

TSV05

TSV05 CTRL LT3
CVTSCAO

AH-T098A-MC
FICHE 2 OF 2

SEP 1982
COPYRIGHT © 1982
MADE IN USA



The main body of the document is a large, dense grid of data. Each cell in the grid contains small, faint text, likely representing a list of items or a detailed schedule. The text is too small to be legible in this scan. The grid is organized into approximately 15 columns and 25 rows. There are some larger, more distinct blocks of text or diagrams interspersed within the grid, particularly in the upper and lower right sections.



.REM_
IDENTIFICATION

PRODUCT ID: AC-T097A-MC
PRODUCT TITLE: CVTSCAO TSV05 CTRL LT3
AUTHOR: DICK GORDON
MAINTAINER: SCOTT SNOWDON
DATE: MARCH 08, 1982

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

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

COPYRIGHT (C) 1982,1982 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

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

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

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

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

1.2 SYSTEM REQUIREMENTS

PDP-11/23 PROCESSOR AND MEMORY
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE 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. CHQUS XXDP+ USERS MANUAL; DOCUMENT NUMBER AC-F348E-MC
DATE: 14 JULY 1980.
2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE; DOCUMENT NUMBER EK-TSV05-UG-001
DATE: AUGUST 1982
3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSV05-TM-001
DATE: AUGUST 1982
4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSV05-IN-001
DATE: AUGUST 1982

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

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

1.5 ASSUMPTIONS

ALL HARDWARE EXCEPT THE HARDWARE UNDER TEST IS ASSUMED TO WORK PROPERLY OR FALSE ERRORS CAN BE REPORTED.
 THE TAPE BEING USED ON THE TSV05 TRANSPORT IS A KNOWN GOOD REEL OF TAPE.
 CVTSAA AND CVTSBA 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 (CHQUS).

2.1 COMMANDS

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

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

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

2.1.1 OPERATOR COMMANDS

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

BOOT THE DIAGNOSTIC MEDIA

```
.R VTSC??
DIAG. RUN-TIME SERVICES REV D. APR 79
CVTSC-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 'DDDDD'.

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN

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

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

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

```
/FLAGS:LOE:IER:BOE
```

2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?" YOU MUST ANSWER "Y" AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 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 = 172520, VECTOR = 224

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

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

UNIT 0

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

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

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

UP TO 4 TSV05 CONTROLLERS PER 11/23 AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

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

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

UNITS (D) ? 8<CR>

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

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

UNIT 3
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 2<CR>
Q-FACTOR (O) 0 ? <CR>

UNIT 4
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 3<CR>
Q-FACTOR (O) 0 ? <CR>

UNIT 5
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 4<CR>
Q-FACTOR (O) 0 ? <CR>

UNIT 6
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 5<CR>
Q-FACTOR (O) 0 ? <CR>

UNIT 7
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 6<CR>
Q-FACTOR (O) 0 ? 1<CR>

UNIT 8
CSR ADDRESS (O) 160000<CR>
SUB-DEVICE # (O) ? 7<CR>
Q-FACTOR (O) 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 (D) ? 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 (D) ? 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
CVTSC HRD ERR 01610 ON UNIT 00 TST 016 SUB 002 PC: 040624
FIFO STATUS (IN WORD 9) INCORRECT AFTER WRITE FIFO

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

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

MESSAGE BUFFER ADDRESS = 047352

MESSAGE BUFFER CONTENTS:

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

ERROR MESSAGE EXAMPLE 2

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

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

TSSR = 100214

TSSR BITS SET: SC,SSR

TERMINATION CLASS CODE = UNRECOVERABLE ERROR

PACKET ADDRESS = 026420

PACKET WORD # = 140010

PACKET WORD # = 000010

PACKET WORD # = 000000

PACKET WORD # = 000024

ERROR MESSAGE EXAMPLE 3

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

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

4.0 PERFORMANCE AND PROGRESS REPORTS

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

SUCCESSFUL RUN EXAMPLE (PDP-11/23)

DR>STA/FLA:PNT:HOE

UNITS (D) ? 1

UNIT 0

DEVICE ADDRESS (0) 172520 ? <CR>

VECTOR (0) 224 ? <CR>

CHANGE SW (L) ? N<CR>

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

TST: 001 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 PDP-11/23 PROCESSOR WITH A LA34 CONSOLE.

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

TEST NUMBER	N/I SECS.	NUMBER ITER	DEF SECS.
1	3	10	7
2	3	8	5
3	38	250	212
4	60	300	240
5	60	300	240
6	120	360	240
7	120	600	480
8	22	90	68

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

Q.V. 7 MINUTES
DEFAULT 31 MINUTES

5.0 DEVICE INFORMATION TABLES

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

CHANGE HW (L) ? . . .
 # UNITS (D) ? <ENTER THE NUMBER OF M7196 CONTROLLERS
 PRESENT TO BE TESTED>
 UNIT 0
 DEVICE ADDRESS (O) 172520 ? <ENTER THE ADDRESS OF THE
 TSBA/TSDB REGISTER>
 VECTOR (O) 224 ? <ENTER ADDRESS OF INTERRUPT
 VECTOR>

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

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

CHANGE SW (L) ?

INHIBIT ITERATIONS (L) N ?

6.0 TEST SUMMARIES

TEST 1: INITIALIZE #4 TEST

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 11/23B 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 11/23B 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 BOUNDRIES, RECORD SIZES (UP TO 64K BYTES IF MEMORY SPACE IS AVAILIABLE), AND BYTE-SWAP (SWP) AND OPPOSITE (OPP) CONRTOL 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 11/23B 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

-

```

2          .TITLE  TSV2 - PROGRAM HEADER
3          .SBTTL  PROGRAM HEADER
4
10         .MCALL  SVC
11 000000 SVC          ; INITIALIZE SUPERVISOR MACROS
12         .ENABLE LC
13         .NLIST  BEX,CND
19 000000 .ENABL  ABS,AMA
20         .=2000
21 002000 BGNMOD  TSV2
002000
22 TSV2::
23
24 :++
25 : THE PROGRAM HEADER IS THE INTERFACE BETWEEN
26 : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
27 :--
28
29 002000 POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
30 002000 HEADER CVTSC,A,0,655.,0
002000 L$NAME::          ;DIAGNOSTIC NAME
002000 .ASCII /C/
002001 .ASCII /V/
002002 .ASCII /T/
002003 .ASCII /S/
002004 .ASCII /C/
002005 .BYTE 0
002006 .BYTE 0
002007 .BYTE 0
002010 L$REV::          ;REVISION LEVEL
002010 .ASCII /A/
002011 L$DEPO::        ;0
002011 .ASCII /0/
002012 L$UNIT::        ;NUMBER OF U'ITS
002012 .WORD 0
002014 L$TIML::        ;LONGEST TEST TIME
002014 .WORD 655.
002016 L$HPCP::        ;PTR. TO H.W. QUES.
002016 .WORD L$HARD
002020 L$SPCP::        ;PTR. TO S.W. QUES.
002020 .WORD L$SOFT
002022 L$HPTP::        ;PTR. TO DEF. H.W. PTABLE
002022 .WORD L$HW
002024 L$SPTP::        ;PTR. TO S.W. PTABLE
002024 .WORD L$SW
002026 L$LADP::        ;DIAG. END ADDRESS
002026 .WORD L$LAST
002030 L$STA::          ;RESERVED FOR APT STATS
002030 .WORD 0
002032 L$CO::           ;
002032 .WORD 0
002034 L$DTYP::        ;DIAGNOSTIC TYPE
002034 .WORD 0
002036 L$APT::         ;APT EXPANSION
002036 .WORD 0
002040 L$DTP::         ;PTR. TO DISPATCH TABLE
002040 .WORD L$DISPATCH

```

002042		LSPRIO::		;DIAGNOSTIC RUN PRIORITY
002042	000000	.WORD	0	
002044		LSENV1::		;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000	.WORD	0	
002046		LSEXP1::		;EXPANSION WORD
002046	000000	.WORD	0	
002050		LSMREV::		;SVC REV AND EDIT #
002050	003	.BYTE	C\$REVIS/	
002051	003	.BYTE	C\$EDIT	
002052		LSEF::		;IAG. EVENT FLAGS
002052	000000	.WORD	0	
002054	000000	.WORD	0	
002056		LSSPC::		
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	022754	.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	022442	.WORD	L\$AU	
002072		L\$DUT::		;PTR. TO DROP UNIT CODE
002072	022540	.WORD	L\$DU	
002074		L\$LUN::		;LUN FOR EXERCISERS TO FILL
002074	000000	.WORD	0	
002076		L\$DESP::		;POINTER TO DIAG. DESCRIPTION
002076	003400	.WORD	L\$DESC	
002100		L\$LOAD::		;GENERATE SPECIAL AUTOLOAD EMT
002100	104035	EMT	E\$LOAD	
002102		L\$ETP::		;POINTER TO ERR TBL
002102	000000	.WORD	0	
002104		L\$ICP::		;PTR. TO INIT CODE
002104	021646	.WORD	L\$INIT	
002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
002106	022726	.WORD	L\$CLEAN	
002110		L\$ACP::		;PTR. TO AUTO CODE
002110	022646	.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	

32
33
34
35
36
37
38
39 002122
002122 000010
002124
002124 023536
002126 024652
002130 027332
002132 034322
002134 046436
002136 055354
002140 074676
002142 104714
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


```
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$HW/2
    002146                               L$HW::
    002146                               DFPTBL::
50
51 002146 172520                         .WORD  172520      ; 1ST (OF 2) REGISTERS.
52 002150 000224                         .WORD  224          ; INTERRUPT VECTOR
53 002152 000200                         .WORD  PRI04       ; INTERRUPT PRIORITY.
54 002154                               ENDHW
    002154                               L10000:
```

```
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      .WORD L10001-L$SW/2
      002156      L$SW::
      002156      SFPTBL::
63
64 002156      000000  TRANSTST::      .WORD  0      ; ENABLE TEST OF TRANSPORT(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
```

7
8
13
19
20 002166
002166
21
22
23
24
25
26
27
28
29
33 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 DIFINITIONS

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.	: START COMMAND WAS ISSUED
000037	EF.RESTART==	31.	: RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE==	30.	: CONTINUE COMMAND WAS ISSUED
000035	EF.NEW==	29.	: A NEW PASS HAS BEEN STARTED
000034	EF.PWR==	28.	: A POWER-FAIL/POWER-UP OCCURRED

```
                ; 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          IDU==     40
000100          ISR==    100
000200          UAM==    200
000400          BOE==    400
001000          PNT==   1000
002000          PRI==   2000
004000          IXE==   4000
010000          IBE==  10000
020000          IER==  20000
040000          LOE==  40000
100000          HOE== 100000
```

34
35 002166

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



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

```
SDPAR3= 172266
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
```

```

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

```

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



```

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

```

211
212
213      ;+
214      ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
215      ;-
216
217      000006      XST0== 6      ;EXTENDED STATUS REGISTER 0 (WORD 4)
218      000010      XST1== 8      ;EXTENDED STATUS REGISTER 1 (WORD 5)
219      000012      XST2== 10     ;EXTENDED STATUS REGISTER 2 (WORD 6)
220      000014      XST3== 12     ;EXTENDED STATUS REGISTER 3 (WORD 7)
221      000016      XST4== 14     ;EXTENDED STATUS REGISTER 4 (WORD 8)
222
223
224      ;+
225      ;
226      ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
227      ;
228      ;-
229
230      000002      PKLOW = 2      ;LOW ORDER CHARACTERISTIC DATA POINTER
231      000004      PKHI  = 4      ;HIGH ORDER CHARACTERISTIC DATA POINTER
232      000006      PKBCNT = 6     ;NUMBER OF BYTES IN DATA PACKET
233
234      000010      EXBCNT=10     ;NUMBER OF BYTES IN EXTENDED DATA PACKET
235
236      ;+
237      ;DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
238      ;-
239      000000      BSELO = 0      ;BYTE 0
240      000001      BSEL1 = 1      ;BYTE 1
241      000002      SEL2  = 2      ;WORD 2
242      000004      SELDATA = 4    ;WORD 3
243
244      ;+
245      ;BSELO SELECT CODES FOR WRITE SUBSYSTEM COMMAND
246      ;-
247      000000      PW.NOP      = 0      ;NO-OP
248      000001      PW.RDRAM    = 1      ;READ RAM
249      000002      PW.WTRAM    = 2      ;WRITE RAM
250      000003      PW.RFIFO    = 3      ;READ FIFO
251      000004      PW.WFIFO    = 4      ;WRITE FIFO
252      000005      PW.RDSTAT   = 5      ;READ STATUS
253      000006      PW.WCTL     = 6      ;WRITE TAPE CONTROL
254      000007      PW.WFMT     = 7      ;WRITE TAPE FORMAT
255      000010      PW.WMISC    = 10     ;WRITE MISCELLANEOUS
256      000011      PW.WNPR     = 11     ;WRITE NPR CONTROL
257      000020      PW.D22     = 20     ;DO MICROTTEST 22
258      000021      PW.D11     = 21     ;DO MICROTTEST 11
259      000022      PW.D13     = 22     ;DO MICROTTEST 13
260      000023      PW.NO1311  = 23     ;DISABLE MICROTTEST 11 AND 13
261      000024      PW.RDXT     = 24     ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
262
263      ;+
264      ;BSEL1 CODES FOR WRITE TAPE CONTROL
265      ;-
266      000200      WC.IFAD     = BIT7   ;IFAD - FORMATTER ADDRESS
267      000100      WC.IOTAD    = BIT6   ;ITADO - TRANSPORT ADDRESS BIT 0

```

```

268      000040      WC.I1TAD      = BIT5      ;ITAD1 - TRANSPORT ADDRESS BIT 1
269      000020      WC.I5RESV     = BIT4      ;IRESV5 - RESERVED #5
270      000010      WC.IREW      = BIT3      ;IREW - REWIND
271      000004      WC.IRWU      = BIT2      ;IRWU - REWIND AND UNLOAD
272      000002      WC.IFEN      = BIT1      ;IFEN - FORMATTER ENABLE
273      000001      WC.IGO       = BIT0      ;GO
274
275      ;+
276      ;BSEL1 CODES FOR WRITE FORMAT
277      ;-
278      000200      WF.IHISP     = BIT7      ;IHISP - HIGH SPEED
279      000100      WF.IWRT      = BIT6      ;IWRT - WRITE
280      000040      WF.IREV      = BIT5      ;IREV - REVERSE
281      000020      WF.IWFM      = BIT4      ;IWFM - WRITE FILE MARK
282      000010      WF.IEDIT     = BIT3      ;IEDIT - EDIT
283      000004      WF.IERASE    = BIT2      ;IERASE - ERASE
284      000002      WF.I3RESV    = BIT1      ;IRESV3 - RESERVED #3
285      000001      WF.I4RESV    = BIT0      ;IRESV4 - RESERVED #4
286
287
288      ;+
289      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
290      ;-
291      000200      MS.EXT       = BIT7      ;INVERT SENSE OF EXTENDED FEATURES SWITCH
292      000020      MS.RSFIFO     = BIT4      ;RESET FIFO AND INPUT PARITY ERRORR
293      000010      MS.RSTAPE     = BIT3      ;RESET TAPE STATUS IN 2 FLIP-FLOPS
294      000006      MS.ATTN      = BIT2!BIT1 ;ATTENTION TRIGGER FIELD
295      000001      MS.RSD       = BIT0      ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
296
297      ;+
298      ; MS.ATTN SUBCODES
299      ;-
300      000000      MSA.NOP      = 0*2      ;NO-OP (NOTHING TRIGGERED)
301      000002      MSA.VOL      = 1*2      ;SIMULATE ON-LINE/OFF-LINE TRANSITION
302      000004      MSA.NRAM     = 2*2      ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
303      000006      MSA.FRAM     = 3*2      ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
304
305      ;+
306      ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
307      ;-
308      000200      NP.IR        = BIT7      ;INTERRUPT REQUEST (0-1 TRANSITION)
309      000100      NP.OUT       = BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
310      000040      NP.LOOP      = BIT5      ;ENABLE TRANSPORT LOOPBACK
311      000020      NP.WRP       = BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
312
313      ;+
314      ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
315      ;-
316      000200      S2.DIM       = BIT7      ;WORD #9 BYTE 2 DATA IN MISS
317      000100      S2.ILW       = BIT6      ;ILW H
318      000040      S2.OUTRDY    = BIT5      ;OUT RDY H
319      000020      S2.INRDY     = BIT4      ;IN RDY H
320      000010      S2.ATIMR     = BIT3      ;TIMER A FLAG H
321      000004      S2.BTIMR     = BIT2      ;TIMER B FLAG H
322      000003      S2.UNDEF     = BIT1+BIT0 ;(UNDEFINED)
323      100000      S1.PARIN     = BIT15     ;WORD #8 BYTE 1 PARIN H
324      040000      S1.I2RESV    = BIT14     ;IRESV2
325      020000      S1.I1RESV    = BIT13     ;IRESV1
326      010000      S1.IEOT      = BIT12     ;IEOT L
    
```

TSV05 REGISTER AND PACKET DEFINITIONS

325	004000	S1.IIDENT	= BIT11	:	IIDENT	H
326	002000	S1.ICER	= BIT10	:	ICER	H
327	001000	S1.IFMK	= BIT9	:	IFMK	H
328	000400	S1.IHER	= BIT8	:	IHER	H
329	000200	S0.ISPEED	= BIT7	:	ISPEED	H
330	000100	S0.IRDY	= BIT6	WORD #8 BYTE 0	IRDY	L
331	000040	S0.IONL	= BIT5	:	IONL	L
332	000020	S0.ILDY	= BIT4	:	ILDY	L
333	000010	S0.IDBY	= BIT3	:	IDBY	L
334	000004	S0.IRWD	= BIT2	:	IRWD	L
335	000002	S0.IFBY	= BIT1	:	IFBY	L
336	000001	S0.IFPT	= BIT0	:	IFPT	L
337				:		
338				:		

```
340 .SBTTL SPECIAL MACROS AND OPDEFS.
341
342
343 :+
344 :SAVE GENERAL REGS 1 TO 5
345 :-
346
347 .MACRO SAVREG
348 JSR R5,REGSAV
349 .ENDM
350
351 :+
352 : MACRO TO FORCE AN ERROR
353 :-
354 .MACRO FORCERROR TAG,NOTSSR
355 .NLIST
356 .IIF NDF LISTALL, .NLIST
357 .LIST
358 .IF B NOTSSR
359 MOV TSSR(R5),R1 ;READ TSSR
360 .ENDC
361 MOV FORCER,FORCER ;IS FORCER SET? (LEAVE C BIT ALONE)
362 BNE TAG ;BR IF YES
363 .NLIST
364 .IIF NDF LISTALL, .LIST
365 .LIST
366 .ENDM
367
368 :+
369 : MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
370 : WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
371 : SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
372 : FORCER TO 17777
373 : TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
374 :-
375 .MACRO FORCEEXIT TAG
376 .NLIST
377 .IIF NDF LISTALL, .NLIST
378 .LIST
379 MOV FORCER,FORCER ;IS FORCER NEGATIVE?
380 BMI TAG ;BR IF YES
381 .NLIST
382 .IIF NDF LISTALL, .LIST
383 .LIST
384 .ENDM
385 :+
386 : MACRO TO INCREMENT ERROR COUNTS
387 :-
388 .MACRO NEXT.ERRNO
389 .NLIST
390 :::IIF NDF LISTALL, .NLIST
391 ERRNO=ERRNO+1
392 :::IIF NDF LISTALL, .LIST
393 .LIST
394 .ENDM
395
396 :+
```

397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420

;MACRO TO PERFORM XOR

;-

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

000000

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

;
; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
;

002166 000000

```
FORCER:: 0 ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -  
; - BY THE MACRO 'IFERROR'). AN ERROR NEED NOT -  
; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
```


.SBTTL GLOBAL DATA SECTION

422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461

002170 000000
002172 000000
002174 000000
002176 000000
002200 000224
002202 000200
002204 000000
002206 000000
002210 000000
002212 000000
002214 000000
002216 000000
002220 000000
002222 000000
002224 000000
002226 000000
002230 000000
002232
002272 000000
002274 000000
002276 000000
002300 000000
002302 000000
002304 000000
002306 000000
002310 000000
002312
002456
002622

```

:++
:THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
:IN MORE THAN ONE TEST.
:--

:THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
:SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.

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

```

463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516

002742
002742 000000
002744 177777
002746 000001
002750 000002
002752 000004
002754 000010
002756 000020
002760 000040
002762 000100
002764 000200
002766 000400
002770 001000
002772 002000
002774 004000
002776 010000
003000 020000
003002 040000
003004 100000
003006 177776
003010 177775
003012 177773
003014 177767
003016 177757
003020 177737
003022 177677
003024 177577
003026 177377
003030 176777
003032 175777
003034 173777
003036 167777
003040 157777
003042 137777
003044 077777
003046 125252
003050 052525
003052

.SBTTL TSTBLK - TEST DATA TABLE

```

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

```

TSTBLK::

```

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

```

TBLEND==.

```

518 .SBTTL GLOBAL ENVIRONMENT STORAGE
519
520 .STORAGE FOR DEVICE REGISTERS
521
522 003052 000000 100000 000000 DUMMY: 0,100000,0,0 ;DUMMY DEVICE REGISTERS...
523 003062 000000 000000 000000 0,0,0,0,0,0,0,0
524 ;...FOR MULTI-UNIT CHECKOUT.
525
526
527 003102 000000 DUFLG:: .WORD 0 ;'DROPPED UNIT' FLAG.
528 ;INHIBITS CODE IN 'CLEAN-UP'.
529 003104 000000 NODEV:: .WORD 0 ;FLAG TO SAY NO DEVICE.
530
531 003106 000000 TEMP1:: .WORD 0 ;SOME TEMP LOCATIONS.
532 003110 000000 TEMP2:: .WORD 0
533 003112 000000 XXCOMM:: .WORD 0 ;XXDP+ COMM BLOCK POINTER.
534 003114 000000 FREE:: .WORD 0 ;1ST FREE MEMORY ADDRESS...
535 003116 000000 FRESIZ:: .WORD 0 ;...AND SIZE (IN WORDS).
536 003120 000000 FREEHI: .WORD 0 ;LAST WORD IN FREE SPACE
537 003122 000000 KTFLG:: .WORD 0 ;KT11, MEM AVAIL FLAG -
538 ;- .WORD 0 = <24K OR NO KT -
539 ;- NZ = >24K AND KT.
540 003124 000000 KTENABLE:: .WORD 0 ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
541 003126 000000 NXMFLG:: .WORD 0 ;SET IF WE CAN TEST CLEARED OTHERWISE
542 003130 000000 NXMLO:: .WORD 0 ;NXM LO ADDRESS BITS
543 003132 000000 NXMHI:: .WORD 0 ;NXM HI ADDRESS BITS FOR DAL'S 16-21
544 003134 000000 T23A:: .WORD 0 ;11/23A FLAG
545 003136 000000 T23B:: .WORD 0 ;11/23B FLAG
546 003140 000000 T3BFLG:: .WORD 0 ;TEST 3B FLAG ^0
547 003142 002000 PST32W:: .WORD 2000 ;32W BLOCK ADDRESS FOR 32K START
548 003144 000000 SIFLAG:: .WORD 0
549 003146 000000 BADDAT:: .WORD 0 ;ACTUAL DATA
550 003150 000000 GDDAT:: .WORD 0 ;EXPECTED DATA
551 003152 000000 LOOPFL:: .WORD 0
552 003154 CTAB:: ;CONFIGURATION TABLES.
553 003154 000000 CTABM:: .WORD 0 ;CONFIG WORK.
554 003156 000000 .WORD 0
555 003160 000000 .WORD 0
556 003162 000000 .WORD 0
557 003164 177777 .WORD -1 ;END OF MEM TABLE.
558 003166
559 ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
560
561 : 0 = UNIT NOT TESTED
562 : 100000 = UNIT ONLINE, NO ERRORS
563 : 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
564 : 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
565 : 160001 = UNIT DROPPED, NOT IDLE AT START
566 : 14XXXX = UNIT DROPPED, ENCOUNTERED XXXX ERRORS
567
568 003166 000000 ERTABL: .BLAW 64.
569 003366 000000 ERTABE: .WORD 0
570
571 003370 000000 SKIPT: .WORD 0 ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST
    
```

573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
602
603
604
605
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644

003372			
003372	124	123	126
003372			
003400			
003400	052	052	052
003400			
003500	003540	003543	003547
003520	003601	003605	003611
003540	123	103	000
003543	102	111	105
003547	123	103	105
003553	122	115	122
003557	116	130	115
003563	116	102	101
003567	102	111	124
003574	102	111	124
003601	123	123	122
003605	117	106	114
003611	102	111	124
003616	102	111	124
00362	102	111	124
003630	102	111	124
003635	102	111	124
003642	102	111	124
003650	124	123	123
003703	124	123	123
003736	040	040	116
003775	045	101	040
004016	045	101	040
004056	045	101	040

```

.SBTTL GLOBAL TEXT MESSAGES
:++
: THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
: MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
: MORE THAN ONE TEST.
:--

:++
: NAMES OF DEVICES SUPPORTED
:--

      DEVTYP <TSV05>
LSDVTYP: .ASCIZ /TSV05/
        .EVEN

:++
: TEST DESCRIPTION
:--

      DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - CHK CABLES-TRANSPORT IF ERR ****>
L$DESC: .ASCIZ /**** TSV05 LOGIC DIAGNOSTIC - CHK CABLES-TRANSPORT IF ERR ****/
        .EVEN

:++
: BIT TO ASCII CONVERSION FOR TSSR REGISTER
:--

TSSRBIT: .WORD 1$,2$,3$,4$,5$,6$,7$,8$
          .WORD 9$,10$,11$,12$,13$,14$,15$,16$
1$: .ASCIZ 'SC'
2$: .ASCIZ 'BIE'
3$: .ASCIZ 'SCE'
4$: .ASCIZ 'RMR'
5$: .ASCIZ 'NXM'
6$: .ASCIZ 'NBA'
7$: .ASCIZ 'BIT9'
8$: .ASCIZ 'BIT8'
9$: .ASCIZ 'SSR'
10$: .ASCIZ 'OFL'
11$: .ASCIZ 'BIT5'
12$: .ASCIZ 'BIT4'
13$: .ASCIZ 'BIT3'
14$: .ASCIZ 'BIT2'
15$: .ASCIZ 'BIT1'
16$: .ASCIZ 'BIT0'
        .EVEN
SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
SFHERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
NXR: .ASCIZ / NON-EXISTANT DEVICE REGISTER/
NXRX: .ASCIZ /%A ADDRESS: %06/
TSSX: .ASCII /%A TSBA,TSSR EXP'D: %06%A,%06%N/
      .ASCIZ /%A TSBA,TSSR REC'D: %06%A,%06/

```

```

645 004115 045 116 045 FUSI: .ASCII /%N%/
646 004121 040 040 125 USI: .ASCIZ / UNEXPECTED INTERRUPT/
647 004150 040 040 111 NSI: .ASCIZ / INTERRUPT EXPECTED, NOT RECEIVED/
648 004213 045 116 045 FNOINTR: .ASCII /%N%/
649 004217 040 040 116 NOINTR: .ASCIZ / NO INTERRUPT WAS GENERATED/
650 004254 040 040 111 IFAULT: .ASCIZ / INTERRUPT FAULT/
651 004276 045 101 040 INTX: .ASCIZ /%A CPU PC: %06% TSBA: %06/
652 004333 040 040 042 NOINIT: .ASCIZ / 'BUS-INIT' DIDN'T INITIALIZE CONTROLLER/
653 004405 040 040 042 NSINIT: .ASCIZ / 'SOFT-INIT' DIDN'T INITIALIZE THE DPU/
654 004455 040 040 042 BRINIT: .ASCIZ / 'BUS-RESET' DIDN'T INITIALIZE THE DPU/
655
656 004525 000 NUL: .ASCIZ //
657 004526 045 116 000 NULCR: .ASCIZ /%N/
658 004531 045 101 040 EXPGOT: .ASCIZ /%A EXP'D: %06%, REC'D: %06/
659 004565 045 116 045 EXPGT2: .ASCIZ /%N% EXP'D: %06%, %06%N% REC'D: %0%, %06/
660 004641 045 101 040 DUAD12: .ASCIZ /%A REG(W) WRITTEN TO: %06% REG(R) READ: EXP'D: %06%, REC'D: %06/
661 004743 122 101 115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
662 005011 040 040 103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
663 005054 127 122 111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
664 005111 124 123 123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
665 005204 124 123 123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
666 005276 106 101 124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
667 005370 105 122 122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
668 005456 045 116 045 NOMEM: .ASCIZ '%N% ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****%N'
669 005552 045 116 045 M8186: .ASCIZ '%N% ***** 11/23A SYSTEM *****%N'
670 005643 045 116 045 M8189: .ASCIZ '%N% ***** 11/23B SYSTEM *****%N'
671 .EVEN
672
673
674

```

```

676
677
678
679
680
681
682
683
684 005734
    005734
685 005734
    005734 013746 003104
    005740 012746 003775
    005744 012746 000002
    005750 010600
    005752 104415
    005754 062706 000006
686 005760 004737 005766
687 005764
    005764
    005764 104423
688
689
690
691
692
693
694 005766 005727
695 005770 000000
696 005772 001402
697 005774 004777 177770
698 006000
    006000 012746 004526
    006004 012746 000001
    006010 010500
    006012 104415
    006014 062706 000004
699 006020 000207
    
```

.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.
:--
    
```

```

        BGNMSG  NXRERR          ;NON-EXISTANT DEVICE REGISTER.
NXERR:: PRINTX  #NXRX,NODEV     ;NODEV = NEXM ADDRESS.
        MOV     NODEV,-(SP)
        MOV     #NXRX,-(SP)
        MOV     #2,-(SP)
        MOV     SP,R0
        TRAP   C$PNTX
        ADD    #6,SP
        JSR   PC,EXTEND        ; PRINT EXTENSION IF REQUIRED.
        ENDMSG
L10002: TRAP   C$MSG
    
```

```

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

```

EXTEND: TST     (PC)+
EXTA:   0
        BEQ    1$
        JSR   PC,@EXTA        ; APPEND EXTENSION TEXT.
        PRINTX #NULCR         ; PRINT A BLANK LINE
        MOV   #NULCR,-(SP)
        MOV   #1,-(SP)
        MOV   SP,R0
        TRAP C$PNTX
        ADD  #4,SP
        RTS  PC
    
```


702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719

.SBTTL PRITSSR - PRINT TSSR CONTENTS

```

:ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
:THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
:BY A MESSAGE PRINTING ROUTINE

```

:INPUTS:

R1 CONTENTS OF TSSR

:SUBORDINATE ROUTINES:

CHKAMB CHECK FOR AMBIGUOUS CONTENTS

PRITSSR:

```

720 006022
721 006022
722 006026 010104
723 006030
    006030 010446
    006032 012746 006505
    006036 012746 000002
    006042 010600
    006044 104414
    006046 062706 000006
724 006052 010400
725 006054 004737 016134
726 006060 103410
727 006062
    006062 012746 006725
    006066 012746 000001
    006072 010600
    006074 104415
    006076 062706 000004
728 006102 010403
729 006104 042703 001476
730 006110 001434
731 006112 012702 002622
732 006116 012701 003500
733 006122 005703
734 006124 001413
735 006126 000241
736 006130 006103
737 006132 103006
738 006134 011100
739 006136 112022
740 006140 001376
741 006142 112762 000054 177777
742 006150 005721
743 006152 000763
744 006154 105042
745 006156
    006156 012746 002622
    006162 012746 006676

```

```

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

```

```

006166 012746 000002      MOV      #2,-(SP)
006172 010600      MOV      SP,R0
006174 104415      TRAP    C$PNTX
006176 062706 000006      ADD      #6,SP
746
747 006202 010403      20$:    MOV      R4,R3          ;GET THE TSSR CONTENTS
748 006204 042703 177761      BIC      #^CTERCLS,R3    ;CLEAR ALL BUT TERMINATION
749 006210 016303 006766      MOV      TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
750 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$PNTX
      006232 062706 000006      ADD      #6,SP
751 006236 010403      MOV      R4,R3          ;TSSR CONTENTS AGAIN
752 006240 042703 177717      BIC      #^CFATERR,R3    ;CLEAR ALL BUT FATAL TERMINATION
753 006244 001416      BEQ     25$             ;DON'T PRINT IF ZERO
754 006246 006203      ASR     R3
755 006250 006203      ASR     R3
756 006252 006203      ASR     R3
757 006254 016303 007326      MOV      TSFCOD(R3),R3   ;ALINE TERMINATION CODE FOR INDEX
758 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$PNTX
      006276 062706 000006      ADD      #6,SP
759 006302 042704 176377      25$:    BIC      #^CHIADDR,R4    ;CLEAR ALL BUT EXTENDED ADDRESS
760 006306 001411      BEQ     30$             ;DON'T PRINT IF ZERO
761 006310      PRINTX  #TEXASC,R4         ;PRINT THE EXTENDED ADDRESS BITS
      006310 010446      MOV      R4,-(SP)
      006312 112746 006525      MOV      #TEXASC,-(SP)
      006316 012746 000002      MOV      #2,-(SP)
      006322 010600      MOV      SP,R0
      006324 104415      TRAP    C$PNTX
      006326 062706 000006      ADD      #6,SP
762 006332 013703 002170      30$:    MOV      EPRTSW,R3        ;PRINT MEASGE BUFFER ADDRESS
763 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$PNTX
      006350 062706 000004      ADD      #4,SP
764 006354 000207      RTS      PC                ;RETURN TO CALLER
765
776 006356      045      116      045  EPRT1:  .ASCIZ  '%NZA *****CHECK CABLES BETWEEN M7196 AND TRANSPORT*****'
777 006446      045      116      045  EPRT2:  .ASCIZ  '%NZA *****CHECK TRANSPORT*****'
783 006505      045      116      045  TSSRFOR: .ASCIZ  '%NZA TSSR = %06'
784 006525      045      116      045  TEXASC:  .ASCIZ  '%NZA Extended Address Bits = %06'
785 006566      045      116      045  TCOASC:  .ASCIZ  '%NZA Termination Class Code = %T'
786 006627      045      116      045  TFCASC:  .ASCIZ  '%NZA Fatal Termination Class Code = %T'
787 006676      045      116      045  TSSDEF:  .ASCIZ  '%NZA TSSR Bits Set: %T'
788 006725      045      116      045  AMBTSSR: .ASCIZ  '%NZA TSSR Contents Are Ambiguous'
789
790 006766 007006 007031 007057 TCOCOD: .EVEN
      .WORD  1$,2$,3$,4$,5$,6$,7$,8$

```

791	007006	116	157	162	1\$:	.ASCIZ	'Normal Termination'
792	007031	124	145	162	2\$:	.ASCIZ	'Termination Condition'
793	007057	124	141	160	3\$:	.ASCIZ	'Tape Status Alert'
794	007101	106	165	156	4\$:	.ASCIZ	'Function Reject'
795	007121	122	145	143	5\$:	.ASCIZ	'Recoverable Error - Tape Position One Record Down'
796	007203	122	145	143	6\$:	.ASCIZ	'Recoverable Error - Tape Was Not Moved'
797	007252	125	156	162	7\$:	.ASCIZ	'Unrecoverable Error'
798	007276	106	141	164	8\$:	.ASCIZ	'Fatal Controller Error'
799						.EVEN	
800							
801	007326	007336	007372	007403	TSFCOD:	.WORD	1\$,2\$,3\$,4\$
802	007336	111	156	164	1\$:	.ASCIZ	'Internal Diagnostic Failure'
803	007372	122	145	163	2\$:	.ASCIZ	'Reserved'
804	007403	102	165	163	3\$:	.ASCIZ	'Bus Interface or Sanity Check Error'
805	007447	122	145	163	4\$:	.ASCIZ	'Reserved'
806						.EVEN	

852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884

.SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

```

:
:
: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::

```

      SAVREG          ;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     ;RETURN TO CALLER

```

```

007702
007702
007706 010203
007710
007720 012700 177400
007724 040001
007726 040002
007730 040003
007732
007732 010346
007734 010146
007736 010246
007740 012746 007764
007744 012746 000004
007750 010600
007752 104414
007754 062706 000012
007760 010300
007762 000207
007764 045 116 045 XORBFOR:

```

.ASCIZ '%N% EXPD: %03% RECV: %03% XOR: %03'

.EVEN

886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913

010032
010032
010036 010203
010040
010050
010050 010346
010052 010146
010054 010246
010056 012746 010102
010062 012746 000004
010066 010600
010070 104414
010072 062706 000012
010076 010300
010100 000207

045 XORFOR: .ASCIZ 'XN% EXPD: %06% RECV: %06% XOF: %06'
 .EVEN

.SBTTL PRI XOR - PRINT EXPD, RECV AND XOR

```

: +
: PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
: THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
: INPUTS:
:       R1      RECEIVED DATA
:       R2      EXPECTED DATA
: OUTPUT:
:       R0      XOR OF EXPECTED/RECEIVED DATA
: -
    
```

```

PRI XOR::
    SAVREG                ;SAVE THE REGISTERS
    MOV R2,R3             ;EXPECTED DATA
    XOR R1,R3             ;FORM THE EXCLUSIVE OR
    PRINTB #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
    MOV R3,-(SP)
    MOV R1,-(SP)
    MOV R2,-(SP)
    MOV #XORFOR,-(SP)
    MOV #4,-(SP)
    MOV SP,R0
    TRAP C$PNTB
    ADD #12,SP
    MOV R3,R0             ;R0 HAS XOR ON RETURN
    RTS                  ;RETURN TO CALLER
    
```

```

915 .SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
916
917 :+
918 :
919 :ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
920 :THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
921 :
922 :INPUTS:
923 :
924 :       R0      OCTAL VALUE TO CONVERT
925 :       R1      TABLE OF POINTERS TO ASCII EQUIVALENT
926 :
927 :-
928
929 PRIEQU:
930 010150      SAVREG
931 010154 000207  RTS      PC      ;SAVE THE REGISTERS
932                                     ;RETURN TO CALLER
933
934
935
936 .SBTTL PRIRAM - PRINT RAM ADDRESS
937 :+
938 :
939 :PRINT CONTROLLER RAM ADDRESS.
940 :THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
941 :
942 :INPUTS:
943 :
944 :       R4      RAM ADDRESS
945 :
946 :-
947 PRIRAM:
948 010156      SAVREG
949 010162      PRINTB #RAMFOR,R4 ;SAVE R1-R5 UNTIL NEXT RETURN
950 010162 010446  MOV      R4,-(SP) ;PRINT RAM ADDRESS IN ERROR
951 010164 012746 010206  MOV      #RAMFOR,-(SP)
952 010170 012746 000002  MOV      #2,-(SP)
953 010174 010600      MOV      SP,R0
954 010176 104414      TRAP     C$PNTB
955 010200 062706 000006  ADD      #6,SP
956 010204 000207      RTS      PC      ;RETURN
957
958 010206      045      116      045 RAMFOR: .ASCIZ '%N%A CONTROLLER RAM ADDRESS = %06'
959                                     .EVEN
960
961
962 .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
963 :+
964 :
965 :PRINT MEMORY ADDRESS
966 :THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
967 :
968 : IMPLICIT INPUTS
969 :
970 :       ERRHI   - HIGH ORDER ADDRESS
971 :       ERRLO   - LOW ORDER ADDRESS
    
```



```

966
967
968 010250
969 010250
970 010254 013700 002226
971 010260 013701 002230
972 010264 010102
973 010266 006101
974 010270 006100
975 010272
    010272 010246
    010274 010046
    010276 012746 010320
    010302 012746 000003
    010306 010600
    010310 104414
    010312 062706 000010
976 010316 000207
977
978 010320 045 116 045 PRIA0: .ASCIZ 'XN% MEMORY ERROR ADDRESS = %01%05'
979 .EVEN
980
981
982 .SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
983
984
985 :PRINT MEMORY ADDRESS
986 :THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
987
988 : IMPLICIT INPUTS
989
990 ERRHI - HIGH ORDER ADDRESS
991 ERRLO - LOW ORDER ADDRESS
992
993
994 010364
995 010364
996 010370 013702 002226
997 010374 013701 002230
998
999
1000
1001 010400
    010400 010146
    010402 012746 010446
    010406 012746 000002
    010412 010600
    010414 104414
    010416 062706 000006
1002 010422
    010422 010246
    010424 012746 010511
    010430 012746 00J002
    010434 010600
    010436 104414
    010440 062706 000006
1003 010444 000207

```

```

:
:-
PRIADD:
    SAVREG
    MOV ERRHI,R0 ;SAVE R1-R5 UNTIL NEXT RETURN
    MOV ERRLO,R1 ;GET HIGH ADDRESS
    MOV R1,R2 ;GET LOW ADDRESS
    ROL R1 ;COPY LOW ADDRESS
    ROL R0 ;SHIFT BIT 15 TO C BIT
    PRINTB #PRIA0,R0,R2 ;SHIFT INTO HIGH ORDER
    MOV R2,-(SP) ;PRINT MEMORY ADDRESS IN ERROR
    MOV R0,-(SP)
    MOV #PRIA0,-(SP)
    MOV #3,-(SP)
    MOV SP,R0
    TRAP C$PNTB
    ADD #10,SP
    RTS PC ;RETURN

045 PRIA0: .ASCIZ 'XN% MEMORY ERROR ADDRESS = %01%05'
.EVEN

.SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
:
:
:PRINT MEMORY ADDRESS
:THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
:
: IMPLICIT INPUTS
:
: ERRHI - HIGH ORDER ADDRESS
: ERRLO - LOW ORDER ADDRESS
:
:-
PRITADD:
    SAVREG
    MOV ERRHI,R2 ;SAVE R1-R5 UNTIL NEXT RETURN
    MOV ERRLO,R1 ;GET HIGH ADDRESS
    :MOV R1,R2 ;GET LOW ADDRESS
    :ROL R1 ;COPY LOW ADDRESS
    :ROL R0 ;SHIFT BIT 15 TO C BIT
    PRINTB #PRIT0,R1 ;SHIFT INTO HIGH ORDER
    MOV R1,-(SP) ;PRINT MEMORY ADDRESS LOW IN ERROR
    MOV #PRIT0,-(SP)
    MOV #2,-(SP)
    MOV SP,R0
    TRAP C$PNTB
    ADD #6,SP
    PRINTB #PRIT1,R2 ;PRINT MEMORY ADDRESS HIGH IN ERROR
    MOV R2,-(SP)
    MOV #PRIT1,-(SP)
    MOV #2,-(SP)
    MOV SP,R0
    TRAP C$PNTB
    ADD #6,SP
    RTS PC ;RETURN

```


010666	005367	177756		DEC	-22(PC)	
010672	001367			BNE	.-20	
1063 010674	005337	010750		DEC	SDELAY	:BUMP DELAY COUNTER DOWN
1064 010700	001356			BNE	15\$:BR, IF MORE DELAY
1065 010702	000411			BR	60\$:BR IF TROUBLE CARRY = CLEAR
1066 010704	016501	000002	20\$:	MOV	TSSR(R5),R1	:READ TSSR
1067 010710	012702	000200		MOV	#SSR,R2	:SET UP EXPECTED
1068 010714	020201		25\$:	CMP	R2,R1	:ARE THEY OK
1069 010716	001401			BEQ	40\$:BR, IF EQUAL = OK
1070 010720	000402			BR	60\$:TROUBLE EXIT
1071 010722	000261		40\$:	SEC		:SET CARRY NO TROUBLE
1072 010724	000401			BR	70\$:EXIT
1073 010726	000241		60\$:	CLC		:CARRY CLEAR = ERROR
1074 010730			70\$:			
1075 010730	010400			MOV	R4,R0	:PASS PACKET ADDRESS
1076 010732	000207			RTS	PC	:RETURN

```
1078  
1079  
1080  
1081  
1082  
1084            010740  
1086  
1087  
1088 010740 000000  
1089  
1090 010742 000000  
1091 010744 000000  
1092 010746 000000  
1093 010750 000000  
1094
```

```
                  :  
                  :  
                  :PACKET FOR SPACE COMMAND  
                  :  
                  :          .=<.+10>&177770  
                  :  
                  :COMMAND WORD  
80$:          .WORD  
                  :NUMBER OF RECORDS TO BE SPACED OVER WORD  
90$:          .WORD  
                  .WORD  
                  .WORD  
SDELAY: .WORD    0                           :DELAY COUNTER  
                  .EVEN
```

1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128 010752
1129 010752
1130 010756 005037 002220
1131 010762 005037 002216
1132 010766 010465 000000
1133 010772 004737 016426
1134 010776 103401
1135 011000 000435
1136 011002 016501 000002
1137 011006 012702 000200
1138 011012 032701 000100
1139 011016 001402
1140 011020 052702 000100
1141 011024 020201
1142 011026 001401
1143 011030 000421
1144 011032 062704 000010
1145 011036 011403
1146 011040 032763 000200 000012
1147 011046 001402
1148 011050 005237 002216
1149 011054
1150 011054 032763 000100 000012
1151 011062 001402
1152 011064 005237 002220

.SBTTL WRTCHR - WRITE CHARACTERISTICS COMMAND

```

:ROUTINE TO ISSUE A WRITE CHARACTERISTICS
:COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED

:INPUT:
      R4      ADDRESS OF PACKET FROM TEST
      R5      FIRST DEVICE UNIBUS ADDRESS
      REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY

:OUTPUT:
      R0      TSSR CONTENTS
      CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
             CLR - WRITE CHARACTERISTICS FAILED

:IMPLICIT OUTPUT:
      MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
      SOFTWARE SWITCHES SET AS FOLLOWS:
             EXTFEA = EXTENDED FEATURES PRESENT
             BENBSW = BUFFER ENABLE SWITCH ON OR OFF

:SIDE EFFECTS:
      -
  
```

```

WRTCHR::
      SAVREG
      CLR      BENBSW          ;CLEAR BUFFER ENABLE SWITCH
      CLR      EXTFEA         ;CLEAR EXTENDED FEATURES SW SWITCH
10$:  MOV      R4,TSDB(R5)    ;SEND OUT COMMAND
      JSR      PC,CHKTSSR    ;WAIT FOR SSR
      BCS     20$            ;BR, IF SSR IS SET AND OK
      BR      60$            ;BR IF TROUBLE CARRY = CLEAR
20$:  MOV      TSSR(R5),R1   ;READ TSSR
      MOV      #SSR,R2       ;SET UP EXPECTED
      BIT     #OFL,R1        ;WAS OFF LINE SET IN TSSR
      BEQ     25$            ;BR, IF NO OFL SET
      BIS     #OFL,R2        ;MAKE THEM LOOK ALIKE
25$:  CMP      R2,R1         ;ARE THEY OK
      BEQ     40$            ;BR, IF EQUAL = OK
      BR      60$            ;TROUBLE EXIT
40$:  ADD      #8.,R4         ;POINT TO WRT CHARA DATA PACKET
      MOV      (R4),R3       ;GET ADDRESS OF MESSAGE BUFFER
      BIT     #X2.EXTF,XST2(R3) ;EXTENDED FEATURES BIT SET?
      BEQ     45$            ;BR IF NO
      INC     EXTFEA         ;SET EXTENDED FEATURES SW SWITCH
45$:  BIT     #X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
      BEQ     50$            ;BR, IF SWITCH NOT SET
      INC     BENBSW        ;SET SOFTWARE SWITCH FOR ENABLED
  
```

1153	011070			50\$:			
1154	011070	000261			SEC		:SET CARRY NO TROUBLE
1155	011072	000401			BR	70\$:EXIT
1156	011074	000241		60\$:	CLC		:CARRY CLEAR = ERROR
1157	011076	016500	000002	70\$:	MOV	TSSR(R5),R0	:RETURN TSSR CONTENTS
1158	011102	000207			RTS	PC	:RETURN
1159							
1160							

1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1206
1208
1209
1210
1211
1212

.SBTTL REWIND - POSITION TAPE (REWIND) COMMAND

```

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

```

011104			
011104			
011110	012704	011200	
011114	010465	000000	
011120	012703	000550	
011124	004737	016340	
011130	103417		
011132			
011132	012727	000372	
011136	000000		
011140	013727	002116	
011144	000000		
011146	005367	177772	
011152	001375		
011154	005367	177756	
011160	001367		
011162	005303		
011164	001357		
011166	000241		
011170	010400		
011172	000207		
011200			
011200	011200		
011200	102010		
011202	000000		

1213
1214
1215

1217
 1218
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1270

.SBTTL CKRAM - COMPARE RAM TO I/O PACKET

:+
 :ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
 :MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.

:INPUT:

R4 ADDRESS OF THE COMMAND PACKET
 R5 FIRST DEVICE UNIBUS ADDRESS

:OUTPUT:

CARRY SET - RAM MATCHES PACKET
 CLR - RAM DOES NOT MATCH PACKET

:IMPLICIT OUTPUT:

THE TABLE RAMDATA IS FILLED WITH THE
 DATA HELD IN RAM.
 RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE

:SIDE EFFECTS:

THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE

CKRAM::

```

SAVREG                                :SAVE THE GENERAL REGISTERS
MOV #RAMDATA,R1                       :ADDRESS TO SAVE THE RAM DATA
MOV #RMPKTBEQ,R2                      :BYTE ADDRESS OF FIRST RAM DATA
CLR R3                                 :CLEAR THE ERROR FLAG
JSR PC,CHKTSSR                         :WAIT FOR SSR
MOVB #0,TSDB(R5)                      :SET MAINTENANCE MODE
10$: JSR PC,CHKTSSR                    :WAIT FOR SSR TO SET
MOV R2,TSDB(R5)                       :SELECT NEXT RAM ADDRESS
JSR PC,CHKTSSR                         :WAIT FOR SSR TO SET
MOVB TSBA(R5),(R1)                   :READ THE RAM DATA
CMPB (R1)+,(R4)+                     :COMPARE TO EXPECTED
BEQ 20$                               :BRANCH IF OK
INC R3                                 :SET ERROR FLAG
20$: INC R2                           :ADDRESS OF NEXT RAM LOCATION
CMP R2,#RMPKTEND                     :REACHED END YET ?
BLE 10$                               :BRANCH TILL ALL READ
TST R3                                 :WAS AN ERROR FOUND ?
BEQ 30$                               :BRANCH IF NOT
CLC                                    :CLEAR CARRY TO SHOW ERROR
BR 50$                                :AND EXIT
30$: SEC                              :SHOW GOOD COMPARE
50$: MOV #8.,RAMSIZ                  :SETUP RAMSIZ FOR PRAMPKT ROUTINE
RTS PC                                :RETURN
  
```

```

011204
011204
011210 012701 002232
011214 012702 000201
011220 005003
011222 004737 016426
011226 112765 000000 000000
011234 004737 016426 10$:
011240 010265 000000
011244 004737 016426
011250 116511 000000
011254 122124
011256 001401
011260 005203
011262 005202 20$:
011264 020227 000210
011270 003761
011272 005703
011274 001402
011276 000241
011300 000401
011302 000261 30$:
011304 012737 000010 002272 50$:
011312 000207
  
```

1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299

```
.SBTTL CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA
:
:ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
:MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
:
:INPUT:
:
:      R4      ADDRESS OF THE CHARACTERISTICS DATA
:      R5      FIRST DEVICE UNIBUS ADDRESS
:
:OUTPUT:
:
:      CARRY   SET - RAM MATCHES PACKET
:             CLR - RAM DOES NOT MATCH PACKET
:
:IMPLICIT OUTPUT:
:
:      THE TABLE RAMDATA IS FILLED WITH THE
:      DATA HELD IN RAM.
:      RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
:
:SIDE EFFECTS:
:
:      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
:-
```

1300 011314
1301 011314
1302 011320 012701 002232
1303 011324 012702 000167
1304 011330 005003
1305 011332 004737 016426
1306 011336 112765 000000 000000
1307 011344 004737 016426 10\$:
1308 011350 010265 000000
1309 011354 004737 016426
1310 011360 116511 000000
1311 011364 122124
1312 011366 001401
1313 011370 005203
1314 011372 005202 20\$:
1315 011374 012737 000010 002272
1316 011402 005737 002216
1317 011406 001407
1318 011410 012737 000012 002272
1319 011416 020227 000200
1320 011422 003750
1321 011424 000403
1322 011426 020227 000176 25\$:
1323 011432 003744
1324 011434 005703 27\$:
1325 011436 001402
1326 011440 000241
1327 011442 000401
1328 011444 000261 30\$:

```
CKRAM2::
  SAVREG
  MOV #RAMDATA,R1 ;SAVE THE GENERAL REGISTERS
  MOV #RMCHBEG,R2 ;ADDRESS TO SAVE THE RAM DATA
  CLR R3 ;BYTE ADDRESS OF FIRST RAM DATA
  JSR PC,CHKTSSR ;CLEAR THE ERROR FLAG
  MOVB #0,TSDB(R5) ;WAIT FOR SSR
  JSR PC,CHKTSSR ;SET MAINTENANCE MODE
  MOV R2,TSDB(R5) ;WAIT FOR SSR TO SET
  JSR PC,CHKTSSR ;SELECT NEXT RAM ADDRESS
  MOVB TSBA(R5),(R1) ;WAIT FOR SSR TO SET
  CMPB (R1)+,(R4)+ ;READ THE RAM DATA
  BEQ 20$ ;COMPARE TO EXPECTED
  INC R3 ;BRANCH IF OK
  INC R2 ;SET ERROR FLAG
  MOV #8.,RAMSIZ ;ADDRESS OF NEXT RAM LOCATION
  TST EXTFEA ;ASSUME EXTFEA NOT SET
  BEQ 25$ ;IS THE SOFTWARE EXTENDED FEATURES SET
  MOV #10.,RAMSIZ ;BR, IF NOT SET
  CMP R2,#RMCHEND ;SET RAMSIZ FOR EXTEND FEATURES
  BLE 10$ ;AT END OF EXTENDED BUFFER
  BR 27$ ;BR, IF NOT AT END YET
  CMP R2,#RMCHEND-2 ;AT END BRANCH
  BLE 10$ ;REACHED END YET ?
  TST R3 ;BRANCH TILL ALL READ
  BEQ 30$ ;WAS AN ERROR FOUND ?
  CLC ;BRANCH IF NOT
  BR 50$ ;CLEAR CARRY TO SHOW ERROR
  SEC ;AND EXIT
  ;SHOW GOOD COMPARE
```

1329 011446 000207
1330

50\$: RTS PC

;RETURN

1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357 011450
1358 011450
1359 011454 010037 002274
1360 011460 010137 002276
1361 011464 005737 003124
1362 011470 001403
1363 011472 004737 017406
1364 011476 010001
1365 011500 005004
1366 011502 005003
1367 011504 010205
1368 011506 011264 002312
1369 011512 011164 002456
1370 011516 022221
1371 011520 001401
1372 011522 005203
1373 011524 062704 000002
1374 011530 020427 000014
1375 011534 003764
1376 011536 032765 000200 000012
1377 011544 001403
1378 011546 020427 000016
1379 011552 003755
1380 011554 005703
1381 011556 001402
1382 011560 000241
1383 011562 000401
1384 011564 000261
1385 011566 000207
1386

```

.SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS
:
:ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
:BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
:ERROR PRINT ROUTINES.
:
:INPUT:
:
:   R0      RCCV MESSAGE BUFFER HIGH ORDER ADDRESS
:   R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
:   R2      EXPD MESSAGE BUFFER ADDRESS
:
:OUTPUT:
:
:   CARRY   SET - MESSAGE BUFFERS MATCH
:   CLR    -MESSAGE BUFFERS DON'T MATCH
:
:IMPLICIT OUTPUT:
:
:   EXPMSG      BUFFER IS SET TO EXPD DATA
:   RECMSG      BUFFER IS SET TO RECV DATA
:   RCVHIADD    SET TO HIGH ORDER ADDRESS OF RECV
:   RCVLOADD    SET TO LOW ORDER ADDRESS OF RECV
:
:CKMSG::
:   SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
:   MOV         R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
:   MOV         R1,RCVLOAD  ;SAVE RECV LOW ADDRESS
:   TST        KTENABLE    ;TESTING ABOVE 28K?
:   BEQ        10$         ;BR IF NO
:   JSR        PC,SETMAP   ;RETURN ADDRESS BIASED TO PAR6 IN R0
:   MOV        R0,R1       ;GET RETURNED ADDRESS BIASED TO PAR6
:   CLR        R4          ;WORD IN BUFFER
:   CLR        R3          ;CLEAR ERROR SEEN FLAG
:   MOV        R2,R5       ;GET EXPD BUFFER ADDRESS
:   MOV        (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
:   MOV        (R1),RECMSG(R4) ;SAVE RECV FOR ERROR REPORT
:   CMP        (R2)+,(R1)+ ;EXPD EQUAL RECV?
:   BEQ        25$         ;BR IF YES
:   INC        R3          ;SET ERROR SEEN FLAG
:   ADD        #2,R4       ;POINT TO NEXT WORD ADDRESS
:   CMP        R4,#14      ;DONE FIRST 7 WORDS?
:   BLE        15$         ;BR IF NO
:   BIT        #X2.EXTF,XST2(R5) ;IS EXTENDED FEATURES SET IN EXPD?
:   BEQ        50$         ;BR IF NO
:   CMP        R4,#16      ;DONE EXTENDED FEATURES WORD?
:   BLE        15$         ;BR IF NO
:   TST        R3          ;ANY ERRORS SEEN?
:   BEQ        55$         ;BR IF NO
:   CLC        ;SET FAILURE
:   BR         60$         ;
:   SEC        ;SET SUCCESS
:   RTS        PC         ;RETURN
:
:10$:
:   CLR        R4
:   CLR        R3
:   MOV        R2,R5
:   MOV        (R2),EXPMSG(R4)
:   MOV        (R1),RECMSG(R4)
:   CMP        (R2)+,(R1)+
:   BEQ        25$
:   INC        R3
:   ADD        #2,R4
:   CMP        R4,#14
:   BLE        15$
:   BIT        #X2.EXTF,XST2(R5)
:   BEQ        50$
:   CMP        R4,#16
:   BLE        15$
:   TST        R3
:   BEQ        55$
:   CLC
:   BR         60$
:
:25$:
:   CLR        R4
:   CLR        R3
:   MOV        R2,R5
:   MOV        (R2),EXPMSG(R4)
:   MOV        (R1),RECMSG(R4)
:   CMP        (R2)+,(R1)+
:   BEQ        25$
:   INC        R3
:   ADD        #2,R4
:   CMP        R4,#14
:   BLE        15$
:   BIT        #X2.EXTF,XST2(R5)
:   BEQ        50$
:   CMP        R4,#16
:   BLE        15$
:   TST        R3
:   BEQ        55$
:   CLC
:   BR         60$
:
:50$:
:   CLR        R4
:   CLR        R3
:   MOV        R2,R5
:   MOV        (R2),EXPMSG(R4)
:   MOV        (R1),RECMSG(R4)
:   CMP        (R2)+,(R1)+
:   BEQ        25$
:   INC        R3
:   ADD        #2,R4
:   CMP        R4,#14
:   BLE        15$
:   BIT        #X2.EXTF,XST2(R5)
:   BEQ        50$
:   CMP        R4,#16
:   BLE        15$
:   TST        R3
:   BEQ        55$
:   CLC
:   BR         60$
:
:55$:
:   SEC
:
:60$:
:   RTS        PC

```

1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415 011570
1416 011570
1417 011574 020327 000144
1418 011600 003412
1419 011602 012703 000144
1420 011606
1421 011626 010037 002274
1422 011632 010137 002276
1423 011636 005737 003124
1424 011642 001403
1425 011644 004737 017406
1426 011650 010001
1427 011652 005004
1428 011654 005005
1429 011656 111264 002312
1430 011662 111164 002456
1431 011666 122221
1432 011670 001401
1433 011672 005205
1434 011674 062704 000001
1435 011700 020403
1436 011702 002001
1437 011704 000764
1438 011706 005705
1439 011710 001402

