAL        ;YES SO RETRY
1637 034224 000207          EXIT:  RTS      PC  ;NO, YOU GOT PROBLEMS
1638                                ;RETURN
1639 034226                RETRY:
1640 034226 012703 000024    2# :    MOV      #20.,R3    ;STARTING RECORD SIZE
1641 034232 013737 003114 032612  MOV      FREE,T23WB    ;STARTING WRITE BUFFER ADDRESS
1642 034240 012737 032632 032610  MOV      #T23WRT,T23PK3 ;WRITE DATA RETRY COMMAND SETUP BY SUBROUTINE
1643 034246 012704 032610          MOV      #T23PK3,R4   ;WRITE DATA RETRY COMMAND SETUP BY SUBROUTINE
1644 034252 010300          MOV      R3,R0        ;SET UP R4 WITH PACKET ADDRESS
1645 034254 004737 017512    JSR      PC,FILLMEM    ;SET PATTERN IN CORRECT REGISTER
1646 034260 010337 032616    MOV      R3,T23SZ      ;FILL MEMORY WITH RECORD SIZE
1647 034264 010465 000000    MOV      R4,T23DB(R5) ;SET UP RECORD SIZE IN PACKET
1648 034270 004737 016340    JSR      PC,WAITF      ;ISSUE COMMAND
1649 034274 016501 000002    MOV      TSSR(R5),R1  ;WAIT FOR SSR TO SET
1650 034300 012702 000200    MOV      #SSR,R2      ;GET TSSR CONTENTS
1651 034304 020102          CMP      R1,R2        ;SET UP EXPECTED
1652 034306 001746          BEQ      EXIT        ;ARE THEY EQUAL
1653 034310 005237 032630          INC      T23TMP       ;BR, IF OK
1654 034314 022737 000005 032630  CMP      #5,T23TMP    ;TRY FIVE TIMES THEN EXIT
1655 034322 001341          BNE      2#           ;DONE FIVE YET?
1656 034324 005237 002212    FATAL:  INC      FATFLG    ;NO GO AGAIN
1660 034330 013702 032512    MOV      T23BFR,R2    ;BUMP COUNT
1661 034334          ERRHRD ERRNO,SCHERR,PKTMES ;LOW ORDER MSGBUF
                                ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C:ERRHRD
                                .WORD     331
                                .WORD     SCHERR
                                .WORD     PKTMES

```

011

1SV: HARDWARE TESTS 1 8

MACRO M1200 23 MAR 84 09:44 PAGE 41-2

1FQ 0132

TEST 3: BASIC WRITE DATA

1662 034344 004737 017272  
1663 034350  
034350  
034350 104401

JSR PC,CKDROP  
ENDTST

!DROP THE UNIT

L10043: TRAP CSETST





TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

1723
1724
1725
1726 034436 004737 016064 5$: JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
1727 034442 103426 BCS 20$ ;BR IF INIT WAS OK
1728 034444 DELAY 250 ;DELAY AWHILE
      034444 012727 000250 MOV #250,(PC)+
      034450 000000 .WORD 0
      034452 013727 002116 MOV L$DLY,(PC)+
      034456 000000 .WORD 0
      034460 005367 177772 DEC -6(PC)
      034464 001375 BNE .-4
      034466 005367 177756 DEC -22(PC)
      034472 001367 BNE .-20
1729 034474 005337 043766 DEC T24DLY ;BUMP DELAY COUNTER
1730 034500 001356 BNE 5$ ;BR, IF MORE DELAY REQUIRED
1731 034502 005237 002212 INC FATFLG ;BUMP COUNT
1735 034506 010001 MOV R0,R1 ;CONTENTS OF TSSR REGISTER
1736 034510 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      034510 104455 TRAP C$ERDF
      034512 000621 .WORD 401
      034514 003650 .WORD SFIERR
      034516 012124 .WORD SFIMSG
1737 034520
1738 034520 012737 000007 043640 20$: MOV #7,T24DSW ;SET DRIVE NUMBER IN PACKET
1739 034526 012704 043620 MOV #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
1740
1741
1742
1743 ;*****
1744 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
1745 ;*****
1746
1747 034532 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
1748 034536 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
1749 034540 005237 002212 INC FATFLG ;BUMP COUNT
1753 034544 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
1754 034546 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      034546 104456 TRAP C$ERHRD
      034550 000622 .WORD 402
      034552 005054 .WORD WRTMSG
      034554 012124 .WORD SFIMSG
1755 034556 005737 002216 24$: TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
1756 034562 001044 BNE 50$ ;BR IF SWITCH IS ON
1757
1758 034564 111137 000200 043751 MOVB #200,T24BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
1759 034572 112737 000010 043750 MOVB #10,T24BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
1760 034600 012704 043730 MOV #T24PK2,R4 ;WRITE SUBSYS MEM PACKET
1761 034604 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
1762 034610 004737 016426 JSR PC,CHKTSSR ;WAIT FOR SSR
1763 034614 103407 BCS 30$ ;BR, IF NO ERROR
1764 034616 010001 MOV R0,R1 ;ERROR?, SAVE TSSR
1765 034620 005237 002212 INC FATFLG ;BUMP COUNT
1769 034624 ERRHRD ERRNO,T24SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      034624 104456 TRAP C$ERHRD
      034626 000623 .WORD 403
      034630 044507 .WORD T24SSR

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

1770 034632 012136          30$:  CKLOOP                ;LOOP IF SELECTED          .WORD  PKTSSR
      034634 104406          ;                               TRAP    C$CLP1
1771 034636 012737 000007 043640  MOV    #7,T24DSW          ;SET DRIVE NUMBER IN PACKET
1772 034644 012704 043620  MOV    #T24PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
1773
1774 ;*****
1775 ;
1776 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
1777 ;
1778 ;*****
1779
1780 034650 004737 010752  JSR    PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
1781 034654 103407          BCS    50$                ;BR, IF COMMAND ISSUED OK
1782 034656 005237 002212  INC    FATFLG             ;BUMP COUNT
1786 034662 010001          MOV    R0,R1              ;SAVE CONTENTS OF TSSR
1787 034664          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      034664 104456          TRAP    C$ERHRD
      034666 000624          .WORD  404
      034670 005054          .WORD  WRTMSG
      034672 012124          .WORD  SFIMSG
1788 034674          50$:  CKLOOP                ;SCOPE LOOP
      034674 104406          TRAP    C$CLP1
1789 034676 016501 000002  MOV    TSSR(R5),R1        ;GET TSSR CONTENTS
1790 034702 032701 000100  BIT    #OFL,R1            ;CHECK FOR THE OFFLINE BIT SET
1791 034706 001006          BNE    60$                ;BR, IF OFFLINE (GOOD)
1792 034710 005237 002212  INC    FATFLG             ;BUMP COUNT
1796 034714          ERRDF  ERRNO,T24OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
      034714 104455          TRAP    C$ERDF
      034716 000625          .WORD  405
      034720 045265          .WORD  T24OFL
      034722 012124          .WORD  SFIMSG
1797 034724          60$:  CKLOOP                ;LOOP IF SELECTED
      034724 104406          TRAP    C$CLP1
1798 034726 012703 043756  MOV    #T24RN,R3          ;POINTER FOR COMMANDS
1799
1800 ;*****
1801 ;
1802 ;TAPE READ COMMAND IN PLACE
1803 ;
1804 ;*****
1805
1806 034732 011337 043740  65$:  MOV    (R3),T24PK3      ;TAPE READ COMMAND IN PLACE
1807 034736 012704 043740  MOV    #T24PK3,R4        ;R4 = POINTER TO PACKET
1808 034742 010465 000000  MOV    R4,TSD8(R5)        ;ISSUE COMMAND
1809 034746 004737 016340  JSR    PC,WAITF           ;WAIT FOR SSR TO SET
1810 034752 016501 000002  MOV    TSSR(R5),R1        ;GET TSSR CONTENTS
1811 034756 012702 100306  MOV    #SSR!SC!OFL!BIT1!BIT2,R2 ;SET UP EXPECTED
1812 034762 020102          CMP    R1,R2              ;ARE THEY EQUAL
1813 034764 001406          BEQ    80$                ;BR, IF OK ESP. FUNCTION REJECT
1814 034766 005237 002212  INC    FATFLG             ;BUMP COUNT
1918 034772          ERRHRD  ERRNO,T24TM,PKTSSR ;TSSR INCORRECT AFTER TAPE MOTION CMD
      034772 104456          TRAP    C$FRHRD
      034774 000626          .WORD  406
      034776 045023          .WORD  T24TM
      035000 012136          .WORD  PKTSSR
1819 035002          80$:  CKLOOP                ;LOOP IF SELECTED

```



TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

1831 ;*
1832 ;
1833 ;TEST 4, SUBTEST 2
1834 ;
1835 ;VERIFIES THAT READ FORWARD COMMANDS WITH SWB=0
1836 ;OPERATES PROPERLY. THE TAPE IS FIRST REWOUND AND THEN
1837 ;WRITTEN WITH A SERIES OF TEST RECORDS VARYING IN
1838 ;LENGTH AND DATA CONTENT. THE TAPE IS THEN REWOUND
1839 ;AGAIN AND THE RECORD READ SEQUENTIALLY AND RESULTS
1840 ;(STATUS, DATA, ETC.) VERIFIED. THE BYTE COUNT ON
1841 ;EACH READ FORWARD COMMAND IS SET TO THE LENGTH OF THE
1842 ;EXPECTED RECORD, SO NO EXCEPTIONAL CONDITIONS SHOULD
1843 ;OCCUR.
1844 ;
1845 ;
1846 ;
1847 ;-
1848 035034          BGNSUB                ;>>>>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
      035034          T4.2:
1849 035034 104402          TRAP          C$BSUB
      035036 004737 046434      JSR      PC,T24RT3      ;SET UP OTHER COMMAND PACKET
1850 035042 004737 046300      JSR      PC,T24REST     ;SET COMMAND PACKET
1851 035046 004737 046372      JSR      PC,T24RT2     ;SET UP OTHER COMMAND PACKET
1852 ;
1853 ;*****
1854 ;
1855 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
1856 ;
1857 ;*****
1858 ;
1859 035052 004737 016064      JSR      PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
1860 035056 103407          BCS      20$           ;BR IF INIT WAS OK
1861 035060 005237 002212      INC      FATFLG         ;BUMP COUNT
1865 035064 010001          MOV      R0,R1          ;CONTENTS OF TSSR REGISTER
1866 035066          ERROF   ERRNO,SFIERR,SFIMSG      ;FATAL ERROR TSSR WAS NOT OK
      035066 104455          TRAP          C$ERDF
      035070 000627          .WORD      407
      035072 003650          .WORD      SFIERR
      035074 012124          .WORD      SFIMSG
1867 035076          20$:
1868 035076 013737 002172 043640 MOV      UNITN,T24DSW      ;SET DRIVE NUMBER IN PACKET
1869 035104 012704 043620      MOV      @T24PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
1870 ;
1871 ;*****
1872 ;
1873 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
1874 ;
1875 ;*****
1876 ;
1877 035110 004737 010752      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
1878 035114 103407          BCS      24$           ;BR, IF COMMAND ISSUED OK
1879 035116 005237 002212      INC      FATFLG         ;BUMP COUNT
1883 035122 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
1884 035124          ERHRD   ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTIC FAILED
      035124 104456          TRAP          C$ERHRD
      035126 000630          .WORD      408
      035130 005054          .WORD      WRTMSG

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

1885 035132 012124
035134 104406
24$: CKLOOP ;LOOP IF SELECTED .WORD SFMSG
TRAP C$CLP1
1886
1887 ;*****
1888 ;
1889 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
1890 ;
1891 ;*****
1892
1893 035136 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
1894 035142 103407 BCS 30$ ;BR, IF NO PROBLEM
1895 035144 010001 MOV R0,R1 ;SAVE TSSR
1896 035146 005237 002212 INC FATFLG ;BUMP COUNT
1900 035152 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
TRAP C$ERHRD
035152 104456 .WORD 409
035154 000631 .WORD T24RWN
035156 045076 .WORD PKTSSR
035160 012136
1901 035162 30$: CKLOOP ;LOOP IF SELECTED
035162 104406 TRAP C$CLP1
1902
1903 ;*****
1904 ;
1905 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
1906 ;
1907 ;*****
1908
1909 035164 013701 043650 MOV T24BFR+6,R1 ;PICK UP XSTO
1910 035170 010102 MOV R1,R2 ;SET UP EXPECTED
1911 035172 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
1912 035176 020102 CMP R1,R2 ;DOES EXP = REC'D
1913 035200 001406 BEQ 40$ ;BR, IF EQUAL (OK)
1914 035202 005237 002212 INC FATFLG ;BUMP COUNT
1918 035206 ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
TRAP C$ERHRD
035206 104456 .WORD 410
035210 000632 .WORD T24BOT
035212 044613 .WORD EXPREC
035214 015564
1919 035216 40$: CKLOOP ;LOOP IF SELECTED
035216 104406 TRAP C$CLP1
1920 035220 012703 000400 MOV #256.,R3 ;RECORD SIZE
1921 035224 013737 003114 043742 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
1922
1923 ;*****
1924 ;
1925 ;WRITE DATA,CVC=1,ACK COMMAND
1926 ;
1927 ;*****
1928
1929 035232 012737 140005 043740 MOV #140005,T24PK3 ;WRITE DATA,CVC=1,ACK COMMAND
1930 035240 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
1931 035244
65$: MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
1932 035244 010300 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
1933 035246 004737 017512 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
1934 035252 010337 043746 MOV R4,TSDB(R5) ;ISSUE COMMAND
1935 035256 010465 000000

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

1936 035262 004737 016340      JSR    PC,WAITF          ;WAIT FOR SSR TO SET
1937 035266 016501 000002      MOV    TSSR(R5),R1      ;GET TSSR CONTENTS
1938 035272 012702 000200      MOV    #SSR,R2         ;SET UP EXPECTED
1939 035276 020102              CMP    R1,R2           ;ARE THEY EQUAL
1940 035300 001406              BEQ    75$             ;BR, IF OK
1941 035302 005237 002212      INC    FATFLG          ;BUMP COUNT
1945 035306              ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERHRD
                                .WORD  411
                                .WORD  WRTErr
                                .WORD  PKTSSR
1946 035316              75$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD  411
                                .WORD  WRTErr
                                .WORD  PKTSSR
1947 035320 005723              TST    (R3)+           ;BUMP RECORD SIZE
1948 035322 022703 000414      CMP    #268.,R3       ;END OF RECORD YET
1949 035326 001346              BNE    65$             ;BR, IF MORE RECORDS TO WRITE
1950 035330              80$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD  411
                                .WORD  WRTErr
                                .WORD  PKTSSR
1951 035332              120$:
1952
1953 ;*****
1954 ;
1955 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
1956 ;
1957 ;*****
1958
1959 035332 004737 011104      JSR    PC,REWIND       ;CALL TAPE REWIND COMMAND
1960 035336 004737 016426      JSR    PC,CHKTSSR     ;SEE HOW TSSR IS
1961 035342 103407              BCS    130$           ;BR, IF NO PROBLEM
1962 035344 010001              MOV    R0,R1          ;SAVE TSSR
1963 035346 005237 002212      INC    FATFLG          ;BUMP COUNT
1967 035352              ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD  412
                                .WORD  T24RWN
                                .WORD  PKTSSR
1968 035362              130$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD  412
                                .WORD  T24RWN
                                .WORD  PKTSSR
1969 035362 104406
1970 ;*****
1971 ;
1972 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
1973 ;
1974 ;*****
1975
1976 035364 013701 043650      MOV    T248FR+6,R1    ;PICK UP XSTO
1977 035370 010102              MOV    R1,R2          ;SET UP EXPECTED
1978 035372 052702 000002      BIS    #BIT1,R2       ;SET BOT BIT IN EXPECTED
1979 035376 020102              CMP    R1,R2          ;DOES EXP = REC'D
1980 035400 001406              BEQ    140$           ;BR, IF EQUAL (OK)
1981 035402 005237 002212      INC    FATFLG          ;BUMP COUNT
1985 035406              ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD  413
                                .WORD  T24BOT
                                .WORD  EXPREC
1986 035416              140$:  CKLOOP          ;LOOP IF SELECTED
                                .WORD  413
                                .WORD  T24BOT
                                .WORD  EXPREC

```





1.1.1

TSV7 - HARDWARE TESTS 1-8      MACRO M1200 23 MAR-84 09:44      PAGE 43-4

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 0141

2036 035610 004737 017272  
2037 035614

999\$;

JSR      PC,CKDROP

;TRY TO DROP THE UNIT

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2039
2040
2041 ; TEST 4, SUBTEST 3
2042
2043 ; VERIFIES THAT READ DATA COMMANDS WITH CVC=1 AND THE
2044 ; SWAP BYTES (SWB) BIT SET OPERATES PROPERLY, THE TEST
2045 ; SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2.
2046 ; THE RESULTS, EXCEPT FOR RAM CONTENTS, SHOULD BE THE SAME.
2047
2048
2049
2050
2051 BGNSUB ; >>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
; T4.3: TRAP C$BSUB
2052 035614 104402 JSR PC,T24RT3 ; SET UP OTHER COMMAND PACKET
2053 035616 004737 046434 JSR PC,T24REST ; SET COMMAND PACKET
2054 035622 004737 046300 JSR PC,T24RT2 ; SET UP OTHER COMMAND PACKET
2055
2056 *****
2057 ; ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
2058
2059 *****
2060
2061
2062 035632 004737 016064 JSR PC,SOFINIT ; DO INITIALIZE ON CONTROLLER
2063 035636 103407 BCS 20$ ; BR IF INIT WAS OK
2064 035640 005237 002212 INC FATFLG ; BUMP COUNT
2068 035644 010001 MOV RO,R1 ; CONTENTS OF TSSR REGISTER
2069 035646 ERRDF ERRNO,SFIERR,SFIMSG ; FATAL ERROR TSSR WAS NOT OK
; TRAP C$ERDF
; .WORD 416
; .WORD SFIERR
; .WORD SFIMSG
2070 035656
2071 035656 013737 002172 043640 20$: MOV UNITN,T24DSW ; SET DRIVE NUMBER IN PACKET
2072 035664 012704 043620 MOV #T24PACKET,R4 ; SUBROUTINE NEEDS PACKET ADDRESS
2073
2074 *****
2075 ; WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
2076
2077 *****
2078
2079
2080 035670 004737 010752 JSR PC,WRTCHR ; ISSUE WRITE CHARACTERISTICS
2081 035674 103407 BCS 24$ ; BR, IF COMMAND ISSUED OK
2082 035676 005237 002212 INC FATFLG ; BUMP COUNT
2086 035702 010001 MOV RO,R1 ; SAVE CONTENTS OF TSSR
2087 035704 ERRHRD ERRNO,WRTMSG,SFIMSG ; WRITE CHARACTERISTIC FAILED
; TRAP C$ERRHD
; .WORD 417
; .WORD WRTMSG
; .WORD SFIMSG
2088 035714 24$: CKLOOP ; LOOP IF SELECTED
; TRAP C$CLP1
2089
2090 *****

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2091 ;
2092 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2093 ;
2094 ;*****
2095 ;
2096 035716 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2097 035722 103407 BCS 30$ ;BR, IF NO PROBLEM
2098 035724 010001 MOV R0,R1 ;SAVE TSSR
2099 035726 005237 002212 INC FATFLG ;BUMP COUNT
2103 035732 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      035732 104456 TRAP C$ERHRD
      035734 000642 .WORD 418
      035736 045076 .WORD T24RWN
      035740 012136 .WORD PKTSSR
2104 035742 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      035742 104406
2105 ;
2106 ;*****
2107 ;
2108 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
2109 ;
2110 ;*****
2111 ;
2112 035744 013701 043650 MOV T24BFR+6,R1 ;PICK UP XSTO
2113 035750 010102 MOV R1,R2 ;SET UP EXPECTED
2114 035752 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
2115 035756 020102 CMP R1,R2 ;DOES EXP = REC'D
2116 035760 001406 BEQ 40$ ;BR, IF EQUAL (OK)
2117 035762 005237 002212 INC FATFLG ;BUMP COUNT
2121 035766 ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      035766 104456 TRAP C$ERHRD
      035770 000643 .WORD 419
      035772 044613 .WORD T24BOT
      035774 015564 .WORD EXPREC
2122 035776 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      035776 104406
2123 036000 012703 000400 MOV #256.,R3 ;RECORD SIZE
2124 036004 013737 003114 043742 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
2125 ;
2126 ;*****
2127 ;
2128 ;WRITE DATA,ACK,SWB,CVC=1 COMMAND
2129 ;
2130 ;*****
2131 ;
2132 036012 012737 150005 043740 MOV #150005,T24PK3 ;WRITE DATA,ACK,SWB,CVC=1 COMMAND
2133 036020 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2134 036024 65$:
2135 036024 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
2136 036026 004737 017512 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
2137 036032 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
2138 036036 010465 000000 MOV R4,T24SDB(R5) ;ISSUE COMMAND
2139 036042 004737 016340 JSR PC,WAIT ;WAIT FOR SSR TO SET
2140 036046 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
2141 036052 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
2142 036056 020102 CMP R1,R2 ;ARE THEY EQUAL
2143 036060 001406 BEQ 75$ ;BR, IF OK

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2144 036062 005237 002212          INC      FATFLG          ;BUMP COUNT
2148 036066          ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      036066 104456          TRAP      C1ERRHRD
      036070 000644          .WORD    420
      036072 005111          .WORD    WRERR
      036074 012136          .WORD    PKTSSR
2149 036076          75$:   CKLOOP          ;LOOP IF SELECTED          TRAP      C1CLP1
      036076 104406
2150 036100 005723          ZPT      (R3),          ;BUMP RECORD SIZE
2151 036102 022703 000414          CMP      #268.,R3      ;END OF RECORD YET
2152 036106 001346          BNE     65$            ;BR, IF MORE RECORDS TO WRITE
2153 036110          80$:   CKLOOP          ;LOOP IF SELECTED          TRAP      C1CLP1
      036110 104406
2154 036112          120$:
2155
2156          ;*****
2157          ;
2158          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2159          ;
2160          ;*****
2161
2162 036112 004737 011104          JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
2163 036116 103407          BCS     130$            ;BR, IF NO PROBLEM
2164 036120 010001          MOV     R0,R1            ;SAVE TSSR
2165 036122 005237 002212          INC     FATFLG          ;BUMP COUNT
2169 036126          ERRHRD  ERRNO,T24RWN,EXPREC ;REWIND NOT ACCEPTED          TRAP      C1ERRHRD
      036126 104456          .WORD    421
      036130 000645          .WORD    T24RWN
      036132 045076          .WORD    EXPREC
2170 036136          130$:   CKLOOP          ;LOOP IF SELECTED          TRAP      C1CLP1
      036136 104406
2171
2172          ;*****
2173          ;
2174          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
2175          ;
2176          ;*****
2177
2178 036140 013701 043650          MOV     T24BFR+6,R1        ;PICK UP XSTO
2179 036144 010102          MOV     R1,R2            ;SET UP EXPECTED
2180 036146 052702 000002          BIS     #BIT1,R2          ;SET BOT BIT IN EXPECTED
2181 036152 020102          CMP     R1,R2            ;DOES EXP = REC'D
2182 036154 001406          BEQ     140$            ;BR, IF EQUAL (OK)
2183 036156 005237 002212          INC     FATFLG          ;BUMP COUNT
2187 036162          ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND          TRAP      C1ERRHRD
      036162 104456          .WORD    422
      036164 000646          .WORD    T24BOT
      036166 044613          .WORD    EXPREC
2188 036172          140$:   CKLOOP          ;LOOP IF SELECTED          TRAP      C1CLP1
      036172 104406
2189 036174 012703 000400          MOV     #256.,R3          ;RECORD SIZE
2190 036200 013737 003114 043742          MOV     FREE,T24RB        ;STARTING READ BUFFER ADDRESS
2191
2192          ;*****
2193          ;

```



TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2241          ;*
2242          ;
2243          ; TEST 4, SUBTEST 4
2244          ;
2245          ; VERIFIES THAT A READ FORWARD COMMAND READING A RECORD
2246          ; LONGER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE
2247          ; STATUS ALERT TERMINATION WITH THE RECORD LENGTH LONG
2248          ; (RLL) BIT SET.
2249          ;
2250          ;
2251          ; -
2252
2253 036370          BGNSUB                      ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
          036370          T4.4:
          036370 104402          TRAP C$BSUB
2254 036372 004737 046434      JSR PC,T24RT3      ;SET UP OTHER COMMAND PACKET
2255 036376 004737 046300      JSR PC,T24REST ;SET COMMAND PACKET
2256 036402 004737 046372      JSR PC,T24RT2  ;SET UP OTHER COMMAND PACKET
2257
2258          ;*****
2259          ;
2260          ; ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
2261          ;
2262          ;*****
2263
2264 036406 004737 016064      JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
2265 036412 103407            BCS 20$      ;BR IF INIT WAS OK
2266 036414 005237 002212    INC FATFLG   ;BUMP COUNT
2270 036420 010001            MOV RO,R1    ;CONTENTS OF TSSR REGISTER
2271 036422            ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
          036422 104455          TRAP C$ERDF
          036424 000651          .WORD 425
          036426 003650          .WORD SFIERR
          036430 012124          .WORD SFIMSG
2272 036432
2273 036432 013737 002172 043640 20$: MOV UNITN,T24DSW ;SET DRIVE NUMBER IN PACKET
2274 036440 012704 043620      MOV 0T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
2275
2276          ;*****
2277          ;
2278          ; WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
2279          ;
2280          ;*****
2281
2282 036444 004737 010752      JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
2283 036450 103407            BCS 24$      ;BR, IF COMMAND ISSUED OK
2284 036452 005237 002212    INC FATFLG   ;BUMP COUNT
2288 036456 010001            MOV RO,R1    ;SAVE CONTENTS OF TSSR
2289 036460            ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
          036460 104456          TRAP C$ERRRD
          036462 000652          .WORD 426
          036464 005054          .WORD WRTMSG
          036466 012124          .WORD SFIMSG
2290 036470            24$: CKLOOP                ;LOOP IF SELECTED
          036470 104406          TRAP C$CLP1
2291
2292          ;*****

```

## TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2293
2294 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2295 ;
2296 ;*****
2297
2298 036472 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2299 036476 103407 BC 30$ ;BR, IF NO PROBLEM
2300 036500 010001 MOV R0,R1 ;SAVE TSSR
2301 036502 005237 002212 INC FATFLG ;BUMP COUNT
2305 036506 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
    036506 104456 TRAP C$ERHRD
    036510 000653 .WORD 427
    036512 045076 .WORD T24RWN
    036514 012136 .WORD PKTSSR
2306 036516 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
    036516 104406
2307
2308 ;*****
2309 ;
2310 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
2311 ;
2312 ;*****
2313
2314 036520 013701 043650 MOV T24BFR+6,R1 ;PICK UP XSTO
2315 036524 010102 MOV R1,R2 ;SET UP EXPECTED
2316 036526 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
2317 036532 020102 CMP R1,R2 ;DOES EXP = REC'D
2318 036534 001406 BEQ 40$ ;BR, IF EQUAL (OK)
2319 036536 005237 002212 INC FATFLG ;BUMP COUNT
2323 036542 ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
    036542 104456 TRAP C$ERHRD
    036544 000654 .WORD 428
    036546 044613 .WORD T24BOT
    036550 015564 .WORD EXPREC
2324 036552 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
    036552 104406
2325 036554 012703 001000 MOV #512,R3 ;RECORD SIZE
2326 036560 013737 003114 043742 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
2327
2328 ;*****
2329 ;
2330 ;WRITE DATA,ACK,CVC=1 COMMAND
2331 ;
2332 ;*****
2333
2334 036566 012737 140005 043740 MOV #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
2335 036574 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2336 036600 65$:
2337 036600 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
2338 036604 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
2339 036610 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
2340 036614 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
2341 036620 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
2342 036624 020102 CMP R1,R2 ;ARE THEY EQUAL
2343 036626 001406 BEQ 75$ ;BR, IF OK
2344 036630 005237 002212 INC FATFLG ;BUMP COUNT
2348 036634 ERRHRD ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA

```

## TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

036634 104456
036636 000655
036640 005111
036642 012136
2349 036644 75$: CKLOOP ;LOOP IF SELECTED
036644 104406
2350 036646 120$:
2351
2352 ;*****
2353 ;
2354 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2355 ;
2356 ;*****
2357
2358 036646 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2359 036652 103407 BCS 130$ ;BR, IF NO PROBLEM
2360 036654 010001 MOV R0,R1 ;SAVE TSSR
2361 036656 005237 002212 INC FATFLG ;BUMP COUNT
2365 036662 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
036662 104456
036664 000656
036666 045076
036670 012136
2366 036672 130$: CKLOOP ;LOOP IF SELECTED
036672 104406
2367
2368 ;*****
2369 ;
2370 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
2371 ;
2372 ;*****
2373
2374 036674 013701 043650 MOV T24BFR+6,R1 ;PICK UP XST0
2375 036700 010102 MOV R1,R2 ;SET UP EXPECTED
2376 036702 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
2377 036706 020102 CMP R1,R2 ;DOES EXP = REC'D
2378 036710 001406 BEQ 140$ ;BR, IF EQUAL (OK)
2379 036712 005237 002212 INC FATFLG ;BUMP COUNT
2383 036716 ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
036716 104456
036720 000657
036722 044613
036724 015564
2384 036726 140$: CKLOOP ;LOOP IF SELECTED
036726 104406
2385 036730 012703 000400 MOV #256.,R3 ;RECORD SIZE
2386 036734 013737 003114 043742 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
2387
2388 ;*****
2389 ;
2390 ;READ DATA,ACK,CVC=1 COMMAND
2391 ;
2392 ;*****
2393
2394 036742 012737 140001 043740 MOV #140001,T24PK3 ;READ DATA,ACK,CVC=1 COMMAND
2395 036750 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2396 036754 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET

```



TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2397 036760 010465 000000          MOV     R4,TSDB(R5)           ;ISSUE COMMAND
2398 036764 004737 016340          JSR     PC,WAITF            ;WAIT FOR SSR TO SET
2399 036770 016501 000002          MOV     TSSR(R5),R1        ;GET TSSR CONTENTS
2400 036774 012702 100204          MOV     *SSR!SC!BIT2,R2    ;SET UP EXPECTED
2401 037000 020102                   CMP     R1,R2              ;ARE THEY EQUAL
2402 037002 001406          BEQ     170$               ;BR, IF OK
2403 037004 005237 002212          INC     FATFLG             ;BUMP COUNT
2407 037010                   ERRHRD  ERRNO,T24TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                             TRAP    C$ERHRD
                             .WORD  432
                             .WORD  T24TRL
                             .WORD  PKTSSR
037010 104456
037012 000660
037014 046144
037016 012136
2408 037020          170$:  CKLOOP              ;LOOP IF SELECTED
037020 104406          TRAP    C$CLP1
2409
2410           ;*****
2411           ;
2412           ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
2413           ;
2414           ;*****
2415           ;
2416 037022 013701 043650          MOV     T24BFR+6,R1        ;GET MESSAGE BUFFER
2417 037026 010102          MOV     R1,R2              ;SET UP EXPECTED
2418 037030 052702 010000          BIS     *BIT12,R2         ;SET THE RLL BIT IN EXPECTED
2419 037034 020102          CMP     R1,R2              ;ARE THEY EQUAL
2420 037036 001406          BEQ     180$               ;BR, IF EQUAL (ALL IS WELL)
2421 037040 005237 002212          INC     FATFLG             ;BUMP COUNT
2425 037044                   ERRHRD  ERRNO,T24LON,EXPREC ;THE RLL BIT WAS NOT SET IN XST0
                             TRAP    C$ERHRD
                             .WORD  433
                             .WORD  T24LON
                             .WORD  EXPREC
037044 104456
037046 000661
037050 045712
037052 015564
2426 037054          180$:  ENDSUR              ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
2427 037054          TRAP    L10056:          TRAP    C$ESUB
037054 104403
2428 037056 023727 002212 000017          CMP     FATFLG,*15.        ;IS ERROR COUNT AT 25
2429 037064 103402          BLO    999$               ;BR, IF LESS THAN 25
2430 037066 004737 017272          JSR     PC,CKDROP          ;TRY TO DROP THE UNIT
2431 037072          999$:

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2433 ;
2434 ;
2435 ; TEST 4, SUBTEST 5
2436 ;
2437 ; VERIFIES THAT A READ FORWARD COMMAND READING A RECORD
2438 ; SHORTER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE
2439 ; STATUS ALERT TERMINATION WITH THE RECORD LENGTH SHORT
2440 ; (RLS) BITS SET. IT IS VERIFIED THAT THE RESIDUAL BYTE
2441 ; COUNT (RBPCR) IN THE MESSAGE BUFFER CONTAINS THE
2442 ; PROPER NONZERO VALUE (E.G. THE DIFFERENCE BETWEEN
2443 ; THE ORIGINAL BYTE COUNT AND THE ACTUAL RECORD
2444 ; LENGTH).
2445 ;
2446 ;
2447 ;
2448 ; -
2449 ;
2450 037072      BGNSUB      ;>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
      037072      104402          T4.5:
      037072 104402          TRAP      C$BSUB
2451 037074 004737 046434      JSR      PC,T24RT3      ;SET UP OTHER COMMAND PACKET
2452 037100 004737 046300      JSR      PC,T24REST     ;SET COMMAND PACKET
2453 037104 004737 046372      JSR      PC,T24RT2     ;SET UP OTHER COMMAND PACKET
2454 ;
2455 ; *****
2456 ;
2457 ; ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
2458 ;
2459 ; *****
2460 ;
2461 037110 004737 016064      JSR      PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
2462 037114 103407          BCS      20$             ;BR IF INIT WAS OK
2463 037116 005237 002212      INC      FATFLG          ;BUMP COUNT
2467 037122 010001          MOV      R0,R1           ;CONTENTS OF TSSR REGISTER
2468 037124      ERDF      ERRNO,SFIERR,SFIMSG    ;FATAL ERROR TSSR WAS NOT OK
      037124 104455          TRAP      C$ERDF
      037126 000662          .WORD    434
      037130 003650          .WORD    SFIERR
      037132 012124          .WORD    SFIMSG
2469 037134      20$:
2470 037134 013737 002172 043640 MOV      UNITN,T24DSW    ;SET DRIVE NUMBER IN PACKET
2471 037142 012704 043620 MOV      @T24PACKET,R4  ;SUBROUTINE NEEDS PACKET ADDRESS
2472 ;
2473 ; *****
2474 ;
2475 ; WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
2476 ;
2477 ; *****
2478 ;
2479 037146 004737 010752      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
2480 037152 103407          BCS      24$             ;BR, IF COMMAND ISSUED OK
2481 037154 005237 002212      INC      FATFLG          ;BUMP COUNT
2485 037160 010001          MOV      R0,R1           ;SAVE CONTENTS OF TSSR
2486 037162      ERHRD      ERRNO,WRTMSG,SFIMSG  ;WRITE CHARACTERISTIC FAILED
      037162 104456          TRAP      C$ERHRD
      037164 000663          .WORD    435
      037166 005054          .WORD    WRTMSG

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2487 037170 012124
037172 104406
2488
2489
2490
2491
2492
2493
2494
2495 037174 004737 011104
2496 037200 103407
2497 037202 010001
2498 037204 005237 002212
2502 037210
037210 104456
037212 000664
037214 045076
037216 012136
2503 037220
037220 104406
2504 037222 012703 000400
2505 037226 013737 003114 043742
2506
2507
2508
2509
2510
2511
2512
2513 037234 012737 140005 043740
2514 037242 012704 043740
2515 037246
2516 037246 010337 043746
2517 037252 010465 000000
2518 037256 004737 016340
2519 037262 016501 000002
2520 037266 012702 000200
2521 037272 020102
2522 037274 001406
2523 037276 005237 002212
2527 037302
037302 104456
037304 000665
037306 005111
037310 012136
2528 037312
037312 104406
2529 037314
2530
2531
2532
2533
2534
2535
2536
2537 037314 004737 011104

```

```

24$: CKLOOP ;LOOP IF SELECTED .WORD SFIMSG
TRAP C$CLP1
;*****
;
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
JSR PC,REWIND ;CALL TAPE REWIND COMMAND
BCS 30$ ;BR, IF NO PROBLEM
MOV R0,R1 ;SAVE TSSR
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
TRAP C$ERHRD
.WORD 436
.WORD T24RWN
.WORD PKTSSR
30$: CKLOOP ;LOOP IF SELECTED
TRAP C$CLP1
MOV #256.,R3 ;RECORD SIZE
MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
;*****
;
;WRITE DATA,ACK,CVC=1 COMMAND
;
;*****
MOV #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
65$:
MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
MOV R4,TSDB(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 75$ ;BR, IF OK
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
TRAP C$ERHRD
.WORD 437
.WORD WRERR
.WORD PKTSSR
75$: CKLOOP ;LOOP IF SELECTED
TRAP C$CLP1
120$:
;*****
;
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
JSR PC,REWIND ;CALL TAPE REWIND COMMAND

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2538 037320 103407          BCS      130$          ;BR, IF NO PROBLEM
2539 037322 010001          MOV      R0,R1         ;SAVE TSSR
2540 037324 005237 002212  INC      FATFLG        ;BUMP COUNT
2544 037330          ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      037330 104456          TRAP     C$ERHRD
      037332 000666          .WORD   438
      037334 045076          .WORD   T24RWN
      037336 012136          .WORD   PKTSSR
2545 037340          130$:  CKLOOP          ;LOOP IF SELECTED          TRAP     C$CLP1
      037340 104406
2546 037342 012703 001000  MOV      #512.,R3      ;RECORD SIZE
2547 037346 013737 003114 043742  MOV      FREE,T24RB    ;STARTING READ BUFFER ADDRESS
2548
2549
2550
2551
2552
2553
2554
2555 037354 012737 140001 043740  MOV      #140001,T24PK3 ;READ DATA,ACK,CVC=1 COMMAND
2556 037362 012704 043740 165$:  MOV      #T24PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
2557 037366 010337 043746  MOV      R3,T24SZ      ;SET UP RECORD SIZE IN PACKET
2558 037372 010465 000000  MOV      R4,T24SDB(R5) ;ISSUE COMMAND
2559 037376 004737 016340  JSR      PC,WAITF      ;WAIT FOR SSR TO SET
2560 037402 016501 000002  MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
2561 037406 012702 100204  MOV      #SSR:SC!BIT2,R2 ;SET UP EXPECTED
2562 037412 020102  CMP      R1,R2         ;ARE THEY EQUAL
2563 037414 001406  BEQ      170$          ;BR, IF OK
2564 037416 005237 002212  INC      FATFLG        ;BUMP COUNT
2568 037422          ERRHRD  ERRNO,T24TRL,EXPREC ;TSSR INCORRECT AFTER READ DATA
      037422 104456          TRAP     C$ERHRD
      037424 000667          .WORD   439
      037426 046144          .WORD   T24TRL
      037430 015564          .WORD   EXPREC
2569 037432          170$:  CKLOOP          ;LOOP IF SELECTED          TRAP     C$CLP1
      037432 104406
2570
2571
2572
2573
2574
2575
2576
2577 037434 013701 043650          MOV      T24BFR+6,R1   ;GET MESSAGE BUFFER
2578 037440 010102          MOV      R1,R2         ;SET UP EXPECTED
2579 037442 052702 040000  BIS      #BIT14,R2     ;SET THE RLS BIT IN EXPECTED
2580 037446 020102  CMP      R1,R2         ;ARE THEY EQUAL
2581 037450 001406  BEQ      180$          ;BR, IF EQUAL (ALL IS WELL)
2582 037452 005237 002212  INC      FATFLG        ;BUMP COUNT
2586 037456          ERRHRD  ERRNO,T24LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
      037456 104456          TRAP     C$ERHRD
      037460 000670          .WORD   440
      037462 045774          .WORD   T24LOP
      037464 015564          .WORD   EXPREC
2587 037466          180$:
2588 037466 013701 043646  MOV      T24BFR+4,R1   ;PICK UP RESIDUAL BYTE COUNTER
2589 037472 012702 000400  MOV      #256.,R2     ;THIS SHOULD BE THE DIFFERENCE

```





TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

037634 012124
2659 037636 24$: CKLOOP ;LOOP IF SELECTED .WORD SFIMSG
037636 104406 TRAP C$CLP1
2660
2661 ;*****
2662 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2663 ;
2664 ;*****
2665
2666
2667 037640 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2668 037644 10 407 BCS 30$ ;BR, IF NO PROBLEM
2669 037646 010001 MOV R0,R1 ;SAVE TSSR
2670 037650 005237 002212 INC FATFLG ;BUMP COUNT
2674 037654 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
037654 104456 TRAP C$ERHRD
037656 006674 .WORD 444
037660 045076 .WORD T24RWN
037662 012136 .WORD PKTSSR
2675 037664 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
037664 104406
2676 037666 012703 000400 MOV #256.,R3 ;RECORD SIZE
2677 037672 013737 003114 043742 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
2678
2679 ;*****
2680 ;WRITE DATA,ACK,CVC=1 COMMAND
2681 ;
2682 ;*****
2683
2684
2685 037700 012737 140005 043740 MOV #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
2686 037706 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2687 037712
2688 037712 010300 65$: MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
2689 037714 004737 017512 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
2690 037720 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
2691 037724 010465 060000 MOV R4,TSDR(R5) ;ISSUE COMMAND
2692 037730 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
2693 037734 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
2694 037740 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
2695 037744 020102 CMP R1,R2 ;ARE THEY EQUAL
2696 037746 001406 BEQ 75$ ;BR, IF OK
2697 037750 005237 002212 INC FATFLG ;BUMP COUNT
2701 037754 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
037754 104456 TRAP C$ERHRD
037756 000675 .WORD 445
037760 005111 .WORD WRERR
037762 012136 .WORD PKTSSR
2702 037764 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
037764 104406
2703 037766 005723 TST (R3)+ ;BUMP RECORD SIZE
2704 037770 022703 000414 CMP #268.,R3 ;END OF RECORD YET
2705 037774 001346 BNE 65$ ;BR, IF MORE RECORDS TO WRITE
2706 037776 80$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
037776 104406
2707 040000 005743 TST -(R3) ;SET BACK TO 512
2708 040002 013737 003114 043742 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2709
2710 ;*****
2711 ;
2712 ;READ REVERSE DATA,ACK COMMAND
2713 ;
2714 ;*****
2715
2716 040010 012737 100401 043740          MOV      #100401,T24PK3      ;READ REVERSE DATA,ACK COMMAND
2717 040016 012704 043740          165$:   MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
2718 040022 010337 043746          MOV      R3,T24SZ          ;SET UP RECORD SIZE IN PACKET
2719 040026 010465 000000          MOV      R4,TSDB(R5)      ;ISSUE COMMAND
2720 040032 004737 016340          JSR      PC,WAITF         ;WAIT FOR SSR TO SET
2721 040036 016501 000002          MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
2722 040042 012702 000200          MOV      #SSR,R2         ;SET UP EXPECTED
2723 040046 020102                  CMP      R1,R2            ;ARE THEY EQUAL
2724 040050 001406                  BEQ      170$             ;BR, IF OK
2725 040052 005237 002212          INC      FATFLG           ;BUMP COUNT
2729 040056          ERRHRD  ERRNO,T24WDC,PKTSSR ;TSSR INCORRECT AFTER READ DATA
           040056 104456                  TRAP     C$ERRHRD
           040060 000676                  .WORD   446
           040062 045426                  .WORD   T24WDC
           040064 012136                  .WORD   PKTSSR
2730 040066          170$:   CKLOOP                    ;LOOP IF SELECTED
           040066 104406                  TRAP     C$CLP1
2731 040070 013702 003114          MOV      FREE,R2          ;GET BUFFER ADDRESS
2732 040074 010304                  MOV      R3,R4            ;CURRENT RECORD SIZE
2733 040076 162704 000400          SUB      #256,,R4        ;FIRST LOCATION IN BUFFER
2734 040102 060204          173$:   ADD      R2,R4            ;SET POINTER TO FRAME (WORD)
2735 040104 021403                  CMP      (R4),R3          ;CHECK DATA READ (R3 DATA ALSO)
2736 040106 001410                  BEQ      180$             ;BR, IF ALL IS WELL
2737 040110 011401                  MOV      (R4),R1         ;RECD DATA
2738 040112 010302                  MOV      R3,R2            ;EXPECTED DATA
2739 040114 005237 002212          INC      FATFLG           ;BUMP COUNT
2743 040120          ERRHRD  ERRNO,T24DTA,EXPREC ;DATA READ NOT * WRITTEN
           040120 104456                  TRAP     C$ERRHRD
           040122 000677                  .WORD   447
           040124 044660                  .WORD   T24DTA
           040126 015564                  .WORD   EXPREC
2744 040130          180$:   CKLOOP                    ;LOOP IF SELECTED
           040130 104406                  TRAP     C$CLP1
2745 040132 005724                  TST      (R4),            ;BUMP TO NEXT LOCATION
2746 040134 160204                  SUB      R2,R4            ;GET RID OF BASE ADDRESS
2747 040136 020403                  CMP      R4,R3            ;END OF RECORD YET
2748 040140 001360                  BNE      173$             ;BR, IF NOT AT END OF RECORD
2749 040142 005743                  TST      -(R3)            ;BUMP RECORD SIZE
2750 040144 022703 000400          CMP      #256,,R3        ;END OF RECORD YET
2751 040150 001322                  BNE      165$             ;BR, IF MORE RECORDS TO WRITE
2752 040152          190$:   CKLOOP                    ;LOOP IF SELECTED
           040152 104406                  TRAP     C$CLP1
2753 040154          ENDSUB                      ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
           040154 104403                  L10060:
           040156 023727 002212 000017        CMP      FATFLG,#15.      ;IS ERROR COUNT AT 25
           040164 103402                  BLO      999$             ;BR, IF LESS THAN 25
           040166 004737 017272          JSR      PC,CKDROP        ;TRY TO DROP THE UNIT
2757 040172          999$:

```



1115

SEQ 0157

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772 040172
040172
040172 104402
2773 040174 004737 046434
2774 040200 004737 046300
2775 040204 004737 046372
2776
2777
2778
2779
2780
2781
2782
2783 040210 004737 016064
2784 040214 103407
2785 040216 005237 002212
2789 040222 010001
2790 040224
040224 104455
040226 000700
040230 003650
040232 012124
2791 040234
2792 040234 013737 002172 043640
2793 040242 012704 043620
2794
2795
2796
2797
2798
2799
2800
2801 040246 004737 010752
2802 040252 103407
2803 040254 005237 002212
2807 040260 010001
2808 040262
040262 104456
040264 000701
040266 005054
040270 012124
2809 040272
040272 104406
2810
2811

```

;
; TEST 4, SUBTEST 7
;
; VERIFIES THAT READ DATA COMMANDS WITH CVC-1 AND THE
; SWAP BYTES (SWB) BIT SET OPERATES PROPERLY. THE TEST
; SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2.
; THE RESULTS, EXCEPT FOR RAM CONTENTS, SHOULD BE THE SAME.
;
;
;-----
;
;-----
; ***** BEGIN SUBTEST *****
; T4.7:
; TRAP C18SUB
; SET UP OTHER COMMAND PACKET
; SET COMMAND PACKET
; SET UP OTHER COMMAND PACKET
;
;-----
; *****
; ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
;
;-----
;
; JSR PC,SOFINIT ; DO INITIALIZE ON CONTROLLER
; BCS 201 ; BR IF INIT WAS OK
; INC FATFLG ; BUMP COUNT
; MOV RO,R1 ; CONTENTS OF TSSR REGISTER
; ERRDF ERRNO,SFIERR,SFIMSG ; FATAL ERROR TSSR WAS NOT OK
; TRAP C1ERDF
; .WORD 448
; .WORD SFIERR
; .WORD SFIMSG
;
; 201:
; MOV UNITN,T24DSW ; SET DRIVE NUMBER IN PACKET
; MOV #T24PACKET,R4 ; SUBROUTINE NEEDS PACKET ADDRESS
;
;-----
; *****
; WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
;
;-----
;
; JSR PC,WRTCHR ; ISSUE WRITE CHARACTERISTICS
; BCS 241 ; BR, IF COMMAND ISSUED OK
; INC FATFLG ; BUMP COUNT
; MOV RO,R1 ; SAVE CONTENTS OF TSSR
; ERRHRD ERRNO,WRTMSG,SFIMSG ; WRITE CHARACTERISTIC FAILED
; TRAP C1ERHRD
; .WORD 449
; .WORD WRTMSG
; .WORD SFIMSG
;
; 241: CKLOOP ; LOOP IF SELECTED
; TRAP C1CLP1
;
;-----

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2812
2813
2814
2815
2816
2817 040274 004737 011104
2818 040300 103407
2819 040302 010001
2820 040304 005237 002212
2824 040310
      040310 104456
      040312 000702
      040314 045075
      040316 012136
2825 040320
      040320 104406
2826 040322 012703 000400
2827 040326 013737 003114 043742
2828
2829
2830
2831
2832
2833
2834
2835 040334 012737 150005 043740
2836 040342 012704 043740
2837 040346
2838 040346 010300
2839 040350 004737 017512
2840 040354 010337 043746
2841 040360 010465 000000
2842 040364 004737 016340
2843 040370 016501 000002
2844 040374 012702 000200
2845 040400 020102
2846 040402 001406
2847 040404 005237 002212
2851 040410
      040410 104456
      040412 000703
      040414 005111
      040416 012136
2852 040420
      040420 104406
2853 040422 005723
2854 040424 022703 000414
2855 040430 001346
2856 040432
      040432 104406
2857 040434 005743
2858 040436 013737 003114 043742
2859
2860
2861
2862
2863

```

```

;
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
;CALL TAPE REWIND COMMAND
;BR, IF NO PROBLEM
;SAVE TSSR
;BUMP COUNT
;REWIND NOT ACCEPTED
;ERRHRD
;WORD 450
;WORD T24RWN
;WORD PKTSSR
30$: CKLOOP ;LOOP IF SELECTED
;TRAP C:ERRHRD
;TRAP C:CLP1
MOV #256.,R3 ;RECORD SIZE
MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
;*****
;WRITE DATA,ACK,CVC=1,SWB COMMAND
;*****
MOV #150005,T24PK3 ;WRITE DATA,ACK,CVC=1,SWB COMMAND
MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
65$: MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
MOV R4,TSDR(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 75$ ;BR, IF OK
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
;TRAP C:ERRHRD
;WORD 451
;WORD WRTErr
;WORD PKTSSR
75$: CKLOOP ;LOOP IF SELECTED
;TRAP C:CLP1
TST (R3), ;BUMP RECORD SIZE
CMP #268.,R3 ;END OF RECORD YET
BNF 65$ ;BR, IF MORE RECORDS TO WRITE
80$: CKLOOP ;LOOP IF SELECTED
;TRAP C:CLP1
TST -(R3) ;SET RECORD SIZE BACK TO 512.
MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
;*****
;READ REVERSE DATA,ACK,SWB COMMAND
;

```





TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2961
2962 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2963 ;
2964 ;*****
2965
2966 040730 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2967 040734 004737 016426 JSR PC,CHKTSSR ;SEE HOW TSSR IS
2968 040740 103407 BCS 30$ ;BR, IF NO PROBLEM
2969 040742 010001 MOV RO,R1 ;SAVE TSSR
2970 040744 005237 002212 INC FATFLG ;BUMP COUNT
2974 040750 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      040750 104456 TRAP C$ERRHRD
      040752 000710 .WORD 456
      040754 045076 .WORD T24RWN
      040756 012136 .WORD PKTSSR
2975 040760 30$: CKLOOP ;LOOP IF SELECTED
      040760 104406 TRAP C$CLP1
2976 040762 012703 001000 MOV #512.,R3 ;RECORD SIZE
2977 040766 013737 003114 043742 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
2978
2979 ;*****
2980 ;WRITE DATA,ACK,CVC=1 COMMAND
2981 ;
2982 ;*****
2983
2984
2985 040774 012737 140005 043740 MOV #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
2986 041002 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2987 041006 65$:
2988 041006 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
2989 041012 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
2990 041016 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
2991 041022 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
2992 041026 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
2993 041032 020102 CMP R1,R2 ;ARE THEY EQUAL
2994 041034 001406 BEQ 75$ ;BR, IF OK
2995 041036 005237 002212 INC FATFLG ;BUMP COUNT
2999 041042 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      041042 104456 TRAP C$ERRHRD
      041044 000711 .WORD 457
      041046 005111 .WORD WRERR
      041050 012136 .WORD PKTSSR
3000 041052 75$: CKLOOP ;LOOP IF SELECTED
      041052 104406 TRAP C$CLP1
3001 041054 012703 000400 MOV #256.,R3 ;SIZE OF RECORD
3002 041060 013737 003114 043742 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
3003
3004 ;*****
3005 ;READ DATA,ACK COMMAND
3006 ;
3007 ;*****
3008
3009
3010 041066 012737 100401 043740 MOV #100401,T24PK3 ;READ DATA,ACK COMMAND
3011 041074 012704 043740 165$: MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
3012 041100 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
3013 041104 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND

```



TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065 041220
      041220
      041220 104402
3066 041222 005737 003126
3067 041226 001002
3068 041230 000137 041616
3069 041234 004737 046434
3070 041240 004737 046300
3071 041244 004737 046372
3072
3073
3074
3075
3076
3077
3078
3079 041250 004737 016064
3080 041254 103407
3081 041256 005237 002212
3085 041262 010001
3086 041264
      041264 104455
      041266 000714
      041270 003650
      041272 012124
3087 041274
3088 041274 013737 002172 043640
3089 041302 012704 043620
3090
3091
3092
3093
3094
3095
3096
3097 041306 004737 010752
3098 041312 103407
3099 041314 005237 002212
3103 041320 010001
3104 041322
      041322 104456