```

.SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
:ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
:BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
:ERROR PRINT ROUTINES.
:INPUT:
      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
      R2      EXPD MESSAGE BUFFER ADDRESS
      R3      NUMBER OF BYTES TO COMPARE
:OUTPUT:
      CARRY   SET - MESSAGE BUFFERS MATCH
             CLR - MESSAGE BUFFERS DON'T MATCH
:IMPLICIT OUTPUT:
      EXPMSG  BUFFER IS SET TO EXPD DATA
      RECMSG  BUFFER IS SET TO RECV DATA
      RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
      RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
-
CKMSG2::
      SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
      CMP R3,#RECMSG-EXPMSG;@@@ IS COUNT ABOVE MAX ALLOWED?
      BLE 5$          ;@@@ BR IF NO
      MOV #RECMSG-EXPMSG,R3;@@@
      PRINTF #DEBUGMSG ;@@@
      MOV #DEBUGMSG,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
      TRAP C$PNTF
      ADD #4,SP
5$:  MOV R0,RCVHIADD  ;SAVE RECV HIGH ADDRESS
      MOV R1,RCVLOAD  ;SAVE RECV LOW ADDRESS
      TST K$ENABLE    ;TESTING ABOVE 28K?
      BEQ 10$         ;BR IF NO
      JSR PC,SETMAP   ;RETURN ADDRESS BIASED TO PAR6 IN R0
      MOV R0,R1       ;GET RETURNED ADDRESS BIASED TO PAR6
10$: CLR R4           ;WORD IN BUFFER
      CLR R5          ;CLEAR ERROR SEEN FLAG
15$: MOVB (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
      MOVB (R1),RECMSG(R4) ;SAVE RECV FOR ERROR REPORT
      CMPB (R2)+,(R1)+  ;EXPD EQUAL RECV?
      BEQ 25$         ;BR IF YES
      INC R5          ;SET ERROR SEEN FLAG
25$: ADD #1,R4        ;POINT TO NEXT BYTE
      CMP R4,R3       ;DONE ALL BYTES?
      BGE 50$         ;BR IF YES
      BR 15$          ;DO NEXT BYTE
50$: TST R5           ;ANY ERRORS SEEN?
      BEQ 55$         ;BR IF NO

```

```
1440 011712 000241
1441 011714 000401
1442 011716 000261
1443 011720 000207
1444
1445 011722 120 122 117 DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';@@D
1446 012012 045 116 045 FERCM: .ASCII /ZNA ***/
1447 012023 040 040 124 ERCM: .ASCIZ / TSSR ERROR CODE REC'D = /
1448 012056 056 056 056 SIMSG: .ASCIZ /... AFTER DOING SOFT INIT/
1449 012111 124 105 123 TINERR: .ASCIZ /TEST: .../
1450 .EVEN
```

CLC :SET FAILURE
BR 60\$
55\$: SEC :SET SUCCESS
60\$: RTS PC :RETURN

1452
 1453
 1454
 1455
 1456
 1457
 1458
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467

```

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

1468 012124
 012124
 1469 012124 004737 006022
 1470 012130 004737 017272
 1471 012134
 012134
 012134 104423

```

BGNMSG SFMSG
SFMSG:: JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR REGISTER
        JSR PC,CKDROP ;DROP UNIT, IF ALLOWED
        ENDMSG
L10003: TRAP C$MSG
  
```

1472
 1473
 1474
 1475
 1476
 1477
 1478
 1479
 1480
 1481
 1482
 1483

```

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

1484 012136
 012136
 1485 012136 004737 006022
 1486 012142 012700 000004
 1487 012146 004737 007460
 1488 012152
 012152
 012152 104423

```

BGNMSG PKTSSR
PKTSSR:: JSR PC,PRITSSR ;PRINT THE CONTENTS OF TSSR REGISTER
         MOV #4,R0 ;NO. OF WORDS IN PACKET
         JSR PC,PRIPKT ;PRINT THE CONTENTS OF COMMAND PACKET
         ENDMSG
L10004: TRAP C$MSG
  
```

1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1499
 1500

```

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

1501 012154
 012154

```

BGNMSG PKTGETS
PKTGETS::
  
```

1502 012154 004737 006022
 1503 012160 012700 000002
 1504 012164 004737 007460
 1505 012170
 012170
 012170 104423

JSR PC,PRITSSR ;PRINT THE CONTENTS OF TSSR REGISTER
 MOV #2,R0 ;NO. OF WORDS IN GET STATUS PACKET
 JSR PC,PRIPKT ;PRINT THE CONTENTS OF COMMAND PACKET
 ENDMSG
 L10005: TRAP C\$MSG

1506
 1507
 1508
 1509
 1510
 1511
 1512
 1513
 1514
 1515
 1516

:+
 :PRINT TSSR ERRORS FOR INITIALIZATION TESTS

:INPUTS:

: R1 TSSR CONTENTS
 : R4 ADDRESS OF COMMAND PACKET
 :-

1517 012172
 012172
 1518 012172 004737 006022
 1519 012176
 012176
 012176 104423

BGNMSG SFFMSG
 SFFMSG:: JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR REGISTER
 ENDMSG
 L10006: TRAP C\$MSG

1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527

.SBTTL PKTMES - PRINT TSSR AND MESSAGE BUFFER

:+
 :PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
 :BUFFER FOR ERROR REPORTS

:INPUTS:

: R1 CONTENTS OF TSSR
 : R2 LOW ORDER MESSAGE BUFFER
 : R3 HIGH ORDER MESSAGE BUFFER ADDRESS
 : NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
 :-

1535 012200
 012200
 1536 012200 004737 006022
 1537 012204 010200
 1538 012206 010301
 1539 012210 004737 014332
 1540 012214
 012214
 012214 104423
 1541

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

1543
 1544
 1545
 1546
 1547
 1548
 1549
 1550
 1551
 1552
 1553
 1554
 1555 012216
 012216
 1556 012216 004737 010364
 1557 012222 016501 000002
 1558 012226 004737 006022
 1559 012232
 012232
 012232 104423
 1560
 1561
 1562
 1563
 1564
 1565
 1566
 1567
 1568
 1569
 1570
 1571
 1572
 1573
 1574 012234
 012234
 1575 012234 012700 000007
 1576 012240 005737 002216
 1577 012244 001402
 1578 012246 012700 000010
 1579 012252 004737 014642
 1580 012256
 012256
 012256 104423
 1581
 1582

```

.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
:      RECMSG  - 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     S$,           ;BR IF NO
      MOV     #8.,R0        ;EXT FEATURE BUFFER IS 8 WORDS
      JSR     PC,PRMSGEXP   ;PRINT EXPD/RCV MESSAGE BUFFERS
      ENDMSG
S$:
L10011:
      TRAP   C$MSG
  
```

```

1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596 012260
      012260
1597 012260 010146 012332
      012260 012746 000002
      012266 012746
      012272 010600
      012274 104415
      012276 062706 000006
1598 012302
      012302 012746 012401
      012306 012746 000001
      012312 010600
      012314 104415
      012316 062706 000004
1599 012322 010100
1600 012324 004737 015212
1601 012330
      012330
      012330 104423
1602 012332 045 116
1603 012401 045 116
1604
1605
  
```

```

.SBTTL FIFEXP - PRINT FIFO EXP/REC DATA
:
:PRINT ROUTINE TO PRINT FIFO EXP/REC DATA
      R1 - BYTE COUNT
:IMPLICIT INPUTS:
      EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
      RECMMSG - 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,R0
TRAP C$PNTX
ADD #6,SP
PRINTX #FIF2MSG ;PRINT HEADER MSG
MOV #FIF2MSG,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #4,SP
MOV R1,R0 ;GET BYTE COUNT
JSR PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
ENDMSG
L10012:
TRAP C$MSG
045 FIF1MSG: .ASCIZ '%N% NUMBER OF BYTES TRANSFERRED = %D2'
045 FIF2MSG: .ASCIZ '%N% FIFO DATA BYTES IN ERROR:'
.EVEN
  
```

```

1607                                     .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
1608                                     :+
1609                                     :PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1610                                     :
1611                                     :IMPLICIT INPUTS:
1612                                     :
1613                                     :EXPMSG - EXPECTED MESSAGE BUFFER
1614                                     :RECMSG - RECEIVED MESSAGE BUFFER
1615                                     :RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1616                                     :RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1617                                     :
1618                                     :-
1619                                     :
1620 012440 BGNMSG MSGSTAT
1621 012440 MSGSTAT::
1622 012440 012701 012502 MOV #STATCOD,R1 ;ASCII ADDRESS TABLE
1623 012444 012100 10$: MOV (R1)+,R0 ;DONE ALL MSG LINES?
1624 012446 001410 BEQ 20$ ;BR IF YES
1625 012450 PRINTX R0 ;PRINT STATUS BIT NAMES
1626 012450 010046 MOV R0,-(SP)
1627 012452 012746 000001 MOV #1,-(SP)
1628 012456 010600 MOV SP,R0
1629 012460 104415 TRAP C$PNTX
1630 012462 062706 000004 ADD #4,SP
1631 012466 000766 BR 10$ ;DO ANOTHER MSG LINE
1632 012470 012700 000012 20$: MOV #10.,R0 ;NUMBER OF WORDS IN A READ STATUS BUFFER
1633 012474 004737 014642 JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
1634 012500 ENDMSG
1635 012500 L10013:
1636 012500 104423 TRAP C$MSG
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653 013154 BGNMSG MSGLOOP
1654 013154 MSGLOOP::
1655 013154 012701 013216 MOV #LOOPCOD,R1 ;ASCII ADDRESS TABLE
  
```

```

1655 013160 012100          10$:  MOV      (R1)+,R0      ;DONE ALL MSG LINES?
1656 013162 001410          BEQ      20$              ;BR IF YES
1657 013164          PRINTX  R0              ;PRINT STATUS BIT NAMES
      013164 010046          MOV      R0,-(SP)
      013166 012746 000001  MOV      #1,-(SP)
      013172 010600          MOV      SP,R0
      013174 104415          TRAP    C$PNTX
      013176 062706 000004  ADD      #4,SP
1658 013202 000766          BR       10$              ;DO ANOTHER MSG LINE
1659 013204 012700 000012  20$:  MOV      #10.,R0      ;NUMBER OF WORDS IN A READ STATUS BUFFER
1660 013210 004737 014642  JSR      PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
1661 013214          ENDMSG
      013214          L10014:
      013214 104423          TRAP    C$MSG
1662
1663 013216 013236 013311 013410 LOOPCOD: .WORD 1$,2$,3$,4$,5$,6$,7$,0
1664 013236          045 116 045 1$: .ASCIZ '%N% Tape Bus Loopback Signals in Word #8:'
1665 013311          045 116 045 2$: .ASCIZ '%N% PARERR<15> IRESV2<14> IRESV1<13>'
1666 013410          045 116 045 3$: .ASCIZ '%N% IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
1667 013507          045 116 045 4$: .ASCIZ '%N% IWFM =>IFMK<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
1668 013606          045 116 045 5$: .ASCIZ '%N% ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDPA <04>'
1669 013705          045 116 045 6$: .ASCIZ '%N% IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
1670 014004          045 116 045 7$: .ASCIZ '%N% IGO =>IFPT<00>'
1671          .EVEN
1672

```

```

1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687 014032
      014032
1688 014032 012700 000012
1689 014036 004737 014642
1690 014042
      014042
      014042 104423
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708 014044
      014044
1709 014044 004737 010250
1710 014050 013701 002222
1711 014054 013702 002224
1712 014060 004737 010032
1713 014064
      014064
      014064 104423
1714
  
```

```

.SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
:+
:PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
: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 MSGSUB
MSGSUB:
      MOV     #10,R0           ;SIZE OF WRITE SUBSYSTEM BUFFER
      JSR    PC,PRMSGEXP     ;PRINT EXPD/RCV MESSAGE BUFFERS
      ENDMSG
L10015:
      TRAP   CSMSG
  
```

```

.SBTTL MEMADD - PRINT MEMORY ADDRESS DATA ERROR
:+
:PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
:IMPLICIT INPUTS:
:
:   ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
:   ERRLO - MEMORY ERROR LOW ORDER ADDRESS
:   EXP   - EXPECTED DATA
:   RECV  - RECEIVED DATA
:-
      BGNMSG MEMADD
MEMADD:
      JSR    PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
      MOV   EXPD,R1         ;GET EXPD DATA
      MOV   RECV,R2        ;GET RECEIVED DATA
      JSR   PC,PRIXOR       ;PRINT EXPD/RCV
      ENDMSG
L10016:
      TRAP   CSMSG
  
```

```

1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737 014066
1738 014066
1739 014072 012701 002232
1740 014076 005002
1741 014100 122124
1742 014102 001005
1743 014104
1744 014114 000436
1745 014116 116105 177777
1746 014122 116403 177777
1747 014126
1748 014136 042703 177400
1749 014142 116137 177777 002224
1750 014150 116437 177777 002222
1751 014156
      014156 010346
      014160 013746 002222
      014164 013746 002224
      014170 010246
      014172 012746 014246
      014176 012746 000005
      014202 010600
      014204 104414
      014206 062706 000014
1752 014212 005202
1753 014214 005737 002272
1754 014220 001404
1755 014222 020237 002272
1756 014226 003724
1757 014230 000403
1758 014232 020227 000010
1759 014236 002720
1760 014240 005037 002272
1761 014244 000207
1762
1763 014246 045 116 045 RAMASC: .ASCIZ '%N%A BYTE: %D2%A RAM: %O3%A Packet: %O3%A XOR:%O3'
```

```

.SBTTL PRAMPKT - PRINT RAM AND PACKET DATA
:
:PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
:WHEN THE RAM DATA DOES NOT MATCH.
:
: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.
:
:IMPLICIT OUTPUTS:
:
:      RAMSIZ  SET TO 0
:
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
```


1764
1765
1766

.EVEN

1768
 1769
 1770
 1771
 1772
 1773
 1774
 1775
 1776
 1777
 1778
 1779
 1780
 1781
 1782
 1783
 1784
 1785 014332
 1786 014332
 1787 014336 010005
 1788 014340 005737 003124
 1789 014344 001001
 1790 014346 005001
 1791 014350 010103
 1792 014352 006100
 1793 014354 006101
 1794 014356
 014356 010546
 014360 010146
 014362 012746 014510
 014366 012746 000003
 014372 010600
 014374 104415
 014376 062706 000010
 1795 014402
 014402 012746 014555
 014406 012746 000001
 014412 010600
 014414 104415
 014416 062706 000004
 1796 014422 005004
 1797 014424 010501
 1798 014426 010300
 1799 014430 001403
 1800 014432 004737 017406
 1801 014436 010005
 1802 014440
 014440 012546
 014442 010446
 014444 012746 014613
 014450 012746 000003
 014454 010600
 014456 104415
 014460 062706 000010
 1803 014464 005204
 1804 014466 020427 000007
 1805 014472 003005

.SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

:
: 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:
      SAVREG      ;SAVE THE REGISTERS
      MOV R0,R5   ;SAVE LOW ORDER ADDRESS
      TST KTENABLE ;ADDRESS ABOVE 28K?
      BNE 10$     ;BR IF YES
      CLR R1      ;SET HIGH ORDER ADDRESS TO 0
      MOV R1,R3   ;SAVE HIGH ORDER ADDRESS
      ROL R0      ;SHIFT BIT15 TO C BIT
      ROL R1      ;SHIFT TO HIGH ORDER FOR PRINTOUT
      PRINTX #PROASC,R1,R5 ;PRINT MESSAGE BUFFER ADDRESS
      MOV R5,-(SP)
      MOV R1,-(SP)
      MOV #PROASC,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C$PNTX
      ADD #10,SP
      PRINTX #PRIASC ;PRINT HEADER FOR CONTENTS
      MOV #PRIASC,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
      TRAP C$PNTX
      ADD #4,SP
      CLR R4      ;NUMBER OF THE NEXT WORD
      MOV R5,R1   ;COPY LOW ORDER ADDRESS
      MOV R3,R0   ;COPY HIGH ORDER ADDRESS
      BEQ 20$     ;BR IF NOT ABOVE 28K
      JSR PC,SETMAP ;SETUP PAR ADDRESS IN R0
      MOV R0,R5   ;GET PAR FORMAT ADDRESS ABOVE 28K
      PRINTX #PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
      MOV (R5)+,-(SP)
      MOV R4,-(SP)
      MOV #PRASC,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C$PNTX
      ADD #10,SP
      INC R4      ;NUMBER OF THE NEXT
      CMP R4,#7   ;DONE ALL YET ?
      BGT 50$     ;BRANCH IF ALL DONE
    
```

1806	014474	002761				BLT	20\$:PRINT FIRST 7 WORDS
1807	014476	032763	000200	000012		BIT	#X2.EXTF,XST2(R3)	:EXTENDED FEATUTES ON ?
1808	014504	001355				BNE	20\$:PRINT EXTENDED STATUS WORD
1809	014506	000207			50\$:	RTS	PC	:RETURN
1810								
1811	014510	045	116	045	PROASC:	.ASCIZ	'%N% Message Buffer Address = %01%05'	
1812	014555	045	116	045	PR1ASC:	.ASCIZ	'%N% Message Buffer Contents:'	
1813	014613	045	116	045	PRASC:	.ASCIZ	'%N% Word%D1%: %0'	
1814						.EVEN		

```

1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830 014642
1831 014642
1832 014646 010005
1833 014650 013700 002276
1834 014654 010004
1835 014656 013701 002274
1836 014662 006100
1837 014664 006101
1838 014666
    014666 010446
    014670 010146
    014672 012746 015022
    014676 012746 000003
    014702 010600
    014704 104415
    014706 062706 000010
1839 014712
    014712 012746 015067
    014716 012746 000001
    014722 010600
    014724 104415
    014726 062706 000004
1840 014732 005004
1841 014734 012701 002312
1842 014740 012702 002456
1843 014744 011100
1844 014746 011203
1845 014750
1846 014760
    014760 010346
    014762 012246
    014764 012146
    014766 010446
    014770 012746 015125
    014774 012746 000005
    015000 010600
    015002 104415
    015004 062706 000014
1847 015010 005204
1848 015012 020405
1849 015014 002001
1850 015016 000752
1851 015020 000207
    
```

```

.SBTTL PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
+
:ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
:
:RO - NUMBER OF WORDS IN BUFFER
:
:IMPLICIT INPUTS:
:
:EXPMSG - EXPECTED MESSAGE BUFFER
:RECMSG - RECEIVED MESSAGE BUFFER
:RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
:RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
:
PRMSGEXP::
    SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
    MOV RO,R5 ;SAVE NUMBER OF WORDS
    MOV RCVLOADD,RO ;GET RECV LOW ADDRESS
    MOV RO,R4 ;COPY LOW ADDRESS
    MOV RCVHIADD,R1 ;GET RECV HIGH ADDRESS
    ROL RO ;SHIFT BIT15 TO C BIT
    ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
    PRINTX #PRMSG0,R1,R4 ;PRINT MESSAGE BUFFER ADDRESS
    MOV R4,-(SP)
    MOV R1,-(SP)
    MOV #PRMSG0,-(SP)
    MOV #3,-(SP)
    MOV SP,RO
    TRAP C$PNTX
    ADD #10,SP
    PRINTX #PRMSG1 ;PRINT HEADER FOR CONTENTS
    MOV #PRMSG1,-(SP)
    MOV #1,-(SP)
    MOV SP,RO
    TRAP C$PNTX
    ADD #4,SP
    CLR R4 ;NUMBER OF THE CURRENT WORD
    MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
    MOV #RECMSG,R2 ;GET RECV BUFFER ADDRESS
    20$: MOV (R1),RO ;GET EXPD
    MOV (R2),R3 ;GET RECV
    XOR RO,R3 ;XOR EXPD/RECV
    PRINTX #PRMSG2,R4,(R1)+,(R2)+,R3
    MOV R3,-(SP)
    MOV (R2)+,-(SP)
    MOV (R1)+,-(SP)
    MOV R4,-(SP)
    MOV #PRMSG2,-(SP)
    MOV #5,-(SP)
    MOV SP,RO
    TRAP C$PNTX
    ADD #14,SP
    INC R4 ;NUMBER OF THE NEXT
    CMP R4,R5 ;DONE ALL YET?
    BGE 50$ ;BR IF YES
    BR 20$ ;DO ANOTHER
    50$: RTS PC ;RETURN
    
```

1852
1853 015022 045 116 045 PRMSG0: .ASCIZ '%N% Message Buffer Address = %01%05'
1854 015067 045 116 045 PRMSG1: .ASCIZ '%N% Message Buffer Contents:'
1855 015125 045 116 045 PRMSG2: .ASCIZ '%N% WORD #D2% EXPD: %06% RECV: %06% XOR: %06%'
1856 .EVEN
1857

```

1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872 015212
1873 015212
1874 015216 010005
1875 015220 005037 002310
1876 015224 005004
1877 015226 012701 002312
1878 015232 012702 002456
1879 015236 111100
1880 015240 042700 177400
1881 015244 110037 015560
1882 015250 111203
1883 015252 042703 177400
1884 015256 110337 015562
1885 015262
1886 015272 12212?
1887 015274 001431
1888 015276 005237 002310
1889 015302 023727 002310 000010
1890 015310 101023
1891 015312
    015312 010346
    015314 013746 015562
    015320 013746 015560
    015324 010446
    015326 012746 015426
    015332 012746 000005
    015336 010600
    015340 104415
    015342 062706 000014
1892 015346
1893 015356 000404
1894 015360
1895 015360
1896 015370
1897 015370 005204
1898 015372 020405
1899 015374 002001
1900 015376 000717
1901 015400
    015400 013746 002310
    015404 012746 015513
    015410 012746 000002
    015414 010600
    015416 104415
    
```

```

.SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
+
:ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
:ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
:
:RO - NUMBER OF BYTES IN BUFFER
:IMPLICIT INPUTS:
:EXPMSG - EXPECTED MESSAGE BUFFER
:RECMSG - RECEIVED MESSAGE BUFFER
-
PRBYTEXP:
  SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
  MOV R0,R5 ;SAVE NUMBER OF BYTES
  CLR PRMNO ;INIT ERROR COUNT
  CLR R4 ;NUMBER OF THE CURRENT BYTE
  MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
  MOV #RECMSG,R2 ;GET RECV BUFFER ADDRESS
20$: MOV (R1),R0 ;GET EXPD BYTE
  BIC #^C<377>,R0 ;CLEAR UPPER BYTE
  MOV R0,PRBEXP ;SAVE FOR ERROR REPORT
  MOV (R2),R3 ;GET RECV BYTE
  BIC #^C<377>,R3 ;CLEAR UPPER BYTE
  MOV R3,PRBREC ;FOR ERROR REPORT
  XOR R0,R3 ;XOR EXPD/RECV
  CMPB (R1)+,(R2)+ ;EXPD = RECV?
  BEQ 30$ ;BR IF YES
  INC PRMNO ;UPDATE ERROR COUNT
  CMP PRMNO,#8. ;PRINTED 8?
  BHI 30$ ;BR IF YES
27$: PRINTX #PRBMSG,R4,PRBEXP,PRBREC,R3
  MOV R3,-(SP)
  MOV PRBREC,-(SP)
  MOV PRBEXP,-(SP)
  MCL R4,-(SP)
  MOV #PRBMSG,-(SP)
  MOV #5,-(SP)
  MOV SP,R0
  TRAP C$PNTX
  ADD #14,SP
  FORCEEXIT 50$ ;@@D
  BR 35$ ;@@D
30$:
35$: FORCERROR 27$,NOTSSR ;@@D
: @@@
  INC R4 ;NUMBER OF THE NEXT
  CMP R4,R5 ;DONE ALL YET?
  BGE 50$ ;BR IF YES
  BR 20$ ;DO ANOTHER
50$: PRINTX #PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
  MOV PRMNO,-(SP)
  MOV #PRBTOT,-(SP)
  MOV #2,-(SP)
  MOV SP,R0
  TRAP C$PNTX
    
```

1902	015420	062706	000006	ADD	#6,SP	
	015424	000207		RTS	PC	;RETURN
1903						
1904	015426	045	116	045	PRBMSG: .ASCIZ	'%N% BYTE #D2% EXPD: %03% RECV: %03% XOR: %03'
1905	015513	045	116	045	PRBTOT: .ASCIZ	'%N% NUMBER OF BYTES IN ERROR = D2'
1906					.EVEN	
1907	015560	000000			PRBEXP: .WORD	0 ;EXPD
1908	015562	000000			PRBREC: .WORD	0 ;RCV
1909						

1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923 015564
015564
1924 015564 004737 010032
1925 015570
015570
015570 104423
1926
1927

```
.SBTTL EXPREC - PRINT EXPD/RECV WORD DATA  
: +  
: PRINT ROUTINE TO DISPLAY EXPD/RECV DATA  
: INPUTS:  
: R1 RECEIVED DATA  
: R2 EXPECTED DATA  
: -  
: BGNMSG EXPREC  
EXPREC:: JSR PC,PRIXOR ;PRINT THE DATA  
: ENDMSG  
L10017: TRAP CSMSG
```


1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942 015572
015572
1943 015572 004737 007702
1944 015576
015576 104423
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969 015600
015600
1970 015600 004737 014066
1971 015604
015604 104423
1972
1973
1974
1975
1976
1977
1978
1979

```
.SBTTL EXPBREC - PRINT EXPD/REC V BYTE DATA
:+
:PRINT ROUTINE TO DISPLAY BYTE EXPD/REC V DATA
:INPUTS:
:      R1      RECEIVED DATA BYTE
:      R2      EXPECTED DATA BYTE
:-

EXPBREC: BGNMSG EXPBREC
:JSR PC,PRIBXOR          ;PRINT THE DATA
:ENDMSG
L10020: TRAP C$MSG

.SBTTL RAMERR - PRINT 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.
:IMPLICIT OUTPUTS:
:      RAMSIZ SET TO 0
:-

RAMERR: BGNMSG RAMERR
:JSR PC,PRAMPKT          ;PRINT RAM/PACKET DATA
:ENDMSG
L10021: TRAP C$MSG

.SBTTL RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
:+
:PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
:INPUTS:
```

```

1980
1981          R4      POINTER TO COMMAND PACKET
1982
1983      :IMPLICIT INPUTS:
1984
1985          RAMDATA  DATA AS READ FROM THE RAM
1986          RAMSIZ   NUMBER OF BYTES IN PACKET
1987                  IF RAMSIZ=0 THEN DEFAULT TO 8.
1988          ERRHI   HIGH ORDER TEST ADDRESS
1989          ERRLO   LOW ORDER TEST ADDRESS
1990
1991      :IMPLICIT OUTPUTS:
1992
1993          RAMSIZ   SET TO 0
1994      :-
1995
1996 015606      BGNMSG  RAMTADD
1997 015606      RAMTADD: JSR      PC,PRITADD      ;PRINT TEST ADDRESS
1998 015612 004737 010354 JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
1999 015616      ENDMSG
015616
015616 104423
2000
2001
2002          .SBTTL  RAMEXP - PRINT RAM EXPD/RECV DATA
2003      :+
2004      :PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2005      :INPUTS:
2006
2007          R1      RECEIVED DATA
2008          R2      EXPECTED DATA
2009          R4      CONTROLLER RAM ADDRESS
2010
2011
2012      :-
2013
2014 015620      BGNMSG  RAMEXP
2015 015620      RAMEXP: BIC      #^C<377>,R1      ;SAVE EXPD RAM DATA BYTE
2016 015624 042701 177400 BIC      #^C<377>,R2      ;SAVE EXPD RAM DATA BYTE
2017 015630 004737 010156 JSR      PC,PRIRAM      ;PRINT THE RAM ADDRESS
2018 015634 004737 010032 JSR      PC,PRIXOR      ;PRINT THE DATA
2019 015640      ENDMSG
015640
015640 104423
2020
2021          .SBTTL  TIMEXP - PRINT TIMER A,B AND EXP/REC
2022      :+
2023      :PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2024      :AND TIMER A,B HEADER MESSAGE
2025      :INPUTS:
2026
2027          R1      RECEIVED DATA
2028          R2      EXPECTED DATA
2029
2030
    
```

```

2031      :-
2032
2033 015642      BGNMSG  TIMEXP
      015642      TIMEXP::
2034 015642      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$PNTX
      015656 062706 000004      ADD      #4,SP
2035 015662 004737 010032      JSR     PC,PRIXOR          ;PRINT THE DATA
2036 015666
      015666      L10024:
      015666 104423      TRAP    C$MSG
2037
2038
2039 015670      045      116      045  TIMSGO: .ASCIZ  '%N% TIMER A STATUS IS IN BIT 3%N% TIMER B STATUS IS IN BIT 2'
2040      .EVEN
    
```

```

2042 .SBTTL BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
2043
2044
2045
2046 :PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
2047
2048 :INPUTS:
2049
2050 : R1 CONTENTS OF TSSR
2051 : R2 DATA WRITTEN (8 BITS)
2052
2053 :-
2054
2055 015770 BGNMSG BADSSR
      015770
2056 015770 010246 BADSSR::
2057 015772 042702 177400      MOV R2, -(SP) ;SAVE DATA TRANSFERRED
2058 015776 010246      BIC #177400, R2 ;GET JUST ONE BYTE
      015776 010246      PRINTB #XFERASC, R2
      016000 012746 016030      MOV R2, -(SP)
      016004 012746 000002      MOV #XFERASC, -(SP)
      016010 010600      MOV #2, -(SP)
      016012 104414      MOV SP, R0
      016014 062706 000006      TRAP C$PNTB
2059 016020 012602      ADD #6, SP
2060 016022 004737 006022      MOV (SP)+, R2 ;RESI0RE R2
2061 016026      JSR PC, PRITSSR ;DECODE TSSR CONTENTS
      016026      ENDMSG
2062 016030 104423      L10025:
      016030 045 116 045 XFERASC: TRAP C$MSG
2063      .ASCIZ '%N% Data Transferred = %03'
```

2065
2066
2067
2068
2069
2070
2071

.SBTTL GLOBAL SUBROUTINES SECTION

:++
: THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
: THAT ARE USED IN MORE THAN ONE TEST.
:--

2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114

016064
016064
016070 012765 000000 000002
016076 004737 016340
016102 016500 000002
016106 010004
016110 042704 176277
016114 052704 002200
016120 020400
016122 001402
016124 000241
016126 000401
016130 000261
016132 000207

```

.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:
:
:       R0      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
:       ERRDF                    ;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
:       BIC     #^C<HIADDR!OFL>,R4
:       BIS     #SSR!NBA,R4      ; R4 HAS EXPECTED CONTENTS
:       CMP     R4,R0            ; ONLY EXPECTED BITS SET ?
:       BEQ     $$                ; BRANCH IF OKAY
:       CLC                     ; CLEAR THE CARRY FOR ERROR
:       BR     10$               ; GO TO EXIT
:       SEC                     ; SET THE CARRY BIT
:       RTS     PC               ; RETURN TO CALLER
:
5$:
10$:

```

2116
 2117
 2118
 2119
 2120
 2121
 2122
 2123
 2124
 2125
 2126
 2127
 2128
 2129
 2130
 2131
 2132
 2133
 2134
 2135
 2136 016134
 2137 016134
 2138 016140 010004
 2139 016142 032700 100000
 2140 016146 001004
 2141 016150 032700 174077
 2142 016154 001023
 2143 016156 000424
 2144 016160 032700 000200
 2145 016164 001011
 2146 016166 032700 000040
 2147 016172 001414
 2148 016174 042704 177761
 2149 016200 020427 000016
 2150 016204 001007
 2151 016206 000410
 2152 016210 032700 000040
 2153 016214 001405
 2154 016216 032700 000006
 2155 016222 001002
 2156 016224 000241
 2157 016226 000401
 2158 016230 000261
 2159 016232 000207
 2160

.SBTTL CHKAMB - CHECK TSSR FOR AMBIGUITY

```

: +
: 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     #SC,R0        ;IS BIT 15 SET ?
        BNE     5$            ;BRANCH IF YES
        BIT     #^C<NBA!OFL!SSR!HIADDR>,R0 ;ANY OTHER BITS SET ?
        BNE     40$          ;MUST BE AN ERROR
        BR      45$          ;RETURN WITH SUCCESS
5$:     BIT     #SSR,R0        ;IS READY BIT SET ?
        BNE     10$         ;BRANCH IF READY BIT IS SET.
        BIT     #BITS,R0     ;IS FATAL ERROR BIT SET ?
        BEQ     40$         ;ERROR IF NOT
        BIC     #^CTERCLS,R4 ;CLEAR ALL BUT TERMINATION CODE
        CMP     R4,#16       ;ALL THREE BITS MUST BE SET
        BNE     40$         ;ERROR IF NOT SET
        BR      45$         ;OK IF ALL ARE SET
10$:    BIT     #BITS,R0     ;IS FATAL ERROR BIT SET ?
        BEQ     45$         ;ERROR IF BIT IS SET WITH SSR
        BIT     #BIT2!BIT1,R0 ;IS THIS A FUNCTION REJECT
        BNE     45$         ;BR, IF TSSR IS OK
40$:    CLC                ;AMBIGUOUS CONTENTS
        BR      50$
55$:    SEC                ;SHOW SUCCESS - NO AMBIGUITY
50$:    RTS      PC         ;RETURN TO CALLER

```

```

2162                .SBTTL ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
2163                :
2164                : DEFAULT DISPLAY INTERRUPT HANDLERS.
2165                : IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2166                : OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
2167                :
2168                :
2169                : BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2170                :
2171                :       IOKCKIN=BIT7      ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
2172                :       IOKSTP=BIT0      ; EXPECT "STOP" INTERRUPT.
2173                :
2174                : INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
2175 016234          INTMASK: .BYTE 0
2176                : INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2177 016235          INTFLAG: .BYTE 0
2178                :
2179                : SAVED INTERRUPT VECTOR:
2180 016236          INTVEC: .WORD 0
2181                : SAVE CPU PC
2182 016240          INTCPC: .WORD 0
2183                :
2184                : SUBROUTINE TO ENABLE INTERRUPTS:
2185 016242          ENAINT: MOV R0,-(SP) ;SAVE R0
2186 016244          MOV IVEC,R0 ;GET POINTER TO VECTORS
2187 016250          MOV #INTR,(R0)+ ;SET UP INTERRUPT VECTOR
2188 016254          MOV #PRI07,(R0)+
2189 016260          MOV (SP)+,R0 ;RESTORE R0
2190 016262          MOV (SP),-(SP)
2191 016264          MOV #0,2(SP) ;SET CPU TO LEVEL 0
2192 016272          RTI
2193                :
2194                : SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2195 016274          DSBINT: MOV (SP),-(SP)
2196 016276          MOV #PRI07,2(SP)
2197 016304          RTI
2198

```



```

2215 .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
2216
2217 : SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2218 :
2219 : INPUTS:
2220 :
2221 : R5 ADDRESS OF FIRST DEVICE REGISTER
2222 :
2223 : OUTPUTS:
2224 :
2225 : R0 CONTENTS OF LAST TSSR READ
2226 : CARRY SET - READY BIT SET
2227 : CLR - TIMEOUT WAITING FOR READY
2228 :
2229 016340 000401 WAITF:: BR 1$ ;NOP WHEN SUPER FIXED
2230 016342 104422 BREAK ; DO A SUPVSR BREAK FIRST.
016342 104422 TRAP C$BRK
2231 016344 012746 003000 1$: MOV #3000,-(SP) ;300 MSEC TIMER
2232 016350 016500 000002 2$: MOV TSSR(R5),R0 ;READ THE TSSR REGISTER
2233 016354 105700 TSTB R0 ;TEST FOR READY BIT SET
2234
2235 016356 100420 BMI 3$ ; EXIT ON STOP FLAG.
2236 016360 012727 000001 DELAY 1 ; WAIT 100 USEC
016360 012727 000001 MOV #1,(PC)+
016364 000000 .WORD 0
016366 013727 002116 MOV L$DLY,(PC)+
016372 000000 .WORD 0
016374 005367 177772 DEC -6(PC)
016400 001375 BNE -4
016402 005367 177756 DEC -22(PC)
016406 001367 BNE -20
2237 016410 005316 DEC (SP) ;REDUCE DELAY COUNT
2238 016412 001356 BNE 2$ ;RETRY UNTIL TIMER EXPIRES
2239 016414 000241 CLC ; C = 0, CONTROLLER STILL RUNNING...
2240 016416 000401 BR 4$ ;...OR HUNG-UP AFTER 300 MSEC.
2241 016420 000261 3$: SEC ; C = 1, CONTROLLER IS STOPPED.
2242 016422 005326 4$: DEC (SP)+ ;RESTORE STACK WITHOUT CHANGING CARRY BIT
2243 016424 000207 RTS PC
    
```

2245
 2246
 2247
 2248
 2249
 2250
 2251
 2252
 2253
 2254
 2255
 2256
 2257
 2258
 2259
 2260
 2261
 2262
 2263
 2264 016426
 2265 016426 004737 016340
 2266 016432 103014
 2267 016434 004737 016134
 2268 016440 103006
 2269 016442 032700 100000
 2270 016446 001405
 2271 016450 032700 074000
 2272 016454 001402
 2273 016456 000241
 2274 016460 000401
 2275 016462 000261
 2276 016464 000207

.SBTTL CHKTSSR - CHECK TSSR FOR READY

```

: +
: THIS ROUTINE WAITS FOR READY IN THE TSSR
: AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
: INPUT:
:         R5      ADDRESS OF CSR REGISTERS
: OUTPUT:
:         R0      CONTENTS OF TSSR
:         CARRY   SET - OKAY
:                CLR - NOT READY AMBIGUOUS, OR SC SET
: -
    
```

```

CHKTSSR:
        JSR      PC, WAITF          :WAIT FOR READY
        BCC      20$                :BRANCH IF TIME OUT
        JSR      PC, CHKAMB         :TSSR AMBIGUOUS?
        BCC      10$                :BR IF YES
        BIT      #SC, R0            :SPECIAL CONDITION SET?
        BEQ      15$                :BR IF NO
        BIT      #<SCE!BIE!RMR!NXM>, R0 :ANY ERROR BITS SET?
        BEQ      15$                :BR IF NO
10$:    CLC                          :SET FAILURE
        BR       20$
15$:    SEC                          :SET SUCCESS
20$:    RTS       PC                 :RETURN TO CALLER
    
```

```

2278 .SBTTL XNXM - CHECK FOR NONEXISTENT MEMORY
2279
2280 :+
2281 : ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
2282 : ON RETURN, IF 'C' = 1, (R1) = NEXM ADDRESS.
2283 : 'C' = 0, ALL ADDRESSES OK.
2284
2285 :CALL: MOV ADR1,R1
2286 : MOV ADR2,R2
2287 : JSR PC,NXM
2288 : RETURN ;TEST 'C' AND PROCEED.
2289 016466 012737 016520 000004 XNXM: MOV #2$,@#4 ; SET BUSERR VECTOR.
2290 016474 012737 000200 000006 : MOV #PRI04,@#6
2291 016502 005003 : CLR R3 ;FLAG.
2292 016504 005711 1$: TST (R1) ;TEST THE ADDRESS(ES).
2293 : ;IF ANY TRAP, CONTINUE AT 2$.
2294 016506 020102 : CMP R1,R2 ;OTHERWISE, CONTINUE HERE.
2295 016510 001407 : BEQ 3$ ;BR IF FINISHED (NO NEXM'S).
2296 016512 062701 000002 : ADD #2,R1 ;SET NEXT ADDRESS...
2297 016516 000772 : BR 1$ ;...AND CONTINUE.
2298
2299 016520 005103 2$: COM R3 ;GOT ONE, SET FLAG...
2300 016522 012716 016530 : MOV #3$, (SP)
2301 016526 000002 : RTI ;...AND DISMISS INTERRUPT...
2302 016530 012700 000004 3$: CLRVEC #4 ;...AND GIVE BACK THE VECTOR.
2303 016536 005703 : MOV #4,R0
2304 016540 001401 : TRAP C$CVEC
2305 016542 000261 : TST R3 ;DID WE CATCH ONE ??
2306 016544 000207 : BEQ .+4 ;NO, 'C' = 0, SKIP NEXT.
2307 : SEC ;YES, 'C' = 1, (R1) = NEXM ADDR.
2308 : RTS PC
2309
2310
2311 .SBTTL TSTLOOP - CHECK ITERATION COUNT
2312
2313 :+
2314 : SUBROUTINE TO EXECUTE TEST ITERATIONS.
2315 : EXIT WITH 'C' SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
2316 : LOOP COUNTER IS SET BY 'BEGIN.TEST' MACRO.
2317
2318 :CALL: LOOPTO ARG
2319
2320 016546 005737 002160 TSTLOOP:: TST NOITS ; ITERATIONS INHIBITED?
2321 016552 001006 : BNE 1$ ; YES.
2322 016554 005737 002174 : TST QVP ; NO.
2323 016560 100403 : BMI 1$ ;LOOPS DISALLOWED IN QUICK PASS.
2324 016562 005337 002206 : DEC LOOPCNT ; BUMP LOOP COUNTER.
2325 016566 001002 : BNE 2$
2326 016570 000241 1$: CLC ;LOOP DISALLOWED, OR DONE.
2327 016572 000401 : BR 3$
2328 016574 000261 2$: SEC ;LOOP ENABLED.
2329 016576 000207 3$: RTS PC
  
```

2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377

016600
016600 010046
016602 005037 003144
016606 005037 017046
016612 005037 005770
016616 105037 016234
016622 013700 002172
016626 006300
016630 005737 003104
016634 001430
016636 100010
016640 052760 160000 003166
016646
016646 104455
016650 000001
016652 003736
016654 005734
016656 000407
016660 052760 160001 003166 3\$:
016666
016666 104455
016670 000002
016672 004333
016674 000000
016676 012737 177777 003102 2\$:
016704
016704 013700 002172
016710 104451
016712

```
.SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
:
: +
: PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
: INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
: IN THE CURRENT RUN SEQUENCE.
: CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
:
: INPUT:
:
: R0 POINTER TO TEST ID ASCIZ STRING
:
: OUTPUT:
:
: R5 ADDRESS OF FIRST DEVICE REGISTER
:
: IMPLICIT OUTPUTS:
:
: TSTCNT UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
:
: SIDE EFFECTS:
:
: INTERRUPT LEVEL IS RASIED TO LEVEL OF
: THE DEVICE UNDER TEST
:
: -
```

```
TSTSETUP::
MOV R0,-(SP) ;SAVE THE TEST ID MESSAGE
CLR SIFLAG ; CLEAR "SOFT INIT" FLAG
CLR ERRK ; CLEAR LOCAL ERROR COUNTER.
CLR EXTA ; CLEAR ERROR EXTENSION FLAG.
CLRB INTMASK ; CLEAR INTERRUPT MASK (CHECK ERROR)
MOV UNITN,R0 ; GET THE UNIT NUMBER,
ASL R0 ; ... AND MAKE IT A WORD OFFSET.
TST NODEV ; DID STARTUP FIND THE DEVICE?
BEQ 4$ ; BR IF YES
BPL 3$ ; BR IF NOT IDLE
BIS #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
ERRDF 1,NXR,NXRERR ; NO DEVICE HERE -- PRINT IT
TRAP C$ERDF
.WORD 1
.WORD NXR
.WORD NXRERR
BR 2$
BIS #160001,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
ERRDF 2,NOINIT ; DEVICE NOT IDLE
TRAP C$ERDF
.WORD 2
.WORD NOINIT
.WORD 0
MOV #-1,DUFLG ; DROP THE UNIT
DODU UNITN
MOV UNITN,R0
TRAP C$DODU
DOCLN ; ABORT THE PASS
```

2378	016712	104444				TRAP	C\$DCLN		
2379	016714	000423				BR	5\$		
2380	016716		4\$:			RFLAGS	RO		: GET THE OPERATOR FLAGS.
	016716	104421				TRAP	C\$RFLA		
2381	016720	032700		001000		BIT	#PNT,RO		: PRINT THE TEST NUMBERS?
2382	016724	001412				BEQ	1\$: BR IF NO
2383	016726	011600				MOV	(SP),RO		:GET THE ID MESSAGE
2384	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		
	016744	104417				TRAP	C\$PNTF		
	016746	062706		000006		ADD	#6,SP		
2385	016752	005237		002204		INC	TSTCNT		: BUMP TEST COUNTER.
2386	016756				1\$:	SETPRI	IPRI		:PRIORITY THAT OF DEVICE
	016756	013700		002202		MOV	IPRI,RO		
	016762	104441				TRAP	C\$SPRI		
2387	016764	005726			5\$:	TST	(SP)+		:FIX UP THE STACK
2388	016766	013705		002176		MOV	CSRADDR,R5		: ADDRESS OF TSV REGISTERS ON UNIBUS
2389	016772	000207				RTS	PC		
2390	016774	045		123	045	TNAM:	.ASCIZ	'%SXT%A Test'	
2391							.EVEN		

```

2393          .SBTTL TSTEND - PRINT ERRORS RECEIVED
2394          :
2395          : AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2396          : IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2397          :
2398 017010 TSTEND: RFLAGS R0
          017010 104421 TRAP CSRFLA
2399 017012 030027 020000 BIT R0,#IER
2400 017016 001412 BEQ 1$ ; RR IF "IER" NOT SET.
2401 017020 PRINTF #ESUM,ERRK ; INT ERROR COUNT.
          017020 013746 017046 MOV ERRK,-(SP)
          017024 012746 017050 MOV #ESUM,-(SP)
          017030 012746 000002 MOV #2,-(SP)
          017034 010600 MOV SP,R0
          017036 104417 TRAP C$PNTF
2402 017040 062706 000006 ADD #6,SP
          017044 000207 1$: RTS PC
2403
2404 017046 000000 ERRK: 0 ; LOCAL ERROR COUNT.
2405 017050 045 101 040 ESUM: .ASCIZ /%A %D%A ERRORS/
2406 017067 105 122 122 EMAXDU: .ASCIZ /ERROR LIMIT REACHED -- DROPPING UNIT/
2407          .EVEN
2408
2409          .SBTTL INCERK - INCREMENT LOCAL ERROR COUNT
2410          :
2411          :+
2412          : ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
2413          :
2413 017134 005237 017046 INCERK: INC ERRK ; INCREMENT LOCAL ERROR COUNT
2414 017140 010046 MOV R0,-(SP) ; SAVE R0
2415 017142 013700 002172 MOV UNITN,R0 ; GET UNIT NUMBER,
2416 017146 006300 ASL R0 ; ... AND MAKE IT A WORD OFFSET.
2417 017150 062700 003166 ADD #ERTABL,R0 ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
2418 017154 005210 INC (R0) ; INCREMENT THE DEVICE ERROR COUNT
2419 017156 032710 007777 BIT #7777,(R0) ; DID WE OVERFLOW THE FIELD?
2420 017162 001001 BNE 1$ ; BR IF NO.
2421 017164 005310 DEC (R0) ; YES -- BACK IT UP TO 7777.
2422 017166 012600 1$: MOV (SP)+,R0 ; RESTORE R0
2423 017170 000207 RTS PC ; RETURN TO CALLER.
2424
2425 017172 010046 CKEMAX: MOV R0,-(SP) ; SAVE R0
2426 017174 013700 002172 MOV UNITN,R0 ; GET UNIT NUMBER
2427 017200 006300 ASL R0 ; ... AND MAKE IT A WORD OFFSET
2428 017202 016000 003166 MOV ERTABL(R0),R0 ; GET ERROR TABLE ENTRY
2429 017206 042700 170000 BIC #170000,R0 ; EXTRACT ERROR COUNT FIELD
2430 017212 020037 002164 CMP R0,GERRMAX ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
2431 017216 103004 BHIS 1$ ; BR IF YES
2432 017220 023737 017046 002162 CMP ERRK,LERRMAX ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2433 017226 103417 BLO 2$ ; BR IF NO
2434 017230 1$: RFLAGS R0 ; GET OPERATOR FLAGS
          017230 104421 TRAP CSRFLA
2435 017232 032700 000040 BIT #IDU,R0 ; IS DROPPING INHIBITED?
2436 017236 001013 BNE 2$ ; BR IF YES.
2437 017240 012737 177777 003102 MOV #-1,DUFLG ; NO -- DROP THE UNIT
2438 017246 ERRDF 4,EMAXDU
          017246 104455 TRAP C$ERDF
          017250 000004 .WORD 4
          017252 017067 .WORD EMAXDU

```

2439	017254	000000		.WORD	0	
	017256			DODU	UNITN	
	017256	013700	002172	MOV	UNITN,RO	
	017262	104451		TRAP	CSDODU	
2440	017264			DOCLN		
	017264	104444		TRAP	CSDCLN	
2441	017266	012600	2\$:	MOV	(SP)+,RO	: RESTORE RO
2442	017270	000207		RTS	PC	: RETURN TO CALLER
2443						


```
2445                    .SBTTL CKDROP - CHECK IF UNIT SHOULD BE DROPPED
2446                    :+
2447                    : CHECK IF UNIT SHOULD BE DROPPED
2448                    :-
2449 017272 010046       CKDROP: MOV     RO,-(SP)
2450 017274             FORCERROR     1$,NOTSSR
2451 017304             RFLAGS     RO
                      TRAP     CSRFLA
2452 017306 104421       BIT     #IDU,RO
                      BNE     1$
2453 017312 001010       MOV     (SP),RO
2454 017314 011600       MOV     #-1,DUFLG
2455 017316 012737 177777 003102   DODU     UNITN
2456 017324             MOV     UNITN,RO
                      TRAP     CSDODU
                      DOCLN                 :ABORT THE PASS
2457 017332 104444       TRAP     CSDCLN
                      1$: MOV     (SP)+,RO
2458 017334 012600       RTS     PC
2459 017336 000207
2460
2461
2462
2463                    .SBTTL CONFIG - DETERMINE CONFIGURATION OF SYSTEM
2464                    :
2465                    : SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
2466                    :
2467                    : CONFIG:
2468 017340             JSR     PC,SOFINIT
2469 017340 004737 016064       RTS     PC
2470 017344 000207
2471
2472
2473
```

```

2475 .SBTTL KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
2476
2477 : SUBROUTINE - ENABLE MEM MGT.
2478 :
2479 017346 005737 003122 KTON: TST KTFLG ; GOT KT?
2480 017352 001403 BEQ 1$ ; NO.
2481 017354 012737 000001 177572 MOV #1,SRO ; YES. ENABLE KT11.
2482 017362 000207 1$: RTS PC
2483
2484
2485
2486 : SUBROUTINE - DISABLE MEM MGT.
2487 :
2488 :
2489 017364 005737 003122 KTOFF: TST KTFLG ; GOT KT11?
2490 017370 001405 BEQ 1$ ; NO.
2491 017372 000240 NOP
2492 017374 000240 NOP
2493 017376 012737 000000 177572 MOV #0,SRO ; DISABLE KT.
2494 017404 000207 1$: RTS PC
2495
2496
    
```

2498
 2499
 2500
 2501
 2502
 2503
 2504
 2505
 2506
 2507
 2508
 2509
 2510
 2511
 2512
 2513
 2514
 2515
 2516
 2517 017406
 2518 017406
 2519 017412 005737 003122
 2520 017416 001433
 2521 017420 010102
 2522 000006
 2523
 2524
 2525
 2526 017452 042701 000177
 2527 017456 020137 003122
 2528 017462 103011
 2529 017464 010137 172354
 2530 017470 042702 160000
 2531 017474 062702 140000
 2532 017500 010200
 2533 017502 000261
 2534 017504 000401
 2535 017506 000241
 2536 017510 000207
 2537

.SBTTL SETMAP - SETUP PAR6 MAPPING

```

:
:
: THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
: AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
: IS RETURNED BIASED TO PAR6.
:
: INPUTS:
:
:      R0      HIGH ORDER ADDRESS BITS
:      R1      LOW ORDER ADDRESS BITS
:
: OUTPUTS:
:
:      R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
:      CARRY   SET IF SUCCESS
:              CLR IF ERROR
:
: SETMAP:
: SAVREG      :SAVE R1-R4 UNTIL NEXT RETURN
: TST         KTFLG      :SYSTEM HAVE ABOVE 28K?
: BEQ         10$        :BR IF NO
: MOV         R1,R2      :SAVE LOW ORDER BITS
: .REPT      6
: ASR        R0          :CONVERT WORD ADDRESS TO 32W BLOCKS
: ROR        R1          :MAKE IT DOUBLE PRECISION
: .ENDR
: BIC        #177,R1     :ALINE FOR LOWER 4K BOUNDARY
: CMP        R1,KTFLG   :HIGHER THAN EXISTING MEMORY?
: BHIS       10$        :BR IF YES
: MOV        R1,#KIPAR6 :SETUP MAPPING REGISTER PAR6
: BIC        #160000,R2 :SETUP DISPLACEMENT IN PAGE
: ADD        #140000,R2 :ADD IN PAR6 BIAS
: MOV        R2,R0      :RETURN IN R0
: SEC        :SET SUCCESS
: BR         15$
:
: 10$:       CLC         :SET FAILURE
: 15$:       RTS        :RETURN
: PC
    
```

```

2539          .SBTTL FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
2540          +
2541          FILL MEMORY WITH A BACKGROUND PATTERN
2542          :
2543          : INPUTS:
2544          :
2545          :     RO = BACKGROUND PATTERN
2546          :     FREE  = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2547          :     KTFLG  = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2548          :
2549          : OUTPUTS:
2550          :
2551          :     NONE
2552          :
2553          :
2554          FILLMEM:
2555          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
2556          JSR PC,KTOFF    ;DISABLE KT.
2557          MOV R0,R3       ;COPY TEST PATTERN
2558          MOV FREE,R1     ;GET FIRST FREE LOCATION
2559          MOV FRESIZ,R2   ;SIZE OF FREE SPACE BELOW 28K.
2560          10$: MOV R3,(R1)+ ;STORE A BACKGROUND WORD
2561          DEC R2          ;DONE ALL MEMORY IN FREE SPACE?
2562          BGT 10$        ;BR IF NO
2563          TST KTFLG      ; GOT KT?
2564          BEQ 55$        ; NO. GET OUT.
2565          JSR PC,KTON     ; YES. ENABLE KT.
2566          CLR R0         ;HIGH ORDER ADDRESS START
2567          MOV PST32W,R1  ;GET >28K START ADDRESS (IN 32W BLOCKS)
2568          .REPT 6
2569          CLC            ;CLEAR C BIT
2570          ROL R1         ;CONVERT BLOCKS TO WORDS
2571          ROL R0         ;MAKE IT DOUBLE PRECISION
2572          .ENDR
2573          30$: JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
2574          MOV R3,(R0)+   ;STORE TEST PATTERN IN >28K ADDRESS
2575          CMP R0,#160000 ;END OF PAR6 MAPPING AREA?
2576          BLO 30$       ;BR IF NO
2577          SUB #20000,R0  ;BACKUP INTO PAR6 MAPPING BEGIN
2578          ADD #200,@#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2579          CMP @#KIPAR6,KTFLG ;END OF MEMORY?
2580          BEQ 50$       ;BR IF YES
2581          TST T23A      ;11/23A?
2582          BEQ 35$      ;NO KEEP GOING
2583          MOV SR0,R4    ;GET SR0 CONTENTS
2584          BIC #177761,R4 ;CLEAR ALL BUT PAGE NUMBER
2585          CMP #16,R4   ;SEE IF PAGE 7
2586          BEQ 50$     ;EXIT IF THERE
2587          35$: TST T23B ;11/23B?
2588          BEQ 45$     ;NO KEEP GOING
2589          CMP @#KIPAR6,#7600 ;REACHED 18 BITS?
2590          BHIS 40$    ;YES
2591          BR 45$      ;NO KEEP GOING
2592          40$: MOV #20,SR3 ;SET 22 BIT RELOCATION
2593          45$: JMP 30$   ;KEEP GOING ON ETC.
2594          50$: JSR PC,KTOFF ;DISABLE KT.
2595          55$: RTS PC
  
```

2596
2597

```

2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621 017750
2622 017750
2623 017754 010003
2624 017756 004737 017364
2625 017762 013701 003114
2626 017766 013702 003116
2627 017772 020311
2628 017774 001411
2629 017776 010137 002230
2630 020002 005037 002226
2631 020006 010337 002222
2632 020012 011137 002224
2633 020016 000474
2634 020020 005721
2635 020022 005302
2636 020024 003362
2637 020026 005737 003122
2638 020032 001472
2639 020034 004737 017346
2640 020040 005000
2641 020042 013701 003142
2642 000006
2643
2644
2645
2646 020076 042701 000177
2647 020102 010046
2648 020104 010146
2649 020106 004737 017406
2650 020112 010004
2651 020114 012601
2652 020116 012600
2653 020120 020314
2654 020122 001411
2655 020124 010037 002226
  
```

```

.SBTTL CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
+
COMPARE MEMORY WITH A BACKGROUND PATTERN
:
INPUTS:
:
RO = BACKGROUND PATTERN
FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
KTFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
:
OUTPUTS:
:
CARRY - SET IF NO ERROR
CARRY - CLR IF ERROR
:
IMPLICIT OUTPUTS:
:
ERRHI - ERROR HIGH ADDRESS
ERRLO - ERROR LOW ADDRESS
EXPD - EXPECTED DATA
RECV - RECEIVED DATA
:
CMPMEM:
SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
MOV R0,R3 ;COPY TEST PATTERN
JSR PC,KTOFF ;DISABLE KT.
MOV FREE,R1 ;GET FIRST FREE LOCATION
MOV FRESIZ,R2 ;SIZE OF FREE SPACE BELOW 28K.
10$: CMP R3,(R1) ;FREE SPACE LOCATION EQUAL TO EXPD?
BEQ 15$ ;BR IF YES
MOV R1,ERRLO ;SAVE ADDRESS IN ERROR
CLR ERRHI ;NO HIGH ADDRESS
MOV R3,EXPD ;SAVE EXPD FOR ERROR REPORT
MOV (R1),RECV ;SAVE RECV FOR ERROR REPORT
BR 50$ ;
15$: TST (R1)+ ;POINT TO NEXT ADDRESS
DEC R2 ;DONE ALL MEMORY IN FREE SPACE?
BGT 10$ ;BR IF NO
TST KTFLG ; GOT KT?
BEQ 55$ ; NO. GET OUT.
JSR PC,KTON ; YES. ENABLE KT.
CLR R0 ;HIGH ORDER ADDRESS START
MOV PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
.REPT 6
ROL R1 ;CONVERT BLOCKS TO WORDS
ROL R0 ;MAKE IT DOUBLE PRECISION
.ENDR
BIC #177,R1 ;ALINE 4K BOUNDARY
MOV R0,-(SP) ;SAVE HIGH ORDER
MOV R1,-(SP) ;SAVE LOW ORDER
JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
MOV R0,R4 ;COPY ADDRESS BIASED TO PAR6
MOV (SP)+,R1 ;RESTORE LOW ORDER IN NON PAR6 FORMAT
MOV (SP)+,R0 ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
30$: CMP R3,(R4) ;ABOVE 28K LOCATION EQUAL EXPD?
BEQ 32$ ;BR IF YES
MOV R0,ERRHI ;SAVE HIGH ORDER IN ERROR
  
```

2656	020130	010137	002230		MOV	R1,ERRLO	:SAVE LOW ORDER IN ERROR
2657	020134	010337	002222		MOV	R3,EXPD	:SAVE EXPD FOR ERROR REPORT
2658	020140	011437	002224		MOV	(R4),RECV	:SAVE RECV FOR ERROR REPORT
2659	020144	000421			BR	50\$:
2660	020146	062701	000002	32\$:	ADD	#2,R1	:UPDATE NON PAR6 ADDRESS
2661	020152	005500			ADC	R0	:MAKE IT DOUBLE PRECISION ADD
2662	020154	062704	000002		ADD	#2,R4	:UPDATE PAR FORMAT ADDRESS
2663	020160	020427	169000		CMP	R4,#160000	:END OF PAR6 MAPPING AREA?
2664	020164	103755			BLO	30\$:BR IF NO
2665	020166	162704	020000		SUB	#20000,R4	:BACKUP INTO PAR6 MAPPING BEGIN
2666	020172	062737	000200	172354	ADD	#200,@#KIPAR6	:POINT TO NEXT 4K BLOCK >28K.
2667	020200	023737	172354	003122	CMP	@#KIPAR6,KTFLG	:END OF MEMORY?
2668	020206	101744			BLOS	30\$:BR IF NO
2669	020210	004737	017364	50\$:	JSR	PC,KTOFF	:TURN OFF MEMORY MAPPING
2670	020214	000241			CLC		:SET FAILURE
2671	020216	000403			BR	60\$:
2672	020220	004737	017364	55\$:	JSR	PC,KTOFF	:TURN OFF MEMORY MAPPING
2673	020224	000261			SEC		:SET SUCCESS
2674	020226	000207		60\$:	RTS	PC	
2675							

2677
 2678
 2679
 2680
 2681
 2682
 2683
 2684
 2685
 2686
 2687
 2688
 2689
 2690
 2691
 2692
 2693
 2694
 2695
 2696
 2697 020230
 2698 020230 010446
 2699 020232 010346
 2700 020234 010246
 2701 020236 010146
 2702 020240 010546
 2703 020242 016605 000012
 2704 020246 004736
 2705 020250 012601
 2706 020252 012602
 2707 020254 012603
 2708 020256 012604
 2709 020260 012605
 2710 020262 000207
 2711