;
;
; TEST 4, SUBTEST 9
;
; VERIFIES THAT A READ REVERSE COMMAND SPECIFYING A DATA
; BUFFER STARTING IN NONEXISTANT MEMORY TERMINATES WITH
; THE PROPER ERROR STATUS WITHOUT MOVING TAPE
;
; *****
;                         CAUTION
;
;           The LSI BUS drivers for all available address lines(16:21)
; are only checked when running on a 11/23B system with more than
; 128K words of memory!
; *****
; -

BGNSUB                                ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
                                     T4.9:                               TRAP      C$B5UB
;
; TST      NXMFLG                      ;DO WE HAVE IT?
; BNE     10$                          ;BR, IF ENOUGH
; JMP     180$                          ;SKIP THIS TEST IF NOT
10$: JSR   PC,T24RT3                    ;SET UP OTHER COMMAND PACKET
     JSR   PC,T24REST                   ;SET COMMAND PACKET
     JSR   PC,T24RT2                    ;SET UP OTHER COMMAND PACKET
;
; *****
;
; ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
;
; *****
;
; JSR     PC,SOFINIT                    ;DO INITIALIZE ON CONTROLLER
; BCS     20$                           ;BR IF INIT WAS OK
; INC     FATFLG                        ;BUMP COUNT
; MOV     R0,R1                         ;CONTENTS OF TSSR REGISTER
; ERDF    ERRNO,SFIERR,SFIMSG          ;FATAL ERROR TSSR WAS NOT OK
                                     TRAP      C$ERDF
                                     .WORD    460
                                     .WORD    SFIERR
                                     .WORD    SFIMSG
20$: MOV   UNITN,T24DSW                 ;SET DRIVE NUMBER IN PACKET
     MOV   @T24PACKET,R4                ;SUBROUTINE NEEDS PACKET ADDRESS
;
; *****
;
; WRITE CHARACTERISTICS COMMAND (CALL TO WRCHR)
;
; *****
;
; JSR     PC,WRCHR                      ;ISSUE WRITE CHARACTERISTICS
; BCS     24$                           ;BR, IF COMMAND ISSUED OK
; INC     FATFLG                        ;BUMP COUNT
; MOV     R0,R1                         ;SAVE CONTENTS OF TSSR
; ERHRD   ERRNO,WRMSG,SFIMSG           ;WRITE CHARACTERISTIC FAILED
                                     TRAP      C$ERHRD
```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

041324 000715 .WORD 461
041326 005054 .WORD WRTMSG
041330 012124 .WORD SFIMSG
3105 041332 24$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
041332 104406
3106
3107 ;*****
3108 ;
3109 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
3110 ;
3111 ;*****
3112
3113 041334 004737 021276 JSR PC,INVERT ;INVERT THE EXTENDED FEATURES SWITCH
3114 041340 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
3115 041344 004737 016426 JSR PC,CHKTSSR ;SEE HOW TSSR IS
3116 041350 103407 BCS 30$ ;BR, IF NO PROBLEM
3117 041352 010001 MOV R0,R1 ;SAVE TSSR
3118 041354 005237 002212 INC FATFLG ;BUMP COUNT
3122 041360 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
041360 104456 TRAP C$ERHRD
041362 000716 .WORD 462
041364 045076 .WORD T24RWN
041366 012136 .WORD PKTSSR
3123 041370 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
041370 104406
3124 041372 012703 000400 MOV #256.,R3 ;RECORD SIZE
3125 041376 013737 003114 043742 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
3126 ;*****
3127 ;
3128 ;WRITE DATA,ACK,CVC=1 COMMAND
3129 ;
3130 ;*****
3131
3132 041404 012737 140005 043740 MOV #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
3133 041412 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
3134 041416
3135 041416 010337 043746 65$: MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
3136 041422 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
3137 041426 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
3138 041432 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
3139 041436 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
3140 041442 020102 CMP R1,R2 ;ARE THEY EQUAL
3141 041444 001406 BEQ 75$ ;BR, IF OK
3142 041446 005237 002212 INC FATFLG ;BUMP COUNT
3146 041452 ERRHRD ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
041452 104456 TRAP C$ERHRD
041454 000717 .WORD 463
041456 005111 .WORD WRTERR
041460 012136 .WORD PKTSSR
3147 041462 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
041462 104406
3148 041464 012703 000400 MOV #256.,R3 ;RECORD SIZE
3149 041470 013737 003130 043742 MOV NXML0,T24RB ;STARTING READ BUFFER ADDRESS
3150 041476 013737 003132 043744 MOV NXMHI,T24RB+2 ;SET ADDRESS BITS 16-17
3151
3152 ;*****
3153 ;

```





TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210 041636           BGNSUB                      ;>>>>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>>>
      041636           T4.10:
      041636 104402           TRAP                C$BSUB
3211 041640 004737 046434     JSR      PC,T24RT3         ;SET COMMAND PACKET UP CLEAR
3212 041644 004737 046300     JSR      PC,T24REST      ;SET COMMAND PACKET
3213 041650 004737 046372     JSR      PC,T24RT2      ;SET UP OTHER COMMAND PACKET
3214
3215
3216
3217
3218
3219
3220
3221 041654 004737 016064     JSR      PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
3222 041660 103407           BCS      20$              ;BR IF INIT WAS OK
3223 041662 005237 002212     INC      FATFLG          ;BUMP COUNT
3227 041666 010001           MOV      R0,R1           ;CONTENTS OF TSSR REGISTER
3228 041670           ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      041670 104455           TRAP                C$ERRDF
      041672 000722           .WORD        466
      041674 003650           .WORD        SFIERR
      041676 012124           .WORD        SFIMSG
3229 041700
3230 041700 013737 002172 043640 20$:  MOV      UNITN,T24DSW    ;SET UP DRIVE NUMBER
3231 041706 012704 043620     MOV      @T24PACKET,R4  ;SUBROUTINE NEEDS PACKET ADDRESS
3232
3233
3234
3235
3236
3237
3238
3239 041712 004737 010752     JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
3240 041716 103407           BCS      24$              ;BR, IF COMMAND ISSUED OK
3241 041720 005237 002212     INC      FATFLG          ;BUMP COUNT
3245 041724 010001           MOV      R0,R1           ;SAVE CONTENTS OF TSSR
3246 041726           ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      041726 104456           TRAP                C$ERRHRD
      041730 000723           .WORD        467
      041732 005054           .WORD        WRTMSG
      041734 012124           .WORD        SFIMSG
3247 041736           24$:  CKLOOP                ;LOOP IF SELECTED
      041736 104406           TRAP                C$CLP1
3248 041740 013737 003114 043742     MOV      FREE,T24RB     ;STARTING WRITE BUFFER ADDRESS
3249

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3250 ;*****
3251 ;
3252 ;ILLEGAL MODE DATA,ACK COMMAND
3253 ;
3254 ;*****
3255
3256 041746 012737 104001 043740      MOV     0104001,T24PK3      ;ILLEGAL MODE DATA,ACK COMMAND
3257 041754 012704 043740          MOV     01T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
3258 041760 012737 000400 043746      MOV     0256.,T24SZ        ;SET UP RECORD SIZE IN PACKET
3259 041766 010465 000000          MOV     R4,T5DB(R5)        ;ISSUE COMMAND
3260 041772 004737 016340          JSR     PC,WAITF           ;WAIT FOR SSR!BIT1!BIT2 TO SET
3261 041776 016501 000002          MOV     T5SR(R5),R1        ;GET T5SR CONTENTS
3262 042002 012702 100206          MOV     05SR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
3263 042006 020102          CMP     R1,R2              ;ARE THEY EQUAL
3264 042010 001406          BEQ     75$                ;BR, IF OK
3265 042012 005237 002212          INC     FATFLG             ;BUMP COUNT
3269 042016          ERRHRD  ERRNO,T24WDF,PKT5SR ;T5SR INCORRECT AFTER READ DATA
                                TRAP     C$ERHRD
                                .WORD    468
                                .WORD    T24WDF
                                .WORD    PKT5SR
3270 042026          75$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP     C$CLP1
3271
3272 ;*****
3273 ;
3274 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
3275 ;
3276 ;*****
3277
3278 042030 013701 043650          MOV     T24BFR+6,R1        ;GET MESSAGE BUFFER
3279 042034 010102          MOV     R1,R2              ;SET UP EXPECTED
3280 042036 052702 001000          BIS     08BIT9,R2          ;SET THE ILC BIT IN EXPECTED
3281 042042 020102          CMP     R1,R2              ;ARE THEY EQUAL
3282 042044 001406          BEQ     180$                ;BR, IF EQUAL (ALL IS WELL)
3283 042046 005237 002212          INC     FATFLG             ;BUMP COUNT
3287 042052          ERRHRD  ERRNO,T24LQ,EXPREC ;THE ILC BIT WAS NOT SET IN XSTO
                                TRAP     C$ERHRD
                                .WORD    469
                                .WORD    T24LQ
                                .WORD    EXPREC
3288 042062          180$:  CKLOOP
                                TRAP     C$CLP1
3289 042064          ENDSUB                    ;>>>>>>>>>> END SUBTEST >>>>>>>>>>
                                .L10064:
                                TRAP     C$ESUB
3290 042066 023727 002212 000017          CMP     FATFLG,015.        ;IS ERROR COUNT AT 25
3291 042074 103402          BLO     999$                ;BR, IF LESS THAN 25
3292 042076 004737 017272          JSR     PC,CKDROP          ;TRY TO DROP THE UNIT
3293 042102          999$:

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307 042102      BGNSUB                ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
      042102      104402                      14.11:
      042102      004737 046434                JSR      PC,T24RT3           ;SET COMMAND PACKET UP CLEAR   TRAP      C$B5UB
3308 042104      004737 046300                JSR      PC,T24REST        ;SET COMMAND PACKET
3309 042110      004737 046372                JSR      PC,T24RT2        ;SET UP OTHER COMMAND PACKET
3310 042114
3311
3312
3313
3314
3315
3316
3317
3318 042120      004737 016064                JSR      PC,SOFINIT        ;DO INITIALIZE ON CONTROLLER
3319 042124      103407                      BCS     20$                ;BR IF INIT WAS OK
3320 042126      005237 002212                INC     FATFLG             ;BUMP COUNT
3324 042132      010001                MOV     R0,R1              ;CONTENTS OF TSSR REGISTER
3325 042134      104455                      ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      042134      000726                      .WORD  470                TRAP      C$ERDF
      042136      003650                      .WORD  SFIERR             .WORD    470
      042140      012124                      .WORD  SFIMSG             .WORD    SFIERR
3326 042144
3327 042144      013737 002172 043640      20$:      MOV     UNITN,T24DSW        ;SET UP DRIVE NUMBER
3328 042152      012704 043620                MOV     @T24PACKET,R4     ;SUBROUTINE NEEDS PACKET ADDRESS
3329
3330
3331
3332
3333
3334
3335
3336 042156      004737 010752                JSR      PC,WRTCHR         ;ISSUE WRITE CHARACTERISTICS
3337 042162      103407                      BCS     24$                ;BR, IF COMMAND ISSUED OK
3338 042164      005237 002212                INC     FATFLG             ;BUMP COUNT
3342 042170      010001                MOV     R0,R1              ;SAVE CONTENTS OF TSSR
3343 042172      104456                      ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
      042172      000727                      .WORD  471                TRAP      C$ERHRD
      042174      005054                      .WORD  WRTMSG             .WORD    471
      042200      012124                      .WORD  SFIMSG             .WORD    WRTMSG
3344 042202      104406                      24$:      CKLOOP                ;LOOP IF SELECTED
      042202
3345 042204      013737 003114 043742                MOV     FREE,T24RB        ;ILLEGAL STARTING READ BUFFER ADDRESS
3346 042212      012737 177700 043744                MOV     @177700,T24RB+2   ;CREATE ILLEGAL ADDRESS

```



TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410 042354          HGNSUB          >>>>>>>>> BEGIN SUBTEST >>>>>>>>>
      042354          104402          T4.12: TRAP CIBSUB
      042354          005737 003126          TST         NXMLG          ;DO WE HAVE IT?
3411 042356          001002          BNE         10$          ;BR, IF ENOUGH
3412 042362          000137 042572          JMP         80$          ;SKIP THIS TEST IF NOT
3413 042364          004737 046434          10$: JSR         PC,T24RT3 ;SET COMMAND PACKET UP CLEAR
3414 042370          004737 046300          JSR         PC,T24REST ;SET COMMAND PACKET
3415 042374          004737 046372          JSR         PC,T24RT2  ;SET UP OTHER COMMAND PACKET
3416 042400
3417
3418
3419
3420
3421
3422
3423
3424 042404          004737 016064          JSR         PC,SOFINIT  ;DO INITIALIZE ON CONTROLLER
3425 042410          103407          BCS         20$          ;BR IF INIT WAS OK
3426 042412          005237 002212          INC         FATELG      ;BUMP COUNT
3430 042416          010001          MOV         R0,R1       ;CONTENTS OF TSSR REGISTER
3431 042420          044455          ERRDF       ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      042420          000732          TRAP       CIERDF
      042422          003650          .WORD     474
      042424          012124          .WORD     SFIERR
      042426          .WORD     SFIMSG
3432 042430          013737 002172 043640 20$: MOV         UNITN,T24DSW ;SET UP DRIVE NUMBER
3433 042430          012704 043620          MOV         @T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
3434 042436
3435
3436
3437
3438
3439
3440
3441
3442 042442          004737 010752          JSR         PC,WRTCHR   ;ISSUE WRITE CHARACTERISTICS
3443 042446          103407          BCS         24$          ;BR, IF COMMAND ISSUED OK
3444 042450          005237 002212          INC         FATELG      ;BUMP COUNT
3448 042454          010001          MOV         R0,R1       ;SAVE CONTENTS OF TSSR
3449 042456          ERRHRD    ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED

```

74

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

042456 104456
042460 000733
042462 005054
042464 012124
3450 042466 24$: CKLOOP                                ;LOOP IF SELECTED
042466 104406
3451 042470 004757 021276                               ;JSR PC,INVERT                    ;INVERT THE EXTENDED FEATURES SWITCH
3452 042474 013737 003130 043742                       ;MOV NXML0,T24RB                 ;SET TO NXM MEMORY ADDRESS
3453 042502 013737 003132 043744                       ;MOV NXMH1,T24RB                 ;SET HIGH BITS
3454
3455
3456
3457
3458
3459
3460
3461 042510 012737 140001 043740                       ;MOV @140001,T24PK3             ;READ, ACK, CVC=1, COMMAND
3462 042516 012704 043740                               ;MOV @T24PK3,R4                 ;SET UP R4 WITH PACKET ADDRESS
3463 042522 012737 000400 043746                       ;MOV @256,,T24SZ                ;SET UP RECORD SIZE IN PACKET
3464 042530 010465 000000                               ;MOV R4,TSD6(R5)                ;ISSUE COMMAND
3465 042534 004737 016340                               ;JSR PC,WAITF                    ;WAIT FOR SSR!BIT1!BIT2 TO SET
3466 042540 016501 000002                               ;MOV TSSR(R5),R1                ;GET TSSR CONTENTS
3467 042544 012702 104210                               ;MOV @SSR!@!NXM!BIT3,R2        ;SET UP EXPECTED
3468 042550 020102                                       ;CMP R1,R2                       ;ARE THEY EQUAL
3469 042552 001406                                       ;BEQ 75$                          ;BR, IF OK
3470 042554 005237 002212                               ;INC FATFLG                       ;BUMP COUNT
3474 042560
042560 104456
042562 000734
042564 044201
042566 012136
3475 042570 75$: CKLOOP                                ;LOOP IF SELECTED
042570 104406
3476 042572 80$:
3477 042572
042572
042572 104403
3478 042574 023727 002212 000017                       ;CMP FATFLG,@15,                ;IS ERROR COUNT AT 25
3479 042602 103402                                       ;BLO 999$                          ;BR, IF LESS THAN 25
3480 042604 004737 017272                               ;JSR PC,CKDROP                    ;TRY TO DROP THE UNIT
3481 042610
999$:

```

TRAP C\$ERRRD  
.WORD 475  
.WORD WRTMSG  
.WORD SFIMSG

TRAP C\$CLP1

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

;READ, ACK, CVC=1, COMMAND  
;SET UP R4 WITH PACKET ADDRESS  
;SET UP RECORD SIZE IN PACKET  
;ISSUE COMMAND  
;WAIT FOR SSR!BIT1!BIT2 TO SET  
;GET TSSR CONTENTS  
;SET UP EXPECTED  
;ARE THEY EQUAL  
;BR, IF OK  
;BUMP COUNT  
;TSSR INCORRECT AFTER READ DATA

TRAP C\$ERRRD  
.WORD 475  
.WORD T24NXM  
.WORD PKTSSR

TRAP C\$CLP1

;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>  
L10066:

TRAP C\$ESUB





TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3535
3536
3537
3538
3539
3540 042712 004737 011104      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
3541 042716 004737 016426      JSR      PC,CHKTSSR     ;SEE HOW TSSR IS
3542 042722 103407              BCS      30$           ;BR, IF NO PROBLEM
3543 042724 010001              MOV      R0,R1         ;SAVE TSSR
3544 042726 005237 002212      INC      FATFLG        ;BUMP COUNT
3548 042732              ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      042732 104456
      042734 000737              TRAP    C$ERRHRD
      042736 045076              .WORD  479
      042740 012136              .WORD  T24RWN
3549 042742              30$:   CKLOOP          ;LOOP IF SELECTED              .WORD  PKTSSR
      042742 104406              TRAP    C$CLP1
3550 042744 012703 000400      MOV      #256.,R3      ;RECORD SIZE
3551 042750 013737 003114 043742  MOV      FREE,T24RB     ;STARTING WRITE BUFFER ADDRESS
3552
3553
3554
3555
3556
3557
3558
3559 042756 012737 100401 043740      MOV      #100401,T24PK3 ;READ REVERSE DATA,ACK COMMAND
3560 042764 012704 043740      MOV      #T24PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
3561 042770
3562 042770 010337 043746      65$:   MOV      R3,T24SZ   ;SET UP RECORD SIZE IN PACKET
3563 042774 010465 000000      MOV      R4,TSD8(R5)   ;ISSUE COMMAND
3564 043000 004737 016340      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
3565 043004 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
3566 043010 012702 100206      MOV      #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
3567 043014 020102              CMP      R1,R2         ;ARE THEY EQUAL
3568 043016 001406              BEQ      75$           ;BR, IF OK
3569 043020 005237 002212      INC      FATFLG        ;BUMP COUNT
3573 043024              ERRHRD  ERRNO,T24WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      043024 104456              TRAP    C$ERRHRD
      043026 000740              .WORD  480
      043030 044541              .WORD  T24WDE
      043032 012136              .WORD  PKTSSR
3574 043034              75$:   CKLOOP          ;LOOP IF SELECTED              TRAP    C$CLP1
      043034 104406
3575
3576
3577
3578
3579
3580
3581
3582 043036 013701 043650      MOV      T24BFR+6,R1   ;GET MESSAGE BUFFER
3583 043042 010102      MOV      R1,R2         ;SET UP EXPECTED
3584 043044 052702 002000      BIS      #BIT10,R2     ;SET THE NEF BIT IN EXPECTED
3585 043050 020102      CMP      R1,R2         ;ARE THEY EQUAL
3586 043052 001406      BEQ      180$          ;BR, IF EQUAL (ALL IS WELL)
3587 043054 005237 002212      INC      FATFLG        ;BUMP COUNT

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

3591	043060				ERRHRD	ERRNO,T24NEF,EXPREC			;THE RLL BIT WAS NOT SET IN XSTO
	043060	104456							TRAP C\$ERHRD
	043062	000741							.WORD 481
	043064	043770							.WORD T24NEF
	043066	015564							.WORD EXPREC
3592	043070		180\$:		CKLOOP				
	043070	104406							TRAP C\$CLP1
3593	043072				ENDSUB				; >>>>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>>>>
	043072								L10067:
	043072	104403							TRAP C\$ESUB
3594	043074	023727	002212	000017	CMP	FATFLG,015.			;IS ERROR COUNT AT 25
3595	043102	103402			BLO	999\$			;BR, IF LESS THAN 25
3596	043104	004737	017272		JSR	PC,CKDROP			;TRY TO DROP THE UNIT
3597	043110								



TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3651
3652 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
3653 ;
3654 ;*****
3655
3656 043212 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
3657 043216 004737 016426 JSR PC,CHKTSSR ;SEE HOW TSSR IS
3658 043222 103407 BCS 30$ ;BR, IF NO PROBLEM
3659 043224 010001 MOV R0,R1 ;SAVE TSSR
3660 043226 005237 002212 INC FATFLG ;BUMP COUNT
3664 043232 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      043232 104456 TRAP C$ERRHD
      043234 000744 .WORD 484
      043236 045076 .WORD T24RWN
      043240 012136 .WORD PKTSSR
3665 043242 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      043242 104406
3666 043244 012703 000400 MOV #256.,R3 ;RECORD SIZE
3667 043250 013737 003114 043742 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
3668
3669 ;*****
3670 ;WRITE DATA,ACK,CVC=1 COMMAND
3671 ;
3672 ;*****
3673
3674
3675 043256 012737 140005 043740 MOV #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
3676 043264 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
3677 043270 65$:
3678 043270 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
3679 043274 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
3680 043300 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
3681 043304 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
3682 043310 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
3683 043314 020102 CMP R1,R2 ;ARE THEY EQUAL
3684 043316 001406 BEQ 75$ ;BR, IF OK
3685 043320 005237 002212 INC FATFLG ;BUMP COUNT
3689 043324 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      043324 104456 TRAP C$ERRHD
      043326 000745 .WORD 485
      043330 005111 .WORD WRERR
      043332 012136 .WORD PKTSSR
3690 043334 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      043334 104406
3691 043336 012703 000400 MOV #256.,R3 ;RECORD SIZE
3692 043342 013737 003114 043742 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
3693
3694 ;*****
3695 ;READ REVERSE DATA,ACK COMMAND
3696 ;
3697 ;*****
3698
3699
3700 043350 012737 100401 043740 MOV #100401,T24PK3 ;READ REVERSE DATA,ACK COMMAND
3701 043356 012704 043740 165$: MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
3702 043362 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
3703 043366 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3704 043372 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
3705 043376 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
3706 043402 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
3707 043406 020102 CMP R1,R2 ;ARE THEY EQUAL
3708 043410 001406 BEQ 170$ ;BR, IF OK
3709 043412 005237 002212 INC FATFLG ;BUMP COUNT
3713 043416 ERRHRD ERRNO,T24TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      043416 104456 TRAP C$ERHRD
      043420 000746 .WORD 486
      043422 046144 .WORD T24TRL
      043424 012136 .WORD PKTSSR
3714 043426 170$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      043426 104406
3715 043430 012703 000400 MOV #256.,R3 ;RECORD SIZE
3716 043434 013737 003114 043742 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
3717
3718 ;*****
3719 ;
3720 ;READ REVERSE DATA,ACK COMMAND
3721 ;
3722 ;*****
3723
3724 043442 012737 100401 043740 MOV #100401,T24PK3 ;READ REVERSE DATA,ACK COMMAND
3725 043450 012704 043740 195$: MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
3726 043454 010337 043746 MOV R3,T24S7 ;SET UP RECORD SIZE IN PACKET
3727 043460 010465 000000 MOV R4,TSD8(R5) ;ISSUE COMMAND
3728 043464 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
3729 043470 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
3730 043474 012702 100204 MOV #SSR!SC!BIT2,R2 ;SET UP EXPECTED
3731 043500 020102 CMP R1,R2 ;ARE THEY EQUAL
3732 043502 001406 BEQ 200$ ;BR, IF OK
3733 043504 005237 002212 INC FATFLG ;BUMP COUNT
3737 043510 ERRHRD ERRNO,T24TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      043510 104456 TRAP C$ERHRD
      043512 000747 .WORD 487
      043514 046144 .WORD T24TRL
      043516 012136 .WORD PKTSSR
3738 043520 200$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      043520 104406
3739 043522 013701 043656 MOV T24BFR+14,R1 ;GET MESSAGE BUFFER (XST3)
3740 043526 010102 MOV R1,R2 ;SET UP EXPECTED
3741 043530 052702 000001 BIS #BIT0,R2 ;SET THE RIB BIT IN EXPECTED
3742 043534 020102 CMP R1,R2 ;ARE THEY EQUAL
3743 043536 001406 BEQ 210$ ;BR, IF EQUAL (ALL IS WELL)
3744 043540 005237 002212 INC FATFLG ;BUMP COUNT
3748 043544 ERRHRD ERRNO,T24LOR,EXPREC ;THE RIB BIT WAS NOT SET IN XST0
      043544 104456 TRAP C$ERHRD
      043546 000750 .WORD 488
      043550 044042 .WORD T24LOR
      043552 015564 .WORD EXPREC
3749 043554 210$: CKLOOP TRAP C$CLP1
      043554 104406
3750 043556 ENDSUB ;>>>>>>>>> END SUBTEST >>>>>>>>>
      043556 L10070: TRAP C$ESUB
      043556 104403
3751 043560 023727 002212 000017 CMP FATFLG,#15. ;IS ERROR COUNT AT 25
3752 043566 103402 BLO 999$ ;BR, IF LESS THAN 25

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3753 043570 004737 017272          JSR    PC,CKDROP          ;TRY TO DROP THE UNIT
3754 043574          999$:
3755          ;
3756          ;
3757          ;
3758 043574 004737 016546          JSR    PC,TSTLOOP        ;DO WE NEED TO ITERATE TEST
3759 043600 103002          BCC    163$              ;BT, IF NO LOOP REQUIRED
3760 043602 000137 034412          JMP    T24LOOP          ;EXECUTE AGAIN
3761 043606          163$:
3762 043606          EXIT    TST          ;ALL DONE THIS TEST
      043606 104432          TRAP   C$EXIT
      043610 002654          .WORD  L10052-

```

## TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3764
3765
3766
3768      043620
3770 043620 100204
3771 043620 100204
3772 043622 043630
3773 043624 000000
3774 043626 000012
3775 043630
3776 043630 043642
3777 043632 000000
3778 043634 000024
3779 043636 000000
3780 043640 000000
3781 043642
3782
3783
3784
3786      043730
3788 043730 100206
3789 043732 043750
3791 043734 000000
3792 043736 000006
3793
3797 043740
3798 043740 100205
3799 043742
3800 043742 003114
3801 043744 000000
3802 043746 000000
3803
3804
3805
3806
3807 043750
3808 043750 010
3809 043751 200
3810 043752 000000
3811 043754 000000
3812
3813
3814
3815
3816
3817 043756 100005
3818 043760 100405
3819 043762 102005
3820 043764 177777
3821 043766 000000
3822
3823

```

```

;
;LOCAL STORAGE FOR THIS TEST
;
      .=<. +10>&177770
T24PACKET:
      .WORD 100204
      .WORD T24DATA
      .WORD 0
      .WORD 10.
T24DATA:
      .WORD 124BFR
      .WORD 0
      .WORD 20.
      .WORD 0
T24DSW: .WORD 0
T24BFR: .BLKW 25.
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
      .=<. +10>&177770
T24PK2:
      .WORD 100206
      .WORD T24BF2
      .WORD 0
      .WORD 6.
T24PK3:
      .WORD 100205
T24RB:
T24WB: .WORD FREE
      .WORD 0
T24SZ: .WORD 0
      .EVEN
;
;
;
T24BF2:
T24BS0: .BYTE 10
T24BS1: .BYTE 200
T24S2: .WORD 0
T24S3: .WORD 0
;
;
      .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T24RN: .WORD 100005
T24WR: .WORD 100405
T24CON: .WORD 102005
      .WORD 177777
T24DLY: .WORD 0

```

```

;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;
;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
;DRIVE SELECTION BITS 2-0
;MESSAGE BUFFER
;WRITE SUB SYS MEM COMMAND, IE AND ACK
;ADDRESS OF SELECT BLOCK DATA
;SIZE OF DATA PACKET
;READ COMMAND, IE AND ACK
;ADDRESS OF WRITE BUFFER
;SIZE OF BUFFER (EXTENT)
;BSEL0 AREA
;BSEL1 AREA
;SEL 2 AREA
;DATA AREA
;READ DATA (NEXT)
;READ DATA RETRY
;WRITE CONTINUOUS
;END OF DATA
;DELAY STORAGE AREA

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3825
3826
3827          ;+
3828          ;LOCAL TEXT MESSAGES FOR TEST
3829          ;-
3830 043770    116    105    106  T24NEF: .ASCIZ 'NEF Not Set After NON-EXECUTABLE FUNCTION'
3831 044042    122    111    102  T24LOR: .ASCIZ 'RIB Not Set After READ REVERSE Into BOT'
3832 044112    124    123    123  T24WDG: .ASCIZ 'TSSR Not Correct After Illegal Buffer Address Bits Set'
3833 044201    124    123    123  T24NXM: .ASCIZ 'TSSR Not Correct After NXM Memory Address In Packet'
3834 044265    124    123    123  T24WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
3835 044342    111    154    154  T24ILA: .ASCIZ 'Illegal Address Bits, Failed To Set ILA Bit In XST0'
3836 044426    111    154    154  T24LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
3837 044507    122    105    101  T24SSR: .ASCIZ 'READ COMMAND Not Accepted'
3838 044541    124    123    123  T24WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
3839 044613    124    141    160  T24BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
3840 044660    104    141    164  T24DTA: .ASCIZ 'Data Written To Tape Not Equal To Data Read From Tape'
3841 044746    122    105    101  T24EOT: .ASCIZ 'READ DATA OVER EOT GAVE NO TAPE STATUS ALERT'
3842 045023    124    123    123  T24TM: .ASCIZ 'TSSR Not Correct After READ COMMAND Reject'
3843 045076    122    145    167  T24RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
3844 045145    122    101    115  T24RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
3845 045220    124    123    123  T24AM3: .ASCIZ 'TSSR Init. Failed After READ COMMAND'
3846 045265    104    162    151  T24OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL." In TSSR'
3847 045340    124    123    123  T24WDD: .ASCIZ 'TSSR Not Correct After READ DATA Command, SWB Bit Set'
3848 045426    124    123    123  T24WDC: .ASCIZ 'TSSR Not Correct After READ DATA Command'
3849 045477    103    126    103  T24VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
3850 045552    124    123    102  T248A: .ASCIZ 'TSBA Not Correct After READ DATA Command'
3851 045623    127    122    111  T24WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
3852 045712    122    145    141  T24LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
3853 045774    122    145    141  T24LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
3854 046056    122    145    163  T24PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
3855 046144    122    145    141  T24TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
3856 046232    102    141    163  TST24ID: .ASCIZ 'Basic Read Data (Forward and Reverse)'
3857
3858          .EVEN
3859          ;+
3860          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3861          ;WRITE SUBSYSTEM MEMORY COMMAND
3862          ;
3863          ;-
3864
3865 046300
3866 046300    T24REST:
3867 046304    012701  043620    SAVREG
3868 046310    012721  100004    MOV     #T24PACKET,R1
3869 046314    012721  043630    MOV     #100004,(R1)+
3870 046320    005021          CLR     #T24DATA,(R1)+
3871 046322    012721  000012    MOV     #10.,(R1)+
3872 046326    012721  043642    MOV     #T24BFR,(R1)+
3873 046332    005021          CLR     (R1)+
3874 046334    012721  000024    MOV     #20.,(R1)+
3875 046340    005021          CLR     (R1)+
3876 046342    012711  000000    MOV     #0,(R1)
3877 046346    012702  000030    MOV     #24.,R2
3878 046352    012762  177777  043642  64$: MOV     #177777,T24BFR(R2)
3879 046360    005742          TST     -(R2)
3880 046362    022702  000000    CMP     #0,R2
3881 046366    001371          BNE    64$

```



TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3882 046370 000207          RTS      PC          ;RETURN
3883
3884
3885 046372          T24RT2:
3886 046372          SAVREG          ;SAVE THE REGISTERS
3887 046376 012701 043730  MOV     #T24PK2,R1      ;START OF THE PACKET
3888 046402 012721 100206  MOV     #100206,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK, IE
3889 046406 012721 043750  MOV     #T24BF2,(R1)+  ;ADDRESS OF DATA BLOCK
3890 046412 005021          CLR     (R1)+           ;EXTENDED ADDRESS
3891 046414 012721 000006  MOV     #6,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
3892 046420 005021          CLR     (R1)+
3893 046422 012701 043750  MOV     #T24BF2,R1     ;POINT TO DATA SEL AREA
3894 046426 005021          CLR     (R1)+
3895 046430 005011          CLR     (R1)
3896 046432 000207          RTS      PC          ;RETURN
3897 046434          T24RT3:
3898 046434          SAVREG          ;SAVE THE REGISTERS
3899 046440 012701 043740  MOV     #T24PK3,R1     ;START OF THE PACKET
3900 046444 012721 000000  MOV     #0,(R1)+       ;CLEAR AREA OUT
3901 046450 012721 000000  MOV     #0,(R1)+       ;ADDRESS OF DATA BLOCK
3902 046454 005021          CLR     (R1)+           ;EXTENDED ADDRESS
3903 046456 012711 000000  MOV     #0,(R1)        ;SIZE OF DATA BLOCK IN BYTES
3904 046462 000207          RTS      PC          ;RETURN
3905 046464          ENDTST
      046464
      046464 104401          L10052: TRAP  C$ETST

```



TEST 5: SPACE RECORDS

```

3965 046546 004737 016064 53: JSR PC,SOF INTT ;DO INITIALIZE ON CONTROLLER
3966 046552 103427 BCS 101 ;BR, IF INIT WAS OK
3967 046554 DELAY 250 ;DELAY IF REQUIRED
      046554 012727 000250
      046560 000000
      046562 013727 002116
      046566 000000
      046570 005367 177772
      046574 001375
      046576 005367 177756
      046602 001367
3968 046604 005337 054032 DEC T25DLT ;DEC DELAY COUNTER
3969 046610 001356 BNE 58 ;BR, IF LOOP IS REQUIRED
3970 046612 005237 002112 INC FATFLG ;BUMP COUNT
3974 046616 016501 000002 MOV TSSR(R5),R1 ;CONTENTS OF TSSR REGISTER
3975 046622 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      046622 104455
      046624 000765 TRAP C1ERRDF
      046626 003650 .WORD 501
      046630 012124 .WORD SFIERR
      .WORD SFIMSG
3976 046632
3977 046632 013737 002172 053700 103: MOV UNITN,T25DSW ;SET UP DRIVE NUMBER
3978 046640 012704 053660 MOV @T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
3979
3980 ;*****
3981 ;
3982 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPCR)
3983 ;
3984 ;*****
3985
3986 046644 004737 010752 JSR PC,WRTPCR ;ISSUE WRITE CHARACTERISTICS
3987 046650 103407 BCS 151 ;BR, IF COMMAND ISSUED OK
3988 046652 005237 002212 INC FATFLG ;BUMP COUNT
3992 046656 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
3993 046660 ERRHRD ERRNO,WRTPMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      046660 104456 TRAP C1ERRRD
      046662 000766 .WORD 502
      046664 005054 .WORD WRTPMSG
      046666 012124 .WORD SFIMSG
3994
3995 ;*****
3996 ;
3997 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
3998 ;
3999 ;*****
4000
4001 046670 151: CKLOOP
      046670 104406 TRAP C1CLP1
4002 046672 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
4003 046676 103407 BCS 301 ;BR, IF NO PROBLEM
4004 046700 010001 MOV R0,R1 ;SAVE TSSR
4005 046702 005237 002212 INC FATFLG ;BUMP COUNT
4009 046706 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      046706 104456 TRAP C1ERRRD
      046710 000767 .WORD 503
      046712 055005 .WORD T25RWN
      046714 012136 .WORD PKTSSR

```

TEST 5: SPACE RECORDS

```

4010 046716 104406 50$: CKLOOP ;LOOP IF SELECTED TRAP C1CLP1
      046716
4011
4012
4013
4014 ;*****
4015 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
4016 ;*****
4017
4018 046720 013701 053710 MOV T25BFR+6,R1 ;PICK UP XSTO
4019 046724 010102 MOV R1,R2 ;SET UP EXPECTED
4020 046726 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
4021 046732 020102 CMP R1,R2 ;DOES EXP = REC'D
4022 046734 001406 BEQ 40$ ;BR, IF EQUAL (OK)
4023 046736 005237 002212 INC FATFLG ;BUMP COUNT
4027 046742 ERRHRD ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      046742 104456 TRAP C1ERHRD
      046744 000770 .WORD 504
      046746 054175 .WORD T25BOT
      046750 015564 .WORD EXPREC
4028 046752 104406 40$: CKLOOP ;LOOP IF SELECTED TRAP C1CLP1
      046752
4029 046754 012703 000400 MOV #256,,R3 ;RECORD SIZE
4030 046760 013737 003114 054002 MOV FREE,T25RB ;STARTING WRITE BUFFER ADDRESS
4031
4032 ;*****
4033 ;WRITE DATA,ACK,CVC=1 COMMAND
4034
4035
4036 ;*****
4037
4038 046766 012737 140005 054000 MOV #140005,T25PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
4039 046774 012704 054000 MOV #T25PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
4040 047000
4041 047000 010337 054006 65$: MOV R3,T25SZ ;SET UP RECORD SIZE IN PACKET
4042 047004 013777 054030 134102 MOV T25CNT,8FREE ;LOAD UP RECORD COUNTER IN WRT BUFFER
4043 047012 062737 000001 054030 ADD #1,T25CNT ;GET READY FOR NEXT RECORD
4044 047020 010465 000000 MOV R4,T25DB(R5) ;ISSUE COMMAND
4045 047024 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
4046 047030 016501 000002 MOV T25R(R5),R1 ;GET T25R CONTENTS
4047 047034 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
4048 047040 020102 CMP R1,R2 ;ARE THEY EQUAL
4049 047042 001411 BEQ 75$ ;BR, IF OK
4050 047044 032701 000004 BIT #BIT2,R1 ;CHECK FOR TAPE STATUS ALERT
4051 047050 001014 BNE 120$ ;BR, IF TSA IS SET (SUSPECT IS EOT)
4052 047052 005237 002212 INC FATFLG ;BUMP COUNT
4056 047056 ERRHRD ERRNO,WRTERR,PKTSSR ;T25R INCORRECT AFTER WRITE DATA
      047056 104456 TRAP C1ERHRD
      047060 000771 .WORD 505
      047062 005111 .WORD WRTERR
      047064 012136 .WORD PKTSSR
4057 047066 104406 75$: CKLOOP ;LOOP IF SELECTED TRAP C1CLP1
      047066
4058 047070 005203 INC R3 ;BUMP RECORD SIZE
4059 047072 022703 001000 CMP #512,,R3 ;END OF RECORD YET
4060 047076 001340 BNE 65$ ;BR, IF MORE RECORDS TO WRITE
4061 047100 000415 BR 125$ ;ENOUGH RECORDS

```

TEST 5: SPACE RECORDS

```

4062 047102      120$:
4063
4064
4065
4066
4067
4068
4069
4070 047102 013701 053710      MOV      T25BFR+6,R1      ;QUICK CHECK FOR EOT SET
4071 047106 010102      MOV      R1,R2           ;SET UP EXPECTED
4072 047110 052702 000001      BIS      @BIT0,R2        ;SET THE EOT BIT X5TO
4073 047114 020102      CMP      R1,R2           ;IS THE EOT BIT SET IN X5TO
4074 047116 001406      BEQ      125$           ;BR, IF SET (GOOD)
4075 047120 005237 002212      INC      FATFLG          ;BUMP COUNT
4079 047124      ERRDF  ERRNO,T25NET,EXPREC ;DEVICE FATAL NOT EOT FOUND ETC.
      047124 104455      TRAP    C$ERRDF
      047126 000772      .WORD  506
      047130 054331      .WORD  T25NET
      047132 015564      .WORD  EXPREC
4080 047134      125$:
4081
4082
4083
4084
4085
4086
4087
4088 047134 004737 011104      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
4089 047140 103407      BCS      130$           ;BR, IF NO PROBLEM
4090 047142 010001      MOV      R0,R1          ;SAVE TSSR
4091 047144 005237 002212      INC      FATFLG          ;BUMP COUNT
4095 047150      ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      047150 104456      TRAP    C$ERRHRD
      047152 000773      .WORD  507
      047154 055005      .WORD  T25RWN
      047156 012136      .WORD  PKTSSR
4096 047160      130$: CKLOOP          ;LOOP IF SELECTED
      047160 104406      TRAP    C$CLP1
4097 047162 012737 000007 053700      MOV      @7,T25DSW      ;SET UP DRIVE NUMBER
4098 047170 012704 053660      MOV      @T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4099
4100
4101
4102
4103
4104
4105
4106 047174 004737 010752      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
4107 047200 103407      BCS      140$           ;BR, IF COMMAND ISSUED OK
4108 047202 005237 002212      INC      FATFLG          ;BUMP COUNT
4112 047206 010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
4113 047210      ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      047210 104456      TRAP    C$ERRHRD
      047212 000774      .WORD  508
      047214 005054      .WORD  WRTMSG
      047216 012124      .WORD  SFIMSG
4114 047220      140$: CKLOOP          ;SCOPE LOOP

```

TEST 5: SPACE RECORDS

```

047220 104406
4115 047222 005737 002216          BIT      EXTFEA          ;CHECK FOR EXTENDED FEATURES          TRAP      C$CLP1
4116 047226 0010          BNE      160$          ;BR IF SWITCH IS ON
4117
4118 047230 112737 000200 054011      MOVW     #200,T25B51      ;WRITE MISCELLANEOUS CONT/READ STATUS
4119 047236 112737 000010 054010      MOVW     #10,T25B50      ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4120 047244 012704 053770          MOV      #T25PK2,R4      ;WRITE SUBSYS MEM PACKET
4121 047250 010465 000000          MOV      R4,T25DB(R5)    ;ISSUE COMMAND
4122 047254 004737 016426          JSR      PC,CHKTSSR      ;WAIT FOR SSR
4123 047260 103407          BCS     150$            ;BR, IF NO ERROR
4124 047262 010001          MOV      R0,R1          ;ERROR, SAVE TSSR
4125 047264 005237 002212          INC      FATFLG         ;BUMP COUNT
4129 047270          ERRHRD  ERRNO,T25SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
                                TRAP      C$ERRRD
                                .WORD     509
                                .WORD     T25SSR
                                .WORD     PKTSSR
                                TRAP      C$CLP1
047270 104456
047272 000775
047274 054034
047276 012136
4130 047300          150$:  CKLOOP          ;LOOP IF SELECTED
047300 104406
4131 047302 012737 000007 053700      MOV      #7,T25DSW      ;SET UP DRIVE NUMBER
4132 047310 012704 053660      MOV      #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4133
4134          ;*****
4135          ;
4136          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPCHR)
4137          ;
4138          ;*****
4139
4140 047314 004737 010752          JSR      PC,WRTPCHR      ;ISSUE WRITE CHARACTERISTICS
4141 047320 103407          BCS     160$            ;BR, IF COMMAND ISSUED OK
4142 047322 005237 002212          INC      FATFLG         ;BUMP COUNT
4146 047326 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
4147 047330          ERRHRD  ERRNO,WRTPMSG,SFMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP      C$ERRRD
                                .WORD     510
                                .WORD     WRTPMSG
                                .WORD     SFMSG
047330 104456
047332 000776
047334 005054
047336 012124
4148 047340          160$:  CKLOOP          ;SCOPE LOOP
047340 104406
4149 047342 016501 000002          MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
4150 047346 032701 000100          BIT      #OFL,R1        ;CHECK FOR THE OFFLINE BIT SET
4151 047352 001006          BNE     170$            ;BR, IF OFFLINE (GOOD)
4152 047354 005237 002212          INC      FATFLG         ;BUMP COUNT
4156 047360          ERRDF  ERRNO,T25OFL,SFMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
                                TRAP      C$ERRDF
                                .WORD     511
                                .WORD     T25OFL
                                .WORD     SFMSG
047360 104455
047362 000777
047364 055054
047366 012124
4157 047370          170$:  CKLOOP          ;LOOP IF SELECTED
047370 104406
                                TRAP      C$CLP1
4158
4159          ;*****
4160          ;
4161          ;SPACE FORWARD COMMAND IN PLACE
4162          ;
4163          ;*****
4164

```

TEST 5: SPACE RECORDS

4165	047372	012737	140010	054000	180\$:	MOV	#140010,T25PK3	;SPACE FORWARD COMMAND IN PLACE
4166	047400	012737	000001	054002		MOV	#1,T25RH	;NUMBER OF RECORDS TO SPACE
4167	047406	012704	054000			MOV	#T25PK3,R4	;R4 = POINTER TO PACKET
4168	047412	010465	000000			MOV	R4,T25DB(R5)	;ISSUE COMMAND
4169	047416	004737	016340			JSR	PC,WAITE	;WAIT FOR SSR TO SET
4170	047422	016501	000002			MOV	TSSR(R5),R1	;GET TSSR CONTENTS
4171	047426	012702	100306			MOV	#SSR!SC!OFL!BIT1!BIT2,R2	;SET UP EXPECTED
4172	047432	020102				CMP	R1,R2	;ARE THEY EQUAL
4173	047434	001406				BEQ	190\$	;BR, IF OK ESP. FUNCTION REJECT
4174	047436	005237	002212			INC	FATFLG	;BUMP COUNT
4178	047442					ERRHRD	ERRNO,T25TM,PKTSSR	;TSSR INCORRECT AFTER TAPE MOTION CMD
	047442	104456						TRAP C\$ERHRD
	047444	001000						.WORD 512
	047446	054242						.WORD T25TM
	047450	012136						.WORD PKTSSR
4179	047452				190\$:	CKLOOP		;LOOP IF SELECTED
	047452	104406						TRAP C\$CLP1
4180	047454					ENDSUB		;##### END SUBTEST #####
	047454							L10072:
	047454	104403						TRAP C\$ESUB
4181	047456	023727	002212	000017		CMP	FATFLG,#15,	;IS ERROR COUNT AT 25
4182	047464	103402				BLO	999\$	;BR, IF LESS THAN 25
4183	047466	004737	017272			JSR	PC,CKDROP	;TRY TO DROP THE UNIT
4184	047472				999\$:			

TEST 5: SPACE RECORDS

```

4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197 047472
    047472
    047472 104402
4198 047474 004737 055216
4199 047500 004737 055352
4200 047504 004737 055310
4201
4202
4203
4204
4205
4206
4207
4208 047510 004737 016064
4209 047514 103407
4210 047516 005237 002212
4214 047522 010001
4215 047524
    047524 104455
    047526 001001
    047530 003650
    047532 012124
4216 047534 012737 000007 053700 10%:
4217
4218 047542 012704 053660
4219
4220
4221
4222
4223
4224
4225
4226 047546 004737 010752
4227 047552 103407
4228 047554 005237 002212
4232 047560 010001
4233 047562
    047562 104456
    047564 001002
    047566 005054
    047570 012124
4234
4235
4236
4237
4238
;
```

```

;
;TEST 5, SUBTEST 2
;
;VERIFIES THAT A SPACE RECORDS REVERSE COMMAND WITH
;THE CLEAR VOLUME CHECK (CVC) BIT CLEAR IS REJECTED IF
;THE VOLUME CHECK (VCK) FLAG IS SET.
;
;
;
;-----
                BGNSUB                                ;>>>>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
                                         T5.2:
                                         TRAP      C$BSUB
                JSR      PC,T25REST                    ;SET COMMAND PACKET
                JSR      PC,T25RT3                     ;SET UP OTHER COMMAND PACKET
                JSR      PC,T25RT2                     ;SET UP OTHER COMMAND PACKET
;*****
;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
;*****
                JSR      PC,SOFINIT                    ;DO INITIALIZE ON CONTROLLER
                BCS      10%                            ;BR IF INIT WAS OK
                INC      FATFLG                        ;BUMP COUNT
                MOV      R0,R1                         ;CONTENTS OF TSSR REGISTER
                ERROF   ERRNO,SFIERR,SFIMSG           ;FATAL ERROR TSSR WAS NOT OK
                                         TRAP      C$EROF
                                         .WORD    513
                                         .WORD    SFIERR
                                         .WORD    SFIMSG
                MOV      #7,125DSW                    ;SET UP DRIVE NUMBER
                MOV      #125PACKET,R4                 ;SUBROUTINE NEEDS PACKET ADDRESS
;*****
;WRITE CHARACTERISTICS COMMAND (CALL 1) WRTCHR)
;*****
                JSR      PC,WRTCHR                     ;ISSUE WRITE CHARACTERISTICS
                BCS      15%                            ;BR, IF COMMAND ISSUED OK
                INC      FATFLG                        ;BUMP COUNT
                MOV      R0,R1                         ;SAVE CONTENTS OF TSSR
                ERRHRD  ERRNO,WRTMSG,SFIMSG           ;WRITE CHARACTERISTIC FAILED
                                         TRAP      C$ERHRD
                                         .WORD    514
                                         .WORD    WRTMSG
                                         .WORD    SFIMSG
;*****
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;

```



TEST 5: SPACE RECORDS

```

4239
4240
4241 047572          ;*****
      047572 104406 15$: CKLOOP
4242 047574 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND TRAP C$CLP1
4243 047600 103407 BCS 30$ ;BR, IF NO PROBLEM
4244 047602 010001 MOV R0,R1 ;SAVE TSSR
4245 047604 005237 002212 INC FATFLG ;BUMP COUNT
4249 047610 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      047610 104456 TRAP C$ERHRD
      047612 001003 .WORD 515
      047614 055005 .WORD T25RWN
      047616 012136 .WORD PKTSSR
4250 047620          30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      047620 104406 140$: TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
4251 047622 005737 002216 BNE 160$ ;BR IF SWITCH IS ON
4253
4254 047630 112737 000200 054011 MOVB #200,T25B51 ;WRITE MISCELLANEOUS CONT/READ STATUS
4255 047636 112737 000010 054010 MOVB #10,T25B50 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4256 047644 012704 053770 MOV #T25PK2,R4 ;WRITE SUBSYS MEM PACKET
4257 047650 010465 000000 MOV R4,T5DB(R5) ;ISSUE COMMAND
4258 047654 004737 016426 JSR PC,CHKTSSR ;WAIT FOR SSR
4259 047660 103407 BCS 150$ ;BR, IF NO ERROR
4260 047662 010001 MOV R0,R1 ;ERROR, SAVE TSSR
4261 047664 005237 002212 INC FATFLG ;BUMP COUNT
4265 047670 ERRHRD ERRNO,T25SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      047670 104456 TRAP C$ERHRD
      047672 001004 .WORD 516
      047674 054034 .WORD T25SSR
      047676 012136 .WORD PKTSSR
4266 047700          150$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      047700 104406 4267 047702 012737 000007 053700 MOV #7,T25DSW ;SET UP DRIVE NUMBER
4268 047710 012704 053660 MOV #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4269
4270 ;*****
4271 ;
4272 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
4273 ;
4274 ;*****
4275
4276 047714 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
4277 047720 103407 BCS 160$ ;BR, IF COMMAND ISSUED OK
4278 047722 005237 002212 INC FATFLG ;BUMP COUNT
4282 047726 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
4283 047730 ERRHRD ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTIC FAILED
      047730 104456 TRAP C$ERHRD
      047732 001005 .WORD 517
      047734 005054 .WORD WRTMSG
      047736 012124 .WORD SFMSG
4284 047740          160$: CKLOOP ;SCOPE LOOP TRAP C$CLP1
      047740 104406 4285 047742 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4286 047746 032701 000100 BIT #0FL,R1 ;CHECK FOR THE OFFLINE BIT SET
4287 047752 001006 BNE 170$ ;BR, IF OFFLINE (GOOD)
4288 047754 005237 002212 INC FATFLG ;BUMP COUNT

```

TEST 5: SPACE RECORDS

```

4292 047760          ERRDF  ERRNO,T25OFL,SFMSG      ;OFF LINE SHOULD HAVE BEEN SET (BAD)
       047760 104455                                     TRAP  C$ERRDF
       047762 001006                                     .WORD 518
       047764 055054                                     .WORD T25OFL
       047766 012124                                     .WORD SFMSG
4293 047770          170$: CKLOOP                   ;LOOP IF SELECTED
       047770 104406                                     TRAP  C$CLP1
4294
4295                ;*****
4296                ;
4297                ;SPACE REVERSE COMMAND IN PLACE
4298                ;
4299                ;*****
4300
4301 047772 012737 100410 054000 180$:  MOV    *100410,T25PK3       ;SPACE REVERSE COMMAND IN PLACE
4302 050000 012737 000001 054002      MOV    *1,T25RB          ;NUMBER OF RECORDS TO SPACE
4303 050006 012704 054000              MOV    *T25PK3,R4       ;R4 - POINTER TO PACKET
4304 050012 010465 000000              MOV    R4,T25DB(R5)     ;ISSUE COMMAND
4305 050016 004737 016340              JSR    PC,WAITF         ;WAIT FOR SSR TO SET
4306 050022 016501 000002              MOV    TSSR(R5),R1     ;GET TSSR CONTENTS
4307 050026 012702 100306              MOV    *SSR!SC!OFL!BIT1!BIT2,R2  ;SET UP EXPECTED
4308 050032 020102                      CMP    R1,R2            ;ARE THEY EQUAL
4309 050034 001406                      BEQ    190$             ;BR, IF OK ESP. FUNCTION REJECT
4310 050036 005237 002212              INC    FATFLG           ;BUMP COUNT
4314 050042          ERRNO  ERRNO,T25TM,PKTSSR       ;TSSR INCORRECT AFTER TAPE MOTION CMD
       050042 104456                                     TRAP  C$ERRRD
       050044 001007                                     .WORD 519
       050046 054242                                     .WORD T25TM
       050050 012136                                     .WORD PKTSSR
4315 050052          190$: CKLOOP                   ;LOOP IF SELECTED
       050052 104406                                     TRAP  C$CLP1
4316 050054          ENDSUB                           ;>>>>>>>>> END SUBTEST >>>>>>>>>
       050054          L10073:
       050054 104403                                     TRAP  C$ESUB
4317 050056 023727 002212 000017      CMP    FATFLG,*15.     ;IS ERROR COUNT AT 25
4318 050064 103402                      BLO    999$            ;BR, IF LESS THAN 25
4319 050066 004737 017272              JSR    PC,CKDROP       ;TRY TO DROP THE UNIT
4320 050072          999$:
  