```

        .SBTTL  REGSAV - SAVE R1-R5 ON STACK
        :+
        :ROUTINE TO
        :SAVE R1 THROUGH R5 ON THE STACK.
        :CALLING SEQUENCE:
        :
        :       JSR      R5,REGSAV
        :
        :THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
        :THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
        :THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
        :REGISTERS.
        :
        :THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
        :CALLED VIA A JSR PC INSTRUCTION
        :-
    REGSAV:
        MOV      R4,-(SP)
        MOV      R3,-(SP)
        MOV      R2,-(SP)
        MOV      R1,-(SP)
        MOV      R5,-(SP)
        MOV      10.(SP),R5
        JSR      PC,@(SP)+
        MOV      (SP)+,R1
        MOV      (SP)+,R2
        MOV      (SP)+,R3
        MOV      (SP)+,R4
        MOV      (SP)+,R5
        RTS      PC
    
```



```

2713 .SBTTL GETPAT - GET 8 BIT PATTERN FROM OPERATOR
2714
2715
2716 :ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
2717
2718 :INPUTS:
2719
2720 :NONE.
2721
2722 :OUTPUTS:
2723
2724 :RO OCTAL NUMBER FROM THE OPERATOR
2725
2726 :CALLING SEQUENCE:
2727
2728 :JSR PC,GETPAT
2729
2730 :-
2731
2732 GETPAT::
2733 1$: SAVREG ;SAVE THE GENERAL REGISTERS
2734 020264 GMANID DATASC,PATDAT,0,377,0,377,NO
020270 104443 TRAP CSGMAN
020272 000406 BR 10000$
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$HILIM
020310
10000$:
2735 020310 BNCOMPLETE 1$ ;RETRY IF ERROR
020310 103367 BCC 1$
2736 020312 013700 020320 MOV PATDAT,RO ;DATA PATTERN FROM OPERATOR
2737 020316 000207 RTS PC ;RETURN TO CALLER
2738
2739 :+
2740 :LOCAL DATA AREA
2741 :-
2742
2743 020320 000000 PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
2744 020322 105 116 124 DATASC: .ASCIZ 'ENTER DATA PATTERN'
2745 .EVEN
  
```

```

2747          .SBTTL  GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
2748          :+
2749          :ROUTINE TO ISSUE A MENU AND GET
2750          :THE OPERATOR'S RESPONSE.
2751          :
2752          :INPUTS:
2753          :
2754          :       R0      ADDRESS OF ASCIZ STRING OF MENU
2755          :       R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
2756          :
2757          :OUTPUTS:
2758          :
2759          :       R0      NUMBER OF THE OPERATOR'S SELECTION
2760          :
2761          :-
2762
2763
2764 020346      GETSEL::
2765 020346      SAVREG                ;SAVE GENERAL REGISTERS
2766 020352      010002      MOV      R0,R2                ;SAVE THE MENU ADDRESS
2767 020354      010203      1$:      MOV      R2,R3                ;START OF MENU STRING
2768 020356      005713      2$:      TST      (R3)                ;END OF ASCII ?
2769 020360      001412      BEQ      3$                    ;BRANCH IF ALL LINES DISPLAYED
2770 020362      PRINTF      #SELASC,(R3)+                ;DISPLAY THE MENU
2771 020362      012346      MOV      (R3)+,-(SP)
2772 020364      012746      020532      MOV      #SELASC,-(SP)
2773 020370      012746      000002      MOV      #2,-(SP)
2774 020374      010600      MOV      SP,R0
2775 020376      104417      TRAP     C$PNTF
2776 020400      062706      000006      ADD      #6,SP
2777 020404      000764      BR       2$
2778 020406      3$:      GMANID      MENASC,MENRES,D,-1,0,-1,NO
2779 020406      104443      TRAP     C$GMAN
2780 020410      000406      BR       10001$
2781 020412      020566      .WORD   MENRES
2782 020414      000042      .WORD   T$CODE
2783 020416      020537      .WORD   MENASC
2784 020420      177777      .WORD   -1
2785 020422      000000      .WORD   T$LOLIM
2786 020424      177777      .WORD   T$HILIM
2787 020426      10001$:
2788 020426      BNCOMPLETE      1$                ;RETRY IF ERROR
2789 020426      103352      BCC      1$
2790 020430      013700      020566      MOV      MENRES,R0                ;GET THE OPERATOR'S REPLY
2791 020434      020001      CMP      R0,R1                ;COMPARE TO MAXIMUM ALLOWED
2792 020436      101411      BLOS     5$                    ;BRANCH IF OK
2793 020440      PRINTF      #MENERR                ;DISPLAY ERROR MESSAGE
2794 020440      012746      020464      MOV      #MENERR,-(SP)
2795 020444      012746      00C001      MOV      #1,-(SP)
2796 020450      010600      MOV      SP,R0
2797 020452      104417      TRAP     C$PNTF
2798 020454      062706      000004      ADD      #4,SP
2799 020460      000735      BR       1$                    ;RETRY
2800 020462      000207      5$:      RTS      PC                ;RETURN TO CALLER
2801 020464      045      116      045      MENERR: .ASCIZ  '%N% *** Menu Selection Too Large ***'
2802 020532      045      116      045      SELASC: .ASCIZ  '%N%T'
2803 020537      105      156      164      MENASC: .ASCIZ  'Enter Menu Selection: '
    
```

2783
2784 020566 000000

MENRES: .EVEN .WORD 0

```

2786 .SBTTL CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
2787
2788
2789 :ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2790
2791 :INPUT:
2792
2793 :       NONE.
2794
2795 :OUTPUT:
2796
2797 :       CARRY  0      MANUAL INTERVENTION NOT ALLOWED
2798 :             1      MANUAL INTERVENTION IS OK
2799
2800 :SIDE EFFECTS:
2801
2802 :       A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2803 :       NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2804 :       ALLOWED.
2805
2806 :-
2807
2808 CHKMAN::
2809       SAVREG                ;SAVE THE REGISTERS
2810       MANUAL                ;SEE IF MANUAL INTERVENTION OK
2811       TRAP C$MANI
2812       BCOMPLETE 1$          ;BRANCH IF ALLOWED
2813       BCS 1$
2814       PRINTF #NOMAN        ;PRINT THE WARNING MESSAGE
2815       MOV #NOMAN,-(SP)
2816       MOV #1,-(SP)
2817       MOV SP,R0
2818       TRAP C$PNTF
2819       ADD #4,SP
2820       CLC                    ;CLEAR CARRY FOR ERROR
2821       RTS PC                  ;RETURN
2822
2823       .ASCIZ '%N% *** Manual Intervention not Allowed - Test Aborted ***'
2824       .even
  
```

```

2819 .SBTTL ENVIRN - SETUP FREE DIAGNOSTIC SPACE
2820
2821 : SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
2822 :
2823 ENVIRN: MEMORY R0
          020720 104431 TRAP C$MEM
2824 020722 010037 003114 MOV R0,FREE ; GET 1ST FREE ADDRESS...
2825 020726 062737 000002 003114 ADD #2,FREE
2826 020734 011037 003116 MOV (R0),FRESIZ ;...AND WORD COUNT.
2827 020740 162737 000004 003116 SUB #4,FRESIZ
2828 020746 013702 002012 MOV L$UNIT,R2 ; GET NUMBER OF UNITS
2829 020752 162737 000007 003116 10$: SUB #7,FRESIZ ; TAKE AWAY 7 WORDS PER UNIT
2830 020760 005302 DEC R2
2831 020762 001373 BNE 10$
2832 020764 013700 003114 MOV FREE,R0 ;GET FIRST FREE ADDRESS
2833 020770 063700 003116 ADD FRESIZ,R0 ;POINT TO LAST FREE ADDRESS
2834 020774 162700 000002 SUB #2,R0 ;BACKUP 1 WORD
2835 021000 010037 003120 MOV R0,FREEH: ;STORE LAST FREE ADDRESS
2836 021004 000240 NOP ;*****
2837 021006 012701 177520 MOV #BDVPCR,R1 ;GET BDV11 PCR ADDRESS
2838 021012 010102 MOV R1,R2 ;COPY TO R2
2839 021014 062702 000002 ADD #2,R2 ;SET THE RANGE
2840 021020 004737 016466 JSR PC,XNXM ;SEE IF WE HAVE ONE
2841 021024 103001 BCC 15$ ;OK TO SET FLAGS
2842 021026 000445 BR 40$ ;RETURN WITH FLAGS CLEAR
2843 021030 013701 177520 15$: MOV BDVPCR,R1 ;SAVE PCR CONTENTS
2844 021034 062701 000001 ADD #1,R1 ;ADD ONE TO IT
2845 021040 012702 177520 MOV #BDVPCR,R2 ;GET BDV11 PCR ADDRESS
2846 021044 005212 INC (R2) ;TRY TO WRITE TO IT
2847 021046 013703 177520 MOV BDVPCR,R3 ;GET RESULTS
2848 021052 020103 CMP R1,R3 ;DID IT CHANGE?
2849 021054 001017 BNE 20$ ;NO, MUST BE 11/23B
2850 021056 005237 003134 INC T23A ;SET THE FLAG
2851 021062 042737 170000 002120 BIC #170000,L$HIME ;SUPERVISOR COULD BE WRONG
2852 021070 000240 NOP ;BR 40$ FOR RELEASE
2853 021072 PRINTF #M8186 ;TELL THE SYSTEM TYPE
          021072 012746 005552 MOV #M8186,-(SP)
          021076 012746 000001 MOV #1,-(SP)
          021102 010600 MOV SP,R0
          021104 104417 TRAP C$PNTF
2854 021106 062706 000004 ADD #4,SP
          021112 000413 BR 40$ ;RETURN
2855 021114 005237 003136 20$: INC T23B ;SET THE FLAG
2856 021120 000240 NOP ;BR 40$ FOR RELEASE
2857 021122 PRINTF #M8189 ;TELL THE SYSTEM TYPE
          021122 012746 005643 MOV #M8189,-(SP)
          021126 012746 000001 MOV #1,-(SP)
          021132 010600 MOV SP,R0
          021134 104417 TRAP C$PNTF
          021136 062706 000004 ADD #4,SP
2858 021142 000207 40$: RTS PC ;RETURN
2859

```

```

2861                                     .SBTTL  KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS
2862
2863                                     :+
2864                                     :ROUTINE TO INIT KT-11
2865                                     :-
2866
2867
2868 021144                               KTINIT:
2869 021144 005037 003122                 CLR    KTFLG                ; INIT >28K MEMORY FLAG
2870 021150 005037 003124                 CLR    KTENABLE           ; INIT TEST >28K FLAG
2871 021154 023727 002120 001577         CMP    L$HIME,#1577       ; GOT ENOUGH MEMORY (>28K)?
2872 021162 101444                        BLOS   9$                 ; NO.
2873 021164 013700 000004                 MOV    @#ERRVEC,R0        ; SAVE OLD ERR VEC PTR.
2874 021170 012737 021262 000004         MOV    #2$,@#ERRVEC       ; SET ERR VEC PTR.
2875 021176 005737 177572                 TST    @#SRO              ; GOT KT11?
2876 021202 000240                        NOP                       ; (TRAP IF NO).
2877 021204 013737 002120 003122         MOV    L$HIME,KTFLG       ; YES. SET KT FLAG.
2878 021212 042737 000177 003122         BIC    #177,KTFLG         ;
2879 021220 010037 000004                 MOV    R0,@#ERRVEC        ; RESTORE OLD ERR VEC PTR.
2880 021224 005000                        CLR    R0                 ; R0 = AR DATA.
2881 021226 012701 172340                 MOV    #KIPAPO,R1         ; R1 = KI REGS PTR.
2882 021232 012761 077406 177740 1$:     MOV    #77406,-40(R1)     ; SET DESCRIPTOR REG.
2883 021240 010021                        MOV    R0,(R1)+          ; SET KIPAR REG.
2884 021242 062700 000200                 ADD    #200,R0            ; BUMP AR DATA BY '4K'.
2885 021246 020027 002000                 CMP    R0,#2000          ; AT 'I/O'?
2886 021252 001367                        BNE    1$                 ; NO.
2887 021254 012741 177600                 MOV    #177600,-(R1)     ; YES. SET KTPAR7 FOR I/O.
2888 021260 000405                        BR     9$                 ;
2889
2890 021262 012716 021270 2$:             MOV    #6$,(SP)          ; SET UP RETURN
2891 021266 000002                        RTI                       ; RTI TO NEXT LOCATION
2892
2893 021270 010037 000004 6$:             MOV    R0,@#ERRVEC       ; RESTORE OLD ERR VEC PTR.
2894
2895 021274 000207 9$:                     RTS    PC
2896

```

```

2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911 021276
2912
2913 021276 005737 002216
2914 021302 001020
2915 021304 012737 100206 021350
2916 021312 012737 021360 021352
2917 021320 012737 000006 021356
2918 021326 012737 100010 021360
2919 021334 012704 021350
2920 021340 004737 010752
2921 021344 000207
2922
2923
2924
2925
2926 021350
2927
2928 021350 000000
2929 021352 000000
2930 021354 000000
2931 021356 000000
2932
2933
2934
2935
2936 021360 000000
2937 021362 000000
2938 021364 000000
2939
2940
    
```

```

:↑
SUBROUTINE TO SET EXTENDED FEATURES SWITCH
Requires that SOFINIT and WRTCHR have been done previous to call.

INPUTS:
R5 CURRENT UNIT NUMBER
OUTPUTS:
The Extended Features Switch is set.

INVERT::

TST EXTFEA ; IS SWITCH SET?
BNE 1$ ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
MOV #100206,CMDPKT ; WRT SUB-SYS MEM CMD
MOV #WSMBK,CMDPKT+2 ; MSG BUF ADDR
MOV #6,CMDPKT+6 ; BYTE COUNT
MOV #100010,WSMBK ; INVERT THE SWITCH
MOV #CMDPKT,R4 ; SET CMDPKT INTO R4
JSR PC,WRTCHR ; DO IT
RTS PC ; RETURN

: COMMAND PACKET.
. = <.+3>&177774 ;MUST BE ON MOD 4 BOUNDRY.

CMDPKT:: 0 ;1ST WORD IS TS05 COMMAND.
0 ;2ND WORD IS THE BUFFER LOW ADDRESS.
0 ;3RD WORD IS THE BUFFER HIGH ADDRESS.
0 ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.

: WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
WSMBK:: 0 ;1ST WORD:: SEL 0
0 ;2ND WORD:: SEL 2
0 ;3RD WORD:: SEL 4
.EVEN
    
```

```

2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952 021366
2953
2954 021366
2955 021372 005037 003126
2956 021376 005037 003130
2957 021402 005037 003132
2958 021406 005737 003136
2959 021412 001407
2960 021414 023727 002120 007777
2961 021422 103406
2962 021424 004737 021542
2963 021430 000427
2964 021432 005737 003134 1$:
2965 021436 001413
2966 021440 023727 002120 005777 2$:
2967 021446 101023
2968 021450 023727 002120 003777
2969 021456 103403
2970 021460 004737 021542
2971 021464 000411
2972 021466 023727 002120 001577 4$:
2973 021474 103410
2974 021476 004737 021542
2975 021502 062737 000077 003132
2976 021510 005237 003126 13$:
2977 021514 000411
2978 021516 000410 14$:
2979 021520
    021520 012746 005456
    021524 012746 000001
    021530 010600
    021532 104417
    021534 062706 000004
2980 021540 000207 15$:
    SAVREG
    CLR NXMFLG
    CLR NXML
    CLR NXMHI
    TST T23B
    BEQ 1$
    CMP LSHIME,#7777
    BLO 2$
    JSR PC,NXMTST
    BR 13$
    TST T23A
    BEQ 4$
    CMP LSHIME,#5777
    BHI 14$
    CMP LSHIME,#3777
    BLO 4$
    JSR PC,NXMTST
    BR 13$
    CMP LSHIME,#1577
    BLO 14$
    JSR PC,NXMTST
    ADD #77,NXMHI
    INC NXMFLG
    BR 15$
    BR 15$
    PRINTF #NOMEM
    MOV #NOMEM,-(SP)
    MOV #1,-(SP)
    MOV SP,R0
    TRAP C$PNTF
    ADD #4,SP
    RTS PC
    ;SAVE THE REGISTERS
    ;CLEAR THE FLAG
    ;CLEAR THE TEST ADDRESS LO
    ;CLEAR THE TEST ADDRESS HI
    ;IS IT A 11/23B?
    ;NO
    ; GREATER THAN 128K
    ; NO
    ;SETUP THE ADDRESS
    ;SET THE FLAG AND EXIT
    ;IS IT A 11/23A?
    ;NO
    ;GREATER THAN 96K
    ;YES,23A/23B WITH 128K MEMORY
    ;GREATER THAN 64K BUT LESS THAN 92K?
    ;NO, CHECK 24K
    ;SETUP THE ADDRESS
    ;SET THE FLAG AND EXIT
    ;GREATER THAN 24K BUT LESS THAN 64K?
    ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
    ;SETUP THE ADDRESS
    ;FOOL THE 11/02 & 11/03
    ;SET THE FLAG
    ;EXIT
    ;NOP FOR PRINTOUT
    ;TELL THEM & EXIT ***NO PRINT*****
    ;RETURN
    
```

```

2981
2982
2983
2984
2985
2986
2987
2988
2989
2990 021542 013701 002120
2991 021546 062701 000200
2992 021552 042701 000177
2993 021556 010102
    ;+
    ; SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
    ;-
    ;OUTPUTS: NXMLO,NXMHI
    ;SETUP WITH NXM ADDRESS
    ;-
    NXMTST: MOV LSHIME,R1
    ADD #200,R1
    BIC #177,R1
    MOV R1,R2
    ;GET TOP OF MEMORY
    ;MAKE IT I/O BLOCK OR OTHER NXM
    ;RESAVE RESULTS
    
```



```
2994          0C0006          .REPT 6
2995          .ASL R1          ;PUT IN PLACE FOR XFER
2996          .ENDR          ;SAVE TEST ADDRESS LOW
2997 021574 010137 003130  MOV R1,NXML0
2998          000012          .REPT 10
2999          .ASR R2          ;PUT IN PLACE FOR XFER
3000          .ENDR
3001 021624 042702 177700  BIC #177700,R2
3002 021630 010237 003132  MOV R2,NXMHI
3003 021634 000207          RTS PC
3004
3005
3006
3007
3008 021636          ENDMOD
```

7
8
9 021636
021636
10
16

.TITLE TSV4 - MISCELLANEOUS SECTIONS
BGNMOD TSV4
TSV4::

```
18  
19 021636 .SBTTL PROTECTION TABLE  
    021636 BGNPROT  
20 021636 177777 177777 177777 LSPROT::  
21 021646 .WORD -1, -1, -1, -1  
22      ENDPROT
```

;NO DEVICE PROTECTION REQUIRED.

24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64

.SBTTL INITIALIZE SECTION

```

:++
:THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
:AT THE BEGINNING OF EACH PASS.
:IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
:IF "CONTINUE", NOTHING IS REQUIRED.

```

```

:--
:+
:INSERT TEMPORARY JUMP TO ODT
:--

```

```

BGNINIT
LSINIT::
40$: CLR EXTFEA
      CLR NXMFLG
      MOV #EPR1,EPR1SW ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
      CLR SIFLAG ;CLEAR "SOFT INIT" FLAG
      CLR KTENABLE ;CLEAR TEST ABOVE 28K FLAG
      CLR RAMSIZ ;CLEAR RAM SIZE FOR RAMERR ROUTINE
      READEF #EF.CONTINUE
      MOV #EF.CONTINUE,R0
      TRAP C$REFG
45: BNCOMPLETE 1$
      BCC 1$
      CMP UNITN,LSUNIT ;UNIT IN RANGE?
      BHIS 4$ ;BR IF NO.
      TST DUFLG ;DROPPED UNIT?
      BMI NXTU ;BR IF YES
      MOV UNITN,R1
      ASL R1
      TST ERTABL(R1)
      BEQ SETU
      BIT #BIT14,ERTABL(R1) ;DROPPED?
      BNE NXTU
      EXIT INIT ;DO NOTHING IF "CONTINUE".
      TRAP C$EXIT
      .WORD L10030-
57: 1$: READEF #EF.NEW
      MOV #EF.NEW,R0
      TRAP C$REFG
58: BNCOMPLETE NXTU ;TAKE NEXT UNIT IF NOT NEW PASS.
      BCC NXTU
      READEF #EF.START
      MOV #EF.START,R0
      TRAP C$REFG
60: BCOMPLETE 2$
      BCS 2$
      READEF #EF.RESTART
      MOV #EF.RESTART,R0
      TRAP C$REFG
62: BNCOMPLETE 31$
      BCC 31$
63: 2$: BRESET ;1ST PASS, BUS-INIT...
      TRAP C$RESET ;BUS RESET.

```

```

021646
021646
021646 005037 002216
021652 005037 003126
021656 012737 006356 002170
021664 005037 003144
021670 005037 003124
021674 005037 002272
021700
021700 012700 000036
021704 104447
021706
021706 103023
021710 023737 002172 002012
021716 103070
021720 005737 003102
021724 100472
021726 013701 002172
021732 006301
021734 005761 003166
021740 001516
021742 032761 040000 003166
021750 001060
021752
021752 104432
021754 000416
021756
021756 012700 000035
021762 104447
021764
021764 103052
021766
021766 012700 000040
021772 104447
021774
021774 103404
021776
021776 012700 000037
022002 104447
022004
022004 103031
022006
022006
022006 104433

```

```

65 022010 005037 002204 CLR TSTCNT ;NUMBER OF TESTS RUN IN PASS
66 022014 005037 002212 CLR FATFLG ;CLEAR FATAL ERROR COUNT
67 022020 005037 003134 CLR T23A ;CLEAR 11/23A FLAG
68 022024 005037 003136 CLR T23B ;CLEAR 11/23B FLAG
69 : MOV #340,-(SP)
70 : MOV #20$,-(SP) ;RETURN TO DEBUGGER
71 : JMP 0.ODT ;:ENTER THE DEBUGGER
72 022030 005037 003370 CLR SKIPT ;CLEAR THE SUBTEST "SKIPPER"
73 022034 20$:
74 022034 012737 177777 002174 MOV #-1,QVP ;...QUICK VERIFY...
75 022042 004737 020720 JSR PC,ENVIRN ;SET ENVIRONMENT.
76 022046 004737 021144 JSR PC,KTINIT ;INITIALIZE KT MEMORY MANAGEMENT
77 022052 012700 003166 MOV #ERTABL,RO
78 022056 005020 30$: CLR (RO)+ ;CLEAR THE ERROR TABLE
79 022060 020027 003366 CMP RO,#ERTABE
80 022064 103774 BLO 30$
81 022066 000404 BR 4$
82 022070 005037 002174 31$: CLR QVP
83 022074 000137 022144 JMP PASRPT ;GO REPORT THE STATUS
84
85 022100 4$:
86 022100 012737 177777 002172 NEWPAS: MOV #-1,UNITN ;INIT UNIT NUMBER...
87 022106 005037 002210 CLR DEVCNT ;CLEAR COUNT OF DEVICES RUNNING
88 022112 NXTU: BREAK
89 022112 104422 TRAP CSBRK
90 022114 005237 002172 INC UNITN ;...AND SET NEXT UNIT NUMBER.
91 022120 023737 002172 002012 CMP UNITN,LSUNIT
92 022126 103423 BLO SETU
93 022130 012737 177777 003102 MOV #-1,DUFLG
94 022136 000401 BR 11$
95 022140 104444 DOCLN
96 022142 000240 11$: TRAP CSDCLN
97 022144 023727 002012 000001 PASRPT: NOP
98 022152 101752 CMP LSUNIT,#1 ;HOW MANY UNITS SELECTED?
99 022154 005737 002210 BLOS NEWPAS ;BR IF ONLY 1
100 022160 001747 TST DEVCNT ;ARE ANY STILL RUNNING?
101 022162 BEQ NEWPAS ;BR IF NO
102 022162 104421 TRAP CSRFLA
103 022164 032700 000100 BIT #ISR,RO ;SHOULD WE PRINT STATISTICS
104 022170 001343 BNE NEWPAS ;BR IF NO
105 022172 DORPT
106 022172 104424 TRAP CSDRPT
107 022174 000741 BR NEWPAS
108
109 022176 10$:
110 022176 013700 002172 SETU: GPWARD UNITN,RO ;GET UNIT N P-TABLE POINTER.
111 022202 104442 MOV UNITN,RO
112 022204 103342 TRAP C$GPHRD ;BR IF UNIT NOT AVAILABLE.
113 022206 005037 003102 BNCOMPLETE NXTU
114 022212 005237 002210 BCC NXTU
115 022216 012001 CLR DUFLG ;CLEAR 'DROPPED' FLAG.
116 022220 010137 002176 INC DEVCNT ;GET 1ST REGISTER ADDRESS.
117 : MOV (RO)+,R1 ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
118 : MOV R1,CSRADDR

```

```

115
116 022224 012001          MOV      (R0)+,R1          ;GET VECTOR ADDRESS.
117                      ;MOV      (R0),R2          ;GET INTERRUPT PRIORITY
118                      ;MOV      R2,IPRI        ;SET INTERRUPT PRIORITY.
119 022226 010137 002200   MOV      R1,IVEC          ;SET INTERRUPT VECTOR POINTER...
120 022232 012721 016306   MOV      #INTR,(R1)+      ;...VECTOR...
121 022236 013721 002202   MOV      IPRI,(R1)+      ;...AND PRIORITY.
122
123 022242                1$:
124                      ;          TST      QVP          ;1ST PASS ??
125                      ;          BEQ      5$          ;NO, SKIP THE PASS 1 STUFF.
126
127
128                      ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
129                      ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
130
131 022242 013701 002172   MOV      UNITN,R1
132 022246 006301          ASL      R1
133 022250 052761 100000 003166  BIS      #BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
134 022256 005037 005770          CLR      EXTA            ;CLEAR ERROR EXTENSION FLAG.
135 022262 023727 002012 000001  CMP      LSUNIT,#1       ;ARE WE TESTING MULTIPLE UNITS?
136 022270 101416          BLOS     10$            ;BR IF NO.
137 022272                RFLAGS    R0            ;YES -- GET OPERATOR FLAGS.
138 022274 104421          TRAP     CSRFLA
139 022300 032700 001000   BIT      #PNT,R0         ;SHOULD WE PRINT UNIT #?
140 022302 001412          BEQ      10$            ;BR IF NOT.
141 022302                PRINTF    #PUNIT,UNITN ;PRINT THE UNIT #
142 022302 013746 002172   MOV      UNITN,-(SP)
143 022306 012746 022374   MOV      #PUNIT,-(SP)
144 022312 012746 000002   MOV      #2,-(SP)
145 022316 010600          MOV      SP,R0
146 022320 104417          TRAP     C$PNTF
147 022322 062706 000006   ADD      #6,SP
148 022326                10$:
149 022326 005037 003104   CLR      NODEV
150 022332 013701 002176   MOV      CSRADDR,R1     ;ADDRESS OF FIRST REGISTER
151 022336 010102          MOV      R1,R2         ;START OF REGISTERS
152 022340 062702 000002   ADD      #TSSR,R2      ;ADDRESS OF TSSR REGISTER
153 022344 004737 016466   JSR      PC,XXNM       ;TEST BOTH CONTROLLER REGISTERS...
154 022350 103005          BCC     2$             ;...AND BR IF ALL OK.
155 022352 010137 003104   MOV      R1,NODEV      ;FLAG DEVICE AS NON-EXISTENT
156 022356 012737 177777 003102  MOV      #-1,DUFLG     ;DROP THIS UNIT.
157 022364
158
159                      2$:
160                      ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
161
162                      5$:
163 022364                SETPRI   #PRI00          ;ENABLE INTERRUPTS.
164 022364 012700 000000   MOV      #PRI00,R0
165 022370 104441          TRAP     C$SPRI
166 022372                ENDINIT
167 022372                L10030:
168 022372 104411          TRAP     C$INIT
169
170 022374 045 116 045 PUNIT: .ASCIZ /%N%N%A***** TESTING UNIT %D2%A *****/
171 .EVEN

```

```

160
161
162
163
164
165
166
167 022442
    022442
168 022442 010001
169 022444 006301
170 022446 052761 100000 003166
171 022454 042761 040000 003166
172 022462
    022462 010046
    022464 012746 022510
    022470 012746 000002
    022474 010600
    022476 104417
    022500 062706 000006
173 022504
    022504 000167
    022506 000026
174 022510 045 116 045 1$:
175
176
177 022536
    022536
    022536 104452
178
179
180
181
182
183
184
185
186
187
188
189 022540
    022540
190 022540 012737 177777 003102
191 022546 010001
192 022550 006301
193 022552 052761 140000 003166
194 022560 000240 000240 000240
195 022566
    022566 010046
    022570 012746 022614
    022574 012746 000002
    022600 010600
    022602 104417
    022604 062706 000006
196 022610
    022610 000167
    022612 000030
    
```

.SBTTL ADD AND DROP UNITS SECTIONS

```

:++
: THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
: TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
: OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
:--
    
```

```

BNAU
L$AU::
MOV    R0,R1           ; GET UNIT TO BE ADDED (R0)
ASL    R1              ; MAKE IT A WORD INDEX
BIS    #10000,ERTABL(R1) ; SET THE 'ACTIVE' BIT
BIC    #40000,ERTABL(R1) ; CLEAR THE 'DROPPED' BIT
PRINTF #1$,R0
MOV    R0,-(SP)
MOV    #1$,-(SP)
MOV    #2,-(SP)
MOV    SP,R0
TRAP   C$PNTF
ADD    #6,SP
EXIT   AU
.WORD  JSJMP
.WORD  L10031-2-.
.ASCIZ /%N% UNIT %D% ADDED/
.EVEN
    
```

```

ENDAU           ; UNUSED.
L10031:
TRAP   C$AU
    
```

```

:++
: THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
: TO BE REMOVED FROM THE TEST LIST.
:
:
    
```

```

: SUPVSR DOES THE 'DROPPING'. THIS IS JUST TO TELL THE MAN.
: 'DROPPED' UNITS ARE RE-SELECTED ON OPERATOR 'STA' OR 'ADD'
: COMMAND, OTHERWISE REMAIN INACTIVE. THE 'DISPLAY' COMMAND
: WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
: WHICH ARE STILL ACTIVE.
: UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.
    
```

```

BGNDU
L$DU:.
MOV    #-1,DUFLG
MOV    R0,R1
ASL    R1
BIS    #140000,ERTABL(R1) ; SAY DROPPED
      240,240,240         ; ?????????
PRINTF #1$,R0
MOV    R0,-(SP)
MOV    #1$,-(SP)
MOV    #2,-(SP)
MOV    SP,R0
TRAP   C$PNTF
ADD    #6,SP
EXIT   DU
.WORD  JSJMP
.WORD  L10032-2-.
    
```

197 022614 045 116
198
199 022644
022644
022644 104453
200
201
202
203 022646
022646
204 022646 013705 002176
205 022652 012703 000550
206 022656 004737 016340
207 022662 103420
208 022664
022664 012727 000372
022670 000000
022672 013727 002116
022676 000000
022700 005367 177772
022704 001375
022706 005367 177756
022712 001367
209 022714 005303
210 022716 001357
211 022720 004737 017272
212 022724
213 022724
022724
022724 104461

045 1\$: .ASCIZ /%N% UNIT %D% DROPPED/
.EVEN
ENDDU
L10032: TRAP C\$DI
:++
: AUTO-DROP CODE SECTION.
:--
BGNAUTO
L\$AUTO::
MOV CSRADDR,R5 ;POINT TO DEVICE REGISTER
MOV #360.,R3 ;ENOUGH TIME FOR 2400' REEL TO REWIND
10\$: JSR PC,WAITF ;WAIT FOR SSR TO SET
BCS 20\$;LEAVE WHEN SSR IS SET
DELAY 250. ;WAIT FOR .25 SECONDS
MOV #250.,(PC)+
.WORD 0
MOV L\$DLY,(PC)+
.WORD 0
DEC -6(PC)
BNE -.4
DEC -22(PC)
BNE -.20
DEC R3 ;BUMP COUNTER DOWN
BNE 10\$;KEEP GOING
JSR PC,CKDROP ;TRY AND DROP UNIT
20\$: ENDAUTO ; UNUSED.
L10033: TRAP C\$AUTO


```

023072 012746 000002      MOV      #2,-(SP)
023076 010600      MOV      SP,R0
023100 104416      TRAP     C$PNTS
023102 062706 000006      ADD      #6,SP
254 023106 000431      BR       4$
255 023110 020227 160001      3$:     CMP      R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
256 023114 001012      BNE     30$           ; BR IF NO.
257 023116      PRINTS  #DEVNRD,R3
023116 010346      MOV      R3,-(SP)
023120 012746 023405      MOV      #DEVNRD,-(SP)
023124 012746 000002      MOV      #2,-(SP)
023130 010600      MOV      SP,R0
023132 104416      TRAP     C$PNTS
023134 062706 000006      ADD      #6,SP
258 023140 000414      BR       4$
259 023142 042702 170000      30$:    BIC      #^C7777,R2
260 023146      PRINTS  #DEVDRD,R3,R2
023146 010246      MOV      R2,-(SP)
023150 010346      MOV      R3,-(SP)
023152 012746 023466      MOV      #DEVDRD,-(SP)
023156 012746 000003      MOV      #3,-(SP)
023162 010600      MOV      SP,R0
023164 104416      TRAP     C$PNTS
023166 062706 000010      ADD      #10,SP
261 023172 062704 000002      4$:     ADD      #2,R4
262 023176 005203      INC      R3
263 023200 020427 003366      CMP      R4,#ERTABE
264 023204 103701      BLO     1$
265 023206 012604      MOV      (SP)+,R4
266 023210 012603      MOV      (SP)+,R3
267 023212 012602      MOV      (SP)+,R2
268 023214      ENDRPT      ; UNUSED.
023214      L10035:
023214 104425      TRAP     C$RPT
269
270
271 023216      045      116      045  DEVSUM: .ASCIZ  /%N%ADEVICE STATUS SUMMARY:%N/
272 023253      045      101      040  DEVONL: .ASCIZ  /%A  UNIT %D3%A  ONLINE,  ERRORS = %D%N/
273 023323      045      101      040  DEVNXR: .ASCIZ  /%A  UNIT %D3%A  DROPPED,  NON-EXISTENT REGISTER%N/
274 023405      045      101      040  DEVNRD: .ASCIZ  /%A  UNIT %D3%A  DROPPED,  NOT READY AT STARTUP%N/
275 023466      045      101      040  DEVDRD: .ASCIZ  /%A  UNIT %D3%A  DROPPED,  ERRORS = %D%N/
276
277
278 023536      ENDMOD
279
280

```

1
2
9
10 023536
 023536
16
24

.TITLE TSV7 - HARDWARE TESTS 1-8

TSV7:: BGNMOD TSV7

26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
73
74
75
76

```

.SBTTL TEST 1: INITIALIZE #4 TEST
:
: 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.)
:
: -
      BGNTST
      T1::
36 023536 012737 006356 002170      MOV      #EPRT1,EPRTSW      ;SET UP PRIMARY ERROR MESSAGE
:
: TEST 1
:
: -
49 023544 004737 016274      JSR      PC,DSBINT      ;DISABLE INTERRUPTS
50 023550 012700 024474      MOV      #TST21ID,R0    ;ASCII MESSAGE TO IDENTIFY TEST
51 023554 004737 016600      JSR      PC,TSTSETUP    ;DO INITIAL TEST SETUP
52 023560 012737 000005 002206      MOV      #5,LOOPCNT     ;PERFORM 5 ITERATIONS
53 023566      T21LOOP:
54 023566 004737 024516      JSR      PC,T21REST     ;SET COMMAND PACKET
55 023572 004737 024606      JSR      PC,T21RT2     ;SET UP OTHER COMMAND PACKET
:
: *****
: ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
: *****
63 023576 012737 176750 024152      11$:    MOV      #65000.,T21DLY  ;SET DELAY ROUTINE
64 023604 004737 016064      JSR      PC,SOFINIT     ;DO INITIALIZE ON CONTROLLER
65 023610 103426      BCS     20$             ;BR IF INIT WAS OK
66 023612      DELAY     250         ;DELAY FOR A REWIND TO FINISH
      MOV      #250,(PC)+
      .WORD   0
      MOV      L$DLY,(PC)+
      .WORD   0
      DEC     -6(PC)
      BNE     -4
      DEC     -22(PC)
      BNE     -20
67 023642 005337 024152      DEC     T21DLY         ;BUMP COUNTER DOWN
68 023646 001356      BNE     11$           ;BR, IF MORE TIME TO GO
69 023650 005237 002212      INC     FATFLG        ;BUMP COUNT
73 023654 010001      MOV     R0,R1         ;CONTENTS OF TSSR REGISTER
74 023656      ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      TRAP   C$ERDF
      .WORD  101
      .WORD  SFIERR
      .WORD  SFIMSG
75 023666      20$:
76 023666 012704 024130      MOV     #T21PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
  
```


024064 000151
024066 024453
024070 015564
130 024072 004737 017272
131 024076 000241
132 024100 106037 024251
133 024104 001316
134 024106
024106 104406
135 024110 004737 016546
136 024114 103002
137 024116 000137 023566
138 024122
024122 104432
024124 000524

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

.WORD 105
.WORD T21OFL
.WORD EXPREC
: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-

140
 141
 142
 144 024130
 146 024130 100004
 147 024130 024140
 148 024132 000000
 149 024134 000012
 150 024136 024154
 151 024140 000000
 152 024140 000024
 153 024142 000000
 154 024144 000000
 155 024146 000000
 156 024150 000000
 157 024152 000000
 158 024154
 159
 160
 161
 163 024240
 165 024240 100206
 166 024240 024250
 167 024242 000000
 168 024244 000006
 169 024246
 170
 171
 172 024250
 173 024250 000
 174 024251 000
 175 024252 000000
 176 024254 000000
 177
 178
 179

:+
 :LOCAL STORAGE FOR THIS TEST
 :-

.=<.+10>8177770

T21PACKET:

.WORD 100004
 .WORD T21DATA
 .WORD 0
 .WORD 10.

T21DATA:

.WORD T21BFR
 .WORD 0
 .WORD 20.

T21DSW: .WORD 0

T21DLY: .WORD 0

T21BFR: .BLKW 25.

:WRITE SUBSYSTEM MEMORY COMMAND PACKET

.=<.+10>8177770

T21PK2:

.WORD 100206
 .WORD T21BF2
 .WORD 0
 .WORD 6.

.EVEN

T21BF2:

T21BS0: .BYTE 0

T21BS1: .BYTE 0

T21S2: .WORD 0

T21S3: .WORD 0

: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

:DRIVE SELECT WORD
 :DELAY COUNTER
 :MESSAGE BUFFER

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

:SIZE OF DATA PACKET

:BSELO AREA --- "COMMAND" BYTE
 :BSEL1 AREA
 :SEL 2 AREA
 :DATA AREA

```

181
182
183      ;+
184      ;LOCAL TEXT MESSAGES FOR TEST
185      ;-
186 024256      127      122      111  T21SSR: .ASCIZ  'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
187 024353      124      123      123  T21AM3: .ASCIZ  'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
188 024453      104      162      151  T21OFL: .ASCIZ  'Drive is OFFLINE'
189 024474      111      156      151  T21ID:  .ASCIZ  'Initialization #4'
190
191
192
193      ;+
194      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
195      ;WRITE SUBSYSTEM MEMORY COMMAND
196      ;-
197
198 024516      T21REST:
199 024516      SAVREG      ;SAVE THE REGISTERS
200 024522      012701      024130      MOV      #T21PACKET,R1      ;START OF THE PACKET
201 024526      012721      100004      MOV      #100004,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,
202 024532      012721      024140      MOV      #T21DATA,(R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
203 024536      005021      CLR      (R1)+      ;EXTENDED ADDRESS
204 024540      012721      000010      MOV      #8,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
205 024544      012721      024154      MOV      #T21BFR,(R1)+      ;ADDRESS OF MESSAGE BUFFER
206 024550      005021      CLR      (R1)+
207 024552      012721      000024      MOV      #20,(R1)+      ;LENGTH OF MESSAGE BUFFER
208 024556      005021      CLR      (R1)+
209 024560      005011      CLR      (R1)
210 024562      012702      000020      MOV      #20,R2      ;NUMBER OF LOCATIONS TO BE CLEARED
211 024566      012762      177777      024154      64$:  MOV      #177777,T21BFR(R2)      ;ALL ONES TO MESSAGE BUFFER
212 024574      005742      TST      -(R2)      ;NEXT LOCATION
213 024576      020227      000000      CMP      R2,#0      ;CHECK R2 FOR ZERO
214 024602      001371      BNE      64$      ;BR, IF NOT AT ZERO YET
215 024604      000207      RTS      PC      ;RETURN
216
217
218 024606      T21RT2:
219 024606      SAVREG      ;SAVE THE REGISTERS
220 024612      012701      024240      MOV      #T21PK2,R1      ;START OF THE PACKET
221 024616      012721      100206      MOV      #100206,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
222 024622      012721      024250      MOV      #T21BF2,(R1)+      ;ADDRESS OF DATA BLOCK
223 024626      005021      CLR      (R1)+      ;EXTENDED ADDRESS
224 024630      012721      000006      MOV      #6,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
225 024634      005021      CLR      (R1)+
226 024636      012701      024250      MOV      #T21BF2,R1      ;ADDRESS OF DATA FOR WRT SUB SYS MEM
227 024642      005021      CLR      (R1)+
228 024644      005011      CLR      (R1)
229 024646      000207      RTS      PC      ;RETURN
230 024650      ENDTST
      024650
      024650      104401
      L10036:      TRAP      C$ETST
  
```


TSV7 - HARDWARE TESTS 1-8
TEST 1: INITIALIZE #4 TEST

MACRO M1113 25-MAY-82 09:19 PAGE 81

J 10

SEQ 0126

232


```

291
292
293
294
295
296
297
298 024744 004737 010752          JSR    PC,WRTCHR          :ISSUE WRITE CHARACTERISTICS
299 024750 103407          BCS    23$                :BR, IF COMMAND ISSUED OK
300 024752 005237 002212          INC    FATFLG            :BUMP COUNT
304 024756 010001          MOV    R0,R1             :SAVE CONTENTS OF TSSR
305 024760          ERRHRD  ERRNO,WRTMSG,SFIMSG :WRITE CHARACTERISTICS FAILED
      024760 104456          TRAP   C$ERHRD
      024762 000312          .WORD 202
      024764 005054          .WORD WRTMSG
      024766 012124          .WORD SFIMSG
306 024770          23$:  CKLOOP          TRAP   C$CLP1
      024770 104406          MOV    T22BFR+6,R1      :PICK UP XT50
307 024772 013701 026250          BIT    #4,R1            :IS UNIT WRITE-LOCKED?
308 024776 032701 000004          BEQ    24$                :NO, PROCEED WITH TESTING
309 025002 001407          INC    FATFLG            :BUMP COUNT
310 025004 005237 002212          ERRDF  ERRNO,T22WLK,SFIMSG :TAPE IS WRITE LOCKED
      025010 104455          TRAP   C$ERDF
      025012 000313          .WORD 203
      025014 027052          .WORD T22WLK
      025016 012124          .WORD SFIMSG
315 025020          DOCLN          TRAP   C$DCLN
      025020 104444          24$:  CKLOOP          TRAP   C$CLP1
316 025022          24$:  CKLOOP          TRAP   C$CLP1
      025022 104406          TST    EXTFEA           :CHECK FOR EXTENDED FEATURES SW SWITCH
317 025024 005737 002216          BNE    50$                :BR IF SWITCH IS ON
318 025030 001041          MOV    #200,T22BS1      :WRITE MISCELLANEOUS CONT/READ STATUS
319 025032 112737 000200 026341  MOV    #10,T22BS0      :FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
320 025040 112737 000010 026340  MOV    #T22PK2,R4      :WRITE SUBSYS MEM PACKET
321 025046 012704 026330          MOV    R4,TSDB(R5)     :ISSUE COMMAND
322 025052 010465 000000          JSR    PC,CHKTSSR      :WAIT FOR SSR
323 025056 004737 016426          BCS    30$                :BR, IF NO ERROR
324 025062 103407          MOV    R0,R1             :ERROR, SAVE TSSR
325 025064 010001          INC    FATFLG            :BUMP COUNT
326 025066 005237 002212          ERRHRD  ERRNO,T22SSR,PKTSSR :TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
330 025072          TRAP   C$ERHRD
      025072 104456          .WORD 204
      025074 000314          .WORD T22SSR
      025076 026360          .WORD PKTSSR
      025100 012136          30$:  CKLOOP          :LOOP IF SELECTED
331 025102          TRAP   C$CLP1
      025102 104406          MOV    #T22PACKET,R4   :SUBROUTINE NEEDS PACKET ADDRESS
      025104 012704 026220
332 025104 012704 026220          MOV    #T22PACKET,R4
333
334
335
336
337
338
339
340 025110 004737 010752          JSR    PC,WRTCHR          :ISSUE WRITE CHARACTERISTICS
    
```



```

384
385
386          :+
387          :TEST 2, SUBTEST 2
388          :VERIFIES THAT A REWIND COMMAND, WITH THE CLEAR VOLUME
389          :CHECK (CVC) BIT CLEAR, IS REJECTED IF THE VOLUME CHECK
390          :(VCK) FLAG IS SET.
391
392
393
394 025274          BGNSUB           ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>>
              025274                   T2.2:
              025274          104402                   TRAP       C$BSUB
395 025276          004737          027172             JSR      PC,T22REST      ;SET COMMAND PACKET
396 025302          004737          027264             JSR      PC,T22RT2      ;SET UP OTHER COMMAND PACKET
397
398          :*****
399          :ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
400          :*****
401
402
403
404 025306          004737          016064             JSR      PC,SOFINIT     ;DO INITIALIZE ON CONTROLLER
405 025312          103407                   BCS      20$            ;BR IF INIT WAS OK
406 025314          005237          002212             INC      FATFLG         ;BUMP COUNT
410 025320          010001                   MOV      R0,R1          ;CONTENTS OF TSSR REGISTER
411 025322          104455                   ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
              025322          000320                                   TRAP     C$ERDF
              025324          003650                                   .WORD   208
              025326          012124                                   .WORD   SFIERR
              025330          012124                                   .WORD   SFIMSG
412 025332          20$:
413 025332          012704          026220             MOV      #T22PACKET,R4  ;SUBROUTINE NEEDS PACKET ADDRESS
414
415          :*****
416          :WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
417          :*****
418
419
420
421 025336          004737          010752             JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
422 025342          103407                   BCS      23$            ;BR, IF COMMAND ISSUED OK
423 025344          005237          002212             INC      FATFLG         ;BUMP COUNT
427 025350          010001                   MOV      R0,R1          ;SAVE CONTENTS OF TSSR
428 025352          104456                   ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
              025352          000321                                   TRAP     C$ERHRD
              025354          005054                                   .WORD   209
              025356          012124                                   .WORD   WRTMSG
              025360          012124                                   .WORD   SFIMSG
429 025362          005737          002216             23$:  TST      EXTFEA
430 025366          001041                   BNE     50$            ;CHECK FOR EXTENDED FEATURES SW SWITCH
431
432 025370          112737          000200          026341             MOVB    #200,T22BS1    ;WRITE MISCELLANEOUS CONT/READ STATUS
433 025376          112737          000010          026340             MOVB    #10,T22BS0     ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
434 025404          012704          026330             MOV     #T22PK2,R4     ;WRITE SUBSYS MEM PACKET
435 025410          010465          000C00             MOV     R4,TSDB(R5)    ;ISSUE COMMAND
436 025414          004737          016426             JSR     PC,CHKTSSR     ;WAIT FOR SSR
    
```


025600
025600 104403
486 025602 023727 002212 000017
487 025610 103402
488 025612 004737 017272
489 025616

999\$:

CMP FATFLG,#15.
BLO 999\$
JSR PC,CKDROP

L10041: TRAP CSESUB
:IS ERROR COUNT AT 25
:BR, IF LESS THAN 25
:TRY TO DROP THE UNIT

491
492
493
494
495
496
497
498
499

:+
:TEST 2, SUBTEST 3
:VERIFIES THAT A REWIND COMMAND WITH CVC=1 CLEARS VCK
:AND RETURNS PROPER STATUS IN THE MESSAGE BUFFER.
:~
:~
:~

```

500 025616             BGNSUB                       ;>>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>>>>
      025616             T2.3:
      025616 104402     TRAP   CSBSUB
501 025620 004737 027172 JSR   PC,T22REST      ;SET COMMAND PACKET
502 025624 004737 027264 JSR   PC,T22RT2      ;SET UP OTHER COMMAND PACKET
503
504 *****
505 :ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
506 *****
507
508
509 JSR   PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
510 025630 004737 016064 BCS   20$              ;BR IF INIT WAS OK
511 025634 103407                    ;BUMP COUNT
512 025636 005237 002212 INC   FATFLG
516 025642 010001                    ;CONTENTS OF TSSR REGISTER
517 025644             ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      025644 104455     TRAP   CSERDF
      025646 000326     .WORD  214
      025650 003650     .WORD  SFIERR
      025652 012124     .WORD  SFIMSG
518 025654             20$:
519 025654 012704 026220 MOV   #T22PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
520
521 *****
522 :WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
523 *****
524
525
526 JSR   PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
527 025660 004737 010752 BCS   23$              ;BR, IF COMMAND ISSUED OK
528 025664 103407                    ;BUMP COUNT
529 025666 005237 002212 INC   FATFLG
533 025672 010001                    ;SAVE CONTENTS OF TSSR
534 025674             ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      025674 104456     TRAP   CSERHRD
      025676 000327     .WORD  215
      025700 005054     .WORD  WRTMSG
      025702 012124     .WORD  SFIMSG
535 025704 005737 002216 23$: TST   EXTFEA      ;CHECK FOR EXTENDED FEATURES SW SWITCH
536 025710 001041 BNE   50$              ;BR IF SWITCH IS ON
537
538 025712 112737 000200 026341 MOVB  #200,T22BS1      ;WRITE MISCELLANEOUS CONT/READ STATUS
539 025720 112737 000010 026340 MOVB  #10,T22BS0      ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
540 025726 012704 026330 MOV   #T22PK2,R4      ;WRITE SUBSYS MEM PACKET
541 025732 010465 000000 MOV   R4,T22SDB(R5)  ;ISSUE COMMAND
542 025736 004737 016426 JSR   PC,CHKTSSR     ;WAIT FOR SSR
543 025742 103407 BCS   30$              ;BR, IF NO ERROR

```



```

544 025744 010001          MOV    R0,R1          ;ERROR, SAVE TSSR
545 025746 005237 002212  INC    FATFLG        ;BUMP COUNT
549 025752          ERRHRD  ERRNO,T22SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
    025752 104456          TRAP  C$ERHRD
    025754 000330          .WORD 216
    025756 026360          .WORD T22SSR
    025760 012136          .WORD PKTSSR
550 025762          30$:  CKLOOP          ;LOOP IF SELECTED
    025762 104406          TRAP  C$CLP1
551 025764 012704 026220  MOV    #T22PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
552
553  :*****
554  :WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
555  :*****
556
557
558
559 025770 004737 010752          JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
560 025774 103407          BCS   50$            ;BR, IF COMMAND ISSUED OK
561 025776 005237 002212  INC    FATFLG        ;BUMP COUNT
565 026002 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
566 026004          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
    026004 104456          TRAP  C$ERHRD
    026006 000331          .WORD 217
    026010 005054          .WORD WRTMSG
    026012 012124          .WORD SFIMSG
567 026014          50$:  CKLOOP          ;SCOPE LOOP
    026014 104406          TRAP  C$CLP1
568 026016 016501 000002  MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
569 026022 032701 000100  BIT   #OFL,R1        ;CHECK FOR THE OFFLINE BIT SET
570 026026 001006          BNE   60$            ;BR, IF OFFLINE (GOOD)
571 026030 005237 002212  INC    FATFLG        ;BUMP COUNT
575 026034          ERRDF  ERRNO,T22OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
    026034 104455          TRAP  C$ERDF
    026036 000332          .WORD 218
    026040 026555          .WORD T22OFL
    026042 012124          .WORD SFIMSG
576 026044          60$:  CKLOOP          ;LOOP IF SELECTED
    026044          TRAP  C$CLP1
577 026046 012737 142010 026330 65$:  MOV    #142010,T22PK2 ;POSITION COMMAND (REWIND MODE)
578 026054 012704 026330  MOV    #T22PK2,R4    ;R4 = POINTER TO PACKET
579 026060 010465 000000  MOV    R4,TSDB(R5)   ;ISSUE COMMAND
580 026064 004737 016340  JSR    PC,WAITF      ;WAIT FOR SSR TO SET
581 026070 016501 000002  MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
582 026074 012702 100306  MOV    #SSR!OFL!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
583 026100 020102          CMP   R1,R2         ;ARE THEY EQUAL
584 026102 001406          BEQ   80$            ;BR, IF OK ESP. FUNCTION REJECT
585 026104 005237 002212  INC    FATFLG        ;BUMP COUNT
589 026110          ERRHRD  ERRNO,T22RWJ,EXPREC ;TSSR INCORRECT AFTER TAPE MOTION CMD
    026110 104456          TRAP  C$ERHRD
    026112 000333          .WORD 219
    026114 026724          .WORD T22RWJ
    026116 015564          .WORD EXPREC
590 026120          80$:  CKLOOP          ;LOOP IF SELECTED
    026120 104406          TRAP  C$CLP1
591 026122 012703 026242  MOV    #T22BFR,R3    ;POINTER TO MESSAGE BUFFER
592 026126 016301 000006  MOV    XST0(R3),R1   ;PICK UP XST0 FROM MESSAGE BUFFER
    
```


613
 614
 615
 617 026220
 619 026220
 620 026220 100204
 621 026222 026230
 622 026224 000000
 623 026226 000012
 624 026230
 625 026230 026242
 626 026232 000000
 627 026234 000024
 628 026236 000000
 629 026240 000007
 630 026242
 631
 632
 633
 635 026330
 637 026330
 638 026330 100206
 639 026332 026340
 640 026334 000000
 641 026336 000006
 642
 643
 644 026340
 645 026340 000
 646 026341 000
 647 026342 000000
 648 026344 000000
 649
 650
 651
 652
 653 026346 100201
 654 026350 100205
 655 026352 100210
 656 026354 100211
 657 026356 177777
 658
 659

```

: +
: 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
  
```

```

661
662
663      ;+
664      ;LOCAL TEXT MESSAGES FOR TEST
665      :-
666 026360      127      122      111 T22SSR: .ASCIZ 'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
667 026455      124      123      123 T22AM3: .ASCIZ 'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
668 026555      104      162      151 T22OFL: .ASCIZ 'Drive 7 Select Failed To Set 'OFL' In TSSR'
669 026630      124      123      123 T22TM: .ASCIZ 'TSSR Incorrect After Tape Motion Command To Off-Line Device'
670 026724      124      123      123 T22RWJ: .ASCIZ 'TSSR Not Correct After REWIND With VCK Set'
671 026777      103      126      103 T22VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
672 027052      052      052      052 T22WLK: .ASCIZ '*****TAPE IS WRITE-LOCKED AND WILL CAUSE ERRORS*****'
673 027137      117      146      146 T22ID: .ASCIZ 'Off-Line And Reject Rewind'
674
675
676
677      ;+
678      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
679      ;WRITE SUBSYSTEM MEMORY COMMAND
680      :-
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
    
```

```

T22REST:
    SAVREG
    MOV #T22PACKET,R1 ;SAVE THE REGISTERS
    MOV #100204,(R1)+ ;START OF THE PACKET
    MOV #T22DATA,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK, IE
    CLR (R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
    MOV #10,(R1)+ ;EXTENDED ADDRESS
    MOV #T22BFR,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
    CLR (R1)+ ;ADDRESS OF MESSAGE BUFFER
    MOV #20,(R1)+ ;LENGTH OF MESSAGE BUFFER
    CLR (R1)+
    MOV #7,(R1) ;SELECT DRIVE SEVEN
    MOV #20,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
    MOV #177777,T22BFR(R2) ;ALL ONES TO MESSAGE BUFFER
    TST -(R2) ;BUMP R2 DOWN
    CMP R2,#0 ;IS R2 AT ZERO YET
    BNE 64$ ;KEEP GOING UNTIL DONE
    RTS PC ;RETURN
    
```

```

T22RT2:
    SAVREG
    MOV #T22PK2,R1 ;SAVE THE REGISTERS
    MOV #100206,(R1)+ ;START OF THE PACKET
    MOV #T22BF2,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK, IE
    CLR (R1)+ ;ADDRESS OF DATA BLOCK
    MOV #6,(R1)+ ;EXTENDED ADDRESS
    CLR (R1)+ ;SIZE OF DATA BLOCK IN BYTES
    MOV #T22BF2,R1 ;POINT TO DATA SEL AREA
    CLR (R1)+
    CLR (R1)
    CLR (R1)
    RTS PC ;LAST LOC TO BE CLEARED
    ENDTST ;RETURN
    
```

```

702 027264
703 027264
704 027270 012701 026330
705 027274 012721 100206
706 027300 012721 026340
707 027304 005021
708 027306 012721 000006
709 027312 005021
710 027314 012701 026340
711 027320 005021
712 027322 005011
713 027324 005011
714 027326 000207
715 027330
    027330
    027330 104401
    
```

L10037: TRAP C\$ETST

+ 716


```

775 027434
776 027434 012737 000007 032460 20$: MOV #7,T23DSW ;SET DRIVE NUMBER IN PACKET
777 027442 012704 032440 MOV #T23PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
778
779
780 :*****
781 :WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
782 :*****
783
784
785 027446 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
786 027452 103407 BCS 23$ ;BR, IF COMMAND ISSUED OK
787 027454 005237 002212 INC FATFLG ;BUMP COUNT
791 027460 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
792 027462 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
027462 104456 TRAP C$ERHRD
027464 000456 .WORD 302
027466 005054 .WORD WRTMSG
027470 012124 .WORD SFIMSG
793 027472 005737 002216 23$: TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
794 027476 001044 BNE 50$ ;BR IF SWITCH IS ON
795
796 027500 112737 000200 032573 MOVB #200,T23BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
797 027506 112737 000010 032572 MOVB #10,T23BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
798 027514 012704 032550 MOV #T23PK2,R4 ;WRITE SUBSYS MEM PACKET
799 027520 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
800 027524 004737 016426 JSR PC,CHKTSSR ;WAIT FOR SSR
801 027530 103407 BCS 30$ ;BR, IF NO ERROR
802 027532 010001 MOV RO,R1 ;ERROR, SAVE TSSR
803 027534 005237 002212 INC FATFLG ;BUMP COUNT
807 027540 ERRHRD ERRNO,T23SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
027540 104456 TRAP C$ERHRD
027542 000457 .WORD 303
027544 032614 .WORD T23SSR
027546 012136 .WORD PKTSSR
808 027550 30$: CKLOOP ;LOOP IF SELECTED
027550 104406 TRAP C$CLP1
809 027552 012737 000007 032460 MOV #7,T23DSW ;SET DRIVE NUMBER IN PACKET
810 027560 012704 032440 MOV #T23PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
811
812 :*****
813 :WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
814 :*****
815
816
817
818 027564 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
819 027570 103407 BCS 50$ ;BR, IF COMMAND ISSUED OK
820 027572 005237 002212 INC FATFLG ;BUMP COUNT
824 027576 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
825 027600 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
027600 104456 TRAP C$ERHRD
027602 000460 .WORD 304
027604 005054 .WORD WRTMSG
027606 012124 .WORD SFIMSG
826 027610 50$: CKLOOP ;SCOPE LOOP
027610 104406 TRAP C$CLP1
  
```



```

916 030044 103407          BCS      23$          ;BR, IF COMMAND ISSUED OK
917 030046 005237 002212  INC      FATFLG      ;BUMP COUNT
921 030052 010001          MOV      R0,R1       ;SAVE CONTENTS OF TSSR
922 030054          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
          030054 104456          TRAP      C$ERHRD
          030056 000464          .WORD     308
          030060 005054          .WORD     WRTMSG
          030062 012124          .WORD     SFIMSG
923 030064          23$:   CKLOOP          ;LOOP IF SELECTED
          030064 104406          TRAP      C$CLP1
924
925
926
927
928
929
930
931 030066 004737 011104          JSR      PC,REWIND    ;CALL THE TAPE REWIND
932 030072 012703 000024          MOV      #20.,R3     ;STARTING RECORD SIZE
933 030076 013737 003114 032562 65$:   MOV      FREE,T23WB  ;STARTING WRITE BUFFER ADDRESS
934
935
936
937
938
939
940
941 030104 012737 140005 032560          MOV      #140005,T23PK3 ;WRITE DATA,CVC=1,ACK COMMAND
942 030112 012737 140005 032602          MOV      #140005,T23WRT ;SETUP FOR RETRY COMMAND
943 030120 052737 004000 032602          BIS      #4000,T23WRT  ;MAKE IT A RETRY
944 030126 012704 032560          MOV      #T23PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
945 030132 010300          MOV      R3,R0       ;SET PATTERN IN CORRECT REGISTER
946 030134 004737 017512          JSR      PC,FILLMEM   ;FILL MEMORY WITH RECORD SIZE
947 030140 010337 032566          MOV      R3,T23SZ    ;SET UP RECORD SIZE IN PACKET
948 030144 010465 000000          MOV      R4,TSDB(R5) ;ISSUE COMMAND
949 030150 004737 016340          JSR      PC,WAITF     ;WAIT FOR SSR TO SET
950 030154 016501 000002          MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
951 030160 012702 000200          MOV      #SSR,R2     ;SET UP EXPECTED
952 030164 020102          CMP      R1,R2       ;ARE THEY EQUAL
953 030166 001402          BEQ      80$         ;BR, IF OK
954 030170 004737 034132          JSR      PC,T23CHK   ;CHECK SPECIAL CONDITION
955 030174          80$:   CKLOOP          ;LOOP IF SELECTED
          030174 104406          TRAP      C$CLP1
956 030176 016501 000000          MOV      TSBA(R5),R1  ;GET TSBA CONTENTS
957 030202 012702 032462          MOV      #T23BFR,R2  ;SET UP EXPECTED
958 030206 062702 000016          ADD      #16,R2      ;SET TO END OF MESSAGE BUFFER
959 030212 005737 002216          TST      EXTFEA     ;CHECK FOR EXTENDED FEATURES SW SET
960 030216 001402          BEQ      85$         ;BR, IF IT NOT SET
961 030220 062702 000002          ADD      #2,R2       ;BUMP R2 FOR EXTRA DATA
962 030224 020102          85$:   CMP      R1,R2   ;ARE THEY EQUAL
963 030226 001406          BEQ      90$         ;BR, IF TSBA IS CORRECT
964 030230 005237 002212          INC      FATFLG     ;BUMP COUNT
968 030234          ERRHRD  ERRNO,T23BA,EXPREC ;TSBA WAS NOT CORRECT AFTER WRITE DATA
          030234 104456          TRAP      C$ERHRD
          030236 000465          .WORD     309
          030240 033575          .WORD     T23BA
          030242 015564          .WORD     EXPREC

```

```

969 030244          90$:  CKLOOP          :LOOP IF SELECTED
      030244 104406          TRAP  C$CLP1
970 030246 020327 007376      CMP    R3,#7376      :ONLY CHECK RAM UNTIL ITS FULL
971 030252 002114          BGE    115$          :IT WRAPS AROUND ETC.
972 030254 004737 034044      JSR    PC,T23RT2     :MAKE SURE PACKET AND DATA ARE CLEAN
973 030260 012737 000400 032574      MOV    #256,T23S2   :STARTING RAM ADDRESS
974 030266 112737 000000 032572      MOVB  #0,T23BS0     :STOP INTERNAL TSV05 DIAGNOSTICS
975 030274 112737 000000 032573      MOVB  #0,T23BS1     :SIZE OF RAM READ
976 030302 012704 032550      MOV    #T23PK2,R4   :SET R4 WITH PACKET ADDRESS
977 030306 010465 000000      MOV    R4,TSDB(R5)  :ISSUE WRITE SUB SYS MEM COMMAND
978 030312 004737 016426      JSR    PC,CHKTSSR   :CHECK TSSR AND WAIT FOR SSR TO SET
979 030316 103407          BCS    92$          :BR, IF NO ERRORS IN TSSR
980 030320 010001          MOV    R0,R1        :SAVE TSSR
981 030322 005237 002212      INC    FATFLG       :BUMP COUNT
985 030326          ERRHRD  ERRNO,T23WSS,PKTSSR :TSSR BAD AFTER WRITE SUB SYS MEM
      030326 104456          TRAP  C$ERHRD
      030330 000466          .WORD 310
      030332 033647          .WORD T23WSS
      030334 012136          .WORD PKTSSR

986 030336          92$:  CKLOOP          :LOOP IF SELECTED
      030336 104406          TRAP  C$CLP1
987 030340 004737 034044      JSR    PC,T23RT2     :MAKE SURE PACKET AND DATA ARE CLEAN
988 030344 012737 000400 032574      MOV    #256,T23S2   :STARTING RAM ADDRESS
989 030352 112737 000001 032572      MOVB  #1,T23BS0     :READ RAM COMMAND FOR WRITE SUB SYS M.
990 030360 112737 000002 032573      MOVB  #2,T23BS1     :SIZE OF RAM READ
991 030366 012704 032550      MOV    #T23PK2,R4   :SET R4 WITH PACKET ADDRESS
992 030372 010465 000000      MOV    R4,TSDB(R5)  :ISSUE WRITE SUB SYS MEM COMMAND
993 030376 004737 016426      JSR    PC,CHKTSSR   :CHECK TSSR AND WAIT FOR SSR TO SET
994 030402 103407          BCS    100$        :BR, IF NO ERRORS IN TSSR
995 030404 010001          MOV    R0,R1        :SAVE TSSR
996 030406 005237 002212      INC    FATFLG       :BUMP COUNT
1000 030412          ERRHRD  ERRNO,T23WSS,PKTSSR :TSSR BAD AFTER WRITE SUB SYS MEM
      030412 104456          TRAP  C$ERHRD
      030414 000467          .WORD 311
      030416 033647          .WORD T23WSS
      030420 012136          .WORD PKTSSR

1001 030422          100$: CKLOOP          :LOOP IF SELECTED
      030422 104406          TRAP  C$CLP1
1002 030424 005001          CLR    R1           :CLEAR REGISTER
1003 030426 005002          CLR    R2           :CLEAR REGISTER
1004 030430 013701 032502      MOV    T23BFR+20,R1 :PICK UP BYTE READ FROM RAM
1005 030434 010302          MOV    R3,R2        :SET UP EXPECTED
1006 030436 020102          CMP    R1,R2        :IS RAM DATA CORRECT
1007 030440 001406          BEQ    110$        :BR, IF OK (EQUAL)
1008 030442 005237 002212      INC    FATFLG       :BUMP COUNT
1012 030446          ERRHRD  ERRNO,T23RNC,EXPREC :RNC=RAM NOT CORRECT
      030446 104456          TRAP  C$ERHRD
      030450 000470          .WORD 312
      030452 033135          .WORD T23RNC
      030454 015564          .WORD EXPREC

1013 030456          110$: CKLOOP          :LOOP IF SELECTED
      030456 104406          TRAP  C$CLP1
1014 030460 005237 032574      INC    T23S2        :BUMP RAM ADDRESS TO BE CHECKED
1015 030464 005237 032574      INC    T23S2        :BUMP RAM ADDRESS TO BE CHECKED
1016 030470 010301          MOV    R3,R1        :GET SIZE OF RECORD
1017 030472 062701 000400      ADD    #256,R1      :FIGURE OUT END RECORD ADDRESS
1018 030476 023701 032574      CMP    T23S2,R1     :AT END OF RAM CHECK YET

```

1019	030502	001333					BNE	95\$:BR, IF MORE TO CHECK
1020	030504	062703	001750			115\$:	ADD	#1000, R3		:NEXT RECORD SIZE/DATA PATTERN
1021	030510	020337	032570				CMP	R3, T23RSZ		:IS R3 OVER MAX RECORD SIZE
1022	030514	002005					BGE	120\$:IF RECORD SIZE IS TOO BIG QUIT
1023	030516	020327	177776				CMP	R3, #65534.		:END OF SUBTEST MAX RECORD SIZE
1024	030522	001402					BEQ	120\$:BR, IF COMPLETED
1025	030524	000137	030076				JMP	65\$:DO MORE RECORDS
1026	030530					120\$:				
1027	030530	004737	034106				JSR	PC, T23RT3		:RESTORE PACKET
1028	030534	004737	034044				JSR	PC, T23RT2		:CLEAN UP PACKET
1029	030540	012737	102010	032550			MOV	#102010, T23PK2		:REWIND (POSITION) COMMAND
1030	030546	012704	032550				MOV	#T23PK2, R4		:LOAD R4 WITH PACKET ADDRESS
1031	030552	010465	000000				MOV	R4, TSDB(R5)		:ISSUE REWIND COMMAND
1032	030556	004737	016426				JSR	PC, CHKTSSR		:WAIT FOR SSR TO SET
1033	030562	103407					BCS	130\$:BR, IF T.JR IS OK (GOOD)
1034	030564	010001					MOV	R0, R1		:SAVE TSSR CONTENTS
1035	030566	005237	002212				INC	FATFLG		:BUMP COUNT
1039	030572						ERRHRD	ERRNO, T23RWN, PKTSSR		:TSSR IS INCORRECT AFTER REWIND
	030572	104456								TRAP C\$ER4RD
	030574	000471								.WORD 313
	030576	033066								.WORD T23RWN
	030600	012136								.WORD PKTSSR
1040	030602					130\$:				
1041	030602						ENDSUB			:>>>>>>>>>> END SUBTEST >>>>>>>>>>
	030602									L10045:
	030602	104403								TRAP C\$ESUB
1042	030604	023727	002212	000017			CMP	FATFLG, #15.		:IS ERROR COUNT AT 25
1043	030612	103402					BLO	999\$:BR, IF LESS THAN 25
1044	030614	004737	017272				JSR	PC, CKDROP		:TRY TO DROP THE UNIT
1045	030620					999\$:				


```

1100
1101      :WRITE DATA,CVC=1,ACK,SWB COMMAND
1102      :*****
1103
1104
1105 030726 012737 150005 032560      MOV      #150005,T23PK3      :WRITE DATA,CVC=1,ACK,SWB COMMAND
1106 030734 012737 150005 032602      MOV      #150005,T23WRT    :SETUP FOR RETRY COMMAND
1107 030742 052737 004000 032602      BIS      #4000,T23WRT     :MAKE IT A RETRY
1108 030750 012704 032560      MOV      #T23PK3,R4      :SET UP R4 WITH PACKET ADDRESS
1109 030754 010300      MOV      R3,R0           :SET PATTERN IN CORRECT REGISTER
1110 030756 004737 017512      JSR      PC,FILLMEM      :FILL MEMORY WITH RECORD SIZE
1111 030762 010337 032566      MOV      R3,T23SZ        :SET UP RECORD SIZE IN PACKET
1112 030766 010465 000000      MOV      R4,TSDB(R5)     :ISSUE COMMAND
1113 030772 004737 016340      JSR      PC,WAITF        :WAIT FOR SSR TO SET
1114 030776 016501 000002      MOV      TSSR(R5),R1     :GET TSSR CONTENTS
1115 031002 012702 000200      MOV      #SSR,R2        :SET UP EXPECTED
1116 031006 020102      CMP      R1,R2           :ARE THEY EQUAL
1117 031010 001402      BEQ      80$             :BR, IF OK
1118 031012 004737 034132      JSR      PC,T23CHK       :CHECK SPECIAL CONDITION
1119 031016      80$:      CKLOOP              :LOOP IF SELECTED
1120 031020 016501 000000      MOV      TSBA(R5),R1     :GET TSBA CONTENTS      TRAP      C$CLP1
1121 031024 012702 032462      MOV      #T23BFR,R2     :SET UP EXPECTED
1122 031030 062707 000016      ADD      #16,R2         :SET TO END OF MESSAGE BUFFER
1123 031034 005737 002216      TST      EXTFEA         :CHECK FOR EXTENDED FEATURES SW SET
1124 031040 001402      BEQ      85$             :BR, IF IT NOT SET
1125 031042 062702 000002      ADD      #2,R2          :BUMP R2 FOR EXTRA DATA
1126 031046 020102      85$:      CMP      R1,R2         :ARE THEY EQUAL
1127 031050 001406      BEQ      90$             :BR, IF TSBA IS CORRECT
1128 031052 005237 002212      INC      FATFLG         :BUMP COUNT
1132 031056      ERRHRD  ERRNO,T23BA,EXPREC :TSBA WAS NOT CORRECT AFTER WRITE DATA
1133 031056 104456      TRAP      C$ERHRD
1134 031060 000474      .WORD    316
1135 031062 033575      .WORD    T23BA
1136 031064 015564      .WORD    EXPREC
1137 031066      90$:      CKLOOP              :LOOP IF SELECTED
1138 031070 020327 007376      CMP      R3,#7376      :ONLY CHECK RAM UNTIL ITS FULL      TRAP      C$CLP1
1139 031074 002115      BGE      115$           :IT WRAPS AROUND ETC.
1140 031076 004737 034044      JSR      PC,T23RT2     :MAKE SURE PACKET AND DATA ARE CLEAN
1141 031102 012737 000400 032574      MOV      #256,T23S2    :STARTING RAM ADDRESS
1142 031110 112737 000000 032572      MOV      #0,T23BS0     :STOP INTERNAL TSV05 DIAGNOSTICS
1143 031116 112737 000000 032573      MOV      #0,T23BS1     :SIZE OF RAM READ
1144 031124 012704 032550      MOV      #T23PK2,R4    :SET R4 WITH PACKET ADDRESS
1145 031130 010465 000000      MOV      R4,TSDB(R5)   :ISSUE WRITE SUB SYS MEM COMMAND
1146 031134 004737 016426      JSR      PC,CHKTSSR    :CHECK TSSR AND WAIT FOR SSR TO SET
1147 031140 103407      BCS      92$           :BR, IF NO ERRORS IN TSSR
1148 031142 010001      MOV      R0,R1         :SAVE TSSR
1149 031144 005237 002212      INC      FATFLG         :BUMP COUNT
1150 031150      ERRHRD  ERRNO,T23WSS,PKTSSR :TSSR BAD AFTER WRITE SUB SYS MEM
1151 031150 104456      TRAP      C$ERHRD
1152 031152 000475      .WORD    317
1153 031154 033647      .WORD    T23WSS
1154 031156 012136      .WORD    PKTSSR
1155 031160      92$:      CKLOOP              :LOOP IF SELECTED
1156 031160 104406      TRAP      C$CLP1
1157 031162 004737 034044      JSR      PC,T23PT2     :MAKE SURE PACKET AND DATA ARE CLEAN

```

Line	Address	Offset	Value	Label	Instruction	Comment
1152	031166	012737	000400	032574	MOV	#256., T23S2 ; STARTING RAM ADDRESS
1153	031174	112737	000001	032572	MOVB	#1, T23BS0 ; READ RAM COMMAND FOR WRITE SUB SYS M.
1154	031202	112737	000002	032573	MOVB	#2, T23BS1 ; SIZE OF RAM READ
1155	031210	012704	032550		MOV	#T23PK2, R4 ; SET R4 WITH PACKET ADDRESS
1156	031214	010465	000000	95\$:	MOV	R4, TSDB(R5) ; ISSUE WRITE SUB SYS MEM CMD (READ RAM)
1157	031220	004737	016426		JSR	PC, CHKTSSR ; CHECK TSSR AND WAIT FOR SSR TO SET
1158	031224	103407			BCS	100\$; BR, IF NO ERRORS IN TSSR
1159	031226	010001			MOV	RO, R1 ; SAVE TSSR
1160	031230	005237	002212		INC	FATFLG ; BUMP COUNT
1164	031234				ERRHRD	ERRNO, T23WSS, PKTSSR ; TSSR BAD AFTER WRITE SUB SYS MEM
	031234	104456				TRAP C\$ERHRD
	031236	000476				.WORD 318
	031240	033647				.WORD T23WSS
	031242	012136				.WORD PKTSSR
1165	031244			100\$:	CKLOOP	; LOOP IF SELECTED
	031244	104406				TRAP C\$CLP1
1166	031246	005001			CLR	R1 ; CLEAR REGISTERS
1167	031250	005002			CLR	R2 ; CLEAR REGISTERS
1168	031252	013701	032502		MOV	T23BFR+20, R1 ; PICK UP BYTE READ FROM RAM
1169	031256	010302			MOV	R3, R2 ; SET UP EXPECTED
1170	031260	000302			SWAB	R2 ; SWAP BYTES
1171	031262	020102			CMP	R1, R2 ; IS RAM DATA CORRECT
1172	031264	001406			BEQ	110\$; BR, IF OK (EQUAL)
1173	031266	005237	002212		INC	FATFLG ; BUMP COUNT
1177	031272				ERRHRD	ERRNO, T23RNC, EXPREC ; RNC=RAM NOT CORRECT
	031272	104456				TRAP C\$ERHRD
	031274	000477				.WORD 319
	031276	033135				.WORD T23RNC
	031300	015564				.WORD EXPREC
1178	031302			110\$:	CKLOOP	; LOOP IF SELECTED
	031302	104406				TRAP C\$CLP1
1179	031304	005237	032574		INC	T23S2 ; BUMP RAM ADDRESS TO BE CHECKED
1180	031310	005237	032574		INC	T23S2 ; BUMP RAM ADDRESS TO BE CHECKED
1181	031314	010301			MOV	R3, R1 ; GET SIZE OF RECORD
1182	031316	062701	000400		ADD	#256., R1 ; FIGURE OUT END RECORD ADDRESS
1183	031322	023701	032574		CMP	T23S2, R1 ; AT END OF RAM CHECK YET
1184	031326	001332			BNE	95\$; BR, IF MORE TO CHECK
1185	031330	062703	001750	115\$:	ADD	#1000., R3 ; NEXT RECORD SIZE/DATA PATTERN
1186	031334	020337	032570		CMP	R3, T23RSZ ; IS R3 OVER MAX RECORD SIZE
1187	031340	002005			BGE	120\$; IF RECORD SIZE IS TOO BIG QUIT
1188	031342	020327	177776		CMP	R3, #65534. ; END OF SUBTEST MAX RECORD SIZE
1189	031346	001402			BEQ	120\$; BR, IF COMPLETED
1190	031350	000137	030720		JMP	65\$; DO MORE RECORDS
1191	031354			120\$:		
1192	031354	004737	034044		JSR	PC, T23RT2 ; CLEAN UP PACKET
1193	031360	012737	102010	032550	MOV	#102010, T23PK2 ; REWIND (POSITION) COMMAND
1194	031366	012704	032550		MOV	#T23PK2, R4 ; LOAD R4 WITH PACKET ADDRESS
1195	031372	010465	000000		MOV	R4, TSDB(R5) ; ISSUE REWIND COMMAND
1196	031376	004737	016426		JSR	PC, CHKTSSR ; WAIT FOR SSR TO SET
1197	031402	103407			BCS	130\$; BR, IF TSSR IS OK (GOOD)
1198	031404	010001			MOV	RO, R1 ; SAVE TSSR CONTENTS
1199	031406	005237	002212		INC	FATFLG ; BUMP COUNT
1203	031412				ERRHRD	ERRNO, T23RWN, PKTSSR ; TSSR IS INCORRECT AFTER REWIND
	031412	104456				TRAP C\$ERHRD
	031414	000500				.WORD 320
	031416	033066				.WORD T23RWN
	031420	012136				.WORD PKTSSR

1204 031422
1205 031422

130\$:

ENDSUB

; >>>>>>>>>> END SUBTEST >>>>>>>>>>
L10046:

031422 104403
1206 031424 023727 002212 000017
1207 031432 103402
1208 031434 004737 017272
1209 031440

999\$:

CMP FATFLG.#15.
BLO 999\$
JSR PC,CKDROP

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


```
1264  
1265  
1266 031540 012737 104405 032560 23$:   MOV     #104405,T23PK3          :WRITE DATA, ACK, ILLEGAL BITS  
1267 031546 013737 003114 032562     MOV     FREE,T23WB           :SET UP WRITE BUFFER ADDRESS  
1268 031554 062737 000001 032562     ADD     #1,T23WB            :MAKE ADDRESS ODD (ILLEGAL)  
1269 031562 012737 000400 032566     MOV     #256.,T23SZ        :SET UP BUFFER SIZE  
1270 031570 012704 032560     MOV     #T23PK3,R4        :R4 = POINTER TO PACKET  
1271 031574 010465 000000     MOV     R4,TSDB(R5)       :ISSUE COMMAND  
1272 031600 004737 016340     JSR     PC,WAITF          :WAIT FOR SSR TO SET  
1273 031604 016501 000002     MOV     TSSR(R5),R1      :GET TSSR CONTENTS  
1274 031610 012702 100206     MOV     #SSR!SC!BIT1!BIT2,R2 :SET UP EXPECTED  
1275 031614 020102         CMP     R1,R2            :ARE THEY EQUAL  
1276 031616 001406         BEQ     80$              :BR, IF OK ESP. FUNCTION REJECT  
1277 031620 005237 002212     INC     FATFLG           :BUMP COUNT  
1281 031624         ERRHRD  ERRNO,T23TM,PKTSSR :TSSR 1 CORRECT AFTER WRITE COMMAND  
           031624 104456         TRAP  C$ERHRD  
           031626 000503         .WORD 323  
           031630 033012         .WORD T23TM  
           031632 012136         .WORD PKTSSR  
1282 031634         80$:  CKLOOP           :LOOP IF SELECTED  
           031634 104406  
1283 031636         ENDSUB          :>>>>>>>>>> END SUBTEST >>>>>>>>>>  
           031636 104403         L10047:  
1284 031640 023727 002212 000017     CMP     FATFLG,#15.      :IS ERROR COUNT AT 25  
1285 031646 103402         BLO    999$             :BR, IF LESS THAN 25  
1286 031650 004737 017272     JSR     PC,CKDROP        :TRY TO DROP THE UNIT  
1287 031654         999$:
```



```

1346 031756           ERRHRD  ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTIC FAILED
      031756 104456          TRAP  C$ERHRD
      031760 000505          .WORD 325
      031762 005054          .WORD WRTMSG
      031764 012124          .WORD SFIMSG

```

1347
1348
1349
1350
1351
1352
1353

```

:*****
:WRITE DATA, ACK, CVC=1
:*****

```

```

1354 031766           23$:
1355 031766 004737 021276      JSR  PC,INVERT      ;INVERT THE EXTENDED FEATURES SWITCH
1356 031772 012737 140005 032560  MOV  #140005,T23PK3 ;WRITE DATA, ACK, CVC=1
1357 032000 013737 003130 032562  MOV  NXML0,T23WB   ;SET UP WRITE BUFFER ADDRESS
1358 032006 013737 003132 032564  MOV  NXMH1,T23WB+2 ;HIGH ORDER ADDRESS BITS
1359 032014 012737 000100 032566  MOV  #64,T23SZ    ;SET UP BUFFER SIZE
1360 032022 012704 032560      MOV  #T23PK3,R4   ;R4 = POINTER TO PACKET
1361 032026 010465 000000      MOV  R4,TSDB(R5)  ;ISSUE COMMAND
1362 032032 004737 016340      JSR  PC,WAITF     ;WAIT FOR SSR TO SET
1363 032036 016501 000002      MOV  TSSR(R5),R1  ;GET TSSR CONTENTS
1364 032042 012702 104210      MOV  #SC!NXM!SSR!BIT3,R2 ;SET UP EXPECTED
1365 032046 020102            CMP  R1,R2        ;ARE THEY EQUAL
1366 032050 001406            BEQ  80$          ;BR, IF OK ESP. FUNCTION REJECT
1367 032052 005237 002212      INC  FATFLG       ;BUMP COUNT
1371 032056           ERRHRD  ERRNO,T23TM,PKTSSR   ;TSSR INCORRECT AFTER WRITE COMMAND
      032056 104456          TRAP  C$ERHRD
      032060 000506          .WORD 326
      032062 033012          .WORD T23TM
      032064 012136          .WORD PKTSSR

1372 032066           80$:  CKLOOP      ;LOOP IF SELECTED
      032066 104406          TRAP  C$CLP1

1373 032070           90$:
1374 032070           ENDSUB      ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
      032070           L10050:
      032070 104403          TRAP  C$ESUB
1375 032072 023727 002212 000017  CMP  FATFLG,#15.  ;IS ERROR COUNT AT 25
1376 032100 103402          BLO  999$        ;BR, IF LESS THAN 25
1377 032102 004737 017272      JSR  PC,CKDROP    ;TRY TO DROP THE UNIT
1378 032106           999$:

```

1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397

```

: +
: TEST 3, SUBTEST 6
: VERIFIES THAT A WRITE DATA COMMAND SPECIFYING A DATA
: BUFFER STARTING IN EXISTANT MEMORY BUT RUNNING INTO
: NONEXISTENT MEMORY TERMINATES WITH THE PROPER ERROR
: STATUS. A LARGE ENOUGH RECORD SIZE IS SPECIFIED SUCH
: THAT TAPE IS ACTUALLY MOVED AND WRITTEN.
: *****
: 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!
: *****
: -
  
```

1398 032106
 032106
 032106 104402
 1399 032110 005737 003126
 1400 032114 001002
 1401 032116 000137 032402
 1402 032122 004737 034106
 1403 032126 004737 033752
 1404 032132 004737 034044
 1405
 1406
 1407
 1408
 1409
 1410
 1411

```

      BGNSUB                               ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
                                          T3.6:
                                          TRAP   CSBSUB
      TST      NXMFLG                      ;DO WE HAVE IT?
      BNE      10$                          ;BR, IF ENOUGH
      JMP      130$                          ;SKIP THIS TEST IF NOT
10$:  JSR      PC,T23RT3                     ;RESTORE PACKET
      JSR      PC,T23REST                    ;SET COMMAND PACKET
      JSR      PC,T23RT2                     ;SET UP OTHER COMMAND PACKET
  
```

1412 032136 004737 016064
 1413 032142 103407
 1414 032144 005237 002212
 1418 032150 010001
 1419 032152
 032152 104455
 032154 000507
 032156 003650
 032160 012124
 1420 032162
 1421 032162 013737 002172 032460
 1422 032170 012704 032440
 1423
 1424
 1425
 1426
 1427
 1428
 1429

```

: *****
: 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   CSERDF
                                          .WORD 327
                                          .WORD SFIERR
                                          .WORD SFIMSG
20$:  MOV      UNITN,T23DSW                   ;SET DRIVE NUMBER IN PACKET
      MOV      #T23PACKET,R4                 ;SUBROUTINE NEEDS PACKET ADDRESS
  
```

1430 032174 004737 010752
 1431 032200 103407
 1432 032202 005237 002212
 1436 032206 010001

```

: *****
: WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
: *****
      JSR      PC,WRTCHR                      ;ISSUE WRITE CHARACTERISTICS
      BCS      23$                          ;BR, IF COMMAND ISSUED OK
      INC      FATFLG                        ;BUMP COUNT
      MOV      R0,R1                          ;SAVE CONTENTS OF TSSR
  
```


1485 032420 004737 016546
1486 032424 103002
1487 032426 000137 027376
1488 032432
1489 032432
 032432 104432
 032434 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-

1491
 1492
 1493
 1495 032440 032440
 1497 032440 100004
 1498 032440 032450
 1499 032442 000000
 1500 032444 000010
 1501 032446 032462
 1502 032450 000000
 1503 032452 000012
 1504 032454 000000
 1505 032456 000000
 1506 032460 000000
 1507 032462
 1508 032462
 1509
 1510
 1511
 1513 032550 032550
 1515 032550 100006
 1516 032550 032572
 1517 032552 000000
 1518 032554 000006
 1519 032556
 1520
 1524 032560 100005
 1525 032560 000000
 1526 032562 000000
 1527 032564 000000
 1528 032566 000000
 1529
 1530
 1531 032570 000000
 1532
 1533
 1534 032572 010
 1535 032572 200
 1536 032573 000000
 1537 032574 000000
 1538 032576 000000
 1539
 1540
 1541 032600 000000
 1542 032602 000000
 1543
 1544
 1545
 1546
 1547 032604 100005
 1548 032606 100405
 1549 032610 102005
 1550 032612 177777
 1551
 1552

:+
 ;LOCAL STORAGE FOR THIS TEST
 :-

.=<.+10>B177770

T23PACKET:

.WORD 100004
 .WORD T23DATA
 .WORD 0
 .WORD 8.

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

T23DATA:

.WORD T23BFR
 .WORD 0
 .WORD 10.
 .WORD 0

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

T23DSW: .WORD 0
 T23BFR: .BLKW 25.

;SELECT DRIVE 0
 ;MESSAGE BUFFER

;WRITE SUBSYSTEM MEMORY COMMAND PACKET

.=<.+10>B177770

T23PK2:

.WORD 100006
 .WORD T23BF2
 .WORD 0
 .WORD 6.

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

T23PK3:

.WORD 100005
 T23WB: .WORD 0
 .WORD 0
 T23SZ: .WORD 0
 .EVEN

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

T23RSZ: .WORD 0

;LARGEST TAPE RECORD IN BYTES

T23BF2:

T23BS0: .BYTE 10
 T23BS1: .BYTE 200
 T23S2: .WORD 0
 T23S3: .WORD 0

;BSEL0 AREA
 ;BSEL1 AREA
 ;SEL 2 AREA
 ;DATA AREA

T23TMP: .WORD 0

;TEMPORARY REGISTER

T23WRT: .WORD 0

;RETRY COMMAND

.EVEN

;TAPE MOTION PACKET COMMAND VALUES

T23WD: .WORD 100005
 T23WDR: .WORD 100405
 T23CON: .WORD 102005
 .WORD 177777

;WRITE DATA (NEXT)
 ;WRITE DATA RETRY
 ;WRITE CONTINUOUS
 ;END OF DATA


```

1554
1555
1556      ;+
1557      ;LOCAL TEXT MESSAGES FOR TEST
1558      ;-
1559 032614 127 122 111 T23SSR: .ASCIZ 'WRITE Command Not Accepted'
1560 032647 105 117 124 T23ET: .ASCIZ 'EOT Not Found In 12000 4k Writes, (Use Shorter Tape)'
1561 032734 127 122 111 T23EOT: .ASCIZ 'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
1562 033012 124 123 123 T23TM: .ASCIZ 'TSSR Not Correct After WRITE Command Reject'
1563 033066 122 145 167 T23RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
1564 033135 122 101 115 T23RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
1565 033210 124 123 123 T23AM3: .ASCIZ 'TSSR Init. Failed After WRITE Command'
1566 033256 104 162 151 T23OFL: .ASCIZ 'Drive 7 Select Failed To Set 'DFL' In TSSR'
1567 033331 124 123 123 T23WDD: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, SWB Bit Set'
1568 033420 124 123 123 T23WDC: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
1569 033522 103 126 103 T23VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
1570 033575 124 123 102 T23BA: .ASCIZ 'TSBA Not Correct After WRITE DATA Command'
1571 033647 127 122 111 T23WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
1572 033736 102 141 163 T23ID: .ASCIZ 'Basic Write'
1573      .EVEN
1574
1575      ;+
1576      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1577      ;WRITE SUBSYSTEM MEMORY COMMAND
1578      ;-
1579
1580
1581 033752
1582 033752
1583 033756 012701 032440
1584 033762 012721 100004
1585 033766 012721 032450
1586 033772 005021
1587 033774 012721 000012
1588 034000 012721 032462
1589 034004 005021
1590 034006 012721 000024
1591 034012 005021
1592 034014 012711 000000
1593 034020 012702 000030
1594 034024 012762 177777 032462 64$:
1595 034032 005742
1596 034034 020227 000000
1597 034040 001371
1598 034042 000207
1599
1600
1601 034044
1602 034044
1603 034050 012701 032550
1604 034054 012721 100006
1605 034060 012721 032572
1606 034064 005021
1607 034066 012721 000006
1608 034072 012701 032572
1609 034076 005021
1610 034100 005021

```

```

T23REST:
      SAVREG      ;SAVE THE REGISTERS
      MOV #T23PACKET,R1 ;START OF THE PACKET
      MOV #100004,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK
      MOV #T23DATA,(R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
      CLR (R1)+ ;EXTENDED ADDRESS
      MOV #10,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
      MOV #T23BFR,(R1)+ ;ADDRESS OF MESSAGE BUFFER
      CLR (R1)+
      MOV #20,(R1)+ ;LENGTH OF MESSAGE BUFFER
      CLR (R1)+
      MOV #0,(R1) ;SELECT DRIVE ZERO
      MOV #24,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
      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      ;SAVE THE REGISTERS
      MOV #T23PK2,R1 ;START OF THE PACKET
      MOV #100006,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK
      MOV #T23BF2,(R1)+ ;ADDRESS OF DATA BLOCK
      CLR (R1)+ ;EXTENDED ADDRESS
      MOV #6,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
      MOV #T23BF2,R1 ;POINT TO DATA SEL AREA
      CLR (R1)+
      CLR (R1)+

```

1611	034102	0G5011			CLR	(R1)		
1612	034104	000207			RTS	PC		:RETURN
1613	034106				T23RT3:			
1614	034106				SAVREG			:SAVE THE REGISTERS
1615	034112	012701	032560		MOV	#T23PK3,R1		:START OF THE PACKET
1616	034116	012721	100005		MOV	#100005,(R1)+		:WRITE TAPE WITH ACK
1617	034122	005021			CLR	(R1)+		:ADDRESS OF DATA BLOCK
1618	034124	005021			CLR	(R1)+		:EXTENDED ADDRESS
1619	034126	005011			CLR	(R1)		:SIZE OF DATA BLOCK
1620	034130	000207			RTS	PC		:RETURN
1621					:*			
1622					:ROUTINE TO RETRY WRITE DATA IN CASE OF BAD TAPE FOR TEST			
1623					:3,SUBTEST 2 & 3			
1624					:INPUTS:			
1625						R1=TSSR		
1626						SUBROUTINE SETS UP T23WRT FOR RETRY		
1627					:-			
1628								
1629								
1630	034132				T23CHK:			
1631	034132				SAVREG			:SAVE THE REGISTERS
1632	034136	005037	032600		CLR	T23TMP		:CLEAR LOCAL REGISTER
1633	034142	032701	100000		BIT	#SC,R1		:IS SC SET IN TSSR?
1634	034146	001452			BEQ	FATAL		:NO, YOU GOT PROBLEMS!
1635	034150	013702	032472		MOV	T23BFR+10,R2		:YES,GET XSTAT1
1636	034154	032702	000002		BIT	#X1.UNC,R2		:IS UNC SET IN XSTAT1?
1637	034160	001401			BEQ	1\$:NO, CHECK COR
1638	034162	000405			BR	RETRY		:YES,DO WRITE DATA RETRY
1639	034164	032702	020000		1\$:	BIT	#X1.COR,R2	:IS COR SET IN XSTAT1 THEN?
1640	034170	001002			BNE	RETRY		:YES SO RETRY
1641	034172	000440			BR	FATAL		:NO, YOU GOT PROBLEMS
1642	034174	000207			EXIT:	RTS	PC	:RETURN
1643								
1644	034176				RETRY:			
1645	034176	012703	000024		2\$:	MOV	#20.,R3	:STARTING RECORD SIZE
1646	034202	013737	003114	032562		MOV	FREE,T23WB	:STARTING WRITE BUFFER ADDRESS
1647	034210	012737	032602	032560		MOV	#T23WRT,T23PK3	:WRITE DATA RETRY COMMAND SETUP BY SUBROUTINE
1648	034216	012704	032560			MOV	#T23PK3,R4	:SET UP R4 WITH PACKET ADDRESS
1649	034222	010300				MOV	R3,R0	:SET PATTERN IN CORRECT REGISTER
1650	034224	004737	017512			JSR	PC,FILLMEM	:FILL MEMORY WITH RECORD SIZE
1651	034230	010337	032566			MOV	R3,T23SZ	:SET UP RECORD SIZE IN PACKET
1652	034234	010465	000000			MOV	R4,TSDB(R5)	:ISSUE COMMAND
1653	034240	004737	016340			JSR	PC,WAITF	:WAIT FOR SSR TO SET
1654	034244	016501	000002			MOV	TSSR(R5),R1	:GET TSSR CONTENTS
1655	034250	012702	000200			MOV	#SSR,R2	:SET UP EXPECTED
1656	034254	020102				CMP	R1,R2	:ARE THEY EQUAL
1657	034256	001746				BEQ	EXIT	:BR, IF OK
1658	034260	005237	032600			INC	T23TMP	:TRY FIVE TIMES THEN EXIT
1659	034264	022737	000005	032600		CMP	#5,T23TMP	:DONE FIVE YET?
1660	034272	001341				BNE	2\$:NO GO AGAIN
1661	034274	005237	002212		FATAL:	INC	FATFLG	:BUMP COUNT
1665	034300	013702	032462			MOV	T23BFR,R2	:LOW ORDER MSGBUF
1666	034304				ERRHRD	ERRNO,SCHERR,PKTMES		:TSSR INCORRECT AFTER WRITE DATA
	034304	104456						TRAP CSERHRD
	034306	000513						.WORD 331
	034310	005276						.WORD SCHERR
	034312	012200						.WORD PKTMES

TSV7 - HARDWARE TESTS 1-8
TEST 3: BASIC WRITE DATA

MACRO M1113 25-MAY-82 09:19 PAGE 94-2 E 13

SEQ 0160

1667 034314 004737 017272
1668 034320
034320
034320 104401

JSR PC,CKDROP
ENDTST

:DROP THE UNIT

L10043: TRAP CSETST


```

1728
1729
1730
1731 034406 004737 016064
1732 034412 103426
1733 034414
    034414 012727 000250
    034420 000000
    034422 013727 002116
    034426 000000
    034430 005367 177772
    034434 001375
    034436 005367 177756
    034442 001367
1734 034444 005337 043736
1735 034450 001356
1736 034452 005237 002212
1740 034456 010001
1741 034460
    034460 104455
    034462 000621
    034464 003650
    034466 012124
1742 034470
1743 034470 012737 000007 043610
1744 034476 012704 043570
1745
1746
1747
1748
1749
1750
1751
1752 034502 004737 010752
1753 034506 103407
1754 034510 005237 002212
1758 034514 010001
1759 034516
    034516 104456
    034520 000622
    034522 005054
    034524 012124
1760 034526 005737 002216
1761 034532 001044
1762
1763 034534 112737 000200 043721
1764 034542 112737 000010 043720
1765 034550 012704 043700
1766 034554 010465 000000
1767 034560 004737 016426
1768 034564 103407
1769 034566 010001
1770 034570 005237 002212
1774 034574
    034574 104456
    034576 000623
    034600 044457

:*****
5$: JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
    BCS 20$ ;BR IF INIT WAS OK
    DELAY 250 ;DELAY AWHILE
    MOV #250,(PC)+
    .WORD 0
    MOV LSDLY,(PC)+
    .WORD 0
    DEC -6(PC)
    BNE -4
    DEC -22(PC)
    BNE -20
    DEC T24DLY ;BUMP DELAY COUNTER
    BNE 5$ ;BR, IF MORE DELAY REQUIRED
    INC FATFLG ;BUMP COUNT
    MOV R0,R1 ;CONTENTS OF TSSR REGISTER
    ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
    TRAP C$ERDF
    .WORD 401
    .WORD SFIERR
    .WORD SFIMSG
20$: MOV #7,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 24$ ;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$ERHRD
    .WORD 402
    .WORD WRTMSG
    .WORD SFIMSG
24$: TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
    BNE 50$ ;BR IF SWITCH IS ON
    MOVB #200,T24BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
    MOVB #10,T24BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
    MOV #T24PK2,R4 ;WRITE SUBSYS MEM PACKET
    MOV R4,TSDB(R5) ;ISSUE COMMAND
    JSR PC,CHKTSSR ;WAIT FOR SSR
    BCS 30$ ;BR, IF NO ERROR
    MOV R0,R1 ;ERROR, SAVE TSSR
    INC FATFLG ;BUMP COUNT
    ERRHRD ERRNO,T24SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
    TRAP C$ERHRD
    .WORD 403
    .WORD T24SSR
    
```

```

1775 034602 012136
      034604 104406
1776 034606 012737 000007 043610
1777 034614 012704 043570
1778
1779
1780
1781
1782
1783
1784
1785 034620 004737 010752
1786 034624 103407
1787 034626 005237 002212
1791 034632 010001
1792 034634
      034634 104456
      034636 000624
      034640 005054
      034642 012124
1793 034644
      034644 104406
1794 034646 016501 000002
1795 034652 032701 000100
1796 034656 001006
1797 034660 005237 002212
1801 034664
      034664 104455
      034666 000625
      034670 045235
      034672 012124
1802 034674
      034674 104406
1803 034676 012703 043726
1804
1805
1806
1807
1808
1809
1810
1811 034702 011337 043710
1812 034706 012704 043710
1813 034712 010465 000000
1814 034716 004737 016340
1815 034722 016501 000002
1816 034726 012702 100306
1817 034732 020102
1818 034734 001406
1819 034736 005237 002212
1823 034742
      034742 104456
      034744 000626
      034746 044773
      034750 012136
1824 034752
    
```

30\$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR

MOV #7,T24DSW ;SET DRIVE NUMBER IN PACKET TRAP C\$CLP1

MOV #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS

:WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)

JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS

BCS 50\$;BR, IF COMMAND ISSUED OK

INC FATFLG ;BUMP COUNT

MOV RO,R1 ;SAVE CONTENTS OF TSSR

ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED

TRAP C\$ERHRD

.WORD 404

.WORD WRTMSG

.WORD SFIMSG

50\$: CKLOOP ;SCOPE LOOP TRAP C\$CLP1

MOV TSSR(R5),R1 ;GET TSSR CONTENTS

BIT #OFL,R1 ;CHECK FOR THE OFFLINE BIT SET

BNE 60\$;BR, IF OFFLINE (GOOD)

INC FATFLG ;BUMP COUNT

ERRDF ERRNO,T24OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)

TRAP C\$ERDF

.WORD 405

.WORD T24OFL

.WORD SFIMSG

60\$: CKLOOP ;LOOP IF SELECTED TRAP C\$CLP1

MOV #T24RN,R3 ;POINTER FOR COMMANDS

:TAPE READ COMMAND IN PLACE

65\$: MOV (R3),T24PK3 ;TAPE READ COMMAND IN PLACE

MOV #T24PK3,R4 ;R4 = POINTER TO 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!OFL!BIT1!BIT2,R2 ;SET UP EXPECTED

CMP R1,R2 ;ARE THEY EQUAL

BEQ 80\$;BR, IF OK ESP. FUNCTION REJECT

INC FATFLG ;BUMP COUNT

ERRHRD ERRNO,T24TM,PKTSSR ;TSSR INCORRECT AFTER TAPE MOTION CMD

TRAP C\$ERHRD

.WORD 406

.WORD T24TM

.WORD PKTSSR

80\$: CKLOOP ;LOOP IF SELECTED

1825	034752	104406								
1826	034754	005723			TST	(R3)+			TRAP	C\$CLP1
1827	034756	022713	177777		CMP	#177777,(R3)			:BUMP TO NEXT COMMAND	
1828	034762	001401			BEQ	90\$:END OF THE COMMANDS YET	
1829	034764	000746			BR	65\$:BR, IF DONE	
1830	034766			90\$:	ENDSUB				:MORE COMMAND(S) TO GO	
	034766								:>>>>>>>>>> END SUBTEST >>>>>>>>>>	
	034766	104403							L10053:	
1831	034770	023727	002212	000017	CMP	FATFLG,#15.			TRAP	C\$ESUB
1832	034776	103402			BLO	999\$:IS ERROR COUNT AT 25	
1833	035000	004737	017272		JSR	PC,CKDROP			:BR, IF LESS THAN 25	
1834	035004			999\$:					:TRY TO DROP THE UNIT	


```

1890 035102 012124
      035104 104406
      035104 104406
1891
1892
1893
1894
1895
1896
1897
1898 035106 004737 011104
1899 035112 103407
1900 035114 010001
1901 035116 005237 002212
1905 035122
      035122 104456
      035124 000631
      035126 045046
      035130 012136
1906 035132
      035132 104406
1907
1908
1909
1910
1911
1912
1913
1914 035134 013701 043620
1915 035140 010102
1916 035142 052702 000002
1917 035146 020102
1918 035150 001406
1919 035152 005237 002212
1923 035156
      035156 104456
      035160 000632
      035162 044563
      035164 015564
1924 035166
      035166 104406
1925 035170 012703 000400
1926 035174 013737 003114 043712
1927
1928
1929
1930
1931
1932
1933
1934 035202 012737 140005 043710
1935 035210 012704 043710
1936 035214
1937 035214 010300
1938 035216 004737 017512
1939 035222 010337 043716
1940 035226 010465 000000

      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 409
      .WORD T24RWN
      .WORD PKTSSR

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

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

      MOV T24BFR+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,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      TRAP C$ERHRD
      .WORD 410
      .WORD T24BOT
      .WORD EXPREC

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

      MOV #256.,R3 ;RECORD SIZE
      MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS

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

      MOV #140005,T24PK3 ;WRITE DATA,CVC=1,ACK 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,TSDB(R5) ;ISSUE COMMAND
    
```

```

1941 035232 004737 016340      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
1942 035236 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
1943 035242 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
1944 035246 020102                CMP      R1,R2        ;ARE THEY EQUAL
1945 035250 001406      BEQ      75$          ;BR, IF OK
1946 035252 005237 002212      INC      FATFLG       ;BUMP COUNT
1950 035256                ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    411
                                .WORD    WRTErr
                                .WORD    PKTSSR
1951 035266                75$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
1952 035270 005723                TST      (R3)+        ;BUMP RECORD SIZE
1953 035272 022703 000414      CMP      #268.,R3     ;END OF RECORD YET
1954 035276 001346      BNE      65$          ;BR, IF MORE RECORDS TO WRITE
1955 035300                80$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
1956 035302                120$:
1957
1958
1959
1960
1961
1962
1963
1964 035302 004737 011104      JSR      PC,REWIND    ;CALL TAPE REWIND COMMAND
1965 035306 004737 016426      JSR      PC,CHKTSSR   ;SEE HOW TSSR IS
1966 035312 103407                BCS      130$         ;BR, IF NO PROBLEM
1967 035314 010001                MOV      R0,R1        ;SAVE TSSR
1968 035316 005237 002212      INC      FATFLG       ;BUMP COUNT
1972 035322                ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    412
                                .WORD    T24RWN
                                .WORD    PKTSSR
1973 035332                130$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
1974
1975
1976
1977
1978
1979
1980
1981 035334 013701 043620                MOV      T24BFR+6,R1  ;PICK UP XSTO
1982 035340 010102                MOV      R1,R2        ;SET UP EXPECTED
1983 035342 052702 000002      BIS      #BIT1,R2     ;SET BO. BIT IN EXPECTED
1984 035346 020102                CMP      R1,R2        ;DOES EXP = REC'D
1985 035350 001406      BEQ      140$         ;BR, IF EQUAL (OK)
1986 035352 005237 002212      INC      FATFLG       ;BUMP COUNT
1990 035356                ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    413
                                .WORD    T24BOT
                                .WORD    EXPREC
1991 035366                140$:  CKLOOP                ;LOOP IF SELECTED
    
```


2041 035560 004737 017272 JSR PC,CKDROP
2042 035564 999\$:

;TRY TO DROP THE UNIT

```

2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056 035564          BGNSUB          :>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
      035564          T4.3:
      035564 104402          JSR      PC,T24RT3          :SET UP OTHER COMMAND PACKET      TRAP      CSBSUB
2057 035566 004737 046404          JSR      PC,T24REST          :SET COMMAND PACKET
2058 035572 004737 046250          JSR      PC,T24RT2          :SET UP OTHER COMMAND PACKET
2059 035576 004737 046342
2060
2061
2062
2063 :*****
2064 :ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
2065 :*****
2066
2067 035602 004737 016064          JSR      PC,SOFINIT          :DO INITIALIZE ON CONTROLLER
2068 035606 103407          BCS      20$                 :BR IF INIT WAS OK
2069 035610 005237 0022'2          INC      FATFLG              :BUMP COUNT
2073 035614 010001          MOV      R0,R1               :CONTENTS OF TSSR REGISTER
2074 035616          ERRDF  ERRNO,SFIERP,SFIMSG :FATAL ERROR TSSR WAS NOT OK
      035616 104455          TRAP      CSERDF
      035620 000640          .WORD    416
      035622 003650          .WORD    SFIERR
      035624 012124          .WORD    SFIMSG
2075 035626          20$:
2076 035626 013737 002172 043610          MOV      UNITN,T24DSW        :SET DRIVE NUMBER IN PACKET
2077 035634 012704 043570          MOV      #T24PACKET,R4      :SUBROUTINE NEEDS PACKET ADDRESS
2078
2079
2080 :*****
2081 :WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
2082 :*****
2083
2084
2085 035640 004737 010752          JSR      PC,WRTCHR          :ISSUE WRITE CHARACTERISTICS
2086 035644 103407          BCS      24$                 :BR, IF COMMAND ISSUED OK
2087 035646 005237 002212          INC      FATFLG              :BUMP COUNT
2091 035652 010001          MOV      R0,R1               :SAVE CONTENTS OF TSSR
2092 035654          ERRHRD ERRNO,WRTMSG,SFIMSG :WRITE CHARACTERISTICS FAILED
      035654 104456          TRAP      CSERHRD
      035656 000641          .WORD    417
      035660 005054          .WORD    WRTMSG
      035662 012124          .WORD    SFIMSG
2093 035664          24$: CKLOOP              :LOOP IF SELECTED
      035664 104406          TRAP      CSCLP1
2094
2095 :*****
    
```

```
2096  
2097  
2098  
2099  
2100  
2101 035666 004737 011104  
2102 035672 103407  
2103 035674 010001  
2104 035676 005237 002212  
2108 035702  
035702 104456  
035704 000642  
035706 045046  
035710 012136  
2109 035712  
035712 104406  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117 035714 013701 043620  
2118 035720 010102  
2119 035722 052702 000002  
2120 035726 020102  
2121 035730 001406  
2122 035732 005237 002212  
2126 035736  
035736 104456  
035740 000643  
035742 044563  
035744 015564  
2127 035746  
035746 104406  
2128 035750 012703 000400  
2129 035754 013737 003114 043712  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137 035762 012737 150005 043710  
2138 035770 012704 043710  
2139 035774  
2140 035774 010300  
2141 035776 004737 017512  
2142 036002 010337 043716  
2143 036006 010465 000000  
2144 036012 004737 016340  
2145 036016 016501 000002  
2146 036022 012702 000200  
2147 036026 020102  
2148 036030 001406  
: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$ERHRD  
.WORD 418  
.WORD T24RWN  
.WORD PKTSSR  
30$: CKLOOP ;LOOP IF SELECTED  
TRAP C$CLP1  
:*****  
:READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)  
:*****  
MOV T24BFR+6,R1 ;PICK UP XSTO  
MOV R1,R2 ;SET UP EXPECTED  
BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED  
CMP R1,R2 ;DOES EXP = REC'D  
BEQ 40$ ;BR, IF EQUAL (OK)  
INC FATFLG ;BUMP COUNT  
ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND  
TRAP C$ERHRD  
.WORD 419  
.WORD T24BOT  
.WORD EXPREC  
40$: CKLOOP ;LOOP IF SELECTED  
TRAP C$CLP1  
MOV #256,R3 ;RECORD SIZE  
MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS  
:*****  
:WRITE DATA,ACK,SWB,CVC=1 COMMAND  
:*****  
MOV #150005,T24PK3 ;WRITE DATA,ACK,SWB,CVC=1 COMMAND  
MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS  
65$: 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,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
```

```

2149 036032 005237 002212          INC    FATFLG          ;BUMP COUNT
2153 036036          ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      036036 104456          TRAP  CSERHRD
      036040 000644          .WORD 420
      036042 005111          .WORD WRTErr
      036044 012136          .WORD PKTSSR
2154 036046          75$:  CKLOOP          ;LOOP IF SELECTED
      036046 104406          TRAP  CSCLP1
2155 036050 005723          TST    (R3)+          ;BUMP RECORD SIZE
2156 036052 022703 000414      CMP    #268.,R3      ;END OF RECORD YET
2157 036056 001346          BNE    65$           ;BR, IF MORE RECORDS TO WRITE
2158 036060          80$:  CKLOOP          ;LOOP IF SELECTED
      036060 104406          TRAP  CSCLP1
2159 036062          120$:
2160
2161
2162
2163
2164
2165
2166
2167 036062 004737 011104          JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
2168 036066 103407          BCS    130$          ;BR, IF NO PROBLEM
2169 036070 010001          MOV    R0,R1         ;SAVE TSSR
2170 036072 005237 002212      INC    FATFLG        ;BUMP COUNT
2174 036076          ERRHRD  ERRNO,T24RWN,EXPREC ;REWIND NOT ACCEPTED
      036076 104456          TRAP  CSERHRD
      036100 000645          .WORD 421
      036102 045046          .WORD T24RWN
      036104 015564          .WORD EXPREC
2175 036106          130$:  CKLOOP          ;LOOP IF SELECTED
      036106 104406          TRAP  CSCLP1
2176
2177
2178
2179
2180
2181
2182
2183 036110 013701 043620          MOV    T24BFR+6,R1   ;PICK UP XSTO
2184 036114 010102          MOV    R1,R2         ;SET UP EXPECTED
2185 036116 052702 000002      BIS    #BIT1,R2      ;SET BOT BIT IN EXPECTED
2186 036122 020102          CMP    R1,R2         ;DOES EXP = REC'D
2187 036124 001406          BEQ    140$          ;BR, IF EQUAL (OK)
2188 036126 005237 002212      INC    FATFLG        ;BUMP COUNT
2192 036132          ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      036132 104456          TRAP  CSERHRD
      036134 000646          .WORD 422
      036136 044563          .WORD T24BOT
      036140 015564          .WORD EXPREC
2193 036142          140$:  CKLOOP          ;LOOP IF SELECTED
      036142 104406          TRAP  CSCLP1
2194 036144 012703 000400          MOV    #256.,R3      ;RECORD SIZE
2195 036150 013737 003114 043712  MOV    FREE,T24RB    ;STARTING READ BUFFER ADDRESS
2196
2197
2198
    
```



```

2298
2299
2300      :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2301      :*****
2302
2303 036442 004737 011104      JSR      PC,REWIND      :CALL TAPE REWIND COMMAND
2304 036446 103407      BCS      30$           :BR, IF NO PROBLEM
2305 036450 010001      MOV      R0,R1        :SAVE TSSR
2306 036452 005237 002212      INC      FATFLG       :BUMP COUNT
2310 036456      ERRHRD  ERRNO,T24RWN,PKTSSR :REWIND NOT ACCEPTED
      036456 104456      TRAP      CSERHRD
      036460 000653      .WORD    427
      036462 045046      .WORD    T24RWN
      036464 012136      .WORD    PKTSSR
2311 036466      30$:   CKLOOP      :LOOP IF SELECTED      TRAP      CSCLP1
      036466 104406
2312
2313      :*****
2314      :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
2315      :*****
2316
2317
2318
2319 036470 013701 043620      MOV      T24BFR+6,R1  :PICK UP XSTO
2320 036474 010102      MOV      R1,R2        :SET UP EXPECTED
2321 036476 052702 000002      BIS      #BIT1,R2     :SET BOT BIT IN EXPECTED
2322 036502 020102      CMP      R1,R2        :DOES EXP = REC'D
2323 036504 001406      BEQ     40$           :BR, IF EQUAL (OK)
2324 036506 005237 002212      INC      FATFLG       :BUMP COUNT
2328 036512      ERRHRD  ERRNO,T24BOT,EXPREC :TAPE NOT AT BOT AFTER REWIND
      036512 104456      TRAP      CSERHRD
      036514 000654      .WORD    428
      036516 044563      .WORD    T24BOT
      036520 015564      .WORD    EXPREC
2329 036522      40$:   CKLOOP      :LOOP IF SELECTED      TRAP      CSCLP1
      036522 104406
2330 036524 012703 001000      MOV      #512.,R3     :RECORD SIZE
2331 036530 013737 003114 043712  MOV      FREE,T24RB   :STARTING WRITE BUFFER ADDRESS
2332
2333      :*****
2334      :WRITE DATA,ACK,CVC=1 COMMAND
2335      :*****
2336
2337
2338
2339 036536 012737 140005 043710  MOV      #140005,T24PK3 :WRITE DATA,ACK,CVC=1 COMMAND
2340 036544 012704 043710      MOV      #T24PK3,R4   :SET UP R4 WITH PACKET ADDRESS
2341 036550      65$:   MOV      R3,T24SZ     :SET UP RECORD SIZE IN PACKET
2342 036550 010337 043716      MOV      R4,TSDB(R5)  :ISSUE COMMAND
2343 036554 010465 000000      JSR      PC,WAITF     :WAIT FOR SSR TO SET
2344 036560 004737 016340      MOV      TSSR(R5),R1  :GET TSSR CONTENTS
2345 036564 016501 000002      MOV      #SSR,R2     :SET UP EXPECTED
2346 036570 C12702 000200      CMP      R1,R2        :ARE THEY EQUAL
2347 036574 020102      BEQ     75$           :BR, IF OK
2348 036576 001406      INC      FATFLG       :BUMP COUNT
2349 036600 005237 002212      ERRHRD  ERRNO,WRERR,PKTSSR :TSSR INCORRECT AFTER WRITE DATA
2353 036604
    