```

TEST 5: SPACE RECORDS

```

4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332 050072
      050072
      050072 104402
4333 050074 004737 055216
4334 050100 004737 055310
4335 050104 004737 055352
4336
4337
4338
4339
4340
4341
4342
4343 050110 004737 016064
4344 050114 103407
4345 050116 005237 002212
4349 050122 010001
4350 050124
      050124 104455
      050126 001010
      050130 003650
      050132 012124
4351 050134 013737 002172 053700 10$:
4352
4353 050142 012704 053660
4354
4355
4356
4357
4358
4359
4360
4361 050146 004737 010752
4362 050152 103407
4363 050154 005237 002212
4367 050160 010001
4368 050162
      050162 104456
      050164 001011
      050166 005054
      050170 012124
4369
4370
4371
4372
4373
4374

;
;TEST 5, SUBTEST 3
;VERIFIES THAT SPACE RECORDS FORWARD CAN SPACE ONE
;RECORD OFF BOT AND CAUSES BOT STATUS TO BE CLEARED.
;
;
;
;
;-----
;              BGNSUB                      ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
;                                             TS. 3:
;                                             TRAP      C$BSUB
;JSR      PC,T25RFST                      ;SET COMMAND PACKET
;JSR      PC,T25RT2                       ;SET UP OTHER COMMAND PACKET
;JSR      PC,T25RT3                       ;SET UP OTHER COMMAND PACKET
;
;*****
;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
;
;*****
;JSR      PC,SOFINIT                      ;DO INITIALIZE ON CONTROLLER
;BCS      10$                             ;BR IF INIT WAS OK
;INC      FATELG                           ;BUMP COUNT
;MOV      R0,R1                            ;CONTENTS OF TSSR REGISTER
;ERRDF   ERRNO,SFIERR,SFIMSG             ;FATAL ERROR TSSR WAS NOT OK
;                                             TRAP      C$ERDF
;                                             .WORD    520
;                                             .WORD    SFIERR
;                                             .WORD    SFIMSG
;
;MOV      UNITN,T25DSW                    ;SET UP DRIVE NUMBER
;
;MOV      @T25PACKET,R4                   ;SUBROUTINE NEEDS PACKET ADDRESS
;
;*****
;WRITE CHARACTERISTICS COMMAND (CALL TO WRCHR)
;
;*****
;JSR      PC,WRCHR                        ;ISSUE WRITE CHARACTERISTICS
;BCS      15$                             ;BR, IF COMMAND ISSUED OK
;INC      FATELG                           ;BUMP COUNT
;MOV      R0,R1                            ;SAVE CONTENTS OF TSSR
;ERRHRD  ERRNO,WRTMSG,SFIMSG             ;WRITE CHARACTERISTICS FAILED
;                                             TRAP      C$ERRRD
;                                             .WORD    521
;                                             .WORD    WRTMSG
;                                             .WORD    SFIMSG
;
;*****
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****

```

TEST 5: SPACE RECORDS

```

4375
4376 050172          15$:   CKLOOP
      050172 104406
4377 050174 004737 011104      JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
4378 050200 103407          BCS    50$            ;BR, IF NO PROBLEM
4379 050202 010001          MOV    R0,R1          ;SAVE TSSR
4380 050204 005237 002212      INC    FATFLG        ;BUMP COUNT
4384 050210          ERRHRD  ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      050210 104456
      050212 001012          TRAP   C$ERHRD
      050214 055005          .WORD 522
      050216 012136          .WORD T25RWN
4385 050220          30$:   CKLOOP          ;LOOP IF SELECTED
      050220 104406          TRAP   C$CLP1
4386
4387 ;*****
4388 ;
4389 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
4390 ;
4391 ;*****
4392
4393 050222 013701 053710      MOV    T25BFR+6,R1   ;PIC. UP XSTO
4394 050226 010102          MOV    R1,R2         ;SET UP EXPECTED
4395 050230 052702 000002      BIS    0BIT1,R2     ;SET BOT BIT IN EXPECTED
4396 050234 020102          CMP    R1,R2        ;DOES EXP = REC'D
4397 050236 001406          BEQ    40$          ;BR, IF EQUAL (OK)
4398 050240 005237 002212      INC    FATFLG        ;BUMP COUNT
4402 050244          ERRHRD  ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      050244 104456          TRAP   C$ERHRD
      050246 001013          .WORD 523
      050250 054175          .WORD T25BOT
      050252 015564          .WORD EXPREC
4403 050254          40$:   CKLOOP          ;LOOP IF SELECTED
      050254 104406          TRAP   C$CLP1
4404 050256 012737 000001 054002  MOV    000001,T25RB  ;NUMBER OF RECORDS TO SPACE OVER
4405
4406 ;*****
4407 ;
4408 ;SPACE FORWARD,ACK,CVC+1 COMMAND
4409 ;
4410 ;*****
4411
4412 050264 012737 140010 054000  MOV    0140010,T25PK3 ;SPACE FORWARD,ACK,CVC+1 COMMAND
4413 050272 012704 054000      MOV    0T25PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
4414 050276
4415 050276 010465 000000      65$:   MOV    R4,T5DB(R5) ;ISSUE COMMAND
4416 050302 004737 016340      JSR    PC,WAITF     ;WAIT FOR SSR TO SET
4417 050306 016501 000002      MOV    TSSR(R5),R1 ;GET TSSR CONTENTS
4418 050312 012702 000200      MOV    0SSR,R2     ;SET UP EXPECTED
4419 050316 020102          CMP    R1,R2        ;ARE THEY EQUAL
4420 050320 001411          BEQ    75$          ;BR, IF OK
4421 050322 032701 000004      BIT    0BIT2,R1    ;CHECK FOR TAPE STATUS ALERT
4422 050326 001006          BNE    75$          ;BR, IF TSA IS SET (SUSPECT IS EOT)
4423 050330 005237 002212      INC    FATFLG        ;BUMP COUNT
4427 050334          ERRHRD  ERRNO,T25WDE,EXPREC ;TSSR INCORRECT AFTER READ DATA
      050334 104456          TRAP   C$ERHRD
      050336 001014          .WORD 524

```

TEST 5: SPACE RECORDS

```

050340 054115 .WORD T25WDE
050342 015564 .WORD EXPREC
4428 050344 75$: CKLOOP ;LOOP IF SELECTED
050344 104406 TRAP C$CLP1
4429 050346
4430
4431 ;*****
4432 ;
4433 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
4434 ;
4435 ;*****
4436
4437 050346 013701 053710 MOV T25BFR+6,R1 ;QUICK CHECK FOR BOT SET
4438 050352 010102 MOV R1,R2 ;SET UP EXPECTED
4439 050354 042702 000002 BIC *BIT1,R2 ;CLEAR THE BOT BIT (XSTO)
4440 050360 020102 CMP R1,R2 ;IS THE BOT BIT SET IN XSTO
4441 050362 001406 BEQ 125$ ;BR, IF SET (GOOD)
4442 050364 005237 002212 INC FATFLG ;BUMP COUNT
4446 050370 ERRHRD ERRNO,T25BNC,EXPREC ;BOT NOT CLEARED AFTER SPACE FROM BOT
050370 104456 TRAP C$ERHRD
050372 001015 .WORD 525
050374 054470 .WORD T25BNC
050376 015564 .WORD EXPREC
4447 050400 125$: CKLOOP TRAP C$CLP1
050400 104406
4448 050402 004737 055352 JSR PC,T25RT3 ;CLEAN UP PACKET
4449 050406 012737 000401 054006 MOV *257.,T25SZ ;SET THE CORRECT SIZE UP
4450
4451 ;*****
4452 ;
4453 ;READ DATA COMMAND IN PLACE
4454 ;
4455 ;*****
4456
4457 050414 012737 140001 054000 MOV *140001,T25PK3 ;READ DATA COMMAND IN PLACE
4458 050422 013737 003114 054002 MOV FREE,T25R8 ;READ BUFFER ADDRESS TO PACKET
4459 050430 012704 054000 MOV *T25PK3,R4 ;R4 = POINTER TO PACKET
4460 050434 010465 000000 MOV R4,T5DB(R5) ;ISSUE COMMAND
4461 050440 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
4462 050444 016501 000002 MOV T5SR(R5),R1 ;GET T5SR CONTENTS
4463 050450 012702 000200 MOV *SSR,R2 ;SET UP EXPECTED
4464 050454 020102 CMP R1,R2 ;ARE THEY EQUAL
4465 050456 001406 BEQ 190$ ;BR, IF OK ESP. FUNCTION REJECT
4466 050460 005237 002212 INC FATFLG ;BUMP COUNT
4470 050464 ERRHRD ERRNO,RDERR,PKTSSR ;T5SR INCORRECT AFTER READ DATA CMD
050464 104456 TRAP C$ERHRD
050466 001016 .WORD 526
050470 005204 .WORD RDERR
050472 012136 .WORD PKTSSR
4471 050474 190$: CKLOOP ;LOOP IF SELECTED
050474 104406 TRAP C$CLP1
4472 050476 017701 132412 MOV *FREE,R1 ;GET FIRST WORD FROM BUFFER
4473 050502 012702 000001 MOV *1,R2 ;SET UP EXPECTED
4474 050506 020102 CMP R1,R2 ;WAS RECORD NUMBERED 1
4475 050510 001406 BEQ 200$ ;BR, IF CORRECT RECORD
4476 050512 005237 002212 INC FATFLG ;BUMP COUNT
4480 050516 ERRHRD ERRNO,T25WNG,EXPREC ;SHOULD HAVE BEEN RECORD NUMBER 1

```

TEST 5: SPACE RECORDS

```

050516 104456
050520 001017
050522 054405
050524 015564
4481 050526 200$: CKLOOP
050526 104406
4482 050530 ENDSUB
050530
050530 104403
4483 050532 023727 002212 000017 CMP FATFLG,#15.
4484 050540 103402 BLO 999$
4485 050542 004737 017272 JSR PC,CKDROP
4486 050546 999$:

```

```

TRAP C$ERHRD
.WORD 527
.WORD T25WNG
.WORD EXPREC

```

```
TRAP C$CLP1
```

```

;>>>>>>>>>> END SUBTEST >>>>>>>>>>
L10074:

```

```

TRAP C$ESUB
;IS ERROR COUNT AT 25
;BR, IF LESS THAN 25
;TRY TO DROP THE UNIT

```

TEST 5: SPACE RECORDS

```

4488      ;
4489      ;
4490      ; TEST 5, SUBTEST 4
4491      ;
4492      ; VERIFIES THAT SPACE RECORDS REVERSE CAN SPACE BACK
4493      ; OVER THE FIRST RECORD ON TAPE.
4494      ;
4495      ;
4496      ;
4497      ;
4498      050546          BGNSUB                                ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>
         050546          104402                                TS.4:
4499      050550  004737  055216          JSR      PC,T25REST          ;SET COMMAND PACKET      TRAP   C$BSUB
4500      050554  004737  055310          JSR      PC,T25RT2         ;SET UP OTHER COMMAND PACKET
4501      050560  004737  055352          JSR      PC,T25RT3         ;SET UP OTHER COMMAND PACKET
4502
4503      ; *****
4504      ;
4505      ; ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
4506      ;
4507      ; *****
4508
4509      050564  004737  016064          JSR      PC,SOFINIT        ;DO INITIALIZE ON CONTROLLER
4510      050570  103407                  BCS      20$                ;BR IF INIT WAS OK
4511      050572  005237  002212          INC      FATFLG             ;BUMP COUNT
4515      050576  010001                  MOV      R0,R1              ;CONTENTS OF TSSR REGISTER
4516      050600          ERRDF  ERRNO,SFIERR,SFIMSG          ;FATAL ERROR TSSR WAS NOT OK
         050600  104455                                TRAP   C$ERDF
         050602  001020                                .WORD 528
         050604  003650                                .WORD SFIERR
         050606  012124                                .WORD SFIMSG
4517      050610  013737  0072  053700  20$:  MOV      UNITN,T25DSW          ;SET UP DRIVE NUMBER
4518
4519      050616  012704  055660          MOV      @T25PACKET,R4     ;SUBROUTINE NEEDS PACKET ADDRESS
4520
4521      ; *****
4522      ;
4523      ; WRITE CHARACTERISTICS COMMAND (CALL. TO WRTCHR)
4524      ;
4525      ; *****
4526
4527      050622  004737  010752          JSR      PC,WRTCHR         ;ISSUE WRITE CHARACTERISTICS
4528      050626  103407                  BCS      25$                ;BR, IF COMMAND ISSUED OK
4529      050630  005237  002212          INC      FATFLG             ;BUMP COUNT
4533      050634  010001                  MOV      R0,R1              ;SAVE CONTENTS OF TSSR
4534      050636          ERRHRD ERRNO,WRTMSG,SFIMSG          ;WRITE CHARACTERISTIC FAILED
         050636  104456                                TRAP   C$ERHRD
         050640  001021                                .WORD 529
         050642  005054                                .WORD WRTMSG
         050644  012124                                .WORD SFIMSG
4535      050646          25$:  CKLOOP                               ;LOOP IF SELECTED      TRAP   C$CLP1
         050646  104406
4536
4537      ; *****
4538      ;
4539      ; ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE

```

```

4540
4541
4542
4543 050650 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
4544 050654 103407 BCS 501 ;BR, IF NO PROBLEM
4545 050656 010001 MOV R0,R1 ;SAVE T25R
4546 050660 005237 002212 INC FATFLG ;BUMP COUNT
4547 050664 ERRHRD ERRNO,T25RWN,PKT5R ;REWIND NOT ACCEPTED
4548 050664 104456 TRAP C$ERRHD
4549 050666 001022 .WORD 530
4550 050670 055005 .WORD T25RWN
4551 050672 012136 .WORD PKT5R
4551 050674 301: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
4552 050674 104406
4553
4554
4555 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (X=10)
4556
4557
4558
4559 050676 013701 053710 MOV T25R+6,R1 ;PICK UP XST0
4560 050702 010102 MOV R1,R2 ;SET UP EXPECTED
4561 050704 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
4562 050710 020102 CMP R1,R2 ;DOES EXP = REC'D
4563 050712 001406 BEQ 401 ;BR, IF EQUAL (OK)
4564 050714 005237 002212 INC FATFLG ;BUMP COUNT
4565 050720 ERRHRD ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
4566 050720 104456 TRAP C$ERRHD
4567 050722 001023 .WORD 531
4568 050724 054175 .WORD T25BOT
4569 050726 015564 .WORD EXPREC
4569 050730 401: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
4570 050730 104406
4571
4572
4573 ;ISSUE SPACE RECORDS COMMAND VALUE IN R3 SETS NUMBER OF RECORDS
4574 ;BIT 15 SETS DIRECTION 0=FORWARD 1=REVERSE
4575
4576
4577
4578 050732 012703 000001 MOV #000001,R3 ;NUMBER OF RECORDS TO SPACE FORWARD
4579 050736 004737 010556 JSR PC,SPACE ;CALL SPACE COMMAND
4580 050742 103410 BCS 501 ;CHECK FOR ERROR
4581 050744 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4582 050750 005237 002212 INC FATFLG ;BUMP COUNT
4583 050754 ERRHRD ERRNO,T25WDE,SFFMSG ;SPACE FORWARD FAILED
4584 050754 104456 TRAP C$ERRHD
4585 050756 001024 .WORD 532
4586 050760 054115 .WORD T25WDE
4587 050762 012172 .WORD SFFMSG
4587 050764 501: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
4588 050766 012737 000001 054002 MOV #1,T25HB ;NUMBER OF RECORDS TO SPACE OVER
4589
4590

```



TEST 5: SPACE RECORDS

```

4591
4592
4593
4594
4595
4596 050774 012757 140410 054000      MOV      0140410,T25PK3      ;SPACE REVERSE,ACK,CVC-1 COMMAND
4597 051002 012704 054000      MOV      0T25PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4598 051006
4599 051006 010465 000000      65$:    MOV      R4,TSDB(R5)        ;ISSUE COMMAND
4600 051017 004737 016340      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
4601 051016 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
4602 051022 012702 000200      MOV      0SSR,R2          ;SET UP EXPECTED
4603 051026 020102
4604 051030 001406      CMP      R1,R2            ;ARE THEY EQUAL
4605 051032 005237 002212      BEQ      75$              ;BR, IF OK
4609 051036      INC      FATFLG           ;BUMP COUNT
      ERRHRD  ERRNO,T25WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      TRAP   C$ERRRD
      .WORD  533
      .WORD  T25WDE
      .WORD  PKTSSR
      051036 104456
      051040 001025
      051042 054115
      051044 012136
4610 051046      75$:    CKLOOP              ;LOOP IF SELECTED
      TRAP   C$CLP1
      051046 104406
4611 051050      120$:
4612 051050 012703 000400      MOV      0256,,R3         ;RECORD SIZE
4613 051054 013737 003114 054002      MOV      FREE,T25RB       ;STARTING READ BUFFER ADDRESS
4614
4615
4616
4617
4618
4619
4620
4621 051062 012737 140001 054000      ;*****
      ;READ DATA,ACK,CVC-1 COMMAND
      ;*****
4622 051070 012704 054000      165$:  MOV      0140001,T25PK3    ;READ DATA,ACK,CVC-1 COMMAND
4623 051074 010337 054006      MOV      0T25PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4624 051100 010465 000000      MOV      R3,T25S2          ;SET UP RECORD SIZE IN PACKET
4625 051104 004737 016340      MOV      R4,TSDB(R5)        ;ISSUE COMMAND
4626 051110 016501 000002      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
4627 051114 012702 000200      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
4628 051110 020102      MOV      0SSR,R2          ;SET UP EXPECTED
4629 051122 001406      CMP      R1,R2            ;ARE THEY EQUAL
4630 051124 005237 002212      BEQ      170$             ;BR, IF OK
4634 051130      INC      FATFLG           ;BUMP COUNT
      ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      TRAP   C$ERRRD
      .WORD  534
      .WORD  RDERR
      .WORD  PKTSSR
      051130 104456
      051132 001026
      051134 005204
      051136 012136
4635 051140      170$:  CKLOOP              ;LOOP IF SELECTED
      TRAP   C$CLP1
      051140 104406
4636 051142 017701 131746      MOV      0FREE,R1          ;GET FIRST WORD FROM BUFFER
4637 051146 012702 000000      MOV      00,R2            ;SET UP EXPECTED
4638 051152 020102      CMP      R1,R2            ;WAS RECORD NUMBERED 1
4639 051154 001406      BEQ      200$             ;BR, IF CORRECT RECORD
4640 051156 005237 002212      INC      FATFLG           ;BUMP COUNT
4644 051162      ERRHRD  ERRNO,T25WNG,EXPREC ;SHOULD HAVE BEEN RECORD NUMBER 1
      TRAP   C$ERRRD
      .WORD  535
      051162 104456
      051164 001027

```

011

TEST 5: SPACE RECORDS

```

051166 054405
051170 015564
4645 051172 200$: CKD DOP
051172 104406
4646 051174 ENDSUB
051174
051174 104405
4647 051176 023727 002212 000017 CMP FATELG, #1%
4648 051204 103402 BLD 999$
4649 051206 004737 017272 JSR PC, CKD DOP
4650 051212 999$:

```

```

.WORD) 125WNG
.WORD) EXPREC
TRAP) C3CPI
;..... END SUBJECT .....
110075:
TRAP) C3ESUB
;IS ERROR COUNT AT 25
;BR, IF LESS THAN 25
;TRY TO DROP THE UNIT

```

TEST 5: SPACE RECORDS

```

4652 ;
4653 ;
4654 ; TEST 5, SUBTEST 5
4655 ;
4656 ; VERIFIES THAT SPACE RECORDS FORWARD CAN SPACE A
4657 ; MULTIPLE NUMBER OF RECORDS (2 THROUGH 64K OR THE
4658 ; MAXIMUM NUMBER OF RECORDS WRITTEN ON THE TAPE,
4659 ; WHICH EVER IS LESS.
4660 ;
4661 ;
4662 ;
4663 ;
4664 ;
4665 051212          BGNSUB                      ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
      051212          TRAP C$B$SUB
      051213 104402
4666 051214 004737 055216 JSR PC,T25REST          ;SET COMMAND PACKET
4667 051220 004737 055310 JSR PC,T25RT2          ;SET UP OTHER COMMAND PACKET
4668 051224 004737 055352 JSR PC,T25RT3          ;SET UP OTHER COMMAND PACKET
4669 051230 013737 054030 054026 MOV T25CNT,T25CN2      ;HOLD NUMBER OF RECORDS
4670 051236 162737 000002 054026 SUB #2,T25CN2          ;ACTUAL NUMBER OF RECORDS ON TAPE
4671 051244 013737 054030 054032 MOV T25CNT,T25DLY      ;SET UP REWIND DELAY COUNTER
4672 ;
4673 ;*****
4674 ;
4675 ; ISSUE CONTROLLER "SOFT" INITIALIZE CARRY BIT CLEAR IF ERROR
4676 ;
4677 ;*****
4678 ;
4679 051252 004737 016064 10$: JSR PC,SOFINIT          ;DO INITIALIZE ON CONTROLLER
4680 051256 103427 BCS 20$          ;BR IF INIT WAS OK
4681 051260 DELAY 250          ;WAIT ABOUT .25 SECONDS
      051260 012727 000250 MOV #250,(PC)+
      051264 000000 .WORD 0
      051266 013727 002116 MOV L$DLY,(PC)+
      051272 000000 .WORD 0
      051274 005367 177772 DEC -6(PC)
      051300 001375 BNE .4
      051302 005367 177756 DEC -22(PC)
      051306 001367 BNE .-20
4682 051310 005337 054032 DEC T25DLY          ;DEC COUNTER
4683 051314 001356 BNE 10$          ;BR, IF MORE LOOPS REQUIRED
4684 051316 016501 000002 MOV TSSR(R5),R1      ;CONTENTS OF TSSR REGISTER
4685 051322 005237 002212 INC FATFLG          ;BUMP COUNT
4689 051326 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      051326 104455 TRAP C$ERRDF
      051330 001030 .WORD 536
      051332 003650 .WORD SFIERR
      051334 012124 .WORD SFIMSG
4690 051336 013737 002172 053700 20$: MOV UNITN,T25DSW      ;SET UP UNIT NUMBER
4691 051336 013737 002172 053700 MOV #T25PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
4692 051344 012704 053660
4693 ;
4694 ;*****
4695 ;
4696 ; WRITE CHARACTERISTICS COMMAND (CALL TO WRITCHR)
4697 ;
  
```

TEST 5: SPACE RECORDS

```

4698
4699
4700 051350 004737 010752      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
4701 051354 103407      BCS      25$           ;BR, IF COMMAND ISSUED OK
4702 051356 005237 002212      INC      FATFLG        ;BUMP COUNT
4706 051362 010001      MOV      R0,R1         ;SAVE CONTENTS OF TSSR
4707 051364      ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP      C$ERHRD
                                .WORD     537
                                .WORD     WRTMSG
                                .WORD     SFIMSG
                                051364 104456
                                051366 001031
                                051370 005054
                                051372 012124
4708 051374      25$:   CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                051374 104406
4709
4710
4711
4712
4713
4714
4715
4716 051376 004737 011104      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
4717 051402 103407      BCS      30$           ;BR, IF NO PROBLEM
4718 051404 010001      MOV      R0,R1         ;SAVE TSSR
4719 051406 005237 002212      INC      FATFLG        ;BUMP COUNT
4723 051412      ERRHRD  ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     538
                                .WORD     T25RWN
                                .WORD     PKTSSR
                                051412 104456
                                051414 001032
                                051416 055005
                                051420 012136
4724 051422      30$:   CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                051422 104406
4725 051424 013701 054026      MOV      T25CN2,R1     ;NUMBER OF RECORDS ON TAPE
4726 051430 012702 177776      MOV      @65534.,R2    ;MAX IT CAN SPACE OVER
4727 051434 020201      CMP      R2,R1         ;WHICH VALUE CAN WE USE
4728 051436 003002      BGT      46$           ;BR, IF @ WRITTEN > 64K
4729 051440 010103      MOV      R1,R3         ;@ WRITTEN CAN BE USED
4730 051442 000401      BR       47$           ;MOVE ON
4731 051444 010203      46$:   MOV      R2,R3     ;USE MAX NUMBER
4732 051446 162703 000001      47$:   SUB      @1,R3    ;DON'T GO ALL THE WAY YET
4733 051452 010337 054002      MOV      R3,T25RB     ;NUMBER OF RECORDS TO SPACE OVER
4734
4735
4736
4737
4738
4739
4740
4741 051456 012737 140010 054000      MOV      @140010,T25PK3 ;SPACE FORWARD,ACK,CVC=1 COMMAND
4742 051464 012704 054000      MOV      @T25PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
4743 051470
4744 051470 013737 054026 054032      65$:   MOV      T25CN2,T25DLY ;NUMBER OF RECORDS USED AS DELAY COUNTER
4745 051476 010465 000000      MOV      R4,T25DB(R5) ;ISSUE COMMAND
4746 051502 004737 016340      67$:   JSR      PC,WAITF  ;WAIT FOR SSR TO SET
4747 051506 016501 000002      MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
4748 051512 012702 000200      MOV      @SSR,R2     ;SET UP EXPECTED
4749 051516 020102      CMP      R1,R2        ;ARE THEY EQUAL
4750 051520 001425      BEQ      75$         ;BR, IF OK

```

TEST 5: SPACE RECORDS

```

4751 051522          DELAY 250          ;DELAY .25 SECONDS
      051522 012727 000250          MOV      #250,(PC)-
      051526 000000          .WORD 0
      051530 013727 002116          MOV      L$DLY,(PC)-
      051534 000000          .WORD 0
      051536 005367 177772          DEC      -6(PC)
      051542 001375          BNE     -4
      051544 005367 177756          DEC      -22(PC)
      051550 001367          BNE     -20
4752 051552 005337 054032          DEC      T25DLY          ;BUMP DOWN COUNTER
4753 051556 001351          BNE     67$          ;BR, IF NOT AT END OF DELAY
4754 051560 005237 002212          INC      FATFLG          ;BUMP COUNT
4758 051564          ERRHRD ERRNO,T25WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      051564 104456          TRAP   C$ERRRD
      051566 001033          .WORD 539
      051570 054115          .WORD T25WDE
      051572 012136          .WORD PKTSSR
4759 051574          75$: CKLOOP          ;LOOP IF SELECTED
      051574 104406          TRAP   C$CLP1
4760 051576 012703 010000          MOV      #4096,R3          ;RECORD SIZE
4761 051602 013737 003114 054002    MOV      FREE,T25RB        ;STARTING READ BUFFER ADDRESS
4762
4763          ;*****
4764          ;
4765          ;READ DATA,ACK COMMAND
4766          ;
4767          ;*****
4768
4769 051610 012737 100001 054000    MOV      #100001,T25PK3    ;READ DATA,ACK COMMAND
4770 051616 012704 054000    165$: MOV      #T25PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4771 051622 010337 054006    MOV      R3,T25SZ          ;SET UP RECORD SIZE IN PACKET
4772 051626 010465 000000    MOV      R4,TSD8(R5)       ;ISSUE COMMAND
4773 051632 004737 016340    JSR      PC,WAITF          ;WAIT FOR SSR TO SET
4774 051636 016501 000002    MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
4775 051642 012702 000200    MOV      #SSR,R2          ;SET UP EXPECTED
4776 051646 020102          CMP      R1,R2            ;ARE THEY EQUAL
4777 051650 001411          BEQ     170$             ;BR, IF OK
4778 051652 032701 000004    BIT     #BIT2,R1          ;CHECK FOR TAPE STATUS ALERT
4779 051656 001006          BNE     170$             ;IF SET ALL IS WELL
4780 051660 005237 002212    INC      FATFLG          ;BUMP COUNT
4784 051664          ERRHRD ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      051664 104456          TRAP   C$ERRRD
      051666 001034          .WORD 540
      051670 005204          .WORD RDERR
      051672 012136          .WORD PKTSSR
4785 051674          170$: CKLOOP          ;LOOP IF SELECTED
      051674 104406          TRAP   C$CLP1
4786 051676 017701 131212          MOV      #FREE,R1          ;GET FIRST WORD FROM BUFFER
4787 051702 013702 054026    MOV      T25CN2,R2        ;SET UP EXPECTED
4788 051706 162702 000001    SUB     #1,R2            ;SHOULD BE LAST RECORD
4789 051712 020102          CMP     R1,R2            ;WAS RECORD NUMBERED R3
4790 051714 001406          BEQ     200$             ;BR, IF CORRECT RECORD
4791 051716 005237 002212    INC      FATFLG          ;BUMP COUNT
4795 051722          ERRHRD ERRNO,T25WNG,EXPREC ;SHOULD HAVE BEEN RECORD NUMBER 1
      051722 104456          TRAP   C$ERRRD
      051724 001035          .WORD 541
      051726 054405          .WORD T25WNG

```



TEST 5: SPACE RECORDS

```

4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816 051752
      051752
      051752 104402
4817 051754 004737 055216      JSR      PC,T25REST      ;SET COMMAND PACKET
4818 051760 013737 054030 054026      MOV      T25CNT,T25CN2  ;SET UP RECORD COUNTER
4819 051766 004737 055310      JSR      PC,T25RT2      ;SET UP OTHER COMMAND PACKET
4820 051772 004737 055352      JSR      PC,T25RT3      ;SET UP OTHER COMMAND PACKET
4821 051776 013737 054030 054032      MOV      T25CNT,T25DLY  ;SET UP REWIND DELAY COUNTER
4822
4823
4824
4825
4826
4827
4828
4829 052004 004737 016064      10$:   JSR      PC,SOFINIT  ;DO INITIALIZE ON CONTROLLER
4830 052010 103427                BCS      20$             ;BR IF INIT WAS OK
4831 052012                DELAY    250             ;WAIT ABOUT .25 SECONDS
      052012 012727 000250                MOV      #250,(PC)+
      052016 000000                .WORD   0
      052020 013727 002116                MOV      L$DLY,(PC)+
      052024 000000                .WORD   0
      052026 005367 177772                DEC      -6(PC)
      052032 001375                BNE     .-4
      052034 005367 177756                DEC      -22(PC)
      052040 001367                BNE     .-20
4832 052042 005337 054032      DEC      T25DLY        ;DEC COUNTER
4833 052046 001356                BNE     10$            ;BR, IF MORE LOOPS REQUIRED
4834 052050 016501 000002      MOV      TSSR(R5),R1    ;CONTENTS OF TSSR REGISTER
4835 052054 005237 002212      INC      FATFLG        ;BUMP COUNT
4839 052060                ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      052060 104455                TRAP    C$ERRDF
      052062 001036                .WORD   54?
      052064 003650                .WORD   SFIERR
      052066 012124                .WORD   SFIMSG
4840 052070 013737 002172 053700 20$:   MOV      UNITN,T25DSW    ;SET UP UNIT NUMBER
4841
4842 052076 012704 053660                MOV      #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4843
4844
4845
4846
4847
4848
4849
;*****
;
;WRITE CHARACTERISTICS COMMAND (CALL TO WRITCHR)
;
;*****
;
;*****
;*****

```

## TEST 5: SPACE RECORDS

```

4850 052102 004737 010752      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
4851 052106 103407              BCS      25$           ;BR, IF COMMAND ISSUED OK
4852 052110 005237 002212      INC      FATFLG        ;BUMP COUNT
4856 052114 010001              MOV      R0,R1         ;SAVE CONTENTS OF TSSR
4857 052116              ERRHRD   ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP      C$ERHRD
                                .WORD    543
                                .WORD    WRTMSG
                                .WORD    SFIMSG
                                052116 104456
                                052120 001037
                                052122 005054
                                052124 012124
4858 052126              25$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                052126 104406
4859
4860 ;*****
4861 ;
4862 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
4863 ;
4864 ;*****
4865
4866 052130 004737 011104      JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
4867 052134 103407              BCS      30$           ;BR, IF NO PROBLEM
4868 052136 010001              MOV      R0,R1         ;SAVE TSSR
4869 052140 005237 002212      INC      FATFLG        ;BUMP COUNT
4873 052144              ERRHRD   ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    544
                                .WORD    T25RWN
                                .WORD    PKTSSR
                                052144 104456
                                052146 001040
                                052150 055005
                                052152 012136
4874 052154              30$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                052154 104406
4875
4876 ;*****
4877 ;
4878 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
4879 ;
4880 ;*****
4881
4882 052156 013701 053710      MOV      T25BFR+6,R1   ;PICK UP XST0
4883 052162 010102              MOV      R1,R2         ;SET UP EXPECTED
4884 052164 052702 000002      BIS      @BIT1,R2      ;SET BOT BIT IN EXPECTED
4885 052170 020102              CMP      R1,R2         ;DOES EXP = REC'D
4886 052172 001406              BEQ      40$           ;BR, IF EQUAL (OK)
4887 052174 005237 002212      INC      FATFLG        ;BUMP COUNT
4891 052200              ERRHRD   ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    545
                                .WORD    T25BOT
                                .WORD    EXPREC
                                052200 104456
                                052202 001041
                                052204 054175
                                052206 015564
4892 052210              40$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                052210 104406
4893 052212 013701 054026      MOV      T25CN2,R1     ;NUMBER OF RECORDS ON TAPE
4894 052216 012702 177776      MOV      @65534.,R2    ;MAX IT CAN SPACE OVER
4895 052222 020201              CMP      R2,R1         ;WHICH VALUE CAN WE USE
4896 052224 003002              BGT      46$           ;BR, IF # WRITTEN > 64K
4897 052226 010103              MOV      R1,R3         ;# WRITTEN CAN BE USED
4898 052230 000401              BR       47$           ;MOVE ON
4899 052232 010203              46$:   MOV      R2,R3   ;USE MAX NUMBER
4900 052234              47$:

```



M L D

TEST 5: SPACE RECORDS

```

4901 052234 010337 054002          MOV      R3,T25RB          ;NUMBER OF RECORDS TO SPACE OVER
4902
4903          ;*****
4904          ;
4905          ;SPACE FORWARD,ACK,CVC=1 COMMAND
4906          ;
4907          ;*****
4908
4909 052240 012737 140010 054000    MOV      #140010,T25PK3    ;SPACE FORWARD,ACK,CVC=1 COMMAND
4910 052246 012704 054000          MOV      #T25PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4911 052252 010465 000000          MOV      R4,T25DB(R5)    ;ISSUE COMMAND
4912 052256 013737 054026 054032    MOV      T25CN2,T25DLY   ;SET UP DELAY COUNTER
4913 052264 004737 016340 48$:    JSR      PC,WAIT         ;WAIT FOR SSR TO SET
4914 052270 016501 000002          MOV      T25SR(R5),R1    ;GET T25SR CONTENTS
4915 052274 012702 000200          MOV      #25SR,R2       ;SET UP EXPECTED
4916 052300 020102          CMP      R1,R2          ;ARE THEY EQUAL
4917 052302 001425          BEQ      50$            ;BR, IF OK
4918 052304          DELAY    250           ;WAIT .25 SECONDS
                                MOV      #250,(PC)
                                .WORD    0
                                MOV      L$DLY,(PC)
                                .WORD    0
                                DEC      -6(PC)
                                BNE      .-4
                                DEC      -22(PC)
                                BNE      .-20
4919 052334 005337 054032          DFC      T25DLY          ;DEC THE DELAY COUNTER
4920 052340 001351          BNE      48$            ;BR, IF COUNTER HASN'T EXPIRED
4921 052342 005237 002212          INC      FATFLG         ;BUMP COUNT
4925 052346          ERRHRD  ERRNO,T25WDE,EXPREC ;T25SR INCORRECT AFTER READ DATA
                                TRAP    C$ERRHRD
                                .WORD    546
                                .WORD    T25WDE
                                .WORD    EXPREC
4926 052356 104456 50$:    CKLOOP
                                TRAP    C$CLP1
4927 052360 013701 054026          MOV      T25CN2,R1      ;NUMBER OF RECORDS ON TAPE
4928 052364 012702 177776          MOV      #65534.,R2    ;MAX IT CAN SPACE OVER
4929 052370 020201          CMP      R2,R1          ;WHICH VALUE CAN WE USE
4930 052372 003002          BGT      55$            ;BR, IF # WRITTEN > 64K
4931 052374 010103          MOV      R1,R3          ;# WRITTEN CAN BE USED
4932 052376 000401          BR       60$            ;MOVE ON
4933 052400 010203 55$:    MOV      R2,R3          ;USE MAX NUMBER
4934 052402 162703 000001 60$:    SUB      #1,R3          ;DON'T GO ALL THE WAY YET
4935 052406 010337 054002          MOV      R3,T25RB      ;NUMBER OF RECORDS TO SPACE OVER
4936
4937          ;*****
4938          ;
4939          ;SPACE REVERSE,ACK,CVC=1 COMMAND
4940          ;
4941          ;*****
4942
4943 052412 012737 140410 054000    MOV      #140410,T25PK3  ;SPACE REVERSE,ACK,CVC=1 COMMAND
4944 052420 012704 054000          MOV      #T25PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
4945 052424 010465 000000          MOV      R4,T25DB(R5)   ;ISSUE COMMAND
4946 052430 013737 054026 054032    MOV      T25CN2,T25DLY  ;SET UP COUNTER
4947 052436 004737 016340 70$:    JSR      PC,WAIT         ;WAIT FOR SSR TO SET

```

TEST 5: SPACE RECORDS

```

4948 052442 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
4949 052446 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
4950 052452 020102              CMP      R1,R2          ;ARE THEY EQUAL
4951 052454 001425              BEQ      75$            ;BR, IF OK
4952 052456              DELAY    250            ;WAIT ABOUT .25 SECONDS
                                MOV      #250,(PC)+
                                .WORD    0
                                MOV      T25DL,(PC)+
                                .WORD    0
                                DEC      -6(PC)
                                BNE     .-4
                                DEC      -22(PC)
                                BNE     .-20
4953 052506 005337 054032      DEC      T25DL          ;BUMP COUNTER DOWN
4954 052512 001351              BNE     70$            ;BR, IF COUNTER HASN'T EXPIRED
4955 052514 005237 002212      INC      FATFLG         ;BUMP COUNT
4959 052520              ERRHRD  ERRNO,T25WE,EXPREC ;TSSR INCORRECT AFTER READ DATA
                                TRAP    C$ERHRD
                                .WORD   547
                                .WORD   T25WE
                                .WORD   EXPREC
                                TRAP    C$CLP1
052520 104456
052522 001043
052524 054115
052526 015564
4960 052530              75$:   CKLOOP          ;LOOP IF SELECTED
052530 104406
4961 052532 012703 010000      MOV      #4096.,R3      ;RECORD SIZE
4962 052536 013737 003114 054002      MOV      FREE,T25RB     ;STARTING READ BUFFER ADDRESS
4963
4964 ;*****
4965 ;
4966 ;READ DATA,ACK COMMAND
4967 ;
4968 ;*****
4969
4970 052544 012737 100001 054000      MOV      #100001,T25PK3 ;READ DATA,ACK COMMAND
4971 052552 012704 054000      165$:  MOV      #T25PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
4972 052556 012700 177777      MOV      #177777,R0     ;SET ALL ONES INTO CORRECT REGISTER
4973 052562 004737 017512      JSR      PC,FILLMEM     ;FILL MEMORY WITH RECORD SIZE
4974 052566 010337 054006      MOV      R3,T25SZ       ;SET UP RECORD SIZE IN PACKET
4975 052572 010465 000000      MOV      R4,T5DB(R5)    ;ISSUE COMMAND
4976 052576 004737 016340      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
4977 052602 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
4978 052606 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
4979 052612 020102              CMP      R1,R2          ;ARE THEY EQUAL
4980 052614 001411              BEQ      170$           ;BR, IF OK
4981 052616 032701 000004      BIT      #BIT2,R1       ;CHECK FOR TAPE STATUS ALERT
4982 052622 001006              BNE     170$           ;BR, IF BIT SET
4983 052624 005237 002212      INC      FATFLG         ;BUMP COUNT
4987 052630              ERRHRD  ERRNO,RDERR,EXPREC ;TSSR INCORRECT AFTER READ DATA
                                TRAP    C$ERHRD
                                .WORD   548
                                .WORD   RDERR
                                .WORD   EXPREC
052630 104456
052632 001044
052634 005204
052636 015564
4988 052640              170$:  CKLOOP          ;LOOP IF SELECTED
052640 104406
4989 052642 017701 130246      MOV      @FREE,R1       ;GET FIRST WORD FROM BUFFER
4990 052646 012702 000001      MOV      #1,R2          ;SET UP EXPECTED
4991 052652 020102              CMP      R1,R2          ;WAS RECORD NUMBERED R3
4992 052654 001406              BEQ      200$           ;BR, IF CORRECT RECORD

```

TEST 5: SPACE RECORDS

```

4993 052656 005237 002212      INC      FATELG      ;BUMP COUNT
4997 052662      ERRHRD  ERRNO,T25WNH,EXPREC ;SHOULD HAVE BEEN RECORD NUMBER 1
      052662 104456                                     TRAP    C$ERHRD
      052664 001045                                     .WORD  549
      052666 054560                                     .WORD  T25WNH
      052670 015564                                     .WORD  EXPREC
4998 052672      200$:   CKL00P                                     TRAP    C$CLP1
      052672 104406                                     ;>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>>
      052674      ENDSUB                                       L10077:
      052674 104403                                     TRAP    C$ESUB
5000 052676 023727 002212 000017  CMP      FATELG,#15, ;IS ERROR COUNT AT 25
5001 052704 103402      RLO      999$          ;BR. IF LESS THAN 25
5002 052706 004737 017272      JSR      PC,CKDROP   ;TR? TO DROP THE UNIT
5003 052712      999$:

```

USER DOCUMENTATION

....B1  
....C1  
....D1  
....E1  
....F1  
....G1  
....H1  
....I1  
....J1  
....K1  
....L1  
....M1  
....N1

....B2  
....C2  
....D2  
....E2  
....F2  
....G2  
....H2  
....I2  
....J2  
....K2  
....L2  
....M2  
....N2

....B3  
....C3  
....D3  
....E3  
....F3  
....G3  
....H3  
....I3  
....J3  
....K3  
....L3  
....M3  
....N3

....B4  
....C4  
....D4  
....E4  
....F4  
....G4  
....H4  
....I4  
....J4  
....K4  
....L4  
....M4  
....N4

....B5  
....C5  
....D5  
....E5  
....F5  
....G5  
....H5  
....I5  
....J5  
....K5  
....L5  
....M5  
....N5

....B6  
....C6  
....D6  
....E6  
....F6  
....G6  
....H6  
....I6  
....J6  
....K6  
....L6  
....M6  
....N6

....B7  
....C7  
....D7  
....E7  
....F7  
....G7  
....H7  
....I7  
....J7  
....K7  
....L7  
....M7  
....N7

....B8  
....C8  
....D8  
....E8  
....F8  
....G8  
....H8  
....I8  
....J8  
....K8  
....L8  
....M8  
....N8

....B9  
....C9  
....D9  
....E9  
....F9  
....G9  
....H9  
....I9  
....J9  
....K9  
....L9  
....M9  
....N9

....B10  
....C10  
....D10  
....E10  
....F10  
....G10  
....H10  
....I10  
....J10  
....K10  
....L10  
....M10  
....N10

....B11  
....C11  
....D11  
....E11  
....F11  
....G11  
....H11  
....I11  
....J11  
....K11  
....L11  
....M11  
....N11

....B12  
....C12  
....D12  
....E12  
....F12  
....G12  
....H12  
....I12  
....J12  
....K12  
....L12  
....M12  
....N12

....B13  
....C13  
....D13  
....E13  
....F13  
....G13  
....H13  
....I13  
....J13  
....K13  
....L13  
....M13  
....N13

....B14  
....C14  
....D14  
....E14  
....F14  
....G14  
....H14  
....I14  
....J14  
....K14  
....L14  
....M14  
....N14

....B15  
....C15  
....D15  
....E15  
....F15  
....G15  
....H15  
....I15  
....J15  
....K15  
....L15  
....M15  
....N15

....B16  
....C16  
....D16  
....E16  
....F16  
....G16  
....H16  
....I16  
....J16  
....K16  
....L16  
....M16

TEST 5: SPACE RECORDS

```

5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016 052712
      052712
      052712 104402
5017 052714 004737 055216
5018 052720 004737 055310
5019 052724 004737 055352
5020
5021
5022
5023
5024
5025
5026
5027 052730 004737 016064
5028 052734 103407
5029 052736 005237 002212
5033 052742 010001
5034 052744
      052744 104455
      052746 001046
      052750 003650
      052752 012124
5035 052754 013737 002172 053700 204:
5036
5037 052762 012704 053660
5038
5039
5040
5041
5042
5043
5044
5045 052766 004737 010752
5046 052772 103407
5047 052774 005237 002212
5051 053000 010001
5052 053002
      053002 104456
      053004 001047
      053006 005054
      053010 012124
5053 053012
      053012 104406
5054
5055
5056

```

```

;
; TEST 5, SUBTEST 7
;
; VERIFIES THAT SPACE RECORDS REVERSE ISSUED WHILE THE
; TAPE IS AT BOT RESULTS IN FUNCTION REJECT TERMINATION
; WITH THE NONEXECUTABLE FUNCTION (NEF) ERROR BIT SET.
;
;
;
; ***** BEGIN SUBTEST *****
; T5.7:
; TRAP C$B5UB
; JSR PC,T25RE1T ;SET COMMAND PACKET
; JSR PC,T25R12 ;SET UP OTHER COMMAND PACKET
; JSR PC,T25R13 ;SET UP OTHER COMMAND PACKET
;
; *****
; ISSUE CONTROLLER "SOFT" INITIALIZE CARRY BIT CLEAR IF ERROR
;
; *****
; JSR PC,SOFTINIT ;DO INITIALIZE ON CONTROLLER
; BCS 204 ;BR IF INIT WAS OK
; INC FATELG ;BUMP COUNT
; MOV R0,R1 ;CONTENTS OF TSSR REGISTER
; ERKRD ERRNO,SFERR,SFMSG ;FATAL ERROR TSSR WAS NOT OK
; TRAP C$ERRD
; .WORD 550
; .WORD SFERR
; .WORD SFMSG
;
; 204: MOV UNITN,T25DSW ;SET UP UNIT NUMBER
;
; MOV @T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
;
; *****
; WRITE CHARACTERISTICS COMMAND (CALL TO WRCHR)
;
; *****
; JSR PC,WRCHR ;ISSUE WRITE CHARACTERISTICS
; BCS 254 ;BR, IF COMMAND ISSUED OK
; INC FATELG ;BUMP COUNT
; MOV R0,R1 ;SAVE CONTENTS OF TSSR
; ERKRD ERRNO,WRMSG,SFMSG ;WRITE CHARACTERISTICS FAILED
; TRAP C$ERRD
; .WORD 551
; .WORD WRMSG
; .WORD SFMSG
;
; 254: CLOOP ;LOOP IF SELECTED
; TRAP C$CLPL
;
; *****
;

```

TEST 6: REREADS

```

5627 056122 004737 010556 JSR PC,SPACE ;CALL SPACE ROUTINE
5628 056126 103412 BCS 150$ ;BR, IF NO PROBLEM WITH SPACE COMMAND
5629 056130 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
5630 056134 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
5631 056140 005237 002212 INC FATFLG ;BUMP COUNT
5635 056144 ERRHRD ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
      056144 104455 TRAP C$ERRHRD
      056146 001140 .WORD 608
      056150 072417 .WORD T26SC
      056152 015564 .WORD EXPREC
5636 056154 150$: CKLOOP TRAP C$CLP1
      056154 104406
5637 056156 013703 072002 MOV T26RSZ,R3 ;RECORD SIZE
5638 056162 013737 003114 071752 MOV FREE,T26RB ;STARTING READ BUFFER ADDRESS
5639
5640 ;*****
5641 ;REREREAD DATA,CVC=1,ACK COMMAND
5642 ;
5643 ;*****
5644
5645
5646 056170 012737 141001 071750 165$: MOV #141001,T26PK3 ;REREREAD DATA,CVC=1,ACK COMMAND
5647 056176 012704 071750 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
5648 056202 010337 071756 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
5649 056206 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
5650 056212 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
5651 056216 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
5652 056222 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
5653 056226 020102 CMP R1,R2 ;ARE THEY EQUAL
5654 056230 001406 BEQ 170$ ;BR, IF OK
5655 056232 005237 002212 INC FATFLG ;BUMP COUNT
5659 056236 ERRHRD ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      056236 104456 TRAP C$ERRHRD
      056240 001141 .WORD 609
      056242 073640 .WORD T26WDC
      056244 012136 .WORD PKTSSR
5660 056246 170$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      056246 104406
5661 056250 013702 003114 MOV FREE,R2 ;CURRENT BUFFER ADDRESS TO R2
5662 056254 010304 MOV R3,R4 ;CURRENT RECORD SIZE
5663 056256 162704 000400 SUB #256.,R4 ;FIRST LOCATION IN BUFFER
5664 056262 060204 173$: ADD R2,R4 ;SET UP POINTER
5665 056264 021403 CMP (R4),R3 ;CHECK DATA READ (R3=DATA ALSO)
5666 056266 001410 BEQ 180$ ;BR, IF ALL IS WELL
5667 056270 011401 MOV (R4),R1 ;RECD DATA
5668 056272 010302 MOV R3,R2 ;EXPECTED DATA
5669 056274 005237 002212 INC FATFLG ;BUMP COUNT
5673 056300 ERRHRD ERRNO,T26DTA,EXPREC ;DATA READ NOT = WRITTEN
      056300 104456 TRAP C$ERRHRD
      056302 001142 .WORD 610
      056304 073062 .WORD T26DTA
      056306 015564 .WORD EXPREC
5674 056310 180$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      056310 104406
5675 056312 005724 TST (R4), ;BUMP TO NEXT LOCATION
5676 056314 160204 SUB R2,R4 ;CORRECT RECORDS SIZE VALUE
5677 056316 020403 CMP R4,R3 ;END OF RECORD YET

```

TEST 5: SPACE RECORDS

```

5110 053154 01213e          759:  CKLOOP          LOOP IF SELECTED      .WORD  PKT5CR
      053156 104406          TRAP          C$CLP1
5111
5112 ;*****
5113 ;
5114 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5115 ;
5116 ;*****
5117
5118 053160 013701 053710      MOV     T25NER*6,R1      ;GET XSTO STATUS WORD
5119 053164 010102          MOV     R1,R2           ;SET UP EXPECTED
5120 053166 052702 002000      HIS     #BIT10,R2      ;SET THE NEF BIT
5121 053172 020102          CMP     R1,R2           ;ARE THEY EQUAL
5122 053174 001406          BEQ    170$           ;BR, IF EQUAL (GOOD)
5123 053176 005237 002212      INC     FATELG          ;BUMP COUNT
5127 053202          ERRHRD  ERRNO,T25NEF,EXPREC ;NEF SHOULD BE SET
      053202 104456          TRAP          C$ERRRD
      053204 001053          .WORD        555
      053206 054643          .WORD        T25NEF
      053210 015564          .WORD        EXPREC
5128 053212          170$:  CKLOOP
      053212 104406          TRAP          C$CLP1
5129 053214          ENDSUB
      053214          L10100:
5130 053216 023727 002212 000017  CMP     FATELG,#15,     ;IS ERROR COUNT AT 25
5131 053224 103402          BLO    999$           ;BR, IF LESS THAN 25
5132 053226 004737 017272      JSR    PC,CKDROP      ;TRY TO DROP THE UNIT
5133 053232          999$:

```





TEST 6: REREADS

```

5742 ;*****
5743 ;
5744 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5745 ;
5746 ;*****
5747
5748 056460 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
5749 056464 103413 BCS 30$ ;BR, IF NO PROBLEM
5750 056466 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
5751 056472 012702 000200 MOV #TSSR,R2 ;SET UP EXPECTED TSSR
5752 056476 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
5753 056500 005237 002212 INC FATFLG ;BUMP COUNT
5757 056504 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERHRD
; .WORD 613
; .WORD T26RWN
; .WORD PKTSSR
5758 056514 30$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
; .WORD 056514 104406
5759 ;*****
5760 ;
5761 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5762 ;
5763 ;*****
5764 ;
5765 ;*****
5766 056516 013701 071660 MOV T26BFR+6,R1 ;PICK UP XSTO
5767 056522 010102 MOV R1,R2 ;SET UP EXPECTED
5768 056524 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
5769 056530 020102 CMP R1,R2 ;DOES EXP = REC'D
5770 056532 001406 BEQ 40$ ;BR, IF EQUAL (OK)
5771 056534 005237 002212 INC FATFLG ;BUMP COUNT
5775 056540 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
; TRAP C$ERHRD
; .WORD 614
; .WORD T26BOT
; .WORD EXPREC
5776 056550 40$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
; .WORD 056550 104406
5777 056552 012703 000400 MOV #256.,R3 ;RECORD SIZE
5778 056556 013737 003114 071752 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
5779 ;*****
5780 ;
5781 ;WRITE DATA,ACK,SWB COMMAND
5782 ;
5783 ;*****
5784 ;
5785 ;*****
5786 056564 012737 110005 071750 MOV #110005,T26PK3 ;WRITE DATA,ACK,SWC COMMAND
5787 056572 012704 071750 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
5788 056576 65$:
5789 056576 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
5790 056600 004737 017512 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
5791 056604 010337 071756 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
5792 056610 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
5793 056614 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
5794 056620 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS

```



TEST 6: REREADS

```

5845 056756 012737 000400 072002      MOV      #256.,T26RSZ      ;SET UP RECORD SIZE
5846
5847      ;*****
5848      ;
5849      ;ISSUE SPACE RECORDS COMMAND  VALUE IN R3 SETS NUMBER OF RECORDS
5850      ;BIT 15 SETS DIRECTION  0=FORWARD  1=REVERSE
5851      ;
5852      ;*****
5853
5854 056764 012703 000001      145$:  MOV      #1,R3          ;SPACE ONE RECORD PARAMETER
5855 056770 004737 010556      JSR      PC,SPACE        ;CALL SPACE ROUTINE
5856 056774 103412      BCS     150$            ;BR, IF NO PROBLEM WITH SPACE COMMAND
5857 056776 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR
5858 057002 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED TSSR
5859 057006 005237 002212      INC      FATFLG         ;BUMP COUNT
5863 057012      ERRHRD  ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
      057012 104456      TRAP    C$ERRHRD
      057014 001152      .WORD  618
      057016 072417      .WORD  T26SC
      057020 015564      .WORD  EXPREC
5864 057022      150$:  CKLOOP                                TRAP    C$CLP1
      057022 104406
5865 057024 013703 072002      MOV      T26RSZ,R3      ;RECORD SIZE
5866 057030 013737 003114 071752      MOV      FREE,T26RB     ;STARTING READ BUFFER ADDRESS
5867
5868      ;*****
5869      ;
5870      ;REREAD DATA,CVC=1,ACK,SWB COMMAND
5871      ;
5872      ;*****
5873
5874 057036 012737 151001 071750      165$:  MOV      #151001,T26PK3 ;REREAD DATA,CVC=1,ACK,SWB COMMAND
5875 057044 012704 071750      MOV      #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
5876 057050 010337 071756      MOV      R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
5877 057054 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
5878 057060 004737 016340      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
5879 057064 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
5880 057070 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED
5881 057074 020102      CMP      R1,R2          ;ARE THEY EQUAL
5882 057076 001406      BEQ     170$            ;BR, IF OK
5883 057100 005237 002212      INC      FATFLG         ;BUMP COUNT
5887 057104      ERRHRD  ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      057104 104456      TRAP    C$ERRHRD
      057106 001153      .WORD  619
      057110 073640      .WORD  T26WDC
      057112 012136      .WORD  PKTSSR
5888 057114      170$:  CKLOOP                                ;LOOP IF SELECTED
      057114 104406      TRAP    C$CLP1
5889 057116 013702 003114      MOV      FREE,R2        ;CURRENT BUFFER ADDRESS TO R2
5890 057122 010304      MOV      R3,R4          ;CURRENT RECORD SIZE
5891 057124 162704 000400      SUB      #256.,R4       ;FIRST LOCATION IN BUFFER
5892 057130 060204      173$:  ADD      R2,R4          ;SET UP POINTER
5893 057132 021403      CMP      (R4),R3        ;CHECK DATA READ (R3=DATA ALSO)
5894 057134 001410      BEQ     180$            ;BR, IF ALL IS WELL
5895 057136 011401      MOV      (R4),R1        ;RECD DATA
5896 057140 010302      MOV      R3,R2          ;EXPECTED DATA
5897 057142 005237 002212      INC      FATFLG         ;BUMP COUNT

```

TEST 6: REREADS

```

5901 057146          ERRHRD ERRNO,T26DTA,EXPREC   ;DATA READ NOT = WRITTEN
      057146 104456
      057150 001154          TRAP   C$ERHRD
      057152 073062          .WORD 620
      057154 015564          .WORD T26DTA
                             .WORD EXPREC
5902 057156          180$:   CKLOOP              ;LOOP IF SELECTED
      057156 104406          TRAP   C$CLP1
5903 057160 005724          IST   (R4)+          ;BUMP TO NEXT LOCATION
5904 057162 160204          SUB   R2,R4          ;CORRECT RECORDS SIZE VALUE
5905 057164 020403          CMP   R4,R3          ;END OF RECORD YET
5906 057166 001360          BNE   173$          ;BR, IF NOT AT END OF RECORD
5907 057170 005723          IST   (R3)+          ;BUMP RECORD SIZE
5908 057172 010337 072002   MOV   R3,T26RSZ      ;STORE RECORD SIZE
5909 057176 022703 000412   CMP   #266.,R3      ;END OF RECORD YET
5910 057202 001270          BNE   145$          ;BR, IF MORE RECORDS TO READ
5911 057204          190$:   CKLOOP              ;LOOP IF SELECTED
      057204 104406          TRAP   C$CLP1
5912 057206          ENDSUB                      ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>
      057206          L10104:
5913 057210 023727 002212 000017   CMP   FATFLG,#15.   ;IS ERROR COUNT AT 25
5914 057216 103402          BLO   999$          ;BR, IF LESS THAN 25
5915 057220 004737 017272          JSR   PC,CKDROP     ;TRY TO DROP THE UNIT
5916 057224          999$:

```



TEST 6: REREADS

```

5972
5973 057300 012704 071630          MOV      #T26PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
5974
5975          ;*****
5976          ;
5977          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPCHR)
5978          ;
5979          ;*****
5980
5981 057304 004737 010752          JSR      PC,WRTPCHR          ;ISSUE WRITE CHARACTERISTICS
5982 057310 103407          BCS     26$                  ;BR, IF COMMAND ISSUED OK
5983 057312 005237 002212          INC     FATFLG              ;BUMP COUNT
5987 057316 010001          MOV     R0,R1              ;SAVE CONTENTS OF TSSR
5988 057320          ERRHRD  ERRNO,WRTPMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
          057320 104456          TRAP   C$ERHRD
          057322 001156          .WORD 622
          057324 005054          .WORD WRTPMSG
          057326 012124          .WORD SFIMSG
5989 057330          26$:   CKLOOP              ;LOOP IF SELECTED
          057330 104406          TRAP   C$CLP1
5990
5991          ;*****
5992          ;
5993          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5994          ;
5995          ;*****
5996
5997 057332 004737 011104          JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
5998 057336 103413          BCS     30$                  ;BR, IF NO PROBLEM
5999 057340 016501 000002          MOV     TSSR(R5),R1        ;GET TSSR
6000 057344 012702 000200          MOV     #SSR,R2           ;SET UP EXPECTED TSSR
6001 057350 010004          MOV     R0,R4             ;PACKET ADDRESS SET UP
6002 057352 005237 002212          INC     FATFLG              ;BUMP COUNT
6006 057356          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
          057356 104456          TRAP   C$ERHRD
          057360 001157          .WORD 623
          057362 073304          .WORD T26RWN
          057364 012136          .WORD PKTSSR
6007 057366          30$:   CKLOOP              ;LOOP IF SELECTED
          057366 104406          TRAP   C$CLP1
6008
6009          ;*****
6010          ;
6011          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6012          ;
6013          ;*****
6014
6015 057370 013701 071660          MOV     T26BFR+6,R1        ;PICK UP XSTO
6016 057374 010102          MOV     R1,R2             ;SET UP EXPECTED
6017 057376 052702 000002          BIS     @BIT1,R2          ;SET BOT BIT IN EXPECTED
6018 057402 020102          CMP     R1,R2             ;DOES EXP = REC'D
6019 057404 001406          BEQ    40$                ;BR, IF EQUAL (OK)
6020 057406 005237 002212          INC     FATFLG              ;BUMP COUNT
6024 057412          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          057412 104456          TRAP   C$ERHRD
          057414 001160          .WORD 624
          057416 073015          .WORD T26BOT

```

TEST 6: REREADS

```

057420 015564
6025 057422          40$:   CKLOOP                ;LOOP IF SELECTED                .WORD   EXPRFC
      057422 104406          ;                                     TRAP    C$CLP1
6026 057424 012703 000400      MOV    $256.,R3                ;RECORD SIZE
6027 057430 013737 003114 071752  MOV    FREE,T26RB            ;STARTING WRITE BUFFER ADDRESS
6028
6029          ;*****
6030          ;
6031          ;WRITE DATA,CVC=1,ACK COMMAND
6032          ;
6033          ;*****
6034
6035 057436 012737 140005 071750      MOV    $140005.T26PK3        ;WRITE DATA.CVC=1,ACK COMMAND
6036 057444 012704 071750          MOV    $T26PK3,R4           ;SET UP R4 WITH PACKET ADDRESS
6037 057450          65$:
6038 057450 010300          MOV    R3,R0                ;SET PATTERN IN CORRECT REGISTER
6039 057452 004737 017512          JSR    PC,FILLMEM           ;FILL MEMORY WITH RECORD SIZE
6040 057456 010337 071756          MOV    R3,T26SZ            ;SET UP RECORD SIZE IN PACKET
6041 057462 013777 071776 123424      MOV    T26CNT,$FREE        ;MOVE TAPE RECORD NUMBER TO BUFFER
6042 057470 062737 000001 071776      ADD    $1,T26CNT           ;NUMBER READY FOR NEXT RECORD
6043 057476 010465 000000          MOV    R4,T26B(R5)         ;ISSUE COMMAND
6044 057502 004737 016340          JSR    PC,WAITF            ;WAIT FOR SSR TO SET
6045 057506 016501 000002          MOV    TSSR(R5),R1        ;GET TSSR CONTENTS
6046 057512 012702 000200          MOV    $SSR,R2            ;SET UP EXPECTED
6047 057516 020102          CMP    R1,R2              ;ARE THEY EQUAL
6048 057520 001406          BEQ   75$                 ;BR, IF OK
6049 057522 005237 002212          INC   FATFLG              ;BUMP COUNT
6053 057526          ERRHRD ERRNO,WRTErr,PKTSSR  ;TSSR INCORRECT AFTER WRITE DATA
      057526 104456          TRAP   C$ERRHRD
      057530 001161          .WORD 625
      057532 005111          .WORD WRTErr
      057534 012156          .WORD PKTSSR
6054 057536          75$:   CKLOOP                ;LOOP IF SELECTED                .WORD   TRAP    C$CLP1
      057536 104406          ;
6055 057540 005723          TST   (R3),                ;BUMP THE RECORD SIZE
6056 057542 022703 000414          CMP   $268.,R3            ;MAXIMUM SIZE YET
6057 057546 001401          BEQ   120$                ;BR, IF AT END OF WRITE SEQUENCE
6058 057550 000737          BR    65$                 ;WRITE MORE RECORDS
6059 057552          120$:
6060 057552 005037 071776          CLR   T26CNT              ;SET RECORD COUNTER BACK TO ZERO
6061
6062          ;*****
6063          ;
6064          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6065          ;
6066          ;*****
6067
6068 057556 004737 011104          JSR   PC,REWIND            ;CALL TAPE REWIND COMMAND
6069 057562 103413          BCS   130$                ;BR, IF NO PROBLEM
6070 057564 016501 000002          MOV   TSSR(R5),R1        ;GET TSSR
6071 057570 012702 000200          MOV   $SSR,R2            ;SET UP EXPECTED TSSR
6072 057574 010004          MOV   R0,R4              ;PACKET ADDRESS SET UP
6073 057576 005237 002212          INC   FATFLG              ;BUMP COUNT
6077 057602          ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      057602 104456          TRAP   C$ERRHRD
      057604 001162          .WORD 626
      057606 073304          .WORD T26RWN

```

TEST 6: REREADS

```

057610 012136
6078 057612 130$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
057612 104406 TRAP C$CLP1
6079
6080 ;*****
6081 ;
6082 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6083 ;
6084 ;*****
6085
6086 057614 013701 071660 MOV T26BFR+6,R1 ;PICK UP XSTO
6087 057620 010102 MOV R1,R2 ;SET UP EXPECTED
6088 057622 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6089 057626 020102 CMP R1,R2 ;DOES EXP = REC'D
6090 057630 001406 BEQ 140$ ;BR, IF EQUAL (OK)
6091 057632 005237 002212 INC FATFLG ;BUMP COUNT
6095 057636 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
057636 104456 TRAP C$ERHRD
057640 001163 .WORD 627
057642 073015 .WORD T26BOT
057644 015564 .WORD EXPREC
6096 057646 140$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
057646 104406
6097
6098 ;*****
6099 ;
6100 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
6101 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
6102 ;
6103 ;*****
6104
6105 057650 012703 000001 MOV #1,R3 ;SPACE 1 RECORD FORWARD
6106 057654 004737 010556 JSR PC,SPACE ;SPACE CALL
6107 057660 012703 000400 MOV #256.,R3 ;RECORD SIZE
6108 057664 013737 003114 071752 150$: MOV FREE,T26RR ;STARTING READ BUFFER ADDRESS
6109
6110 ;*****
6111 ;
6112 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6113 ;
6114 ;*****
6115
6116 057672 012737 161001 071750 MOV #161001,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6117 057700 012704 071750 165$: MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6118 057704 010337 71756 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6119 057710 010465 000000 MOV R4,T26SDB(R5) ;ISSUE COMMAND
6120 057714 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
6121 057720 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6122 057724 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6123 057730 020102 CMP R1,R2 ;ARE THEY EQUAL
6124 057732 001406 BEQ 170$ ;BR, IF OK
6125 057734 005237 002212 INC FATFLG ;BUMP COUNT
6129 057740 ERRHRD ERRNO,T26RRG,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
057740 104456 TRAP C$ERHRD
057742 001164 .WORD 628
057744 072322 .WORD T26RRG
057746 012136 .WORD PKTSSR

```



TEST 6: REREADS

```

7373
7374 064264 012704 071630          MOV    #T26PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
7375
7376          ;*****
7377          ;
7378          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPCHR)
7379          ;
7380          ;*****
7381
7382 064270 004737 010752          JSR    PC,WRTPCHR          ;ISSUE WRITE CHARACTERISTICS
7383 064274 103407          BCS    26$                ;BR, IF COMMAND ISSUED OK
7384 064276 005237 002212          INC    FATFLG            ;BUMP COUNT
7388 064302 010001          MOV    R0,R1            ;SAVE CONTENTS OF TSSR
7389 064304          ERRHRD  ERRNO,WRTPMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
          064304 104456          TRAP   C$ERRHRD
          064306 001252          .WORD 682
          064310 005054          .WORD WRTPMSG
          064312 012124          .WORD SFIMSG
7390 064314          2c$:  CKLOOP          ;LOOP IF SELECTED
          064314 104406          TRAP   C$CLP1
7391
7392          ;*****
7393          ;
7394          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE.
7395          ;
7396          ;*****
7397
7398 064316 004737 011104          JSR    PC,REWIND          ;CALL TAPE REWIND COMMAND
7399 064322 103413          BCS    30$                ;BR, IF NO PROBLEM
7400 064324 016501 000002          MOV    TSSR(R5),R1       ;GET TSSR
7401 064330 012702 000200          MOV    #SSR,R2          ;SET UP EXPECTED TSSR
7402 064334 010004          MOV    R0,R4            ;PACKET ADDRESS SET UP
7403 064336 005237 002212          INC    FATFLG            ;BUMP COUNT
7407 064342          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
          064342 104456          TRAP   C$ERRHRD
          064344 001253          .WORD 683
          064346 073304          .WORD T26RWN
          064350 012136          .WORD PKTSSR
7408 064352          30$:  CKLOOP          ;LOOP IF SELECTED
          064352 104406          TRAP   C$CLP1
7409
7410          ;*****
7411          ;
7412          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
7413          ;
7414          ;*****
7415
7416 064354 013701 071660          MOV    T26BFR+6,R1       ;PICK UP XST0
7417 064360 010102          MOV    R1,R2            ;SET UP EXPECTED
7418 064362 052702 000002          BIS    #BIT1,R2          ;SET BOT BIT IN EXPECTED
7419 064366 020102          CMP    R1,R2            ;DOES EXP = REC'D
7420 064370 001406          BEQ   40$                ;BR, IF EQUAL (OK)
7421 064372 005237 002212          INC    FATFLG            ;BUMP COUNT
7425 064376          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          064376 104456          TRAP   C$ERRHRD
          064400 001254          .WORD 684
          064402 073015          .WORD T26BOT

```

TEST 6: REREADS

```

056002 104406
5576 056004 005723          ;BUMP RECORD SIZE          TRAP  C$CLP1
5577 056006 022703 000414    ;END OF RECORD SET
5578 056012 001346          ;BR, IF MORE RECORDS TO WRITE
5579 056014 104406          ;LOOP IF SELECTED
                                TRAP  C$CLP1
5580 056016 104406
5581
5582
5583
5584
5585
5586
5587
5588 056016 004737 011104      JSR    PC,REWIND          ;CALL TAPE REWIND COMMAND
5589 056022 103413          BCS    130$              ;BR, IF NO PROBLEM
5590 056024 016501 000002     MOV    TSSR(R5),R1       ;GET TSSR
5591 056030 012702 000200     MOV    #SSR,R2          ;SET UP EXPECTED TSSR
5592 056034 010004          MOV    R0,R4            ;PACKET ADDRESS SET UP
5593 056036 005237 002212     INC    FATFLG           ;BUMP COUNT
5597 056042 104456          ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP  C$ERHRD
                                .WORD 606
                                .WORD T26RWN
                                .WORD PKTSSR
5598 056052 130$: CKLOOP      ;LOOP IF SELECTED
056052 104406          TRAP  C$CLP1
5599
5600
5601
5602
5603
5604
5605
5606 056054 013701 071660     MOV    T26BFR+6,R1      ;PICK UP XST0
5607 056060 010102          MOV    R1,R2            ;SET UP EXPECTED
5608 056062 052702 000002     BIS    #BIT1,R2         ;SET BOT BIT IN EXPECTED
5609 056066 020102          CMP    R1,R2            ;DOES EXP = REC'D
5610 056070 001406          BEQ    140$             ;BR, IF EQUAL (OK)
5611 056072 005237 002212     INC    FATFLG           ;BUMP COUNT
5615 056076 104456          ERRHRD ERRNO,T26BOT,PKTSSR ;TAPE NOT AT BOT AFTER REWIND
                                TRAP  C$ERHRD
                                .WORD 607
                                .WORD T26BOT
                                .WORD PKTSSR
5616 056106 104406          TRAP  C$CLP1
5617 056110 012737 000400 072002 MOV    #256.,T26RSZ     ;SET RECORD SIZE
5618
5619
5620
5621
5622
5623
5624
5625
5626 056116 012703 000001     145$: MOV    #1,R3      ;SPACE ONE RECORD PARAMETER

```

TEST 6: REREADS

```

5627 056122 004737 010556      JSR      PC,SPACE      ;CALL SPACE ROUTINE
5628 056126 103412              BCL      150$          ;BR, IF NO PROBLEM WITH SPACE COMMAND
5629 056130 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR
5630 056134 012702 000200      MOV      QSSR,R2      ;SET UP EXPECTED TSSR
5631 056140 005237 002212      INC      FATFLG       ;BUMP COUNT
5635 056144              ERRHRD  ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
                    056144 104456              TRAP      C$ERRHRD
                    056146 001140              .WORD    608
                    056150 072417              .WORD    T26SC
                    056152 015564              .WORD    EXPREC
5636 056154              150$:  CKLOOP
                    056154 104406              TRAP      C$CLP1
5637 056156 013703 072002      MOV      T26RSZ,R3    ;RECORD SIZE
5638 056162 013737 003114 071752  MOV      FREE,T26RB   ;STARTING READ BUFFER ADDRESS
5639
5640 ;*****
5641 ;
5642 ;REREREAD DATA,CVC=1,ACK COMMAND
5643 ;
5644 ;*****
5645
5646 056170 012737 141001 071750      MOV      Q141001,T26PK3 ;REREREAD DATA,CVC=1,ACK COMMAND
5647 056176 012704 071750      165$:  MOV      QT26PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
5648 056202 010337 071756      MOV      R3,T26SZ     ;SET UP RECORD SIZE IN PACKET
5649 056206 010465 000000      MOV      R4,TSD8(R5)  ;ISSUE COMMAND
5650 056212 004737 016340      JSR      PC,WAITF     ;WAIT FOR SSR TO SET
5651 056216 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
5652 056222 012702 000200      MOV      QSSR,R2     ;SET UP EXPECTED
5653 056226 020102              CMP      R1,R2        ;ARE THEY EQUAL
5654 056230 001406              BEQ      170$         ;BR, IF OK
5655 056232 005237 002212      INC      FATFLG       ;BUMP COUNT
5659 056236              ERRHRD  ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                    056236 104456              TRAP      C$ERRHRD
                    056240 001141              .WORD    609
                    056242 073640              .WORD    T26WDC
                    056244 012136              .WORD    PKTSSR
5660 056246              170$:  CKLOOP          ;LOOP IF SELECTED
                    056246 104406              TRAP      C$CLP1
5661 056250 013702 003114      MOV      FREE,R2     ;CURRENT BUFFER ADDRESS TO R2
5662 056254 010304              MOV      R3,R4       ;CURRENT RECORD SIZE
5663 056256 162704 000400      SUB      Q256.,R4    ;FIRST LOCATION IN BUFFER
5664 056262 060204              173$:  ADD      R2,R4    ;SET UP POINTER
5665 056264 021403              CMP      (R4),R3     ;CHECK DATA READ (R3=DATA ALSO)
5666 056266 001410              BEQ      180$         ;BR, IF ALL IS WELL
5667 056270 011401              MOV      (R4),R1     ;RECD DATA
5668 056272 010302              MOV      R3,R2      ;EXPECTED DATA
5669 056274 005237 002212      INC      FATFLG       ;BUMP COUNT
5673 056300              ERRHRD  ERRNO,T26DTA,EXPREC ;DATA READ NOT = WRITTEN
                    056300 104456              TRAP      C$ERRHRD
                    056302 001142              .WORD    610
                    056304 073062              .WORD    T26DTA
                    056306 015564              .WORD    EXPREC
5674 056310              180$:  CKLOOP          ;LOOP IF SELECTED
                    056310 104406              TRAP      C$CLP1
5675 056312 005724              TST      (R4),       ;BUMP TO NEXT LOCATION
5676 056314 160204              SUB      R2,R4       ;CORRECT RECORDS SIZE VALUE
5677 056316 020403              CMP      R4,R3       ;END OF RECORD YET

```

( )

TEST 6: REREADS

```

5678 056320 001360
5679 056322 005723
5680 056324 010337 072002
5681 056330 022703 000412
5682 056334 001270
5683 056336
          190$: CKLOOP
          056336 104406
5684 056340
          056340
          056340 104403
5685 056342 023727 002212 000017
5686 056350 103402
5687 056352 004737 017272
5688 056356
          999$:

```

```

BNE 173$
TST (R3)+
MOV R3,126RSZ
CMP #266,R3
BNE 145$
ENDSUB
CMP #ATELG,015.
BLO 999$
JSR PC,CKDROP

```

```

;BR, IF NOT AT END OF RECORD
;BUMP RECORD SIZE
;RESET RECORD SIZE
;END OF RECORD TEST
;BR, IF MORE RECORDS TO READ
;LOOP IF SELECTED
          TRAP C$CLP1
;=====> END SUBTEST <=====>
          L1010$:
          TRAP C$ESUB
;IS ERROR COUNT AT 25
;BR, IF LESS THAN 25
;TRY TO DROP THE UNIT

```

TEST 6: REREADS

```

5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703 056356          BGNSUB                ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
      056356          T6.2:                TRAP      C$BSUB
      056356 104402
5704 056360 004737 074540      JSR      PC,T26REST          ;SET COMMAND PACKET
5705 056364 004737 074632      JSR      PC,T26RT2          ;SET UP OTHER COMMAND PACKET
5706 056370 004737 074674      JSR      PC,T26RT3          ;SET UP OTHER COMMAND PACKET
5707
5708
5709
5710
5711
5712
5713
5714 056374 004737 016064      JSR      PC,SOFINIT          ;DO INITIALIZE ON CONTROLLER
5715 056400 103407             BCS      26$                ;BR IF INIT WAS OK
5716 056402 005237 002212      INC      FATFLG             ;BUMP COUNT
5720 056406 010001             MOV      R0,R1              ;CONTENTS OF TSSR REGISTER
5721 056410             ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      056410 104455
      056412 001143             TRAP      C$ERDF
      056414 003650             .WORD    611
      056416 012124             .WORD    SFIERR
5722 056420 013737 002172 071650 20$:  MOV      UNITN,T26DSW          ;SET UP UNIT NUMBER
5723
5724 056426 012704 071630      MOV      @T26PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
5725
5726
5727
5728
5729
5730
5731
5732 056432 004737 010752      JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
5733 056436 103407             BCS      26$                ;BR, IF COMMAND ISSUED OK
5734 056440 005237 002212      INC      FATFLG             ;BUMP COUNT
5738 056444 010001             MOV      R0,R1              ;SAVE CONTENTS OF TSSR
5739 056446             ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
      056446 104456             TRAP      C$FRHRD
      056450 001144             .WORD    612
      056452 005054             .WORD    WRTMSG
      056454 012124             .WORD    SFIMSG
5740 056456             26$:  CKLOOP                ;LOOP IF SELECTED
      056456 104406             TRAP      C$CLP1
5741

```

TEST 6: REREADS

```

5742 ;*****
5743 ;
5744 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5745 ;
5746 ;*****
5747
5748 056460 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
5749 056464 103413 BCS 30$ ;BR, IF NO PROBLEM
5750 056466 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
5751 056472 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
5752 056476 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
5753 056500 005237 002212 INC FATFLG ;BUMP COUNT
5757 056504 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERHRD
; .WORD 613
; .WORD T26RWN
; .WORD PKTSSR
5758 056514 30$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
; .WORD
5759 ;*****
5760 ;
5761 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5762 ;
5763 ;*****
5764 ;
5765
5766 056516 013701 071660 MOV T26BFR+6,R1 ;PICK UP XSTO
5767 056522 010102 MOV R1,R2 ;SET UP EXPECTED
5768 056524 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
5769 056530 020102 CMP R1,R2 ;DOES EXP = REC'D
5770 056532 001406 BEQ 40$ ;BR, IF EQUAL (OK)
5771 056534 005237 002212 INC FATFLG ;BUMP COUNT
5775 056540 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
; TRAP C$ERHRD
; .WORD 614
; .WORD T26BOT
; .WORD EXPREC
5776 056550 40$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
; .WORD
5777 056552 012703 000400 MOV #256.,R3 ;RECORD SIZE
5778 056556 013737 003114 071752 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
5779 ;*****
5780 ;
5781 ;WRITE DATA,ACK,SWB COMMAND
5782 ;
5783 ;*****
5784 ;
5785
5786 056564 012737 110005 071750 MOV #110005,T26PK3 ;WRITE DATA,ACK,SWB COMMAND
5787 056572 012704 071750 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
5788 056576 65$:
5789 056576 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
5790 056600 004737 017512 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
5791 056604 010337 071756 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
5792 056610 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
5793 056614 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
5794 056620 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS

```

TEST 6: REREADS

```

5795 056624 012702 000200      MOV     #SSR,R2           ;SET UP EXPECTED
5796 056630 020102      CMP     R1,R2           ;ARE THEY EQUAL
5797 056632 001406      BEQ     75$             ;BR IF OK
5798 056634 005237 002212      INC     FATFLG          ;BUMP COUNT
5802 056640      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      056640 104456                                TRAP   C$ERRHRD
      056642 001147                                .WORD  615
      056644 005111                                .WORD  WRTErr
      056646 012136                                .WORD  PKTSSR
5803 056650      75$:  CKLOOP                ;LOOP IF SELECTED
      056650 104406                                TRAP   C$CLP1
5804 056652 005723      TST     (R3)+           ;BUMP RECORD SIZE
5805 056654 022703 000414      CMP     #268.,R3       ;END OF RECORD YET
5806 056660 001346      BNE     65$             ;BR, IF MORE RECORDS TO WRITE
5807 056662      80$:  CKLOOP                ;LOOP IF SELECTED
      056662 104406                                TRAP   C$CLP1
5808 056664      120$:
5809
5810      ;*****
5811      ;
5812      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5813      ;
5814      ;*****
5815
5816 056664 004737 011104      JSR     PC,REWIND       ;CALL TAPE REWIND COMMAND
5817 056670 103413      BCS     130$           ;BR, IF NO PROBLEM
5818 056672 016501 000002      MOV     TSSR(R5),R1    ;GET TSSR
5819 056676 012702 000200      MOV     #SSR,R2       ;SET UP EXPECTED TSSR
5820 056702 010004      MOV     RC,R4          ;PACKET ADDRESS SET UP
5821 056704 005237 002212      INC     FATFLG          ;BUMP COUNT
5825 056710      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      056710 104456                                TRAP   C$ERRHRD
      056712 001150                                .WORD  616
      056714 073304                                .WORD  T26RWN
      056716 012136                                .WORD  PKTSSR
5826 056720      130$:  CKLOOP                ;LOOP IF SELECTED
      056720 104406                                TRAP   C$CLP1
5827
5828      ;*****
5829      ;
5830      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5831      ;
5832      ;*****
5833
5834 056722 013701 071660      MOV     T26BFR+6,R1    ;PICK UP XSTO
5835 056726 010102      MOV     R1,R2          ;SET UP EXPECTED
5836 056730 052702 000002      BIS     #BIT1,R2       ;SET BOT BIT IN EXPECTED
5837 056734 020102      CMP     R1,R2          ;DOES EXP = REC'D
5838 056736 001406      BEQ     140$           ;BR, IF EQUAL (OK)
5839 056740 005237 002212      INC     FATFLG          ;BUMP COUNT
5843 056744      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      056744 104456                                TRAP   C$ERRHRD
      056746 001151                                .WORD  617
      056750 073015                                .WORD  T26BOT
      056752 015564                                .WORD  EXPREC
5844 056754      140$:  CKLOOP                ;LOOP IF SELECTED
      056754 104406                                TRAP   C$CLP1

```

TEST 6: REREADS

```

5845 056756 012737 000400 072002      MOV      #256.,T26RSZ      ;SET UP RECORD SIZE
5846
5847      ;*****
5848      ;
5849      ;ISSUE SPACE RECORDS COMMAND  VALUE IN R3 SETS NUMBER OF RECORDS
5850      ;BIT 15 SETS DIRECTION - 0=FORWARD  1=REVERSE
5851      ;
5852      ;*****
5853
5854 056764 012703 000001      145$:   MOV      #1,R3          ;SPACE ONE RECORD PARAMETER
5855 056770 004737 010556      JSR      PC,SPACE        ;CALL SPACE ROUTINE
5856 056774 103412                BCS      150$            ;BR, IF NO PROBLEM WITH SPACE COMMAND
5857 056776 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR
5858 057002 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED TSSR
5859 057006 005237 002212      INC      FATFLG          ;BUMP COUNT
5863 057012                ERRHRD   ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
      057012 104456                TRAP    C$ERRHRD
      057014 001152                .WORD  618
      057016 072417                .WORD  T26SC
      057020 015564                .WORD  EXPREC
5864 057022                150$:   CKLOOP                                TRAP    C$CLP1
      057022 104406
5865 057024 013703 072002      MOV      T26RSZ,R3      ;RECORD SIZE
5866 057030 013737 003114 071752      MOV      FREE,T26R8     ;STARTING READ BUFFER ADDRESS
5867
5868      ;*****
5869      ;
5870      ;REREAD DATA,CVC=1,ACK,SWB COMMAND
5871      ;
5872      ;*****
5873
5874 057036 012737 151001 071750      165$:   MOV      #151001,T26PK3 ;REREAD DATA,CVC=1,ACK,SWB COMMAND
5875 057044 012704 071750      MOV      #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
5876 057050 010337 071756      MOV      R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
5877 057054 010465 000000      MOV      R4,T26SDB(R5)  ;ISSUE COMMAND
5878 057060 004737 016340      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
5879 057064 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
5880 057070 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
5881 057074 020102                CMP      R1,R2          ;ARE THEY EQUAL
5882 057076 001406                BEQ      170$            ;BR, IF OK
5883 057100 005237 002212      INC      FATFLG          ;BUMP COUNT
5887 057104                ERRHRD   ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      057104 104456                TRAP    C$ERRHRD
      057106 001153                .WORD  619
      057110 073640                .WORD  T26WDC
      057112 012136                .WORD  PKTSSR
5888 057114                170$:   CKLOOP                                TRAP    C$CLP1
      057114 104406
5889 057116 013702 003114      MOV      FREE,R2        ;CURRENT BUFFER ADDRESS TO R2
5890 057122 010304      MOV      R3,R4          ;CURRENT RECORD SIZE
5891 057124 162704 000400      SUB      #256.,R4       ;FIRST LOCATION IN BUFFER
5892 057130 060204      173$:   ADD      R2,R4          ;SET UP POINTER
5893 057132 021403      CMP      (R4),R3        ;CHECK DATA READ (R3=DATA ALSO)
5894 057134 001410      BEQ      180$            ;BR, IF ALL IS WELL
5895 057136 011401      MOV      (R4),R1        ;RECD DATA
5896 057140 010302      MOV      R3,R2          ;EXPECTED DATA
5897 057142 005237 002212      INC      FATFLG          ;BUMP COUNT

```



TEST 6: REREADS

```

5901 057146          ERRHRD  ERRNO,T26DTA,EXPREC  ;DATA READ NOT = WRITTEN
      057146 104456
      057150 001154
      057152 073062
      057154 015564
5902 057156          180$:  CKLOOP                ;LOOP IF SELECTED
      057156 104406
      5903 057160 005724          TST          (R4)+      ;BUMP TO NEXT LOCATION
      5904 057162 160204          SUB          R2,R4      ;CORRECT RECORDS SIZE VALUE
      5905 057164 020403          CMP          R4,R3      ;END OF RECORD YET
      5906 057166 001360          BNE          173$      ;BR, IF NOT AT END OF RECORD
      5907 057170 005723          TST          (R3)+      ;BUMP RECORD SIZE
      5908 057172 010337 072002   MOV          R3,T26RSZ  ;STORE RECORD SIZE
      5909 057176 022703 000412   CMP          #266.,R3  ;END OF RECORD YET
      5910 057202 001270          BNE          145$      ;BR, IF MORE RECORDS TO READ
      5911 057204          190$:  CKLOOP                ;LOOP IF SELECTED
      057204 104406
      5912 057206          ENDSUB
      057206
      057206 104403
      5913 057210 023727 002212 000017  CMP          FATFLG,#15. ;IS ERROR COUNT AT 25
      5914 057216 103402          BLO          999$      ;BR, IF LESS THAN 25
      5915 057220 004737 017272          JSR          PC,CKDROP ;TRY TO DROP THE UNIT
      5916 057224          999$:

```

```

TRAP  C$ERHRD
.WORD 620
.WORD T26DTA
.WORD EXPREC
TRAP  C$CLP1
TRAP  C$CLP1
TRAP  C$CLP1
TRAP  C$ESUB

```



TEST 6: REREADS

```

5972
5973 057300 012704 071630          MOV      #T26PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
5974
5975          ;*****
5976          ;
5977          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPCHR)
5978          ;
5979          ;*****
5980
5981 057304 004737 010752          JSR      PC,WRTPCHR          ;ISSUE WRITE CHARACTERISTICS
5982 057310 103407          BCS     26$                ;BR, IF COMMAND ISSUED OK
5983 057312 005237 002212          INC     FATFLG            ;BUMP COUNT
5987 057316 010001          MOV     R0,R1             ;SAVE CONTENTS OF TSSR
5988 057320          ERRHRD  ERRNO,WRTPMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
          057320 104456          TRAP   C$ERHRD
          057322 001156          .WORD 622
          057324 005054          .WORD WRTPMSG
          057326 012124          .WORD SFIMSG
5989 057330          26$:   CKLOOP          ;LOOP IF SELECTED
          057330 104406          TRAP   C$CLP1
5990
5991          ;*****
5992          ;
5993          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE.
5994          ;
5995          ;*****
5996
5997 057332 004737 011104          JSR     PC,REWIND          ;CALL TAPE REWIND COMMAND
5998 057336 103413          BCS     30$                ;BR, IF NO PROBLEM
5999 057340 016501 000002          MOV     TSSR(R5),R1       ;GET TSSR
6000 057344 012702 000200          MOV     #TSSR,R2         ;SET UP EXPECTED TSSR
6001 057350 010004          MOV     R0,R4             ;PACKET ADDRESS SET UP
6002 057352 005237 002212          INC     FATFLG            ;BUMP COUNT
6006 057356          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
          057356 104456          TRAP   C$ERHRD
          057360 001157          .WORD 623
          057362 073304          .WORD T26RWN
          057364 012136          .WORD PKTSSR
6007 057366          30$:   CKLOOP          ;LOOP IF SELECTED
          057366 104406          TRAP   C$CLP1
6008
6009          ;*****
6010          ;
6011          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6012          ;
6013          ;*****
6014
6015 057370 013701 071660          MOV     T26BFR+6,R1       ;PICK UP XSTO
6016 057374 010102          MOV     R1,R2             ;SET UP EXPECTED
6017 057376 052702 000002          BIS     #BIT1,R2         ;SET BOT BIT IN EXPECTED
6018 057402 020102          CMP     R1,R2             ;DOES EXP = REC'D
6019 057404 001406          BEQ     40$                ;BR, IF EQUAL (OK)
6020 057406 005237 002212          INC     FATFLG            ;BUMP COUNT
6024 057412          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          057412 104456          TRAP   C$ERHRD
          057414 001160          .WORD 624
          057416 073015          .WORD T26BOT

```

## TEST 6: REREADS

```

057420 015564
6025 057422          40$:   CKLOOP                               ;LOOP IF SELECTED          .WORD  EXPREC
      057422 104406          ;                               TRAP    C$CLP1
6026 057424 012703 000400          MOV    #256.,R3          ;RECORD SIZE
6027 057430 013737 003114 071752  MOV    FREE,T26RB       ;STARTING WRITE BUFFER ADDRESS
6028
6029          ;*****
6030          ;
6031          ;WRITE DATA,CVC=1,ACK COMMAND
6032          ;
6033          ;*****
6034
6035 057436 012737 140005 071750  MOV    #140005,T26PK3   ;WRITE DATA,CVC=1,ACK COMMAND
6036 057444 012704 071750          MOV    #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
6037 057450          65$:
6038 057450 010300          MOV    R3,R0           ;SET PATTERN IN CORRECT REGISTER
6039 057452 004737 017512          JSR    PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
6040 057456 010337 071756          MOV    R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
6041 057462 013777 071776 123424  MOV    T26CNT,#FREE    ;MOVE TAPE RECORD NUMBER TO BUFFER
6042 057470 062737 000001 071776  ADD    #1,T26CNT       ;NUMBER READY FOR NEXT RECORD
6043 057476 010465 000000          MOV    R4,T26DB(R5)   ;ISSUE COMMAND
6044 057502 004737 016340          JSR    PC,WAITF       ;WAIT FOR SSR TO SET
6045 057506 016501 000002          MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
6046 057512 012702 000200          MOV    #SSR,R2        ;SET UP EXPECTED
6047 057516 020102          CMP    R1,R2          ;ARE THEY EQUAL
6048 057520 001406          BEQ    75$            ;BR, IF OK
6049 057522 005237 002212          INC    FATFLG         ;BUMP COUNT
6053 057526          ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      057526 104456          TRAP   C$ERRHRD
      057530 001161          .WORD 625
      057532 005111          .WORD WRTErr
      057534 012136          .WORD PKTSSR
6054 057536          75$:   CKLOOP                               ;LOOP IF SELECTED          .WORD  EXPREC
      057536 104406          ;                               TRAP    C$CLP1
6055 057540 005723          TST    (R3),          ;BUMP THE RECORD SIZE
6056 057542 022703 000414          CMP    #268.,R3      ;MAXIMUM SIZE YET
6057 057546 001401          BEQ    120$          ;BR, IF AT END OF WRITE SEQUENCE
6058 057550 000737          BR     65$           ;WRITE MORE RECORDS
6059 057552          120$:
6060 057552 005037 071776          CLR    T26CNT         ;SET RECORD COUNTER BACK TO ZERO
6061
6062          ;*****
6063          ;
6064          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6065          ;
6066          ;*****
6067
6068 057556 004737 011104          JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
6069 057562 103413          BCS    130$          ;BR, IF NO PROBLEM
6070 057564 016501 000002          MOV    TSSR(R5),R1    ;GET TSSR
6071 057570 012702 000200          MOV    #SSR,R2        ;SET UP EXPECTED TSSR
6072 057574 010004          MOV    R0,R4          ;PACKET ADDRESS SET UP
6073 057576 005237 002212          INC    FATFLG         ;BUMP COUNT
6077 057602          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      057602 104456          TRAP   C$ERRHRD
      057604 001162          .WORD 626
      057606 073304          .WORD T26RWN

```

TEST 6: REREADS

```

6078 057610 012136
057612 130$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
057612 104406 ;TRAP C$CLP1
6079
6080 ;*****
6081 ;
6082 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6083 ;
6084 ;*****
6085
6086 057614 013701 071660 MOV T26BFR+6,R1 ;PICK UP XSTO
6087 057620 010102 MOV R1,R2 ;SET UP EXPECTED
6088 057622 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6089 057626 020102 CMP R1,R2 ;DOES EXP = REC'D
6090 057630 001406 BEQ 140$ ;BR, IF EQUAL (OK)
6091 057632 005237 002212 INC FATFLG ;BUMP COUNT
6095 057636 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
057636 104456 ;TRAP C$ERHRD
057640 001163 ;.WORD 627
057642 073015 ;.WORD T26BOT
057644 015564 ;.WORD EXPREC
6096 057646 140$: CKLOOP ;LOOP IF SELECTED .WORD C$CLP1
057646 104406 ;TRAP
6097
6098 ;*****
6099 ;
6100 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
6101 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
6102 ;
6103 ;*****
6104
6105 057650 012703 000001 MOV #1,R3 ;SPACE 1 RECORD FORWARD
6106 057654 004737 010556 JSR PC,SPACE ;SPACE CALL
6107 057660 012703 000400 MOV #256.,R3 ;RECORD SIZE
6108 057664 013737 003114 071752 150$: MOV FREE,T26RR ;STARTING READ BUFFER ADDRESS
6109
6110 ;*****
6111 ;
6112 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6113 ;
6114 ;*****
6115
6116 057672 012737 161001 071750 MOV #161001,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6117 057700 012704 071750 165$: MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6118 057704 010337 071756 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6119 057710 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6120 057714 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
6121 057720 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6122 057724 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6123 057730 020102 CMP R1,R2 ;ARE THEY EQUAL
6124 057732 001406 BEQ 170$ ;BR, IF OK
6125 057734 005237 002212 INC FATFLG ;BUMP COUNT
6129 057740 ERRHRD ERRNO,T26RRG,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
057740 104456 ;TRAP C$ERHRD
057742 001164 ;.WORD 628
057744 072322 ;.WORD T26RRG
057746 012136 ;.WORD PKTSSR

```

TEST 6: REREADS

```

6130 057750          170$:  CKLOOP                      ;LOOP IF SELECTED
      057750 104406                                     TRAP      C$CLP1
6131 057752 005723                                     TST      (R3)+      ;BUMP RECORD SIZE
6132 057754 062737 000001 071776                   ADD      #1,T26CNT  ;BUMP TAPE RECORD COUNTER
6133
6134 ;*****
6135 ;
6136 ;READ DATA, CVC=1, ACK COMMAND
6137 ;
6138 ;*****
6139
6140 057760 012737 140001 071750                   MOV      #140001,T26PK3 ;READ DATA, CVC=1, ACK COMMAND
6141 057770 010337 071756                           MOV      R3,T26SZ     ;SET SIZE INTO PACKET
6142 057774 010465 000000                           MOV      R4,T26CNT    ;ISSUE READ DATA COMMAND
6143 060000 004737 016340                           JSR      PC,WAITF     ;WAIT FOR SSR
6144 060004 016501 000002                           MOV      TSSR(R5),R1 ;PICK UP THE TSSR
6145 060010 012702 002000                           MOV      #SSR,R2     ;SET UP EXPECTED
6146 060014 020102                                CMP      R1,R2        ;IS THE TSSR OK
6147 060016 001406                                BEQ      195$         ;BR, IF TSSR OK (GOOD)
6148 060020 005237 002212                           INC      FATFLG      ;BUMP COUNT
6152 060024                                ERRHRD  ERRNO,RDERR,PKTSSR ;READ DATA COMMAND FAILED
      060024 104456                                     TRAP      C$ERHRD
      060026 001165                                     .WORD    629
      060030 005204                                     .WORD    RDERR
      060032 012136                                     .WORD    PKTSSR
6153 060034          195$:  CKLOOP                      ;LOOP IF SELECTED
      060034 104406                                     TRAP      C$CLP1
6154 060036 017701 123052                           MOV      #FREE,R1    ;FIRST WORD FROM READ BUFFER
6155 060042 013702 071776                           MOV      T26CNT,R2   ;SET UP EXPECTED
6156 060046 020102                                CMP      R1,R2        ;IS TAPE POSITION CORRECT
6157 060050 001406                                BEQ      197$         ;KEEP GOING POSITION OK
6158 060052 005237 002212                           INC      FATFLG      ;BUMP COUNT
6162 060056                                ERRHRD  ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
      060056 104456                                     TRAP      C$ERHRD
      060060 001166                                     .WORD    630
      060062 072006                                     .WORD    T26WNG
      060064 015564                                     .WORD    EXPREC
6163 060066          197$:  CKLOOP                      ;LOOP IF SELECTED
      060066 104406                                     TRAP      C$CLP1
6164 060070 022703 000412                           CMP      #266.,R3    ;AT MAX SIZE YET
6165 060074 001401                                BEQ      200$         ;BR, IF AT END OF THE SUBTEST
6166 060076 000672                                BR       150$         ;KEEP GOING MORE RECORDS
6167 060100          200$:
6168 060100                                ENDSUB              ;>>>>>>>>>> END SUBTEST >>>>>>>>>>
      060100 104403                                -10105:
6169 060102 023727 002212 000017                   CMP      FATFLG,#15, ;IS ERROR COUNT A 25
6170 060110 103402                                BLO     999$         ;BR, IF LESS THAN 25
6171 060112 004737 017272                           JSR      PC,CKDRUP   ;TRY TO DROP THE UNIT
6172 060116          999$:

```



TEST 6: REREADS

```

6227 060222 060222 104406 26$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
;*****
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;*****
6235 060224 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6236 060230 103413 BCS 30$ ;BR, IF NO PROBL'M
6237 060232 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
6238 060236 012702 000200 MOV *SSR,R2 ;SET UP EXPECTED TSSR
6239 060242 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
6240 060244 005237 002212 INC FATFLG ;BUMP COUNT
6244 060250 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED TRAP C$FRHRD
;*****
;*****
6245 060260 060260 104406 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
;*****
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;*****
6253 060262 013701 071660 MOV T26BFR*6,R1 ;PICK UP XSTO
6254 060266 010102 MOV R1,R2 ;SET UP EXPECTED
6255 060270 052702 000002 BIS *BIT1,R2 ;SET BOT BIT IN EXPECTED
6256 060274 020102 CMP R1,R2 ;DOES EXP = REC'D
6257 060276 001406 BEQ 40$ ;BR, IF EQUAL (OK)
6258 060300 005237 002212 INC FATFLG ;BUMP COUNT
6262 060304 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND TRAP C$ERHRD
;*****
;*****
6263 060314 060314 104406 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
6264 060316 012703 000400 MOV *256.,R3 ;RECORD SIZE
6265 060322 013737 003114 071752 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
;*****
;WRITE DATA,CVC=1,ACK COMMAND
;*****
6273 060330 012737 140005 071750 MOV *140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
6274 060336 012704 071750 MOV *T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6275 060342 65$: MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
6276 060342 010300 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
6277 060344 004737 017512 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6278 060350 010337 071756

```





TEST 6: REREADS

```

060532 001175
060534 073015
060536 015564
6334 060540 140$: CKLOOP ;LOOP IF SELECTED
060540 104406 TRAP C$CLP1
6335
6336 ;*****
6337 ;
6338 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
6339 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
6340 ;
6341 ;*****
6342
6343 060542 012703 000001 MOV #1,R3 ;SET UP SPACE FORWARD 1
6344 060546 004737 010556 JSR PC,SPACE ;ISSUE SPACE COMMAND
6345 060552 012703 000400 MOV #256.,R3 ;RECORD SIZE
6346 060556 013737 003114 071752 150$: MOV FREE,T26RB ;STARTING READ BUFFER ADDRESS
6347
6348 ;*****
6349 ;
6350 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6351 ;
6352 ;*****
6353
6354 060564 012737 171001 071750 MOV #171001,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6355 060572 012704 071750 165$: MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6356 060576 010337 071756 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6357 060602 010465 000000 MOV R4,T26SDB(R5) ;ISSUE COMMAND
6358 060606 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
6359 060612 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6360 060616 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6361 060622 020102 CMP R1,R2 ;ARE THEY EQUAL
6362 060624 001406 BEQ 170$ ;BR, IF OK
6363 060626 005237 002212 INC FATFLG ;BUMP COUNT
6367 060632 ERRHRD ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
060632 104456 TRAP C$ERHRD
060634 001176 .WORD 638
060636 072225 .WORD T26RRF
060640 012136 .WORD PKTSSR
6368 060642 170$: CKLOOP ;LOOP IF SELECTED
060642 104406 TRAP C$CLP1
6369 060644 017701 122244 MOV #FREE,R1 ;FIRST WORD FROM READ BUFFER
6370 060650 013702 071776 MOV T26CNT,R2 ;SET UP EXPECTED
6371 060654 000302 SWAB R2 ;SWAP BYTES IN EXPECTED
6372 060656 020102 CMP R1,R2 ;IS TAPE POSITION CORRECT
6373 060660 001406 BEQ 190$ ;KEEP GOING POSITION OK
6374 060662 005237 002212 INC FATFLG ;BUMP COUNT
6378 060666 ERRHRD ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
060666 104456 TRAP C$ERHRD
060670 001177 .WORD 639
060672 072006 .WORD T26WNG
060674 015564 .WORD EXPREC
6379 060676 190$: CKLOOP
060676 104406 TRAP C$CLP1
6380 060700 005723 TST (R3), ;NEXT RECORD SIZE
6381 060702 062737 000001 071776 ADD #1,T26CNT ;BUMP TAPE RECORD COUNTER
6382

```

TEST 6: REREADS

```

6383 ;*****
6384 ;
6385 ;READ DATA, CVC=1, ACK COMMAND
6386 ;
6387 ;*****
6388
6389 060710 012737 140001 071750      MOV      #140001,T26PK3      ;READ DATA, CVC=1, ACK COMMAND
6390 060716 010337 071756            MOV      R3,T26SZ          ;SET SIZE INTO PACKET
6391 060722 010465 000000            MOV      R4,TSDB(R5)      ;ISSUE READ DATA COMMAND
6392 060726 004737 016340            JSR      PC,WAIF         ;WAIT FOR SSR
6393 060732 016501 000002            MOV      TSSR(R5),R1      ;PICK UP THE TSSR
6394 060736 012702 000200            MOV      #SSR,R2         ;SET UP EXPECTED
6395 060742 020102                    CMP      R1,R2            ;IS THE TSSR OK
6396 060744 001406                    BEQ      215$             ;BR, IF TSSR OK (GOOD)
6397 060746 005237 002212            INC      FATFLG           ;BUMP COUNT
6401 060752                    ERRHRD  ERRNO,T26RDF,PKTSSR ;READ DATA COMMAND FAILED
        060752 104456                                TRAP      C$ERHRD
        060754 001200                                .WORD    640
        060756 072156                                .WORD    T26RDF
        060760 012136                                .WORD    PKTSSR
6402 060762                    215$: CKLOOP                    ;LOOP IF SELECTED
        060762 104406                                TRAP      C$CLP1
6403 060764 017701 122124            MOV      #FREE,R1         ;FIRST WORD FROM READ BUFFER
6404 060770 013702 071776            MOV      T26CNT,R2       ;SET UP EXPECTED
6405 060774 020102                    CMP      R1,R2            ;IS TAPE POSITION CORRECT
6406 060776 001406                    BEQ      217$             ;KEEP GOING POSITION OK
6407 061000 005237 002212            INC      FATFLG           ;BUMP COUNT
6411 061004                    ERRHRD  ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
        061004 104456                                TRAP      C$ERHRD
        061006 001201                                .WORD    641
        061010 072006                                .WORD    T26WNG
        061012 015564                                .WORD    EXPREC
6412 061014                    217$: CKLOOP                    ;LOOP IF SELECTED
        061014 104406                                TRAP      C$CLP1
6413 061016 022703 000410            CMP      #264.,R3        ;AT MAX SIZE YET
6414 061022 001401                    BEQ      220$             ;BR, IF AT END OF THE SUBTEST
6415 061024 000654                    BR       150$             ;KEEP GOING MORE RECORDS
6416 061026                    220$:
6417 061026                    ENDSUB                    ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
        061026 L10106:
        061026 104403                                TRAP      C$ESUB
6418 061030 023727 002212 000017      CMP      FATFLG,#15      ;IS ERROR COUNT AT 25
6419 061036 103402                    BLO     999$             ;BR, IF LESS THAN 25
6420 061040 004737 017272            JSR      PC,CKDROP        ;TRY TO DROP THE UNIT
6421 061044                    999$:
    
```