```

```

036604 104456
036606 000655
036610 005111
036612 012136
2354 036614 104406 75$: CKLOOP ;LOOP IF SELECTED TRAP CSERHRD
036614 104406 TRAP CSCLP1
2355 036616
2356
2357
2358
2359
2360
2361
2362
2363 036616 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2364 036622 103407 BCS 130$ ;BR, IF NO PROBLEM
2365 036624 010001 MOV R0,R1 ;SAVE TSSR
2366 036626 005237 002212 INC FATFLG ;BUMP COUNT
2370 036632 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
036632 104456 TRAP CSERHRD
036634 000656 .WORD 430
036636 045046 .WORD T24RWN
036640 012136 .WORD PKTSSR
2371 036642 104406 130$: CKLOOP ;LOOP IF SELECTED TRAP CSCLP1
036642 104406
2372
2373
2374
2375
2376
2377
2378
2379 036644 013701 043620 MOV T24BFR+6,R1 ;PICK UP XST0
2380 036650 010102 MOV R1,R2 ;SET UP EXPECTED
2381 036652 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
2382 036656 020102 CMP R1,R2 ;DOES EXP = REC'D
2383 036660 001406 BEQ 140$ ;BR, IF EQUAL (OK)
2384 036662 005237 002212 INC FATFLG ;BUMP COUNT
2388 036666 ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
036666 104456 TRAP CSERHRD
036670 000657 .WORD 431
036672 044563 .WORD T24BOT
036674 015564 .WORD EXPREC
2389 036676 104406 140$: CKLOOP ;LOOP IF SELECTED TRAP CSCLP1
036676 104406
2390 036700 012703 000400 MOV #256.,R3 ;RECORD SIZE
2391 036704 013737 003114 043712 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
2392
2393
2394
2395
2396
2397
2398
2399 036712 012737 140001 043710 MOV #140001,T24PK3 ;READ DATA,ACK,CVC=1 COMMAND
2400 036720 012704 043710 165$: MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2401 036724 010337 043716 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
    
```


2438
 2439
 2440
 2441
 2442
 2443
 2444
 2445
 2446
 2447
 2448
 2449
 2450
 2451
 2452
 2453
 2454
 2455
 2456
 2457
 2458
 2459
 2460
 2461
 2462
 2463
 2464
 2465
 2466
 2467
 2468
 2472
 2473
 2474
 2475
 2476
 2477
 2478
 2479
 2480
 2481
 2482
 2483
 2484
 2485
 2486
 2490
 2491
 2491
 2491
 2491

```

: *
: TEST 4, SUBTEST 5
: VERIFIES THAT A READ FORWARD COMMAND READING A RECORD
: SHORTER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE
: STATUS ALERT TERMINATION WITH THE RECORD LENGTH SHORT
: (RLS) BITS SET. IT IS VERIFIED THAT THE RESIDUAL BYTE
: COUNT (RBPCR) IN THE MESSAGE BUFFER CONTAINS THE
: PROPER NONZERO VALUE (E.G. THE DIFFERENCE BETWEEN
: THE ORIGINAL BYTE COUNT AND THE ACTUAL RECORD
: LENGTH).
: -
: *****
: 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   CSERDF           ;TRAP
      .WORD  434              ;.WORD
      .WORD  SFIERR          ;.WORD
      .WORD  SFIMSG          ;.WORD
20$:
      MOV     UNITN,T24DSW     ;SET DRIVE NUMBER IN PACKET
      MOV     #T24PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
: *****
: WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
: *****
      JSR     PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
      BCS    24$               ;BR, IF COMMAND ISSUED OK
      INC    FATFLG           ;BUMP COUNT
      MOV    R0,R1            ;SAVE CONTENTS OF TSSR
      ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      TRAP   CSERHRD         ;TRAP
      .WORD  435              ;.WORD
      .WORD  WRTMSG          ;.WORD
    
```

037042
 037042
 037042 104402
 037044 004737 046404
 037050 004737 046250
 037054 004737 046342
 037060 004737 016064
 037064 103407
 037066 005237 002212
 037072 010001
 037074 104455
 037076 000662
 037100 003650
 037102 012124
 037104
 037104 013737 002172 043610
 037112 012704 043570
 037116 004737 010752
 037122 103407
 037124 005237 002212
 037130 010001
 037132 104456
 037134 000663
 037136 005054

```

2492 037140 012124
037142 104406
2493
2494
2495
2496
2497
2498
2499
2500 037144 004737 011104
2501 037150 103407
2502 037152 010001
2503 037154 005237 002212
2507 037160
037160 104456
037162 000664
037164 045046
037166 012136
2508 037170
037170 104406
2509 037172 012703 000400
2510 037176 013737 003114 043712
2511
2512
2513
2514
2515
2516
2517
2518 037204 012737 140005 043710
2519 037212 012704 043710
2520 037216
2521 037216 010337 043716
2522 037222 010465 000000
2523 037226 004737 016340
2524 037232 016501 000002
2525 037236 012702 000200
2526 037242 020102
2527 037244 001406
2528 037246 005237 002212
2532 037252
037252 104456
037254 000665
037256 005111
037260 012136
2533 037262
037262 104406
2534 037264
2535
2536
2537
2538
2539
2540
2541
2542 037264 004737 011104
    
```

24\$: CKLOOP ;LOOP IF SELECTED .WORD SFIMSG
 TRAP CSCLP1

 :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 CSERHRD
 .WORD 436
 .WORD T24RWN
 .WORD PKTSSR

30\$: CKLOOP ;LOOP IF SELECTED TRAP CSCLP1
 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,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA

TRAP CSERHRD
 .WORD 437
 .WORD WRTErr
 .WORD PKTSSR

75\$: CKLOOP ;LOOP IF SELECTED TRAP CSCLP1
 120\$:

 :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE

JSR PC,REWIND ;CALL TAPE REWIND COMMAND

```

2543 037270 103407          BCS      130$          :BR, IF NO PROBLEM
2544 037272 010001          MOV      R0,R1        :SAVE TSSR
2545 037274 005237 002212  INC      FATFLG       :BUMP COUNT
2549 037300          ERRHRD  ERRNO,T24RWN,PKTSSR :REWIND NOT ACCEPTED
      037300 104456          TRAP    C$ERHRD
      037302 000666          .WORD  438
      037304 045046          .WORD  T24RWN
      037306 012136          .WORD  PKTSSR
2550 037310          130$:  CKLOOP        :LOOP IF SELECTED          TRAP    C$CLP1
      037310 104406
2551 037312 012703 001000          MOV      #512.,R3     :RECORD SIZE
2552 037316 013737 003114 043712  MOV      FREE,T24RB   :STARTING READ BUFFER ADDRESS
2553
2554 :*****
2555 :
2556 :READ DATA,ACK,CVC=1 COMMAND
2557 :
2558 :*****
2559
2560 037324 012737 140001 043710  MOV      #140001,T24PK3 :READ DATA,ACK,CVC=1 COMMAND
2561 037332 012704 043710 165$:  MOV      #T24PK3,R4   :SET UP R4 WITH PACKET ADDRESS
2562 037336 010337 043716  MOV      R3,T24SZ     :SET UP RECORD SIZE IN PACKET
2563 037342 010465 000000  MOV      R4,TSDB(R5)  :ISSUE COMMAND
2564 037346 004737 016340  JSR      PC,WAITF     :WAIT FOR SSR TO SET
2565 037352 016501 000002  MOV      TSSR(R5),R1  :GET TSSR CONTENTS
2566 037356 012702 100204  MOV      #SSR!SC!BIT2,R2 :SET UP EXPECTED
2567 037362 020102          CMP      R1,R2        :ARE THEY EQUAL
2568 037364 001406          BEQ     170$          :BR, IF OK
2569 037366 005237 002212  INC      FATFLG       :BUMP COUNT
2573 037372          ERRHRD  ERRNO,T24TRL,EXPREC :TSSR INCORRECT AFTER READ DATA
      037372 104456          TRAP    C$ERHRD
      037374 000667          .WORD  439
      037376 046114          .WORD  T24TRL
      037400 015564          .WORD  EXPREC
2574 037402          170$:  CKLOOP        :LOOP IF SELECTED          TRAP    C$CLP1
      037402 104406
2575
2576 :*****
2577 :
2578 :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
2579 :
2580 :*****
2581
2582 037404 013701 043620          MOV      T24BFR+6,R1  :GET MESSAGE BUFFER
2583 037410 010102          MOV      R1,R2        :SET UP EXPECTED
2584 037412 052702 040000  BIS      #BIT14,R2    :SET THE RLS BIT IN EXPECTED
2585 037416 020102          CMP      R1,R2        :ARE THEY EQUAL
2586 037420 001406          BEQ     180$          :BR, IF EQUAL (ALL IS WELL)
2587 037422 005237 002212  INC      FATFLG       :BUMP COUNT
2591 037426          ERRHRD  ERRNO,T24LOP,EXPREC :THE RLL BIT WAS NOT SET IN XSTO
      037426 104456          TRAP    C$ERHRD
      037430 000670          .WORD  440
      037432 045744          .WORD  T24LOP
      037434 015564          .WORD  EXPREC
2592 037436          180$:
2593 037436 013701 043616          MOV      T24BFR+4,R1  :PICK UP RESIDUAL BYTE COUNTER
2594 037442 012702 000400          MOV      #256.,R2    :THIS SHOULD BE THE DIFFERENCE
    
```

2595	037446	020102			CMP	R1,R2					:IS THE DIFFERENCE CORRECT	
2596	037450	001406			BEQ	190\$:BR, IF CORRECT	
2597	037452	005237	002212		INC	FATFLG					:BUMP COUNT	
2601	037456				ERRHRD	ERRNO,T24PBP,EXPREC					:RBPDR NOT CORRECT	
	037456	104456										TRAP CSERHRD
	037460	000671										.WORD 441
	037462	046026										.WORD T24PBP
	037464	015564										.WORD EXPREC
2602	037466			190\$:	CKLOOP						:LOOP IF SELECTED	
2603	037470	104406			ENDSUB							TRAP CSCLP1
	037470										:>>>>>>>>>> END SUBTEST >>>>>>>>>>	
	037470	104403									L10057:	
2604	037472	023727	002212	000017	CMP	FATFLG,#15.						TRAP CSSESUB
2605	037500	103402			BLO	999\$:I ^c ERROR COUNT AT 25	
2606	037502	004737	017272		JSR	PC,CKDROP					:BR, IF LESS THAN 25	
2607	037506			999\$:							:TRY TO DROP THE UNIT	


```

2664 037604 012124
037606 104406
2665
2666
2667
2668
2669
2670
2671
2672 037610 004737 011104
2673 037614 103407
2674 037616 010001
2675 037620 005237 002212
2679 037624
037624 104456
037626 000674
037630 045046
037632 012136
2680 037634
037634 104406
2681 037636 012703 000400
2682 037642 013737 003114 043712
2683
2684
2685
2686
2687
2688
2689
2690 037650 012737 140005 043710
2691 037656 012704 043710
2692 037662
2693 037662 010300
2694 037664 004737 017512
2695 037670 010337 043716
2696 037674 010465 000000
2697 037700 004737 016340
2698 037704 016501 000002
2699 037710 012702 000200
2700 037714 020102
2701 037716 001406
2702 037720 005237 002212
2706 037724
037724 104456
037726 000675
037730 005111
037732 012136
2707 037734
037734 104406
2708 037736 005723
2709 037740 022703 000414
2710 037744 001346
2711 037746
037746 104406
2712 037750 005743
2713 037752 013737 003114 043712

```

```

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$ERHRD
.WORD 444
.WORD T24RWN
.WORD PKTSSR
30$: 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,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,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
BEC 75$ ;BR, IF OK
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
TRAP C$ERHRD
.WORD 445
.WORD WRERR
.WORD PKTSSR
75$: 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
TST -(R3) ;SET BACK TO 512.
MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS

```

```

2714
2715
2716
2717
2718
2719
2720
2721 037760 012737 100401 043710
2722 037766 012704 043710
2723 037772 010337 043716
2724 037776 010465 000000
2725 040002 004737 016340
2726 040006 016501 000002
2727 040012 012702 000200
2728 040016 020102
2729 040020 001406
2730 040022 005237 002212
2734 040026
      040026 104456
      040030 000676
      040032 045376
      040034 012136
2735 040036
      040036 104406
2736 040040 013702 003114
2737 040044 010304
2738 040046 162704 000400
2739 040052 060204
2740 040054 021403
2741 040056 001410
2742 040060 011401
2743 040062 010302
2744 040064 005237 002212
2748 040070
      040070 104456
      040072 000677
      040074 044630
      040076 015564
2749 040100
      040100 104406
2750 040102 005724
2751 040104 160204
2752 040106 020403
2753 040110 001360
2754 040112 005743
2755 040114 022703 000400
2756 040120 001322
2757 040122
      040122 104406
2758 040124
      040124
      040124 104403
2759 040126 023727 002212 000017
2760 040134 103402
2761 040136 004737 017272
2762 040142

*****
:READ REVERSE DATA,ACK COMMAND
*****
165$:  MOV #100401,T24PK3 :READ REVERSE DATA,ACK COMMAND
      MOV #T24PK3,R4 :SET UP R4 WITH PACKET ADDRESS
      MOV R3,T24SZ :SET UP RECORD SIZE IN PACKET
      MOV R4,TSDB(R5) :ISSUE COMMAND
      JSR PC,WAITF :WAIT FOR SSR TO SET
      MOV TSSR(R5),R1 :GET TSSR CONTENTS
      MOV #SSR,R2 :SET UP EXPECTED
      CMP R1,R2 :ARE THEY EQUAL
      BEQ 170$ :BR, IF OK
      INC FATFLG :BUMP COUNT
      ERRHRD ERRNO,T24WDC,PKTSSR :TSSR INCORRECT AFTER READ DATA
      TRAP CSERHRD
      .WORD 446
      .WORD T24WDC
      .WORD PKTSSR
170$:  CKLOOP :LOOP IF SELECTED
      TRAP CSCLP1
      MOV FREE,R2 :GET BUFFER ADDRESS
      MOV R3,R4 :CURRENT RECORD SIZE
      SUB #256.,R4 :FIRST LOCATION IN BUFFER
173$:  ADD R2,R4 :SET POINTER TO FRAME (WORD)
      CMP (R4),R3 :CHECK DATA READ (R3=DATA ALSO)
      BEQ 180$ :BR, IF ALL IS WELL
      MOV (R4),R1 :RECD DATA
      MOV R3,R2 :EXPECTED DATA
      INC FATFLG :BUMP COUNT
      ERRHRD ERRNO,T24DTA,EXPREC :DATA READ NOT = WRITTEN
      TRAP CSERHRD
      .WORD 447
      .WORD T24DTA
      .WORD EXPREC
180$:  CKLOOP :LOOP IF SELECTED
      TRAP CSCLP1
      TST (R4)+ :BUMP TO NEXT LOCATION
      SUB R2,R4 :GET RID OF BASE ADDRESS
      CMP R4,R3 :END OF RECORD YET
      BNE 173$ :BR, IF NOT AT FND OF RECORD
      TST -(R3) :BUMP RECORD SIZE
      CMP #256.,R3 :END OF RECORD YET
      BNE 165$ :BR, IF MORE RECORDS TO WRITE
190$:  CKLOOP :LOOP IF SELECTED
      TRAP CSCLP1
      ENDSUB :>>>>>>>>>> END SUBTEST >>>>>>>>>
      L10060:
      TRAP CSSESUB
2999$:  CMP FATFLG,#15. :IS ERROR COUNT AT 25
      BLO 999$ :BR, IF LESS THAN 25
      JSR PC,CKDROP :TRY TO DROP THE UNIT

```



```

2817
2818      :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2819      :
2820      :*****
2821
2822 040244 004737 011104      JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
2823 040250 103407      BCS    30$           ;BR, IF NO PROBLEM
2824 040252 010001      MOV    R0,R1         ;SAVE TSSR
2825 040254 005237 002212      INC    FATFLG        ;BUMP COUNT
2829 040260      ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      040260 104456
      040262 000702      TRAP   CSERHRD
      040264 045046      .WORD 450
      040266 012136      .WORD T24RWN
2830 040270      30$:   CKLOOP      ;LOOP IF SELECTED      .WORD  PKTSSR
      040270 104406      TRAP   CSCLP1
2831 040272 012703 000400      MOV    #256.,R3      ;RECORD SIZE
2832 040276 013737 003114 043712  MOV    FREE,T24RB    ;STARTING WRITE BUFFER ADDRESS
2833
2834      :*****
2835      :WRITE DATA,ACK,CVC=1,SWB COMMAND
2836      :
2837      :*****
2838
2839
2840 040304 012737 150005 043710      MOV    #150005,T24PK3 ;WRITE DATA,ACK,CVC=1,SWB COMMAND
2841 040312 012704 043710      MOV    #T24PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
2842 040316      65$:
2843 040316 010300      MOV    R3,R0         ;SET PATTERN IN CORRECT REGISTER
2844 040320 004737 017512      JSR    PC,FILLMEM    ;FILL MEMORY WITH RECORD SIZE
2845 040324 010337 043716      MOV    R3,T24SZ     ;SET UP RECORD SIZE IN PACKET
2846 040330 010465 000000      MOV    R4,TSDB(R5)  ;ISSUE COMMAND
2847 040334 004737 016340      JSR    PC,WAITF      ;WAIT FOR SSR TO SET
2848 040340 016501 000002      MOV    TSSR(R5),R1  ;GET TSSR CONTENTS
2849 040344 012702 000200      MOV    #SSR,R2      ;SET UP EXPECTED
2850 040350 020102      CMP    R1,R2        ;ARE THEY EQUAL
2851 040352 001406      BEQ    75$          ;BR, IF OK
2852 040354 005237 002212      INC    FATFLG        ;BUMP COUNT
2856 040360      ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      040360 104456      TRAP   CSERHRD
      040362 000703      .WORD 451
      040364 005111      .WORD WRERR
      040366 012136      .WORD  PKTSSR
2857 040370      75$:   CKLOOP      ;LOOP IF SELECTED      TRAP   CSCLP1
      040370 104406
2858 040372 005723      TST    (R3)+         ;BUMP RECORD SIZE
2859 040374 022703 000414      CMP    #268.,R3     ;END OF RECORD YET
2860 040400 001346      BNE    65$          ;BR, IF MORE RECORDS TO WRITE
2861 040402      80$:   CKLOOP      ;LOOP IF SELECTED
      040402 104406      TRAP   CSCLP1
2862 040404 005743      TST    -(R3)        ;SET RECORD SIZE BACK TO 512.
2863 040406 013737 003114 043712  MOV    FREE,T24RB    ;STARTING READ BUFFER ADDRESS
2864
2865      :*****
2866      :READ REVERSE DATA,ACK,SWB COMMAND
2867      :
2868
    
```

```

2869
2870
2871 040414 012737 110401 043710
2872 040422 012704 043710 165$: MOV #110401,T24PK3 ;READ REVERSE DATA,ACK,SWB COMMAND
2873 040426 010337 043716 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2874 040432 010465 000000 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
2875 040436 004737 016340 MOV R4,T24S2 ;ISSUE COMMAND
2876 040442 016501 000002 JSR PC,WAITF ;WAIT FOR SSR TO SET
2877 040446 012702 000200 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
2878 040452 020102 MOV #SSR,R2 ;SET UP EXPECTED
2879 040454 001406 CMP R1,R2 ;ARE THEY EQUAL
2880 040456 005237 002212 BEQ 170$ ;BR, IF OK
2884 040462 INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T24WDC,EXPREC ;TSSR INCORRECT AFTER READ DATA
040462 104456 TRAP C$ERHRD
040464 000704 .WORD 452
040466 045376 .WORD T24WDC
040470 015564 .WORD EXPREC
2885 040472 170$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
040472 104406
2886 040474 013702 003114 MOV FREE,R2 ;GET BUFFER ADDRESS
2887 040500 010304 MOV R3,R4 ;CURRENT RECORD SIZE
2888 040502 162704 000400 SUB #256.,R4 ;FIRST LOCATION IN BUFFER
2889 040506 060204 173$: ADD R2,R4 ;SET POINTER TO FRAME (WORD)
2890 040510 021403 CMP (R4),R3 ;CHECK DATA READ (R3=DATA ALSO)
2891 040512 001410 BEQ 180$ ;BR, IF ALL IS WELL
2892 040514 011401 MOV (R4),R1 ;RECD DATA
2893 040516 010302 MOV R3,R2 ;EXPECTED DATA
2894 040520 005237 002212 INC FATFLG ;BUMP COUNT
2898 040524 ERRHRD ERRNO,T24DTA,EXPREC ;DATA READ NOT = WRITTEN
040524 104456 TRAP C$ERHRD
040526 000705 .WORD 453
040530 044630 .WORD T24DTA
040532 015564 .WORD EXPREC
2899 040534 180$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
040534 104406
2900 040536 005724 TST (R4)+ ;BUMP TO NEXT LOCATION
2901 040540 160204 SUB R2,R4 ;GET RID OF BASE ADDRESS
2902 040542 020403 CMP R4,R3 ;END OF RECORD YET
2903 040544 001360 BNE 173$ ;BR, IF NOT AT END OF RECORD
2904 040546 005743 TST -(R3) ;BUMP RECORD SIZE
2905 040550 022703 000400 CMP #256.,R3 ;END OF RECORD YET
2906 040554 001322 BNE 165$ ;BR, IF MORE RECORDS TO WRITE
2907 040556 190$: CKLOOP ;LOOP IF SELECTED
040556 104406 TRAP C$CLP1
2908 040560 ENDSUB ;>>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>
040560 L10061: TRAP C$ESUB
040560 104403
2909 040562 023727 002212 000017 CMP FATFLG,#15. ;IS ERROR COUNT AT 25
2910 040570 103402 BLO 999$ ;BR, IF LESS THAN 25
2911 040572 004737 017272 JSR PC,CKDROP ;TRY TO DROP THE UNIT
2912 040576 999$:
  
```

2914


```

2968
2969      :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2970      :
2971      :*****
2972
2973 040700 004737 011104      JSR      PC,REWIND      :CALL TAPE REWIND COMMAND
2974 040704 004737 016426      JSR      PC,CHKTSSR     :SEE HOW TSSR IS
2975 040710 103407              BCS      30$            :BR, IF NO PROBLEM
2976 040712 010001              MOV      P0,R1          :SAVE TSSR
2977 040714 005237 002212      INC      FATFLG         :BUMP COUNT
2981 040720              ERRHRD  ERRNO,T24RWN,PKTSSR :REWIND NOT ACCEPTED
      040720 104456
      040722 000710              TRAP     C$ERHRD
      040724 045046              .WORD   456
      040726 012136              .WORD   T24RWN
      .WORD   PKTSSR
2982 040730      30$:  CKLOOP              :LOOP IF SELECTED              TRAP     C$CLP1
      040730 104406
2983 040732 012703 001000      MOV      #512.,R3      :RECORD SIZE
2984 040736 013737 003114 043712  MOV      FREE,T24RB    :STARTING WRITE BUFFER ADDRESS
2985
2986      :*****
2987      :WRITE DATA,ACK,CVC=1 COMMAND
2988      :
2989      :*****
2990
2991
2992 040744 012737 140005 043710      MOV      #140005,T24PK3 :WRITE DATA,ACK,CVC=1 COMMAND
2993 040752 012704 043710      MOV      #T24PK3,R4    :SET UP R4 WITH PACKET ADDRESS
2994 040756      65$:
2995 040756 010337 043716      MOV      R3,T24SZ      :SET UP RECORD SIZE IN PACKET
2996 040762 010465 000000      MOV      R4,TSDB(R5)   :ISSUE COMMAND
2997 040766 004737 016340      JSR      PC,WAITF      :WAIT FOR SSR TO SET
2998 040772 016501 000002      MOV      TSSR(R5),R1   :GET TSSR CONTENTS
2999 040776 012702 000200      MOV      #SSR,R2       :SET UP EXPECTED
3000 041002 020102              CMP      R1,R2         :ARE THEY EQUAL
3001 041004 001406              BEQ     75$            :BR, IF OK
3002 041006 005237 002212      INC      FATFLG         :BUMP COUNT
3006 041012              ERRHRD  ERRNO,WRTErr,PKTSSR :TSSR INCORRECT AFTER WRITE DATA
      041012 104456              TRAP     C$ERHRD
      041014 000711              .WORD   457
      041016 005111              .WORD   WRTErr
      041020 012136              .WORD   PKTSSR
3007 041022      75$:  CKLOOP              :LOOP IF SELECTED              TRAP     C$CLP1
      041022 104406
3008 041024 012703 000400      MOV      #256.,R3      :SIZE OF RECORD
3009 041030 013737 003114 043712  MOV      FREE,T24RB    :STARTING READ BUFFER ADDRESS
3010
3011      :*****
3012      :READ DATA,ACK COMMAND
3013      :
3014      :*****
3015
3016
3017 041036 012737 100401 043710      MOV      #100401,T24PK3 :READ DATA,ACK COMMAND
3018 041044 012704 043710      MOV      #T24PK3,R4    :SET UP R4 WITH PACKET ADDRESS
3019 041050 010337 043716      MOV      R3,T24SZ      :SET UP RECORD SIZE IN PACKET
3020 041054 010465 000000      MOV      R4,TSDB(R5)   :ISSUE COMMAND
  
```


3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071

```
:+
: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 11/238 system with more than
:                               128K words of memory!
:*****
:-
```

```
3072 041170 BGNSUB ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
      041170 ;                               T4.9:
      041170 104402 TRAP CSBSUB
3073 041172 005737 003126 TST NXMFLG ;DO WE HAVE IT?
3074 041176 001002 BNE 10$ ;BR, IF ENOUGH
3075 041200 000137 041566 JMP 180$ ;SKIP THIS TEST IF NOT
3076 041204 004737 046404 10$: JSR PC,T24RT3 ;SET UP OTHER COMMAND PACKET
3077 041210 004737 046250 JSR PC,T24REST ;SET COMMAND PACKET
3078 041214 004737 046342 JSR PC,T24RT2 ;SET UP OTHER COMMAND PACKET
```

```
:*****
:ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
:*****
```

```
3085
3086 041220 004737 016064 JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
3087 041224 103407 BCS 20$ ;BR IF INIT WAS OK
3088 041226 005237 002212 INC FATFLG ;BUMP COUNT
3092 041232 010001 MOV R0,R1 ;CONTENTS OF TSSR REGISTER
3093 041234 104455 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      041236 000714 TRAP CSERDF
      041240 003650 .WORD 460
      041242 012124 .WORD SFIERR
3094 041244 013737 002172 043610 20$: MOV UNITN,T24DSW ;SET DRIVE NUMBER IN PACKET
3095 041244 013737 002172 043610 MOV #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
```

```
:*****
:WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
:*****
```

```
3103
3104 041256 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
3105 041262 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
3106 041264 005237 002212 INC FATFLG ;BUMP COUNT
3110 041270 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
3111 041272 104456 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      TRAP CSERHRD
```

```

041274 0G0715
041276 005054
041300 012124
3112 041302 24$: CKLOOP ;LOOP IF SELECTED
041302 104406 TRAP C$CLP1
3113
3114 :*****
3115 :
3116 :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
3117 :
3118 :*****
3119
3120 041304 004737 021276 JSR PC,INVERT ;INVERT THE EXTENDED FEATURES SWITCH
3121 041310 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
3122 041314 004737 016426 JSR PC,CHKTSSR ;SEE HOW TSSR IS
3123 041320 103407 BCS 30$ ;BR, IF NO PROBLEM
3124 041322 010001 MOV R0,R1 ;SAVE TSSR
3125 041324 005237 002212 INC FATFLG ;BUMP COUNT
3129 041330 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
041330 104456 TRAP C$ERHRD
041332 000716 .WORD 462
041334 045046 .WORD T24RWN
041336 012136 .WORD PKTSSR
3130 041340 30$: CKLOOP ;LOOP IF SELECTED
041340 104406 TRAP C$CLP1
3131 041342 012703 000400 MOV #256.,R3 ;RECORD SIZE
3132 041346 013737 003114 043712 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
3133 :*****
3134 :
3135 :WRITE DATA,ACK,CVC=1 COMMAND
3136 :
3137 :*****
3138
3139 041354 012737 140005 043710 MOV #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
3140 041362 012704 043710 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
3141 041366 65$:
3142 041366 010337 043716 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
3143 041372 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
3144 041376 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
3145 041402 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
3146 041406 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
3147 041412 020102 CMP R1,R2 ;ARE THEY EQUAL
3148 041414 001406 BEQ 75$ ;BR, IF OK
3149 041416 005237 002212 INC FATFLG ;BUMP COUNT
3153 041422 ERRHRD ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
041422 104456 TRAP C$ERHRD
041424 000717 .WORD 463
041426 005111 .WORD WRTErr
041430 012136 .WORD PKTSSR
3154 041432 75$: CKLOOP ;LOOP IF SELECTED
041432 104406 TRAP C$CLP1
3155 041434 012703 000400 MOV #256.,R3 ;RECORD SIZE
3156 041440 013737 003130 043712 MOV NXML0,T24RB ;STARTING READ BUFFER ADDRESS
3157 041446 013737 003132 043714 MOV NXMHI,T24RB+2 ;SET ADDRESS BITS 16-17
3158 :*****
3159 :
3160 :

```


3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3252
3253
3254
3255
3256

:*
:TEST 4, SUBTEST 10
:VERIFIES THAT ILLEGAL MODE-FIELD CODES IN THE READ
:COMMAND CAUSE A FUNCTION REJECT TERMINATION WITH THE
:ILLEGAL COMMAND (ILC) ERROR BIT SET.
:~

```
          BGNSUB                ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
                                           T4.10:
                                           TRAP      C$BSUB
3217 041606 041606 104402                JSR      PC,T24RT3    ;SET COMMAND PACKET UP CLEAR
041606                                JSR      PC,T24REST  ;SET COMMAND PACKET
3218 041610 004737 046404                JSR      PC,T24RT2  ;SET UP OTHER COMMAND PACKET
3219 041614 004737 046250
3220 041620 004737 046342
```

:*****
: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    466
                                           .WORD    SFIERR
                                           .WORD    SFIMSG
```

```
20$:
          MOV      UNITN,T24DSW        ;SET DRIVE NUMBER
          MOV      #T24PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
```

:*****
:WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
:*****

```
          JSR      PC,WRTCHR           ;ISSUE WRITE CHARACTERISTICS
          BCS      24$                ;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    467
                                           .WORD    WRTMSG
                                           .WORD    SFIMSG
```

```
24$:
          CKLOOP                        ;LOOP IF SELECTED
                                           TRAP      C$CLP1
          MOV      FREE,T24RB          ;STARTING WRITE BUFFER ADDRESS
```



```

3354
3355
3356
3357
3358
3359
3360
3361 042170 012757 140001 043710      MOV      #140001,T24PK3      ;LEGAL MODE,ACK,CVC=1,READ COMMAND
3362 042176 012704 043710              MOV      #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
3363 042202 012737 000400 043716      MOV      #256,T24SZ        ;SET UP RECORD SIZE IN PACKET
3364 042210 010465 000000              MOV      R4,T5DB(R5)       ;ISSUE COMMAND
3365 042214 004737 016340              JSR      PC,WAITF          ;WAIT FOR SSR!BIT1!BIT2 TO SET
3366 042220 016501 000002              MOV      T5SR(R5),R1       ;GET T5SR CONTENTS
3367 042224 012702 100206              MOV      #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
3368 042230 020102              CMP      R1,R2            ;ARE THEY EQUAL
3369 042232 001406              BEQ      75$              ;BR, IF CK
3370 042234 005237 002212              INC      FATFLG           ;BUMP COUNT
3374 042240              ERNHRD  ERRNC,T24WDG,PKTSSR ;T5SR INCORRECT AFTER READ DATA
                                TRAP      CSERHRD
                                .WORD      472
                                .WORD      T24WDG
                                .WORD      PKTSSR
3375 042250 104406 75$: CKLOOP                          ;LOOP IF SELECTED
                                TRAP      CSCLP1
                                .WORD      CSCLP1
3376
3377
3378
3379
3380
3381
3382
3383 042252 013701 043620              MOV      T24BFR+6,R1       ;GET MESSAGE BUFFER
3384 042256 010102              MOV      R1,R2            ;SET UP EXPECTED
3385 042260 052702 000400              BIS      #BIT8,R2         ;SET THE ILA BIT IN EXPECTED
3386 042264 020102              CMP      R1,R2            ;ARE THEY EQUAL
3387 042266 001406              BEQ      180$            ;BR, IF EQUAL (ALL IS WELL)
3388 042270 005237 002212              INC      FATFLG           ;BUMP COUNT
3392 042274              ERNHRD  ERRNO,T24ILA,EXPREC ;THE ILA BIT WAS NOT SET IN XSTO
                                TRAP      CSERHRD
                                .WORD      473
                                .WORD      T24ILA
                                .WORD      EXPREC
3393 042304 104406 180$: CKLOOP
                                TRAP      CSCLP1
                                .WORD      CSCLP1
3394 042306              ENDSUB                    ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
                                L10065:
                                TRAP      CSesub
                                .WORD      CSesub
3395 042310 023727 002212 000017      CMP      FATFLG,#15.      ;IS ERROR COUNT AT 25
3396 042316 103402              BLO      999$            ;BR, IF LESS THAN 25
3397 042320 004737 017272              JSR      PC,CKDROP        ;TRY TO DROP THE UNIT
3398 042324 999$:

```



```

3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502 042560                BGNSUB                      ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
      042560
      042560 104402            T4.13:
3503 042562 004737 046404   JSR      PC,T24RT3     ;SET COMMAND PACKET UP CLEAR TRAP CSBSUB
3504 042566 004737 046250   JSR      PC,T24REST    ;SET COMMAND PACKET
3505 042572 004737 046342   JSR      PC,T24RT2     ;SET UP OTHER COMMAND PACKET
3506
3507
3508
3509
3510
3511
3512
3513 042576 004737 016064   JSR      PC,SOFINIT    ;DO INITIALIZE ON CONTROLLER
3514 042602 103407          BCS      20$           ;BR IF INIT WAS OK
3515 042604 005237 002212   INC      FATFLG        ;BUMP COUNT
3519 042610 010001          MOV      RO,R1         ;CONTENTS OF TSSR REGISTER
3520 042612          ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      042612 104455          TRAP      CSERDF
      042614 000735          .WORD   477
      042616 003650          .WORD   SFIERR
      042620 012124          .WORD   SFIMSG
3521 042622
3522 042622 013737 002172 043610 20$:   MOV      UNITN,T24DSW  ;SET UP DRIVE NUMBER
3523 042630 012704 043570   MOV      #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
3524
3525
3526
3527
3528
3529
3530
3531 042634 004737 010752   JSR      PC,WRTCHR     ;ISSUE WRITE CHARACTERISTICS
3532 042640 103407          BCS      24$           ;BR, IF COMMAND ISSUED OK
3533 042642 005237 002212   INC      FATFLG        ;BUMP COUNT
3537 042646 010001          MOV      RO,R1         ;SAVE CONTENTS OF TSSR
3538 042650          ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      042650 104456          TRAP      CSERHRD
      042652 000736          .WORD   478
      042654 005054          .WORD   WRTMSG
      042656 012124          .WORD   SFIMSG
3539 042660          24$:   CKLOOP                       ;LOOP IF SELECTED
      042660 104406          TRAP      C$CLP1
3540
3541
  
```

```

3542
3543      :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
3544      :
3545      :*****
3546
3547 042662 004737 011104      JSR      PC,REWIND      :CALL TAPE REWIND COMMAND
3548 042666 004737 016426      JSR      PC,CHKTSSR     :SEE HOW TSSR IS
3549 042672 103407              BCS      30$            :BR, IF NO PROBLEM
3550 042674 010001              MOV      R0,R1         :SAVE TSSR
3551 042676 005237 002212      INC      FATFLG        :BUMP COUNT
3555 042702              EPRHRD   ERRNO,T24RWN,PKTSSR :REWIND NOT ACCEPTED
      042702 104456              TRAP    CSERHRD
      042704 000737              .WORD  479
      042706 045046              .WORD  T24RWN
      042710 012136              .WORD  PKTSSR
3556 042712      30$:      CKLOOP      :LOOP IF SELECTED      TRAP    CSCLP1
      042712 104406
3557 042714 012703 000400      MOV      #256,R3       :RECORD SIZE
3558 042720 013737 003114 043712  MOV      FREE,T24RB     :STARTING WRITE BUFFER ADDRESS
3559
3560      :*****
3561      :READ REVERSE DATA,ACK COMMAND
3562      :
3563      :*****
3564
3565
3566 042726 012737 100401 043710  MOV      #100401,T24PK3 :READ REVERSE DATA,ACK COMMAND
3567 042734 012704 043710      MOV      #T24PK3,R4    :SET UP R4 WITH PACKET ADDRESS
3568 042740      65$:
3569 042740 010337 043716      MOV      R3,T24SZ      :SET UP RECORD SIZE IN PACKET
3570 042744 010465 000000      MOV      R4,TSDB(R5)   :ISSUE COMMAND
3571 042750 004737 016340      JSR      PC,WAITF      :WAIT FOR SSR TO SET
3572 042754 016501 000002      MOV      TSSR(R5),R1   :GET TSSR CONTENTS
3573 042760 012702 100206      MOV      #SSR!SC!BIT1!BIT2,R2 :SET UP EXPECTED
3574 042764 020102              CMP      R1,R2         :ARE THEY, EQUAL
3575 042766 001406              BEQ      75$           :BR, IF OK
3576 042770 005237 002212      INC      FATFLG        :BUMP COUNT
3580 042774              ERRHRD   ERRNO,T24WDE,PKTSSR :TSSR INCORRECT AFTER READ DATA
      042774 104456              TRAP    CSERHRD
      042776 000740              .WORD  480
      043000 044511              .WORD  T24WDE
      043002 012136              .WORD  PKTSSR
3581 043004      75$:      CKLOOP      :LOOP IF SELECTED      TRAP    CSCLP1
      043004 104406
3582
3583      :*****
3584      :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
3585      :
3586      :*****
3587
3588
3589 043006 013701 043620      MOV      T24BFR+6,R1   :GET MESSAGE BUFFER
3590 043012 010102              MOV      R1,R2         :SET UP EXPECTED
3591 043014 052702 002000      BIS      #BIT10,R2     :SET THE NEF BIT IN EXPECTED
3592 043020 020102              CMP      R1,R2         :ARE THEY EQUAL
3593 043022 001406              BEQ      180$          :BR, IF EQUAL (ALL IS WELL)
3594 043024 005237 002212      INC      FATFLG        :BUMP COUNT
    
```

3598	043030				ERRHRD	ERRNO,T24NEF,EXPREC		;THE RLL BIT WAS NOT SET IN XSTO
	043030	104456						TRAP C\$ERHRD
	043032	000741						.WORD 481
	043034	043740						.WORD T24NEF
	043036	015564						.WORD EXPREC
3599	043040		180\$:	CKLOOP				TRAP C\$CLP1
	043040	104406						
3600	043042			ENDSUB				;>>>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>>>>
	043042							L10067:
3601	043044	104403						TRAP C\$ESUB
	043044	023727	002212	000017	CMP	FATFLG.#15.		;IS ERROR COUNT AT 25
3602	043052	103402			BLO	999\$;BR, IF LESS THAN 25
3603	043054	004737	017272		JSR	PC,CKDROP		;TRY TO DROP THE UNIT
3604	043060		999\$:					


```

3658
3659      :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
3660      :*****
3661
3662
3663 043162 004737 011104      JSR      PC,REWIND      :CALL TAPE REWIND COMMAND
3664 043166 004737 016426      JSR      PC,CHKTSSR     :SEE HOW TSSR IS
3665 043172 103407              BCS      30$            :BR, IF NO PROBLEM
3666 043174 010001              MOV      R0,R1         :SAVE TSSR
3667 043176 005237 002212      INC      FATFLG        :BUMP COUNT
3671 043202              ERRHRD   ERRNO,T24RWN,PKTSSR :REWIND NOT ACCEPTED
      043202 104456
      043204 000744              TRAP    C$ERHRD
      043206 045046              .WCRD   484
      043210 012136              .WORD   T24RWN
      043212              .WORD   PKTSSR
3672 043212              30$:   CKLOOP          :LOOP IF SELECTED
      043212 104406              TRAP    C$CLP1
3673 043214 012703 000400      MOV      #256.,R3      :RECORD SIZE
3674 043220 013737 003114 043712  MOV      FREE,T24RB    :STARTING WRITE BUFFER ADDRESS
3675
3676      :*****
3677      :WRITE DATA,ACK,CVC=1 COMMAND
3678      :*****
3679
3680
3681
3682 043226 012737 140005 043710      MOV      #140005,T24PK3 :WRITE DATA,ACK,CVC=1 COMMAND
3683 043234 012704 043710      MOV      #T24PK3,R4    :SET UP R4 WITH PACKET ADDRESS
3684 043240
3685 043240 010337 043716      65$:   MOV      R3,T24SZ  :SET UP RECORD SIZE IN PACKET
3686 043244 010465 000000      MOV      R4,TSDB(R5)   :ISSUE COMMAND
3687 043250 004737 016340      JSR      PC,WAITF      :WAIT FOR SSR TO SET
3688 043254 016501 000002      MOV      TSSR(R5),R1   :GET TSSR CONTENTS
3689 043260 012702 000200      MOV      #SSR,R2       :SET UP EXPECTED
3690 043264 020102              CMP      R1,R2         :ARE THEY EQUAL
3691 043266 001406              BEQ     75$            :BR, IF OK
3692 043270 005237 002212      INC      FATFLG        :BUMP COUNT
3696 043274              ERRHRD   ERRNO,WRERR,PKTSSR :TSSR INCORRECT AFTER READ DATA
      043274 104456              TRAP    C$ERHRD
      043276 000745              .WORD   485
      043300 005111              .WORD   WRERR
      043302 012136              .WORD   PKTSSR
3697 043304              75$:   CKLOOP          :LOOP IF SELECTED
      043304 104406              TRAP    C$CLP1
3698 043306 012703 000400      MOV      #256.,R3      :RECORD SIZE
3699 043312 013737 003114 043712  MOV      FREE,T24RB    :STARTING READ BUFFER ADDRESS
3700
3701      :*****
3702      :READ REVERSE DATA,ACK COMMAND
3703      :*****
3704
3705
3706
3707 043320 012737 100401 043710      165$:  MOV      #100401,T24PK3 :READ REVERSE DATA,ACK COMMAND
3708 043326 012704 043710      MOV      #T24PK3,R4    :SET UP R4 WITH PACKET ADDRESS
3709 043332 010337 043716      MOV      R3,T24SZ     :SET UP RECORD SIZE IN PACKET
3710 043336 010465 000000      MOV      R4,TSDB(R5)   :ISSUE COMMAND
    
```



```
60 043540 004737 017272                    JSR     PC,CKDROP                    ;TRY TO DROP THE UNIT
3761 043544                    999$:                   
3762                    :                   
3763                    :                   
3764                    :                   
3765 043544 004737 016546                    JSR     PC,TSTLOOP                    ;DO WE NEED TO ITERATE TEST
3766 043550 103002                    BCC     163$                    ;BR, IF NO LOOP REQUIRED
3767 043552 000137 034362                    JMP     T24LOOP                    ;EXECUTE AGAIN
3768 043556                    163$:                   
3769 043556                    EXIT     TST                    ;ALL DONE THIS TEST
         043556 104432                    TRAP     C$EXIT
         043560 002654                    .WORD     L10052-
```

```

3771
3772      ;+
3773      ;LOCAL STORAGE FOR THIS TEST
3774      ;-
3775      043570      .=<.+10>&177770
3777 043570      T24PACKET:      ;COMMAND PACKET FOR TEST
3778 043570 100204      .WORD 100204      ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
3779 043572 043600      .WORD T24DATA      ;ADDRESS OF CHARACTERISTICS BLOCK
3780 043574 000000      .WORD 0
3781 043576 000012      .WORD 10.      ;STARTING VALUE OF BLOCK SIZE
3782 043600      T24DATA:      ;CHARACTERISTICS DATA BLOCK
3783 043600 043612      .WORD T24BFR      ;ADDRESS OF MESSAGE BUFFER
3784 043602 000000      .WORD 0
3785 043604 000024      .WORD 20.      ;LENGTH OF MESSAGE BUFFER
3786 043606 000000      .WORD 0
3787 043610 000000      T24DSW: .WORD 0      ;DRIVE SELECTION BITS 2-0
3788 043612      T24BFR: .BLKW 25.      ;MESSAGE BUFFER
3789
3790      ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
3791
3793      043700      .=<..10>&177770
3795 043700      T24PK2:      ;WRITE SUB SYS MEM COMMAND, IE AND ACK
3796 043700 100206      .WORD 100206      ;ADDRESS OF SELECT BLOCK DATA
3797 043702 043720      .WORD T24BF2
3798 043704 000000      .WORD 0
3799 043706 000006      .WORD 6.      ;SIZE OF DATA PACKET
3800
3804 043710      T24PK3:      ;READ COMMAND, IE AND ACK
3805 043710 100205      .WORD 100205
3806 043712      T24RB:      ;ADDRESS OF WRITE BUFFER
3807 043712 003114      T24WB: .WORD FREE
3808 043714 000000      .WORD 0
3809 043716 000000      T24SZ: .WORD 0      ;SIZE OF BUFFER (EXTENT)
3810
3811      .EVEN
3812
3813
3814 043720      T24BF2:
3815 043720 010      T24BS0: .BYTE 10      ;BSEL0 AREA
3816 043721 200      T24BS1: .BYTE 200      ;BSEL1 AREA
3817 043722 000000      T24S2: .WORD 0      ;SEL 2 AREA
3818 043724 000000      T24S3: .WORD 0      ;DATA AREA
3819
3820
3821      .EVEN
3822      ;TAPE MOTION PACKET COMMAND VALUES
3823
3824 043726 100005      T24RN: .WORD 100005      ;READ DATA (NEXT)
3825 043730 100405      T24WDR: .WORD 100405      ;READ DATA RETRY
3826 043732 102005      T24CON: .WORD 102005      ;WRITE CONTINUOUS
3827 043734 177777      .WORD 177777      ;END OF DATA
3828 043736 000000      T24DLY: .WORD 0      ;DELAY STORAGE AREA
3829
3830

```

```

3832
3833
3834          ;+
3835          ;LOCAL TEXT MESSAGES FOR TEST
3836          ;-
3837 043740    116    105    106  T24NEF: .ASCIZ 'NEF Not Set After NON-EXECUTABLE FUNCTION'
3838 044012    122    111    102  T24LOR: .ASCIZ 'RIB Not Set After READ REVERSE Into BOT'
3839 044062    124    123    123  T24WDG: .ASCIZ 'TSSR Not Correct After Illegal Buffer Address Bits Set'
3840 044151    124    123    123  T24NXM: .ASCIZ 'TSSR Not Correct After NXM Memory Address In Packet'
3841 044235    124    123    123  T24WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
3842 044312    111    154    154  T24ILA: .ASCIZ 'Illegal Address Bits, Failed To Set ILA Bit In XST0'
3843 044376    111    154    154  T24LOO: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
3844 044457    122    105    101  T24SSR: .ASCIZ 'READ COMMAND Not Accepted'
3845 044511    124    123    123  T24WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
3846 044563    124    141    160  T24BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
3847 044630    104    141    164  T24DTA: .ASCIZ 'Data Written To Tape Not Equal To Data Read From Tape'
3848 044716    122    105    101  T24EOT: .ASCIZ 'READ DATA OVER EOT GAVE NO TAPE STATUS ALERT'
3849 044773    124    123    123  T24TM: .ASCIZ 'TSSR Not Correct After READ COMMAND Reject'
3850 045046    122    145    167  T24RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
3851 045115    122    101    115  T24RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
3852 045170    124    123    123  T24AM3: .ASCIZ 'TSSR Init. Failed After READ COMMAND'
3853 045235    104    162    151  T24OFL: .ASCIZ 'Drive 7 Select Failed To Set 'OFL' In TSSR'
3854 045310    124    23     123  T24WDD: .ASCIZ 'TSSR Not Correct After READ DATA Command, SWB Bit Set'
3855 045376    124    123    123  T24WDC: .ASCIZ 'TSSR Not Correct After READ DATA Command'
3856 045447    103    126    103  T24VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
3857 045522    124    123    102  T24BA: .ASCIZ 'TSBA Not Correct After READ DATA Command'
3858 045573    127    122    111  T24WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
3859 045662    122    145    141  T24LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
3860 045744    122    145    141  T24LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
3861 046026    122    145    163  T24PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
3862 046114    122    145    141  T24TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
3863 046202    *02    141    163  TST24ID: .ASCIZ 'Basic Read Data (Forward and Reverse)'

```

```

3864          .EVEN
3865          ;+
3866          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3867          ;WRITE SUBSYSTEM MEMORY COMMAND
3868          ;-
3869          ;-
3870          ;-
3871          ;-

```

```

3872 046250    T24REST:
3873 046250          SAVREG          ;SAVE THE REGISTERS
3874 046254    012701 043570    MOV #T24PACKET,R1          ;START OF THE PACKET
3875 046260    012721 100004    MOV #100004,(R1)+        ;WRITE SUBSYSTEM MEM. WITH ACK,
3876 046264    012721 043600    MOV #T24DATA,(R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
3877 046270    005021          CLR (R1)+                ;EXTENDED ADDRESS
3878 046272    012721 000012    MOV #10,(R1)+           ;SIZE OF DATA BLOCK IN BYTES
3879 046276    012721 0436.2    MOV #T24BFR,(R1)+       ;ADDRESS OF MESSAGE BUFFER
3880 046302    005021          CLR (R1)+
3881 046304    012721 000024    MOV #20,(R1)+           ;LENGTH OF MESSAGE BUFFER
3882 046310    005021          CLR (R1)+
3883 046312    012711 000000    MOV #0,(R1)              ;SELECT DRIVE ZERO
3884 046316    012702 000030    MOV #24,R2               ;NUMBER OF LOCATIONS TO BE CLEARED
3885 046322    012762 177777 043612 64$: MOV #177777,T24BFR(R2)   ;ALL ONES TO MESSAGE BUFFER
3886 046330    005742          TST -(R2)                ;NEXT LOCATION
3887 046332    022702 000000    CMP #0,R2                ;CHECK FOR END OF LOOP
3888 046336    001371          BNE 64$                  ;KEEP GOING UNTIL DONE

```

3889	046340	000207		RTS	PC		:RETURN
3890							
3891							
3892	046342			T24RT2:			
3893	046342				SAVREG		:SAVE THE REGISTERS
3894	046346	012701	043700		MOV #T24PK2,R1		:START OF THE PACKET
3895	046352	012721	100206		MOV #100206,(R1)+		:WRITE SUBSYSTEM MEM. WITH ACK, IE
3896	046356	012721	043720		MOV #T24BF2,(R1)+		:ADDRESS OF DATA BLOCK
3897	046362	005021			CLR (R1)+		:EXTENDED ADDRESS
3898	046364	012721	000006		MOV #6,(R1)+		:SIZE OF DATA BLOCK IN BYTES
3899	046370	005021			CLR (R1)+		
3900	046372	012701	043720		MOV #T24BF2,R1		:POINT TO DATA SEL AREA
3901	046376	005021			CLR (R1)+		
3902	046400	005011			CLR (R1)		
3903	046402	000207			RTS	PC	:RETURN
3904	046404						
3905	046404				T24RT3:		
3906	046410	012701	043710		SAVREG		:SAVE THE REGISTERS
3907	046414	012721	000000		MOV #T24PK3,R1		:START OF THE PACKET
3908	046420	012721	000000		MOV #0,(R1)+		:CLEAR AREA OUT
3909	046424	005021			MOV #0,(R1)+		:ADDRESS OF DATA BLOCK
3910	046426	012711	000000		CLR (R1)+		:EXTENDED ADDRESS
3911	046432	000207			MOV #0,(R1)		:SIZE OF DATA BLOCK IN BYTES
3912	046434				RTS	PC	:RETURN
	046434				ENDTST		
	046434	104401					L10052: TRAP C\$ETST


```
3972 046516 004737 016064      5$:   JSR      PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
3973 046522 103427                BCS      10$                ;BR IF INIT WAS OK
3974 046524                DELAY    250                ;DELAY IF REQUIRED
                                MOV      #250,(PC)+
                                .WORD    0
                                MOV      L$DLY,(PC)+
                                .WORD    0
                                DEC      -6(PC)
                                BNE     .-4
                                DEC      -22(PC)
                                BNE     .-20
3975 046554 005337 054002                DEC      T25DLY            ;DEC DELAY COUNTER
3976 046560 001356                BNE     5$                  ;BR, IF LOOP IS REQUIRED
3977 046562 005237 002212                INC      FATFLG            ;BUMP COUNT
3981 046566 016501 000002                MOV      TSSR(R5),R1      ;CONTENTS OF TSSR REGISTER
3982 046572                ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                TRAP    C$ERDF
                                .WORD    501
                                .WORD    SFIERR
                                .WORD    SFIMSG
3983 046602                10$:
3984 046602 013737 002172 053650   MOV      UNITN,T25DSW      ;SET UP DRIVE NUMBER
3985 046610 012704 053630   MOV      #T25PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
3986
3987   :*****
3988   :
3989   :WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
3990   :
3991   :*****
3992
3993 046614 004737 010752                JSR      PC,WRTCHR         ;ISSUE WRITE CHARACTERISTICS
3994 046620 103407                BCS      15$                ;BR, IF COMMAND ISSUED OK
3995 046622 005237 002212                INC      FATFLG            ;BUMP COUNT
3999 046626 010001                MOV      R0,R1             ;SAVE CONTENTS OF TSSR
4000 046630                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP    C$ERHRD
                                .WORD    502
                                .WORD    WRTMSG
                                .WORD    SFIMSG
4001
4002   :*****
4003   :
4004   :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
4005   :
4006   :*****
4007
4008 046640                15$:   CKLOOP
4009 046642 004737 011104                JSR      PC,REWIND        ;CALL TAPE REWIND COMMAND
4010 046646 103407                BCS      30$                ;BR, IF NO PROBLEM
4011 046650 010001                MOV      R0,R1             ;SAVE TSSR
4012 046652 005237 002212                INC      FATFLG            ;BUMP COUNT
4016 046656                ERRHRD  ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD    503
                                .WORD    T25RWN
                                .WORD    PKTSSR
```

```

4017 046666          30$:  CKLOOP                      ;LOOP IF SELECTED
      046666 104406                                TRAP  C$CLP1
4018
4019
4020
4021
4022
4023
4024
4025 046670 013701 053660          MOV  T25BFR+6,R1      ;PICK UP XSTO
4026 046674 010102          MOV  R1,R2           ;SET UP EXPECTED
4027 046676 052702 000002          BIS  #BIT1,R2        ;SET BOT BIT IN EXPECTED
4028 046702 020102          CMP  R1,R2           ;DOES EXP = REC'D
4029 046704 001406          BEQ  40$             ;BR, IF EQUAL (OK)
4030 046706 005237 002212          INC  FATFLG          ;BUMP COUNT
4034 046712          ERRHRD ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      046712 104456                                TRAP  C$ERHRD
      046714 000770                                .WORD 504
      046716 054145                                .WORD T25BOT
      046720 015564                                .WORD EXPREC
4035 046722          40$:  CKLOOP                      ;LOOP IF SELECTED
      046722 104406                                TRAP  C$CLP1
4036 046724 012703 000400          MOV  #256.,R3        ;RECORD SIZE
4037 046730 013737 003114 053752  MOV  FREE,T25RB      ;STARTING WRITE BUFFER ADDRESS
4038
4039
4040
4041
4042
4043
4044
4045 046736 012737 140005 053750  MOV  #140005,T25PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
4046 046744 012704 053750          MOV  #T25PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
4047 046750          65$:
4048 046750 010337 053756          MOV  R3,T25SZ        ;SET UP RECORD SIZE IN PACKET
4049 046754 013777 054000 134132  MOV  T25CNT,@FREE    ;LOAD UP RECORD COUNTER IN WRT BUFFER
4050 046762 062737 000001 054000  ADD  #1,T25CNT        ;GET READY FOR NEXT RECORD
4051 046770 010465 000000          MOV  R4,TSDB(R5)     ;ISSUE COMMAND
4052 046774 004737 016340          JSR  PC,WAITF        ;WAIT FOR SSR TO SET
4053 047000 016501 000002          MOV  TSSR(R5),R1     ;GET TSSR CONTENTS
4054 047004 012702 000200          MOV  #SSR,R2         ;SET UP EXPECTED
4055 047010 020102          CMP  R1,R2           ;ARE THEY EQUAL
4056 047012 001411          BEQ  75$             ;BR, IF OK
4057 047014 032701 000004          BIT  #BIT2,R1        ;CHECK FOR TAPE STATUS ALERT
4058 047020 001014          BNE  120$            ;BR, IF TSA IS SET (SUSPECT IS EOT)
4059 047022 005237 002212          INC  FATFLG          ;BUMP COUNT
4063 047026          ERRHRD ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      047026 104456                                TRAP  C$ERHRD
      047030 000771                                .WORD 505
      047032 005111                                .WORD WRTERR
      047034 012136                                .WORD PKTSSR
4064 047036          75$:  CKLOOP                      ;LOOP IF SELECTED
      047036 104406                                TRAP  C$CLP1
4065 047040 005203          INC  R3              ;BUMP RECORD SIZE
4066 047042 022703 001000          CMP  #512.,R3       ;END OF RECORD YET
4067 047046 001340          BNE  65$             ;BR, IF MORE RECORDS TO WRITE
4068 047050 000415          BR   125$           ;ENOUGH RECORDS
  
```


4069 047052
4070
4071
4072
4073
4074
4075
4076
4077 047052 013701 053660
4078 047056 010102
4079 047060 052702 000001
4080 047064 020102
4081 047066 001406
4082 047070 005237 002212
4086 047074
047074 104455
047076 000772
047100 054301
047102 015564

120\$:

:READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)

MOV T25BFR+6,R1 ;QUICK CHECK FOR EOT SET
MOV R1,R2 ;SET UP EXPECTED
BIS #BIT0,R2 ;SET THE EOT BIT XSTO
CMP R1,R2 ;IS THE EOT BIT SET IN XSTO
BEQ 125\$;BR, IF SET (GOOD)
INC FATFLG ;BUMP COUNT
ERRDF ERRNO,T25NET,EXPREC ;DEVICE FATAL NOT EOT FOUND ETC.
TRAP C\$ERDF
.WORD 506
.WORD T25NET
.WORD EXPREC

4087 047104
4088
4089
4090
4091
4092
4093
4094

125\$:

:ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE

4095 047104 004737 011104
4096 047110 103407
4097 047112 010001
4098 047114 005237 002212
4102 047120
047120 104456
047122 000773
047124 054755
047126 012136

JSR PC,REWIND ;CALL TAPE REWIND COMMAND
BCS 130\$;BR, IF NO PROBLEM
MOV R0,R1 ;SAVE TSSR
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
TRAP C\$ERHRD
.WORD 507
.WORD T25RWN
.WORD PKTSSR

4103 047130
047130 104406
4104 047132 012737 000007 053650
4105 047140 012704 053630
4106
4107
4108
4109
4110
4111
4112

130\$:

CKLOOP ;LOOP IF SELECTED TRAP C\$CLP1
MOV #7,T25DSW ;SET UP DRIVE NUMBER
MOV #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS

4113 047144 004737 010752
4114 047150 103407
4115 047152 005237 002212
4119 047156 010001
4120 047160
047160 104456
047162 000774
047164 005054
047166 012124

:WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)

JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
BCS 140\$;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 508
.WORD WRTMSG
.WORD SFIMSG

4121 047170

140\$:

CKLOOP ;SCOPE LOOP

```

047170 104406
4122 047172 005737 002216          TST      EXTFEA          ;CHECK FOR EXTENDED FEATURES          TRAP      C$CLP1
4123 047176 001044          BNE      160$          ;BR IF SWITCH IS ON
4124
4125 047200 112737 000200 053761      MOV      #200,T25BS1    ;WRITE MISCELLANEOUS CONT/READ STATUS
4126 047206 112737 000010 053760      MOV      #10,T25BS0    ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4127 047214 012704 053740      MOV      #T25PK2,R4    ;WRITE SUBSYS MEM PACKET
4128 047220 010465 000000      MOV      R4,T25DB(R5)  ;ISSUE COMMAND
4129 047224 004737 016426      JSR      PC,CHKTSSR    ;WAIT FOR SSR
4130 047230 103407      BCS      150$          ;BR, IF NO ERROR
4131 047232 010001      MOV      R0,R1         ;ERROR, SAVE TSSR
4132 047234 005237 002212      INC      FATFLG        ;BUMP COUNT
4136 047240          ERRHRD  ERRNO,T25SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
                                TRAP      C$ERHRD
                                .WORD    509
                                .WORD    T25SSR
                                .WORD    PKTSSR
                                TRAP      C$CLP1
047240 104456
047242 000775
047244 054004
047246 012136
4137 047250          150$:  CKLOOP          ;LOOP IF SELECTED
047250 104406
4138 047252 012737 000007 053650      MOV      #7,T25DSW    ;SET UP DRIVE NUMBER
4139 047260 012704 053630      MOV      #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4140
4141
4142
4143
4144
4145
4146
4147 047264 004737 010752      JSR      PC,WRTCHR     ;ISSUE WRITE CHARACTERISTICS
4148 047270 103407      BCS      160$          ;BR, IF COMMAND ISSUED OK
4149 047272 005237 002212      INC      FATFLG        ;BUMP COUNT
4153 047276 010001      MOV      R0,R1         ;SAVE CONTENTS OF TSSR
4154 047300          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP      C$ERHRD
                                .WORD    510
                                .WORD    WRTMSG
                                .WORD    SFIMSG
047300 104456
047302 000776
047304 005054
047306 012124
4155 047310          160$:  CKLOOP          ;SCOPE LOOP
047310 104406
4156 047312 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
4157 047316 032701 000100      BIT      #OFL,R1      ;CHECK FOR THE OFFLINE BIT SET
4158 047322 001006      BNE      170$          ;BR, IF OFFLINE (GOOD)
4159 047324 005237 002212      INC      FATFLG        ;BUMP COUNT
4163 047330          ERRDF  ERRNO,T25OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
                                TRAP      C$ERDF
                                .WORD    511
                                .WORD    T25OFL
                                .WORD    SFIMSG
047330 104455
047332 000777
047334 055024
047336 012124
4164 047340          170$:  CKLOOP          ;LOOP IF SELECTED
047340 104406
                                TRAP      C$CLP1
4165
4166
4167
4168
4169
4170
4171
:*****
:SPACE FORWARD COMMAND IN PLACE
:*****
```



```

4246 ;*****
4247
4248 047542 15$: CKLOOP
      047542 104406
4249 047544 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND TRAP C$CLP1
4250 047550 103407 BCS 30$ ;BR, IF NO PROBLEM
4251 047552 010001 MOV R0,R1 ;SAVE TSSR
4252 047554 005237 002212 INC FATFLG ;BUMP COUNT
4256 047560 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      047560 104456
      047562 001003 TRAP C$ERHRD
      047564 054755 .WORD 515
      047566 012136 .WORD T25RWN
      .WORD PKTSSR
4257 047570 30$: CKLOOP ;LOOP IF SELECTED
      047570 104406 TRAP C$CLP1
4258 047572 005737 002216 140$: TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
4259 047576 001044 BNE 160$ ;BR IF SWITCH IS ON
4260
4261 047600 112737 000200 053761 MOVB #200,T25BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
4262 047606 112737 000010 053760 MOVB #10,T25BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4263 047614 012704 053740 MOV #T25PK2,R4 ;WRITE SUBSYS MEM PACKET
4264 047620 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
4265 047624 004737 016426 JSR PC,CHKTSSR ;WAIT FOR SSR
4266 047630 103407 BCS 150$ ;BR, IF NO ERROR
4267 047632 010001 MOV R0,R1 ;ERROR, SAVE TSSR
4268 047634 005237 002212 INC FATFLG ;BUMP COUNT
4272 047640 ERRHRD ERRNO,T25SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      047640 104456 TRAP C$ERHRD
      047642 001004 .WORD 516
      047644 054004 .WORD T25SSR
      047646 012136 .WORD PKTSSR
4273 047650 150$: CKLOOP ;LOOP IF SELECTED
      047650 104406 TRAP C$CLP1
4274 047652 012737 000007 053650 MOV #7,T25DSW ;SET UP DRIVE NUMBER
4275 047660 012704 053630 MOV #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4276
4277 ;*****
4278 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
4279 ;*****
4280
4281
4282
4283 047664 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
4284 047670 103407 BCS 160$ ;BR, IF COMMAND ISSUED OK
4285 047672 005237 002212 INC FATFLG ;BUMP COUNT
4289 047676 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
4290 047700 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      047700 104456 TRAP C$ERHRD
      047702 001005 .WORD 517
      047704 005054 .WORD WRTMSG
      047706 012124 .WORD SFIMSG
4291 047710 160$: CKLOOP ;SCOPE LOOP
      047710 104406 TRAP C$CLP1
4292 047712 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4293 047716 032701 000100 BIT #OFL,R1 ;CHECK FOR THE OFFLINE BIT SET
4294 047722 001006 BNE 170$ ;BR, IF OFFLINE (GOOD)
4295 047724 005237 002212 INC FATFLG ;BUMP COUNT

```



```

4382
4383 050142          15$:  CKLOOP
      050142 104406
4384 050144 004737 011104      JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND      TRAP  C$CLP1
4385 050150 103407          BCS    30$            ;BR, IF NO PROBLEM
4386 050152 010001          MOV    R0,R1          ;SAVE TSSR
4387 050154 005237 002212      INC    FATFLG         ;BUMP COUNT
4391 050160          ERRHRD  ERRNG,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      050160 104456
      050162 001012          TRAP  C$SERHRD
      050164 054755          .WORD 522
      050166 012136          .WORD T25RWN
4392 050170          30$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      050170 104406
4393
4394 :*****
4395 :
4396 :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
4397 :
4398 :*****
4399
4400 050172 013701 053660      MOV    T25BFR+6,R1   ;PICK UP XSTO
4401 050176 010102          MOV    R1,R2         ;SET UP EXPECTED
4402 050200 052702 000002      BIS    #BIT1,R2     ;SET BOT BIT IN EXPECTED
4403 050204 020102          CMP    R1,R2        ;DOES EXP = REC'D
4404 050206 001406          BEQ    40$          ;BR, IF EQUAL (OK)
4405 050210 005237 002212      INC    FATFLG         ;BUMP COUNT
4409 050214          ERRHRD  ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      050214 104456          TRAP  C$SERHRD
      050216 001013          .WORD 523
      050220 054145          .WORD T25BOT
      050222 015564          .WORD EXPREC
4410 050224          40$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      050224 104406
4411 050226 012737 000001 053752  MOV    #000001,T25RB ;NUMBER OF RECORDS TO SPACE OVER
4412
4413 :*****
4414 :
4415 :SPACE FORWARD,ACK,CVC=1 COMMAND
4416 :
4417 :*****
4418
4419 050234 012737 140010 053750  MOV    #140010,T25PK3 ;SPACE FORWARD,ACK,CVC=1 COMMAND
4420 050242 012704 053750      MOV    #T25PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
4421 050246          65$:
4422 050246 010465 000000      MOV    R4,T5DB(R5)  ;ISSUE COMMAND
4423 050252 004737 016340      JSR    PC,WAITF     ;WAIT FOR SSR TO SET
4424 050256 016501 000002      MOV    TSSR(R5),R1 ;GET TSSR CONTENTS
4425 050262 012702 000200      MOV    #SSR,R2     ;SET UP EXPECTED
4426 050266 020102          CMP    R1,R2        ;ARE THEY EQUAL
4427 050270 001411          BEQ    75$          ;BK, IF OK
4428 050272 032701 000004      BIT    #BIT2,R1     ;CHECK FOR TAPE STATUS ALERT
4429 050276 001006          BNE    75$          ;BR, IF TSA IS SET (SUSPECT IS EOT)
4430 050300 005237 002212      INC    FATFLG         ;BUMP COUNT
4434 050304          ERRHRD  ERRNO,T25WDE,EXPREC ;TSSR INCORRECT AFTER READ DATA
      050304 104456          TRAP  C$SERHRD
      050306 001014          .WORD 524

```



```

050310 054065 .WORD T25WDE
050312 015564 .WORD EXPREC
4435 050314 75$: CKLOOP ;LOOP IF SELECTED
050314 104406 TRAP C$CLP1
4436 050316 120$:
4437
4438
4439
4440 :*****
4441 :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
4442 :*****
4443
4444 050316 013701 053660 MOV T25BFR+6,R1 ;QUICK CHECK FOR BOT SET
4445 050322 010102 MOV R1,R2 ;SET UP EXPECTED
4446 050324 042702 000002 BIC #BIT1,R2 ;CLEAR THE BOT BIT (XSTO)
4447 050330 020102 CMP R1,R2 ;IS THE EOT BIT SET IN XSTO
4448 050332 001406 BEQ 125$ ;BR, IF SET (GOOD)
4449 050334 005237 002212 INC FATFLG ;BUMP COUNT
4453 050340 ERRHRD ERRNO,T25BNC,EXPREC ;BOT NOT CLEARED AFTER SPACE FROM BOT
050340 104456 TRAP C$ERHRD
050342 001015 .WORD 525
050344 054440 .WORD T25BNC
050346 015564 .WORD EXPREC
4454 050350 25$: CKLOOP TRAP C$CLP1
050350 104406
4455 050352 004737 055322 JSR PC,T25RT3 ;CLEAN UP PACKET
4456 050356 012737 000401 053756 MOV #257.,T25SZ ;SET THE CORRECT SIZE UP
4457
4458 :*****
4459 :READ DATA COMMAND IN PLACE
4460 :*****
4461
4462
4463
4464 050364 012737 140001 053750 MOV #140001,T25PK3 ;READ DATA COMMAND IN PLACE
4465 050372 013737 003114 053752 MOV FREE,T25RB ;READ BUFFER ADDRESS TO PACKET
4466 050400 012704 053750 MOV #T25PK3,R4 ;R4 = POINTER TO PACKET
4467 050404 010465 000000 MOV R4,T5DB(R5) ;ISSUE COMMAND
4468 050410 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
4469 050414 016501 000002 MOV T5SR(R5),R1 ;GET T5SR CONTENTS
4470 050420 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
4471 050424 020102 CMP R1,R2 ;ARE THEY EQUAL
4472 050426 001406 BEQ 190$ ;BR, IF OK ESP. FUNCTION REJECT
4473 050430 005237 002212 INC FATFLG ;BUMP COUNT
4477 050434 ERRHRD ERRNO,RDERR,PKT5SR ;T5SR INCORRECT AFTER READ DATA CMD
050434 104456 TRAP C$ERHRD
050436 001016 .WORD 526
050440 005204 .WORD RDERR
050442 012136 .WORD PKT5SR
4478 050444 190$: CKLOOP ;LOOP IF SELECTED
050444 104406 TRAP C$CLP1
4479 050446 017701 132442 MOV @FREE,R1 ;GET FIRST WORD FROM BUFFER
4480 050452 012702 000001 MOV #1,R2 ;SET UP EXPECTED
4481 050456 020102 CMP R1,R2 ;WAS RECORD NUMBERED 1
4482 050460 001406 BEQ 200$ ;BR, IF CORRECT RECORD
4483 050462 005237 002212 INC FATFLG ;BUMP COUNT
4487 050466 ERRHRD ERRNO,T25WNG,EXPREC ;SHOULD HAVE BEEN RECORD NUMBER 1

```



```
4495  
4496  
4497  
4498  
4499  
4500  
4501  
4502  
4503  
4504  
4505 050516          BGNSUB          :>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>  
050516          TS.4:  
050516 104402          TRAP          C$BSUB  
4506 050520 004737 055166      JSR          PC,T25REST      ;SET COMMAND PACKET  
4507 050524 004737 055260      JSR          PC,T25RT2      ;SET UP OTHER COMMAND PACKET  
4508 050530 004737 055322      JSR          PC,T25RT3      ;SET UP OTHER COMMAND PACKET  
4509  
4510  
4511  
4512  
4513  
4514  
4515  
4516 050534 004737 016064      JSR          PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER  
4517 050540 103407          BCS          20$            ;BR IF INIT WAS OK  
4518 050542 005237 002212      INC          FATFLG         ;BUMP COUNT  
4522 050546 010001          MOV          R0,R1          ;CONTENTS OF TSSR REGISTER  
4523 050550          ERRDF          ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK  
050550 104455          TRAP          C$ERDF  
050552 001020          .WORD        528  
050554 003650          .WORD        SFIERR  
050556 012124          .WORD        SFIMSG  
4524 050560 013737 002172 053650 20$:  MOV          UNITN,T25DSW      ·SET UP DRIVE NUMBER  
4525  
4526 050566 012704 053630      MOV          #T25PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS  
4527  
4528  
4529  
4530  
4531  
4532  
4533  
4534 050572 004737 010752      JSR          PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS  
4535 050576 103407          BCS          25$            ;BR, IF COMMAND ISSUED OK  
4536 050600 005237 002212      INC          FATFLG         ;BUMP COUNT  
4540 050604 010001          MOV          R0,R1          ;SAVE CONTENTS OF TSSR  
4541 050606          ERRHRD          ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED  
050606 104456          TRAP          C$ERHRD  
050610 001021          .WORD        529  
050612 005054          .WORD        WRTMSG  
050614 012124          .WORD        SFIMSG  
4542 050616          25$:  CKLOOP          ;LOOP IF SELECTED  
050616 104406          TRAP          C$CLP1  
4543  
4544  
4545  
4546  
:*****  
:ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
```

```
4547  
4548  
4549  
4550 050620 004737 011104      JSR    PC,REWIND      :CALL TAPE REWIND COMMAND  
4551 050624 103407      BCS    30$            :BR, IF NO PROBLEM  
4552 050626 010001      MOV    R0,R1          :SAVE TSSR  
4553 050630 005237 002212      INC    FATFLG         :BUMP COUNT  
4557 050634      ERRHRD  ERRNO,T25RWN,PKTSSR :REWIND NOT ACCEPTED  
                                TRAP    C$ERHRD  
                                .WORD  530  
                                .WORD  T25RWN  
                                .WORD  PKTSSR  
4558 050644 30$:      CKLOOP                :LOOP IF SELECTED  
050644 104406      TRAP    C$CLP1  
4559  
4560  
4561  
4562  
4563  
4564  
4565  
4566 050646 013701 053660      MOV    T25BFR+6,R1   :PICK UP XSTO  
4567 050652 010102      MOV    R1,R2         :SET UP EXPECTED  
4568 050654 052702 000002      BIC    #BIT1,R2      :SET BOT BIT IN EXPECTED  
4569 050660 020102      CMP    R1,R2         :DOES EXP = REC'D  
4570 050662 001406      BEQ    40$           :BR, IF EQUAL (OK)  
4571 050664 005237 002212      INC    FATFLG         :BUMP COUNT  
4575 050670      ERRHRD  ERRNO,T25BOT,EXPREC :TAPE NOT AT BOT AFTER REWIND  
                                TRAP    C$ERHRD  
                                .WORD  531  
                                .WORD  T25BOT  
                                .WORD  EXPREC  
4576 050700 40$:      CKLOOP                :LOOP IF SELECTED  
050700 104406      TRAP    C$CLP1  
4577  
4578  
4579  
4580  
4581  
4582  
4583  
4584  
4585 050702 012703 000001      MOV    #000001,R3    :NUMBER OF RECORDS TO SPACE FORWARD  
4586 050706 004737 010556      JSR    PC,SPACE      :CALL SPACE COMMAND  
4587 050712 103410      BCS    50$           :CHECK FOR ERROR  
4588 050714 016501 000002      MOV    TSSR(R5),R1   :GET TSSR CONTENTS  
4589 050720 005237 002212      INC    FATFLG         :BUMP COUNT  
4593 050724      ERRHRD  ERRNO,T25WDE,SFFMSG :SPACE FORWARD FAILED  
                                TRAP    C$ERHRD  
                                .WORD  532  
                                .WORD  T25WDE  
                                .WORD  SFFMSG  
4594 050734 50$:      CKLOOP                :LOOP IF SELECTED  
050734 104406      TRAP    C$CLP1  
4595 050736 012737 000001 053752      MOV    #1,T25RB      :NUMBER OF RECORDS TO SPACE OVER  
4596  
4597
```

```

4598
4599
4600
4601
4602
4603 050744 012737 140410 053750      MOV      #140410,T25PK3      :SPACE REVERSE,ACK,CVC=1 COMMAND
4604 050752 012704 053750      MOV      #T25PK3,R4        :SET UP R4 WITH PACKET ADDRESS
4605 050756      65$:
4606 050756 010465 000000      MOV      R4,TSD8(R5)       :ISSUE COMMAND
4607 050762 004737 016340      JSR      PC,WAITF          :WAIT FOR SSR TO SET
4608 050766 016501 000002      MOV      TSSR(R5),R1       :GET TSSR CONTENTS
4609 050772 012702 000200      MOV      #SSR,R2          :SET UP EXPECTED
4610 050776 020102      CMP      R1,R2            :ARE THEY EQUAL
4611 051000 001406      BEQ      75$              :BR, IF OK
4612 051002 005237 002212      INC      FATFLG           :BUMP COUNT
4616 051006      ERRHRD  ERRNO,T25WDE,PKTSSR :TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    533
                                .WORD    T25WDE
                                .WORD    PKTSSR
4617 051016      75$:  CKLOOP              :LOOP IF SELECTED
                                TRA"    C$CLP1
4618 051020      120$:
4619 051020 012703 000400      MOV      #256.,R3         :RECORD SIZE
4620 051024 013737 003114 053752      MOV      FREE,T25RB       :STARTING READ BUFFER ADDRESS
4621
4622
4623
4624
4625
4626
4627
4628 051032 012737 140001 053750      MOV      #140001,T25PK3   :READ DATA,ACK,CVC=1 COMMAND
4629 051040 012704 053750      MOV      #T25PK3,R4       :SET UP R4 WITH PACKET ADDRESS
4630 051044 010337 053756      MOV      R3,T25SZ        :SET UP RECORD SIZE IN PACKET
4631 051050 010465 000000      MOV      R4,TSD8(R5)     :ISSUE COMMAND
4632 051054 004737 016340      JSR      PC,WAITF          :WAIT FOR SSR TO SET
4633 051060 016501 000002      MOV      TSSR(R5),R1     :GET TSSR CONTENTS
4634 051064 012702 000200      MOV      #SSR,R2        :SET UP EXPECTED
4635 051070 020102      CMP      R1,R2            :ARE THEY EQUAL
4636 051072 001406      BEQ      170$            :BR, IF OK
4637 051074 005237 002212      INC      FATFLG           :BUMP COUNT
4641 051100      ERRHRD  ERRNO,RDERR,PKTSSR :TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    534
                                .WORD    RDERR
                                .WORD    PKTSSR
4642 051110      170$:  CKLOOP              :LOOP IF SELECTED
                                TRAP      C$CLP1
4643 051112 017701 131776      MOV      @FREE,R1         :GET FIRST WORD FROM BUFFER
4644 051116 012702 000000      MOV      #0,R2           :SET UP EXPECTED
4645 051122 020102      CMP      R1,R2            :WAS RECORD NUMBERED 1
4646 051124 001406      BEQ      200$            :BR, IF CORRECT RECORD
4647 051126 005237 002212      INC      FATFLG           :BUMP COUNT
4651 051132      ERRHRD  ERRNO,T25WNG,EXPREC :SHOULD HAVE BEEN RECORD NUMBER 1
                                TRAP      C$ERHRD
                                .WORD    535
051132 104456
051134 001027

```

TSV7 - HARDWARE TESTS 1-8
TEST 5: SPACE RECORDS

MACRO M1113 25-MAY-82 09.19 PAGE 118-3

1 2

SEQ 0227

	051136	054355					
	051140	015564					
4652	051142		200\$:	CKLOOP			
	051142	104406					
4653	051144			ENDSUB			
	051144						
	051144	104403					
4654	051146	023727	002212	000017	CMP	FATFLG,#15.	
4655	051154	103402			BLO	999\$	
4656	051156	004737	017272		JSR	PC,CKDROP	
4657	051162			999\$:			

.WORD T25WNG
.WORD EXPREC
TRAP C\$CLP1
:>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>>>>>>>
L10075:
TRAP C\$ESUB
:IS ERROR COUNT AT 25
:BR, IF LESS THAN 25
:TRY TO DROP THE UNIT


```
4705 ;*****
4706
4707 051320 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
4708 051324 103407 BCS 25$ ;BR, IF COMMAND ISSUED OK
4709 051326 005237 002212 INC FATFLG ;BUMP COUNT
4713 051332 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
4714 051334 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
; TRAP C$ERHRD
; .WORD 537
; .WORD WRTMSG
; .WORD SFIMSG
4715 051344 104406 25$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
4716
4717 ;*****
4718 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
4719 ;*****
4720
4721
4722
4723 051346 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
4724 051352 103407 BCS 30$ ;BR, IF NO PROBLEM
4725 051354 010001 MOV R0,R1 ;SAVE TSSR
4726 051356 005237 002212 INC FATFLG ;BUMP COUNT
4730 051362 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERHRD
; .WORD 538
; .WORD T25RWN
; .WORD PKTSSR
4731 051372 104406 30$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
4732 051374 013701 053776 MOV T25CN2,R1 ;NUMBER OF RECORDS ON TAPE
4733 051400 012702 177776 MOV #65534.,R2 ;MAX IT CAN SPACE OVER
4734 051404 020201 CMP R2,R1 ;WHICH VALUE CAN WE USE
4735 051406 003002 BGT 46$ ;BR, IF # WRITTEN > 64K
4736 051410 010103 MOV R1,R3 ;# WRITTEN CAN BE USED
4737 051412 000401 BR 47$ ;MOVE ON
4738 051414 010203 46$: MOV R2,R3 ;USE MAX NUMBER
4739 051416 162703 000001 47$: SUB #1,R3 ;DON'T GO ALL THE WAY YET
4740 051422 010337 053752 MOV R3,T25RB ;NUMBER OF RECORDS TO SPACE OVER
4741
4742 ;*****
4743 ;SPACE FORWARD,ACK,CVC=1 COMMAND
4744 ;*****
4745
4746
4747
4748 051426 012737 140010 053750 MOV #140010,T25PK3 ;SPACE FORWARD,ACK,CVC=1 COMMAND
4749 051434 012704 053750 MOV #T25PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
4750 051440
4751 051440 013737 053776 054002 65$: MOV T25CN2,T25DLY ;NUMBER OF RECORDS USED AS DELAY COUNTER
4752 051446 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
4753 051452 004737 016340 67$: JSR PC,WAITF ;WAIT FOR SSR TO SET
4754 051456 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4755 051462 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
4756 051466 020102 CMP R1,R2 ;ARE THEY EQUAL
4757 051470 001425 BEQ 75$ ;BR, IF OK
```



```
4758 051472          DELAY 250          ;DELAY .25 SECONDS
      051472 012727 000250          MOV #250,(PC)+
      051476 000000          .WORD 0
      051500 013727 002116          MOV L$DLY,(PC)+
      051504 000000          .WORD 0
      051506 005367 177772          DEC -6(PC)
      051512 001375          BNE -4
      051514 005367 177756          DEC -22(PC)
      051520 001367          BNE -20
4759 051522 005337 054002          DEC T25DLY          ;BUMP DOWN COUNTER
4760 051526 001351          BNE 67$            ;BR, IF NOT AT END OF DELAY
4761 051530 005237 002212          INC FATFLG         ;BUMP COUNT
4765 051534          ERRHRD ERRNO,T25WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      051534 104456          TRAP C$ERHRD
      051536 001033          .WORD 539
      051540 054065          .WORD T25WDE
      051542 012136          .WORD PKTSSR
4766 051544          75$: CKLOOP          ;LOOP IF SELECTED
      051544 104406          TRAP C$CLP1
4767 051546 012703 010000          MOV #4096.,R3      ;RECORD SIZE
4768 051552 013737 003114 053752    MOV FREE,T25RB     ;STARTING READ BUFFER ADDRESS
4769
4770          ;*****
4771          ;
4772          ;READ DATA,ACK COMMAND
4773          ;
4774          ;*****
4775
4776 051560 012737 100001 053750    165$: MOV #100001,T25PK3 ;READ DATA,ACK COMMAND
4777 051566 012704 053750          MOV #125PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
4778 051572 010337 053756          MOV R3,T25SZ      ;SET UP RECORD SIZE IN PACKET
4779 051576 010465 000000          MOV R4,TSDB(R5)   ;ISSUE COMMAND
4780 051602 004737 016340          JSR PC,WAITF      ;WAIT FOR SSR TO SET
4781 051606 016501 000002          MOV TSSR(R5),R1   ;GET TSSR CONTENTS
4782 051612 012702 000200          MOV #SSR,R2       ;SET UP EXPECTED
4783 051616 020102          CMP R1,R2         ;ARE THEY EQUAL
4784 051620 001411          BEQ 170$          ;BR, IF OK
4785 051622 032701 000004          BIT #BIT2,R1      ;CHECK FOR TAPE STATUS ALERT
4786 051626 001006          BNE 170$          ;IF SET ALL IS WELL
4787 051630 005237 002212          INC FATFLG        ;BUMP COUNT
4791 051634          ERRHRD ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      051634 104456          TRAP C$ERHRD
      051636 001034          .WORD 540
      051640 005204          .WORD RDERR
      051642 012136          .WORD PKTSSR
4792 051644          170$: CKLOOP          ;LOOP IF SELECTED
      051644 104406          TRAP C$CLP1
4793 051646 017701 131242          MOV @FREE,R1      ;GET FIRST WORD FROM BUFFER
4794 051652 013702 053776          MOV T25CN2,R2    ;SET UP EXPECTED
4795 051656 162702 000001          SUB #1,R2         ;SHOULD BE LAST RECORD
4796 051662 020102          CMP R1,R2         ;WAS RECORD NUMBERED R3
4797 051664 001406          BEQ 200$          ;BR, IF CORRECT RECORD
4798 051666 005237 002212          INC FATFLG        ;BUMP COUNT
4802 051672          ERRHRD ERRNO,T25WNG,EXPREC ;SHOULD HAVE BEEN RECORD NUMBER 1
      051672 104456          TRAP C$ERHRD
      051674 001035          .WORD 541
      051676 054355          .WORD T25WNG
```

TSV7 - HARDWARE TESTS 1-8
 TEST 5: SPACE RECORDS

MACRO M1113 25-MAY-82 09:19 PAGE 119-3 ^M 2

SEQ 0231

4803	051700	015564					
	051702		200\$:	CKLOOP			
4804	051702	104406					
	051704			ENDSUB			
4805	051704	104403					
	051706	023727	002212	000017	CMP	FATFLG,#15.	
4806	051714	103402			BLO	999\$	
4807	051716	004737	017272		JSR	PC,CKDROP	
4808	051722				999\$:		

```

        .WORD  EXPREC
        TRAP  C$CLP1
;>>>>>>>>>> END SUBTEST >>>>>>>>>>
        L10076:
        TRAP  C$ESUB
        ;IS ERROR COUNT AT 25
        ;BR, IF LESS THAN 25
        ;TRY TO DROP THE UNIT

```



```

4857 052052 004737 010752      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
4858 052056 103407      BCS      25$           ;BR, IF COMMAND ISSUED OK
4859 052060 005237 002212      INC      FATFLG       ;BUMP COUNT
4863 052064 010001      MOV      R0,R1        ;SAVE CONTENTS OF TSSR
4864 052066      ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP      C$ERHRD
                                .WORD     543
                                .WORD     WRTMSG
                                .WORD     SFIMSG
4865 052076      25$:    CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
4866 052076 104406
4867
4868
4869
4870
4871
4872
4873 052100 004737 011104      JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
4874 052104 103407      BCS      30$           ;BR, IF NO PROBLEM
4875 052106 010001      MOV      R0,R1        ;SAVE TSSR
4876 052110 005237 002212      INC      FATFLG       ;BUMP COUNT
4880 052114      ERRHRD  ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     544
                                .WORD     T25RWN
                                .WORD     PKTSSR
4881 052124      30$:    CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
4882 052124 104406
4883
4884
4885
4886
4887
4888
4889 052126 013701 053660      MOV      T25BFR+6,R1  ;PICK UP XSTO
4890 052132 010102      MOV      R1,R2        ;SET UP EXPECTED
4891 052134 052702 000002      BIS      #BIT1,R2     ;SET BOT BIT IN EXPECTED
4892 052140 020102      CMP      R1,R2        ;DOES EXP = REC'D
4893 052142 001406      BEQ      40$          ;BR, IF EQUAL (OK)
4894 052144 005237 002212      INC      FATFLG       ;BUMP COUNT
4898 052150      ERRHRD  ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD     545
                                .WORD     T25BOT
                                .WORD     EXPREC
4899 052160      40$:    CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
4900 052160 104406
4901 052162 013701 053776      MOV      T25CN2,R1    ;NUMBER OF RECORDS ON TAPE
4902 052166 012702 177776      MOV      #65534.,R2   ;MAX IT CAN SPACE OVER
4903 052172 020201      CMP      R2,R1        ;WHICH VALUE CAN WE USE
4904 052174 003002      BGT      46$          ;BR, IF # WRITTEN > 64K
4905 052176 010103      MOV      R1,R3        ;# WRITTEN CAN BE USED
4906 052200 000401      BR       47$          ;MOVE ON
4907 052204 010203      46$:    MOV      R2,R3 ;USE MAX NUMBER
                                47$:

```

```
4908 052204 010337 053752          MOV      R3,T25RB          ;NUMBER OF RECORDS TO SPACE OVER
4909
4910          :*****
4911          :SPACE FORWARD,ACK,CVC=1 COMMAND
4912          :*****
4913
4914
4915
4916 052210 012737 140010 053750          MOV      #140010,T25PK3          ;SPACE FORWARD,ACK,CVC=1 COMMAND
4917 052216 012704 053750          MOV      #T25PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
4918 052222 010465 000000          MOV      R4,TSDB(R5)          ;ISSUE COMMAND
4919 052226 013737 053776 054002          MOV      T25CN2,T25DLY          ;SET UP DELAY COUNTER
4920 052234 004737 016340          48$: JSR      PC,WAITF          ;WAIT FOR SSR TO SET
4921 052240 016501 000002          MOV      TSSR(R5),F1          ;GET TSSR CONTENTS
4922 052244 012702 000200          MOV      #SSR,R2          ;SET UP EXPECTED
4923 052250 020102          CMP      R1,R2          ;ARE THEY EQUAL
4924 052252 001425          BEQ      50$          ;BR, IF OK
4925 052254          DELAY    250          ;WAIT .25 SECONDS
          MOV      #250,(PC)+
          .WORD    0
          MOV      L$DLY,(PC)+
          .WORD    0
          DEC      -6(PC)
          BNE      -4
          DEC      -22(PC)
          BNE      -20
4926 052304 005337 054002          DEC      T25DLY          ;DEC THE DELAY COUNTER
4927 052310 001351          BNE      48$          ;BR, IF COUNTER HASN'T EXPIRED
4928 052312 005237 002212          INC      FATFLG          ;BUMP COUNT
4932 052316          ERRHRD  ERRNO,T25WDE,EXPREC          ;TSSR INCORRECT AFTER READ DATA
          TRAP    C$ERHRD
          .WORD    546
          .WORD    T25WDE
          .WORD    EXPREC
4933 052326          50$: CKLOOP          TRAP    C$CLP1
          MOV      T25CN2,R1          ;NUMBER OF RECORDS ON TAPE
4934 052330 013701 053776          MOV      #65534.,R2          ;MAX IT CAN SPACE OVER
4935 052334 012702 177776          CMP      R2,R1          ;WHICH VALUE CAN WE USE
4936 052340 020201          BGT      55$          ;BR, IF # WRITTEN > 64K
4937 052342 003002          MOV      R1,R3          ;# WRITTEN CAN BE USED
4938 052344 010103          BR       60$          ;MOVE ON
4939 052346 000401          55$: MOV      R2,R3          ;USE MAX NUMBER
4940 052350 010203          60$: SUB      #1,R3          ;DON'T GO ALL THE WAY YET
4941 052352 162703 000001          MOV      R3,T25RB          ;NUMBER OF RECORDS TO SPACE OVER
4942 052356 010337 053752
4943
4944          :*****
4945          :SPACE REVERSE,ACK,CVC=1 COMMAND
4946          :*****
4947
4948
4949
4950 052362 012737 140410 053750          MOV      #140410,T25PK3          ;SPACE REVERSE,ACK,CVC=1 COMMAND
4951 052370 012704 053750          MOV      #T25PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
4952 052374 010465 000000          MOV      R4,TSDB(R5)          ;ISSUE COMMAND
4953 052400 013737 053776 054002          MOV      T25CN2,T25DLY          ;SET UP COUNTER
4954 052406 004737 016340          70$: JSR      PC,WAITF          ;WAIT FOR SSR TO SET
```

```
4955 052412 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
4956 052416 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
4957 052422 020102                CMP      R1,R2          ;ARE THEY EQUAL
4958 052424 001425                BEQ      75$            ;BR, IF OK
4959 052426                DELAY    250            ;WAIT ABOUT .25 SECONDS
                                MOV      #250,(PC)+
                                .WORD    0
                                MOV      L$DLY,(PC)+
                                .WORD    0
                                DEC      -6(PC)
                                BNE      -.4
                                DEC      -22(PC)
                                BNE      -.20
                                052426 012727 000250
                                052432 000000
                                052434 013727 002116
                                052440 000000
                                052442 005367 177772
                                052446 001375
                                052450 005367 177756
                                052454 001367
4960 052456 005337 054002      DEC      T25DLY         ;BUMP COUNTER DOWN
4961 052462 001351                BNE      70$            ;BR, IF COUNTER HASN'T EXPIRED
4962 052464 005237 002212      INC      FATFLG         ;BUMP COUNT
4966 052470                ERRHRD  ERRNO,T25WDE,EXPREC ;TSSR INCORRECT AFTER READ DATA
                                TRAP    C$ERHRD
                                .WORD    547
                                .WORD    T25WDE
                                .WORD    EXPREC
                                052470 104456
                                052472 001043
                                052474 054065
                                052476 015564
4967 052500                75$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                052500 104406
4968 052502 012703 010000      MOV      #4096,R3       ;RECORD SIZE
4969 052506 013737 003114 053752  MOV      FREE,T25RB     ;STARTING READ BUFFER ADDRESS
4970
4971
4972
4973
4974
4975
4976
4977 052514 012737 100001 053750      MOV      #100001,T25PK3 ;READ DATA,ACK COMMAND
4978 052522 012704 053750 165$:  MOV      #T25PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
4979 052526 012700 177777      MOV      #177777,R0     ;SET ALL ONES INTO CORRECT REGISTER
4980 052532 004737 017512      JSR      PC,FILLMEM     ;FILL MEMORY WITH RECORD SIZE
4981 052536 010337 053756      MOV      R3,T25SZ      ;SET UP RECORD SIZE IN PACKET
4982 052542 010465 000000      MOV      R4,T5DB(R5)   ;ISSUE COMMAND
4983 052546 004737 016340      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
4984 052552 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
4985 052556 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
4986 052562 020102                CMP      R1,R2          ;ARE THEY EQUAL
4987 052564 001411                BEQ      170$          ;BR, IF OK
4988 052566 032701 000004      BIT      #BIT2,R1      ;CHECK FOR TAPE STATUS ALERT
4989 052572 001006                BNE      170$          ;BR, IF BIT SET
4990 052574 005237 002212      INC      FATFLG         ;BUMP COUNT
4994 052600                ERRHRD  ERRNO,RDERR,EXPREC ;TSSR INCORRECT AFTER READ DATA
                                TRAP    C$ERHRD
                                .WORD    548
                                .WORD    RDERR
                                .WORD    EXPREC
                                052600 104456
                                052602 001044
                                052604 005204
                                052606 015564
4995 052610                170$: CKLOOP           ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                052610 104406
4996 052612 017701 130276      MOV      @FREE,R1      ;GET FIRST WORD FROM BUFFER
4997 052616 012702 000001      MOV      #1,R2         ;SET UP EXPECTED
4998 052622 020102                CMP      R1,R2          ;WAS RECORD NUMBERED R3
4999 052624 001406                BEQ      200$          ;BR, IF CORRECT RECORD
```

```
5000 052626 005237 002212      INC      FATFLG      ;BUMP COUNT
5004 052632      ERRHRD  ERRNO,T25WNH,EXPREC ;SHOULD HAVE BEEN RECORD NUMBER 1
      052632 104456                               TRAP      C$ERHRD
      052634 001045                               .WORD    549
      052636 054530                               .WORD    T25WNH
      052640 015564                               .WORD    EXPREC
5005 052642      200$:  CKLOOP
      052642 104406                               TRAP      C$CLP1
5006 052644      ENDSUB                          ;>>>>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>>>
      052644      L10077:
      052644 104403                               TRAP      C$ESUB
5007 052646 023727 002212 000017  CMP      FATFLG,#15. ;IS ERROR COUNT AT 25
5008 052654 103402      BLO      999$      ;BR, IF LESS THAN 25
5009 052656 004737 017272      JSR      PC,CKDROP ;TRY TO DROP THE UNIT
5010 052662      999$:
```

```

5012
5013
5014
5015
5016
5017
5018
5019
5020
5021
5022
5023 052662
      052662
      052662 104402
5024 052664 004737 055166      JSR     PC,T25REST      ;SET COMMAND PACKET      TRAP     C$BSUB
5025 052670 004737 055260      JSR     PC,T25RT2       ;SET UP OTHER COMMAND PACKET
5026 052674 004737 055322      JSR     PC,T25RT3       ;SET UP OTHER COMMAND PACKET
5027
5028
5029
5030
5031
5032
5033
5034 052700 004737 016064      JSR     PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
5035 052704 103407          BCS     20$             ;BR IF INIT WAS OK
5036 052706 005237 002212      INC     FATFLG          ;BUMP COUNT
5040 052712 010001          MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
5041 052714          ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      052714 104455          TRAP     C$ERDF
      052716 C01046          .WORD   550
      052720 003650          .WORD   SFIERR
      052722 012124          .WORD   SFIMSG
5042 052724 013737 002172 053650 20$:  MOV     UNITN,T25DSW     ;SET UP UNIT NUMBER
5043
5044 052732 012704 053630      MOV     #T25PACKET,R4  ;SUBROUTINE NEEDS PACKET ADDRESS
5045
5046
5047
5048
5049
5050
5051
5052 052736 004737 010752      JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
5053 052742 103407          BCS     25$             ;BR, IF COMMAND ISSUED OK
5054 052744 005237 002212      INC     FATFLG          ;BUMP COUNT
5058 052750 010001          MOV     R0,R1           ;SAVE CONTENTS OF TSSR
5059 052752          ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      052752 104456          TRAP     C$ERHRD
      052754 001047          .WORD   551
      052756 005054          .WORD   WRTMSG
      052760 012124          .WORD   SFIMSG
5060 052762          25$:  CKLOOP          ;LOOP IF SELECTED
      052762 104406          TRAP     C$CLP1
5061
5062
5063
    
```



```
5064                                     ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5065                                     ;*****
5066                                     ;*****
5067                                     ;*****
5068 052764 004737 011104                JSR    PC,REWIND                ;CALL TAPE REWIND COMMAND
5069 052770 103407                        BCS    30$                      ;BR, IF NO PROBLEM
5070 052772 010001                        MOV    R0,R1                    ;SAVE TSSR
5071 052774 005237 002212                INC    FATFLG                   ;BUMP COUNT
5075 053000                                ERRHRD ERRNO,T25RWN,PKTSSR      ;REWIND NOT ACCEPTED
                                104456                                TRAP  C$ERHRD
                                001050                                .WORD 552
                                053004 054755                                .WORD T25RWN
                                053006 012136                                .WORD PKTSSR
5076 053010 104406                30$:  CKLOOP                    ;LOOP IF SELECTED
                                TRAP  C$CLP1
5077                                     ;*****
5078                                     ;*****
5079                                     ;*****
5080                                     ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5081                                     ;*****
5082                                     ;*****
5083                                     ;*****
5084 053012 013701 053660                MOV    T25BFR+6,R1              ;PICK UP XSTO
5085 053016 010102                        MOV    R1,R2                    ;SET UP EXPECTED
5086 053020 052702 000002                BIS    #BIT1,R2                 ;SET BOT BIT IN EXPECTED
5087 053024 020102                        CMP    R1,R2                    ;DOES EXP = REC'D
5088 053026 001406                        BEQ    40$                      ;BR, IF EQUAL (OK)
5089 053030 005237 002212                INC    FATFLG                   ;BUMP COUNT
5093 053034                                ERRHRD ERRNO,T25BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
                                104456                                TRAP  C$ERHRD
                                053036 001051                                .WORD 553
                                053040 054145                                .WORD T25BOT
                                053042 015564                                .WORD EXPREC
5094 053044 104406                40$:  CKLOOP                    ;LOOP IF SELECTED
                                TRAP  C$CLP1
5095 053046 012737 000001 053752        MOV    #1,T25RB                 ;NUMBER OF RECORDS TO SPACE OVER
5096                                     ;*****
5097                                     ;*****
5098                                     ;*****
5099                                     ;SPACE REVERSE,ACK COMMAND
5100                                     ;*****
5101                                     ;*****
5102                                     ;*****
5103 053054 012737 100410 053750        MOV    #100410,T25PK3           ;SPACE REVERSE,ACK COMMAND
5104 053062 012704 053750                MOV    #T25PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
5105 053066                                65$:
5106 053066 010465 000000                MOV    R4,T5DB(R5)             ;ISSUE COMMAND
5107 053072 004737 016340                JSR    PC,WAITF                 ;WAIT FOR SSR TO SET
5108 053076 016501 000002                MOV    TSSR(R5),R1             ;GET TSSR CONTENTS
5109 053102 012702 100206                MOV    #SSR!SC!BIT1!BIT2,R2   ;SET UP EXPECTED
5110 053106 020102                        CMP    R1,R2                    ;ARE THEY EQUAL
5111 053110 001406                        BEQ    75$                      ;BR, IF OK
5112 053112 005237 002212                INC    FATFLG                   ;BUMP COUNT
5116 053116                                ERRHRD ERRNO,T25WDE,PKTSSR      ;TSSR INCORRECT AFTER READ DATA
                                104456                                TRAP  C$ERHRD
                                053120 001052                                .WORD 554
                                053122 054065                                .WORD T25WDE
```

5117	053124	012136							
	053126		75\$:	CKLOOP		:LOOP IF SELECTED	.WORD	PKTSSR	
	053126	104406					TRAP	C\$CLP1	
5118									
5119									
5120									
5121									
5122									
5123									
5124									
5125	053130	013701	053660		MOV	T25BFR+6,R1		:GET XST0 STATUS WORD	
5126	053134	010102			MOV	R1,R2		:SET UP EXPECTED	
5127	053136	052702	002000		BIS	#BIT10,R2		:SET THE NEF BIT	
5128	053142	020102			CMP	R1,R2		:ARE THEY EQUAL	
5129	053144	001406			BEQ	170\$:BR, IF EQUAL (GOOD)	
5130	053146	005237	002212		INC	FATFLG		:BUMP COUNT	
5134	053152				ERRHRD	ERRNO,T25NEF,EXPREC		:NEF SHOULD BE SET	
	053152	104456						TRAP	C\$ERHRD
	053154	001053						.WORD	555
	053156	054613						.WORD	T25NEF
	053160	015564						.WORD	EXPREC
5135	053162			170\$:	CKLOOP				
	053162	104406						TRAP	C\$CLP1
5136	053164				ENDSUB				
	053164	104403							
	053164								
5137	053166	023727	002212	000017	CMP	FATFLG,#15.		L10100: TRAP	C\$ESUB
5138	053174	103402			BLO	999\$:IS ERROR COUNT AT 25	
5139	053176	004737	017272		JSR	PC,CKDROP		:BR, IF LESS THAN 25	
5140	053202				999\$:			:TRY TO DROP THE UNIT	


```
5194
5195
5196
5197
5198
5199
5200 053304 004737 011104      JSR      PC,REWIND      :CALL TAPE REWIND COMMAND
5201 053310 103407              BCS      30$           :BR, IF NO PROBLEM
5202 053312 010001              MOV      R0,R1         :SAVE TSSR
5203 053314 005237 002212      INC      FATFLG        :BUMP COUNT
5207 053320              ERRHRD   ERRNO,T25RWN,PKTSSR :REWIND NOT ACCEPTED
5207 053320 104456              TRAP     C$ERHRD
5207 053322 001056              .WORD   558
5207 053324 054755              .WORD   T25RWN
5207 053326 012136              .WORD   PKTSSR
5208 053330 30$:      CKLOOP              :LOOP IF SELECTED
5208 053330 104406              TRAP     C$CLP1
5209
5210
5211
5212
5213
5214
5215
5216 053332 013701 053660      MOV      T25BFR+6,R1   :PICK UP XSTO
5217 053336 010102              MOV      R1,R2         :SET UP EXPECTED
5218 053340 052702 000002      BIS      #BIT1,R2      :SET BOT BIT IN EXPECTED
5219 053344 020102              CMP      R1,R2         :DOES EXP = REC'D
5220 053346 001406              BEQ     40$           :BR, IF EQUAL (OK)
5221 053350 005237 002212      INC      FATFLG        :BUMP COUNT
5225 053354              ERRHRD   ERRNO,T25BOT,EXPREC :TAPE NOT AT BOT AFTER REWIND
5225 053354 104456              TRAP     C$ERHRD
5225 053356 001057              .WORD   559
5225 053360 054145              .WORD   T25BOT
5225 053362 015564              .WORD   EXPREC
5226 053364 40$:      CKLOOP              TRAP     C$CLP1
5227 053366 012737 000001 053752      MOV      #1,T25RB      :NUMBER OF RECORDS TO SPACE OVER
5228
5229
5230
5231
5232
5233
5234
5235 053374 012737 140210 053750      MOV      #140210,T25PK3 :SPACE FORWARD,IE,ACK,CVC=1 COMMAND
5236 053402 012704 053750              MOV      #T25PK3,R4    :SET UP R4 WITH PACKET ADDRESS
5237 053406 010465 000000              MOV      R4,T25DB(R5)  :ISSUE COMMAND
5238 053412 004737 016340              JSR      PC,WAITF      :WAIT FOR SSR TO SET
5239 053416 016501 000002              MOV      TSSR(R5),R1   :GET TSSR CONTENTS
5240 053422 012702 000200              MOV      #SSR,R2       :SET UP EXPECTED
5241 053426 020102              CMP      R1,R2         :ARE THEY EQUAL
5242 053430 001406              BEQ     75$           :BR, IF OK
5243 053432 005237 002212      INC      FATFLG        :BUMP COUNT
5247 053436              ERRHRD   ERRNO,T25WDE,EXPREC :TSSR INCORRECT AFTER READ DATA
5247 053436 104456              TRAP     C$ERHRD
5247 053440 001060              .WORD   560
```


TSV7 - HARDWARE TESTS 1-8
TEST 5: SPACE RECORDS

MACRO M1113 25-MAY-82 09:19 PAGE ^{L 3} 123-3

053620 001532

SEQ 0243

.WORD L10071-.

5296			:+		
5297			:LOCAL STORAGE FOR THIS TEST		
5298			:-		
5300		053630			
5302	053630		T25PACKET: .=<.+10>B177770		
5303	053630	100004	.WORD 100004		:COMMAND PACKET FOR TEST
5304	053632	053640	.WORD T25DATA		:WRITE CHARACTERISTICS COMMAND, WITH ACK
5305	053634	000000	.WORD 0		:ADDRESS OF CHARACTERISTICS BLOCK
5306	053636	000010	.WORD 8.		:STARTING VALUE OF BLOCK SIZE
5307	053640		T25DATA:		:CHARACTERISTICS DATA BLOCK
5308	053640	053652	.WORD T25BFR		:ADDRESS OF MESSAGE BUFFER
5309	053642	000000	.WORD 0		
5310	053644	000012	.WORD 10.		:LENGTH OF MESSAGE BUFFER
5311	053646	000000	.WORD 0		
5312	053650	000000	T25DSW: .WORD 0		:SELECT DRIVE ZERO
5313	053652		T25BFR: .BLKW 25.		:MESSAGE BUFFER
5314					
5315			:WRITE SUBSYSTEM MEMORY COMMAND PACKET		
5316			:		
5318		053740			
5320	053740		T25PK2: .-<.+10>B177770		
5321	053740	100006	.WORD 100006		:WRITE SUB SYS MEM COMMAND, AND ACK
5322	053742	053760	.WORD T25BF2		:ADDRESS OF SELECT BLOCK DATA
5323	053744	000000	.WORD 0		
5324	053746	000006	.WORD 6.		:SIZE OF DATA PACKET
5325					
5329	053750		T25PK3:		
5330	053750	100005	.WORD 100005		:READ COMMAND, AND ACK
5331	053752		T25RB:		
5332	053752	003114	T25WB: .WORD FREE		:ADDRESS OF WRITE BUFFER
5333	053754	000000	.WORD 0		
5334	053756	000000	T25SZ: .WORD 0		:SIZE OF BUFFER (EXTENT)
5335			.EVEN		
5336			:		
5337			:		
5338			:		
5339	053760		T25BF2:		
5340	053760	010	T25BS0: .BYTE 10		:BSEL0 AREA
5341	053761	200	T25BS1: .BYTE 200		:BSEL1 AREA
5342	053762	000000	T25S2: .WORD 0		:SEL 2 AREA
5343	053764	000000	T25S3: .WORD 0		:DATA AREA
5344			:		
5345			:		
5346			.EVEN		
5347			:TAPE MOTION PACKET COMMAND VALUES		
5348					
5349	053766	100005	T25RN: .WORD 100005		:READ DATA (NEXT)
5350	053770	100405	T25WDR: .WORD 100405		:READ DATA RETRY
5351	053772	102005	T25CON: .WORD 102005		:WRITE CONTINUOUS
5352	053774	177777	.WORD 177777		:END OF DATA
5353					
5354					

```

5356 053776 000000      T25CN2: .WORD 0          ;COUNTER FOR RECORDS
5357 054000 000000      T25CNT: .WORD 0          ;COUNTER FOR RECORDS
5358 054002 000000      T25DLY: .WORD 0          ;COUNTER FOR RECORDS
5359
5360
5361      ;+
5362      ;LOCAL TEXT MESSAGES FOR TEST
5363      ;-
5364 054004      127      122      111  T25SSR: .ASCIZ 'WRITE SUBSYSTEM Miscellaneous Read Status Failed'
5365 054065      124      123      123  T25WDE: .ASCIZ 'TSSR Not Correct After POSITION (SPACE) Command'
5366 054145      124      141      160  T25BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
5367 054212      124      123      123  T25TM: .ASCIZ 'TSSR Not Correct After POSITION (Space Command) Reject'
5368 054301      127      162      151  T25NET: .ASCIZ 'Write Tape, Status Alert, But No EOT Sensed'
5369 054355      123      160      141  T25WNG: .ASCIZ 'Space Forward Failed To Position On Correct Record'
5370 054440      123      160      141  T25BNC: .ASCIZ 'Space Forward, From BOT, Failed To Clear BOT Indication'
5371 054530      123      160      141  T25WNH: .ASCIZ 'Space Reverse Failed To Position On Correct Record'
5372 054613      123      160      141  T25NEF: .ASCIZ 'Space Reverse, At BOT, Failed To Set NEF (XST0)'
5373 054673      123      160      141  T25RIB: .ASCIZ 'Space Reverse, Into BOT, Failed To Set RIB (XST3)'
5374 054755      122      145      167  T25RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
5375 055024      104      162      151  T25OFL: .ASCIZ 'Drive 7 Select Failed To Set 'OFL' In TSSR'
5376 055077      124      123      123  T25WDC: .ASCIZ 'TSSR Not Correct After READ DATA Command'
5377 055150      123      160      141  T25ID: .ASCIZ 'Space Records'
5378
5379
5380
5381      ;+
5382      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5383      ;WRITE SUBSYSTEM MEMORY COMMAND
5384      ;-
5385
5386 055166      T25REST:
5387 055166      SAVREG          ;SAVE THE REGISTERS
5388 055172      012701 053630      MOV #T25PACKET,R1      ;START OF THE PACKET
5389 055176      012721 100004      MOV #100004,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK
5390 055202      012721 053640      MOV #T25DATA,(R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
5391 055206      005021      CLR (R1)+              ;EXTENDED ADDRESS
5392 055210      012721 000012      MOV #10,(R1)+          ;SIZE OF DATA BLOCK IN BYTES
5393 055214      012721 053652      MOV #T25BFR,(R1)+      ;ADDRESS OF MESSAGE BUFFER
5394 055220      005021      CLR (R1)+              ;
5395 055222      012721 000024      MOV #20,(R1)+          ;LENGTH OF MESSAGE BUFFER
5396 055226      005021      CLR (R1)+              ;
5397 055230      012711 000000      MOV #0,(R1)            ;SELECT DRIVE ZERO
5398 055234      012702 000030      MOV #24,R2             ;NUMBER OF LOCATIONS TO BE CLEARED
5399 055240      012762 177777 053652 64$: MOV #177777,T25BFR(R2) ;ALL ONES TO MESSAGE BUFFER
5400 055246      005742      TST -(R2)              ;NEXT LOCATION
5401 055250      022702 000000      CMP #0,R2              ;IS R2 AT ZERO YET
5402 055254      001371      BNE 64$                ;KEEP GOING UNTIL DONE
5403 055256      000207      RTS PC                 ;RETURN
5404
5405 055260      T25RT2:
5406 055260      SAVREG          ;SAVE THE REGISTERS
5407 055264      012701 053740      MOV #T25PK2,R1         ;START OF THE PACKET
5408 055270      012721 100006      MOV #100006,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK.
5409 055274      012721 053760      MOV #T25BF2,(R1)+      ;ADDRESS OF DATA BLOCK
5410 055300      005021      CLR (R1)+              ;EXTENDED ADDRESS
5411 055302      012721 000006      MOV #6,(R1)+           ;SIZE OF DATA BLOCK IN BYTES
5412 055306      005021      CLR (R1)+              ;

```


5413	055310	012701	053760
5414	055314	005021	
5415	055316	005011	
5416	055320	000207	
5417	055322		
5418	055322		
5419	055326	012701	053750
5420	055332	012721	000000
5421	055336	012721	000000
5422	055342	005021	
5423	055344	012721	000000
5424	055350	000207	
5425	055352		
	055352		
	055352	104401	

T25RT3:

```
MOV #T25BF2,R1
CLR (R1)+
CLR (R1)
RTS PC
SAVREG
MOV #T25PK3,R1
MOV #0,(R1)+
MOV #0,(R1)+
CLR (R1)+
MOV #0,(R1)+
RTS PC
ENDTST
```

:POINT TO DATA SEL AREA

:RETURN

:SAVE THE REGISTERS
:START OF THE PACKET
:WRITE SUBSYSTEM MEM. WITH ACK,
:ADDRESS OF DATA BLOCK
:EXTENDED ADDRESS
:SIZE OF DATA BLOCK IN BYTES
:RETURN

L10071: TRAP C\$ETST


```
5485  
5486  
5487  
5488  
5489  
5490 055440 004737 016064  
5491 055444 103426  
5492 055446  
055446 012727 000250  
055452 000000  
055454 013727 002116  
055460 000000  
055462 005367 177772  
055466 001375  
055470 005367 177756  
055474 001367  
5493 055476 005337 071754  
5494 055502 001356  
5495 055504 005237 002212  
5499 055510 010001  
5500 055512  
055512 104455  
055514 001131  
055516 003650  
055520 012124  
5501 055522  
5502 055522 013737 002172 071620  
5503 055530 012704 071600  
5504  
5505  
5506  
5507  
5508  
5509  
5510  
5511 055534 004737 010752  
5512 055540 103407  
5513 055542 005237 002212  
5517 055546 010001  
5518 055550  
055550 104456  
055552 001132  
055554 005054  
055556 012124  
5519 055560  
055560 104406  
5520  
5521  
5522  
5523  
5524  
5525  
5526  
5527 055562 004737 011104  
5528 055566 103413  
5529 055570 016501 000002  
5530 055574 012702 000200
```

```
;  
:ISSUE CONTROLLER 'SOFT' INITIALIZE - CARRY BIT CLEAR IF ERROR  
:*****  
10$: JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER  
BCS 20$ ;BR IF INIT WAS OK  
DELAY 250 ;DELAY FOR A REWIND TO FINISH  
MOV #250,(PC)+  
.WORD 0  
MOV LSDLY,(PC)+  
.WORD 0  
DEC -6(PC)  
BNE -4  
DEC -22(PC)  
BNE -20  
DEC T26DLY ;DEC COUNTER  
BNE 10$ ;BR, IF DELAY NOT READY  
INC FATFLG ;BUMP COUNT  
MOV R0,R1 ;CONTENTS OF TSSR REGISTER  
ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK  
TRAP C$ERDF  
.WORD 601  
.WORD SFIERR  
.WORD SFIMSG  
20$: MOV UNITN,T26DSW ;SET UP UNIT NUMBER  
MOV #T26PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS  
:*****  
:WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)  
:*****  
JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS  
BCS 26$ ;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$ERHRD  
.WORD 602  
.WORD WRTMSG  
.WORD SFIMSG  
26$: 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  
MOV #SSR,R2 ;SET UP EXPECTED TSSR
```

```

5531 055600 010004          MOV    R0,R4          ;PACKET ADDRESS SET UP
5532 055602 005237 002212  INC    FATFLG        ;BUMP COUNT
5536 055606          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
          055606 104456          TRAP  C$ERHRD
          055610 001133          .WORD 603
          055612 073254          .WORD T26RWN
          055614 012136          .WORD PKTSSR
5537 055616          30$:  CKLOOP          ;LOOP IF SELECTED
          055616 104406          TRAP  C$CLP1
5538
5539
5540
5541
5542
5543
5544
          :*****
          :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
          :*****
5545 055620 013701 071630          MOV    T26BFR+6,R1   ;PICK UP XSTO
5546 055624 010102          MOV    R1,R2         ;SET UP EXPECTED
5547 055626 052702 000002          BIS    #BIT1,R2     ;SET BOT BIT IN EXPECTED
5548 055632 020102          CMP    R1,R2         ;DOES EXP = REC'D
5549 055634 001406          BEQ    40$          ;BR, IF EQUAL (OK)
5550 055636 005237 002212  INC    FATFLG        ;BUMP COUNT
5554 055642          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          055642 104456          TRAP  C$ERHRD
          055644 001134          .WORD 604
          055646 072765          .WORD T26BOT
          055650 015564          .WORD EXPREC
5555 055652          40$:  CKLOOP          ;LOOP IF SELECTED
          055652 104406          TRAP  C$CLP1
5556 055654 012703 000400          MOV    #256.,R3     ;RECORD SIZE
5557 055660 013737 003114 071722  MOV    FREE,T26RB   ;STARTING WRITE BUFFER ADDRESS
5558
5559
5560
5561
5562
5563
5564
          :*****
          :WRITE DATA,ACK,CVC=1 COMMAND
          :*****
5565 055666 012737 140005 071720  MOV    #140005,T26PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
5566 055674 012704 071720  MOV    #T26PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
5567 055700          65$:
5568 055700 010300          MOV    R3,R0        ;SET PATTERN IN CORRECT REGISTER
5569 055702 004737 017512  .ISR   PC,FILLMEM   ;FILL MEMORY WITH RECORD SIZE
5570 055706 010337 071726  MOV    R3,T26SZ     ;SET UP RECORD SIZE IN PACKET
5571 055712 010465 000000  MOV    R4,TSDB(R5)  ;ISSUE COMMAND
5572 055716 004737 016340  JSR    PC,WAITF     ;WAIT FOR SSR TO SET
5573 055722 016501 000002  MOV    TSSR(R5),R1  ;GET TSSR CONTENTS
5574 055726 012702 000200  MOV    #SSR,R2     ;SET UP EXPECTED
5575 055732 020102          CMP    R1,R2        ;ARE THEY EQUAL
5576 055734 001406          BEQ    75$          ;BR, IF OK
5577 055736 005237 002212  INC    FATFLG        ;BUMP COUNT
5581 055742          ERRHRD  ERRNO,WRTErr,EXPREC ;TSSR INCORRECT AFTER WRITE DATA
          055742 104456          TRAP  C$ERHRD
          055744 001135          .WORD 605
          055746 005111          .WORD WRTErr
          055750 015564          .WORD EXPREC
5582 055752          75$:  CKLOOP          ;LOOP IF SELECTED

```



```

5634 056072 004737 010556      JSR      PC,SPACE      ;CALL SPACE ROUTINE
5635 056076 103412              BCS      150$          ;BR, IF NO PROBLEM WITH SPACE COMMAND
5636 056100 016501 000002      MOV      TSSR(R5),R1  ;GET TSSR
5637 056104 012702 000200      MOV      #SSR,R2     ;SET UP EXPECTED TSSR
5638 056110 005237 002212      INC      FATFLG      ;BUMP COUNT
5642 056114              ERRHRD   ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
      056114 104456              TRAP     C$ERHRD
      056116 001140              .WORD   608
      056120 072367              .WORD   T26SC
      056122 015564              .WORD   EXPREC
5643 056124              150$:   CKLOOP
      056124 104406              TRAP     C$CLP1
5644 056126 013703 071752      MOV      T26RSZ,R3   ;RECORD SIZE
5645 056132 013737 003114 071722  MOV      FREE,T26RB  ;STARTING READ BUFFER ADDRESS
5646
5647
5648
5649
5650
5651
5652
5653 056140 012737 141001 071720  165$:   MOV      #141001,T26PK3 ;REREREAD DATA,CVC=1,ACK COMMAND
5654 056146 012704 071720      MOV      #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
5655 056152 010337 071726      MOV      R3,T26SZ   ;SET UP RECORD SIZE IN PACKET
5656 056156 010465 000000      MOV      R4,TSDB(R5);ISSUE COMMAND
5657 056162 004737 016340      JSR      PC,WAITF   ;WAIT FOR SSR TO SET
5658 056166 016501 000002      MOV      TSSR(R5),R1;GET TSSR CONTENTS
5659 056172 012702 000200      MOV      #SSR,R2   ;SET UP EXPECTED
5660 056176 020102              CMP      R1,R2     ;ARE THEY EQUAL
5661 056200 001406              BEQ      170$      ;BR, IF OK
5662 056202 005237 002212      INC      FATFLG      ;BUMP COUNT
5666 056206              ERRHRD   ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      056206 104456              TRAP     C$ERHRD
      056210 001141              .WORD   609
      056212 073610              .WORD   T26WDC
      056214 012136              .WORD   PKTSSR
5667 056216              170$:   CKLOOP      ;LOOP IF SELECTED
      056216 104406              TRAP     C$CLP1
5668 056220 013702 003114      MOV      FREE,R2   ;CURRENT BUFFER ADDRESS TO R2
5669 056224 010304              MOV      R3,R4    ;CURRENT RECORD SIZE
5670 056226 162704 000400      SUB      #256.,R4  ;FIRST LOCATION IN BUFFER
5671 056232 060204              173$:   ADD      R2,R4    ;SET UP POINTER
5672 056234 021403              CMP      (R4),R3  ;CHECK DATA READ (R3=DATA ALSO)
5673 056236 001410              BEQ      180$      ;BR, IF ALL IS WELL
5674 056240 011401              MOV      (R4),R1  ;RECD DATA
5675 056242 010302              MOV      R3,R2   ;EXPECTED DATA
5676 056244 005237 002212      INC      FATFLG      ;BUMP COUNT
5680 056250              ERRHRD   ERRNO,T26DTA,EXPREC ;DATA READ NOT = WRITTEN
      056250 104456              TRAP     C$ERHRD
      056252 001142              .WORD   610
      056254 073032              .WORD   T26DTA
      056256 015564              .WORD   EXPREC
5681 056260              180$:   CKLOOP      ;LOOP IF SELECTED
      056260 104406              TRAP     C$CLP1
5682 056262 005724              TST      (R4)+    ;BUMP TO NEXT LOCATION
5683 056264 160204              SUB      R2,R4    ;CORRECT RECORDS SIZE VALUE
5684 056266 020403              CMP      R4,R3    ;END OF RECORD YET

```


5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709

↑
:TEST 6, SUBTEST 2
:VERIFIES THAT THE REREAD PREVIOUS COMMAND WITH OPP=0
:AND SWB=1 OPERATES PROPERLY. THE TEST SEQUENCE IS
:THE SAME AS THAT USED IN SUBTEST 1, BUT IT IS
:VERIFIED THAT DATA STORED BY THE COMMAND CONTAINS
:SWAPPED BYTES.