TEST 6: REREADS

```

061144 104406                                TRAP      C$CLP1
6476
6477
6478
6479
6480
6481
6482
6483 061146 004737 011104                    JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
6484 061152 103413                            BCS     30$                ;BR, IF NO PROBLEM
6485 061154 016501 000002                    MOV     TSSR(R5),R1        ;GET TSSR
6486 061160 012702 000200                    MOV     #SSR,R2           ;SET UP EXPECTED TSSR
6487 061164 010004                            MOV     R0,R4              ;PACKET ADDRESS SET UP
6488 061166 005237 002212                    INC     FATFLG             ;BUMP COUNT
6492 061172                                ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     644
                                .WORD     T26RWN
                                .WORD     PKTSSR
061172 104456
061174 001204
061176 073304
061200 012136
6493 061202 30$: CKLOOP                        ;LOOP IF SELECTED
061202 104406                                TRAP      C$CLP1
6494
6495
6496
6497
6498
6499
6500
6501 061204 013701 071660                    MOV     T26BFR+6,R1       ;PICK UP XSTO
6502 061210 010102                            MOV     R1,R2             ;SET UP EXPECTED
6503 061212 052702 000002                    BIS     #BIT1,R2          ;SET BOT BIT IN EXPECTED
6504 061216 020102                            CMP     R1,R2             ;DOES EXP = REC'D
6505 061220 001406                            BEQ     40$                ;BR, IF EQUAL (OK)
6506 061222 005237 002212                    INC     FATFLG             ;BUMP COUNT
6510 061226                                ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD     645
                                .WORD     T26BOT
                                .WORD     EXPREC
061226 104456
061230 001205
061232 073015
061234 015564
6511 061236 40$: CKLOOP                        ;LOOP IF SELECTED
061236 104406                                TRAP      C$CLP1
6512 061240 012703 001000                    MOV     #512,R3           ;RECORD SIZE
6513 061244 013737 003114 071752            MOV     FREE,T26RB        ;STARTING WRITE BUFFER ADDRESS
6514
6515
6516
6517
6518
6519
6520
6521 061252 012737 140005 071750            MOV     #140005,T26PK3    ;WRITE DATA,CVC-1,ACK COMMAND
6522 061260 012704 071750                    MOV     #T26PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
6523 061264
6524 061264 010337 071756 65$:
6525 061270 010465 000000                    MOV     R3,T26SZ          ;SET UP RECORD SIZE IN PACKET
6526 061274 004737 016340                    MOV     R4,TSDB(R5)       ;ISSUE COMMAND
6527 061300 016501 000002                    JSR     PC,WAITF           ;WAIT FOR SSR TO SET
                                MOV     TSSR(R5),R1        ;GET TSSR CONTENTS

```



TEST 6: REREADS

```

6579 061456 012703 000777         MOV     #511.,R3        ;SET RECORD SIZE
6580 061462 013737 003114 071752 MOV     FREE,T26RB     ;STARTING READ BUFFER ADDRESS
6581
6582                               ;*****
6583                               ;
6584                               ;REREAD DATA,CVC=1,ACK COMMAND
6585                               ;
6586                               ;*****
6587
6588 061470 012737 141001 071750 MOV     #141001,T26PK3 ;REREAD DATA,CVC=1,ACK COMMAND
6589 061476 012704 071750 365$:  MOV     #T26PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
6590 061502 010337 071756       MOV     R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
6591 061506 010465 000000       MOV     R4,TSDB(R5)    ;ISSUE COMMAND
6592 061512 004737 016340       JSR     PC,WAITF       ;WAIT FOR SSR TO SET
6593 061516 016501 000002       MOV     TSSR(R5),R1   ;GET TSSR CONTENTS
6594 061522 012702 100204       MOV     #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6595 061526 020102               CMP     R1,R2         ;ARE THEY EQUAL
6596 061530 001406               BEQ     370$          ;BR, IF OK
6597 061532 005237 002212       INC     FATFLG        ;BUMP COUNT
6601 061536               ERRHRD ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
           061536 104456                TRAP   C$ERRRD
           061540 001211                .WORD 649
           061542 074362                .WORD T26TRL
           061544 012136                .WORD PKTSSR
6602 061546               370$:  CKLOOP                      ;LOOP IF SELECTED
           061546 104406                TRAP   C$CLP1
6603
6604                               ;*****
6605                               ;
6606                               ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
6607                               ;
6608                               ;*****
6609
6610 061550 013701 071660         MOV     T26BFR+6,R1   ;GET MESSAGE BUFFER
6611 061554 010102               MOV     R1,R2         ;SET UP EXPECTED
6612 061556 052702 010000       BIS     #BIT12,R2    ;SET THE RLL BIT IN EXPECTED
6613 061562 020102               CMP     R1,R2         ;ARE THEY EQUAL
6614 061564 001406               BEQ     380$          ;BR, IF EQUAL (ALL IS WELL)
6615 061566 005237 002212       INC     FATFLG        ;BUMP COUNT
6619 061572               ERRHRD ERRNO,T26LON,EXPREC ;THE RLL BIT WAS NOT SET IN XST0
           061572 104456                TRAP   C$ERRRD
           061574 001212                .WORD 650
           061576 074130                .WORD T26LON
           061600 015564                .WORD EXPREC
6620 061602               380$:  CKLOOP                      ;
           061602 104406                TRAP   C$CLP1
6621 061604               ENDSUB                      ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
           061604               L.10107:
           061604 104403               TRAP   C$ESUB
6622 061606 023727 002212 000017 CMP     FATFLG,#15.    ;IS ERROR COUNT AT 25
6623 061614 103402               BLD     999$          ;BR, IF LESS THAN 25
6624 061616 004737 017272       JSR     PC,CKDROP     ;TRY TO DROP THE UNIT
6625 061622               999$:

```

TEST 6: REREADS

```

6627      ;
6628      ;
6629      ;TEST 6, SUBTEST 6
6630      ;
6631      ;VERIFIES THAT A REREAD PREVIOUS COMMAND READING A
6632      ;RECORD SHORTER THAN THE SPECIFIED BYTE COUNT CAUSES
6633      ;TAPE STATUS ALERT TERMINATION WITH THE RECORD LENGTH
6634      ;SHORT (RLS) BIT SET. IT IS VERIFIED THAT THE
6635      ;RESIDUAL BYTE COUNTER (RBPCR) IN THE MESSAGE BUFFER
6636      ;CONTAINS THE PROPER NONZERO VALUE (E.G., THE
6637      ;DIFFERENCE BETWEEN THE ORIGINAL BYTE COUNT AND THE
6638      ;ACTUAL RECORD LENGTH). RESULTS ARE VERIFIED FOR BOTH
6639      ;STATES OF OPP (0 AND 1).
6640      ;
6641      ;
6642      ;
6643      ;
6644      BGNSUB                               ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
                                T6.6:
                                TRAP      C$BSUB
6645 061622 104402                       JSR      PC,T26REST           ;SET COMMAND PACKET
6646 061624 004737 074540                 JSR      PC,T26RT2          ;SET UP OTHER COMMAND PACKET
6647 061634 004737 074674                 JSR      PC,T26RT3          ;SET UP OTHER COMMAND PACKET
6648
6649      ;*****
6650      ;
6651      ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
6652      ;
6653      ;*****
6654      ;
6655 061640 004737 016064                   JSR      PC,SOFINIT         ;DO INITIALIZE ON CONTROLLER
6656 061644 103407                         BCS      20$                ;BR IF INIT WAS OK
6657 061646 005237 002212                   INC      FATFLG             ;BUMP COUNT
6661 061652 010001                         MOV      R0,R1              ;CONTENTS OF TSSR REGISTER
6662 061654 ERRDF ERRNO,S$FIERR,S$FIMSG    ;FATAL ERROR TSSR WAS NOT OK
                                TRAP      C$ERDF
                                .WORD     651
                                .WORD     S$FIERR
                                .WORD     S$FIMSG
6663 061664 013737 002172 071650 20$:    MOV      UNITN,T26DSW        ;SET UP UNIT NUMBER
6664
6665 061672 012704 071630                   MOV      @T26PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
6666
6667      ;*****
6668      ;
6669      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
6670      ;
6671      ;*****
6672      ;
6673 061676 004737 010752                       JSR      PC,WRTPHR          ;ISSUE WRITE CHARACTERISTICS
6674 061702 103407                         BCS      26$                ;BR, IF COMMAND ISSUED OK
6675 061704 005237 002212                   INC      FATFLG             ;BUMP COUNT
6679 061710 010001                         MOV      R0,R1              ;SAVE CONTENTS OF TSSR
6680 061712 ERRHRD ERRNO,WRTPHR,S$FIMSG    ;WRITE CHARACTERISTICS FAILED
                                TRAP      C$ERHRD
                                .WORD     652
                                .WORD     WRTPHR
061712 104456
061714 001214
061716 005054

```



TEST 6: REREADS

```

6681 061720 012124
6681 061722
6681 061722 104406
6682
6683
6684
6685
6686
6687
6688
6689 061724 004737 011104
6690 061730 103413
6691 061732 016501 000002
6692 061736 012702 000200
6693 061742 010004
6694 061744 005237 002212
6698 061750
6698 061750 104456
6698 061752 001215
6698 061754 073304
6698 061756 012136
6699 061760
6699 061760 104406
6700
6701
6702
6703
6704
6705
6706
6707 061762 013701 071660
6708 061766 010102
6709 061770 052702 000002
6710 061774 020102
6711 061776 001406
6712 062000 005237 002212
6716 062004
6716 062004 104456
6716 062006 001216
6716 062010 073015
6716 062012 015564
6717 062014
6717 062014 104406
6718 062016 012703 000400
6719 062022 013737 003114 071752
6720
6721
6722
6723
6724
6725
6726
6727 062030 012737 140005 071750
6728 062036 012704 071750
6729 062042
6730 062042 010337 071756
6731 062046 010465 000000

```

```

26$: CKLOOP ;LOOP IF SELECTED .WORD SFIMSG
TRAP C$CLP1

;*****
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE.
;*****

JSR PC,REWIND ;CALL TAPE REWIND COMMAND
BCS 30$ ;BR, IF NO PROBLEM
MOV TSSR(R5),R1 ;GET TSSR
MOV #SSR,R2 ;SET UP EXPECTED TSSR
MOV R0,R4 ;PACKET ADDRESS SET UP
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
TRAP C$ERHRD
.WORD 653
.WORD T26RWN
.WORD PKTSSR

30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1

;*****
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;*****

MOV T26BFR+6,R1 ;PICK UP XSTO
MOV R1,R2 ;SET UP EXPECTED
BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
CMP R1,R2 ;DOES EXP = REC'D
BEQ 40$ ;BR, IF EQUAL (OK)
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
TRAP C$ERHRD
.WORD 654
.WORD T26BOT
.WORD EXPREC

40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1

MOV #256.,R3 ;RECORD SIZE
MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS

;*****
;WRITE DATA,CVC=1,ACK COMMAND
;*****

MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS

65$: MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
MOV R4,TSDB(R5) ;ISSUE COMMAND

```

TEST 6: REREADS

```

6732 062052 004737 016340      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
6733 062056 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
6734 062062 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
6735 062066 020102                CMP      R1,R2        ;ARE THEY EQUAL
6736 062070 001406      BEQ      75$          ;BR, IF OK
6737 062072 005237 002212      INC      FATFLG       ;BUMP COUNT
6741 062076                ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    655
                                .WORD    WRTErr
                                .WORD    PKTSSR
                                TRAP      C$CLP1
                                .WORD    656
                                .WORD    T26TRL
                                .WORD    PKTSSR
6742 062106                75$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    657
                                .WORD    T26LOP
                                .WORD    EXPREC
6743 062110 012703 001000      MOV      #512.,R3     ;RECORD SIZE
6744 062114 013737 003114 071752  MOV      FREE,T26RB    ;STARTING READ BUFFER ADDRESS
6745
6746      ;*****
6747      ;
6748      ;REREAD PREVIOUS,ACK,CVC=1,OPP=1
6749      ;
6750      ;*****
6751
6752 062122 012737 161001 071750  MOV      #161001,T26PK3 ;REREAD PREVIOUS,ACK,CVC=1,OPP=1
6753 062130 012704 071750 165$:  MOV      #T26PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
6754 062134 010337 071756      MOV      R3,T26SZ     ;SET UP RECORD SIZE IN PACKET
6755 062140 010465 000000      MOV      R4,TSDB(R5)  ;ISSUE COMMAND
6756 062144 004737 016340      JSR      PC,WAITF     ;WAIT FOR SSR TO SET
6757 062150 016501 000002      MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
6758 062154 012702 100204      MOV      #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6759 062160 020102                CMP      R1,R2        ;ARE THEY EQUAL
6760 062162 001406      BEQ      170$         ;BR, IF OK
6761 062164 005237 002212      INC      FATFLG       ;BUMP COUNT
6765 062170                ERRHRD  ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    656
                                .WORD    T26TRL
                                .WORD    PKTSSR
6766 062200                170$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    657
                                .WORD    T26LOP
                                .WORD    EXPREC
6767
6768      ;*****
6769      ;
6770      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6771      ;
6772      ;*****
6773
6774 062202 013701 071660      MOV      T26+15,R1    ;GET MESSAGE BUFFER
6775 062206 010102                MOV      R1,R1        ;SET UP EXPECTED
6776 062210 052702 040000      BIS      14,RC        ;SET THE RLS BIT IN EXPECTED
6777 062214 020102                CMP      R1,R2        ;ARE THEY EQUAL
6778 062216 001406      BEQ      180$         ;BR, IF NORMAL (ALL IS WELL)
6779 062220 005237 002212      INC      FATFLG       ;BUMP COUNT
6783 062224                ERRHRD  ERRNO,T26LOP    ;THE RLS BIT WAS NOT SET IN XSTO
                                TRAP      C$ERHRD
                                .WORD    657
                                .WORD    T26LOP
                                .WORD    EXPREC

```

TEST 6: REREADS

```

6784 062234      180$:  CKLOOP
      062234 104406
6785 062236 013701 071656      MOV    T26BFR+4,R1      ;PICK UP RESIDUAL BYTE COUNTER
6786 062242 012702 000400      MOV    #256.,R2        ;THIS SHOULD BE THE DIFFERENCE
6787 062246 020102                CMP    R1,R2           ;IS THE DIFFERENCE CORRECT
6788 062250 001406                BEQ    190$            ;BR, IF CORRECT
6789 062252 005237 002212      INC    FATFLG          ;BUMP COUNT
6793 062256      ERRHRD  ERRNO,T26PBP,EXPREC ;RBPCR NOT CORRECT
      062256 104456
      062260 001222                TRAP   C$ERHRD
      062262 074274                .WORD 658
      062264 015564                .WORD T26PBP
      .WORD EXPREC
6794 062266      190$:  CKLOOP                ;LOOP IF SELECTED
      062266 104406                TRAP   C$CLP1
6795 062270 012703 001000      MOV    #512.,R3        ;RECORD SIZE
6796 062274 013737 003114 071752  MOV    FREE,T26RB      ;STARTING READ BUFFER ADDRESS
6797
6798 ;*****
6799 ;
6800 ;REREAD PREVIOUS,ACK,CVC=1,OPP=0
6801 ;
6802 ;*****
6803
6804 062302 012737 141001 071750      MOV    #141001,T26PK3 ;REREAD PREVIOUS,ACK,CVC=1,OPP=0
6805 062310 012704 071750      MOV    #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
6806 062314 010337 071756      MOV    R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
6807 062320 010465 000000      MOV    R4,T26SDB(R5) ;ISSUE COMMAND
6808 062324 004737 016340      JSR    PC,WAITF       ;WAIT FOR SSR TO SET
6809 062330 016501 000002      MOV    T26SSR(R5),R1 ;GET T26SSR CONTENTS
6810 062334 012702 100204      MOV    #T26SSR!SC!BIT2,R2 ;SET UP EXPECTED
6811 062340 020102                CMP    R1,R2           ;ARE THEY EQUAL
6812 062342 001406                BEQ    270$            ;BR, IF OK
6813 062344 005237 002212      INC    FATFLG          ;BUMP COUNT
6817 062350      ERRHRD  ERRNO,T26TRL,PKTSSR ;T26SSR INCORRECT AFTER READ DATA
      062350 104456                TRAP   C$ERHRD
      062352 001223                .WORD 659
      062354 074362                .WORD T26TRI
      062356 012136                .WORD PKTSSR
6818 062360      270$:  CKLOOP                ;LOOP IF SELECTED
      062360 104406                TRAP   C$CLP1
6819
6820 ;*****
6821 ;
6822 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6823 ;
6824 ;*****
6825
6826 062362 013701 071660      MOV    T26BFR+6,R1    ;GET MESSAGE BUFFER
6827 062366 010102                MOV    R1,R2           ;SET UP EXPECTED
6828 062370 052702 040000      BIS    #BIT14,R2      ;SET THE RLS BIT IN EXPECTED
6829 062374 020102                CMP    R1,R2           ;ARE THEY EQUAL
6830 062376 001406                BEQ    280$            ;BR, IF EQUAL (ALL IS WELL)
6831 062400 005237 002212      INC    FATFLG          ;BUMP COUNT
6835 062404      ERRHRD  ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
      052404 104456                TRAP   C$ERHRD
      062406 001224                .WORD 650
      062410 074212                .WORD T26LOP

```

TEST 6: REREADS

```

062412 015564
6836 062414 104406 280$: CKLOOP .WORD EXPREC
062414 104406 TRAP C$CLP1
6837 062416 013701 071656 .MOV T26BFR+4,R1 ;PICK UP RESIDUAL BYTE COUNTER
6838 062422 012702 000400 .MOV 4256.,R2 ;THIS SHOULD BE THE DIFFERENCE
6839 062426 020102 .CMP R1,R2 ;IS THE DIFFERENCE CORRECT
6840 062430 001405 .REQ 290$ ;BR, IF CORRECT
6844 062434 .ERRHRD ERRNO,T26PBP,EXPREC ;RBPOR NOT CORRECT
062434 104456 TRAP C$ERRHD
062436 001224 .WORD 660
062440 074274 .WORD T26PBP
062442 015564 .WORD EXPREC
6845 062444 290$: CKLOOP ;LOOP IF SELECTED
062444 104406 TRAP C$CLP1
6846 062446 .ENDSUB ;***** END SUBTEST *****
062446 104403 (10110: TRAP C$ESUB
6847 062450 023727 000012 000017 .CMP FATELG,015, ;IS ERROR COUNT AT 25
6848 062456 103402 .RLO 999$ ;BR, IF LESS THAN 25
6849 062460 004737 017272 .JSR PC,CKDROP ;TRY TO DROP THE UNIT
6850 062464 999$:

```



TEST 6: REREADS

```

062554 104456
062556 001226
062560 005054
062562 012124
6909 062564 26$: CKLOOP ;LOOP IF SELECTED
062564 104406 TRAP C$ERHRD
6910 ;*****
6911 ;
6912 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6913 ;
6914 ;*****
6915
6916
6917 062566 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6918 062572 103413 BCS 30$ ;BR, IF NO PROBLEM
6919 062574 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
6920 062600 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
6921 062604 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
6922 062606 005237 002212 INC FATFLG ;BUMP COUNT
6926 062612 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
062612 104456 TRAP C$ERHRD
062614 001227 .WORD 663
062616 073304 .WORD T26RWN
062620 012136 .WORD PKTSSR
6927 062622 30$: CKLOOP ;LOOP IF SELECTED
062622 104406 TRAP C$CLP1
6928 ;*****
6929 ;
6930 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
6931 ;
6932 ;*****
6933
6934
6935 062624 013701 071660 MOV T26BFR+6,R1 ;PICK UP XST0
6936 062630 010102 MOV R1,R2 ;SET UP EXPECTED
6937 062632 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6938 062636 020102 CMP R1,R2 ;DOES EXP = REC'D
6939 062640 001406 BEQ 40$ ;BR, IF EQUAL (OK)
6940 062642 005237 002212 INC FATFLG ;BUMP COUNT
6944 062646 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
062646 104456 TRAP C$ERHRD
062650 001230 .WORD 664
062652 073015 .WORD T26BOT
062654 015564 .WORD EXPREC
6945 062656 40$: CKLOOP ;LOOP IF SELECTED
062656 104406 TRAP C$CLP1
6946 062660 012703 000400 MOV #256.,R3 ;RECORD SIZE
6947 062664 013737 003114 071752 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
6948 ;*****
6949 ;
6950 ;WRITE DATA,CVC=1,ACK COMMAND
6951 ;
6952 ;*****
6953
6954
6955 062672 012737 140005 071750 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
6956 062700 012704 071750 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS

```

TEST 6: REREADS

```

6957 062704
6958 062704 010300
6959 062706 004737 017512
6960 062712 010337 071756
6961 062716 010465 000000
6962 062722 004737 016340
6963 062726 016501 000002
6964 062732 012702 000200
6965 062736 020102
6966 062740 001406
6967 062742 005237 002212
6971 062746
    062746 104456
    062750 001231
    062752 005111
    062754 012136
6972 062756
    062756 104406
6973 062760 005723
6974 062762 022703 000414
6975 062766 001346
6976 062770
    062770 104406
6977 062772
6978
6979
6980
6981
6982
6983
6984
6985 062772 004737 011104
6986 062776 103413
6987 063000 016501 000002
6988 063004 012702 000200
6989 063010 010004
6990 063012 005237 002212
6994 063016
    063016 104456
    063020 001232
    063022 073304
    063024 012136
6995 063026
    063026 104406
6996
6997
6998
6999
7000
7001
7002
7003 063030 013701 071660
7004 063034 010102
7005 063036 052702 000002
7006 063042 020102
7007 063044 001406
7008 063046 005237 002212

```

```

65$:  MOV      R3,R0           ;SET PATTERN IN CORRECT REGISTER
      JSR     PC,FILL MEM     ;FILL MEMORY WITH RECORD SIZE
      MOV     R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
      MOV     R4,TSD8(R5)    ;ISSUE COMMAND
      JSR     PC,WAITF       ;WAIT FOR SSR TO SET
      MOV     TSSR(R5),R1    ;GET TSSR CONTENTS
      MOV     *SSR,R2        ;SET UP EXPECTED
      CMP     R1,R2          ;ARE THEY EQUAL
      BEQ     75$           ;BR, IF OK
      INC     FATFLG         ;BUMP COUNT
      ERRHRD ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP  C$ERHRD
                                .WORD 665
                                .WORD WRTErr
                                .WORD PKTSSR
75$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP  C$CLP1
      TST     (R3)+          ;BUMP RECORD SIZE
      CMP     *268.,R3      ;END OF RECORD YET
      BNE     65$           ;BR, IF MORE RECORDS TO WRITE
80$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP  C$CLP1
120$:
;*****
;
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
      JSR     PC,REWIND      ;CALL TAPE REWIND COMMAND
      BCS     130$          ;BR, IF NO PROBLEM
      MOV     TSSR(R5),R1   ;GET TSSR
      MOV     *SSR,R2       ;SET UP EXPECTED TSSR
      MOV     R0,R4         ;PACKET ADDRESS SET UP
      INC     FATFLG        ;BUMP COUNT
      ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP  C$ERHRD
                                .WORD 666
                                .WORD T26RWN
                                .WORD PKTSSR
130$: CKLOOP                ;LOOP IF SELECTED
                                TRAP  C$CLP1
;*****
;
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
;
;*****
      MOV     T26BFR+6,R1   ;PICK UP XST0
      MOV     R1,R2         ;SET UP EXPECTED
      BIS     *BIT1,R2      ;SET BOT BIT IN EXPECTED
      CMP     R1,R2         ;DOES EXP = REC'D
      BEQ     140$         ;BR, IF EQUAL (OK)
      INC     FATFLG        ;BUMP COUNT

```

TEST 6: REREADS

```

7012 063052          ERRHRD  ERRNO,T26BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      063052 104456          TRAP          C$ERHRD
      063054 001233          .WORD        667
      063056 073015          .WORD        T26BOT
      063060 015564          .WORD        EXPREC
7013 063062          140$:  CKLOOP                    ;LOOP IF SELECTED
      063062 104406          TRAP          C$CLP1
7014 063064 012737 000400 072002          MOV      #256.,T26RSZ      ;STORE START RECORD SIZE
7015 063072 000420          BR          150$          ;SKIP THE SPACE THIS TIME
7016
7017 ;*****
7018 ;
7019 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
7020 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
7021 ;
7022 ;*****
7023
7024 063074 012703 000001 145$:  MOV      #1,R3          ;SPACE ONE RECORD PARAMETER
7025 063100 004737 010556          JSR      PC,SPACE        ;CALL SPACE ROUTINE
7026 063104 103413          BCS     150$          ;BR, IF NO PROBLEM WITH SPACE COMMAND
7027 063106 016501 000002          MOV      TSSR(R5),R1     ;GET TSSR
7028 063112 012702 000200          MOV      #SSR,R2        ;SET UP EXPECTED TSSR
7029 063116 010004          MOV      R0,R4          ;PACKET ADDRESS SET UP
7030 063120 005237 002212          INC      FATFLG         ;BUMP COUNT
7034 063124          ERRHRD  ERRNO,T26SC,EXPREC      ;POSITION (SPACE RECORDS) FAILED
      063124 104456          TRAP          C$ERHRD
      063126 001234          .WORD        668
      063130 072417          .WORD        T26SC
      063132 015564          .WORD        EXPREC
7035 063134          150$:  CKLOOP                    ;LOOP IF SELECTED
      063134 104406          TRAP          C$CLP1
7036 063136 013703 072002          MOV      T26RSZ,R3      ;RECORD SIZE
7037 063142 013737 003114 071752          MOV      FREE,T26RB     ;STARTING READ BUFFER ADDRESS
7038
7039 ;*****
7040 ;
7041 ;REREREAD DATA,CVC=1,ACK COMMAND
7042 ;
7043 ;*****
7044
7045 063150 012737 141401 071750          MOV      #141401,T26PK3 ;REREREAD DATA,CVC=1,ACK COMMAND
7046 063156 012704 071750          MOV      #T26PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
7047 063162 010337 071756          MOV      R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
7048 063166 010465 000000          MOV      R4,TSDB(R5)   ;ISSUE COMMAND
7049 063172 004737 016340          JSR      PC,WAITF      ;WAIT FOR SSR TO SET
7050 063176 016501 000002          MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
7051 063202 012702 000200          MOV      #SSR,R2        ;SET UP EXPECTED
7052 063206 020102          CMP      R1,R2         ;ARE THEY EQUAL
7053 063210 001406          BEQ     170$          ;BR, IF OK
7054 063212 005237 002212          INC      FATFLG         ;BUMP COUNT
7058 063216          ERRHRD  ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      063216 104456          TRAP          C$ERHRD
      063220 001235          .WORD        669
      063222 073640          .WORD        T26WDC
      063224 012136          .WORD        PKTSSR
7059 063226          170$:  CKLOOP                    ;LOOP IF SELECTED
      063226 104406          TRAP          C$CLP1

```







TEST 6: REREADS

```

7141 ;*****
7142 ;
7143 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7144 ;
7145 ;*****
7146
7147 063440 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
7148 063444 103413 BCS 30$ ;BR, IF NO PROBLEM
7149 063446 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
7150 063452 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
7151 063456 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
7152 063460 005237 002212 INC FATFLG ;BUMP COUNT
7153 063464 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
7154 063464 104456 TRAP C$ERHRD
7155 063466 001241 .WORD 673
7156 063470 073304 .WORD T26RWN
7157 063472 012136 .WORD PKTSSR
7158 063474 104406 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
7159 ;*****
7160 ;
7161 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
7162 ;
7163 ;*****
7164
7165 063476 013701 071660 MOV T26BFR+6,R1 ;PICK UP XST0
7166 063502 010102 MOV R1,R2 ;SET UP EXPECTED
7167 063504 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
7168 063510 020102 CMP R1,R2 ;DOES EXP = REC'D
7169 063512 001406 DEQ 40$ ;BR, IF EQUAL (OK)
7170 063514 005237 002212 INC FATFLG ;BUMP COUNT
7171 063520 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
7172 063520 104456 TRAP C$ERHRD
7173 063522 001242 .WORD 674
7174 063524 073015 .WORD T26BOT
7175 063526 015564 .WORD EXPREC
7176 063530 104406 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
7177 063532 012703 000400 MOV #256.,R3 ;RECORD SIZE
7178 063536 013737 003114 071752 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
7179 ;*****
7180 ;
7181 ;WRITE DATA,CVC-1,ACK,SWB COMMAND
7182 ;
7183 ;*****
7184
7185 063544 012737 150005 071750 MOV #150005,T26PK3 ;WRITE DATA,CVC-1,ACK,SWB COMMAND
7186 063552 012704 071750 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
7187 063556 65$:
7188 063560 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
7189 063564 004737 017512 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
7190 063568 010337 071756 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
7191 063570 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
7192 063574 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
7193 063600 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS

```

```

7194 063604 012702 000200      MOV     #SSR,R2      ;SET UP EXPECTED
7195 063610 020102      CMP     R1,R2      ;ARE THEY EQUAL
7196 063612 001406      BEQ    75$         ;BR, IF OK
7197 063624 005237 002212      INC     FATFLG     ;BUMP COUNT
7201 063620      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      063620 104456      TRAP   C$ERRHRD
      063622 001243      .WORD 675
      063624 005111      .WORD WRTErr
      063626 012136      .WORD PKTSSR
7202 063630      75$:   CKLOOP      ;LOOP IF SELECTED      TRAP   C$CLP1
      063630 103406
7203 063632 005723      TST    (R3)+      ;BUMP RECORD SIZE
7204 063634 022703 000414      CMP    #268.,R3  ;END OF RECORD YET
7205 063640 001346      BNE    65$         ;BR, IF MORE RECORDS TO WRITE
7206 063642      80$:   CKLOOP      ;LOOP IF SELECTED      TRAP   C$CLP1
      063642 104406
7207 063644      120$:
7208
7209      ;*****
7210      ;
7211      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7212      ;
7213      ;*****
7214
7215 063648 004707 011104      JSR    PC,REWIND  ;CALL TAPE REWIND COMMAND
7216 063650 103413      BUS    130$       ;BR, IF NO PROBLEM
7217 063652 016507 000002      MOV    TSSR(R5),R1 ;GET TSSR
7218 063654 012702 000200      MOV    #SSR,R2   ;SET UP EXPECTED TSSR
7219 063656 010004      MOV    R0,R4     ;PACKET ADDRESS SET UP
7220 063658 005237 002212      INC    FATFLG    ;BUMP COUNT
7224 063660      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      063670 104456      TRAP   C$ERRHRD
      063672 001244      .WORD 676
      063674 073304      .WORD T26RWN
      063676 012136      .WORD PKTSSR
7225 063700      130$:   CKLOOP      ;LOOP IF SELECTED      TRAP   C$CLP1
      063700 104406
7226
7227      ;*****
7228      ;
7229      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7230      ;
7231      ;*****
7232
7233 063702 013701 071660      MOV    T26BFR+6,R1 ;PICK UP XSTO
7234 063706 010102      MOV    R1,R2     ;SET UP EXPECTED
7235 063710 052702 000002      BIS    #BIT1,R2  ;SET BOT BIT IN EXPECTED
7236 063714 020102      CMP    R1,R2     ;DOES EXP = REC'D
7237 063716 001406      BEQ    140$      ;BR, IF EQUAL (OK)
7238 063720 005237 002212      INC    FATFLG    ;BUMP COUNT
7242 063724      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      063724 104456      TRAP   C$ERRHRD
      063726 001245      .WORD 677
      063730 073015      .WORD T26BOT
      063732 015564      .WORD EXPREC
7243 063734      140$:   CKLOOP      ;LOOP IF SELECTED      TRAP   C$CLP1
      063734 104406

```







TEST 6: REREADS

```

7373
7374 064264 012704 071630          MOV    #T26PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
7375
7376          ;*****
7377          ;
7378          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPCHR)
7379          ;
7380          ;*****
7381
7382 064270 004737 010752          JSR    PC,WRTPCHR          ;ISSUE WRITE CHARACTERISTICS
7383 064274 103407          BCS    26$                ;BR, IF COMMAND ISSUED OK
7384 064276 005237 002212          INC    FATFLG            ;BUMP COUNT
7388 064302 010001          MOV    R0,R1             ;SAVE CONTENTS OF TSSR
7389 064304          ERRHRD  ERRNO,WRTPMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
          064304 104456          TRAP   C$ERHRD
          064306 001252          .WORD 682
          064310 005054          .WORD WRTPMSG
          064312 012124          .WORD SFIMSG
7390 064314          2c$:  CKLOOP          ;LOOP IF SELECTED
          064314 104406          TRAP   C$CLP1
7391
7392          ;*****
7393          ;
7394          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7395          ;
7396          ;*****
7397
7398 064316 004737 011104          JSR    PC,REWIND          ;CALL TAPE REWIND COMMAND
7399 064322 103413          BCS    30$                ;BR, IF NO PROBLEM
7400 064324 016501 000002          MOV    TSSR(R5),R1       ;GET TSSR
7401 064330 012702 000200          MOV    #SSR,R2          ;SET UP EXPECTED TSSR
7402 064334 010004          MOV    R0,R4             ;PACKET ADDRESS SET UP
7403 064336 005237 002212          INC    FATFLG            ;BUMP COUNT
7407 064342          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
          064342 104456          TRAP   C$ERHRD
          064344 001253          .WORD 683
          064346 073304          .WORD T26RWN
          064350 012136          .WORD PKTSSR
7408 064352          30$:  CKLOOP          ;LOOP IF SELECTED
          064352 104406          TRAP   C$CLP1
7409
7410          ;*****
7411          ;
7412          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7413          ;
7414          ;*****
7415
7416 064354 013701 071660          MOV    T26BFR+6,R1       ;PICK UP XSTO
7417 064360 010102          MOV    R1,R2             ;SET UP EXPECTED
7418 064362 052702 000002          BIS    #BIT1,R2          ;SET BOT BIT IN EXPECTED
7419 064366 020102          CMP    R1,R2             ;DOES EXP = REC'D
7420 064370 001406          BEQ    40$                ;BR, IF EQUAL (OK)
7421 064372 005237 002212          INC    FATFLG            ;BUMP COUNT
7425 064376          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          064376 104456          TRAP   C$ERHRD
          064400 001254          .WORD 684
          064402 073015          .WORD T26BOT

```



TEST 6: REREADS

```

064404 015564
7426 064406 015564 30$: CKLOOP ;LOOP IF SELECTED ;WORD EXPREC
064406 104406
7427 064410 012703 000400 MOV 0256,,R3 ;RECORD SIZE TRAP C$CLP1
7428 064414 013737 003114 071752 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
7429
7430 ;*****
7431 ;
7432 ;WRITE DATA,CVC=1,ACK COMMAND
7433 ;
7434 ;*****
7435
7436 064422 012737 140005 071750 MOV 0140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
7437 064430 012704 071750 MOV 0T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
7438 064434 65$:
7439 064434 010337 071756 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
7440 064440 013777 071776 116446 MOV T26CNT,0FREE ;MOVE TAPE RECORD NUMBER TO BUFFER
7441 064446 062737 000001 071776 ADD 01,T26CNT ;NUMBER READY FOR NEXT RECORD
7442 064454 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
7443 064460 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
7444 064464 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
7445 064470 012702 000200 MOV 0SSR,R2 ;SET UP EXPECTED
7446 064474 020102 CMP R1,R2 ;ARE THEY EQUAL
7447 064476 001406 BEQ 75$ ;BR, IF OK
7448 064500 005237 002212 INC FATFLG ;BUMP COUNT
7452 064504 ERRHRD ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
064504 104456 TRAP C$ERRRD
064506 001255 ;WORD 685
064510 005111 ;WORD WRTErr
064512 012136 ;WORD PKTSSR
7453 064514 75$: CKLOOP ;LOOP IF SELECTED ;WORD TRAP C$CLP1
064514 104406
7454 064516 005723 TST (R3), ;BUMP THE RECORD SIZE
7455 064520 022703 000414 CMP 0268,,R3 ;MAXIMUM SIZE YET
7456 064524 001401 BEQ 120$ ;BR, IF AT END OF WRITE SEQUENCE
7457 064526 000742 BR 65$ ;WRITE MORE RECORDS
7458 064530 120$:
7459 064530 005037 071776 CLR T26CNT ;SET RECORD COUNTER BACK TO ZERO
7460
7461 ;*****
7462 ;
7463 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7464 ;
7465 ;*****
7466
7467 064534 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
7468 064540 103411 BCS 130$ ;BR, IF NO PROBLEM
7469 064542 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
7470 064546 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
7471 064550 005237 002212 INC FATFLG ;BUMP COUNT
7475 064554 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
064554 104456 TRAP C$ERRRD
064556 001256 ;WORD 686
064560 073304 ;WORD T26RWN
064562 012136 ;WORD PKTSSR
7476 064564 130$: CKLOOP ;LOOP IF SELECTED ;WORD TRAP C$CLP1
064564 104406

```

TEST 6: REREADS

```

7477
7478
7479
7480
7481
7482
7483
7484 064566 013701 071660
7485 064572 010102
7486 064574 052702 000002
7487 064600 020102
7488 064602 001406
7489 064604 005237 002212
7493 064610
      064610 104456
      064612 001257
      064614 073015
      064616 015564
7494 064620
      064620 104406
7495 064622 012737 000400 072002
7496 064630 000420
7497
7498
7499
7500
7501
7502
7503
7504
7505 064632 012703 000001
7506 064636 004737 010556
7507 064642 103413
7508 064644 016501 000002
7509 064650 012702 000200
7510 064654 010004
7511 064656 005237 002212
7515 064662
      064662 104456
      064664 001260
      064666 072417
      064670 012136
7516 064672
      064672 104406
7517 064674 013703 072002
7518 064700 013737 003114 071752
7519
7520
7521
7522
7523
7524
7525
7526 064706 012737 161401 071750
7527 064714 012704 071750
7528 064720 010337 071756
7529 064724 010465 000000

```

```

;*****
;
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;
;*****
      MOV     T26BFR+6,R1      ;PICK UP XSTO
      MOV     R1,R2           ;SET UP EXPECTED
      BIS     @BIT1,R2        ;SET BOT BIT IN EXPECTED
      CMP     R1,R2           ;DOES EXP = REC'D
      BFE     135$           ;BR, IF EQUAL (OK)
      INC     FATFLG          ;BUMP COUNT
      ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP   C$ERRHRD
                                .WORD  687
                                .WORD  T26BOT
                                .WORD  EXPREC
135$:  CKLOOP                    ;LOOP IF SELECTED
                                TRAP   C$CLP1
      MOV     @256.,T26RSZ    ;STARTING RECORD SIZE
      BR     140$            ;SKIP OVER THE SPCF THIS TIME
;*****
;
;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
;BIT 15 SETS DIRECTION - 0-FORWARD 1-REVERSE
;
;*****
132$:  MOV     @000001,R3     ;SET UP SPACE COMMAND (1 FORWARD)
      JSR     PC,SPACE       ;CALL SPACE ROUTINE
      BCS     140$           ;BR, IF NO TROUBLE
      MOV     TSSR(R5),R1    ;GET TSSR
      MOV     @SSR,R2        ;SET UP EXPECTED TSSR
      MOV     R0,R4          ;PACKET ADDRESS SET UP
      INC     FATFLG          ;BUMP COUNT
      ERRHRD ERRNO,T26SC,PKTSSR ;SPACE (FORWARD) FAILED
                                TRAP   C$ERRHRD
                                .WORD  688
                                .WORD  T26SC
                                .WORD  PKTSSR
140$:  CKLOOP                    ;LOOP IF SELECTED
                                TRAP   C$CLP1
      MOV     T26RSZ,R3      ;RECORD SIZE
150$:  MOV     FREE,T26RB    ;STARTING READ BUFFER ADDRESS
;*****
;
;REREAD DATA,CVC-1,ACK, OPP COMMAND
;
;*****
165$:  MOV     @161401,T26PK3 ;REREAD DATA,CVC-1,ACK, OPP COMMAND
      MOV     @T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
      MOV     R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
      MOV     R4,TSDB(R5)    ;ISSUE COMMAND

```





TEST 6: REREADS

```

7617 065170      26$:  CKLOOP                      ;LOOP IF SELECTED
      065170 104406                                TRAP      C$CLP1
7618
7619      ;*****
7620      ;
7621      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7622      ;
7623      ;*****
7624
7625 065172 004737 011104      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
7626 065176 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR
7627 065202 012702 000200      MOV      *SSR,R2         ;SET UP EXPECTED TSSR
7628 065206 103407              BCS      30$              ;BR, IF NO PROBLEM
7629 065210 010004              MOV      R0,R4           ;PACKET ADDRESS SET UP
7630 065212 005237 002212      INC      FATFLG          ;BUMP COUNT
7634 065216              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      065216 104456                                TRAP      C$ERRHRD
      065220 001265                                .WORD    693
      065222 073304                                .WORD    T26RWN
      065224 012136                                .WORD    PKTSSR
7635 065226      30$:  CKLOOP                      ;LOOP IF SELECTED
      065226 104406                                TRAP      C$CLP1
7636
7637      ;*****
7638      ;
7639      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7640      ;
7641      ;*****
7642
7643 065230 013701 071560      MOV      T26BFR+6,R1     ;PICK UP XSTO
7644 065234 010102              MOV      R1,R2           ;SET UP EXPECTED
7645 065236 052702 000002      BIS      *BIT1,R2        ;SET BOT BIT IN EXPECTED
7646 065242 020102              CMP      R1,R2           ;DOES EXP = REC'D
7647 065244 001406              BEQ      40$              ;BR, IF EQUAL (OK)
7648 065246 005237 002212      INC      FATFLG          ;BUMP COUNT
7652 065252              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      065252 104456                                TRAP      C$ERRHRD
      065254 001266                                .WORD    694
      065256 073015                                .WORD    T26BOT
      065260 015564                                .WORD    EXPREC
7653 065262      40$:  CKLOOP                      ;LOOP IF SELECTED
      065262 104406                                TRAP      C$CLP1
7654 065264 012703 000400      MOV      *256.,R3        ;RECORD SIZE
7655 065270 013737 003114 071752  MOV      FREE,T26RB      ;STARTING WRITE BUFFER ADDRESS
7656
7657      ;*****
7658      ;
7659      ;WRITE DATA,CVC=1,ACK COMMAND
7660      ;
7661      ;*****
7662
7663 065276 012737 140005 071750      MOV      *140005,T26PK3  ;WRITE DATA,CVC=1,ACK COMMAND
7664 065304 012704 071750      MOV      *T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
7665 065310      65$:
7666 065310 010337 071756      MOV      R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
7667 065314 013777 071776 115572  MOV      T26CNT,*FREE    ;MOVE TAPE RECORD NUMBER TO BUFFER
7668 065322 062737 000001 071776  ADD      *1,T26CNT       ;NUMBER READY FOR NEXT RECORD

```

TEST 6: REREADS

```

7669 065330 010465 000000      MOV      R4,TSDB(R5)          ;ISSUE COMMAND
7670 065334 004737 016340      JSR      PC,WAITF           ;WAIT FOR SSR TO SET
7671 065340 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
7672 065344 012702 000200      MOV      #SSR,R2           ;SET UP EXPECTED
7673 065350 020102              CMP      R1,R2             ;ARE THEY EQUAL
7674 065352 001406              BEQ      75$               ;BR, IF OK
7675 065354 005237 002212      INC      FATFLG            ;BUMP COUNT
7679 065360              ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    695
                                .WORD    WRTERR
                                .WORD    PKTSSR
                                TRAP      C$CLP1
065360 104456
065362 001267
065364 005111
065366 012136
7680 065370              75$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
065370 104406
7681 065372 005723              TST      (R3)+             ;BUMP THE RECORD SIZE
7682 065374 022703 000414      CMP      #268.,R3         ;MAXIMUM SIZE YET
7683 065400 001401              BEQ      120$             ;BR, IF AT END OF WRITE SEQUENCE
7684 065402 000742              BR       65$              ;WRITE MORE RECORDS
7685 065404
7686 065404 005037 071776      120$:  CLR      T26CNT      ;SET RECORD COUNTER BACK TO ZERO
7687
7688 ;*****
7689 ;
7690 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7691 ;
7692 ;*****
7693
7694 065410 004737 011104      JSR      PC,REWIND         ;CALL TAPE REWIND COMMAND
7695 065414 103411              BCS     130$              ;BR, IF NO PROBLEM
7696 065416 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR
7697 065422 010004              MOV     RO,R4             ;PACKET ADDRESS SET UP
7698 065424 005237 002212      INC      FATFLG            ;BUMP COUNT
7702 065430              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    696
                                .WORD    T26RWN
                                .WORD    PKTSSR
065430 104456
065432 001270
065434 073304
065436 012136
7703 065440              130$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
065440 104406
7704
7705 ;*****
7706 ;
7707 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7708 ;
7709 ;*****
7710
7711 065442 013701 071660      MOV      T26BFR+6,R1       ;PICK UP XSTO
7712 065446 010102              MOV     R1,R2             ;SET UP EXPECTED
7713 065450 052702 000002      BIS     #BIT1,R2          ;SET BOT BIT IN EXPECTED
7714 065454 020102              CMP     R1,R2             ;DOES EXP = REC'D
7715 065456 001406              BEQ     135$              ;BR, IF EQUAL (OK)
7716 065460 005237 002212      INC     FATFLG            ;BUMP COUNT
7720 065464              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    697
                                .WORD    T26BOT
                                .WORD    EXPREC
065464 104456
065466 001271
065470 073015
065472 015564

```

TEST 6: REREADS

```

7721 065474          135$:  CKLOOP                      ;LOOP IF SELECTED
      065474 104406          TRAP          C$CLP1
7722 065476 012737 000400 072002      MOV      #256.,T26RSZ      ;START RECORD SIZE
7723 065504 000420          BR          140$          ;SKIP OVER SPACE
7724
7725 ;*****
7726 ;
7727 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
7728 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
7729 ;
7730 ;*****
7731
7732 065506 012703 000001 136$:  MOV      #000001,R3      ;SET UP SPACE COMMAND (1 FORWARD)
7733 065512 004737 010556          JSR      PC,SPACE      ;CALL SPACE ROUTINE
7734 065516 103413          BCS     140$          ;BR, IF NO TROUBLE
7735 065520 016501 000002          MOV      TSSR(R5),R1    ;GET TSSR
7736 065524 012702 000200          MOV      #SSR,R2      ;SET UP EXPECTED TSSR
7737 065530 010004          MOV      R0,R4        ;PACKET ADDRESS SET UP
7738 065532 005237 002212          INC     FATFLG        ;BUMP COUNT
7742 065536          ERRHRD  ERRNO,T26SC,PKTSSR ;SPACE (FORWARD) FAILED
      065536 104456          TRAP          C$ERHRD
      065540 001272          .WORD     698
      065542 072417          .WORD     T26SC
      065544 012136          .WORD     PKTSSR
7743 065546          140$:  CKLOOP                      ;LOOP IF SELECTED
      065546 104406          TRAP          C$CLP1
7744 065550 013703 072002          MOV      T26RSZ,R3     ;RECORD SIZE
7745 065554 013737 003114 071752 150$:  MOV      FREE,T26RB     ;STARTING READ BUFFER ADDRESS
7746
7747 ;*****
7748 ;
7749 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
7750 ;
7751 ;*****
7752
7753 065562 012737 161401 071750          MOV      #161401,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
7754 065570 012704 071750          165$:  MOV      #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
7755 065574 010337 071756          MOV      R3,T26SZ     ;SET UP RECORD SIZE IN PACKET
7756 065600 010465 000000          MOV      R4,TSDB(R5)  ;ISSUE COMMAND
7757 065604 004737 016340          JSR      PC,WAITF     ;WAIT FOR SSR TO SET
7758 065610 016501 000002          MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
7759 065614 012702 000200          MOV      #SSR,R2     ;SET UP EXPECTED
7760 065620 020102          CMP     R1,R2        ;ARE THEY EQUAL
7761 065622 001406          BEQ     170$        ;BR, IF OK
7762 065624 005237 002212          INC     FATFLG        ;BUMP COUNT
7766 065630          ERRHRD  ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      065630 104456          TRAP          C$ERHRD
      065632 001273          .WORD     699
      065634 072225          .WORD     T26RRF
      065636 012136          .WORD     PKTSSR
7767 065640          170$:  CKLOOP                      ;LOOP IF SELECTED
      065640 104406          TRAP          C$CLP1
7768 065642 017701 115246          MOV      #FREE,R1     ;FIRST WORD FROM READ BUFFER
7769 065646 013702 071776          MOV      T26CNT,R2   ;SET UP EXPECTED
7770 065652 020102          CMP     R1,R2        ;IS TAPE POSITION CORRECT
7771 065654 001406          BEQ     190$        ;KEEP GOING POSITION OK
7772 065656 005237 002212          INC     FATFLG        ;BUMP COUNT

```

TEST 6: REREADS

```

7776 065662          ERRHRD  ERRNO,T26WNG,EXPREC      ;TAPE POSITION INCORRECT
      065662 104456
      065664 001274
      065666 072006
      065670 015564
7777 065672          190$:  CKLOOP
      065672 104406
7778 065674 062737 000001 071776  ADD      #1,T26CNT      ;BUMP TAPE RECORD COUNTER
7779 065702 005723          TST      (R3)          ;NEXT RECORD SIZE
7780 065704 010337 072002  MOV      R3,T26R5Z     ;STORE RECORD SIZE
7781 065710 022703 000412  CMP      #266.,R3     ;AT MAX SIZE YET
7782 065714 001402          BEQ      220$          ;BR, IF AT END OF THE SUBTEST
7783 065716 000137 065506  JMP      136$          ;KEEP GOING MORE RECORDS
7784 065722          220$:
7785 065722          ENDSUB
      065722
      065722 104403
7786 065724 023727 002212 000017  CMP      FATFLG,#15.   ;IS ERROR COUNT AT 25
7787 065732 103402          BLO      999$          ;BR, IF LESS THAN 25
7788 065734 004737 017272  JSR      PC,CKDROP     ;TRY TO DROP THE UNIT
7789 065740          999$:
    
```





TEST 6: REREADS

```

066040 104406                                     TRAP    C$CLP1
7844
7845
7846
7847
7848
7849
7850
7851 066042 004737 011104             JSR     PC,REWIND             ;CALL TAPE REWIND COMMAND
7852 066046 016501 000002             MOV     TSSR(R5),R1          ;GET TSSR
7853 066052 012702 000200             MOV     #SSR,R2              ;SET UP EXPECTED TSSR
7854 066056 103407                     BCS     30$                   ;BR, IF NO PROBLEM
7855 066060 010004                     MOV     R0,R4                 ;PACKET ADDRESS SET UP
7856 066062 005237 002212             INC     FATFLG                ;BUMP COUNT
7860 066066                     ERRHRD  ERRNO,T26RWN,PKTSSR   ;REWIND NOT ACCEPTED
066066 104456                                     TRAP    C$ERRHRD
066070 001277                                     .WORD  703
066072 073304                                     .WORD  T26RWN
066074 012136                                     .WORD  PKTSSR
7861 066076 104406             30$:   CKLOOP                   ;LOOP IF SELECTED             TRAP    C$CLP1
7862
7863
7864
7865
7866
7867
7868
7869 066100 013701 071660             MOV     T26BFR+6,R1          ;PICK UP XSTO
7870 066104 010102             MOV     R1,R2                 ;SET UP EXPECTED
7871 066106 052702 000002             BIS     #BIT1,R2             ;SET BOT BIT IN EXPECTED
7872 066112 020102             CMP     R1,R2                 ;DOES EXP = REC'D
7873 066114 001406             REQ     40$                   ;BR, IF EQUAL (OK)
7874 066116 005237 002212             INC     FATFLG                ;BUMP COUNT
7878 066122             ERRHRD  ERRNO,T26BOT,EXPREC   ;TAPE NOT AT BOT AFTER REWIND
066122 104456                                     TRAP    C$ERRHRD
066124 001300                                     .WORD  704
066126 073015                                     .WORD  T26BOT
066130 015564                                     .WORD  EXPREC
7879 066132 104406             40$:   CKLOOP                   ;LOOP IF SELECTED             TRAP    C$CLP1
066132 104406
7880 066134 012703 001000             MOV     #S12.,R3             ;RECORD SIZE
7881 066140 013737 003114 071752       MOV     FREE,T26RB           ;STARTING WRITE BUFFER ADDRESS
7882
7883
7884
7885
7886
7887
7888
7889 066146 012737 140005 071750       MOV     #140005,T26PK3       ;WRITE DATA,CVC=1,ACK COMMAND
7890 066154 012704 071750             MOV     #T26PK3,R4           ;SET UP R4 WITH PACKET ADDRESS
7891 066160
7892 066160 010337 071756             65$:   MOV     R3,T26SZ         ;SET UP RECORD SIZE IN PACKET
7893 066164 010465 000000             MOV     R4,TSD8(R5)          ;ISSUE COMMAND
7894 066170 004737 016340             JSR     PC,WAITF              ;WAIT FOR SSR TO SET
7895 066174 016501 000002             MOV     TSSR(R5),R1          ;GET TSSR CONTENTS

```

TEST 6: REREADS

```

7896 066200 012702 000200      MOV    #SSR,R2      ;SET UP EXPECTED
7897 066204 020102             CMP    R1,R2       ;ARE THEY EQUAL
7898 066206 001406             BEQ    75$         ;BR, IF OK
7899 066210 005237 002212      INC    FATFLG      ;BUMP COUNT
7903 066214             ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERRRD
                                .WORD   705
                                .WORD   WRTERR
                                .WORD   PKTSSR
                                066214 104456
                                066216 001301
                                066220 005111
                                066222 012136
7904 066224             75$:   CKLOOP      ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                066224 104406
7905
7906 ;*****
7907 ;
7908 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7909 ;
7910 ;*****
7911
7912 066226 004737 011104      JSR    PC,REWIND   ;CALL TAPE REWIND COMMAND
7913 066232 016501 000002      MOV    TSSR(R5),R1 ;GET TSSR
7914 066236 012702 000200      MOV    #SSR,R2    ;SET UP EXPECTED TSSR
7915 066242 103407             BCS    130$       ;BR, IF NO PROBLEM
7916 066244 010004             MOV    R0,R4      ;PACKET ADDRESS SET UP
7917 066246 005237 002212      INC    FATFLG      ;BUMP COUNT
7921 066252             ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERRRD
                                .WORD   706
                                .WORD   T26RWN
                                .WORD   PKTSSR
                                066252 104456
                                066254 001302
                                066256 073304
                                066260 012136
7922 066262             130$:  CKLOOP      ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                066262 104406
7923
7924 ;*****
7925 ;
7926 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7927 ;
7928 ;*****
7929
7930 066264 013701 071660      MOV    T26RFR+6,R1 ;PICK UP XSTO
7931 066270 010102             MOV    R1,R2      ;SET UP EXPECTED
7932 066272 052702 000002      BIS    #BIT1,R2   ;SET BOT BIT IN EXPECTED
7933 066276 020102             CMP    R1,R2      ;DOES EXP = REC'D
7934 066300 001406             BEQ    140$       ;BR, IF EQUAL (OK)
7935 066302 005237 002212      INC    FATFLG      ;BUMP COUNT
7939 066306             ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERRRD
                                .WORD   707
                                .WORD   T26BOT
                                .WORD   EXPREC
                                066306 104456
                                066310 001303
                                066312 073015
                                066314 015564
7940 066316             140$:  CKLOOP      ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                066316 104406
7941 066320 005303             DEC    R3          ;SET RECORD SIZE TO 511.
7942 066322 013737 003114 071752 MOV    FREE,T26RB ;STARTING READ BUFFER ADDRESS
7943
7944 ;*****
7945 ;
7946 ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND

```

TEST 6: REREADS

```

7947
7948
7949
7950 066330 012737 161401 071750
7951 066336 012704 071750 165$: MOV #161401,T26PK3 ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
7952 066342 010337 071756 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
7953 066346 010465 000000 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
7954 066352 004737 016340 MOV R4,TSDB(R5) ;ISSUE COMMAND
7955 066356 016501 000002 JSR PC,WAITF ;WAIT FOR SSR TO SET
7956 066362 012702 100204 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
7957 066366 020102 MOV #SSR!SC!BIT2,R2 ;SET UP EXPECTED
7958 066370 001406 CMP R1,R2 ;ARE THEY EQUAL
7959 066372 005237 002212 BEQ 170$ ;BR, IF OK
7963 066376 ERRHRD ERRNO,T26TRL,PKTSSR ;BUMP COUNT
;TSSR INCORRECT AFTER REREAD DATA
066376 104456 TRAP C$ERHRD
066400 001304 .WORD 708
066402 074362 .WORD T26TRL
066404 012136 .WORD PKTSSR
7964 066406 170$: CKLOOP ;LOOP IF SELECTED
066406 104406 TRAP C$CLP1
7965
7966
7967
7968
7969 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7970
7971
7972 066410 013701 071660 MOV T26BFR+6,R1 ;GET MESSAGE BUFFER
7973 066414 010102 MOV R1,R2 ;SET UP EXPECTED
7974 066416 052702 010000 BIS #BIT12,R2 ;SET THE RLL BIT IN EXPECTED
7975 066422 020102 CMP R1,R2 ;ARE THEY EQUAL
7976 066424 001406 BEQ 180$ ;BR, IF EQUAL (ALL IS WELL)
7977 066426 005237 002212 INC FATFLG ;BUMP COUNT
7981 066432 ERRHRD ERRNO,T26LON,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
066432 104456 TRAP C$ERHRD
066434 001305 .WORD 709
066436 074130 .WORD T26LON
06644 015564 .WORD EXPREC
7982 066442 180$: CKLOOP TRAP C$CLP1
066442 104406
7983 066444 012703 000777 MOV #511,R3 ;SET UP SIZE OF RECORD
7984 066450 013737 003114 071752 MOV FREE,T26RB ;STARTING READ BUFFER ADDRESS
7985
7986
7987
7988 ;REREAD DATA,CVC=1,ACK COMMAND
7989
7990
7991
7992 066456 012737 141401 071750
7993 066464 012704 071750 365$: MOV #141401,T26PK3 ;REREAD DATA,CVC=1,ACK COMMAND
7994 066470 010337 071756 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
7995 066474 010465 000000 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
7996 066500 004737 016340 MOV R4,TSDB(R5) ;ISSUE COMMAND
7997 066504 016501 000002 JSR PC,WAITF ;WAIT FOR SSR TO SET
7998 066510 012702 100204 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
7999 066514 020102 MOV #SSR!SC!BIT2,R2 ;SET UP EXPECTED
CMP R1,R2 ;ARE THEY EQUAL

```

TEST 6: REREADS

```

8000 066516 001406          BEQ      370$          ;BR, IF OK
8001 066520 005237 002212  INC      FATFLG          ;BUMP COUNT
8005 066524          ERRHRD  ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      066524 104456          TRAP      C$ERHRD
      066526 001306          .WORD    710
      066530 074362          .WORD    T26TRL
      066532 012136          .WORD    PKTSSR
8006 066534          370$:  CKLOOP          ;LOOP IF SELECTED
      066534 104406          TRAP      C$CLP1
8007
8008
8009
8010
8011
8012
8013
      ;*****
      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
      ;*****
8014 066536 013701 071660  MOV      T26BFR+6,R1    ;GET MESSAGE BUFFER
8015 066542 010102          MOV      R1,R2          ;SET UP EXPECTED
8016 066544 052702 010000  BIS      @BIT12,R2     ;SET THE RLL BIT IN EXPECTED
8017 066550 020102          CMP      R1,R2          ;ARE THEY EQUAL
8018 066552 001406          BEQ      380$          ;BR, IF EQUAL (ALL IS WELL)
8019 066554 005237 002212  INC      FATFLG          ;BUMP COUNT
8023 066560          ERRHRD  ERRNO,T26LON,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
      066560 104456          TRAP      C$ERHRD
      066562 001307          .WORD    711
      066564 074130          .WORD    T26LON
      066566 015564          .WORD    EXPREC
8024 066570          380$:  CKLOOP
      066570 104406          TRAP      C$CLP1
8025 066572          ENDSUB          ;>>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>>
      066572 104403          L.10115;
      066572 104403          TRAP      C$ESUB
8026 066574 023727 002212 000017  CMP      FATFLG,@15.    ;IS ERROR COUNT AT 25
8027 066602 103402          BLO     999$          ;BR, IF LESS THAN 25
8028 066604 004737 017272  JSR      PC,CKDROP     ;TRY TO DROP THE UNIT
8029 066610          999$:

```



TEST 6: REREADS

```

8085 066706 012124
      066710 26$: CKLOOP ;LOOP IF SELECTED .WORD 5FIMSG
      066710 104406 TRAP C$CLP1
8086
8087 ;*****
8088 ;
8089 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8090 ;
8091 ;*****
8092
8093 066712 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
8094 066716 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
8095 066722 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
8096 066726 103407 BCS 30$ ;BR, IF NO PROBLEM
8097 066730 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
8098 066732 005237 002212 INC FATFLG ;BUMP COUNT
8102 066736 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      066736 104456 TRAP C$ERRHRD
      066740 001312 .WORD 714
      066742 073304 .WORD T26RWN
      066744 012136 .WORD PKTSSR
8103 066746 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      066746 104406
8104
8105 ;*****
8106 ;
8107 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8108 ;
8109 ;*****
8110
8111 066750 013701 071660 MOV T26FR+6,R1 ;PICK UP XSTO
8112 066754 010102 MOV R1,R2 ;SET UP EXPECTED
8113 066756 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
8114 066762 020102 CMP R1,R2 ;DOES EXP = REC'D
8115 066764 001406 BEQ 40$ ;BR, IF EQUAL (OK)
8116 066766 005237 002212 INC FATFLG ;BUMP COUNT
8120 066772 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      066772 104456 TRAP C$ERRHRD
      066774 001313 .WORD 715
      066776 073015 .WORD T26BOT
      067000 015564 .WORD EXPREC
8121 067002 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      067002 104406
8122 067004 012703 000400 MOV #256,R3 ;RECORD SIZE
8123 067010 013737 003114 071752 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
8124
8125 ;*****
8126 ;
8127 ;WRITE DATA,CVC=1,ACK COMMAND
8128 ;
8129 ;*****
8130
8131 067016 012737 140005 071750 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
8132 067024 012704 071750 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8133 067030
8134 067030 010337 071756 65$: MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
8135 067034 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND

```

TEST 6: REREADS

```

8136 067040 004737 016340      JSR      PC,WAITE          ;WAIT FOR SSR TO SET
8137 067044 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
8138 067050 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED
8139 067054 020102              CMP      R1,R2           ;ARE THEY EQUAL
8140 067056 001406              BEQ      75$             ;BR, IF OK
8141 067060 005237 002212      INC      FATFLG          ;BUMP COUNT
8145 067064              ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERRHRD
                                .WORD    716
                                .WORD    WRTERR
                                .WORD    PKTSSR
      067064 104456
      067066 001314
      067070 005111
      067072 012136
8146 067074              75$:   CKLOOP              ;LOOP IF SELECTED
      067074 104406
                                TRAP      C$CLP1
8147 067076              120$:
8148
8149
8150
8151
8152
8153
8154
      ;*****
      ;
      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
      ;
      ;*****
8155 067076 004737 011104      JSR      PC,REWIND        ;CALL TAPE REWIND COMMAND
8156 067102 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR
8157 067106 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED TSSR
8158 067112 103407              BCS      130$           ;BR, IF NO PROBLEM
8159 067114 010004              MOV      R0,R4           ;PACKET ADDRESS SET UP
8160 067116 005237 002212      INC      FATFLG          ;BUMP COUNT
8164 067122              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERRHRD
                                .WORD    717
                                .WORD    T26RWN
                                .WORD    PKTSSR
      067122 104456
      067124 001315
      067126 073304
      067130 012136
8165 067132              130$:  CKLOOP              ;LOOP IF SELECTED
      067132 104406
                                TRAP      C$CLP1
8166
8167
8168
8169
8170
8171
8172
      ;*****
      ;
      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
      ;
      ;*****
8173 067134 013701 071660      MOV      T26BFR+6,R1      ;PICK UP XST0
8174 067140 010102              MOV      R1,R2           ;SET UP EXPECTED
8175 067142 052702 000002      BIS      #BIT1,R2         ;SET BOT BIT IN EXPECTED
8176 067146 020102              CMP      R1,R2           ;DOES EXP = REC'D
8177 067150 001406              BEQ      135$           ;BR, IF EQUAL (OK)
8178 067152 005237 002212      INC      FATFLG          ;BUMP COUNT
8182 067156              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERRHRD
                                .WORD    718
                                .WORD    T26BOT
                                .WORD    EXPREC
      067156 104456
      067160 001316
      067162 073015
      067164 015564
8183 067166              135$:  CKLOOP              ;LOOP IF SELECTED
      067166 104406
                                TRAP      C$CLP1
8184 067170 012703 001000      MOV      #512,R3         ;RECORD SIZE
8185 067174 013737 003114 071752 MOV      FREE,T26R8       ;STARTING READ BUFFER ADDRESS
8186

```



TEST 6: REREADS

```

8187
8188
8189
8190
8191
8192
8193 067202 012737 161401 071750
8194 067210 012704 071750
8195 067214 010337 071756
8196 067220 010465 000000
8197 067224 004737 016340
8198 067230 016501 000002
8199 067234 012702 100204
8200 067240 020102
8201 067242 001406
8202 067244 005237 002212
8206 067250
      067250 104456
      067252 001317
      067254 074362
      067256 012136
8207 067260
      067260 104406
8209
8209
8210
8211
8212
8213
8214
8215 067262 013701 071660
8216 067266 010102
8217 067270 052702 040000
8218 067274 020102
8219 067276 001406
8220 067300 005237 002212
8224 067304
      067304 104456
      067306 001320
      067310 074212
      067312 015564
8225 067314
      067314 104406
8226 067316 013701 071656
8227 067322 012702 000400
8228 067326 020102
8229 067330 001405
8233 067334
      067334 104456
      067336 001320
      067340 074274
      067342 015564
8234 067344
      067344 104406
8235 067346 012703 001000
8236 067352 013737 003114 071752
8237

```

```

;*****
;REREAD NEXT,ACK,CVC=1,OPP=1
;*****
165$: MOV    #161401,T26PK3      ;REREAD NEXT,ACK,CVC=1,OPP=1
      MOV    #T26PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
      MOV    R3,T26SZ          ;SET UP RECORD SIZE IN PACKET
      MOV    R4,TSDB(R5)       ;ISSUE COMMAND
      JSR    PC,WAITF          ;WAIT FOR SSR TO SET
      MOV    TSSR(R5),R1       ;GET TSSR CONTENTS
      MOV    #SSR!SC!BIT2,R2   ;SET UP EXPECTED
      CMP    R1,R2             ;ARE THEY EQUAL
      BEQ    170$              ;BR, IF OK
      INC    FATFLG            ;BUMP COUNT
      ERRHRD ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP    C$ERHRD
                                .WORD   719
                                .WORD   T26TRL
                                .WORD   PKTSSR
170$: CKI OOP                    ;LOOP IF SELECTED
                                TRAP    C$CLP1
;*****
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;*****
180$: MOV    T26BFR+6,R1       ;GET MESSAGE BUFFER
      MOV    R1,R2             ;SET UP EXPECTED
      BIS    #BIT14,R2         ;SET THE RLS BIT IN EXPECTED
      CMP    R1,R2             ;ARE THEY EQUAL
      BEQ    180$              ;BR, IF EQUAL (ALL IS WELL)
      INC    FATFLG            ;BUMP COUNT
      ERRHRD ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
                                TRAP    C$ERHRD
                                .WORD   720
                                .WORD   T26LOP
                                .WORD   EXPREC
180$: CKLOOP
                                TRAP    C$CLP1
      MOV    T26BFR+4,R1       ;PICK UP RESIDUAL BYTE COUNTER
      MOV    #256.,R2          ;THIS SHOULD BE THE DIFFERENCE
      CMP    R1,R2             ;IS THE DIFFERENCE CORRECT
      BEQ    190$              ;BR, IF CORRECT
      ERRHRD ERRNO,T26PBP,EXPREC ;RBPCR NOT CORRECT
                                TRAP    C$ERHRD
                                .WORD   720
                                .WORD   T26PBP
                                .WORD   EXPREC
190$: CKLOOP                    ;LOOP IF SELECTED
                                TRAP    C$CLP1
      MOV    #512.,R3          ;RECORD SIZE
      MOV    FREE,T26RB        ;STARTING READ BUFFER ADDRESS

```



(1)

TEST 6: REREADS

```
8287 067526 023727 002212 000017      CMP    FATFLG,015.      ;IS ERROR COUNT AT 25
8288 067534 103402                      BLO    999$            ;BR, IF LESS THAN 25
8289 067536 004737 017272                      JSR    PC,CKDROP      ;TRY TO DROP THE UNIT
8290 067542                                      999$;
```

TEST 6: REREADS

8292  
 8293  
 8294  
 8295  
 8296  
 8297  
 8298  
 8299  
 8300  
 8301  
 8302  
 8303  
 8304  
 8305  
 8306  
 8307  
 8308  
 8309  
 8310 067542  
 067542  
 067542 104402  
 8311 067544 005737 003126  
 8312 067550 001002  
 8313 067552 000137 070534  
 8314 067556 004737 074540  
 8315 067562 005037 071776  
 8316 067566 004737 074632  
 8317 067572 004737 074674  
 8318  
 8319  
 8320  
 8321  
 8322  
 8323  
 8324  
 8325 067576 004737 016064  
 8326 067602 103407  
 8327 067604 005237 002212  
 8331 067610 010001  
 8332 067612  
 067612 104455  
 067614 001323  
 067616 003650  
 067620 012124  
 8333 067622 013737 002172 071650 20\$:  
 8334  
 8335 067630 012704 071630  
 8336  
 8337  
 8338  
 8339  
 8340  
 8341  
 8342  
 8343 067634 004737 010752  
 8344 067640 103407  
 8345 067642 005237 002212

```

; *
;
; TEST 6, SUBTEST 13
;
; VERIFIES THAT A DATA BUFFER ADDRESS REFERENCING
; NONEXISTENT MEMORY CAUSES RECOVERABLE ERROR
; TERMINATION (TC=4 OR 5) WITH NXM=1 AND THAT THE TAPE
; IS ULTIMATELY POSITIONED PROPERLY. ALL COMBINATIONS
; OF REREAD PREVIOUS/NEXT AND OPP=0/1 ARE TESTED.
; *****
;                       CAUTION
;
;       The LSI BUS drivers for all available address lines(16-21)
; are only checked when running on a 11/23B system with more than
; 128K words of memory!
; *****
; -
;
; BGNSUB                                ; >>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
;                                         T6.13:
;                                         TRAP     C$BSUB
;
;   TST     NXMFLG                         ; DO WE HAVE IT?
;   BNE     10$                            ; BR, IF ENOUGH
;   JMP     200$                           ; SKIP THIS TEST IF NOT
;10$:   JSR     PC,T26REST                  ; SET COMMAND PACKET
;       CLR     T26CNT                     ; CLEAR TAPE RECORD COUNTER
;       JSR     PC,T26RT2                  ; SET UP OTHER COMMAND PACKET
;       JSR     PC,T26RT3                  ; SET UP OTHER COMMAND PACKET
;
; *****
;
; ; ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
;
; *****
;
;       JSR     PC,SOFINIT                  ; DO INITIALIZE ON CONTROLLER
;       BCS     20$                        ; BR IF INIT WAS OK
;       INC     FATFLG                      ; BUMP COUNT
;       MOV     R0,R1                      ; CONTENTS OF TSSR REGISTER
;       ERRDF  ERRNO,SFIERR,SFIMSG        ; FATAL ERROR TSSR WAS NOT OK
;                                         TRAP     C$ERDF
;                                         .WORD   723
;                                         .WORD   SFIERR
;                                         .WORD   SFIMSG
;
;20$:   MOV     UNITN,T26DSW                ; SET UP UNIT NUMBER
;
;       MOV     @T26PACKET,R4              ; SUBROUTINE NEEDS PACKET ADDRESS
;
; *****
;
; ; WRITE CHARACTERISTICS COMMAND (CALL TO WRCHR)
;
; *****
;
;       JSR     PC,WRCHR                    ; ISSUE WRITE CHARACTERISTICS
;       BCS     26$                        ; BR, IF COMMAND ISSUED OK
;       INC     FATFLG                      ; BUMP COUNT

```

TEST 6: REREADS

```

8349 067646 010001          MOV     R0,R1          ;SAVE CONTENTS OF TSSR
8350 067650          ERRHRD  ERRNO,WRTMSG,SFMSG  ;WRITE CHARACTERISTICSC FAILED
      067650 104456          TRAP     C$ERHRD
      067652 001324          .WORD   724
      067654 005054          .WORD   WRTMSG
      067656 012124          .WORD   SFMSG
8351 067660          26$:   CKLOOP          ;LOOP IF SELECTED
      067660 104406          TRAP     C$CLP1
8352
8353          ;*****
8354          ;
8355          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8356          ;
8357          ;*****
8358
8359 067662 004737 021276      JSR     PC,INVERT      ;INVERT THE EXTENDED FEATURES SWITCH
8360 067666 004737 011104      JSR     PC,REWIND      ;CALL TAPE REWIND COMMAND
8361 067672 103411          BCS     30$           ;BR, IF NO PROBLEM
8362 067674 016501 000002      MOV     TSSR(R5),R1    ;GET TSSR
8363 067700 010004          MOV     R0,R4          ;PACKET ADDRESS SET UP
8364 067702 005237 002212      INC     FATFLG         ;BUMP COUNT
8368 067706          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      067706 104456          TRAP     C$ERHRD
      067710 001325          .WORD   725
      067712 073304          .WORD   T26RWN
      067714 012136          .WORD   PKTSSR
8369 067716          30$:   CKLOOP          ;LOOP IF SELECTED
      067716 104406          TRAP     C$CLP1
8370
8371          ;*****
8372          ;
8373          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8374          ;
8375          ;*****
8376
8377 067720 013701 071660      MOV     T26BFR+6,R1    ;PICK UP XSTO
8378 067724 010102          MOV     R1,R2          ;SET UP EXPECTED
8379 067726 052702 000002      BIS     #BIT1,R2       ;SET BOT BIT IN EXPECTED
8380 067732 020102          CMP     R1,R2          ;DOES EXP = REC'D
8381 067734 001406          BEQ     40$           ;BR, IF EQUAL (OK)
8382 067736 005237 002212      INC     FATFLG         ;BUMP COUNT
8386 067742          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      067742 104456          TRAP     C$ERHRD
      067744 001326          .WORD   726
      067746 073015          .WORD   T26BOT
      067750 015564          .WORD   EXPREC
8387 067752          40$:   CKLOOP          ;LOOP IF SELECTED
      067752 104406          TRAP     C$CLP1
8388 067754 013737 003114 071752  MOV     FREE,T26R8     ;STARTING WRITE BUFFER ADDRESS
8389
8390          ;*****
8391          ;
8392          ;WRITE DATA,CVC=1,ACK COMMAND
8393          ;
8394          ;*****
8395
8396 067762 012737 140005 071750  MOV     #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND

```

TEST 6: REREADS

```

8397 067770 012704 071750      MOV      #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
8398 067774 012737 000400 071756 65$:  MOV      #256.,T26SZ    ;SET UP RECORD SIZE IN PACKET
8399 070002 013777 071776 113104      MOV      T26CNT,#FREE   ;MOVE TAPE RECORD NUMBER TO BUFFER
8400 070010 062737 000001 071776      ADD      #1,T26CNT      ;NUMBER READY FOR NEXT RECORD
8401 070016 010465 000000      MOV      R4,TSD8(R5)    ;ISSUE COMMAND
8402 070022 004737 016340      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
8403 070026 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
8404 070032 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED
8405 070036 020102      CMP      R1,R2         ;ARE THEY EQUAL
8406 070040 001406      BEQ      75$           ;BR, IF OK
8407 070042 005237 002212      INC      FATFLG        ;BUMP COUNT
8411 070046      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      070046 104456      TRAP    C$ERHRD
      070050 001327      .WORD  727
      070052 005111      .WORD  WRTErr
      070054 012136      .WORD  PKTSSR
8412 070056      75$:  CKLOOP          ;LOOP IF SELECTED
      070056 104406      TRAP    C$CLP1
8413 070060 022737 000013 071776      CMP      #11.,T26CNT    ;CHECK NUMBER OF RECORDS WRITTEN
8414 070066 001401      BEQ      120$          ;BR, IF AT END OF WRITE SEQUENCE
8415 070070 000741      BR       65$           ;WRITE MORE RECORDS
8416 070072      120$:  CLR      T26CNT        ;SET RECORD COUNTER BACK TO ZERO
8417 070072 005037 071776
8418
8419      ;*****
8420      ;
8421      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8422      ;
8423      ;*****
8424
8425 070076 004737 011104      JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
8426 070102 103411      BCS     130$          ;BR, IF NO PROBLEM
8427 070104 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR
8428 070110 010004      MOV      R0,R4         ;PACKET ADDRESS SET UP
8429 070112 005237 002212      INC      FATFLG        ;BUMP COUNT
8433 070116      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      070116 104456      TRAP    C$ERHRD
      070120 001330      .WORD  728
      070122 073304      .WORD  T26RWN
      070124 012136      .WORD  PKTSSR
8434 070126      130$:  CKLOOP          ;LOOP IF SELECTED
      070126 104406      TRAP    C$CLP1
8435
8436      ;*****
8437      ;
8438      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8439      ;
8440      ;*****
8441
8442 070130 013701 071660      MOV      T26BFR+6,R1    ;PICK UP XSTO
8443 070134 010102      MOV      R1,R2         ;SET UP EXPECTED
8444 070136 052702 000002      BIS     #BIT1,R2       ;SET BOT BIT IN EXPECTED
8445 070142 020102      CMP     R1,R2         ;DOES EXP = REC'D
8446 070144 001406      BEQ     140$          ;BR, IF EQUAL (OK)
8447 070146 005237 002212      INC     FATFLG        ;BUMP COUNT
8451 070152      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      070152 104456      TRAP    C$ERHRD

```

## TEST 6: REREADS

```

      070154 001331
      070156 073015
      070160 015564
8452 070162 140$: CKLOOP
      070162 104406
8453 070164 012703 071766
8454 070170 013737 003130 071752
8455 070176 013737 003132 071754
8456
8457
8458
8459
8460
8461
8462
8463 070204 011337 071750
8464 070210 012704 071750
8465 070214 012737 000400 071756
8466 070222 010465 000000
8467 070226 004737 016340
8468 070232 016501 000002
8469 070236 012702 104210
8470 070242 020102
8471 070244 001414
8472 070246 031327 001000
8473 070252 001403
8474 070254 030127 000002
8475 070260 001006
8476 070262
8477 070262 005237 002212
8481 070266
      070266 104456
      070270 001332
      070272 072225
      070274 012136
8482 070276 170$: CKLOOP
      070276 104406
8483
8484
8485
8486
8487
8488
8489
8490 070300 012737 140001 071750
8491 070306 012737 000400 071756
8492 070314 003037 071754
8493 070320 013737 003114 071752
8494 070326 010465 000000
8495 070332 004737 016340
8496 070336 016501 000002
8497 070342 012702 000200
8498 070346 020102
8499 070350 001406
8500 070352 005237 002212
8504 070356
      070356 104456
      .WORD 729
      .WORD T26BOT
      .WORD EXPREC
      TRAP C$CLP1
      ;COMMAND BUFFER ADDRESS
      ;STARTING READ BUFFER ADDRESS
      ;SET UP HIGH ORDER ADDRESS BITS
      ;*****
      ;REREAD DATA,IE,ACK, OPP COMMAND
      ;*****
      MOV (R3),T26PK3 ;REREAD DATA,IE,ACK, OPP COMMAND
      MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
      MOV #256.,T26SZ ;SET UP RECORD SIZE IN PACKET
      MOV R4,TSDB(R5) ;ISSUE COMMAND
      JSR PC,WAITF ;WAIT FOR SSR TO SET
      MOV TSSR(R5),R1 ;GET TSSR CONTENTS
      MOV #SSR!NXM!SC!BIT3,R2 ;SET UP EXPECTED
      CMP R1,R2 ;ARE THEY EQUAL
      BEQ 170$ ;BR, IF OK
      BIT (R3),#BIT9 ;CHECK FOR A READ COMMAND
      BEQ 168$ ;BR, IF IT WAS A READ COMMAND
      BIT R1,#BIT1 ;WAS BIT1 SET
      BNE 170$ ;BR, IF REREAD AND BIT1 SET
      168$: INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      TRAP C$ERHRD
      .WORD 730
      .WORD T26RRF
      .WORD PKTSSR
      ;*****
      ;READ DATA, ACK,CVC=1 COMMAND
      ;*****
      MOV #140001,T26PK3 ;READ DATA, ACK,CVC=1 COMMAND
      MOV #256.,T26SZ ;SET SIZE INTO PACKET
      CLR T26RB+2 ;CLEAR OUT HIGH ADDRESS BITS
      MOV FREE,T26RB ;GIVE READ A GOOD BUFFER
      MOV R4,TSDB(R5) ;ISSUE READ DATA COMMAND
      JSR PC,WAITF ;WAIT FOR SSR
      MOV TSSR(R5),R1 ;PICK UP THE TSSR
      MOV #SSR,R2 ;SET UP EXPECTED
      CMP R1,R2 ;IS THE TSSR OK
      BEQ 180$ ;BR, IF TSSR OK (GOOD)
      INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,ROERR,PKTSSR ;READ DATA COMMAND FAILED
      TRAP C$ERHRD

```

TEST 6: REREADS

```

070360 001333 .WORD 731
070362 005204 .WORD RDERR
070364 012136 .WORD PKTSSR
8505 070366 180$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
070366 104406 ;FIRST WORD FROM READ BUFFER
8506 070370 017701 112520 MOV $FREE,R1 ;SET UP EXPECTED
8507 070374 012702 000001 MOV $1,R2 ;IS TAPE POSITION CORRECT
8508 070400 020102 CMP R1,R2 ;KEEP GOING POSITION OK
8509 070402 001406 BEQ 190$ ;BUMP COUNT
8510 070404 005237 002212 INC FATFLG ;TAPE POSITION INCORRECT
8514 070410 ERRHRD ERRNO,T26WNG,EXPREC TRAP C$ERHRD
070410 104456 .WORD 732
070412 001334 .WORD T26WNG
070414 072006 .WORD EXPREC
070416 015564
8515 070420 190$: CKLOOP TRAP C$CLP1
070420 104406
8516
8517 ;*****
8518 ;
8519 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8520 ;
8521 ;*****
8522
8523 070422 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
8524 070426 103411 BCS 194$ ;BR, IF NO PROBLEM
8525 070430 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
8526 070434 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
8527 070436 005237 002212 INC FATFLG ;BUMP COUNT
8531 070442 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
070442 104456 TRAP C$ERHRD
070444 001335 .WORD 733
070446 073304 .WORD T26RWN
070450 012136 .WORD PKTSSR
8532 070452 194$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
070452 104406
8533
8534 ;*****
8535 ;
8536 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8537 ;
8538 ;*****
8539
8540 070454 013701 071660 MOV T26BFR+6,R1 ;PICK UP XSTO
8541 070460 010102 MOV R1,R2 ;SET UP EXPECTED
8542 070462 052702 000002 BIS $B111,R2 ;SET BOT BIT IN EXPECTED
8543 070466 020102 CMP R1,R2 ;DOES EXP = REC'D
8544 070470 001406 BEQ 196$ ;BR, IF EQUAL (OK)
8545 070472 005237 002212 INC FATFLG ;BUMP COUNT
8549 070476 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
070476 104456 TRAP C$ERHRD
070500 001336 .WORD 734
070502 073015 .WORD T26BOT
070504 015564 .WORD EXPREC
8550 070506 196$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
070506 104406
8551 070510 010302 MOV R3,R2 ;SAVE R3 FOR A MOMENT
    
```







TEST 6: REREADS

```

8625
8626
8627
8628
8629
8630
8631
8632 070656 004737 011104 26$: JSR PC,REWIND ;CALL TAPE REWIND COMMAND
8633 070662 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
8634 070664 012702 000200 MOV *SSR,R2 ;SET UP EXPECTED TSSR
8635 070672 103407 BCS 50$ ;BR, IF NO PROBLEM
8636 070674 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
8637 070676 005237 002212 INC FATFLG ;BUMP COUNT
8641 070702 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      070702 104456 TRAP C$ERRHRD
      070704 001341 .WORD 737
      070706 073304 .WORD T26RWN
      070710 012136 .WORD PKTSSR
8642 070712 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      070712 104406
8643
8644
8645
8646
8647
8648
8649
8650 070714 013701 071660 MOV T26BFR+6,R1 ;PICK UP XSTO
8651 070720 010102 MOV R1,R2 ;SET UP EXPECTED
8652 070722 052702 000002 BIS *BIT1,R2 ;SET BOT BIT IN EXPECTED
8653 070726 020102 CMP R1,R2 ;DOES EXP = REC'D
8654 070730 001406 BEQ 40$ ;BR, IF EQUAL (OK)
8655 070732 005237 002212 INC FATFLG ;BUMP COUNT
8659 070736 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      070736 104456 TRAP C$ERRHRD
      070740 001342 .WORD 738
      070742 073015 .WORD T26BOT
      070744 015564 .WORD EXPREC
8660 070746 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      070746 104406
8661 070750 012737 000400 071756 MOV *256.,T26SZ ;SET UP RECORD SIZE IN PACKET
8662 070756 013737 003114 071752 MOV FREE,T26RB ;ADDRESS OF READ BUFFER
8663 070764 005703 TST R3 ;CHECK NUMBER OF TIMES THROUGH HERE
8664 070766 001404 BEQ 50$ ;BR, IF FIRST TIME THROUGH HERE
8665
8666
8667
8668
8669
8670
8671
8672 070770 012737 161001 071750 MOV *161001,T26PK3 ;REREAD,CVC=1,ACK COMMAND
8673 070776 000403 BR 55$ ;SKIP NEXT COMMAND
8674
8675
8676
8677

```

TEST 6: REREADS

```

8678
8679
8680
8681 071000 012737 141001 071750 50$: MOV 0141001,T26PK3 ;REREAD,ACK COMMAND
8682 071006 55$:
8683 071006 012704 071750 MOV 0T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8684 071012 65$:
8685 071012 010465 000000 MOV R4,TSD8(R5) ;ISSUE COMMAND
8686 071016 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
8687 071022 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8688 071026 012702 100206 MOV 0SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
8689 071032 020102 CMP R1,R2 ;ARE THEY EQUAL
8690 071034 001406 BEQ 75$ ;BR, IF OK
8691 071036 005237 002212 INC FATFLG ;BUMP COUNT
8695 071042 ERRHRD ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      071042 104456 TRAP C$ERHRD
      071044 001343 .WORD 739
      071046 072743 .WORD T26WDE
      071050 012136 .WORD PKTSSR
8696 071052 75$: CKLOOP ;LOOP IF SELECTED
      071052 104406 TRAP C$CLP1
8697
8698
8699
8700 ;*****
8701 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
8702 ;*****
8703
8704 071054 013701 071660 MOV T26BFR+6,R1 ;GET XST0 STATUS WORD
8705 071060 010102 MOV R1,R2 ;SET UP EXPECTED
8706 071062 052702 002000 BIS 0BIT10,R2 ;SET THE NEF BIT
8707 071066 020102 CMP R1,R2 ;ARE THEY EQUAL
8708 071070 001406 BEQ 170$ ;BR, IF EQUAL (GOOD)
8709 071072 005237 002212 INC FATFLG ;BUMP COUNT
8713 071076 ERRHRD ERRNO,T26NEF,EXPREC ;NEF SHOULD BE SET
      071076 104456 TRAP C$ERHRD
      071100 001344 .WORD 740
      071102 072074 .WORD T26NEF
      071104 015564 .WORD EXPREC
8714 071106 170$: CKLOOP TRAP C$CLP1
      071106 104406
8715 071110 005103 COM R3 ;RESET THE SWITCH
8716 071112 001261 BNE 26$ ;BR, IF FIRST TIME THROUGH HERE
8717 071114 ENDSUB
      071114 L10120:
      071114 104403 TRAP C$ESUB
8718 071116 023727 002212 000017 CMP FATFLG,015, ;IS ERROR COUNT AT 25
8719 071124 103402 BLO 999$ ;BR, IF LESS THAN 25
8720 071126 004737 017272 JSR PC,CKDROP ;TRY TO DROP THE UNIT
8721 071132 999$:

```



TEST 6: REREADS

```

071236 104406 TRAP C$CLP1
8776
8777 ;*****
8778 ;
8779 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8780 ;
8781 ;*****
8782
8783 071240 004737 011104 26$: JSR PC,REWIND ;CALL TAPE REWIND COMMAND
8784 071244 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
8785 071250 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
8786 071254 103407 BCS 30$ ;BR, IF NO PROBLEM
8787 071256 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
8788 071260 005237 002212 INC FATFLG ;BUMP COUNT
8792 071264 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
071264 104456 TRAP C$ERHRD
071266 001347 .WORD 743
071270 073304 .WORD T26RWN
071272 012136 .WORD PKTSSR
8793 071274 30$: CKLOOP ;LOOP IF SELECTED
071274 104406 TRAP C$CLP1
8794
8795 ;*****
8796 ;
8797 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
8798 ;
8799 ;*****
8800
8801 071276 013701 071660 MOV T26BFR+6,R1 ;PICK UP XST0
8802 071302 010102 MOV R1,R2 ;SET UP EXPECTED
8803 071304 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
8804 071310 020102 CMP R1,R2 ;DOES EXP = REC'D
8805 071312 001406 BEQ 40$ ;BR, IF EQUAL (OK)
8806 071314 005237 002212 INC FATFLG ;BUMP COUNT
8810 071320 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
071320 104456 TRAP C$ERHRD
071322 001350 .WORD 744
071324 073015 .WORD T26BOT
071326 015564 .WORD EXPREC
8811 071330 40$: CKLOOP
071330 104406 TRAP C$CLP1
8812
8813 ;*****
8814 ;
8815 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
8816 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
8817 ;
8818 ;*****
8819
8820 071332 012703 000001 MOV #000001,R3 ;SET UP SPACE FORWARD 1 RECORD
8821 071336 004737 010556 JSR PC,SPACE ;ISSUE SPACE COMMAND
8822 071342 103411 BCS 75$ ;BR, IF OK
8823 071344 016501 000002 MOV TSSR(R5),R1 ;GET STATUS DATA
8824 071350 010004 MOV R0,R4 ;GET PACKET ADDRESS
8825 071352 005237 002212 INC FATFLG ;BUMP COUNT
8829 071356 ERRHRD ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
071356 104456 TRAP C$ERHRD

```

TEST 6: REREADS

```

      071360 001351                                ,WORD 745
      071362 072743                                ,WORD T26WDE
      071364 012136                                ,WORD PKTSSR
8830 071366 104406      75$:   CKLOOP                                ;LOOP IF SELECTED
      071366 104406                                TRAP   C$CLP1
8831
8832
8833
8834
8835
8836
8837
8838
8839 071370 012703 100001      MOV     #100001,R3                ;SET SPACE REVERSE 1 RECORD
8840 071374 004737 010556      JSR     PC,SPACE                ;ISSUE COMMAND
8841 071400 103411      BCS     175$                     ;GO ON IF ALL IS WELL
8842 071402 016501 000002      MOV     TSSR(R5),R1            ;GET TSSR CONTENTS
8843 071406 010004      MOV     R0,R4                   ;SET UP EXPECTED (PACKET CONTENTS)
8844 071410 005237 002212      INC     FATFLG                 ;BUMP COUNT
8848 071414      ERRHRD  ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      071414 104456                                TRAP   C$ERHRD
      071416 001352                                ,WORD 746
      071420 072743                                ,WORD T26WDE
      071422 012136                                ,WORD PKTSSR
8849 071424      175$:   CKLOOP                                ;LOOP IF SELECTED
      071424 104406                                TRAP   C$CLP1
8850 071426 013737 003114 071752      MOV     FREE,T26RB             ;ADDRESS OF BUFFER
8851 071434 005737 072000      TST     T26CNU                 ;CHECK FOR TIMES THROUGH HERE
8852 071440 001404      BEQ     176$                     ;BR, IF FIRST TIME THROUGH
8853
8854
8855
8856
8857
8858
8859
8860 071442 012737 161001 071750      MOV     #161001,T26PK3        ;REREAD (PREVIOUS),IE,ACK,OPP=1 CMD.
8861 071450 000403      BR      178$                     ;SKIP NEXT COMMAND
8862
8863
8864
8865
8866
8867
8868
8869 071452 012737 141001 071750      176$:  MOV     #141001,T26PK3        ;REREAD ,ACK,OPP=1 COMMAND
8870 071460      178$:
8871 071460 012704 071750      MOV     #T26PK3,R4             ;SET UP R4 WITH PACKET ADDRESS
8872 071464 010465 000000      MOV     R4,TSD8(R5)           ;ISSUE COMMAND
8873 071470 004737 016340      JSR     PC,WAITF               ;WAIT FOR SSR TO SET
8874 071474 016501 000002      MOV     TSSR(R5),R1            ;GET TSSR CONTENTS
8875 071500 012702 100204      MOV     #SSR!SC!BIT2,R2        ;SET UP EXPECTED
8876 071504 020102      CMP     R1,R2                   ;ARE THEY EQUAL
8877 071506 001406      BEQ     180$                     ;BR, IF OK
8878 071510 005237 002212      INC     FATFLG                 ;BUMP COUNT
8882 071514      ERRHRD  ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      071514 104456                                TRAP   C$ERHRD

```





TEST 6: REREADS

8911			;		
8912			;	LOCAL STORAGE FOR THIS TEST	
8913			;		
8915		071630	;		
8917	071630		T26PACKET:		;
8918	071630	014004		.WORD 14004	;
8919	071632	071640		.WORD T26DATA	;
8920	071634	000000		.WORD 0	;
8921	071636	000012		.WORD 10.	;
8922	071640		T26DATA:		;
8923	071640	071652		.WORD T26BFR	;
8924	071642	000000		.WORD 0	;
8925	071644	000024		.WORD 20.	;
8926	071646	000000		.WORD 0	;
8927	071650	000000	T26DSW:	.WORD 0	;
8928	071652		T26BFR:	.BLKW 25.	;
8929			;		;
8930			;	WRITE SUBSYSTEM MEMORY COMMAND PACKET	
8931			;		
8933		071740	;		
8935	071740		T26PK2:		;
8936	071740	100006		.WORD 100006	;
8937	071742	071760		.WORD T26BF2	;
8938	071744	000000		.WORD 0	;
8939	071746	000006		.WORD 6.	;
8940					;
8944	071750		T26PK3:		;
8945	071750	140005		.WORD 140005	;
8946	071752		T26RB:		;
8947	071752	003114	T26WB:	.WORD FREE	;
8948	071754	000000		.WORD 0	;
8949	071756	000000	T26SZ:	.WORD 0	;
8950				.EVEN	;
8951			;		;
8952			;		;
8953			;		;
8954	071760		T26BF2:		;
8955	071760	010	T26BS0:	.BYTE 10	;
8956	071761	200	T26BS1:	.BYTE 200	;
8957	071762	000000	T26S2:	.WORD 0	;
8958	071764	000000	T26S3:	.WORD 0	;
8959			;		;
8960			;		;
8961			;	.EVEN	;
8962			;	TAPE MOTION PACKET COMMAND VALUES	
8963					
8964	071766	140001	T26RN:	.WORD 140001	;
8965	071770	141401		.WORD 141401	;
8966	071772	161401		.WORD 161401	;
8967	071774	177777		.WORD 177777	;
8968					;
8969			;		;
8970	071776	000000	T26CNT:	.WORD 0	;
8971	072000	000000	T26CNU:	.WORD 0	;
8972					;
8973	072002	000000	T26RSZ:	.WORD 0	;
8974					;

TSV7 - HARDWARE TESTS 1-8

MACRO M1200 23 MAR-84 09:44 PAGE 87-1

SEQ 0293

TEST 6: REREADS

8975 072004 000000  
8976

T26DLY: .WORD 0

;DELAY COUNTER AREA

TEST 6: REREADS

```

8978
8979
8980          ;*
8981          ;LOCAL TEXT MESSAGES FOR TEST
8982          ;-
8983
8984 072006    124    141    160 T26WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
8985 072074    122    105    122 T26NEF: .ASCIZ 'REREAD PREVIOUS, At BOT, Failed To Set NEF (XSTO)'
8986 072156    124    123    123 T26RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
8987 072225    122    105    122 T26RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
8988 072322    122    105    122 T26RRG: .ASCIZ 'REREAD Previous (Read Reverse, Space Forward) Command Failed'
8989 072417    120    117    123 T26SC:  .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
8990 072501    122    111    102 T26LOR: .ASCIZ 'RIA NOT SET AFTER READ REVERSE INTO BOT'
8991 072551    124    123    123 T26WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
8992 072626    111    154    154 T26LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
8993 072707    122    105    122 T26SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
8994 072743    124    123    123 T26WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
8995 073015    124    141    160 T26BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
8996 073062    104    141    164 T26DTA: .ASCIZ 'Data Written To Tape Not Equal To Data Read From Tape'
8997 073150    122    105    122 T26EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
8998 073227    124    123    123 T26TM:  .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
8999 073304    122    145    167 T26RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
9000 073353    122    101    115 T26RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
9001 073426    124    123    123 T26AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
9002 073475    104    162    151 T26OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
9003 073550    124    123    123 T26WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
9004 073640    124    123    123 T26WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
9005 073713    103    126    103 T26VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
9006 073766    124    123    102 T26BA:  .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
9007 074041    127    122    111 T26WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
9008 074130    122    145    141 T26LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XSTO'
9009 074212    122    145    141 T26LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XSTO'
9010 074274    122    145    163 T26PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
9011 074362    122    145    141 T26TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
9012 074450    104    141    164 T26NEQ: .ASCIZ 'Data REREAD From Tape Not Correct, After SWB=1'
9013 074527    122    145    162 TST26ID: .ASCIZ 'Rereads'
9014
9015          .EVEN
9016
9017          ;*
9018          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
9019          ;WRITE SUBSYSTEM MEMORY COMMAND
9020          ;-
9021
9022 074540    T26REST:
9023 074540          SAVREG
9024 074544    012701 071630          MOV     #T26PACKET,R1          ;SAVE THE REGISTERS
9025 074550    012721 140004          MOV     #140004,(R1)+         ;START OF THE PACKET
9026 074554    012721 071640          MOV     #T26DATA,(R1)+       ;WRITE SUBSYSTEM MEM. WITH ACK, CVC=1
9027 074560    005021                    CLR     (R1)+                 ;ADDRESS OF CHARAISTICS DATA BLOCK
9028 074562    012721 000012          MOV     #10,(R1)+            ;EXTENDED ADDRESS
9029 074566    012721 071652          MOV     #T26BFR,(R1)+        ;SIZE OF DATA BLOCK IN BYTES
9030 074572    005021                    CLR     (R1)+                 ;ADDRESS OF MESSAGE BUFFER
9031 074574    012721 000024          MOV     #20,(R1)+            ;LENGTH OF MESSAGE BUFFER
9032 074600    005021                    CLR     (R1)+
9033 074602    012711 000000          MOV     #0,(R1)              ;SELECT DRIVE ZERO (0)
9034 074606    012702 000030          MOV     #24,(R2)             ;NUMBER OF LOCATIONS TO BE CLEARED

```

TEST 6: REREADS

```

9035 074612 012762 177777 071652 64$: MOV #177777,T26BFR(R2) ;ALL ONE'S TO MESSAGE BUFFER
9036 074620 005742 TST -(R2) ;NEXT LOCATION
9037 074622 020227 000000 CMP R2,#0 ;CHECK FOR END OF LOOP
9038 074626 001371 BNE 64$ ;KEEP GOING UNTIL DONE
9039 074630 000207 RTS PC ;RETURN
9040
9041
9042 074632 T26RT2:
9043 074632 SAVREG ;SAVE THE REGISTERS
9044 074636 012701 071740 MOV #T26PK2,R1 ;START OF THE PACKET
9045 074642 012721 140006 MOV #140006,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1,
9046 074646 012721 071760 MOV #T26BF2,(R1)+ ;ADDRESS OF DATA BLOCK
9047 074652 005021 CLR (R1)+ ;EXTENDED ADDRESS
9048 074654 012721 000006 MOV #6,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
9049 074660 005021 CLR (R1)+
9050 074662 012701 071760 MOV #T26BF2,R1 ;POINT TO DATA SEL. AREA
9051 074666 005021 CLR (R1)+
9052 074670 005011 CLR (R1)
9053 074672 000207 RTS PC ;RETURN
9054 074674 T26RT3:
9055 074674 SAVREG ;SAVE THE REGISTERS
9056 074700 012701 071750 MOV #T26PK3,R1 ;START OF THE PACKET
9057 074704 012721 000000 MOV #0,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
9058 074710 012721 000000 MOV #0,(R1)+ ;ADDRESS OF DATA BLOCK
9059 074714 005021 CLR (R1)+ ;EXTENDED ADDRESS
9060 074716 012711 000000 MOV #0,(R1) ;SIZE OF DATA BLOCK IN BYTES
9061 074722 000207 RTS PC ;RETURN
9062 074724 ENDTST
074724
074724 104401 L10102: TRAP C$ETST

```



TEST 7: WRITE DATA RETRI

```

075040 005367 177772
075044 001375
075046 005367 177756
075052 001367
9118 075054 005337 101612      DEC      T27DLY      ;BUMP COUNTER
9119 075060 001356      BNE      10$        ;BR, IF COUNTER NOT DONE
9120 075062 005237 002212      INC      FATFLG     ;BUMP COUNT
9124 075066 010001      MOV      R0,R1      ;CONTENTS OF TSSR REGISTER
9125 075070      ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
075070 104455      TRAP     C$ERRDF
075072 001275      .WORD   701
075074 003650      .WORD   SFIERR
075076 012124      .WORD   SFIMSG
9126 075100 013737 002172 101460 20$:  MOV      UNITN,T27DSW ;SET UP DRIVE NUMBER
9127 075106 012704 101440      MOV      @T27PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
9128
9129      ;*****
9130      ;
9131      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
9132      ;
9133      ;*****
9134
9135 075112 004737 010752      JSR      PC,WRTCHR   ;ISSUE WRITE CHARACTERISTICS
9136 075116 103407      BCS      25$        ;BR, IF COMMAND ISSUED OK
9137 075120 005237 002212      INC      FATFLG     ;BUMP COUNT
9141 075124 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
9142 075126      ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
075126 104456      TRAP     C$ERRHRD
075130 001276      .WORD   702
075132 005054      .WORD   WRTMSG
075134 012124      .WORD   SFIMSG
9143 075136      25$:  CKLOOP      ;LOOP IF SELECTED
075136 104406      TRAP     C$CLP1
9144
9145      ;*****
9146      ;
9147      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9148      ;
9149      ;*****
9150
9151 075140 004737 011104      JSR      PC,REWIND  ;CALL TAPE REWIND COMMAND
9152 075144 103407      BCS      30$        ;BR, IF NO PROBLEM
9153 075146 010004      MOV      R0,R4      ;SET UP REWIND PACKET ADDRESS
9154 075150 005237 002212      INC      FATFLG     ;BUMP COUNT
9158 075154      ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
075154 104456      TRAP     C$ERRHRD
075156 001277      .WORD   703
075160 102765      .WORD   T27RWN
075162 012136      .WORD   PKTSSR
9159 075164      30$:  CKLOOP      ;LOOP IF SELECTED
075164 104406      TRAP     C$CLP1
9160
9161      ;*****
9162      ;
9163      ;READ MESSAGE BUFFER EXPENDED STATUS REGISTER ZERO (XSTO)
9164      ;
9165      ;*****

```

TEST 7: WRITE DATA RETRY

```

9166
9167 075166 013701 101470      MOV     T27BFR+6,R1      ;PICK UP XSTO
9168 075172 010102              MOV     R1,R2           ;SET UP EXPECTED
9169 075174 052702 000002      BIS     #BIT1,R2        ;SET BOT BIT IN EXPECTED
9170 075200 020102              CMP     R1,R2           ;DOES EXP = REC'D
9171 075202 001406              BEQ     40$             ;BR, IF EQUAL (OK)
9172 075204 005237 002212      INC     FATFLG          ;BUMP COUNT
9176 075210              ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP     C$ERHRD
                                .WORD    704
                                .WORD    T27BOT
                                .WORD    EXPREC
                                075210 104456
                                075212 001300
                                075214 102461
                                075216 015564
9177 075220              40$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP     C$CLP1
                                075220 104406
9178 075222 012737 000400 101566  MOV     #256.,T27SZ      ;SET UP RECORD SIZE
9179 075230 013737 003114 101562  MOV     FREE,T27WC       ;ADDRESS OF WRITE BUFFER
9180
9181 ;*****
9182 ;
9183 ;WRITE DATA RETRY,ACK,CVC=1 COMMAND
9184 ;
9185 ;*****
9186
9187 075236 012737 141005 101560  MOV     #141005,T27PK3   ;WRITE DATA RETRY,ACK,CVC=1 COMMAND
9188 075244 012704 101560          MOV     #T27PK3,R4       ;SET UP R4 WITH PACKET ADDRESS
9189 075250 010465 070000          MOV     R4,TSD8(R5)      ;ISSUE COMMAND
9190 075254 05737 016340          JSR     PC,WAITF         ;WAIT FOR SSR TO SET
9191 075260 016501 000002          MOV     TSSR(R5),R1      ;GET TSSR CONTENTS
9192 075264 012702 100206          MOV     #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
9193 075270 020102              CMP     R1,R2           ;ARE THEY EQUAL
9194 075272 001406              BEQ     75$             ;BR, IF OK
9195 075274 005237 002212      INC     FATFLG          ;BUMP COUNT
9199 075300              ERRHRD  ERRNO,T27WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP     C$ERHRD
                                .WORD    705
                                .WORD    T27WDE
                                .WORD    PKTSSR
                                075300 104456
                                075302 001301
                                075304 102372
                                075306 012136
9200 075310              75$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP     C$CLP1
                                075310 104406
9201
9202 ;*****
9203 ;
9204 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9205 ;
9206 ;*****
9207
9208 075312 013701 101470      MOV     T27BFR+6,R1      ;GET XSTO STATUS WORD
9209 075316 010102              MOV     R1,R2           ;SET UP EXPECTED
9210 075320 052702 002000      BIS     #BIT10,R2       ;SET THE NEF BIT
9211 075324 020102              CMP     R1,R2           ;ARE THEY EQUAL
9212 075326 001406              BEQ     170$            ;BR, IF EQUAL (GOOD)
9213 075330 005237 002212      INC     FATFLG          ;BUMP COUNT
9217 075334              ERRHRD  ERRNO,T27NEF,EXPREC ;NEF SHOULD BE SET
                                TRAP     C$ERHRD
                                .WORD    706
                                .WORD    T27NEF
                                .WORD    EXPREC
                                075334 104456
                                075336 001302
                                075340 104131
                                075342 015564

```

TEST 7: WRITE DATA RETRI

```

9218 075344          1105:  CIO,DRP
      075344 104406
9219 075346          END,DR
      075346
      075346 104403
9220 075350 023727 002212 000017  CMP  FATH(6,015,
9221 075356 103402          HLD  9999
9222 075360 004737 017272          JSR  PC,CIO,DRP
9223 075364          9999:

```

TRAP C\$CLP1  
 110123: TRAP C\$ESUB  
 ;IF ERROR COUNT AT 25,  
 ;BR, IF LESS THAN 25  
 ;TRY TO DROP THE UNIT



TEST 7: WRITE DATA RETRY

```

9225
9226
9227
9228
9229
9230
9231
9232
9233
9234
9235
9236
9237
9238 075364          BGNSUB                      ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
      075364          ;>>>>>>>>>>>> T7.2:                      TRAP      C$B50B
      075364 104402
9239 075366 004737 104564      JSR      PC,T27REST          ;SET COMMAND PACKET
9240 075372 004737 104656      JSR      PC,T27RT2          ;SET UP OTHER COMMAND PACKET
9241 075376 004737 104720      JSR      PC,T27RT3          ;SET UP OTHER COMMAND PACKET
9242
9243
9244
9245
9246
9247
9248
9249 075402 004737 016064      JSR      PC,SOFINIT          ;DO INITIALIZE ON CONTROLLER
9250 075406 103407                BCS      20$                ;BR IF INIT WAS OK
9251 075410 005237 002212      INC      FATFLG            ;BUMP COUNT
9255 075414 010001                MOV      R0,R1              ;SAVE CONTENTS OF TSSR REGISTER
9256 075416                ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      075416 104455                TRAP      C$ERDF
      075420 001303                .WORD    70$
      075422 003650                .WORD    SFIERR
      075424 012124                .WORD    SFIMSG
9257 075426 013737 002172 101460 20$: MOV      UNITN,T27DSW          ;SET UP DRIVE NUMBER
9258 075434 012704 101440      MOV      @T27PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
9259
9260
9261
9262
9263
9264
9265
9266 075440 004737 010752      JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
9267 075444 103407                BCS      25$                ;BR, IF COMMAND ISSUED OK
9268 075446 005237 002212      INC      FATFLG            ;BUMP COUNT
9272 075452 010001                MOV      R0,R1              ;SAVE CONTENTS OF TSSR
9273 075454                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      075454 104456                TRAP      C$ERDF
      075456 001304                .WORD    70B
      075460 005054                .WORD    WRTMSG
      075462 012124                .WORD    SFIMSG
9274 075464                25$:  CKLOOP                      ;LOOP IF SELECTED
      075464 104406                TRAP      C$CLP1
9275
9276

```

TEST 7: WRITE DATA RETR

```

9277
9278 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9279 ;
9280 ;*****
9281
9282 075466 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
9283 075472 103411 BCS 26$ ;BR, IF NO PROBLEM
9284 075474 010004 MOV R0,R4 ;SET UP REWIND PACKET ADDRESS
9285 075476 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9286 075502 005237 002212 INC FATFLG ;BUMP COUNT
9290 075506 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERRHRD
; .WORD 709
; .WORD T27RWN
; .WORD PKTSSR
9291 075516 26$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
9292 075520 012703 000400 MOV #256.,R3 ;STARTING RECORD SIZE
9293 075524 013737 003114 101562 MOV FREE,T27WB ;STARTING WRITE BUFFER ADDRESS
9294
9295 ;*****
9296 ;
9297 ;WRITE DATA,CVC-1,ACK COMMAND
9298 ;
9299 ;*****
9300
9301 075532 012737 140005 101560 MOV #140005,T27PK3 ;WRITE DATA,CVC-1,ACK COMMAND
9302 075540 012704 101560 MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
9303 075544 010337 101566 MOV R3,T27SZ ;SET UP RECORD SIZE IN PACKET
9304 075550 010465 000000 MOV R4,T508(R5) ;ISSUE COMMAND
9305 075554 004737 016540 JSR PC,WAITF ;WAIT FOR SSR TO SET
9306 075560 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9307 075564 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9308 075570 020102 CMP R1,R2 ;ARE THEY EQUAL
9309 075572 001406 BEQ 28$ ;BR, IF OK
9310 075574 005237 002212 INC FATFLG ;BUMP COUNT
9314 075600 ERRHRD ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
; TRAP C$ERRHRD
; .WORD 710
; .WORD WRTErr
; .WORD PKTSSR
9315 075610 28$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
9316 075610 104406
9317
9318 ;*****
9319 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9320 ;
9321 ;*****
9322
9323 075612 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
9324 075616 103411 BCS 30$ ;BR, IF NO PROBLEM
9325 075620 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9326 075624 010004 MOV R0,R4 ;SET UP REWIND PACKET ADDRESS
9327 075626 005237 002212 INC FATFLG ;BUMP COUNT
9331 075632 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERRHRD
; .WORD 710
; .WORD WRTErr
; .WORD PKTSSR
075632 104456

```

TEST 7: WRITE DATA RETRY

```

075634 001307
075636 102765
075640 012136
9332 075642 30$: CKLOOP ;LOOP IF SELECTED
075642 104406 TRAP C$CLP1
9333
9334 ;*****
9335 ;
9336 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9337 ;
9338 ;*****
9339
9340 075644 013701 101470 MOV T27BFR+6,R1 ;PICK UP XSTC
9341 075650 010102 MOV R1,R2 ;SET UP EXPECTED
9342 075652 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
9343 075656 020102 CMP R1,R2 ;DOES EXP = REC'D
9344 075660 001406 BEQ 40$ ;BR, IF EQUAL (OK)
9345 075662 005237 002212 INC FATFLG ;BUMP COUNT
9349 075666 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
075666 104456 TRAP C$ERHRD
075670 001310 .WORD 712
075672 102461 .WORD T27BOT
075674 015564 .WORD EXPREC
9350 075676 40$: CKLOOP ;LOOP IF SELECTED
075676 104406 TRAP C$CLP1
9351
9352 ;*****
9353 ;
9354 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
9355 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
9356 ;
9357 ;*****
9358
9359 075700 012703 000001 MOV #1,R3 ;PARAMETER SPACE FORWARD 1 RECORD
9360 075704 004737 010556 JSR PC,SPACE ;CALL SPACE RECORDS ROUTINE
9361 075710 103413 BCS 50$ ;BR, IF NO ERRORS
9362 075712 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9363 075716 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9364 075722 010004 MOV R0,R4 ;SET UP REWIND PACKET ADDRESS
9365 075724 005237 002212 INC FATFLG ;BUMP COUNT
9369 075730 ERRHRD ERRNO,T27SCF,PKTSSR ;SPACE RECORDS COMMAND FAILED
075730 104456 TRAP C$ERHRD
075732 001311 .WORD 713
075734 104227 .WORD T27SCF
075736 012136 .WORD PKTSSR
9370 075740 50$: CKLOOP ;LOOP IF SELECTED
075740 104406 TRAP C$CLP1
9371
9372 ;*****
9373 ;
9374 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
9375 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
9376 ;
9377 ;*****
9378
9379 075742 012703 100001 MOV #100001,R3 ;PARAMETER SPACE REVERSE 1 RECORD
9380 075746 004737 010556 JSR PC,SPACE ;CALL SPACE RECORDS ROUTINE

```





TEST 7: WRITE DATA RETRY

```

9483 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9484 ;
9485 ;*****
9486 ;
9487 076250 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
9488 076254 103407 BCS 30$ ;BR, IF NO PROBLEM
9489 076256 010004 MOV R0,R4 ;SET UP REWIND PACKET ADDRESS
9490 076260 005237 002212 INC FATFLG ;BUMP COUNT
9494 076264 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
;
; TRAP C$ERHRD
; .WORD 719
; .WORD T27RWN
; .WORD PKTSSR
9495 076274 30$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
; .WORD
9496 ;
9497 ;*****
9498 ;
9499 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
9500 ;
9501 ;*****
9502 ;
9503 076276 013701 101470 MOV T27BFR+6,R1 ;PICK UP XST0
9504 076302 010102 MOV R1,R2 ;SET UP EXPECTED
9505 076304 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
9506 076310 020102 CMP R1,R2 ;DOES EXP = RECD
9507 076312 001406 BEQ 40$ ;BR, IF EQUAL (OK)
9508 076314 005237 002212 INC FATFLG ;BUMP COUNT
9512 076320 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
;
; TRAP C$ERHRD
; .WORD 720
; .WORD T27BOT
; .WORD EXPREC
9513 076330 40$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
; .WORD
9514 076332 012703 000024 MOV #20.,R3 ;STARTING RECORD SIZE
9515 076336 013737 003114 101562 MOV FREE,T27WB ;STARTING WRITE BUFFER ADDRESS
9516 ;
9517 ;*****
9518 ;
9519 ;WRITE DATA,CVC=1,ACK COMMAND
9520 ;
9521 ;*****
9522 ;
9523 076344 012737 140005 101560 65$: MOV #140005,T27PK3 ;WRITE DATA,CVC=1,ACK COMMAND
9524 076352 012704 101560 MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
9525 076356 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
9526 076360 004737 017512 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
9527 076364 010337 101566 MOV R3,T27SZ ;SET UP RECORD SIZE IN PACKET
9528 076370 010465 000000 MOV R4,T5DB(R5) ;ISSUE COMMAND
9529 076374 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
9530 076400 016501 000002 MOV T5SR(R5),R1 ;GET T5SR CONTENTS
9531 076404 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9532 076410 020102 CMP R1,R2 ;ARE THEY EQUAL
9533 076412 001406 BEQ 80$ ;BR, IF OK
9534 076414 005237 002212 INC FATFLG ;BUMP COUNT
9538 076420 ERRHRD ERRNO,WRTErr,PKTSSR ;T5SR INCORRECT AFTER WRITE DATA

```

TEST 7: WRITE DATA RETRY

```

076420 104456
076422 001321
076424 005111
076426 012136
9539 076430 80$: CKLOOP ;LOOP IF SELECTED
076430 104406 TRAP C$ERHRD
9540 TRAP C$CLP1
9541 ;*****
9542 ;
9543 ;WRITE DATA RETRY,CVC=1,ACK COMMAND
9544 ;
9545 ;*****
9546
9547 076432 012737 141005 101560 MOV #141005,T27PK3 ;WRITE DATA RETRY,CVC=1,ACK COMMAND
9548 076440 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
9549 076444 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
9550 076450 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9551 076454 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9552 076460 020102 CMP R1,R2 ;ARE THEY EQUAL
9553 076462 001406 BEQ 90$ ;BR, IF OK
9554 076464 005237 002212 INC FATFLG ;BUMP COUNT
9558 076470 ERRHRD ERRNO,T27WRF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA RETRY
076470 104456 TRAP C$ERHRD
076472 001322 .WORD 722
076474 104366 .WORD T27WRF
076476 012136 .WORD PKTSSR
9559 076500 90$: CKLOOP ;LOOP IF SELECTED
076500 104406 TRAP C$CLP1
9560 076502 005723 TST (R3)+ ;BUMP RECORD SIZE COUNTER
9561 076504 020327 000050 CMP R3,#40. ;AT 40 SIZE YET
9562 076510 001315 BNE 65$ ;BR, IF MORE RECORDS TO WRITE
9563
9564 ;*****
9565 ;
9566 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9567 ;
9568 ;*****
9569
9570 076512 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
9571 076516 103407 BCS 230$ ;BR, IF NO PROBLEM
9572 076520 010001 MOV R0,R1 ;SAVE TSSR
9573 076522 005237 002212 INC FATFLG ;BUMP COUNT
9577 076526 ERRHRD ERRNO,T27RWN,EXPREC ;REWIND NOT ACCEPTED
076526 104456 TRAP C$ERHRD
076530 001323 .WORD 723
076532 102765 .WORD T27RWN
076534 015564 .WORD EXPREC
9578 076536 230$: CKLOOP ;LOOP IF SELECTED
076536 104406 TRAP C$CLP1
9579
9580 ;*****
9581 ;
9582 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
9583 ;
9584 ;*****
9585
9586 076540 013701 101470 MOV T27BFR+6,R1 ;PICK UP XST0

```

TEST 7: WRITE DATA RETRI

```

9587 076544 010102          MOV      R1,R2          ;SET UP EXPECTED
9588 076546 052702 000002    BIS      @BIT1,R2       ;SET BOT BIT IN EXPECTED
9589 076552 020102          CMP      R1,R2          ;DOES EXP - REC'D
9590 076554 001406          BEQ      240$           ;BR, IF EQUAL (OK)
9591 076556 005237 002212    INC      FATFLG         ;BUMP COUNT
9595 076562          ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          076562 104456          TRAP      C$ERHRD
          076564 001324          .WORD    724
          076566 102461          .WORD    T27BOT
          076570 015564          .WORD    EXPREC
9596 076572          240$:  CKLOOP          ;LOOP IF SELECTED
          076572 104406          TRAP      C$CLP1
9597 076574 012703 000024    MOV      @20.,R3        ;STARTING RECORD SIZE
9598 076600 013737 003114 101562    MOV      FREE,T27RB     ;STARTING READ BUFFER ADDRESS
9599
9600          ;*****
9601          ;
9602          ;READ DATA,ACK COMMAND
9603          ;
9604          ;*****
9605
9606 076606 012737 100001 101560 265$:  MOV      @100001,T27PK3 ;READ DATA,ACK COMMAND
9607 076614 012704 101560    MOV      @T27PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
9608 076620 010337 101566    MOV      R3,T27SZ       ;SET UP RECORD SIZE IN PACKET
9609 076624 010465 000000    MOV      R4,TSDB(R5)    ;ISSUE COMMAND
9610 076630 004737 016340    JSR      PC,WAITF       ;WAIT FOR SSR TO SET
9611 076634 016501 000002    MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
9612 076640 012702 000200    MOV      @SSR,R2        ;SET UP EXPECTED
9613 076644 020102          CMP      R1,R2          ;ARE THEY EQUAL
9614 076646 001406          BEQ      280$           ;BR, IF OK
9615 076650 005237 002212    INC      FATFLG         ;BUMP COUNT
9619 076654          ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
          076654 104456          TRAP      C$ERHRD
          076656 001325          .WORD    725
          076660 005204          .WORD    RDERR
          076662 012136          .WORD    PKTSSR
9620 076664          280$:  CKLOOP          ;LOOP IF SELECTED
          076664 104406          TRAP      C$CLP1
9621 076666 013737 003114    MOV      FREE,R2        ;GET BUFFER ADDRESS
9622 076672 010304          MOV      R3,R4          ;GET RECORD SIZE
9623 076674 162704 000024    SUB      @20.,R4        ;POINT BACK TO 1ST RECORD
9624 076700 060204          285$:  ADD      R2,R4          ;POINT TO 1ST LOC IN BUFFER
9625 076702 021403          CMP      (R4),R3        ;DATA WRITTEN = READ
9626 076704 001410          BEQ      290$           ;BR, IF DATA OK (GOOD)
9627 076706 011401          MOV      (R4),R1        ;PICK UP BAD DATA
9628 076710 010302          MOV      R3,R2          ;SET UP EXPECTED
9629 076712 005237 002212    INC      FATFLG         ;BUMP COUNT
9633 076716          ERRHRD  ERRNO,T27DTA,EXPREC ;DATA IN BUFFER NOT CORRECT
          076716 104456          TRAP      C$ERHRD
          076720 001326          .WORD    726
          076722 104446          .WORD    T27DTA
          076724 015564          .WORD    EXPREC
9634 076726          290$:  CKLOOP          ;LOOP IF SELECTED
          076726 104406          TRAP      C$CLP1
9635 076730 005724          TST      (R4),          ;BUMP TO NEXT ADDRESS
9636 076732 160204          SUB      R2,R4          ;BACK TO RECORD SIZE
9637 076734 020403          CMP      R4,R3          ;AT END OF RECORD YET

```







TEST 7: WRITE DATA RETRY

```

9696 077116 005237 002212      INC     FATFLG      ;BUMP COUNT
9700 077122 010001              MOV     R0,R1      ;SAVE CONTENTS OF TSSR
9701 077124              ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
          077124 104456              TRAP   C$ERHRD
          077126 001330              .WORD 728
          077130 005054              .WORD WRTMSG
          077132 012124              .WORD SFIMSG
9702 077134 104406      23$:   CKLOOP              ;LOOP IF SELECTED
          077134 104406              TRAP   C$CLP1
9703
9704      ;*****
9705      ;
9706      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9707      ;
9708      ;*****
9709
9710 077136 004737 011104      JSR     PC,REWIND   ;CALL TAPE REWIND COMMAND
9711 077142 103411      BCS    30$         ;BR, IF NO PROBLEM
9712 077144 016501 000002      MOV     TSSR(R5),R1 ;GET TSSR CONTENTS
9713 077150 010004      MOV     R0,R4      ;GET PACKET ADDRESS
9714 077152 005237 002212      INC     FATFLG      ;BUMP COUNT
9718 077156              ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
          077156 104456              TRAP   C$ERHRD
          077160 001331              .WORD 729
          077162 102765              .WORD T27RWN
          077164 012136              .WORD PKTSSR
9719 077166 104406      30$:   CKLOOP              ;LOOP IF SELECTED
          077166 104406              TRAP   C$CLP1
9720
9721      ;*****
9722      ;
9723      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9724      ;
9725      ;*****
9726
9727 077170 013701 101470      MCV     T27BFR+6,R1 ;PICK UP XSTO
9728 077174 010102      MOV     R1,R2      ;SET UP EXPECTED
9729 077176 052702 000002      BIS    @BIT1,R2    ;SET BOT BIT IN EXPECTED
9730 077202 020102      CMP     R1,R2      ;DOES EXP = RECD
9731 077204 001406      BEQ    40$         ;BR, IF EQUAL (==)
9732 077206 005237 002212      INC     FATFLG      ;BUMP COUNT
9736 077212              ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          077212 104456              TRAP   C$ERHRD
          077214 001332              .WORD 730
          077216 102461              .WORD T27BOT
          077220 015564              .WORD EXPREC
9737 077222 104406      40$:   CKLOOP              ;LOOP IF SELECTED
          077222 104406              TRAP   C$CLP1
9738 077224 012703 000024      MOV     @20.,R3    ;STARTING RECORD SIZE
9739 077230 013737 003114 101562      MOV     FREE,T27WB ;STARTING WRITE BUFFER ADDRESS
9740
9741      ;*****
9742      ;
9743      ;WRITE DATA,CVC=1,ACK COMMAND
9744      ;
9745      ;*****
9746

```

TEST 7: WRITE DATA RETRY

```

9747 077236 012737 140005 101560 65$:  MOV      #140005,T27PK3      ;WRITE DATA,CVC=1,ACK COMMAND
9748 077244 012704 101560          MOV      #T27PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
9749 077250 010300          MOV      R3,R0          ;SET PATTERN IN CORRECT REGISTER
9750 077252 004737 017512          JSR      PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
9751 077256 010337 101566          MOV      R3,T27SZ       ;SET UP RECORD SIZE IN PACKET
9752 077262 010465 000000          MOV      R4,TSDB(R5)    ;ISSUE COMMAND
9753 077266 004737 016340          JSR      PC,WAITF       ;WAIT FOR SSR TO SET
9754 077272 016501 000002          MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
9755 077276 012702 000200          MOV      #SSR,R2       ;SET UP EXPECTED
9756 077302 020102          CMP      R1,R2         ;ARE THEY EQUAL
9757 077304 001406          BEQ      80$           ;BR, IF OK
9758 077306 005237 002212          INC      FATFLG        ;BUMP COUNT
9762 077312          ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
          077312 104456          TRAP    C$ERRHRD
          077314 001333          .WORD  731
          077316 005111          .WORD  WRTErr
          077320 012136          .WORD  PKTSSR
9763 077322          80$:  CKLOOP          ;LOOP IF SELECTED
          077322 104406          TRAP    C$CLP1
9764
9765          ;*****
9766          ;
9767          ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
9768          ;
9769          ;*****
9770
9771 077324 012737 111005 101560          MOV      #111005,T27PK3 ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
9772 077332 010465 000000          MOV      R4,TSDB(R5)    ;ISSUE COMMAND
9773 077336 004737 016340          JSR      PC,WAITF       ;WAIT FOR SSR TO SET
9774 077342 016501 000002          MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
9775 077346 012702 000200          MOV      #SSR,R2       ;SET UP EXPECTED
9776 077352 020102          CMP      R1,R2         ;ARE THEY EQUAL
9777 077354 001406          BEQ      90$           ;BR, IF OK
9778 077356 005237 002212          INC      FATFLG        ;BUMP COUNT
9782 077362          ERRHRD  ERRNO,T27WRF,EXPREC ;TSSR INCORRECT AFTER WRITE DATA RETRY
          077362 104456          TRAP    C$ERRHRD
          077364 001334          .WORD  732
          077366 104366          .WORD  T27WRF
          077370 015564          .WORD  EXPREC
9783 077372          90$:  CKLOOP          ;LOOP IF SELECTED
          077372 104406          TRAP    C$CLP1
9784 077374 005723          TST      (R3)+         ;BUMP RECORD SIZE COUNTER
9785 077376 020327 000050          CMP      R3,#40        ;AT 40 SIZE YET
9786 077402 001315          BNE      65$           ;BR, IF MORE RECORDS TO WRITE
9787
9788          ;*****
9789          ;
9790          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9791          ;
9792          ;*****
9793
9794 077404 004737 011104          JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
9795 077410 103411          BCS      230$          ;BR, IF NO PROBLEM
9796 077412 016501 000002          MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
9797 077416 010004          MOV      R0,R4         ;GET PACKET ADDRESS
9798 077420 005237 002212          INC      FATFLG        ;BUMP COUNT
9802 077424          ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED

```

TEST 7: WRITE DATA RETRY

```

077424 104456
077426 001335
077430 102765
077432 012136
9803 077434 230$: CKLOOP ;LOOP IF SELECTED
077434 104405 TRAP C$ERRRD
9804 TRAP C$CLP1
9805
9806 ;*****
9807 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9808 ;
9809 ;*****
9810
9811 077436 013701 101470 MOV T27BFR+6,R1 ;PICK UP XSTO
9812 077442 010102 MOV R1,R2 ;SET UP EXPECTED
9813 077444 052702 000002 BIS @BIT1,R2 ;SET BOT BIT IN EXPECTED
9814 077450 020102 CMP R1,R2 ;DOES EXP = REC'D
9815 077452 001406 BEQ 240$ ;BR, IF EQUAL (OK)
9816 077454 005237 002212 INC FATFLG ;BUMP COUNT
9820 077460 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
077460 104456 TRAP C$ERRRD
077462 001336 .WORD 734
077464 102461 .WORD T27BOT
077466 015564 .WORD EXPREC
9821 077470 240$: CKLOOP ;LOOP IF SELECTED
077470 104406 TRAP C$CLP1
9822 077472 012703 000024 MOV @20.,R3 ;STARTING RECORD SIZE
9823 077476 013737 003114 101562 MOV FREE,T27RB ;STARTING READ BUFFER ADDRESS
9824
9825 ;*****
9826 ;
9827 ;READ DATA,ACK COMMAND
9828 ;
9829 ;*****
9830
9831 077504 012737 100001 101560 265$: MOV @100001,T27PK3 ;READ DATA,ACK COMMAND
9832 077512 012704 101560 MOV @T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
9833 077516 010337 101566 MOV R3,T27SZ ;SET UP RECORD SIZE IN PACKET
9834 077522 010465 000000 MOV R4,TSD8(R5) ;ISSUE COMMAND
9835 077526 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
9836 077532 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9837 077536 012702 000200 MOV @SSR,R2 ;SET UP EXPECTED
9838 077542 020102 CMP R1,R2 ;ARE THEY EQUAL
9839 077544 001406 BEQ 280$ ;BR, IF OK
9840 077546 005237 002212 INC FATFLG ;BUMP COUNT
9844 077552 ERRHRD ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
077552 104456 TRAP C$ERRRD
077554 001337 .WORD 735
077556 005204 .WORD RDERR
077560 012136 .WORD PKTSSR
9845 077562 280$: CKLOOP ;LOOP IF SELECTED
077562 104406 TRAP C$CLP1
9846 077564 013702 003114 MOV FREE,R2 ;GET BUFFER ADDRESS
9847 077570 010304 MOV R3,R4 ;GET RECORD SIZE
9848 077572 162704 000024 SUB @20.,R4 ;POINT BACK TO 1ST RECORD
9849 077576 060204 285$: ADD R2,R4 ;POINT TO 1ST LOC IN BUFFER
9850 077600 000303 SWAB R3 ;SWAP BYTES SWB=1 ETC.

```



TEST 7: WRITE DATA RETRY

9876  
9877  
9878  
9879  
9880  
9881  
9882  
9883  
9884  
9885  
9886  
9887  
9888  
9889  
9890  
9891  
9892  
9893  
9894  
9895  
9896  
9897  
9898  
9899  
9900  
9901  
9902  
9903  
9904  
9905  
9906  
9907  
9908  
9909  
9910  
9911

;  
;  
;TEST 7, SUBTEST 5  
;  
;VERIFIES THAT A WRITE DATA RETRY COMMAND IS  
;PERFORMING THE "ERASE" PART OF THE OPERATION BY  
;PERFORMING THE FOLLOWING SERIES OF STEPS.  
;  
;1. THE TAPE IS REWOUND AND A SERIES OF RECORDS ARE  
; WRITTEN WITH THE NORMAL WRITE DATA COMMAND. THIS  
; SHOULD RESULT IN RECORDS SEPARATED BY THE  
; STANDARD INTERRECORD GAP.  
;  
;2. A PROGRAM TIMING VALUE IS CALIBRATED BY REWINDING  
; THE TAPE AND THEN COUNTING THE NUMBER OF CYCLES  
; THROUGH A PROGRAMMED LOOP REQUIRED TO SPACE OVER  
; THE SERIES OF RECORDS WRITTEN IN PREVIOUS  
; STEP  
;  
;3. THE TAPE IS AGAIN REWOUND AND THE SAME SERIES OF  
; RECORDS WRITTEN AGAIN, THIS TIME USING THE WRITE  
; DATA RETRY COMMAND. THIS SHOULD RESULT IN  
; RECORDS SEPARATED BY A LONG INTERRECORD GAP.  
;  
;4. THE TAPE IS AGAIN REWOUND, THE SPACING COMMAND  
; ISSUED, AND THE NUMBER OF TIMMING LOOP CYCLES  
; COUNTED TO COMPLETE THE OPERATION.  
;  
;5. THE TWO LOOP COUNTS ARE COMPARED, CHECKING TO SEE  
; THAT THEY DIFFER BY A SIGNIFICANT AMOUNT.  
;  
;  
;  
;  
;  
;-

```
9912 077672          BGNSUB          ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
      077672          T7.5:
      077672 104402          TRAP      C$BSUB
9913 077674 005037 002214      CLR      INTRECV          ; INTERRUPT INDICATOR
9914 077700 005037 101606      CLR      T27CNT          ; TIMER FOR WRITE DATA SPACING
9915 077704 005037 101610      CLR      T27CNU          ; TIMER FOR WRITE DATA RETRY SPACING
9916 077710 004737 104564      JSR      PC,T27REST      ; SET COMMAND PACKET
9917 077714 004737 104656      JSR      PC,T27RT2      ; SET UP OTHER COMMAND PACKET
9918 077720 004737 104720      JSR      PC,T27RT3      ; SET UP OTHER COMMAND PACKET
9919 077724 012737 176750 101612  MOV      #65000.,T27DLY      ; SET UP DELAY COUNTER
9920
9921 ;*****
9922 ;
9923 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
9924 ;
9925 ;*****
9926
9927 077732 004737 016064      10$: JSR      PC,SOFINIT          ;DO INITIALIZE ON CONTROLLER
9928 077736 103426          BCS      20$          ;BR IF INIT WAS OK
9929 077740          DELAY      250          ;DELAY ABOUT .25 SEC
      077740 012727 000250          MOV      #250,(PC)+
```

TEST 7: WRITE DATA RETRI

```

077744 000000
077746 013727 002116
077752 000000
077754 005367 177772
077760 001375
077762 005367 177756
077766 001367
9930 077770 005337 101612
9931 077774 001356
9932 077776 005237 002212
9936 100002 010001
9937 100004
      100004 104455
      100006 001341
      100010 003650
      100012 012124
9938 100014 013737 002172 101460 20$:
9939
9940 100022 012704 101440
9941
9942
9943
9944
9945
9946
9947
9948 100026 004737 010752
9949 100032 103407
9950 100034 005237 002212
9954 100040 010001
9955 100042
      100042 104456
      100044 001342
      100046 005054
      100050 012124
9956 100052
      100052 104406
9957
9958
9959
9960
9961
9962
9963
9964 100054 004737 011104
9965 100060 103411
9966 100062 016501 000002
9967 100066 010004
9968 100070 005237 002212
9972 100074
      100074 104456
      100076 001343
      100100 102765
      100102 012136
9973 100104
      100104 104406
9974

```

```

      .WORD 0
      MOV 1$DLY,(PC)
      .WORD 0
      DEC -6(PC)
      BNE .-4
      DEC -22(PC)
      BNE .-20
      DEC T27DLY ;BUMP COUNTER
      BNE 10$ ;BR, IF COUNTER NOT DONE
      INC FATFLG ;BUMP COUNT
      MOV R0,R1 ;CONTENTS OF TSSR REGISTER
      ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      TRAP C$ERDF
      .WORD 737
      .WORD SFIERR
      .WORD SFIMSG
      MOV UNITN,T27DSW ;SET UP UNIT NUMBER
      MOV #T27PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
;*****
;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
;*****
      JSR PC,WRTPHR ;ISSUE WRITE CHARACTERISTICS
      BCS 23$ ;BR, IF COMMAND ISSUED OK
      INC FATFLG ;BUMP COUNT
      MOV R0,R1 ;SAVE CONTENTS OF TSSR
      ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      TRAP C$ERHRD
      .WORD 738
      .WORD WRTMSG
      .WORD SFIMSG
      CKLOOP ;LOOP IF SELECTED
      TRAP C$CLP1
;*****
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;*****
      JSR PC,REWIND ;CALL TAPE REWIND COMMAND
      BCS 30$ ;BR, IF NO PROBLEM
      MOV TSSR(R5),R1 ;GET TSSR CONTENTS
      MOV R0,R4 ;GET PACKET ADDRESS
      INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
      TRAP C$ERHRD
      .WORD 739
      .WORD T27RWN
      .WORD PKTSSR
      CKLOOP ;LOOP IF SELECTED
      TRAP C$CLP1

```



TEST 7: WRITE DATA RETRY

```

9975 ;*****
9976 ;
9977 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9978 ;
9979 ;*****
9980
9981 100106 013701 101470      MOV      T27BFR+6,R1      ;PICK UP XSTO
9982 100112 010102          MOV      R1,R2           ;SET UP EXPECTED
9983 100114 052702 000002    BIS      #BIT1,R2       ;SET BOT BIT IN EXPECTED
9984 100120 020102          CMP      R1,R2           ;DOES EXP = REC'D
9985 100122 001406          BEQ      40$            ;BR, IF EQUAL (OK)
9986 100124 005237 002212    INC      FATFLG         ;BUMP COUNT
9990 100130          ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    740
                                .WORD    T27BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
100130 104456
100132 001344
100134 102461
100136 015564
9991 100140          40$:  CKLOOP                ;LOOP IF SELECTED
100140 104406          TRAP      C$CLP1
9992 100142 012703 000144    MOV      #100.,R3       ;NUMBER OF RECORDS TO BE WRITTEN
9993 100146 013737 003114 101562  MOV      FREE,T27WB     ;STARTING WRITE BUFFER ADDRESS
9994
9995 ;*****
9996 ;
9997 ;WRITE DATA,ACK,CVC=1 COMMAND
9998 ;
9999 ;*****
10000
10001 100154 012737 140005 101560 65$:  MOV      #140005,T27PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
10002 100162 012704 101560    MOV      #T27PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
10003 100166 012737 000024 101566    MOV      #20.,T27SZ    ;SET UP RECORD SIZE IN PACKET
10004 100174 010465 000000    MOV      R4,T5DB(R5)   ;ISSUE COMMAND
10005 100200 004737 016340    JSR      PC,WAITF      ;WAIT FOR SSR TO SET
10006 100204 016501 000002    MOV      TSSR(R5),R1   ;SET TSSR CONTENTS
10007 100210 012702 000200    MOV      #SSR,R2       ;SET UP EXPECTED
10008 100214 020102          CMP      R1,R2         ;ARE THEY EQUAL
10009 100216 001406          BEQ      70$            ;BR, IF OK
10010 100220 005237 002212    INC      FATFLG         ;BUMP COUNT
10014 100224          ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    741
                                .WORD    WRTErr
                                .WORD    PKTSSR
                                TRAP      C$CLP1
100224 104456
100226 001345
100230 005111
100232 012136
10015 100234          70$:  CKLOOP                ;LOOP IF SELECTED
100234 104406          TRAP      C$CLP1
10016 100236 005303          DEC      R3             ;DEC RECORD COUNTER
10017 100240 001345          BNE     65$            ;BR, IF MORE RECORDS TO WRITE
10018
10019 ;*****
10020 ;
10021 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10022 ;
10023 ;*****
10024
10025 100242 004737 011104    JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
10026 100246 103411          BCS     130$           ;BR, IF NO PROBLEM
10027 100250 016501 000002    MOV      TSSR(R5),R1   ;GET TSSR CONTENTS

```

TEST 7: WRITE DATA RETRY

```

10028 100254 010004          MOV     R0,R4          ;GET PACKET ADDRESS
10029 100256 005237 002212  INC     FATFLG        ;BUMP COUNT
10033 100262          ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
10034 100272          130$:  CKLOOP          ;LOOP IF SELECTED
10035 100272 104406          TRAP   C$ERHRD
10036          .WORD   742
10037          .WORD   T27RWN
10038          .WORD   PKTSSR
10039          TRAP   C$CLP1
10040          ;*****
10041          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
10042          ;*****
10042 100274 013701 101470  MOV     T27BFR+6,R1    ;PICK UP XSTO
10043 100300 010102          MOV     R1,R2          ;SET UP EXPECTED
10044 100302 052702 000002  BIS     #BIT1,R2       ;SET BOT BIT IN EXPECTED
10045 100306 020102          CMP     R1,R2          ;DOES EXP = REC'D
10046 100310 001406          BEQ    140$           ;BR, IF EQUAL (OK)
10047 100312 005237 002212  INC     FATFLG        ;BUMP COUNT
10051 100316          ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
10052 100326          140$:  CKLOOP          ;LOOP IF SELECTED
10053 100330 012704 101560  TRAP   C$ERHRD
10054 100334 012737 000010 101562  MOV     #T27PK3,R4     ;SET UP PACKET ADDRESS
10055          MOV     #10,T27RB    ;SET UP RECORDS TO SPACE OVER
10056          ;*****
10057          ;ACK,CVC=1,SPACE FORWARD COMMAND
10058          ;*****
10062 100342 012737 140010 101560  MOV     #140010,T27PK3 ;ACK,CVC=1,SPACE FORWARD COMMAND
10063 100350 010465 000000 150$:  MOV     R4,TSDB(R5) ;ISSUE COMMAND
10064 100354 005237 101606 152$:  INC     T27CNT     ;BUMP TIMER
10065 100360          DELAY  1              ;DELAY ABOUT 100US
10066 100360 012727 000001          MOV     #1,(PC)+
10067 100364 000000          .WORD  0
10068 100366 013727 002116          MOV     L$DLY,(PC)+
10069 100372 000000          .WORD  0
10070 100374 005367 177772          DEC     -6(PC)
10071 100400 001375          BNE    .-4
10072 100402 005367 177756          DEC     -22(PC)
10073 100406 001367          BNE    .-20
10074 100410 016501 000002          MOV     TSSR(R5),R1   ;GET TSSR
10075 100414 032701 000200          BIT     #BIT7,R1     ;CHECK FOR TSSR'S SSR SET
10076 100420 001755          BEQ    152$           ;KEEP COUNTING UNTIL SET
10077 100422 016501 000002          MOV     TSSR(R5),R1   ;GET STATUS FROM TSSR
10078 100426 012702 000200          MOV     #SSR,R2       ;SET UP EXPECTED
10079 100432 020301          CMP     R2,R1         ;WAS EVERYTHING OK
10080 100434 001406          BEQ    160$           ;BR, IF ALL IS WELL

```

TEST 7: WRITE DATA RETRI

```

10073 100436 005237 002212      INC      FATFLG      ;BUMP COUNT
10077 100442      ERRHRD  ERRNO,T27SCF,PKTSSR ;SPACE FORWARD DIDN'T WORK OUT
      100442 104456      TRAP      C$ERHRD
      100444 001350      .WORD    744
      100446 104227      .WORD    T27SCF
      100450 012136      .WORD    PKTSSR
10078 100452      160$:  CKLOOP      ;LOOP IF SELECTED
      100452 104406      TRAP      C$CLP1
10079
10080      ;*****
10081      ;
10082      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10083      ;
10084      ;*****
10085
10086 100454 004737 011104      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
10087 100460 004737 016426      JSR      PC,CHKTSSR     ;SEE HOW TSSR IS
10088 100464 103407      BCS      170$           ;BR, IF NO PROBLEM
10089 100466 010001      MOV      R0,R1          ;SAVE TSSR
10090 100470 005237 002212      INC      FATFLG      ;BUMP COUNT
10094 100474      ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
      100474 104456      TRAP      C$ERHRD
      100476 001351      .WORD    745
      100500 102765      .WORD    T27RWN
      100502 012136      .WORD    PKTSSR
10095 100504      170$:  CKLOOP      ;LOOP IF SELECTED
      100504 104406      TRAP      C$CLP1
10096
10097      ;*****
10098      ;
10099      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
10100      ;
10101      ;*****
10102
10103 100506 013701 101470      MOV      T27BFR+6,R1    ;PICK UP XST0
10104 100512 010102      MOV      R1,R2          ;SET UP EXPECTED
10105 100514 052702 000002      BIS      @BIT1,R2       ;SET BOT BIT IN EXPECTED
10106 100520 020102      CMP      R1,R2          ;DOES EXP = REC'D
10107 100522 001406      BEQ      175$           ;BR, IF EQUAL (OK)
10108 100524 005237 002212      INC      FATFLG      ;BUMP COUNT
10112 100530      ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      100530 104456      TRAP      C$ERHRD
      100532 001352      .WORD    746
      100534 102461      .WORD    T27BOT
      100536 015564      .WORD    EXPREC
10113 100540      175$:  CKLOOP      ;LOOP IF SELECTED
      100540 104406      TRAP      C$CLP1
10114 100542 012703 000144      MOV      @100.,R3       ;STARTING RECORD SIZE
10115 100546 013737 003114 101562 177$:  MOV      FREE,T27WB     ;STARTING WRITE BUFFER ADDRESS
10116
10117      ;*****
10118      ;
10119      ;WRITE DATA,CVC=1,ACK COMMAND
10120      ;
10121      ;*****
10122
10123 100554 012737 140005 101560      MOV      @140005,T27PK3 ;WRITE DATA.CVC=1,ACK COMMAND

```

TEST 7: WRITE DATA RETRY

```

10124 100562 012704 101560      MOV      #T27PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
10125 100566 012737 000024 101566  MOV      #20.,T27SZ     ;SET UP RECORD SIZE IN PACKET
10126 100574 010465 000000      MOV      R4,T27SDB(R5)  ;ISSUE COMMAND
10127 100600 004737 016340      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
10128 100604 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
10129 100610 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
10130 100614 020102              CMP      R1,R2          ;ARE THEY EQUAL
10131 100616 001406              BEQ      180$           ;BR, IF OK
10132 100620 005237 002212      INC      FATFLG         ;BUMP COUNT
10136 100624              ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERRRD
                                .WORD    747
                                .WORD    WRTErr
                                .WORD    PKTSSR
                                TRAP      C$CLP1
10137 100634 104456      180$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
10138 100636 005303              DEC      R3             ;COUNT NUMBER OF RECORDS
10139 100640 001342              BNE     177$           ;BR, IF MORE RECORDS TO WRITE
10140
10141      ;*****
10142      ;
10143      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10144      ;
10145      ;*****
10146
10147 100642 004737 011104      JSR      PC,REWIND      ;ISSUE REWIND
10148 100646 103411              BCS     182$           ;BR, IF ALL IS WELL
10149 100650 010004              MOV     R0,R4          ;GET PACKET ADDRESS
10150 100652 016501 000002      MOV     TSSR(R5),R1    ;GET TSSR CONTENTS
10151 100656 005237 002212      INC     FATFLG         ;BUMP COUNT
10155 100662              ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND FAILED
                                TRAP      C$ERRRD
                                .WORD    748
                                .WORD    T27RWN
                                .WORD    PKTSSR
10156 100672 104406      182$:  CKLOOP           ;SELECT LOOP MAYBE
                                TRAP      C$CLP1
10157
10158      ;*****
10159      ;
10160      ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
10161      ;BIT 15 SETS DIRECTION - 0=FORWARD  1=REVERSE
10162      ;
10163      ;*****
10164
10165 100674 012703 000001      MOV     #1.,R3         ;SPACE 1 RECORD FORWARD
10166 100700 004737 010556      JSR     PC,SPACE       ;ISSUE SPACE COMMAND
10167 100704 103411              BCS     185$           ;BR, IF COMMAND OK
10168 100706 010004              MOV     R0,R4          ;GET PACKET ADDRESS
10169 100710 016501 000002      MOV     TSSR(R5),R1    ;GET TSSR STATUS
10170 100714 005237 002212      INC     FATFLG         ;BUMP COUNT
10174 100720              ERRHRD  ERRNO,T27SCF,PKTSSR ;SPACE FORWARD COMMAND FAILED
                                TRAP      C$ERRRD
                                .WORD    749
                                .WORD    T27SCF
                                .WORD    PKTSSR
10175 100730              185$:  CKLOOP           ;LOOP IF SELECTED

```

TEST 7: WRITE DATA RETRY

```

10176 100730 104400 000144      MOV      #100.,R3          ;NUMBER OF RECORDS TO BE WRITTEN
10177 100732 012703 000144      MOV      #FREE,T27WB      ;STARTING WRITE BUFFER ADDRESS
10178
10179 ;*****
10180 ;
10181 ;WRITE DATA RETRY,ACK COMMAND
10182 ;
10183 ;*****
10184
10185 100744 012737 101005 101560 190$:  MOV      #101005,T27PK3    ;WRITE DATA RETRY,ACK COMMAND
10186 100752 012704 101560      MOV      #T27PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
10187 100756 012737 000024 101566  MOV      #20.,T27SZ      ;SET UP RECORD SIZE IN PACKET
10188 100764 010465 000000      MOV      R4,TSD8(R5)     ;ISSUE COMMAND
10189 100770 004737 016340      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
10190 100774 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
10191 101000 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
10192 101004 020102      CMP      R1,R2          ;ARE THEY EQUAL
10193 101006 001406      BEQ      200$           ;BR, IF OK
10194 101010 005237 002212      INC      FATFLG         ;BUMP COUNT
10198 101014      ERRHRD  ERRNO,T27WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
10198 101014 104456      TRAP    C$ERRHRD
10198 101016 001356      .WORD  750
10198 101020 103321      .WORD  T27WDC
10198 101022 C12136      .WORD  PKTSSR
10199 101024      200$:  CKLOOP           ;LOOP IF SELECTED
10199 101024 104406      TRAP    C$CLP1
10200 101026 013737 003114 101562  MOV      #FREE,T27WB      ;STARTING WRITE BUFFER ADDRESS
10201
10202 ;*****
10203 ;
10204 ;WRITE DATA,CVC=1,ACK COMMAND
10205 ;
10206 ;*****
10207
10208 101034 012737 140005 101560      MOV      #140005,T27PK3  ;WRITE DATA,CVC=1,ACK COMMAND
10209 101042 012704 101560      MOV      #T27PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
10210 101046 012737 000024 101566  MOV      #20.,T27SZ      ;SET UP RECORD SIZE IN PACKET
10211 101054 010465 000000      MOV      R4,TSD8(R5)     ;ISSUE COMMAND
10212 101060 004737 016340      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
10213 101064 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
10214 101070 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
10215 101074 020102      CMP      R1,R2          ;ARE THEY EQUAL
10216 101076 001406      BEQ      210$           ;BR, IF OK
10217 101100 005237 002212      INC      FATFLG         ;BUMP COUNT
10221 101104      ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
10221 101104 104456      TRAP    C$ERRHRD
10221 101106 001357      .WORD  751
10221 101110 005111      .WORD  WRERR
10221 101112 012136      .WORD  PKTSSR
10222 101114      210$:  CKLOOP           ;LOOP IF SELECTED
10222 101114 104406      TRAP    C$CLP1
10223 101116 005303      DEC      R3             ;BUMP DOWN RECORD COUNTER
10224 101120 001311      BNE     190$           ;BR, IF MORE RECORDS TO WRITE RETRY
10225
10226 ;*****
10227 ;

```

TEST 7: WRITE DATA RETRY

```

10228 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10229 ;
10230 ;*****
10231 ;
10232 101122 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
10233 101126 103411 BCS 230$ ;BR, IF NO PROBLEM
10234 101130 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10235 101134 010004 MOV R0,R4 ;GET PACKET ADDRESS
10236 101136 005237 002212 INC FATFLG ;BUMP COUNT
10240 101142 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
101142 104456 TRAP C$ERHRD
101144 001360 .WORD 752
101146 102765 .WORD T27RWN
101150 012136 .WORD PKTSSR
10241 101152 230$: CKLOOP ;LOOP IF SELECTED
101152 104406 TRAP C$CLP1
10242 ;*****
10243 ;
10244 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
10245 ;
10246 ;*****
10247 ;
10248 ;
10249 101154 013701 101470 MOV T27BFR+6,R1 ;PICK UP XST0
10250 101160 010102 MOV R1,R2 ;SET UP EXPECTED
10251 101162 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
10252 101166 020102 CMP R1,R2 ;DOES EXP = REC'D
10253 101170 001406 BEQ 240$ ;BR, IF EQUAL (OK)
10254 101172 005237 002212 INC FATFLG ;BUMP COUNT
10258 101176 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
101176 104456 TRAP C$ERHRD
101200 001361 .WORD 753
101202 102461 .WORD T27BOT
101204 015564 .WORD EXPREC
10259 101206 240$: CKLOOP ;LOOP IF SELECTED
101206 104406 TRAP C$CLP1
10260 101210 012704 101560 MOV #T27PK3,R4 ;SET UP PACKET ADDRESS
10261 101214 012737 000010 101562 MOV #10,T27RB ;SET UP RECORDS TO SPACE OVER
10262 ;*****
10263 ;
10264 ;ACK,CVC=1,SPACE FORWARD COMMAND
10265 ;
10266 ;*****
10267 ;
10268 ;
10269 101222 012737 140010 101560 MOV #140010,T27PK3 ;ACK,CVC=1,SPACE FORWARD COMMAND
10270 101230 010465 000000 250$: MOV R4,T5DB(R5) ;ISSUE COMMAND
10271 101234 005237 101610 252$: INC T27CNU ;BUMP TIMER
10272 101240 DELAY 1 ;DELAY ABOUT 100US
101240 012727 000001 MOV #1,(PC)+
101244 000000 .WORD 0
101246 013727 002116 MOV L$DLY,(PC)+
101252 000000 .WORD 0
101254 005367 177772 DEC -6(PC)
101260 001375 BNE .-4
101262 005367 177756 DEC -22(PC)
101266 001367 BNE .-20

```



## TEST 7: WRITE DATA RETRY

```

10315
10316
10317
10321 101440
10322 101440 100004
10323 101442 101450
10324 101444 000000
10325 101446 000012
10326 101450
10327 101450 101462
10328 101452 000000
10329 101454 000024
10330 101456 000000
10331 101460 000000
10332 101462
10333
10334
10335
10337 101550
10339 101550
10340 101550 100006
10341 101552 101570
10342 101554 000000
10343 101556 000006
10344
10348 101560
10349 101560 100005
10350 101562
10351 101562 003114
10352 101564 000000
10353 101566 000000
10354
10355
10356
10357
10358 101570
10359 101570 010
10360 101571 200
10361 101572 000000
10362 101574 000000
10363
10364
10365
10366
10367
10368 101576 100205
10369 101600 100605
10370 101602 102205
10371 101604 177777
10372
10373
10374 101606 000000
10375 101610 000000
10376 101612 000000
10377

;
;LOCAL STORAGE FOR THIS TEST
;
;
T27PACKET:
;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH , ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
;SELECT DRIVE 0
;MESSAGE BUFFER

T27DATA:
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
;=<.+10>E177770
T27PK2:
;WRITE SUB SYS MEM COMMAND, AND ACK
;ADDRESS OF SELECT BLOCK DATA
;SIZE OF DATA PACKET

T27PK3:
;REREAD COMMAND, AND ACK
;ADDRESS OF WRITE BUFFER
;SIZE OF BUFFER (EXTENT)

T27RB:
T27WB:
T27SZ:
;
;
T27BF2:
T27BS0:
T27BS1:
T27S2:
T27S3:
;
;
;TAPES MOTION PACKET COMMAND VALUES
;REREAD DATA (NEXT)
;REREAD DATA RETRY
;WRITE CONTINUOUS
;END OF DATA

T27CNT:
T27CNU:
T27DLY:
;TAPES TIMER COUNTER STORAGE AREA
;TAPES TIMER COUNTER STORAGE AREA
;DELAY COUNTER

```



## TEST 7: WRITE DATA RETRY

```

10379
10380
10381          ;+
10382          ;LOCAL TEXT MESSAGES FOR TEST
10383          ;-
10384
10385 101614      124      141      160  T27WNG: .ASCIZ  'Tape Position Incorrect After REREAD Previous (OPP-1)'
10386 101702      124      123      123  T27RDF: .ASCIZ  'TSSR Incorrect After READ DATA Command'
10387 101751      122      105      122  T27RRF: .ASCIZ  'REREAD Previous (Space Reverse, Read Forward) Command Failed'
10388 102046      120      117      123  T27SC:  .ASCIZ  'POSITION (Space Command) Failed, TSSR Not Correct'
10389 102130      122      111      102  T27LOR: .ASCIZ  'RIB NOT SET AFTER READ REVERSE INTO BOT'
10390 102100      124      123      123  T27WDF: .ASCIZ  'TSSR Not Correct After Illegal Mode Bits Set'
10391 102255      111      154      154  T27LOQ: .ASCIZ  'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
10392 102336      122      105      122  T27SSR: .ASCIZ  'REREAD COMMAND Not Accepted'
10393 102372      124      123      123  T27WDE: .ASCIZ  'TSSR Not Correct After WRITE DATA RETRY Command, At BOT'
10394 102461      124      141      160  T27BOT: .ASCIZ  'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
10395 102554      127      122      111  T27TIM: .ASCIZ  'WRITE DATA RETRY'S Erase Tape Not Long Enough'
10396 102631      122      105      122  T27EOT: .ASCIZ  'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
10397 102710      124      123      123  T27TM:  .ASCIZ  'TSSR Not Correct After REREAD COMMAND Reject'
10398 102765      122      145      167  T27RWN: .ASCIZ  'Rewind (POSITION) Command Not Accepted'
10399 103034      122      101      115  T27RNC: .ASCIZ  'RAM Error, Correct Data Pattern Not In Ram'
10400 103107      124      123      123  T27AM3: .ASCIZ  'TSSR Init, Failed After REREAD COMMAND'
10401 103156      104      162      151  T27OFL: .ASCIZ  'Drive 7 Select Failed To Set "OFL" In TSSR'
10402 103231      124      123      123  T27WDD: .ASCIZ  'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
10403 103321      124      123      123  T27WDC: .ASCIZ  'TSSR Not Correct After REREAD DATA Command'
10404 103374      103      126      103  T27VCK: .ASCIZ  'CVC Set, Didn't Reset VCK In Message Buffer'
10405 103447      124      123      102  T27BA:  .ASCIZ  'TSBA Not Correct After REREAD DATA Command'
10406 103522      127      122      111  T27WSS: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
10407 103611      122      145      141  T27LON: .ASCIZ  'Reading Long Record Failed To Set RLL Bit In XST0'
10408 103673      122      145      141  T27LOP: .ASCIZ  'Reading Long Record Failed To Set RLS Bit In XST0'
10409 103755      122      145      163  T27PBP: .ASCIZ  'Residual Byte Count Incorrect After Short Record Read'
10410 104043      122      145      141  T27TRL: .ASCIZ  'Reading Long Record Failed To Give Tape Status Alert'
10411 104131      127      122      111  T27NEF: .ASCIZ  'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
10412 104227      124      123      123  T27SCF: .ASCIZ  'TSSR Not Correct After SPACE RECORDS Command'
10413 104304      124      123      123  T27TSA: .ASCIZ  'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
10414 104366      124      123      123  T27WRF: .ASCIZ  'TSSR Not Correct After WRITE DATA RETRY Command'
10415 104446      104      141      164  T27DTA: .ASCIZ  'Data Compare Error, Data Read From Tape Not Equal To Written'
10416 104543      127      162      151  TS127ID: .ASCIZ  'Write Data Retry'
10417
10418          .EVEN
10419
10420          ;+
10421          ;
10422          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
10423          ;WRITE SUBSYSTEM MEMORY COMMAND
10424          ;
10425          ;-
10426
10425 104564
10426 104564
10427 104570      012701      101440
10428 104574      012721      100004
10429 104600      012721      101450
10430 104604      005021
10431 104606      012721      000012
10432 104612      012721      101462
10433 104616      005021
10434 104620      012721      000024
10435 104624      005021

          T27REST:
          SAVREG
          MOV      #T27PACKET,R1          ;SAVE THE REGISTERS
          MOV      #100004,(R1)+        ;START OF THE PACKET
          MOV      #T27DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,
          CLR      (R1)+                ;ADDRESS OF CHARAISTICS DATA BLOCK
          MOV      #10.,(R1)+          ;EXTENDED ADDRESS
          MOV      #T27BFR,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
          CLR      (R1)+                ;ADDRESS OF MESSAGE BUFFER
          MOV      #20.,(R1)+          ;LENGTH OF MESSAGE BUFFER
          CLR      (R1)+