5710 056326
056326
056326 104402
5711 056330 004737 074510
5712 056334 004737 074602
5713 056340 004737 074644
5714
5715
5716
5717
5718
5719
5720

BGNSUB :>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
T6.2:
TRAP C\$BSUB
JSR PC,T26REST ;SET COMMAND PACKET
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
:*****

5721 056344 004737 016064
5722 056350 103407
5723 056352 005237 002212
5727 056356 010001
5728 056360

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 611
.WORD SFIERR
.WORD SFIMSG

5729 056370 013737 002172 071620 20\$:
5730
5731 056376 012704 071600
5732
5733
5734
5735
5736
5737
5738

MOV UNITN,T26DSW ;SET UP UNIT NUMBER
MOV #T26PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS

:WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
:*****

5739 056402 004737 010752
5740 056406 103407
5741 056410 005237 002212
5745 056414 010001
5746 056416

JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
BCS 26\$;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 612
.WORD WRTMSG
.WORD SFIMSG

5747 056426
056426 104406
5748

26\$: CKLOOP ;LOOP IF SELECTED
TRAP C\$CLP1


```

5749
5750
5751
5752
5753
5754
5755 056430 004737 011104      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
5756 056434 103413      BCS      30$            ;BR, IF NO PROBLEM
5757 056436 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR
5758 056442 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED TSSR
5759 056446 010004      MOV      R0,R4         ;PACKET ADDRESS SET UP
5760 056450 005237 002212      INC      FATFLG        ;BUMP COUNT
5764 056454      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      056454 104456      TRAP      C$ERHRD
      056456 001145      .WORD    613
      056460 073254      .WORD    T26RWN
      056462 012136      .WORD    PKTSSR
5765 056464      30$:   CKLOOP          ;LOOP IF SELECTED      TRAP      C$CLP1
      056464 104406
5766
5767
5768
5769
5770
5771
5772
5773 056466 013701 071630      MOV      T26BFR+6,R1   ;PICK UP XST0
5774 056472 010102      MOV      R1,R2         ;SET UP EXPECTED
5775 056474 052702 000002      BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
5776 056500 020102      CMP      R1,R2         ;DOES EXP = REC'D
5777 056502 001406      BEQ      40$           ;BR, IF EQUAL (OK)
5778 056504 005237 002212      INC      FATFLG        ;BUMP COUNT
5782 056510      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      056510 104456      TRAP      C$ERHRD
      056512 001146      .WORD    614
      056514 072765      .WORD    T26BOT
      056516 015564      .WORD    EXPREC
5783 056520      40$:   CKLOOP          ;LOOP IF SELECTED      TRAP      C$CLP1
      056520 104406
5784 056522 012703 000400      MOV      #256.,R3      ;RECORD SIZE
5785 056526 013737 003114 071722      MOV      FREE,T26RB    ;STARTING WRITE BUFFER ADDRESS
5786
5787
5788
5789
5790
5791
5792
5793 056534 012737 110005 071720      MOV      #110005,T26PK3 ;WRITE DATA,ACK,SWB COMMAND
5794 056542 012704 071720      MOV      #T26PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
5795 056546      65$:   MOV      R3,R0    ;SET PATTERN IN CORRECT REGISTER
5796 056546 010300      JSR      PC,FILLMEM    ;FILL MEMORY WITH RECORD SIZE
5797 056550 004737 017512      MOV      R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
5798 056554 010337 071726      MOV      R4,TSD8(R5)   ;ISSUE COMMAND
5799 056560 010465 000000      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
5800 056564 094737 016340      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
5801 056570 016501 000002

```

```

5802 056574 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
5803 056600 020102      CMP      R1,R2      ;ARE THEY EQUAL
5804 056602 001406      BEQ      75$      ;BR, IF OK
5805 056604 005237 002212      INC      FATFLG      ;BUMP COUNT
5809 056610      ERRHRD  ERRNO,WRterr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      056610 104456      TRAP      CSERHRD
      056612 001147      .WORD    615
      056614 005111      .WORD    WRERR
      056616 012136      .WORD    PKTSSR
5810 056620      75$:  CKLOOP      ;LOOP IF SELECTED      TRAP      CSCLP1
      056620 104406
5811 056622 005723      TST      (R3)+      ;BUMP RECORD SIZE
5812 056624 022703 000414      CMP      #268.,R3   ;END OF RECORD YET
5813 056630 001346      BNE      65$      ;BR, IF MORE RECORDS TO WRITE
5814 056632      80$:  CKLOOP      ;LOOP IF SELECTED      TRAP      CSCLP1
      056632 104406
5815 056634      120$:
5816
5817      ;*****
5818      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5819      ;*****
5820
5821
5822
5823 056634 004737 011104      JSR      PC,REWIND   ;CALL TAPE REWIND COMMAND
5824 056640 103413      BCS      130$      ;BR, IF NO PROBLEM
5825 056642 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR
5826 056646 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED TSSR
5827 056652 010004      MOV      R0,R4      ;PACKET ADDRESS SET UP
5828 056654 005237 002212      INC      FATFLG      ;BUMP COUNT
5832 056660      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      056660 104456      TRAP      CSERHRD
      056662 001150      .WORD    616
      056664 073254      .WORD    T26RWN
      056666 012136      .WORD    PKTSSR
5833 056670      130$:  CKLOOP      ;LOOP IF SELECTED      TRAP      CSCLP1
      056670 104406
5834
5835      ;*****
5836      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
5837      ;*****
5838
5839
5840
5841 056672 013701 071630      MOV      T26BFR+6,R1 ;PICK UP XST0
5842 056676 010102      MOV      R1,R2      ;SET UP EXPECTED
5843 056700 052702 000002      BIS      #BIT1,R2   ;SET BOT BIT IN EXPECTED
5844 056704 020102      CMP      R1,R2      ;DOES EXP = REC'D
5845 056706 001406      BEQ      140$      ;BR, IF EQUAL (OK)
5846 056710 005237 002212      INC      FATFLG      ;BUMP COUNT
5850 056714      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      056714 104456      TRAP      CSERHRD
      056716 001151      .WORD    617
      056720 072765      .WORD    T26BOT
      056722 015564      .WORD    EXPREC
5851 056724      140$:  CKLOOP      ;LOOP IF SELECTED      TRAP      CSCLP1
      056724 104406

```

```
5852 056726 012737 000400 071752      MOV      #256.,T26RSZ      ;SET UP RECORD SIZE
5853
5854      ;*****
5855      ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
5856      ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
5857      ;*****
5858
5859
5860
5861 056734 012703 000001      145$:  MOV      #1,R3      ;SPACE ONE RECORD PARAMETER
5862 056740 004737 010556      JSR      PC,SPACE      ;CALL SPACE ROUTINE
5863 056744 103412      BCS     150$      ;BR, IF NO PROBLEM WITH SPACE COMMAND
5864 056746 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR
5865 056752 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED TSSR
5866 056756 005237 002212      INC      FATFLG      ;BUMP COUNT
5870 056762      ERRHRD  ERRNO,T26SC,EXPREC      ;POSITION (SPACE RECORDS) FAILED
      TRAP      C$ERHRD
      .WORD     618
      .WORD     T26SC
      .WORD     EXPREC
5871 056772      150$:  CKLOOP      TRAP      C$CLP1
      056772 104406
5872 056774 013703 071752      MOV      T26RSZ,R3      ;RECORD SIZE
5873 057000 013737 003114 071722      MOV      FREE,T26RB      ;STARTING READ BUFFER ADDRESS
5874
5875      ;*****
5876      ;REREAD DATA,CVC=1,ACK,SWB COMMAND
5877      ;*****
5878
5879
5880
5881 057006 012737 151001 071720      165$:  MOV      #151001,T26PK3      ;REREAD DATA,CVC=1,ACK,SWB COMMAND
5882 057014 012704 071720      MOV      #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
5883 057020 010337 071726      MOV      R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
5884 057024 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
5885 057030 004737 016340      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
5886 057034 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
5887 057040 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
5888 057044 020102      CMP      R1,R2      ;ARE THEY EQUAL
5889 057046 001406      BEQ     170$      ;BR, IF OK
5890 057050 005237 002212      INC      FATFLG      ;BUMP COUNT
5894 057054      ERRHRD  ERRNO,T26WDC,PKTSSR      ;TSSR INCORRECT AFTER REREAD DATA
      TRAP      C$ERHRD
      .WORD     619
      .WORD     T26WDC
      .WORD     PKTSSR
5895 057064      170$:  CKLOOP      ;LOOP IF SELECTED
      TRAP      C$CLP1
      057064 104406
5896 057066 013702 003114      MOV      FREE,R2      ;CURRENT BUFFER ADDRESS TO R2
5897 057072 010304      MOV      R3,R4      ;CURRENT RECORD SIZE
5898 057074 162704 000400      SUB     #256.,R4      ;FIRST LOCATION IN BUFFER
5899 057100 060204      173$:  ADD      R2,R4      ;SET UP POINTER
5900 057102 021403      CMP     (R4),R3      ;CHECK DATA READ (R3=DATA ALSO)
5901 057104 001410      BEQ     180$      ;BR, IF ALL IS WELL
5902 057106 011401      MOV     (R4),R1      ;RECD DATA
5903 057110 010302      MOV     R3,R2      ;EXPECTED DATA
5904 057112 005237 002212      INC     FATFLG      ;BUMP COUNT
```

```
5908 057116          ERRHRD  ERRNO,T26DTA,EXPREC      ;DATA READ NOT = WRITTEN
      057116 104456
      057120 001154
      057122 073032
      057124 015564
5909 057126          180$:  CKLOOP                ;LOOP IF SELECTED
      057126 104406
      5910 057130 005724          TST      (R4)+          ;BUMP TO NEXT LOCATION
      5911 057132 160204          SUB      R2,R4         ;CORRECT RECORDS SIZE VALUE
      5912 057134 020403          CMP      R4,R3         ;END OF RECORD YET
      5913 057136 001360          BNE     173$          ;BR, IF NOT AT END OF RECORD
      5914 057140 005723          TST      (R3)+          ;BUMP RECORD SIZE
      5915 057142 010337 071752   MOV      R3,T26RSZ     ;STORE RECORD SIZE
      5916 057146 022703 000412   CMP      #266.,R3     ;END OF RECORD YET
      5917 057152 001270          BNE     145$          ;BR, IF MORE RECORDS TO READ
      5918 057154          190$:  CKLOOP                ;LOOP IF SELECTED
      057154 104406
      5919 057156          ENDSUB
      057156
      057156 104403
      5920 057160 023727 002212 000017  CMP      FATFLG,#15.   ;IS ERROR COUNT AT 25
      5921 057166 103402          BLO     999$          ;BR, IF LESS THAN 25
      5922 057170 004737 017272   JSR     PC,CKDROP     ;TRY TO DROP THE UNIT
      5923 057174          999$:
```

5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957

TEST 6, SUBTEST 3

VERIFIES THAT THE REREAD PREVIOUS COMMAND WITH OPP=1 (READ REVERSE, SPACE FORWARD) AND SWB=0 OPERATES PROPERLY. THE TAPE IS FIRST REWOUND AND THEN WRITTEN WITH A SERIES OF TEST RECORDS VARYING IN LENGTH AND DATA CONTENT. THE FIRST FOUR BYTES OF EACH RECORD CONTAIN ITS RECORD NUMBER (INDICATING POSITION ON TAPE). THE TAPE IS THEN REWOUND AGAIN. FOR EACH TEST RECORD, THE FOLLOWING SEQUENCE IS EXECUTED.

- 1. THE REREAD PREVIOUS COMMAND WITH OPP=1 IS ISSUED AND THE RESULTS CHECKED
- 2. A READ FORWARD COMMAND IS THEN ISSUED AND THE DATA IS CHECKED TO VERIFY THAT THE TAPE WAS POSITIONED PROPERLY AFTER THE REREAD PREVIOUS COMMAND (E.G. THE TAPE SHOULD HAVE BEEN LEFT POSITIONED AT THE START OF THE TEST RECORD). THE READ FORWARD COMMAND LEAVES THE TAPE POSITIONED PROPERLY AT THE START OF THE NEXT TEST RECORD.

THE BYTE COUNT ON EACH REREAD PREVIOUS COMMAND IS SET TO THE LENGTH OF THE EXPECTED RECORD, SO NO EXCEPTIONAL CONDITIONS SHOULD OCCUR.

```
5958 057174          BGNSUB          ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>>
      057174          T6.3:
      057174 104402          TRAP          CSBSUB
5959 057176 004737 074510    JSR          PC,T26REST    ;SET COMMAND PACKET
5960 057202 005037 071746    CLR          T26CNT        ;CLEAR TAPE RECORD COUNTER
5961 057206 004737 074602    JSR          PC,T26RT2     ;SET UP OTHER COMMAND PACKET
5962 057212 004737 074644    JSR          PC,T26RT3     ;SET UP OTHER COMMAND PACKET
5963
5964 *****
5965 :ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
5966 *****
5967
5968
5969
5970 057216 004737 016064    JSR          PC,SOFINIT    ;DO INITIALIZE ON CONTROLLER
5971 057222 103407          BCS          20$          ;BR IF INIT WAS OK
5972 057224 005237 002212    INC          FATFLG        ;BUMP COUNT
5976 057230 010001          MOV          R0,R1         ;CONTENTS OF TSSR REGISTER
5977 057232          ERRDF    ERRNC,SF!ERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      057232 104455          TRAP          CSERDF
      057234 001155          .WORD        621
      057236 003650          .WORD        SFIERR
      057240 012124          .WORD        SFIMSG
5978 057242 013737 002172 071620 20$:  MOV          UNITN,T26DSW ;SET UP UNIT NUMBER
```

```
5979
5980 057250 012704 071600          MOV      #T26PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
5981
5982          ;*****
5983          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
5984          ;*****
5985
5986
5987
5988 057254 004737 010752          JSR      PC,WRTCHR              ;ISSUE WRITE CHARACTERISTICS
5989 057260 103407                  BCS     26$                    ;BR, IF COMMAND ISSUED OK
5990 057262 005237 002212          INC     FATFLG                 ;BUMP COUNT
5994 057266 010001                  MOV     RO,R1                  ;SAVE CONTENTS OF TSSR
5995 057270          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP     CSERHRD
                                .WORD    622
                                .WORD    WRTMSG
                                .WORD    SFIMSG
                                TRAP     C$CLP1
                                CSERHRD
                                622
                                WORD    WRTMSG
                                WORD    SFIMSG
057270 104456
057272 001156
057274 005054
057276 012124
5996 057300          26$:  CKLOOP                    ;LOOP IF SELECTED
057300 104406          TRAP     C$CLP1
5997
5998          ;*****
5999          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6000          ;*****
6001
6002
6003
6004 057302          '737 011104          JSR      PC,REWIND              ;CALL TAPE REWIND COMMAND
6005 057306          13                    BCS     30$                    ;BR, IF NO PROBLEM
6006 057310          016501 000002          MOV     TSSR(R5),R1            ;GET TSSR
6007 057314          012702 000200          MOV     #SSR,R2                ;SET UP EXPECTED TSSR
6008 057320 010004                  MOV     RO,R4                  ;PACKET ADDRESS SET UP
6009 057322 005237 002212          INC     FATFLG                 ;BUMP COUNT
6013 057326          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP     CSERHRD
                                .WORD    623
                                .WORD    T26RWN
                                .WORD    PKTSSR
                                TRAP     C$CLP1
                                CSERHRD
                                623
                                WORD    T26RWN
                                WORD    PKTSSR
057326 104456
057330 001157
057332 073254
057334 012136
6014 057336          30$:  CKLOOP                    ;LOOP IF SELECTED
057336 104406          TRAP     C$CLP1
6015
6016          ;*****
6017          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6018          ;*****
6019
6020
6021
6022 057340 013701 071630          MOV     T26BFR+6,R1            ;PICK UP XSTO
6023 057344 010102                  MOV     R1,R2                  ;SET UP EXPECTED
6024 057346 052702 000002          BIS     #BIT1,R2                ;SET BOT BIT IN EXPECTED
6025 057352 020102                  CMP     R1,R2                  ;DOES EXP = REC'D
6026 057354 001406                  BEQ     40$                    ;BR, IF EQUAL (OK)
6027 057356 005237 002212          INC     FATFLG                 ;BUMP COUNT
6031 057362          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP     CSERHRD
                                .WORD    624
                                .WORD    T26BOT
057362 104456
057364 001160
057366 072765
```



```
6085 057560 012136          130$:  CKLOOP                ;LOOP IF SELECTED          .WORD  PKTSSR
      057562                ;                               TRAP    C$CLP1
      057562 104406
6086
6087
6088
6089
6090
6091
6092
6093 057564 013701 071630          MOV    T26BFR+6,R1          ;PICK UP XSTO
6094 057570 010102          MOV    R1,R2                ;SET UP EXPECTED
6095 057572 052702 000002          BIS    #BIT1,R2             ;SET BOT BIT IN EXPECTED
6096 057576 020102          CMP    F1,R2                ;DOES EXP = REC'D
6097 057600 001406          BEQ    140$                  ;BR, IF EQUAL (OK)
6098 057602 005237 002212          INC    FATFLG                ;BUMP COUNT
6102 057606                ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      057606 104456                TRAP    C$ERHRD
      057610 001163                .WORD  627
      057612 072765                .WORD  T26BOT
      057614 015564                .WORD  EXPREC
6103 057616                140$:  CKLOOP                ;LOOP IF SELECTED          .WORD  PKTSSR
      057616 104406                TRAP    C$CLP1
6104
6105
6106
6107
6108
6109
6110
6111
6112 057620 012703 000001          MOV    #1,R3                 ;SPACE 1 RECORD FORWARD
6113 057624 004737 010556          JSR    PC,SPACE              ;SPACE CALL
6114 057630 012703 000400          MOV    #256,R3               ;RECORD SIZE
6115 057634 013737 003114 071722 150$:  MOV    FREE,T26RB            ;STARTING READ BUFFER ADDRESS
6116
6117
6118
6119
6120
6121
6122
6123 057642 012737 161001 071720 165$:  MOV    #161001,T26PK3        ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6124 057650 012704 071720          MOV    #T26PK3,R4           ;SET UP R4 WITH PACKET ADDRESS
6125 057654 010337 071726          MOV    R3,T26SZ              ;SET UP RECORD SIZE IN PACKET
6126 057660 010465 000000          MOV    R4,TSDB(R5)           ;ISSUE COMMAND
6127 057664 004737 016340          JSR    PC,WAITF              ;WAIT FOR SSR TO SET
6128 057670 016501 000002          MOV    1SSR(R5),R1           ;GET TSSR CONTENTS
6129 057674 012702 000200          MOV    #SSR,R2               ;SET UP EXPECTED
6130 057700 020102          CMP    R1,R2                 ;ARE THEY EQUAL
6131 057702 001406          BEQ    170$                  ;BR, IF OK
6132 057704 005237 002212          INC    FATFLG                ;BUMP COUNT
6136 057710                ERRHRD  ERRNO,T26RRG,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      057710 104456                TRAP    C$ERHRD
      057712 001164                .WORD  628
      057714 072272                .WORD  T26RRG
      057716 012136                .WORD  PKTSSR
```



```
6137 057720 170$: CKLOOP ;LOOP IF SELECTED
      057720 104406 ;BUMP RECORD SIZE TRAP CSCLP1
6138 057722 005723 TST (R3)+ ;BUMP TAPE RECORD COUNTER
6139 057724 062737 000001 071746 ADD #1,T26CNT
6140
6141 :*****
6142 :READ DATA, CVC=1, ACK COMMAND
6143 :*****
6144
6145
6146
6147 057732 012737 140001 071720 MOV #140001,T26PK3 ;READ DATA, CVC=1, ACK COMMAND
6148 057740 010337 071726 MOV R3,T26SZ ;SET SIZE INTO PACKET
6149 057744 010465 000000 MOV R4,TSDB(R5) ;ISSUE READ DATA COMMAND
6150 057750 004737 016340 JSR PC,WAITF ;WAIT FOR SSR
6151 057754 016501 000002 MOV TSSR(R5),R1 ;PICK UP THE TSSR
6152 057760 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6153 057764 020102 CMP R1,R2 ;IS THE TSSR OK
6154 057766 001406 BEQ 195$ ;BR, IF TSSR OK (GOOD)
6155 057770 005237 002212 INC FATFLG ;BUMP COUNT
6159 057774 ERRHRD ERRNO,RDERR,PKTSSR ;READ DATA COMMAND FAILED
      057774 104456 TRAP CSERHRD
      057776 001165 .WORD 629
      060000 005204 .WORD RDERR
      060002 012136 .WORD PKTSSR
6160 060004 195$: CKLOOP ;LOOP IF SELECTED
      060004 104406 TRAP CSCLP1
6161 060006 017701 123102 MOV @FREE,R1 ;FIRST WORD FROM READ BUFFER
6162 060012 013702 071746 MOV T26CNT,R2 ;SET UP EXPECTED
6163 060016 020102 CMP R1,R2 ;IS TAPE POSITION CORRECT
6164 060020 001406 BEQ 197$ ;KEEP GOING POSITION OK
6165 060022 005237 002212 INC FATFLG ;BUMP COUNT
6169 060026 ERRHRD ERRNO,T26WNG,EXPRES ;TAPE POSITION INCORRECT
      060026 104456 TRAP CSERHRD
      060030 001166 .WORD 630
      060032 071756 .WORD T26WNG
      060034 015564 .WORD EXPRES
6170 060036 197$: CKLOOP
      060036 104406 TRAP CSCLP1
6171 060040 022703 000412 CMP #266.,R3 ;AT MAX SIZE YET
6172 060044 001401 BEQ 200$ ;BR, IF AT END OF THE SUBTEST
6173 060046 000672 BR 150$ ;KEEP GOING MORE RECORDS
6174 060050
6175 060050 200$: ENDSUB ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
      060050 L10105:
      060050 104403 TRAP CSesub
6176 060052 023727 002212 000017 CMP FATFLG,#15. ;IS ERROR COUNT AT 25
6177 060060 103402 BLO 999$ ;BR, IF LESS THAN 25
6178 060062 004737 017272 JSR PC,CKDROP ;TRY TO DROP THE UNIT
6179 060066 999$:
```

6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191
6192
6193
6194
6195
6196

```

:+
:TEST 6, SUBTEST 4
:VERIFIES THAT THE REREAD PREVIOUS COMMAND WITH OPP=1
:AND SWB=1 OPERATES PROPERLY. THE TEST SEQUENCE IS
:THE SAME THAT IS USED IN SUBTEST 3, BUT IT IS
:VERIFIED THAT DATA STORED BY THE COMMAND CONTAINS
:SWAPPED BYTES.
:-
  
```

```

6196 060066          BGNSUB                      ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
      060066          T6.4:
      060066 104402          TRAP          C$BSUB
6197 060070 004737 074510      JSR          PC,T26REST      ;SET COMMAND PACKET
6198 060074 005037 071746      CLR          T26CNT      ;CLEAR TAPE RECORD COUNTER
6199 060100 004737 074602      JSR          PC,T26RT2   ;SET UP OTHER COMMAND PACKET
6200 060104 004737 074644      JSR          PC,T26RT3   ;SET UP OTHER COMMAND PACKET
  
```

```

6201
6202
6203
6204 :*****
6205 :ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
6206 :*****
6207
  
```

```

6208 060110 004737 016064      JSR          PC,SOFINIT  ;DO INITIALIZE ON CONTROLLER
6209 060114 103407          BCS          20$        ;BR IF INIT WAS OK
6210 060116 005237 002212      INC          FATFLG      ;BUMP COUNT
6214 060122 010001          MOV          R0,R1      ;CONTENTS OF TSSR REGISTER
6215 060124          ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      060124 104455          TRAP          C$ERDF
      060126 001167          .WORD        631
      060130 003650          .WORD        SFIERR
      060132 012124          .WORD        SFIMSG
6216 060134 013737 002172 071620 20$: MOV          UNITN,T26DSW ;SET UP UNIT NUMBER
6217
6218 060142 012704 071600      MOV          #T26PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
6219
  
```

```

6220 :*****
6221 :WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
6222 :*****
6223
6224
6225
  
```

```

6226 060146 004737 010752      JSR          PC,WRTCHR   ;ISSUE WRITE CHARACTERISTICS
6227 060152 103407          BCS          26$        ;BR, IF COMMAND ISSUED OK
6228 060154 005237 002212      INC          FATFLG      ;BUMP COUNT
6232 060160 010001          MOV          R0,R1      ;SAVE CONTENTS OF TSSR
6233 060162          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      060162 104456          TRAP          C$FRHRD
      060164 001170          .WORD        632
      060166 005054          .WORD        WRTMSG
      060170 012124          .WORD        SFIMSG
  
```

```
6234 060172 104406 26$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
060172
6235
6236
6237
6238 :*****
6239 :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6240 :*****
6241
6242 060174 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6243 060200 103413 BCS 30$ ;BR, IF NO PROBLEM
6244 060202 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
6245 060206 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
6246 060212 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
6247 060214 005237 002212 INC FATFLG ;BUMP COUNT
6251 060220 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
060220 104456 TRAP C$ERHRD
060222 001171 .WORD 633
060224 073254 .WORD T26RWN
060226 012136 .WORD PKTSSR
6252 060230 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
060230 104406
6253
6254 :*****
6255 :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6256 :*****
6257
6258
6259
6260 060232 013701 071630 MOV T26BFR+6,R1 ;PICK UP XSTO
6261 060236 010102 MOV R1,R2 ;SET UP EXPECTED
6262 060240 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6263 060244 020102 CMP R1,R2 ;DOES EXP = REC'D
6264 060246 001406 BEQ 40$ ;BR, IF EQUAL (OK)
6265 060250 005237 002212 INC FATFLG ;BUMP COUNT
6269 060254 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
060254 104456 TRAP C$ERHRD
060256 001172 .WORD 634
060260 071765 .WORD T26BOT
060262 011564 .WORD EXPREC
6270 060264 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
060264 104406
6271 060266 012703 000400 MOV #256.,R3 ;RECORD SIZE
6272 060272 013737 003114 071722 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
6273
6274 :*****
6275 :WRITE DATA,CVC=1,ACK COMMAND
6276 :*****
6277
6278
6279
6280 060300 012737 140005 071720 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
6281 060306 012704 071720 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6282 060312 65$:
6283 060312 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
6284 060314 004737 017512 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
6285 060320 010337 071726 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
```



```

060502 0G1175
060504 072765
060506 015564
6341 060510 140$: CKLOOP ;LOOP IF SELECTED
060510 104406 TRAP C$CLP1
6342
6343
6344 :*****
6345 :ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
6346 :BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
6347 :*****
6348
6349
6350 060512 012703 000001 MOV #1,R3 ;SET UP SPACE FORWARD 1
6351 060516 004737 010556 JSR PC,SPACE ;ISSUE SPACE COMMAND
6352 060522 012703 000400 MOV #256,R3 ;RECORD SIZE
6353 060526 013737 003114 071722 150$: MOV FREE,T26RB ;STARTING READ BUFFER ADDRESS
6354
6355 :*****
6356 :REREAD DATA,CVC=1,ACK, OPP COMMAND
6357 :*****
6358
6359
6360
6361 060534 012737 171001 071720 165$: MOV #171001,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6362 060542 012704 071720 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6363 060546 010337 071726 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6364 050552 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6365 060556 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
6366 060562 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6367 060566 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6368 060572 020102 CMP R1,R2 ;ARE THEY EQUAL
6369 060574 001406 BEQ 170$ ;BR, IF OK
6370 060576 005237 002212 INC FATFLG ;BUMP COUNT
6374 060602 ERRHRD ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
060602 104456 TRAP C$ERHRD
060604 001176 .WORD 638
060606 072175 .WORD T26RRF
060610 012136 .WORD PKTSSR
6375 060612 170$: CKLOOP ;LOOP IF SELECTED
060612 104406 TRAP C$CLP1
6376 060614 017701 122274 MOV @FREE,R1 ;FIRST WORD FROM READ BUFFER
6377 060620 013702 071746 MOV T26CNT,R2 ;SET UP EXPECTED
6378 060624 000302 SWAB R2 ;SWAP BYTES IN EXPECTED
6379 060626 020102 CMP R1,R2 ;IS TAPE POSITION CORRECT
6380 060630 001406 BEQ 190$ ;KEEP GOING POSITION OK
6381 060632 005237 002212 INC FATFLG ;BUMP COUNT
6385 060636 ERRHRD ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
060636 104456 TRAP C$ERHRD
060640 001177 .WORD 639
060642 071756 .WORD T26WNG
060644 015564 .WORD EXPREC
6386 060646 190$: CKLOOP
060646 104406 TRAP C$CLP1
6387 060650 005723 TST (R3)+ ;NEXT RECORD SIZE
6388 060652 062737 000001 071746 ADD #1,T26CNT ;BUMP TAPE RECORD COUNTER
6389

```



```
061114 104406 TRAP C$CLP1
6483
6484
6485
6486
6487
6488
6489
6490 061116 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6491 061122 103413 BCS 30$ ;BR, IF NO PROBLEM
6492 061124 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
6493 061130 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
6494 061134 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
6495 061136 005237 002212 INC FATFLG ;BUMP COUNT
6499 061142 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
061142 104456 TRAP C$ERHRD
061144 001204 .WORD 644
061146 073254 .WORD T26RWN
061150 012136 .WORD PKTSSR
6500 061152 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
061152 104406
6501
6502
6503
6504
6505
6506
6507
6508 061154 013701 071630 MOV T26BFR+6,R1 ;PICK UP XST0
6509 061160 010102 MOV R1,R2 ;SET UP EXPECTED
6510 061162 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6511 061166 020102 CMP R1,R2 ;DOES EXP = REC'D
6512 061170 001406 BEQ 40$ ;BR, IF EQUAL (OK)
6513 061172 005237 002212 INC FATFLG ;BUMP COUNT
6517 061176 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
061176 104456 TRAP C$ERHRD
061200 001205 .WORD 645
061202 072765 .WORD T26BOT
061204 015564 .WORD EXPREC
5518 061206 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
061206 104406
6519 061210 012703 001000 MOV #512.,R3 ;RECORD SIZE
6520 061214 013737 003114 071722 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
6521
6522
6523
6524
6525
6526
6527
6528 061222 012737 140005 071720 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
6529 061230 012704 071720 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6530 061234
6531 061234 010337 071726 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6532 061240 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6533 061244 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
6534 061250 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
```



```
6535 061254 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
6536 061260 020102      CMP      R1,R2      ;ARE THEY EQUAL
6537 061262 001406      BEQ      75$      ;BR, IF OK
6538 061264 005237 002212      INC      FATFLG      ;BUMP COUNT
6542 061270      ERRHRD  ERRNO,WRterr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
        061270 104456      TRAP      C$ERHRD
        061272 001206      .WORD    646
        061274 005111      .WORD    WRterr
        061276 012136      .WORD    PKTSSR
6543 061300      75$:  CKLOOP      ;LOOP IF SELECTED
        061300 104406      TRAP      C$CLP1
6544 061302 005303      DEC      R3      ;SET RECORD SIZE TO 511.
6545 061304 013737 003114 071722      MOV      FREE,T26RB ;STARTING READ BUFFER ADDRESS
6546
6547      ;*****
6548      ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
6549      ;*****
6550
6551
6552
6553 061312 012737 161001 071720      MOV      #161001,T26PK3 ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
6554 061320 012704 071720      165$:  MOV      #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6555 061324 010337 071726      MOV      R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6556 061330 010465 000000      MOV      R4,TSDB(R5) ;ISSUE COMMAND
6557 061334 004737 016340      JSR      PC,WAITF ;WAIT FOR SSR TO SET
6558 061340 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
6559 061344 012702 100204      MOV      #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6560 061350 020102      CMP      R1,R2      ;ARE THEY EQUAL
6561 061352 001406      BEQ      170$      ;BR, IF OK
6562 061354 005237 002212      INC      FATFLG      ;BUMP COUNT
6566 061360      ERRHRD  ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
        061360 104456      TRAP      C$ERHRD
        061362 001207      .WORD    647
        061364 074332      .WORD    T26TRL
        061366 012136      .WORD    PKTSSR
6567 061370      170$:  CKLOOP      ;LOOP IF SELECTED
        061370 104406      TRAP      C$CLP1
6568
6569      ;*****
6570      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6571      ;*****
6572
6573
6574
6575 061372 013701 071630      MOV      T26BFR+6,R1 ;GET MESSAGE BUFFER
6576 061376 010102      MOV      R1,R2      ;SET UP EXPECTED
6577 061400 052702 010000      BIS      #BIT12,R2 ;SET THE RLL BIT IN EXPECTED
6578 061404 020102      CMP      R1,R2      ;ARE THEY EQUAL
6579 061406 001406      BEQ      180$      ;BR, IF EQUAL (ALL IS WELL)
6580 061410 005237 002212      INC      FATFLG      ;BUMP COUNT
6584 061414      ERRHRD  ERRNO,T26LON,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
        061414 104456      TRAP      C$ERHRD
        061416 001210      .WORD    648
        061420 074100      .WORD    T26LON
        061422 015564      .WORD    EXPREC
6585 061424      180$:  CKLOOP      ;LOOP IF SELECTED
        061424 104406      TRAP      C$CLP1
```

```

6586 061426 012703 000777         MOV    #511.,R3           ;SET RECORD SIZE
6587 061432 013737 003114 071722  MOV    FREE,T26RB       ;STARTING READ BUFFER ADDRESS
6588
6589                               ;*****
6590                               ;REREAD DATA,CVC=1,ACK COMMAND
6591                               ;*****
6592
6593
6594
6595 061440 012737 141001 071720 365$:  MOV    #141001,T26PK3   ;REREAD DATA,CVC=1,ACK COMMAND
6596 061446 012704 071720         MOV    #T26PK3,R4       ;SET UP R4 WITH PACKET ADDRESS
6597 061452 010337 071726         MOV    R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
6598 061456 010465 000000         MOV    R4,T26SDB(R5)   ;ISSUE COMMAND
6599 061462 004737 016340         JSR    PC,WAITF        ;WAIT FOR SSR TO SET
6600 061466 016501 000002         MOV    T26SSR(R5),R1   ;GET T26SSR CONTENTS
6601 061472 012702 100204         MOV    #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6602 061476 020102                 CMP    R1,R2           ;ARE THEY EQUAL
6603 061500 001406                 BEQ    370$            ;BR, IF OK
6604 061502 005237 002212         INC    FATFLG          ;BUMP COUNT
6608 061506                 ERRHRD ERRNO,T26TRL,PKTSSR ;T26SSR INCORRECT AFTER REREAD DATA
6609 061516 104456                 TRAP   C$ERRHRD
6610 061516 104406                 .WORD 649
6611                               .WORD T26TRL
6612                               .WORD PKTSSR
6613                               TRAP   C$CLP1
6614                               ;*****
6615                               ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6616                               ;*****
6617 061520 013701 071630         MOV    T26BFR+6,R1     ;GET MESSAGE BUFFER
6618 061524 010102                 MOV    R1,R2           ;SET UP EXPECTED
6619 061526 052702 010000         BIS    #BIT12,R2       ;SET THE RLL BIT IN EXPECTED
6620 061532 020102                 CMP    R1,R2           ;ARE THEY EQUAL
6621 061534 001406                 BEQ    380$            ;BR, IF EQUAL (ALL IS WELL)
6622 061536 005237 002212         INC    FATFLG          ;BUMP COUNT
6626 061542                 ERRHRD ERRNO,T26LON,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
6627 061542 104456                 TRAP   C$ERRHRD
6628 061544 001212                 .WORD 650
6629 061546 074100                 .WORD T26LON
6630 061550 015564                 .WORD EXPREC
6631 061552 104406                 TRAP   C$CLP1
6632 061552 104403                 TRAP   C$ESUB
6633 061554 023727 002212 000017     CMP    FATFLG,#15.     ;IS ERROR COUNT AT 25
6634 061556 103402                 BLO   999$            ;BR, IF LESS THAN 25
6635 061566 004737 017272         JSR    PC,CKDROP      ;TRY TO DROP THE UNIT
6636 061572 999$:

```



```
6688 061670 012124
      061672
      061672 104406
      26$: CKLOOP ;LOOP IF SELECTED .WORD SFIMSG
                                     TRAP C$CLP1
6689
6690
6691
6692
6693
6694
6695
6696 061674 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6697 061700 103413 BCS 30$ ;BR, IF NO PROBLEM
6698 061702 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
6699 061706 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
6700 061712 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
6701 061714 005237 002212 INC FATFLG ;BUMP COUNT
6705 061720 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      061720 104456 TRAP C$ERHRD
      061722 001215 .WORD 653
      061724 073254 .WORD T26RWN
      061726 012136 .WORD PKTSSR
6706 061730 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      061730 104406
6707
6708
6709
6710
6711
6712
6713
6714 061732 013701 071630 MOV T26BFR+6,R1 ;PICK UP XST0
6715 061736 010102 MOV R1,R2 ;SET UP EXPECTED
6716 061740 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6717 061744 020102 CMP R1,R2 ;DOES EXP = REC'D
6718 061746 001406 BEQ 40$ ;BR, IF EQUAL (OK)
6719 061750 005237 002212 INC FATFLG ;BUMP COUNT
6723 061754 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      061754 104456 TRAP C$ERHRD
      061756 001216 .WORD 654
      061760 072765 .WORD T26BOT
      061762 015564 .WORD EXPREC
6724 061764 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      061764 104406
6725 061766 012703 000400 MOV #256.,R3 ;RECORD SIZE
6726 061772 013. '7 003114 071722 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
6727
6728
6729
6730
6731
6732
6733
6734 062000 012737 140005 071720 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
6735 062006 012704 071720 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6736 062012
6737 062012 010337 071726 65$: MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6738 062016 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
```

```
6739 062022 004737 016340      JSR      PC, WAITF      ;WAIT FOR SSR TO SET
6740 062026 016501 000002      MOV      TSSR(R5), R1   ;GET TSSR CONTENTS
6741 062032 012702 000200      MOV      #SSR, R2      ;SET UP EXPECTED
6742 062036 020102                CMP      R1, R2        ;ARE THEY EQUAL
6743 062040 001406      BEQ      75$           ;BR, IF OK
6744 062042 005237 002212      INC      FATFLG        ;BUMP COUNT
6748 062046                ERRHRD   ERRNO, WRTERR, PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    655
                                .WORD    WRTERR
                                .WORD    PKTSSR
                                TRAP      C$CLP1
6749 062056                75$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
6750 062060 012703 001000      MOV      #512., R3     ;RECORD SIZE
6751 062064 013737 003114 071722  MOV      FREE, T26RB   ;STARTING READ BUFFER ADDRESS
6752
6753      ;*****
6754      ;REREAD PREVIOUS, ACK, CVC=1, OPP=1
6755      ;*****
6756
6757
6758
6759 062072 012737 161001 071720      MOV      #161001, T26FK3 ;REREAD PREVIOUS, ACK, CVC=1, OPP=1
6760 062100 012704 071720 165$:   MOV      #T26PK3, R4    ;SET UP R4 WITH PACKET ADDRESS
6761 062104 010337 071726      MOV      R3, T26SZ     ;SET UP RECORD SIZE IN PACKET
6762 062110 010465 000000      MOV      R4, TSDB(R5)  ;ISSUE COMMAND
6763 062114 004737 016340      JSR      PC, WAITF     ;WAIT FOR SSR TO SET
6764 062120 016501 000002      MOV      TSSR(R5), R1  ;GET TSSR CONTENTS
6765 062124 012702 100204      MOV      #SSR!SC!BIT2, R2 ;SET UP EXPECTED
6766 062130 020102                CMP      R1, R2        ;ARE THEY EQUAL
6767 062132 001406      BEQ      170$         ;BR, IF OK
6768 062134 005237 002212      INC      FATFLG        ;BUMP COUNT
6772 062140                ERRHRD   ERRNO, T26TRL, PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    656
                                .WORD    T26TRL
                                .WORD    PKTSSR
6773 062150                170$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
6774
6775      ;*****
6776      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6777      ;*****
6778
6779
6780
6781 062152 013701 071630      MOV      T26BFR+6, R1  ;GET MESSAGE BUFFER
6782 062156 010102                MOV      R1, R2        ;SET UP EXPECTED
6783 062160 052702 040000      BIS      #BIT14, R2    ;SET THE RLS BIT IN EXPECTED
6784 062164 020102                CMP      R1, R2        ;ARE THEY EQUAL
6785 062166 001406      BEQ      180$         ;BR, IF EQUAL (ALL IS WELL)
6786 062170 005237 002212      INC      FATFLG        ;BUMP COUNT
6790 062174                ERRHRD   ERRNO, T26LOP, EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
                                TRAP      C$ERHRD
                                .WORD    657
                                .WORD    T26LOP
                                .WORD    EXPREC
062174 104456
062176 001221
062200 074162
062202 015564
```

```
6791 062204 180$: CKLOOP
      062204 104406
6792 062206 013701 071626      MOV T26BFR+4,R1      ;PICK UP RESIDUAL BYTE COUNTER
6793 062212 012702 000400      MOV #256.,R2      ;THIS SHOULD BE THE DIFFERENCE
6794 062216 020102      CMP R1,R2      ;IS THE DIFFERENCE CORRECT
6795 062220 001406      BEQ 190$      ;BR, IF CORRECT
6796 062222 005237 002212      INC FATFLG      ;BUMP COUNT
6800 062226      ERRHRD ERRNO,T26PBP,EXPREC ;RBPOR NOT CORRECT
      062226 104456      TRAP C$ERHRD
      062230 001222      .WORD 658
      062232 074244      .WORD T26PBP
      062234 015564      .WORD EXPREC
6801 062236 190$: CKLOOP      ;LOOP IF SELECTED
      062236 104406      TRAP C$CLP1
6802 062240 012703 001000      MOV #512.,R3      ;RECORD SIZE
6803 062244 013737 003114 071722      MOV FREE,T26RB      ;STARTING READ BUFFER ADDRESS
6804
6805      ;*****
6806      ;REREAD PREVIOUS,ACK,CVC=1,OPP=0
6807      ;*****
6808
6809
6810
6811 062252 012737 141001 071720      MOV #141001,T26PK3 ;REREAD PREVIOUS,ACK,CVC=1,OPP=0
6812 062260 012704 071720      MOV #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
6813 062264 010337 071726      MOV R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
6814 062270 010465 000000      MOV R4,TSDB(R5)      ;ISSUE COMMAND
6815 062274 004737 016340      JSR PC,WAITF      ;WAIT FOR SSR TO SET
6816 062300 016501 000002      MOV TSSR(R5),R1      ;GET TSSR CONTENTS
6817 062304 012702 100204      MOV #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6818 062310 020102      CMP R1,R2      ;ARE THEY EQUAL
6819 062312 001406      BEQ 270$      ;BR, IF OK
6820 062314 005237 002212      INC FATFLG      ;BUMP COUNT
6824 062320      ERRHRD ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      062320 104456      TRAP C$ERHRD
      062322 001223      .WORD 659
      062324 074332      .WORD T26TRL
      062326 012136      .WORD PKTSSR
6825 062330 270$: CKLOOP      ;LOOP IF SELECTED
      062330 104406      TRAP C$CLP1
6826
6827
6828      ;*****
6829      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6830      ;*****
6831
6832
6833 062332 013701 071630      MOV T26BFP+6,R1      ;GET MESSAGE BUFFER
6834 062336 010102      MOV R1,R2      ;SET UP EXPECTED
6835 062340 052702 040000      BIS #BIT14,R2      ;SET THE RLS BIT IN EXPECTED
6836 062344 020102      CMP R1,R2      ;ARE THEY EQUAL
6837 062346 001406      BEQ 280$      ;BR, IF EQUAL (ALL IS WELL)
6838 062350 005237 002212      INC FATFLG      ;BUMP COUNT
6842 062354      ERRHRD ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
      062354 104456      TRAP C$ERHRD
      062356 001224      .WORD 660
      062360 074162      .WORD T26LOP
```

```

6843 062362 015564           280$: CKLOOP          .WORD  EXPREC
        062364 104406             TRAP  C$CLP1
6844 062366 013701 071626    MOV    T26BFR+4,R1    ;PICK UP RESIDUAL BYTE COUNTER
6845 062372 012702 000400    MOV    #256.,R2     ;THIS SHOULD BE THE DIFFERENCE
6846 062376 020102             CMP    R1,R2       ;IS THE DIFFERENCE CORRECT
6847 062400 001405             BEQ    290$        ;BR, IF CORRECT
6851 062404             ERRHRD  ERRNO,T26PBP,EXPREC ;RBPCR NOT CORRECT
        062404 104456             TRAP  C$ERHRD
        062406 001224             .WORD  660
        062410 074244             .WORD  T26PBP
        062412 015564             .WORD  EXPREC
6852 062414           290$: CKLOOP          ;LOOP IF SELECTED
        062414 104406             TRAP  C$CLP1
6853 062416             ENDSUB           ;>>>>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>>>
        062416 104403             L10110:
6854 062420 023727 002212 000017    CMP    FATFLG,#15.    ;IS ERROR COUNT AT 25
6855 062426 103402             BLO    999$         ;BR, IF LESS THAN 25
6856 062430 004737 017272             JSR    PC,CKDROP    ;TRY TO DROP THE UNIT
6857 062434           999$:

```

6859
6860
6861
6862
6863
6864
6865
6866
6867
6868
6869
6870
6871
6872
6873
6874
6875
6876
6877
6878
6879

:+
:TEST 6, SUBTEST 7
:
:VERIFIES THAT THE REREAD NEXT COMMAND WITH OPP=0
:AND SWB=0 OPERATES PROPERLY. THE TAPE IS FIRST
:REWOUND AND THEN WRITTEN WITH A SERIES OF TEST
:RECORDS VARYING IN LENGTH AND DATA CONTENT. THE TAPE
:IS THEN REWOUND AGAIN. FOR EACH TEST RECORD, THE
:TAPE IS SPACED FORWARD ONE RECORD AND A REREAD
:NEXT COMMAND ISSUED. RESULTS (STATUS, DATA,
:ETC.) ARE VERIFIED. THE BYTE COUNT ON EACH REREAD
:NEXT COMMAND IS SET TO THE LENGTH OF THE EXPECTED
:RECORD, SO NO EXCEPTIONAL CONDITIONS SHOULD OCCUR.
:-

062434
062434
062434 104402
6880 062436 004737 074510
6881 062442 004737 074602
6882 062446 004737 074644
6883
6884
6885
6886
6887
6888
6889
6890 062452 004737 016064
6891 062456 103407
6892 062460 005237 002212
6896 062464 010001
6897 062466
062466 104455
062470 001225
062472 003650
062474 012124
6898 062476 013737 002172 071620 20\$:
6899
6900 062504 012704 071600
6901
6902
6903
6904
6905
6906
6907
6908 062510 004737 010752
6909 062514 103407
6910 062516 005237 002212
6914 062522 010001
6915 062524

BGNSUB ;>>>>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>>>>>
T6.7:

TRAP C\$BSUB
JSR PC,T26REST ;SET COMMAND PACKET
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 661
.WORD SFIERR
.WORD SFIMSG
MOV UNITN,T26DSW ;SET UP UNIT NUMBER
MOV #T26PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS

:WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)

JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
BCS 26\$;BR, IF COMMAND ISSUED OK
INC FATFLG ;BUMP COUNT
MOV R0,R1 ;SAVE CONTENTS OF TSSR
ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED


```
062524 104456 TRAP C$ERHRD
062526 001226 .WORD 662
062530 005054 .WORD WRTMSG
062532 012124 .WORD SFIMSG
6916 062534 26$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
062534 104406
6917
6918
6919
6920 :*****
6921 :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6922 :*****
6923
6924 062536 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6925 062542 103413 BCS 30$ ;BR, IF NO PROBLEM
6926 062544 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
6927 062550 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
6928 062554 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
6929 062556 005237 002212 INC FATFLG ;BUMP COUNT
6933 062562 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
062562 104456 TRAP C$ERHRD
062564 001227 .WORD 663
062566 073254 .WORD T26RWN
062570 012136 .WORD PKTSSR
6934 062572 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
062572 104406
6935
6936 :*****
6937 :
6938 :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6939 :*****
6940
6941
6942 062574 013701 071630 MOV T26BFR+6,R1 ;PICK UP XSTO
6943 062600 010102 MOV R1,R2 ;SET UP EXPECTED
6944 062602 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6945 062606 020102 CMP R1,R2 ;DOES EXP = REC'D
6946 062610 001406 BEQ 40$ ;BR, IF EQUAL (OK)
6947 062612 005237 002212 INC FATFLG ;BUMP COUNT
6951 062616 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
062616 104456 TRAP C$ERHRD
062620 001230 .WORD 664
062622 072765 .WORD T26BOT
062624 015564 .WORD EXPREC
6952 062626 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
062626 104406
6953 062630 012703 000400 MOV #256.,R3 ;RECORD SIZE
6954 062634 013737 003114 071722 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
6955
6956 :*****
6957 :
6958 :WRITE DATA,CVC=1,ACK COMMAND
6959 :*****
6960
6961
6962 062642 012737 140005 071720 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
6963 062650 012704 071720 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
```

```

6964 062654
6965 062654 010300
6966 062656 004737 017512
6967 062662 010337 071726
6968 062666 010465 000000
6969 062672 004737 016340
6970 062676 016501 000002
6971 062702 012702 000200
6972 062706 020102
6973 062710 001406
6974 062712 005237 002212
6978 062716
      062716 104456
      062720 001231
      062722 005111
      062724 C12136
6979 062726
      062726 174406
6980 062730 065723
6981 062732 022703 000414
6982 062736 001346
6983 062740
      062740 104406
6984 062742
6985
6986
6987
6988
6989
6990
6991
6992 062742 004737 011104
6993 062746 103413
6994 062750 016501 000002
6995 062754 012702 000200
6996 062760 010004
6997 062762 005237 002212
7001 062766
      062766 104456
      062770 001232
      062772 073254
      062774 012136
7002 062776
      062776 104406
7003
7004
7005
7006
7007
7008
7009
7010 063000 013701 071630
7011 063004 010102
7012 063006 052702 000002
7013 063012 020102
7014 063014 001406
7015 063016 005237 002212

65$:
      MOV      R3,R0          ;SET PATTERN IN CORRECT REGISTER
      JSR      PC,FILLMEM     ;FILL MEMORY WITH RECORD SIZE
      MOV      R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
      MOV      #SSR,R2        ;SET UP EXPECTED
      CMP      R1,R2          ;ARE THEY EQUAL
      BEQ      75$            ;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$CLP1
:*****
:READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
:*****
      MOV      T26BFR+6,R1    ;PICK UP XSTO
      MOV      R1,R2          ;SET UP EXPECTED
      BIS      #BIT1,R2       ;SET BOT BIT IN EXPECTED
      CMP      R1,R2          ;DOES EXP = REC'D
      BEQ      140$          ;BR, IF EQUAL (OK)
      INC      FATFLG         ;BUMP COUNT

```

```
7019 063022          ERRHRD  ERRNO,T26BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      063022 104456          TRAP          C$ERHRD
      063024 001233          .WORD          667
      063026 072765          .WORD          T26BOT
      063030 015564          .WORD          EXPREC
7020 063032          140$:  CKLOOP          ;LOOP IF SELECTED
      063032 104406          TRAP          C$CLP1
7021 063034 012737 000400 071752      MOV      #256.,T26RSZ      ;STORE START RECORD SIZE
7022 063042 000420          BR        150$            ;SKIP THE SAPCE THIS TIME
7023
7024          :*****
7025          :
7026          :ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
7027          :BIT 15 SETS DIRECTION - 0=FORWARD  1=REVERSE
7028          :
7029          :*****
7030
7031 063044 012703 000001      145$:  MOV      #1,R3          ;SPACE ONE RECORD PARAMETER
7032 063050 004737 010556      JSR      PC,SPACE        ;CALL SPACE ROUTINE
7033 063054 103413          BCS      150$            ;BR, IF NO PROBLEM WITH SPACE COMMAND
7034 063056 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR
7035 063062 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED TSSR
7036 063066 010004          MOV      R0,R4           ;PACKET ADDRESS SET UP
7037 063070 005237 002212      INC      FATFLG          ;BUMP COUNT
7041 063074          ERRHRD  ERRNO,T26SC,EXPREC      ;POSITION (SPACE RECORDS) FAILED
      063074 104456          TRAP          C$ERHRD
      063076 001234          .WORD          668
      063100 072367          .WORD          T26SC
      063102 015564          .WORD          EXPREC
7042 063104          150$:  CKLOOP          ;LOOP IF SELECTED
      063104 104406          TRAP          C$CLP1
7043 063106 013703 071752      MOV      T26RSZ,R3       ;RECORD SIZE
7044 063112 013737 003114 071722      MOV      FREE,T26RB      ;STARTING READ BUFFER ADDRESS
7045
7046          :*****
7047          :
7048          :REREREAD DATA,CVC=1,ACK COMMAND
7049          :
7050          :*****
7051
7052 063120 012737 141401 071720      165$:  MOV      #141401,T26PK3 ;REREREAD DATA,CVC=1,ACK COMMAND
7053 063126 012704 071720      MOV      #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
7054 063132 010337 071726      MOV      R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
7055 063136 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
7056 063142 004737 016340      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
7057 063146 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
7058 063152 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
7059 063156 020102          CMP      R1,R2           ;ARE THEY EQUAL
7060 063160 001406          BEQ      170$            ;BR, IF OK
7061 063162 005237 002212      INC      FATFLG          ;BUMP COUNT
7065 063166          ERRHRD  ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      063166 104456          TRAP          C$ERHRD
      063170 001235          .WORD          669
      063172 073610          .WORD          T26WDC
      063174 012136          .WORD          PKTSSR
7066 063176          170$:  CKLOOP          ;LOOP IF SELECTED
      063176 104406          TRAP          C$CLP1
```



```
7148  
7149  
7150  
7151  
7152  
7153  
7154 063410 004737 011104  
7155 063414 103413  
7156 063416 016501 000002  
7157 063422 012702 000200  
7158 063426 010004  
7159 063430 005237 002212  
7163 063434  
063434 104456  
063436 001241  
063440 073254  
063442 012136  
7164 063444 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1  
063444 104406  
7165  
7166  
7167  
7168  
7169  
7170  
7171  
7172 063446 013701 071630  
7173 063452 010102  
7174 063454 052702 000002  
7175 063460 020102  
7176 063462 001406  
7177 063464 005237 002212  
7181 063470  
063470 104456  
063472 001242  
063474 072765  
063476 015564  
7182 063500 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1  
063500 104406  
7183 063502 012703 000400  
7184 063506 013737 003114 071722  
7185  
7186  
7187  
7188  
7189  
7190  
7191  
7192 063514 012737 150005 071720  
7193 063522 012704 071720  
7194 063526  
7195 063526 010300  
7196 063530 004737 017512  
7197 063534 010337 071726  
7198 063540 010465 000000  
7199 063544 004737 016340  
7200 063550 016501 000002
```