```

TEST 7: WRITE DATA RETRY

```

10436 104626 012711 000000      MOV     #0,(R1)           ;SELECT DRIVE ZERO
10437 104632 012702 000030      MOV     #24,,R2          ;NUMBER OF LOCATIONS TO BE CLEARED
10438 104636 012762 177777 101462 64$;  MOV     #177777,T27BFR(R2) ;ALL ONES TO MESSAGE BUFFER
10439 104644 005742              TST     (R2)             ;NEXT LOCATION
10440 104646 022702 000000      CMP     #0,R2           ;AT END OF LOOP YET
10441 104652 001371              BNE     64$              ;KEEP GOING UNTIL DONE
10442 104654 000207              RTS     PC               ;RETURN
10443
10444
10445 104656              T27RT2:
10446 104656              SAVREG
10447 104662 012701 101550      MOV     #T27PK2,R1      ;SAVE THE REGISTERS
10448 104666 012721 100006      MOV     #100006,(R1)+   ;START OF THE PACKET
10449 104672 012721 101570      MOV     #T27BFR2,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
10450 104676 005021              CLR     (R1)+           ;ADDRESS OF DATA BLOCK
10451 104700 012721 000006      MOV     #6,,(R1)+      ;EXTENDED ADDRESS
10452 104704 005021              CLR     (R1)+           ;SIZE OF DATA BLOCK IN BYTES
10453 104706 012701 101570      MOV     #T27BFR2,R1    ;POINT TO DATA SEL AREA
10454 104712 005021              CLR     (R1)+
10455 104714 005011              CLR     (R1)+
10456 104716 000207              RTS     PC               ;RETURN
10457 104720
10458 104720              T27RT3:
10459 104724 012701 101560      SAVREG
10460 104730 005021              MOV     #T27PK3,R1     ;SAVE REGISTERS
10461 104732 005021              CLR     (R1)+          ;SET UP POINTER ADDRESS
10462 104734 005021              CLR     (R1)+          ;COMMAND SPACE
10463 104736 005011              CLR     (R1)+          ;ADDRESS OF DATA BLOCK
10464 104740 000207              CLR     (R1)+          ;EXTENDED ADDRESS
10465 104742              RTS     PC               ;SIZE OF DATA TRANSFER BLOCK
10465 104742              ENDTST                  ;RETURN
104742 104401              L10122: TRAP C$ETST

```

DFO

TEST 8: WRITE/READ TAPE MARK

10524	105060	103407			BCS	24:		;BR, IF COMMAND ISSUED OK
10525	105062	005237	002212		INC	FATFLG		;BUMP COUNT
10529	105066	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR
10530	105070				ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTIC FAILED
	105070	104456						TRAP C\$ERHRD
	105072	001442						.WORD 802
	105074	005054						.WORD WRTMSG
	105076	012124						.WORD SFIMSG
10531	105100					24:	CKLOOP	
	105100	104406						TRAP C\$CLP1
10532	105102	005737	002216		TST	EXTFEA		;CHECK FOR EXTENDED FEATURES SW SWITCH
10533	105106	001044			BNE	50:		;BR IF SWITCH IS ON
10534								
10535	105110	112737	000200	110261	MOVB	#200,T28BS1		;WRITE MISCELLANEOUS CONT/READ STATUS
10536	105116	112737	000010	110260	MOVB	#10,T28BS0		;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
10537	105124	012704	110240		MOV	#T28PK2,R4		;WRITE SUBSYS MEM PACKET
10538	105130	010465	000000		MOV	R4,TSD8(R5)		;ISSUE COMMAND
10539	105134	004737	016426		JSR	PC,CHKTSSR		;WAIT FOR SSR
10540	105140	103407			BCS	30:		;BR, IF NO ERROR
10541	105142	010001			MOV	R0,R1		;ERROR, SAVE TSSR
10542	105144	005237	002212		INC	FATFLG		;BUMP COUNT
10546	105150				ERRHRD	ERRNO,T28SSR,PKTSSR		;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
	105150	104456						TRAP C\$ERHRD
	105152	001443						.WORD 803
	105154	110755						.WORD T28SSR
	105156	012136						.WORD PKTSSR
10547	105160					30:	CKLOOP	;LOOP IF SELECTED
	105160	104406						TRAP C\$CLP1
10548	105162	012704	110130		MOV	#T28PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS
10549	105166	012737	000007	110150	MOV	#7,T28DSW		;SELECT DRIVE 7
10550	105174	004737	010752		JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS
10551	105200	103407			BCS	50:		;BR, IF COMMAND ISSUED OK
10552	105202	005237	002212		INC	FATFLG		;BUMP COUNT
10556	105206	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR
10557	105210				ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTIC FAILED
	105210	104456						TRAP C\$ERHRD
	105212	001444						.WORD 804
	105214	005054						.WORD WRTMSG
	105216	012124						.WORD SFIMSG
10558	105220					50:	CKLOOP	;SCOPE LOOP
	105220	104406						TRAP C\$CLP1
10559	105222	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
10560	105226	032701	000100		BIT	#0FL,R1		;CHECK FOR THE OFFLINE BIT SET
10561	105232	001006			BNE	60:		;BR, IF OFFLINE (GOOD)
10562	105234	005237	002212		INC	FATFLG		;BUMP COUNT
10566	105240				ERRDF	ERRNO,T28OFL,SFIMSG		;OFF LINE SHOULD HAVE BEEN SET (BAD)
	105240	104455						TRAP C\$ERDF
	105242	001445						.WORD 805
	105244	111310						.WORD T28OFL
	105246	012124						.WORD SFIMSG
10567	105250					60:	CKLOOP	;LOOP IF SELECTED
	105250	104406						TRAP C\$CLP1
10568	105252	012703	110276		MOV	#T28RN,R3		;POINTER FOR COMMANDS
10569	105256	011337	110250		MOV	(R3),T28PK3		;TAPE READ COMMAND IN PLACE
10570	105262	012704	110250		MOV	#T28PK3,R4		;R4 = POINTER TO PACKET
10571	105266	010465	000000		MOV	R4,TSD8(R5)		;ISSUE COMMAND
10572	105272	004737	016340		JSR	PC,WAITF		;WAIT FOR SSR TO SET

TEST 8: WRITE/READ TAPE MARK

```

10573 105276 016501 000002      MOV      TSSR(R5),R1                    ;GET TSSR CONTENTS
10574 105302 012702 100306      MOV      #55R!SC!OFL!BIT1!BIT2,R2       ;SET UP EXPECTED
10575 105306 020102      CMP      R1,R2                            ;ARE THEY EQUAL
10576 105310 001406      BEQ      80$                            ;BR, IF OK ESP. FUNCTION REJECT
10577 105312 005237 002212      INC      FATFLG                         ;BUMP COUNT
10581 105316                    ERRHRD   ERRNO,T28TM,PKTSSR            ;TSSR INCORRECT AFTER FORMAT CMD
         105316 104456                                                       TRAP      C$ERHRD
         105320 001446                                                       .WORD     806
         105322 111164                                                       .WORD     T28TM
         105324 012136                                                       .WORD     PKTSSR
10582 105326                    80$:      CKLOOP                            ;LOOP IF SELECTED
         105326 104406                                                       TRAP      C$CLP1
10583 105330 005723                    TST      (R3)+                            ;POINT TO NEXT COMMAND
10584 105332 022713 177777      CMP      #177777,(R3)                    ;END OF THE COMMANDS YET
10585 105336 001401                    BEQ      90$                            ;BR, IF DONE
10586 105340 000746                    BR      65$                            ;MORE COMMAND(S) TO GO
10587 105342                    90$:      ENDSUB                            ;>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
10588 105342                                                                       L10131:
         105342 104403                                                       TRAP      C$ESUB
10589 105344 023727 002212 000017      CMP      FATFLG,#15.                    ;IS ERROR COUNT AT 25
10590 105352 103402                    BLO      999$                            ;BR, IF LESS THAN 25
10591 105354 004737 017272                    JSR      PC,CKDROP                       ;TRY TO DROP THE UNIT
10592 105360                    999$:     

```



TEST 8: WRITE/READ TAPE MARK

```

10645 105552          75$:  CKLOOP          ;LOOP IF SELECTED
      105552 104406          ;GET MESSAGE BUFFER      TRAP  C$CLP1
10646 105554 013701 110160  MOV  T28BFR*6,R1
10647 105560 010102      MOV  R1,R2          ;SET UP EXPECTED
10648 105562 052702 001000  BIS  #BIT9,R2      ;SET THE ILC BIT IN EXPECTED
10649 105566 020102      CMP  R1,R2          ;ARE THEY EQUAL
10650 105570 001406      BEQ  180$          ;BR, IF EQUAL (ALL IS WELL)
10651 105572 005237 002212  INC  FATFLG        ;BUMP COUNT
10655 105576          ERRHRD ERRNO,T28LOQ,EXPREC ;THE ILC BIT WAS NOT SET IN XSTO
      105576 104456          TRAP  C$ERHRD
      105600 001452          .WORD 810
      105602 110674          .WORD T28LOQ
      105604 015564          .WORD EXPREC
10656 105606          180$:  CKLOOP          TRAP  C$CLP1
      105606 104406
10657 105610 005723      TST  (R3),        ;BUMP TO NEXT ADDRESS (COMMAND)
10658 105612 021327 177777  CMP  (R3),#177777 ;CHECK FOR END OF COMMAND BUFFER
10659 105616 001323      BNE  30$          ;BR, IF MORE COMMANDS TO TRY
10660 105620          190$:  CKLOOP          ;LOOP IF SELECTED
      105620 104406          TRAP  C$CLP1
10661 105622          ENDSUB          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
      105622          L10132:
      105622 104403          TRAP  C$ESUB
10662 105624 023727 002212 000017  CMP  FATFLG,#15. ;IS ERROR COUNT AT 25
10663 105632 103402      BLO  999$        ;BR, IF LESS THAN 25
10664 105634 004737 017272      JSR  PC,CKDROP   ;TRY TO DROP THE UNIT
10665 105640          999$:

```

## TEST 8: WRITE/READ TAPE MARK

```

10667      ;
10668      ;
10669      ; TEST 8, SUBTEST 3
10670      ;
10671      ; VERIFIES THAT WRITE TAPE MARK COMMANDS OPERATE
10672      ; PROPERLY, AND THAT READ COMMANDS SUBSEQUENTLY ISSUED
10673      ; TO DETECT THE WRITTEN TAPE MARKS TERMINATE WITH TAPE
10674      ; STATUS ALERT WITH THE TAPE MARK DETECTED (TMK) STATUS
10675      ; BIT SET. THE FOLLOWING SEQUENCE IS EXECUTED.
10676      ;
10677      ; 1. THE CONTROLLER IS INITIALIZED AND TAPE REWOUND.
10678      ; THIS SETS THE VOLUME CHECK (VCK) STATUS BIT.
10679      ;
10680      ; 2. A WRITE TAPE MARK COMMAND WITH CVC=1 IS ISSUED
10681      ; AND PROPER TERMINATION AND STATUS IS VERIFIED
10682      ; (I.E. VCK=0 AND TMK=1).
10683      ;
10684      ; 3. SEVERAL MORE WRITE TAPE MARK COMMANDS, THESE WITH
10685      ; CVC=0 ARE ISSUED AND PROPER TERMINATION (NORMAL)
10686      ; AND STATUS (TMK) VERIFIED.
10687      ;
10688      ; 4. A READ REVERSE COMMAND IS ISSUED AND PROPER
10689      ; TERMINATION (TAPE STATUS ALERT) AND STATUS (TMK)
10690      ; VERIFIED. IT IS ALSO VERIFIED THAT NO DATA IS
10691      ; TRANSFERRED INTO MEMORY.
10692      ;
10693      ; 5. A SPACE RECORDS REVERSE COMMAND IS ISSUED AND
10694      ; PROPER TERMINATION (TAPE STATUS ALERT) AND STATUS
10695      ; (TMK) VERIFIED.
10696      ;
10697      ; 6. THE TAPE IS REWOUND AND A READ FORWARD COMMAND IS
10698      ; ISSUED AND PROPER TERMINATION (TAPE STATUS ALERT)
10699      ; AND STATUS (TMK) VERIFIED. IT IS ALSO VERIFIED
10700      ; THAT NO DATA IS TRANSFERRED INTO MEMORY.
10701      ;
10702      ; 7. A SPACE RECORDS REVERSE COMMAND THAT CONTAINS A
10703      ; RECORD COUNT GREATER THAN 1 IS ISSUED AND IT IS
10704      ; VERIFIED THAT TAPE STATUS ALERT TERMINATION
10705      ; OCCURED, TMK=1 AND THAT RBPCR (RESIDUAL
10706      ; BYTE/RECORD COUNTER) CONTAINS THE PROPER NONZERO
10707      ; VALUE. THIS OPERATION VERIFIES THAT DETECTION OF
10708      ; THE TAPE MARK CAUSES THE SPACE RECORDS OPERATION
10709      ; TO BE PREMATURELY TERMINATED. THIS SHOULD LEAVE
10710      ; THE POSITION JUST BEFORE THE FIRST RECORD ON
10711      ; TAPE.
10712      ;
10713      ; 8. TAPE POSITION IS VERIFIED BY ISSUING ANOTHER
10714      ; SPACE RECORDS REVERSE COMMAND AND VERIFYING THAT
10715      ; TAPE STATUS ALERT TERMINATION OCCURS, WITH THE
10716      ; REVERSE INTO BOT (RIB) STATUS ERROR BIT SET.
10717      ;
10718      ; 9. A SPACE RECORDS FORWARD COMMAND THAT CONTAINS A
10719      ; RECORD COUNT GREATER THAN 1 IS ISSUED AND IT IS
10720      ; VERIFIED THAT TAPE STATUS ALERT TERMINATION
10721      ; OCCURED, TMK=1, AND THAT RBPCR (RESIDUAL
10722      ; BYTE/RECORD COUNTER) CONTAINS THE PROPER NONZERO
10723      ; VALUE. THIS OPERATION VERIFIES THAT DETECTION OF

```





## TEST 8: WRITE/READ TAPE MARK

```

106032 111241 .WORD T28RWN
106034 012136 .WORD PKTSSR
10769 106036 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
106036 104406
10770 106040 013701 110160 MOV T28BFR+6,R1 ;PICK UP XSTO
10771 106044 010102 MOV R1,R2 ;SET UP EXPECTED
10772 106046 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
10773 106052 020102 CMP R1,R2 ;DOES EXP = REC'D
10774 106054 001406 BEQ 40$ ;BR, IF EQUAL (OK)
10775 106056 005237 002212 INC FATFLG ;BUMP COUNT
10779 106062 ERRHRD ERRNO,T28BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
106062 104456 TRAP C$ERHRD
106064 001456 .WORD 814
106066 111117 .WORD T28BOT
106070 015564 .WORD EXPREC
10780 106072 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
106072 104406
10781 106074 005737 002216 42$: TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
10782 106100 001024 BNE 50$ ;BR IF SWITCH IS ON
10783 106102 112737 000200 110261 MOVB #200,T28BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
10784 106110 112737 000010 110260 MOVB #10,T28BS0 ;FUNC. SEL. BIT (TURN ON EXTFEA SWITCH)
10785 106116 012704 110240 MOV #T28PK2,R4 ;WRITE SUBSYS MEM PACKET
10786 106122 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
10787 106126 004737 016426 JSR PC,CHKTSSR ;WAIT FOR SSR
10788 106132 101407 BCS 50$ ;BR, IF NO FRUR
10789 106134 010001 MOV R0,R1 ;ERROR, SAVE TSSR
10790 106136 005237 002212 INC FATFLG ;BUMP COUNT
10794 106142 ERRHRD ERRNO,T28SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
106142 104456 TRAP C$ERHRD
106144 001457 .WORD 815
106146 110755 .WORD T28SSR
106150 012136 .WORD PKTSSR
10795 106152 50$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
106152 104406
10796 106154 012737 000007 110150 MOV #7,T28DSW ;SET UP DRIVE NUMBER
10797 106162 012704 110130 MOV #T28PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
10798 106166 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
10799 106172 103407 BCS 60$ ;BR, IF COMMAND ISSUED OK
10800 106174 005237 002212 INC FATFLG ;BUMP COUNT
10804 106200 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
10805 106202 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
106202 104456 TRAP C$ERHRD
106204 001460 .WORD 816
106206 005054 .WORD WRTMSG
106210 012124 .WORD SFIMSG
10806 106212 60$: CKLOOP ;SCOPE LOOP TRAP C$CLP1
106212 104406
10807 106214 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10808 106220 032701 000100 BIT #OFL,R1 ;CHECK FOR THE OFFLINE BIT SET
10809 106224 001006 BNE 65$ ;BR, IF OFFLINE (GOOD)
10810 106226 005237 002212 INC FATFLG ;BUMP COUNT
10814 106232 ERRDF ERRNO,T28OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
106232 104455 TRAP C$ERDF
106234 001461 .WORD 817
106236 111310 .WORD T28OFL
106240 012124 .WORD SFIMSG
10815 106242 65$: CKLOOP ;LOOP IF SELECTED

```

TEST 8: WRITE/READ TAPE MARK

```

10816 106242 104406          TRAP C$CLP1
10816 106244 013737 002172 110150      MOV    UNITN,T28DSW      ;SET UP DRIVE NUMBER
10817 106252 012704 110130      MOV    #T28PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
10818 106256 004737 010752      JSR    PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
10819 106262 103407          BCS    68$              ;BR, IF COMMAND ISSUED OK
10820 106264 005237 002212      INC    FATFLG           ;BUMP COUNT
10824 106270 010001          MOV    R0,R1           ;SAVE CONTENTS OF TSSR
10825 106272          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
106272 104456          TRAP C$ERHRD
106274 001462          .WORD 818
106276 005054          .WORD WRTMSG
106300 012124          .WORD SFIMSG
10826 106302          68$: CKLOOP           ;LOOP IF SELECTED
106302 104406          TRAP C$CLP1
10827 106304 012737 140011 110250      MOV    #140011,T28PK3   ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
10828 106312 012704 110250      MOV    #T28PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
10829 106316 010465 000000      MOV    R4,TSDB(R5)     ;ISSUE COMMAND
10830 106322 004737 016340      JSR    PC,WAITF        ;WAIT FOR SSR TO SET
10831 106326 016501 000002      MOV    TSSR(R5),R1     ;GET TSSR CONTENTS
10832 106332 012702 000200      MOV    #SSR,R2        ;SET UP EXPECTED
10833 106336 020102          CMP    R1,R2          ;ARE THEY EQUAL
10834 106340 001406          BEQ    70$            ;BR, IF OK
10835 106342 005237 002212      INC    FATFLG           ;BUMP COUNT
10839 106346          ERRHRD  ERRNO,T28WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE TAPE MARK
106346 104456          TRAP C$ERHRD
106350 001463          .WORD 819
106352 111363          .WORD T28WDC
106354 012136          .WORD PKTSSR
10840 106356          70$: CKLOOP           ;LOOP IF SELECTED
106356 104406          TRAP C$CLP1
10841 106360 013701 110160      MOV    T28BFR+6,R1     ;PICK UP XSTO (VCK CHECK)
10842 106364 010102          MOV    R1,R2          ;SET UP EXPECTED
10843 106366 042702 000020      BIC    #BIT4,R2        ;VCK SHOULD BE 0
10844 106372 020102          CMP    R1,R2          ;IS VCK SET CORRECTLY
10845 106374 001406          BEQ    80$            ;BR, IF VCK IS CLEAR
10846 106376 005237 002212      INC    FATFLG           ;BUMP COUNT
10850 106402          ERRHRD  ERRNO,T28VCK,EXPREC ;VCK WAS NOT CLEAR AFTER CVC=1
106402 104456          TRAP C$ERHRD
106404 001464          .WORD 820
106406 111442          .WORD T28VCK
106410 015564          .WORD EXPREC
10851 106412          80$: CKLOOP           ;LOOP IF SELECTED
106412 104406          TRAP C$CLP1
10852 106414 013701 110160      MOV    T28BFR+6,R1     ;PICK UP XSTO (CHECK TMK)
10853 106420 010102          MOV    R1,R2          ;SET UP EXPECTED
10854 106422 052702 100000      BIS    #BIT15,R2       ;TMK SHOULD BE SET
10855 106426 020102          CMP    R1,R2          ;WAS TMK SET
10856 106430 001406          BEQ    90$            ;BR, IF TMK WAS SET
10857 106432 005237 002212      INC    FATFLG           ;BUMP COUNT
10861 106436          ERRHRD  ERRNO,T28TMK,EXPREC ;TMK WAS NOT SET AFTER WRT TAPE MARK
106436 104456          TRAP C$ERHRD
106440 001465          .WORD 821
106442 111515          .WORD T28TMK
106444 015564          .WORD EXPREC
10862 106446          90$: CKLOOP           ;LOOP IF SELECTED
106446 104406          TRAP C$CLP1
10863 106450 004737 011104      JSR    PC,REWIND       ;CALL TAPE REWIND COMMAND

```

TEST 8: WRITE/READ TAPE MARK

```

10864 106454 103411          BCS      130$           ;BR, IF NO PROBLEM
10865 106456 010004          MOV      RO,R4         ;SAVE PACKET ADDRESS
10866 106460 016501 000002   MOV      TSSR(R5),R1   ;GET TSSR STATUS
10867 106464 005237 002212   INC      FATFLG        ;BUMP COUNT
10871 106470          ERRHRD   ERRNO,T28RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     822
                                .WORD     T28RWN
                                .WORD     PKTSSR
                                TRAP      C$CLP1
10872 106500          130$:   CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
10873 106500 104406          MOV      T28BFR+6,R1   ;PICK UP XSTO
10874 106502 013701 110160   MOV      R1,R2         ;SET UP EXPECTED
10875 106506 010102          MOV      R1,R2         ;SET BOT BIT IN EXPECTED
10876 106510 052702 000002   BIS      %BIT1,R2      ;DOES EXP = REC'D
10877 106514 020102          CMP      R1,R2         ;BR, IF EQUAL (OK)
10878 106516 001406          BEQ      140$         ;BUMP COUNT
10878 106520 005237 002212   INC      FATFLG        ;TAPE NOT AT BOT AFTER REWIND
10882 106524          ERRHRD   ERRNO,T28BOT,EXPREC
                                TRAP      C$ERHRD
                                .WORD     823
                                .WORD     T28BOT
                                .WORD     EXPREC
                                TRAP      C$CLP1
10883 106534          140$:   CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
10884 106534 104406          MOV      #10.,R3       ;NUMBER OF RECORDS TO WRITE TM
10884 106536 012703 000012   MOV      #140011,T28PK3 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
10885 106542 012737 140011 110250  MOV      #T28PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
10886 106550 012704 110250          MOV      R4,TSDB(R5)   ;ISSUE COMMAND
10887 106554 010465 000000 155$:   MOV      R4,TSDB(R5)   ;WAIT FOR SSR TO SET
10888 106560 004737 016340          JSR      PC,WAITF      ;PICK UP TSSR
10889 106564 016501 000002          MOV      TSSR(R5),R1   ;SET UP EXPECTED (SSR ONLY)
10890 106570 012702 000200          MOV      #SSR,R2       ;WAS STATUS GOOD
10891 106574 020102          CMP      R1,R2         ;BR, IF TERMINATION WAS GOOD
10892 106576 001406          BEQ      165$         ;BUMP COUNT
10893 106600 005237 002212   INC      FATFLG        ;TSSR NOT CORRECT AFTER WRT TAPE M.
10897 106604          ERRHRD   ERRNO,T28WDC,PKTSSR
                                TRAP      C$ERHRD
                                .WORD     824
                                .WORD     T28WDC
                                .WORD     PKTSSR
                                TRAP      C$CLP1
10898 106614          165$:   CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
10899 106614 104406          MOV      T28BFR+6,R1   ;PICK UP XSTO
10899 106616 013701 110160   MOV      R1,R2         ;SET UP EXPECTED
10900 106622 010102          MOV      R1,R2         ;SET TMK BIT IN EXPECTED
10901 106624 052702 100000   BIS      %BIT15,R2     ;DOES EXP = REC'D
10902 106630 020102          CMP      R1,R2         ;BR, IF EQUAL (OK)
10903 106632 001406          BEQ      180$         ;BUMP COUNT
10904 106634 005237 002212   INC      FATFLG        ;TMK NOT SET AFTER WRT TAPE MARK
10908 106640          ERRHRD   ERRNO,T28TMK,EXPREC
                                TRAP      C$ERHRD
                                .WORD     825
                                .WORD     T28TMK
                                .WORD     EXPREC
                                TRAP      C$CLP1
10909 106650          180$:   CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
10910 106650 104406          DEC      R3            ;BUMP COUNTER DOWN
10911 106652 005303          BNE      155$         ;BR, IF LESS THAN 10 TAPE MARKS
10912 106654 001337          MOV      #177777,R0   ;VALUE 10 WRITTEN TO MEMORY
10912 106656 012700 177777

```

TEST 8: WRITE/READ TAPE MARK

10913	106662	004737	017512		JSR	PC,FILLMEM		;FILL MEM WITH ALL ONES
10914	106666	013737	003114	110252	MOV	FREE,T28WB		;STARTING READ BUFFER ADDRESS
10915	106674	012737	140401	110250	MOV	#140401,T28PK3		;READ REVERSE,ACK, COMMAND
10916	106702	012704	110250		MOV	#T28PK3,R4		;SET UP R4 WITH PACKET ADDRESS
10917	106706	013737	000024	110256	MOV	20.,T28SZ		;SET UP RECORD SIZE IN PACKET
10918	106714	010465	000000		MOV	R4,TSD8(R5)		;ISSUE COMMAND
10919	106720	004737	016340		JSR	PC,WAITF		;WAIT FOR SSR TO SET
10920	106724	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
10921	106730	012702	100204		MOV	#SSR!SC!BIT2,R2		;SET UP EXPECTED
10922	106734	020102			CMP	R1,R2		;ARE THEY EQUAL
10923	106736	001406			BEQ	200\$		;BR, IF OK
10924	106740	005237	002212		INC	FATFLG		;BUMP COUNT
10928	106744				ERRHRD	ERRNO,T28RDF,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA
	106744	104456					TRAP	C\$ERHRD
	106746	001472					.WORD	826
	106750	110454					.WORD	T28RDF
	106752	012136					.WORD	PKTSSR
10929	106754			200\$:	CKLOOP			;LOOP IF SELECTED
	106754	104406					TRAP	C\$CLP1
10930	106756	013701	110160		MOV	T28BFR+6,R1		;PICK UP XSTO
10931	106762	010102			MOV	R1,R2		;SET UP EXPECTED
10932	106764	052702	100000		BIS	#BIT15,R2		;TMK SHOULD BE SET
10933	106770	020102			CMP	R1,R2		;IS TMK SET
10934	106772	001406			BEQ	210\$		;BR, IF TMK WAS SET (GOOD)
10935	106774	005237	002212		INC	FATFLG		;BUMP COUNT
10939	107000				ERRHRD	ERRNO,T28RRM,EXPREC		;TMK NOT SET AFTER READ REV
	107000	104456					TRAP	C\$ERHRD
	107002	001473					.WORD	827
	107004	111567					.WORD	T28RRM
	107006	015564					.WORD	EXPREC
10940	107010			210\$:	CKLOOP			;LOOP IF SELECTED
	107010	104406					TRAP	C\$CLP1
10941	107012	017701	074076		MOV	#FREE,R1		;FIRST LOC IN READ BUFFER
10942	107016	012702	177777		MOV	#177777,R2		;EXPECTED IF NO DATA TRANS.
10943	107022	020102			CMP	R1,R2		;DID ANY DATA GET TRANSFERRED
10944	107024	001406			BEQ	220\$		;BR, IF NO DATA TRANS (GOOD)
10945	107026	005237	002212		INC	FATFLG		;BUMP COUNT
10949	107032				ERRHRD	ERRNO,T28DTR,EXPREC		;DATA TRANSFERRED ON READ TAPE MARK
	107032	104456					TRAP	C\$ERHRD
	107034	001474					.WORD	828
	107036	112002					.WORD	T28DTR
	107040	015564					.WORD	EXPREC
10950	107042			220\$:	CKLOOP			;LOOP IF SELECTED
	107042	104406					TRAP	C\$CLP1
10951	107044	012737	100410	110250	MOV	#100410,T28PK3		;SPACE REVERSE,ACK, COMMAND
10952	107052	012737	000001	110252	MOV	#1,T28RB		;NUMBER OF RECORDS TO SPACE BACK
10953	107060	012704	110250		MOV	#T28PK3,R4		;SET UP R4 WITH PACKET ADDRESS
10954	107064	010465	000000		MOV	R4,TSD8(R5)		;ISSUE COMMAND
10955	107070	004737	016340		JSR	PC,WAITF		;WAIT FOR SSR TO SET
10956	107074	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
10957	107100	012702	100204		MOV	#SSR!SC!BIT2,R2		;SET UP EXPECTED
10958	107104	020102			CMP	R1,R2		;ARE THEY EQUAL
10959	107106	001406			BEQ	222\$		;BR, IF OK
10960	107110	005237	002212		INC	FATFLG		;BUMP COUNT
10964	107114				ERRHRD	ERRNO,T28RDG,PKTSSR		;TSSR INCORRECT AFTER SPACE CMD.
	107114	104456					TRAP	C\$ERHRD
	107116	001475					.WORD	829

TEST 8: WRITE/READ TAPE MARK

10965	107120	110535						.WORD	T28RDG
	107122	012136						.WORD	PKTSSR
	107124	104406		222\$:	CKLOOP		;LOOP IF SELECTED		
	107124	104406						TRAP	C\$CLP1
10966	107126	013701	110160		MOV	T28BFR+6,R1	;PICK UP XSTO		
10967	107132	010102			MOV	R1,R2	;SET UP EXPECTED		
10968	107134	052702	100000		BIS	#BIT15,R2	;TMK SHOULD BE SET		
10969	107140	020102			CMP	R1,R2	;IS TMK SET		
10970	107142	001406			BEQ	226\$	;BR, IF TMK WAS SET (GOOD)		
10971	107144	005237	002212		INC	FATFLG	;BUMP COUNT		
10975	107150				ERRHRD	ERRNO,T28RRN,EXPREC	;TMK NOT SET AFTER SPACE REV		
	107150	104456						TRAP	C\$ERHRD
	107152	001476						.WORD	830
	107154	111645						.WORD	T28RRN
	107156	015564						.WORD	EXPREC
10976	107160			226\$:	CKLOOP		;LOOP IF SELECTED		
	107160	104406						TRAP	C\$CLP1
10977	107162	004737	011104		JSR	PC,REWIND	;CALL TAPE REWIND COMMAND		
10978	107166	103411			BCS	230\$	;BR, IF NO PROBLEM		
10979	107170	010004			MOV	R0,R4	;SAVE PACKET ADDRESS		
10980	107172	016501	000002		MOV	TSSR(R5),R1	;GET TSSR		
10981	107176	005237	002212		INC	FATFLG	;BUMP COUNT		
10985	107202				ERRHRD	ERRNO,T28RWN,PKTSSR	;REWIND NOT ACCEPTED		
	107202	104456						TRAP	C\$ERHRD
	107204	001477						.WORD	831
	107206	111241						.WORD	T28RWN
	107210	012136						.WORD	PKTSSR
10986	107212			230\$:	CKLOOP		;LOOP IF SELECTED		
	107212	104406						TRAP	C\$CLP1
10987	107214	013701	110160		MOV	T28BFR+6,R1	;PICK UP XSTO		
10988	107220	010102			MOV	R1,R2	;SET UP EXPECTED		
10989	107222	052702	000002		BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED		
10990	107226	020102			CMP	R1,R2	;DOES EXP = REC'D		
10991	107230	001406			BEQ	240\$	;BR, IF EQUAL (OK)		
10992	107232	005237	002212		INC	FATFLG	;BUMP COUNT		
10996	107236				ERRHRD	ERRNO,T28BOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	107236	104456						TRAP	C\$ERHRD
	107240	001500						.WORD	832
	107242	111117						.WORD	T28BOT
	107244	015564						.WORD	EXPREC
10997	107246			240\$:	CKLOOP		;LOOP IF SELECTED		
	107246	104406						TRAP	C\$CLP1
10998	107250	012700	177777		MOV	#177777,R0	;VALUE TO WRITTEN TO MEMORY		
10999	107254	004737	017512		JSR	PC,FILLMEM	;FILL MEM WITH ALL ONES		
11000	107260	013737	003114	110252	MOV	FREE,T28RB	;STARTING READ BUFFER ADDRESS		
11001	107266	012737	100001	110250	MOV	#100001,T28PK3	;READ FORWARD,ACK, COMMAND		
11002	107274	012704	110250		MOV	#T28PK3,R4	;SET UP R4 WITH PACKET ADDRESS		
11003	107300	013737	000024	110256	MOV	20.,T28SZ	;SET UP RECORD SIZE IN PACKET		
11004	107306	010465	000000		MOV	R4,TSOB(R5)	;ISSUE COMMAND		
11005	107312	004737	016340		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
11006	107316	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
11007	107322	012702	100204		MOV	#SSR!SC!BIT2,R2	;SET UP EXPECTED		
11008	107326	020102			CMP	R1,R2	;ARE THEY EQUAL		
11009	107330	001406			BEQ	245\$	;BR, IF OK		
11010	107332	005237	002212		INC	FATFLG	;BUMP COUNT		
11014	107336				ERRHRD	ERRNO,T28WDE,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA		
	107336	104456						TRAP	C\$ERHRD

## TEST 8: WRITE/READ TAPE MARK

```

107550 015564
11062 107552          270$: CKLOOP          ;LOOP IF SELECTED          .WORD  EXPREC
107552 104406          TRAP  C$CLP1
11063 107554 013701 110156          MOV  T28BFR+4,R1          ;PICK UP RESIDUAL BYTE COUNTER
11064 107560 012702 000004          MOV  #4,R2          ;SHOULD BE THE DIFFERENCE
11065 107564 020102          CMP  R1,R2          ;IS COUNTER CORRECT
11066 107566 001406          BEQ  280$          ;BR, IF COUNTER CORRECT
11067 107570 005237 002212          INC  FATFLG          ;BUMP COUNT
11071 107574          ERRHRD  ERRNO,T28BFP,EXPREC          ;RESIDUAL BYTE COUNTER NOT CORRECT
107574 104456          TRAP  C$ERHRD
107576 001506          .WORD  838
107600 110371          .WORD  T28BFP
107602 015564          .WORD  EXPREC
11072 107604          280$: CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
107604 104406          ;SPACE REVERSE,ACK, COMMAND
11073 107606 012737 100410 110250          MOV  #100410,T28PK3          ;NUMBER OF RECORDS TO SPACE BACK
11074 107614 012737 000001 110252          MOV  #1,T28RB          ;SET UP R4 WITH PACKET ADDRESS
11075 107622 012704 110250          MOV  #T28PK3,R4          ;ISSUE COMMAND
11076 107626 010465 000000          MOV  R4,TSD8(R5)          ;WAIT FOR SSR TO SET
11077 107632 004737 016340          JSR  PC,WAITF          ;GET TSSR CONTENTS
11078 107636 016501 000002          MOV  TSSR(R5),R1          ;SET UP EXPECTED
11079 107642 012702 100204          MOV  #SSR!SC!BIT2,R2          ;ARE THEY EQUAL
11080 107646 020102          CMP  R1,R2          ;BR, IF OK
11081 107650 001406          BEQ  290$          ;BUMP COUNT
11082 107652 005237 002212          INC  FATFLG          ;TSSR INCORRECT AFTER SPACE CMD.
11086 107656          ERRHRD  ERRNO,T28RDG,PKTSSR          TRAP  C$ERHRD
107656 104456          .WORD  839
107660 001507          .WORD  T28RDG
107662 110535          .WORD  PKTSSR
107664 012136
11087 107666          290$: CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
107666 104406          ;PICK UP XST3
11088 107670 013701 110166          MOV  T28BFR+14,R1          ;SET UP EXPECTED
11089 107674 010102          MOV  R1,R2          ;RIB SHOULD BE SET
11090 107676 052702 000001          BIS  #BIT0,R2          ;IS RIB SET
11091 107702 020102          CMP  R1,R2          ;BR, IF RIB WAS SET (GOOD)
11092 107704 001406          BEQ  300$          ;BUMP COUNT
11093 107706 005237 002212          INC  FATFLG          ;RIB NOT SET AFTER READ REV
11097 107712          ERRHRD  ERRNO,T28RIB,EXPREC          TRAP  C$ERHRD
107712 104456          .WORD  840
107714 001510          .WORD  T28RIB
107716 110314          .WORD  EXPREC
107720 015564
11098 107722          300$: CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
107722 104406          ;SPACE FORWARD,ACK, COMMAND
11099 107724 012737 100010 110250          MOV  #100010,T28PK3          ;NUMBER OF RECORDS TO SPACE FORW.
11100 107732 012737 000005 110252          MOV  #5,T28RB          ;SET UP R4 WITH PACKET ADDRESS
11101 107740 012704 110250          MOV  #T28PK3,R4          ;ISSUE COMMAND
11102 107744 010465 000000          MOV  R4,TSD8(R5)          ;WAIT FOR SSR TO SET
11103 107750 004737 016340          JSR  PC,WAITF          ;GET TSSR CONTENTS
11104 107754 016501 000002          MOV  TSSR(R5),R1          ;SET UP EXPECTED
11105 107760 012702 100204          MOV  #SSR!SC!BIT2,R2          ;ARE THEY EQUAL
11106 107764 020102          CMP  R1,R2          ;BR, IF OK
11107 107766 001406          BEQ  310$          ;BUMP COUNT
11108 107770 005237 002212          INC  FATFLG          ;TSSR INCORRECT AFTER SPACE CMD.
11112 107774          ERRHRD  ERRNO,T28RDF,EXPREC          TRAP  C$ERHRD
107774 104456

```

SYMBOL TABLE

P.IE = 000200	SPM6 112610	TSREJ = 000006	T\$\$CLE = 010034	T22WRT 026400
P.INIT = 000013	SPM7 112640	TSSDEF 006676	T\$\$DU = 010032	T23A 003134 G
P.MODE = 007400	SR0 = 177572	TSSR = 000002 G	T\$\$HAR = 010134	T23AM3 033240
P.OPP = 020000	SR1 = 177574	TSSRBI 003500 G	T\$\$HW = 010000	T23B 003136 G
P.POSI = 000010	SR2 = 177576	TSSRFO 006505	T\$\$INI = 010030	T23BA 033625
P.READ = 000001	SR3 = 172516	TSSRH = 000003 G	T\$\$MSG = 010025	T23BFR 032512
P.SWR = 010000	SSR = 000200	TSSX 004016	T\$\$PRD = 010027	T23BF2 032622
P.WRIT = 000005	STATCO 012502	TSTBLK 002742 G	T\$\$RPT = 010035	T23BS0 032622
P.WRTC = 000004	SVCGBL = 000000	TSTCNT 002204 G	T\$\$SOF = 010135	T23BS1 032623
P.WRTS = 000006	SVCINS = 000000	TSTEND 017010	T\$\$SRV = 010026	T23CHK 034162
QVP 002174 G	SVCSUB = 000001	TSTFLA 002304 G	T\$\$SUB = 010133	T23CON 032640
RAMASC 014246	SVCTAG = 000000	TSTL00 016546 G	T\$\$SW = 010001	T23DAT 032500
RAMDAT 002232 G	SVCTST = 000001	TSTPTR 002306 G	T\$\$TES = 010130	T23DSW 032510
RAMERR 015600 G	S\$LSYM = 010000	TSTSET 016600 G	T1 023564 G	T23E0T 032764
RAMEXP 015620 G	SO.IDB = 000010	TST21I 024524	T2 024702 G	T23ET 032677
RAMFOR 010206	SO.IFB = 000002	TST22I 027167	T2.1 024732	T23L00 027426
RAMSIZ 002272 G	SO.IFP = 000001	TST23I 033766	T2.2 025324	T23OFL 033306
RAMTAD 015606 G	SO.ILD = 000020	TST24I 046232	T2.3 025646	T23PAC 032470
RCVHIA 002274 G	SO.ION = 000040	TST25I 055200	T21AM3 024403	T23PK2 032600
RCVLOA 002276 G	SO.IRD = 000100	TST26I 074527	T21BFR 024204	T23PK3 032610
RDERR 005204	SO.IRW = 000004	TST27I 104543	T21BF2 024300	T23RES 034002
RECMGSG 002456 G	SO.ISP = 000200	TST28I 112161	T21BS0 024300	T23RNC 033165
RECV 002224 G	S1.ICE = 002000	TSV2 002000 G	T21BS1 024301	T23RSZ 032620
REGSAV 020230	S1.IEO = 010000	TSV3 002166 G	T21DAT 024170	T23RT2 034074
RETERR 005370	S1.IFM = 001000	TSV4 021636 G	T21DLY 024202	T23RT3 034136
RETRY 034226	S1.IHE = 000400	TSV6 112366 G	T21DSW 024200	T23RWN 033116
REWIND 011104 G	S1.IID = 004000	TSV7 023564 G	T21L00 023614	T23SSR 032644
RMCHBE = 000167	S1.IIR = 020000	TTIBFR = 177562 G	T21OFL 024503	T23SZ 032616
RMCHEN = 000200	S1.I2R = 040000	TTICSR = 177560 G	T21PAC 024160	T23S2 032624
RMMSGB = 000215	S1.PAK = 100000	T\$IVEC = 000060 G	T21PK2 024270	T23S3 032626
RMMSGE = 000234	S2.ATI = 000010	T\$ARGC = 000003	T21RES 024546	T23TM 033042
RMPKTB = 000201	S2.BTI = 000004	T\$CODE = 001130	T21RT2 024636	T23TMP 032630
RMPKTE = 000210	S2.DIM = 000200	T\$ERRN = 001513	T21SSR 024306	T23VCK 033552
RMR = 010000	S2.ILW = 000100	T\$EXCP = 000000	T21S2 024302	T23WB 032612
RWPACK 011200	S2.INR = 000020	T\$FLAG = 000040	T21S3 024304	T23WD 032634
SC = 100000	S2.OUT = 000040	T\$GMAN = 000000	T22AM3 026505	T23WDC 033450
SCE = 020000	S2.UND = 000003	T\$HILI = 000776	T22BFR 026272	T23WDD 033361
SCHERR 005276	TBLEND = 003052 G	T\$LAST = 000001	T22BF2 026370	T23WDR 032636
SCME 005011	TCOASC 006566	T\$LOLI = 000000	T22BS0 026370	T23WRT 032632
SDELAY 010750	TCOCOD 006766	T\$LSYM = 010000	T22BS1 026371	T23WSS 033677
SELASC 020532	TEMP1 003106 G	T\$LINO = 000010	T22DAT 026260	T24AM3 045220
SELDAT = 000004	TEMP2 003110 G	T\$NEST = 177777	T22FOR 026404	T24BA 045552
SEL2 = 000002	TERCLS = 000016	T\$NS0 = 000000	T22L00 024732	T24BFR 043642
SETMAP 017406	TESTNO = 000010	T\$NS1 = 000005	T22OFL 026605	T24BF2 043750
SETU 022224	TEXASC 006325	T\$NS2 = 000002	T22PAC 026250	T24BOT 044613
SFFMSG 012172 G	TFCASC 006627	T\$PTNU = 000000	T22PK2 026360	T24BS0 043750
SFHERR 003703	TIMEXP 015642 G	T\$SAVL = 177777	T22POS 026402	T24BS1 043751
SFIERR 003650	TIMSGO 015670	T\$SEGL = 177777	T22RD 026376	T24CON 043762
SFIMSG 012124 G	TINERR 012111	T\$SUBN = 000003	T22RES 027222	T24DAT 043630
SFPTBL 002156 G	TMPBFR 002622 G	T\$TAGL = 177777	T22RT2 027314	T24DLY 043766
SIFLAG 003144 G	TNAM 016774	T\$TAGN = 010136	T22RWJ 026754	T24DSW 043640
SIMSG 012056	TRANST 002156 G	T\$TEMP = 000000	T22SSR 026410	T24DTA 044660
SKIPT 003370	TSBA = 000000 G	T\$TEST = 000010	T22S2 026372	T24E01 044746
SOFINI 016064 G	TSBAH = 000001 G	T\$TSTM = 177777	T22S3 026374	T24ILA 044342
SPACE 010556 G	TSDB = 000000 G	T\$TSTS = 000001	T22TM 026660	T24LON 045712
SPM1 112530	TSDBH = 000001 G	T\$\$AU = 010031	T22VCK 027027	T24L00 034412
SPM4 112560	TSFCOD 007326	T\$\$AUT = 010033	T22WLK 027102	T24LOP 045774

SYMBOL TABLE

T24LOQ	044426	T25SSR	054034	T26WDC	073640	T27WDR	101600	T4	034352	G
T24LOR	044042	T25SZ	054006	T26WDD	073550	T27WNG	101614	T4.1	034412	
T24NEF	043770	T25S2	054012	T26WDE	072743	T27WRF	104366	T4.10	041636	
T24NXM	044201	T25S3	054014	T26WDF	072551	T27WSS	103522	T4.11	042102	
T24OFL	045265	T25TM	054242	T26WNG	072006	T28BFR	110152	T4.12	042354	
T24PAC	043620	T25WB	054002	T26WSS	074041	T28BF2	110260	T4.13	042610	
T24PBP	046056	T25WDC	055127	T27AM3	103107	T28BOT	111117	T4.14	043110	
T24PK2	043730	T25WDE	054115	T27BA	103447	T28BS0	110260	T4.2	035034	
T24PK3	043740	T25WDR	054020	T27BFR	101462	T28BS1	110261	T4.3	035614	
T24RB	043742	T25WNG	054405	T27BF2	101770	T28CNT	110306	T4.4	036370	
T24RES	046300	T25WNH	054560	T27BOT	102451	T28CNU	110310	T4.5	037072	
T24RN	043756	T26AM3	073426	T27BS0	102451	T28CON	110302	T4.6	037536	
T24RNC	045145	T26BA	073765	T27BS1	102451	T28DAT	110140	T4.7	040172	
T24RT2	046372	T26BFR	071652	T27CNT	101606	T28DLY	110312	T4.8	040626	
T24RT3	046434	T26BF2	071760	T27CNU	101610	T28DSW	110150	T4.9	041220	
T24RWN	045076	T26BOT	073015	T27CON	101602	T28DTA	112064	T5	046466	G
T24SSR	041507	T26US0	071760	T27DAT	101450	T28DTR	112002	T5.1	046516	
T24SZ	043746	T26BS1	071761	T27DLY	101612	T28IMV	110266	T5.2	047472	
T24S2	043752	T26CNT	071776	T27DSW	101460	T28L00	105000	T5.3	050072	
T24S3	043754	T26CNU	072000	T27DTA	104446	T28LOQ	110674	T5.4	050546	
T24TM	045023	T26DAT	071640	T27EOT	102631	T28OFL	111310	T5.5	051212	
T24TRL	046144	T26DLY	072004	T27LON	103611	T28PAC	110130	T5.6	051752	
T24VCK	045477	T26DSW	071650	T27L00	074766	T28PBP	110371	T5.7	052712	
T24WB	043742	T26DTA	073062	T27L0P	103673	T28PK2	110240	T5.8	053232	
T24WDC	045426	T26EOT	073150	T27LOQ	102255	T28PK3	110250	T6	055404	G
T24WDD	045340	T26LON	074130	T27LOR	102130	T28RB	110252	T6.1	055444	
T24WDE	044541	T26L00	055444	T27NEF	104131	T28RDF	110454	T6.10	065064	
T24WDF	044265	T26LOP	074212	T27OFL	103156	T28RDG	110535	T6.11	065740	
T24WDG	044112	T26LOQ	072626	T27PAC	101440	T28RES	112206	T6.12	066610	
T24WDR	043760	T26LOR	072501	T27PBP	103755	T28RIB	110314	T6.13	067542	
T24WSS	045623	T26NEF	072074	T27PK2	101550	T28RN	110276	T6.14	070552	
T25BFR	053702	T26NEQ	074450	T27PK3	101560	T28RRM	111567	T6.15	071132	
T25BF2	054010	T26OFL	073475	T27RB	101562	T28RRN	111645	T6.2	056356	
T25BNC	054470	T26PAC	071630	T27RDF	101702	T28RRP	111724	T6.3	057224	
T25BOT	054175	T26PBP	074274	T27RES	104564	T28RT2	112300	T6.4	060116	
T25BS0	054010	T26PK2	071740	T27RN	101576	T28RT3	112342	T6.5	061044	
T25BS1	054011	T26PK3	071750	T27RNC	103034	T28RWN	111241	T6.6	061622	
T25CNT	054030	T26RB	071752	T27RRF	101751	T28SSR	110755	T6.7	062464	
T25CN2	054026	T26RDF	072156	T27RT2	104656	T28SZ	110256	T6.8	063336	
T25CON	054022	T26RES	074540	T27RT3	104720	T28S2	110262	T6.9	064210	
T25DAT	053670	T26RN	071766	T27RWN	102765	T28S3	110264	T7	074726	G
T25DLY	054032	T26RNC	073353	T27SC	102046	T28TM	111164	T7.1	074766	
T25DSW	053700	T26RRF	072225	T27SCF	104227	T28TMK	111515	T7.2	075364	
T25L00	046516	T26RRG	072322	T27SSR	102336	T28VCK	111442	T7.3	076146	
T25NEF	054643	T26RSZ	072002	T27SZ	101566	T28WB	110252	T7.4	076770	
T25NET	054331	T26RT2	074632	T27S2	101572	T28WDC	111363	T7.5	077672	
T25OFL	055054	T26RT3	074674	T27S3	101574	T28WDE	111026	T8	104744	G
T25PAC	053660	T26RWN	073304	T27TIM	102554	T28WDF	110617	T8.1	105000	
T25PK2	053770	T26SC	072417	T27TM	102710	T28WDR	110300	T8.2	105360	
T25PK3	054000	T26SSR	072707	T27TRL	104043	T3	027362	T8.3	105640	
T25RB	054002	T26SZ	071756	T27TSA	104300	T3BFLG	003140	UAM	= 000200	G
T25RES	055216	T26S2	071762	T27VCK	103377	T3.1	027426	UNITN	= 002172	G
T25RIB	054723	T26S3	071764	T27WB	101562	T3.2	030000	UNREC	= 000006	
T25RN	054016	T26TM	073227	T27WDC	103321	T3.3	030650	USI	004121	
T25RT2	055310	T26TRL	074362	T27WDD	103231	T3.4	031470	WAITF	016340	G
T25RT3	055352	T26VCK	073713	T27WDE	102372	T3.5	031704	WC.IFA	= 000200	
T25RWN	055005	T26WB	071752	T27WDF	102200	T3.6	032136	WC.IFE	= 000002	



SYMBOL TABLE

WC.IG0= 000001	WRTErr 005111	XSOILA= 000400	X\$AL5= 000040	X2.UNI= 000007
WC.IRI= 000010	WRMSG 005054	XSOILC= 001000	X\$OFFG= 000400	X2.WCF= 002000
WC.IRW= 000004	WSMOK 021360 G	XSOLEI= 020000	X\$TRUK= 000020	X3.DCK= 000010
WC.IOT= 000100	XFERAS 016030	XSOHOT= 000200	X1.COR= 020000	X3.MBZ= 000006
WC.IIT= 000040	XNXM 016466	XSONEF= 002000	X1.DLT= 100000	X3.MDF= 177400
WC.I5R= 000020	XORBF0 007764	XSOONL= 000100	X1.MBZ= 017575	X3.OPI= 000100
WF.IED= 000010	XORFUR 010102	XSOPEF= 000010	X1.RHP= 000400	X3.REV= 000040
WF.IER= 000004	XST0 = 000006 G	XSORIL= 010000	X1.SPA= 040000	X3.RIH= 000001
WF.IHI= 000200	XST1 = 000010 G	XSORLS= 040000	X1.UNC= 000001	X3.SPA= 000200
WF.IRE= 000040	XST2 = 000012 G	XSGTMK= 100000	X2.BUF= 000100	X3.TRF= 000010
WF.IWF= 000020	XST3 = 000014 G	XSOVCK= 000020	X2.EXT= 000200	X4.HSP= 100000
WF.IWR= 000100	XST4 = 000016 G	XSOWLF= 004000	X2.OPM= 100000	X4.MBZ= 017400
WF.I3R= 000002	XSOBOT= 000002	XSOWLK= 000004	X2.RCE= 040000	X4.RCE= 040000
WF.I4R= 000001	XSOBOT= 000001	XXCOMM 005112 G	X2.REV= 000077	X4.TSM= 020000
WRTCHR 010752 G	XSOIE = 000040	X\$ALWA= 000000	X2.SPA= 035400	X4.WRC= 000377

. ABS. 113004 000  
 000000 001  
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 28880 WORDS ( 113 PAGES)  
 DYNAMIC MEMORY: 20060 WORDS ( 77 PAGES)  
 ELAPSED TIME: 00:51:20  
 CNTSCAO.BIC,CNTSCAO.SEQ/-SP=SVC34/ML,TSV1C,TSV22C,TSV3B,TSV4,TSV7A,TSV6

....B1  
....C1  
....D1  
....E1  
....F1  
....G1  
....H1  
....I1  
....J1  
....K1  
....L1  
....M1  
....N1

....B2  
....C2  
....D2  
....E2  
....F2  
....G2  
....H2  
....I2  
....J2  
....K2  
....L2  
....M2  
....N2

....B3  
....C3  
....D3  
....E3  
....F3  
....G3  
....H3  
....I3  
....J3  
....K3  
....L3  
....M3  
....N3

....B4  
....C4  
....D4  
....E4  
....F4  
....G4  
....H4  
....I4  
....J4  
....K4  
....L4  
....M4  
....N4

....B5  
....C5  
....D5  
....E5  
....F5  
....G5  
....H5  
....I5  
....J5  
....K5  
....L5  
....M5  
....N5

....B6  
....C6  
....D6  
....E6  
....F6  
....G6  
....H6  
....I6  
....J6  
....K6  
....L6  
....M6  
....N6

....B7  
....C7  
....D7  
....E7  
....F7  
....G7  
....H7  
....I7  
....J7  
....K7  
....L7  
....M7  
....N7

....B8  
....C8  
....D8  
....E8  
....F8  
....G8  
....H8  
....I8  
....J8  
....K8  
....L8  
....M8  
....N8

....B9  
....C9  
....D9  
....E9  
....F9  
....G9  
....H9  
....I9  
....J9  
....K9  
....L9  
....M9  
....N9

....B10  
....C10  
....D10  
....E10  
....F10  
....G10  
....H10  
....I10  
....J10  
....K10  
....L10  
....M10  
....N10

....B11  
....C11  
....D11  
....E11  
....F11  
....G11  
....H11  
....I11  
....J11  
....K11  
....L11  
....M11  
....N11

....B12