: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 673  
.WORD T26RWN  
.WORD PKTSSR
```

: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 674  
.WORD T26BOT  
.WORD EXPREC
```

:WRITE DATA,CVC=1,ACK,SWB COMMAND

```
MOV #150005,T26PK3 ;WRITE DATA,CVC=1,ACK,SWB COMMAND  
MOV #T26PK3,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,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
```

```

7201 063554 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
7202 063560 020102      CMP      R1,R2      ;ARE THEY EQUAL
7203 063562 001406      BEQ      75$      ;BR, IF OK
7204 063564 005237 002212      INC      FATFLG      ;BUMP COUNT
7208 063570      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      063570 104456      TRAP      C$ERHRD
      063572 001243      .WORD     675
      063574 005111      .WORD     WRTErr
      063576 012136      .WORD     PKTSSR
7209 063600      75$:  CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
      063600 104406
7210 063602 005723      TST      (R3)+      ;BUMP RECORD SIZE
7211 063604 022703 000414      CMP      #268.,R3   ;END OF RECORD YET
7212 063610 001346      BNE      65$      ;BR, IF MORE RECORDS TO WRITE
7213 063612      80$:  CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
      063612 104406
7214 063614      120$:
7215
7216      ;*****
7217      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7218      ;*****
7219
7220
7221
7222 063614 004737 011104      JSR      PC,REWIND   ;CALL TAPE REWIND COMMAND
7223 063620 103413      BCS      130$      ;BR, IF NO PROBLEM
7224 063622 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR
7225 063626 012702 000200      MOV      #SSR,R2   ;SET UP EXPECTED TSSR
7226 063632 010004      MOV      R0,R4     ;PACKET ADDRESS SET UP
7227 063634 005237 002212      INC      FATFLG    ;BUMP COUNT
7231 063640      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      063640 104456      TRAP      C$ERHRD
      063642 001244      .WORD     676
      063644 073254      .WORD     T26RWN
      063646 012136      .WORD     PKTSSR
7232 063650      130$: CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
      063650 104406
7233
7234      ;*****
7235      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
7236      ;*****
7237
7238
7239
7240 063652 013701 071630      MOV      T26BFR+6,R1 ;PICK UP XST0
7241 063656 010102      MOV      R1,R2     ;SET UP EXPECTED
7242 063660 052702 000002      BIS      #BIT1,R2   ;SET BOT BIT IN EXPECTED
7243 063664 020102      CMP      R1,R2     ;DOES EXP = REC'D
7244 063666 001406      BEQ      140$      ;BR, IF EQUAL (OK)
7245 063670 005237 002212      INC      FATFLG    ;BUMP COUNT
7249 063674      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      063674 104456      TRAP      C$ERHRD
      063676 001245      .WORD     677
      063700 072765      .WORD     T26BOT
      063702 015564      .WORD     EXPREC
7250 063704      140$: CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
      063704 104406

```

```
7251 063706 012737 000400 071752      MOV      #256.,T26RSZ      ;START RECORD SIZE
7252 063714 000420                BR      150$              ;SKIP SACE THIS TIME
7253
7254
7255      :*****
7256      :ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
7257      :BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
7258      :*****
7259
7260
7261 063716 012703 000001      145$:  MOV      #1,R3              ;SPACE ONE RECORD PARAMETER
7262 063722 004737 010556      JSR      PC,SPACE          ;CALL SPACE ROUTINE
7263 063726 103413                BCS     150$              ;BR, IF NO PROBLEM WITH SPACE COMMAND
7264 063730 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR
7265 063734 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED TSSR
7266 063740 010004                MOV     RO,R4              ;PACKET ADDRESS SET UP
7267 063742 005237 002212      INC     FATFLG            ;BUMP COUNT
7271 063746                ERRHRD  ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
7271 063746 104456                TRAP   C$ERHRD
7271 063750 001246                .WORD  678
7271 063752 072367                .WORD  T26SC
7271 063754 015564                .WORD  EXPREC
7272 063756                150$:  CKLOOP
7272 063756 104406                TRAP   C$CLP1
7273 063760 013703 071752      MOV      T26RSZ,R3        ;RECORD SIZE
7274 063764 013737 003114 071722      MOV      FREE,T26RB       ;STARTING READ BUFFER ADDRESS
7275
7276      :*****
7277      :REREAD DATA,ACK,CVC=1,SWB COMMAND
7278      :*****
7279
7280
7281
7282 063772 012737 151401 071720      165$:  MOV      #151401,T26PK3    ;REREAD DATA,ACK,CVC=1,SWB COMMAND
7283 064000 012704 071720      MOV      #T26PK3,R4       ;SET UP R4 WITH PACKET ADDRESS
7284 064004 010337 071726      MOV      R3,T26SZ         ;SET UP RECORD SIZE IN PACKET
7285 064010 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
7286 064014 004737 016340      JSR      PC,WAITF         ;WAIT FOR SSR TO SET
7287 064020 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
7288 064024 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
7289 064030 020102                CMP     R1,R2              ;ARE THEY EQUAL
7290 064032 001406                BEQ    170$              ;BR, IF OK
7291 064034 005237 002212      INC     FATFLG            ;BUMP COUNT
7295 064040                ERRHRD  ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
7295 064040 104456                TRAP   C$ERHRD
7295 064042 001247                .WORD  679
7295 064044 073610                .WORD  T26WDC
7295 064046 012136                .WORD  PKTSSR
7296 064050                170$:  CKLOOP
7296 064050 104406                ;LOOP IF SELECTED
7296 064050 104406                TRAP   C$CLP1
7297 064052 013702 003114      MOV      FREE,R2          ;CURRENT BUFFER ADDRESS TO R2
7298 064056 010304                MOV     R3,R4              ;CURRENT RECORD SIZE
7299 064060 162704 000400      SUB     #256.,R4          ;FIRST LOCATION IN BUFFER
7300 064064 060204      173$:  ADD     R2,R4              ;SET UP POINTER
7301 064066 021403                CMP     (R4),R3           ;CHECK DATA READ (R3=DATA ALSO)
7302 064070 001410                BEQ    180$              ;BR, IF ALL IS WELL
7303 064072 011401                MOV     (R4),R1           ;RECD DATA
```


7326
7327
7328
7329
7330
7331
7332
7333
7334
7335
7336
7337
7338
7339
7340
7341
7342
7343
7344
7345
7346
7347
7348
7349
7350
7351
7352
7353
7354
7355
7356
7357
7358

```

: *
: TEST 6, SUBTEST 9
: VERIFIES THAT THE REREAD NEXT COMMAND WITH OPP=1
: (READ FORWARD, SPACE REVERSE) AND SWB=0 OPERATES
: PROPERLY. THE TAPE IS FIRST REWOUND AND THEN WRITTEN
: WITH A SERIES OF TEST RECORDS VARYING IN LENGTH AND
: DATA CONTENT; THE FIRST FOUR BYTES OF EACH RECORD
: CONTAIN ITS RECORD NUMBER (INDICATING POSITION ON
: TAPE). THE TAPE IS THEN REWOUND AGAIN. FOR EACH
: TEST RECORD, THE FOLLOWING SEQUENCE IS EXECUTED.
: 1. THE REREAD NEXT COMMAND WITH OPP=1 IS ISSUED
: AND THE RESULTS CHECKED
: 2. A READ FORWARD COMMAND IS THEN ISSUED AND THE
: DATA IS CHECKED TO VERIFY THAT THE TAPE WAS
: POSITIONED PROPERLY AFTER THE REREAD NEXT
: COMMAND (E.G. THE TAPE SHOULD HAVE BEEN LEFT
: POSITIONED AT THE START OF THE TEST RECORD). THE
: READ FORWARD COMMAND LEAVES THE TAPE POSITIONED
: PROPERLY AT THE START OF THE NEXT TEST RECORD.
: THE BYTE COUNT ON EACH REREAD NEXT COMMAND IS SET
: TO THE LENGTH OF THE EXPECTED RECORD, SO NO
: EXCEPTIONAL CONDITIONS SHOULD OCCUR.
    
```

```

7359 064160      BGNSUB                  ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
       064160      T6.9:
       064160      104402                  TRAP          CSBSUB
7360 064162      004737 074510            JSR          PC,T26REST          ;SET COMMAND PACKET
7361 064166      005037 071746            CLR          T26CNT             ;CLEAR TAPE RECORD COUNTER
7362 064172      004737 074602            JSR          PC,T26RT2          ;SET UP OTHER COMMAND PACKET
7363 064176      004737 074644            JSR          PC,T26RT3          ;SET UP OTHER COMMAND PACKET
7364
7365 :*****
7366 :ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
7367 :*****
7368
7369
7370
7371 064202      004737 016064            JSR          PC,SOFINIT         ;DO INITIALIZE ON CONTROLLER
7372 064206      103407                  BCS          20$                ;BR IF INIT WAS OK
7373 064210      005237 002212            INC          FATFLG             ;BUMP COUNT
7377 064214      010001                  MOV          R0,R1              ;CONTENTS OF TSSR REGISTER
7378 064216      104455                  ERRDF       ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
       064216      104455                  TRAP          CSERDF
       064220      001251                  .WORD       681
       064222      003650                  .WORD       SFIERR
       064224      012124                  .WORD       SFIMSG
7379 064226      013737 002172 071620 20$: MOV          UNITN,T26DSW      ;SET UP UNIT NUMBER
    
```

```
7380
7381 064234 012704 071600          MOV      #T26PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
7382
7383          :*****
7384          :WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
7385          :*****
7386
7387
7388
7389 064240 004737 010752          JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
7390 064244 103407          BCS     26$                ;BR, IF COMMAND ISSUED OK
7391 064246 005237 002212          INC     FATFLG            ;BUMP COUNT
7395 064252 010001          MOV     R0,R1             ;SAVE CONTENTS OF TSSR
7396 064254          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
                                TRAP      C$ERHRD
                                .WORD    682
                                .WORD    WRTMSG
                                .WORD    SFIMSG
                                TRAP      C$CLP1
                                064254 104456
                                064256 001252
                                064260 005054
                                064262 012124
7397 064264          26$: CKLOOP          ;LOOP IF SELECTED
                                064264 104406
                                TRAP      C$CLP1
7398
7399          :*****
7400          :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7401          :*****
7402
7403
7404
7405 064266 004737 011104          JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
7406 064272 103413          BCS     30$                ;BR, IF NO PROBLEM
7407 064274 016501 000002          MOV     TSSR(R5),R1       ;GET TSSR
7408 064300 012702 000200          MOV     #SSR,R2           ;SET UP EXPECTED TSSR
7409 064304 010004          MOV     R0,R4             ;PACKET ADDRESS SET UP
7410 064306 005237 002212          INC     FATFLG            ;BUMP COUNT
7414 064312          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    683
                                .WORD    T26RWN
                                .WORD    PKTSSR
                                TRAP      C$CLP1
                                064312 104456
                                064314 001253
                                064316 073254
                                064320 012136
7415 064322          30$: CKLOOP          ;LOOP IF SELECTED
                                064322 104406
                                TRAP      C$CLP1
7416
7417          :*****
7418          :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7419          :*****
7420
7421
7422
7423 064324 013701 071630          MOV     T26BFR+6,R1       ;PICK UP XSTO
7424 064330 010102          MOV     R1,R2             ;SET UP EXPECTED
7425 064332 052702 000002          BIS     #BIT1,R2          ;SET BOT BIT IN EXPECTED
7426 064336 020102          CMP     R1,R2             ;DOES EXP = REC'D
7427 064340 001406          BEQ    40$                ;BR, IF EQUAL (OK)
7428 064342 005237 002212          INC     FATFLG            ;BUMP COUNT
7432 064346          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    684
                                .WORD    T26BOT
                                064346 104456
                                064350 001254
                                064352 072765
```

```
7433 064354 015564          40$:  CKLOOP          ;LOOP IF SELECTED          .WORD  EXPREC
      064356          ;RECORD SIZE          TRAP  C$CLP1
      064356 104406
7434 064360 012703 000400      MOV  #256.,R3          ;RECORD SIZE
7435 064364 013737 003114 071722  MOV  FREE,T26RB       ;STARTING WRITE BUFFER ADDRESS
7436
7437
7438
7439
7440
7441
7442
7443 064372 012737 140005 071720      MOV  #140005,T26PK3   ;WRITE DATA,CVC=1,ACK COMMAND
7444 064400 012704 071720      MOV  #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
7445 064404          65$:
7446 064404 010337 071726      MOV  R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
7447 064410 013777 071746 116476      MOV  T26CNT,@FREE    ;MOVE TAPE RECORD NUMBER TO BUFFER
7448 064416 062737 000001 071746      ADD  #1,T26CNT       ;NUMBER READY FOR NEXT RECORD
7449 064424 010465 000000      MOV  R4,T26DB(R5)    ;ISSUE COMMAND
7450 064430 004737 016340      JSR  PC,WAITF        ;WAIT FOR SSR TO SET
7451 064434 016501 000002      MOV  TSSR(R5),R1     ;GET TSSR CONTENTS
7452 064440 012702 000200      MOV  #SSR,R2        ;SET UP EXPECTED
7453 064444 020102          CMP  R1,R2          ;ARE THEY EQUAL
7454 064446 001406          BEQ  75$            ;BR, IF OK
7455 064450 005237 002212      INC  FATFLG         ;BUMP COUNT
7459 064454          ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      064454 104456          TRAP  C$ERHRD
      064456 001255          .WORD  685
      064460 005111          .WORD  WRERR
      064462 012136          .WORD  PKTSSR
7460 064464          75$:  CKLOOP          ;LOOP IF SELECTED          .WORD  EXPREC
      064464 104406          TRAP  C$CLP1
7461 064466 005723
7462 064470 022703 000414      TST  (R3)+          ;BUMP THE RECORD SIZE
7463 064474 001401          CMP  #268.,R3       ;MAXIMUM SIZE YET
7464 064476 000742          BEQ  120$           ;BR, IF AT END OF WRITE SEQUENCE
7465 064500          BR   65$            ;WRITE MORE RECORDS
7466 064500 005037 071746      120$: CLR  T26CNT         ;SET RECORD COUNTER BACK TO ZERO
7467
7468
7469
7470
7471
7472
7473
7474 064504 004737 011104      JSR  PC,REWIND       ;CALL TAPE REWIND COMMAND
7475 064510 103411          BCS  130$           ;BR, IF NO PROBLEM
7476 064512 016501 000002      MOV  TSSR(R5),R1     ;GET TSSR
7477 064516 010004          MOV  R0,R4          ;PACKET ADDRESS SET UP
7478 064520 005237 002212      INC  FATFLG         ;BUMP COUNT
7482 064524          ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      064524 104456          TRAP  C$ERHRD
      064526 001256          .WORD  686
      064530 073254          .WORD  T26RWN
      064532 012136          .WORD  PKTSSR
7483 064534          130$: CKLOOP          ;LOOP IF SELECTED          .WORD  EXPREC
      064534 104406          TRAP  C$CLP1
```

```
7484
7485
7486
7487
7488
7489
7490
7491 064536 013701 071630      MOV      T26BFR+6,R1      ;PICK UP XSTO
7492 064542 010102      MOV      R1,R2           ;SET UP EXPECTED
7493 064544 052702 000002      BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
7494 064550 020102      CMP      R1,R2           ;DOES EXP = REC'D
7495 064552 001406      BEQ      135$            ;BR, IF EQUAL (OK)
7496 064554 005237 002212      INC      FATFLG          ;BUMP COUNT
7500 064560      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      064560 104456      TRAP      C$ERHRD
      064562 001257      .WORD     687
      064564 072765      .WORD     T26BOT
      064566 015564      .WORD     EXPREC
7501 064570 104406      135$:   CKLOOP          ;LOOP IF SELECTED      TRAP      C$CLP1
      064570 012737 000400 071752      MOV      #256.,T26RSZ    ;STARTING RECORD SIZE
7502 064572 012737 000400 071752      BR       140$           ;SKIP OVER THE SPACE THIS TIME
7503 064600 000420
7504
7505
7506
7507
7508
7509
7510
7511
7512 064602 012703 000001      132$:   MOV      #000001,R3 ;SET UP SPACE COMMAND (1 FORWARD)
7513 064606 004737 010556      JSR      PC,SPACE        ;CALL SPACE ROUTINE
7514 064612 103413      BCS      140$           ;BR, IF NO TROUBLE
7515 064614 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR
7516 064620 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED TSSR
7517 064624 010004      MOV      R0,R4          ;PACKET ADDRESS SET UP
7518 064626 005237 002212      INC      FATFLG          ;BUMP COUNT
7522 064632      ERRHRD  ERRNO,T26SC,PKTSSR ;SPACE (FORWARD) FAILED
      064632 104456      TRAP      C$ERHRD
      064634 001260      .WORD     688
      064636 072367      .WORD     T26SC
      064640 012136      .WORD     PKTSSR
7523 064642      140$:   CKLOOP          ;LOOP IF SELECTED      TRAP      C$CLP1
      064642 104406
7524 064644 013703 071752      MOV      T26RSZ,R3      ;RECORD SIZE
7525 064650 013737 003114 071722      150$:   MOV      FREE,T26RB ;STARTING READ BUFFER ADDRESS
7526
7527
7528
7529
7530
7531
7532
7533 064656 012737 161401 071720      165$:   MOV      #161401,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
7534 064664 012704 071720      MOV      #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
7535 064670 010337 071726      MOV      R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
7536 064674 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
```



```

7624 065140 104406 26$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      065140
7625
7626 :*****
7627 :
7628 :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7629 :
7630 :*****
7631
7632 065142 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
7633 065146 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
7634 065152 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
7635 065156 103407 BCS 30$ ;BR, IF NO PROBLEM
7636 065160 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
7637 065162 005237 002212 INC FATFLG ;BUMP COUNT
7641 065166 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      065166 104456 TRAP C$ERHRD
      065170 001265 .WORD 693
      065172 073254 .WORD T26RWN
      065174 012136 .WORD PKTSSR
7642 065176 104406 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      065176
7643
7644 :*****
7645 :
7646 :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7647 :
7648 :*****
7649
7650 065200 013701 071630 MOV T26BFR+6,R1 ;PICK UP XSTO
7651 065204 010102 MOV R1,R2 ;SET UP EXPECTED
7652 065206 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
7653 065212 020102 CMP R1,R2 ;DOES EXP = REC'D
7654 065214 001406 BEQ 40$ ;BR, IF EQUAL (OK)
7655 065216 005237 002212 INC FATFLG ;BUMP COUNT
7659 065222 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      065222 104456 TRAP C$ERHRD
      065224 001266 .WORD 694
      065226 072765 .WORD T26BOT
      065230 015564 .WORD EXPREC
7660 065232 104406 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      065232
7661 065234 012703 000400 MOV #256.,R3 ;RECORD SIZE
7662 065240 013737 003114 071722 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
7663
7664 :*****
7665 :
7666 :WRITE DATA,CVC=1,ACK COMMAND
7667 :
7668 :*****
7669
7670 065246 012737 140005 071720 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
7671 065254 012704 071720 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
7672 065260
7673 065260 010337 071726 65$: MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
7674 065264 013777 071746 115622 MOV T26CNT,@FREE ;MOVE TAPE RECORD NUMBER TO BUFFER
7675 065272 062737 000001 071746 ADD #1,T26CNT ;NUMBER READY FOR NEXT RECORD
  
```



```

7676 065300 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
7677 065304 004737 016340      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
7678 065310 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
7679 065314 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
7680 065320 020102                CMP      R1,R2           ;ARE THEY EQUAL
7681 065322 001406                BEQ      75$             ;BR, IF OK
7682 065324 005237 002212      INC      FATFLG          ;BUMP COUNT
7686 065330                ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP  C$ERHRD
                                .WORD 695
                                .WORD WRTErr
                                .WORD PKTSSR
                                TRAP  C$CLP1
                                .WORD 696
                                .WORD T26RWN
                                .WORD PKTSSR
                                TRAP  C$CLP1
                                .WORD 697
                                .WORD T26BOT
                                .WORD EXPREC
7687 065340                75$:  CKLOOP            ;LOOP IF SELECTED
                                TRAP  C$CLP1
                                .WORD 695
                                .WORD T26RWN
                                .WORD PKTSSR
7688 065342 005723                TST      (R3)+           ;BUMP THE RECORD SIZE
7689 065344 022703 000414      CMP      #268.,R3       ;MAXIMUM SIZE YET
7690 065350 001401                BEQ      120$            ;BR, IF AT END OF WRITE SEQUENCE
7691 065352 000742                BR       65$             ;WRITE MORE RECORDS
7692 065354                120$:
7693 065354 005037 071746      CLR      T26CNT          ;SET RECORD COUNTER BACK TO ZERO
7694
7695
7696
7697
7698
7699
7700
7701 065360 004737 011104      JSR      PC,REWIND       ;CALL TAPE REWIND COMMAND
7702 065364 103411                BCS      130$            ;BR, IF NO PROBLEM
7703 065366 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR
7704 065372 010004                MOV      R0,R4           ;PACKET ADDRESS SET UP
7705 065374 005237 002212      INC      FATFLG          ;BUMP COUNT
7709 065400                ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP  C$ERHRD
                                .WORD 696
                                .WORD T26RWN
                                .WORD PKTSSR
7710 065410                130$:  CKLOOP            ;LOOP IF SELECTED
                                TRAP  C$CLP1
                                .WORD 697
                                .WORD T26BOT
                                .WORD EXPREC
7711
7712
7713
7714
7715
7716
7717
7718 065412 013701 071630      MOV      T26BFR+6,R1    ;PICK UP XSTO
7719 065416 010102                MOV      R1,R2           ;SET UP EXPECTED
7720 065420 052702 000002      BIS      #BIT1,R2       ;SET BOT BIT IN EXPECTED
7721 065424 020102                CMP      R1,R2           ;DOES EXP = REC'D
7722 065426 001406                BEQ      135$            ;BR, IF EQUAL (OK)
7723 065430 005237 002212      INC      FATFLG          ;BUMP COUNT
7727 065434                ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP  C$ERHRD
                                .WORD 697
                                .WORD T26BOT
                                .WORD EXPREC
7728 065434 104456
7729 065436 001271
7730 065440 072765
7731 065442 015564

```

```

7 28 065444          135$:  CKLOOP                                ;LOOP IF SELECTED
      065444 104406                                TRAP  C$CLP1
7729 065446 012737 000400 071752  MOV  #256.,T26RSZ      ;START RECORD SIZE
7730 065454 000420  BR  140$                          ;SKIP OVER SPACE
7731
7732
7733
7734
7735
7736
7737
7738
7739 065456 012703 000001 136$:  MOV  #000001,R3      ;SET UP SPACE COMMAND (1 FORWARD)
7740 065462 004737 010556  JSR  PC,SPACE      ;CALL SPACE ROUTINE
7741 065466 103413  BCS  140$          ;BR, IF NO TROUBLE
7742 065470 016501 000002  MOV  TSSR(R5),R1    ;GET TSSR
7743 065474 012702 000200  MOV  #SSR,R2       ;SET UP EXPECTED TSSR
7744 065500 010004  MOV  R0,R4         ;PACKET ADDRESS SET UP
7745 065502 005237 002212  INC  FATFLG        ;BUMP COUNT
7749 065506  ERRHRD ERRNO,T26SC,PKTSSR ;SPACE (FORWARD) FAILED
      065506 104456                                TRAP  C$ERHRD
      065510 001272                                .WORD 698
      065512 072367                                .WORD T26SC
      065514 012136                                .WORD PKTSSR
7750 065516          140$:  CKLOOP                                ;LOOP IF SELECTED
      065516 104406                                TRAP  C$CLP1
7751 065520 013703 071752  MOV  T26RSZ,R3      ;RECORD SIZE
7752 065524 013737 003114 071722 150$:  MOV  FREE,T26RB      ;STARTING READ BUFFER ADDRESS
7753
7754
7755
7756
7757
7758
7759
7760 065532 012737 161401 071720  MOV  #161401,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
7761 065540 012704 071720 165$:  MOV  #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
7762 065544 010337 071726  MOV  R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
7763 065550 010465 000000  MOV  R4,TSDB(R5)   ;ISSUE COMMAND
7764 065554 004737 016340  JSR  PC,WAITF      ;WAIT FOR SSR TO SET
7765 065560 016501 000002  MOV  TSSR(R5),R1   ;GET TSSR CONTENTS
7766 065564 012702 000200  MOV  #SSR,R2       ;SET UP EXPECTED
7767 065570 020102  CMP  R1,R2         ;ARE THEY EQUAL
7768 065572 001406  BEQ  170$          ;BR, IF OK
7769 065574 005237 002212  INC  FATFLG        ;BUMP COUNT
7773 065600  ERRHRD ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      065600 104456                                TRAP  C$ERHRD
      065602 001273                                .WORD 699
      065604 072175                                .WORD T26RRF
      065606 012136                                .WORD PKTSSR
7774 065610          170$:  CKLOOP                                ;LOOP IF SELECTED
      065610 104406                                TRAP  C$CLP1
7775 065612 017701 115276  MOV  @FREE,R1      ;FIRST WORD FROM READ BUFFER
7776 065616 013702 071746  MOV  T26CNT,R2     ;SET UP EXPECTED
7777 065622 020102  CMP  R1,R2         ;IS TAPE POSITION CORRECT
7778 065624 001406  BEQ  190$          ;KEEP GOING POSITION OK
7779 065626 005237 002212  INC  FATFLG        ;BUMP COUNT

```



```
066010 104406 TRAP C$CLP1
7851
7852
7853
7854
7855
7856
7857
7858 066012 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
7859 066016 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
7860 066022 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
7861 066026 103407 BCS 30$ ;BR, IF NO PROBLEM
7862 066030 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
7863 066032 005237 002212 INC FATFLG ;BUMP COUNT
7867 066036 ERRHRD ERRNO,T26RWN,PXTSSR ;REWIND NOT ACCEPTED
066036 104456 TRAP C$ERHRD
066040 001277 .WORD 703
066042 073254 .WORD T26RWN
066044 012136 .WORD PKTSSR
7868 066046 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
066046 104406
7869
7870
7871
7872
7873
7874
7875
7876 066050 013701 071630 MOV T26BFR+6,R1 ;PICK UP XSTO
7877 066054 010102 MOV R1,R2 ;SET UP EXPECTED
7878 066056 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
7879 066062 020102 CMP R1,R2 ;DOES EXP = REC'D
7880 066064 001406 BEQ 40$ ;BR, IF EQUAL (OK)
7881 066066 005237 002212 INC FATFLG ;BUMP COUNT
7885 066072 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
066072 104456 TRAP C$ERHRD
066074 001300 .WORD 704
066076 072765 .WORD T26BOT
066100 015564 .WORD EXPREC
7886 066102 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
066102 104406
7887 066104 012703 001000 MOV #512.,R3 ;RECORD SIZE
7888 066110 013737 003114 071722 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
7889
7890
7891
7892
7893
7894
7895
7896 066116 012737 140005 071720 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
7897 066124 012704 071720 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
7898 066130 65$:
7899 066130 010337 071726 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
7900 066134 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
7901 066140 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
7902 066144 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
```

```
7903 066150 012702 000200      MOV    #SSR,R2      ;SET UP EXPECTED
7904 066154 020102      CMP    R1,R2       ;ARE THEY EQUAL
7905 066156 001406      BEQ    75$         ;BR, IF OK
7906 066160 005237 002212      INC    FATFLG      ;BUMP COUNT
7910 066164      ERRHRD ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      066164 104456      TRAP   C$ERHRD
      066166 001301      .WORD 705
      066170 005111      .WORD WRTErr
      066172 012136      .WORD PKTSSR
7911 066174      75$:   CKLOOP      ;LOOP IF SELECTED
      066174 104406      TRAP   C$CLP1
7912
7913
7914
7915
7916
7917
7918
7919 066176 004737 011104      JSR    PC,REWIND   ;CALL TAPE REWIND COMMAND
7920 066202 016501 000002      MOV    TSSR(R5),R1 ;GET TSSR
7921 066206 012702 000200      MOV    #SSR,R2    ;SET UP EXPECTED TSSR
7922 066212 103407      BCS    130$       ;BR, IF NO PROBLEM
7923 066214 010004      MOV    R0,R4      ;PACKET ADDRESS SET UP
7924 066216 005237 002212      INC    FATFLG      ;BUMP COUNT
7928 066222      ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      066222 104456      TRAP   C$ERHRD
      066224 001302      .WORD 706
      066226 073254      .WORD T26RWN
      066230 012136      .WORD PKTSSR
7929 066232      130$: CKLOOP      ;LOOP IF SELECTED
      066232 104406      TRAP   C$CLP1
7930
7931
7932
7933
7934
7935
7936
7937 066234 013701 071630      MOV    T26BFR+6,R1 ;PICK UP XST0
7938 066240 010102      MOV    R1,R2      ;SET UP EXPECTED
7939 066242 052702 000002      BIS    #BIT1,R2   ;SET BOT BIT IN EXPECTED
7940 066246 020102      CMP    R1,R2      ;DOES EXP = REC'D
7941 066250 001406      BEQ    140$       ;BR, IF EQUAL (OK)
7942 066252 005237 002212      INC    FATFLG      ;BUMP COUNT
7946 066256      ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      066256 104456      TRAP   C$ERHRD
      066260 001303      .WORD 707
      066262 072765      .WORD T26BOT
      066264 015564      .WORD EXPREC
7947 066266      140$: CKLOOP      ;LOOP IF SELECTED
      066266 104406      TRAP   C$CLP1
7948 066270 005303      DEC    R3          ;SET RECORD SIZE TO 511.
7949 066272 013737 003114 071722      MOV    FREE,T26RB ;STARTING READ BUFFER ADDRESS
7950
7951
7952
7953
:*****
:REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
```

```
7954  
7955  
7956  
7957 066300 012737 161401 071720  
7958 066306 012704 071720  
7959 066312 010337 071726  
7960 066316 010465 000000  
7961 066322 004737 016340  
7962 066326 016501 000002  
7963 066332 012702 100204  
7964 066336 020102  
7965 066340 001406  
7966 066342 005237 002212  
7970 066346  
066346 104456  
066350 001304  
066352 074332  
066354 012136  
7971 066356  
066356 104406  
7972  
7973  
7974  
7975  
7976  
7977  
7978  
7979 066360 013701 071630  
7980 066364 010102  
7981 066366 052702 010000  
7982 066372 020102  
7983 066374 001406  
7984 066376 005237 002212  
7988 066402  
066402 104456  
066404 001305  
066406 074100  
066410 015564  
7989 066412  
066412 104406  
7990 066414 012703 000777  
7991 066420 013737 003114 071722  
7992  
7993  
7994  
7995  
7996  
7997  
7998  
7999 066426 012737 141401 071720  
8000 066434 012704 071720  
8001 066440 010337 071726  
8002 066444 010465 000000  
8003 066450 004737 016340  
8004 066454 016501 000002  
8005 066460 012702 100204  
8006 066464 020102  
:*****  
165$: MOV #161401,T26PK3 ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND  
MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS  
MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET  
MOV R4,TSDB(R5) ;ISSUE COMMAND  
JSR PC,WAITF ;WAIT FOR SSR TO SET  
MOV TSSR(R5),R1 ;GET TSSR CONTENTS  
MOV #SSR!SC!BIT2,R2 ;SET UP EXPECTED  
CMP R1,R2 ;ARE THEY EQUAL  
BEQ 170$ ;BR, IF OK  
INC FATFLG ;BUMP COUNT  
ERRHRD ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA  
TRAP C$ERHRD  
.WORD 708  
.WORD T26TRL  
.WORD PKTSSR  
170$: CKLOOP ;LOOP IF SELECTED  
TRAP C$CLP1  
:*****  
:READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)  
:*****  
7979 066360 013701 071630 MOV T26BFR+6,R1 ;GET MESSAGE BUFFER  
7980 066364 010102 MOV R1,R2 ;SET UP EXPECTED  
7981 066366 052702 010000 BIS #BIT12,R2 ;SET THE RLL BIT IN EXPECTED  
7982 066372 020102 CMP R1,R2 ;ARE THEY EQUAL  
7983 066374 001406 BEQ 180$ ;BR, IF EQUAL (ALL IS WELL)  
7984 066376 005237 002212 INC FATFLG ;BUMP COUNT  
7988 066402 ERRHRD ERRNO,T26LON,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO  
TRAP C$ERHRD  
.WORD 709  
.WORD T26LON  
.WORD EXPREC  
180$: CKLOOP TRAP C$CLP1  
7990 066414 012703 000777 MOV #511.,R3 ;SET UP SIZE OF RECORD  
7991 066420 013737 003114 071722 MOV FREE,T26RB ;STARTING READ BUFFER ADDRESS  
:*****  
:REREAD DATA,CVC=1,ACK COMMAND  
:*****  
8000 066434 012704 071720 365$: MOV #141401,T26PK3 ;REREAD DATA,CVC=1,ACK 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  
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
```



```
8092 066656 012124
      066660 104406
      066660 104406
8093
8094
8095
8096
8097
8098
8099
8100 066662 004737 011104
8101 066666 016501 000002
8102 066672 012702 000200
8103 066676 103407
8104 066700 010004
8105 066702 005237 002212
8109 066706
      066706 104456
      066710 001312
      066712 073254
      066714 012136
8110 066716
      066716 104406
8111
8112
8113
8114
8115
8116
8117
8118 066720 013701 071630
8119 066724 010102
8120 066726 052702 000002
8121 066732 020102
8122 066734 001406
8123 066736 005237 002212
8127 066742
      066742 104456
      066744 001313
      066746 072765
      066750 015564
8128 066752
      066752 104406
8129 066754 012703 000400
8130 066760 013737 003114 071722
8131
8132
8133
8134
8135
8136
8137
8138 066766 012737 140005 071720
8139 066774 012704 071720
8140 067000
8141 067000 010337 071726
8142 067004 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
      MOV TSSR(R5),R1 ;GET TSSR
      MOV #SSR,R2 ;SET UP EXPECTED TSSR
      BCS 30$ ;BR, IF NO PROBLEM
      MOV R0,R4 ;PACKET ADDRESS SET UP
      INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP C$ERHRD
                                .WORD 714
                                .WORD T26RWN
                                .WORD PKTSSR
8110 30$: 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$ERHRD
                                .WORD 715
                                .WORD T26BOT
                                .WORD EXPREC
8128 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      MOV #256,R3 ;RECORD SIZE
      MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
8132
8133
8134
8135
8136
8137
8138
8139
8140
8141
8142
65$:
      MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK 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
```

```

8143 067010 004737 016340      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
8144 067014 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
8145 067020 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
8146 067024 020102              CMP      R1,R2        ;ARE THEY EQUAL
8147 067026 001406      BEQ      75$          ;BR, IF OK
8148 067030 005237 002212      INC      FATFLG       ;BUMP COUNT
8152 067034              ERRHRD  ERRNO,WRterr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      067034 104456              TRAP    C$ERHRD
      067036 001314              .WORD  716
      067040 005111              .WORD  WRterr
      067042 012136              .WORD  PKTSSR
8153 067044              75$:   CKLOOP          ;LOOP IF SELECTED
      067044 104406              TRAP    C$CLP1
8154 067046              120$:
8155
8156      ;*****
8157      ;
8158      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8159      ;
8160      ;*****
8161
8162 067046 004737 011104      JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
8163 067052 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR
8164 067056 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED TSSR
8165 067062 103407      BCS     130$          ;BR, IF NO PROBLEM
8166 067064 010004      MOV      R0,R4        ;PACKET ADDRESS SET UP
8167 067066 005237 002212      INC      FATFLG       ;BUMP COUNT
8171 067072              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      067072 104456              TRAP    C$ERHRD
      067074 001315              .WORD  717
      067076 073254              .WORD  T26RWN
      067100 012136              .WORD  PKTSSR
8172 067102              130$:  CKLOOP          ;LOOP IF SELECTED
      067102 104406              TRAP    C$CLP1
8173
8174      ;*****
8175      ;
8176      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8177      ;
8178      ;*****
8179
8180 067104 013701 071630      MOV      T26FR+6,R1    ;PICK UP XSTO
8181 067110 010102      MOV      R1,R2        ;SET UP EXPECTED
8182 067112 052702 000002      BIS     #BIT1,R2      ;SET BOT BIT IN EXPECTED
8183 067116 020102      CMP      R1,R2        ;DOES EXP = REC'D
8184 067120 001406      BEQ     135$          ;BR, IF EQUAL (OK)
8185 067122 005237 002212      INC     FATFLG       ;BUMP COUNT
8189 067126              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      067126 104456              TRAP    C$ERHRD
      067130 001316              .WORD  718
      067132 072765              .WORD  T26BOT
      067134 015564              .WORD  EXPREC
8190 067136              135$:  CKLOOP          ;LOOP IF SELECTED
      067136 104406              TRAP    C$CLP1
8191 067140 012703 001000      MOV     #512,R3       ;RECORD SIZE
8192 067144 013737 003114 071722  MOV     FREE,T26RB    ;STARTING READ BUFFER ADDRESS
8193

```

```

8194
8195
8196
8197
8198
8199
8200 067152 012737 161401 071720
8201 067160 012704 071720
8202 067164 010337 071726
8203 067170 010465 000000
8204 067174 004737 016340
8205 067200 016501 000002
8206 067204 012702 100204
8207 067210 020102
8208 067212 001406
8209 067214 075237 002212
8213 067220
      067220 104456
      067222 001317
      067224 074332
      067226 012136
8214 067230
      067230 104406
8215
8216
8217
8218
8219
8220
8221
8222 067232 013701 071630
8223 067236 010102
8224 067240 052702 040000
8225 067244 020102
8226 067246 001406
8227 067250 005237 002212
8231 067254
      067254 104456
      067256 001320
      067260 074162
      067262 015564
8232 067264
      067264 104406
8233 067266 013701 071626
8234 067272 012702 000400
8235 067276 020102
8236 067300 001405
8240 067304
      067304 104456
      067306 001320
      067310 074244
      067312 015564
8241 067314
      067314 104406
8242 067316 012703 001000
8243 067322 013737 003114 071722
8244

```

```

:*****
: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$ERHRD
                                .WORD 719
                                .WORD T26TRL
                                .WORD PKTSSR
170$:  CKLOOP                      ;LOOP IF SELECTED
                                TRAP  C$CLP1
:*****
:READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
:*****
      MOV      T26BFR+6,R1        ;GET MESSAGE BUFFER
      MOV      R1,R2              ;SET UP EXPECTED
      BIS      #BIT14,R2          ;SET THE RLS BIT IN EXPECTED
      CMP      R1,R2              ;ARE THEY EQUAL
      BEQ      180$              ;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                      ;LOOP IF SELECTED
                                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 ;RBPCL 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

```


TSV7 - HARDWARE TESTS 1-8
TEST 6: REREADS

MACRO M1113 25-MAY-82 09:19 PAGE 137-5

K 8

SEQ 0307

8294 067476 023727 002212 000017
8295 067504 103402
8296 067506 004737 017272
8297 067512

999\$:

CMP FATFLG,#15.
BLO 999\$
JSR PC,CKDROP

:IS ERROR COUNT AT 25
:BR, IF LESS THAN 25
:TRY TO DROP THE UNIT


```

8356 067616 010001          MOV      RO,R1          ;SAVE CONTENTS OF TSSR
8357 067620          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      067620 104456          TRAP      CSERHRD
      067622 001324          .WORD    724
      067624 005054          .WORD    WRTMSG
      067626 012124          .WORD    SFIMSG
8358 067630          26$:   CKLOOP          ;LOOP IF SELECTED
      067630 104406          TRAP      CSCLP1
8359
8360
8361
8362
8363
8364
8365
8366 067632 004737 021276      JSR      PC,INVERT      ;INVERT THE EXTENDED FEATURES SWITCH
8367 067636 004737 011104      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
8368 067642 103411          BCS      30$            ;BR, IF NO PROBLEM
8369 067644 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR
8370 067650 010004          MOV      RO,R4          ;PACKET ADDRESS SET UP
8371 067652 005237 002212      INC      FATFLG         ;BUMP COUNT
8375 067656          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      067656 104456          TRAP      CSERHRD
      067660 001325          .WORD    725
      067662 073254          .WORD    T26RWN
      067664 012136          .WORD    PKTSSR
8376 067666          30$:   CKLOOP          ;LOOP IF SELECTED
      067666 104406          TRAP      CSCLP1
8377
8378
8379
8380
8381
8382
8383
8384 067670 013701 071630      MOV      T26BFR+6,R1    ;PICK UP XSTO
8385 067674 010102          MOV      R1,R2          ;SET UP EXPECTED
8386 067676 052702 000002      BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
8387 067702 020102          CMP      R1,R2          ;DOES EXP = REC'D
8388 067704 001406          BEQ      40$            ;BR, IF EQUAL (OK)
8389 067706 005237 002212      INC      FATFLG         ;BUMP COUNT
8393 067712          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      067712 104456          TRAP      CSERHRD
      067714 001326          .WORD    726
      067716 072765          .WORD    T26BOT
      067720 015564          .WORD    EXPREC
8394 067722          40$:   CKLOOP          ;LOOP IF SELECTED
      067722 104406          TRAP      CSCLP1
8395 067724 013737 003114 071722      MOV      FREE,T26RB     ;STARTING WRITE BUFFER ADDRESS
8396
8397
8398
8399
8400
8401
8402
8403 067732 012737 140005 071720      MOV      #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
  
```



```
8404 067740 012704 071720          MOV      #T26PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
8405 067744 012737 000400 071726 65$: MOV      #256.,T26SZ        ;SET UP RECORD SIZE IN PACKET
8406 067752 013777 071746 113134 MOV      T26CNT,@FREE      ;MOVE TAPE RECORD NUMBER TO BU:FER
8407 067760 062737 000001 071746 ADD      #1,T26CNT        ;NUMBER READY FOR NEXT RECORD
8408 067766 010465 000000          MOV      R4,TSDB(R5)      ;ISSUE COMMAND
8409 067772 004737 016340          JSR      PC,WAITF        ;WAIT FOR SSR TO SET
8410 067776 016501 000002          MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
8411 070002 012702 000200          MOV      #SSR,R2        ;SET UP EXPECTED
8412 070006 020102          CMP      R1,R2          ;ARE THEY EQUAL
8413 070010 001406          BEQ      75$            ;BR, IF OK
8414 070012 005237 002212          INC      FATFLG        ;BUMP COUNT
8418 070016          ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      070016 104456          TRAP    C$ERHRD
      070020 001327          .WORD  727
      070022 005111          .WORD  WRTERR
      070024 012136          .WORD  PKTSSR
8419 070026          75$:  CKLOOP          ;LOOP IF SELECTED
      070026 104406          TRAP    C$CLP1
8420 070030 022737 000013 071746          CMP      #11.,T26CNT    ;CHECK NUMBER OF RECORDS WRITTEN
8421 070036 001401          BEQ      120$          ;BR, IF AT END OF WRITE SEQUENCE
8422 070040 000741          BR       65$            ;WRITE MORE RECORDS
8423 070042          120$: CLR      T26CNT        ;SET RECORD COUNTER BACK TO ZERO
8424 070042 005037 071746
8425
8426
8427
8428
8429
8430
8431
      :*****
      :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
      :*****
8432 070046 004737 011104          JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
8433 070052 103411          BCS     130$          ;BR, IF NO PROBLEM
8434 070054 016501 000002          MOV      TSSR(R5),R1   ;GET TSSR
8435 070060 010004          MOV      R0,R4        ;PACKET ADDRESS SET UP
8436 070062 005237 002212          INC      FATFLG        ;BUMP COUNT
8440 070066          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      070066 104456          TRAP    C$ERHRD
      070070 001330          .WORD  728
      070072 073254          .WORD  T26RWN
      070074 012136          .WORD  PKTSSR
8441 070076          130$: CKLOOP          ;LOOP IF SELECTED
      070076 104406          TRAP    C$CLP1
8442
8443
8444
8445
8446
8447
8448
      :*****
      :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
      :*****
8449 070100 013701 071630          MOV      T26BFR+6,R1   ;PICK UP XSTO
8450 070104 010102          MOV      R1,R2        ;SET UP EXPECTED
8451 070106 052702 000002          BIS      #BIT1,R2     ;SET BOT BIT IN EXPECTED
8452 070112 020102          CMP      R1,R2        ;DOES EXP = REC'D
8453 070114 001406          BEQ      140$          ;BR, IF EQUAL (OK)
8454 070116 005237 002212          INC      FATFLG        ;BUMP COUNT
8458 070122          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      070122 104456          TRAP    C$ERHRD
```

```
070124 001331
070126 072765 .WORD 729
070130 015564 .WORD T26BOT
8459 070132 140$: CKLOOP ;LOOP IF SELECTED .WORD EXPREC
070132 104406 TRAP C$CLP1
8460 070134 012703 071736 MOV #T26RN,R3 ;COMMAND BUFFER ADDRESS
8461 070140 013737 003130 071722 150$: MOV NXML0,T26RB ;STARTING READ BUFFER ADDRESS
8462 070146 013737 003132 071724 MOV NXMH1,T26RB+2 ;SET UP HIGH ORDER ADDRESS BITS
8463
8464 ;*****
8465 ;REREAD DATA,IE,ACK, OPP COMMAND
8466 ;*****
8467
8468
8469
8470 070154 011337 071720 MOV (R3),T26PK3 ;REREAD DATA,IE,ACK, OPP COMMAND
8471 070160 012704 071720 165$: MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8472 070164 012737 000400 071726 MOV #256.,T26SZ ;SET UP RECORD SIZE IN PACKET
8473 070172 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
8474 070176 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
8475 070202 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8476 070206 012702 104210 MOV #SSR!NXM!SC!BIT3,R2 ;SET UP EXPECTED
8477 070212 020102 CMP R1,R2 ;ARE THEY EQUAL
8478 070214 001414 BEQ 170$ ;BR, IF OK
8479 070216 031327 001000 BIT (R3),#BIT9 ;CHECK FOR A READ COMMAND
8480 070222 001403 BEQ 168$ ;BR, IF IT WAS A READ COMMAND
8481 070224 030127 000002 BIT R1,#BIT1 ;WAS BIT1 SET
8482 070230 001006 BNE 170$ ;BR, IF REREAD AND BIT1 SET
8483 070232
8484 070232 005237 002212 168$: INC FATFLG ;BUMP COUNT
8488 070236 ERRHRD ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
070236 104456 TRAP C$ERHRD
070240 001332 .WORD 730
070242 072175 .WORD T26RRF
070244 012136 .WORD PKTSSR
8489 070246 170$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
070246 104406
8490
8491 ;*****
8492 ;READ DATA, ACK,CVC=1 COMMAND
8493 ;*****
8494
8495
8496
8497 070250 012737 140001 071720 MOV #140001,T26PK3 ;READ DATA, ACK,CVC=1 COMMAND
8498 070256 012737 000400 071726 MOV #256.,T26SZ ;SET SIZE INTO PACKET
8499 070264 005037 071724 CLR T26RB+2 ;CLEAR OUT HIGH ADDRESS BITS
8500 070270 013737 003114 071722 MOV FREE,T26RB ;GIVE READ A GOOD BUFFER
8501 070276 010465 000000 MOV R4,TSDB(R5) ;ISSUE READ DATA COMMAND
8502 070302 004737 016340 JSR PC,WAITF ;WAIT FOR SSR
8503 070306 016501 000002 MOV TSSR(R5),R1 ;PICK UP THE TSSR
8504 070312 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
8505 070316 020102 CMP R1,R2 ;IS THE TSSR OK
8506 070320 001406 BEQ 180$ ;BR, IF TSSR OK (GOOD)
8507 070322 005237 002212 INC FATFLG ;BUMP COUNT
8511 070326 ERRHRD ERRNO,RDERR,PKTSSR ;READ DATA COMMAND FAILED
070326 104456 TRAP C$ERHRD
```

```
070330 001333
070332 005204
070334 012136
8512 070336 180$: CKLOOP ;LOOP IF SELECTED .WORD 731
070336 104406 ;FIRST WORD FROM READ BUFFER TRAP C$CLP1
8513 070340 017701 112550 MOV @FREE,R1 ;SET UP EXPECTED
8514 070344 012702 000001 MOV #1,R2 ;IS TAPE POSITION CORRECT
8515 070350 020102 CMP R1,R2 ;KEEP GOING POSITION OK
8516 070352 001406 BEQ 190$ ;BUMP COUNT
8517 070354 005237 002212 INC FATFLG ;TAPE POSITION INCORRECT
8521 070360 ERRHRD ERRNO,T26WNG,EXPREC TRAP C$ERHRD
070360 104456 .WORD 732
070362 001334 .WORD T26WNG
070364 071756 .WORD EXPREC
070366 015564
8522 070370 190$: CKLOOP TRAP C$CLP1
070370 104406
8523
8524
8525
8526
8527
8528
8529
:*****
:ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
:*****
8530 070372 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
8531 070376 103411 BCS 194$ ;BR, IF NO PROBLEM
8532 070400 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
8533 070404 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
8534 070406 005237 002212 INC FATFLG ;BUMP COUNT
8538 070412 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
070412 104456 TRAP C$ERHRD
070414 001335 .WORD 733
070416 073254 .WORD T26RWN
070420 012136 .WORD PKTSSR
8539 070422 194$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
070422 104406
8540
8541
8542
8543
8544
8545
8546
:*****
:READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
:*****
8547 070424 013701 071630 MOV T26BFR+6,R1 ;PICK UP XST0
8548 070430 010102 MOV R1,R2 ;SET UP EXPECTED
8549 070432 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
8550 070436 020102 CMP R1,R2 ;DOES EXP = REC'D
8551 070440 001406 BEQ 196$ ;BR, IF EQUAL (OK)
8552 070442 005237 002212 INC FATFLG ;BUMP COUNT
8556 070446 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
070446 104456 TRAP C$ERHRD
070450 001336 .WORD 734
070452 072765 .WORD T26BOT
070454 015564 .WORD EXPREC
8557 070456 196$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
070456 104406
8558 070460 010302 MOV R3,R2 ;SAVE R3 FOR A MOMENT
```



```
8632
8633
8634
8635
8636
8637
8638
8639 070626 004737 011104
8640 070632 016501 000002
8641 070636 012702 000200
8642 070642 103407
8643 070644 010004
8644 070646 005237 002212
8648 070652
      070652 104456
      070654 001341
      070656 073254
      070660 012136
8649 070662
      070662 104406
8650
8651
8652
8653
8654
8655
8656
8657 070664 013701 071630
8658 070670 010102
8659 070672 052702 000002
8660 070676 020102
8661 070700 001406
8662 070702 005237 002212
8666 070706
      070706 104456
      070710 001342
      070712 072765
      070714 015564
8667 070716
      070716 104406
8668 070720 012737 000400 071726
8669 070726 013737 003114 071722
8670 070734 005703
8671 070736 001404
8672
8673
8674
8675
8676
8677
8678
8679 070740 012737 161001 071720
8680 070746 000403
8681
8682
8683
8684
```

```
*****
:ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
*****
26$: JSR PC,REWIND ;CALL TAPE REWIND COMMAND
      MOV TSSR(R5),R1 ;GET TSSR
      MOV #SSR,R2 ;SET UP EXPECTED TSSR
      BCS 30$ ;BR, IF NO PROBLEM
      MOV R0,R4 ;PACKET ADDRESS SET UP
      INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                           TRAP C$ERHRD
                                           .WORD 737
                                           .WORD T26RWN
                                           .WORD PKTSSR
30$: CKLOOP ;LOOP IF SELECTED
                                           TRAP C$CLP1
*****
:READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
*****
      MOV T26BFR+6,R1 ;PICK UP XST0
      MOV R1,R2 ;SET UP EXPECTED
      BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
      CMP R1,R2 ;DOES EXP = REC'D
      BEQ 40$ ;BR, IF EQUAL (OK)
      INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                           TRAP C$ERHRD
                                           .WORD 738
                                           .WORD T26BOT
                                           .WORD EXPREC
40$: CKLOOP ;LOOP IF SELECTED
                                           TRAP C$CLP1
      MOV #256.,T26SZ ;SET UP RECORD SIZE IN PACKET
      MOV FREE,T26RB ;ADDRESS OF READ BUFFER
      TST R3 ;CHECK NUMBER OF TIMES THROUGH HERE
      BEQ 50$ ;BR, IF FIRST TIME THROUGH HERE
*****
:REREAD,CVC=1,ACK COMMAND
*****
      MOV #161001,T26PK3 ;REREAD,CVC=1,ACK COMMAND
      BR 55$ ;SKIP NEXT COMMAND
*****
:REREAD,ACK COMMAND
```

```
8685  
8686  
8687  
8688 070750 012737 141001 071720 50$: MOV #141001,T26PK3 ;REREAD,ACK COMMAND  
8689 070756 55$:  
8690 070756 012704 071720 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS  
8691 070762 65$:  
8692 070762 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND  
8693 070766 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET  
8694 070772 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS  
8695 070776 012702 100206 MOV #SSR!SC.BIT1!BIT2,R2 ;SET UP EXPECTED  
8696 071002 020102 CMP R1,R2 ;ARE THEY EQUAL  
8697 071004 001406 BEQ 75$ ;BR, IF OK  
8698 071006 005237 002212 INC FATFLG ;BUMP COUNT  
8702 071012 ERRHRD ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA  
071012 104456 TRAP C$ERHRD  
071014 001343 .WORD 739  
071016 072713 .WORD T26WDE  
071020 012136 .WORD PKTSSR  
8703 071022 75$: CKLOOP ;LOOP IF SELECTED  
071022 104406 TRAP C$CLP1  
8704  
8705  
8706  
8707  
8708  
8709  
8710  
8711 071024 013701 071630 MOV T26BFR+6,R1 ;GET XST0 STATUS WORD  
8712 071030 010102 MOV R1,R2 ;SET UP EXPECTED  
8713 071032 052702 002000 BIS #BIT10,R2 ;SET THE NEF BIT  
8714 071036 020102 CMP R1,R2 ;ARE THEY EQUAL  
8715 071040 001406 BEQ 170$ ;BR, IF EQUAL (GOOD)  
8716 071042 005237 002212 INC FATFLG ;BUMP COUNT  
8720 071046 ERRHRD ERRNO,T26NEF,EXPREC ;NEF SHOULD BE SET  
071046 104456 TRAP C$ERHRD  
071050 001344 .WORD 740  
071052 072044 .WORD T26NEF  
071054 015564 .WORD EXPREC  
8721 071056 170$: CKLOOP TRAP C$CLP1  
071056 104406  
8722 071060 005103 COM R3 ;RESET THE SWITCH  
8723 071062 001261 BNE 26$ ;BR, IF FIRST TIME THROUGH HERE  
8724 071064 ENDSUB  
071064  
071064 104403 L10120:  
8725 071066 023727 002212 000017 CMP FATFLG,#15. ;IS ERROR COUNT AT 25 TRAP C$ESUB  
8726 071074 103402 BLO 999$ ;BR, IF LESS THAN 25  
8727 071076 004737 017272 JSR PC,CKDROP ;TRY TO DROP THE UNIT  
3728 071102 999$:
```



```
071206 104406 TRAP C$CLP1
8783
8784
8785 *****
8786 :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8787 :
8788 *****
8789
8790 071210 004737 011104 26$: JSR PC,REWIND ;CALL TAPE REWIND COMMAND
8791 071214 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
8792 071220 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
8793 071224 103407 BCS 30$ ;BR, IF NO PROBLEM
8794 071226 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
8795 071230 005237 002212 INC FATFLG ;BUMP COUNT
8799 071234 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
071234 104456 TRAP C$ERHRD
071236 001347 .WORD 743
071240 073254 .WORD T26RWN
071242 012136 .WORD PKTSSR
8800 071244 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
071244 104406
8801
8802 *****
8803 :
8804 :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
8805 :
8806 *****
8807
8808 071246 013701 071630 MOV T26BFR+6,R1 ;PICK UP XST0
8809 071252 010102 MOV R1,R2 ;SET UP EXPECTED
8810 071254 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
8811 071260 020102 CMP R1,R2 ;DOES EXP = REC'D
8812 071262 001406 BEQ 40$ ;BR, IF EQUAL (OK)
8813 071264 005237 002212 INC FATFLG ;BUMP COUNT
8817 071270 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
071270 104456 TRAP C$ERHRD
071272 001350 .WORD 744
071274 072765 .WORD T26BOT
071276 015564 .WORD EXPREC
8818 071300 40$: CKLOOP TRAP C$CLP1
071300 104406
8819
8820 *****
8821 :
8822 :ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
8823 :BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
8824 :
8825 *****
8826
8827 071302 012703 000001 MOV #000001,R3 ;SET UP SPACE FORWARD 1 RECORD
8828 071306 004737 010556 JSR PC,SPACE ;ISSUE SPACE COMMAND
8829 071312 103411 BCS 75$ ;BR, IF OK
8830 071314 016501 000002 MOV TSSR(R5),R1 ;GET STATUS DATA
8831 071320 010004 MOV R0,R4 ;GET PACKET ADDRESS
8832 071322 005237 002212 INC FATFLG ;BUMP COUNT
8836 071326 ERRHRD ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
071326 104456 TRAP C$ERHRD
```

```
071330 001351 .WORD 745
071332 072713 .WORD T26WDE
071334 012136 .WORD PKTSSR
8837 071336 75$: CKLOOP :LOOP IF SELECTED TRAP C$CLP1
071336 104406
8838
8839 :*****
8840 :ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
8841 :BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
8842 :*****
8843
8844
8845
8846 071340 012703 100001 MOV #100001,R3 ;SET SPACE REVERSE 1 RECORD
8847 071344 004737 010556 JSR PC,SPACE ;ISSUE COMMAND
8848 071350 103411 BCS 175$ ;GO ON IF ALL IS WELL
8849 071352 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8850 071356 010004 MOV R0,R4 ;SET UP EXPECTED (PACKET CONTENTS)
8851 071360 005237 002212 INC FATFLG ;BUMP COUNT
8855 071364 ERRHRD ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
071364 104456 TRAP C$ERHRD
071366 001352 .WORD 746
071370 072713 .WORD T26WDE
071372 012136 .WORD PKTSSR
8856 071374 175$: CKLOOP :LOOP IF SELECTED TRAP C$CLP1
071374 104406
8857 071376 013737 003114 071722 MOV FREE,T26RB ;ADDRESS OF BUFFER
8858 071404 005737 071750 TST T26CNU ;CHECK FOR TIMES THROUGH HERE
8859 071410 001404 BEQ 176$ ;BR, IF FIRST TIME THROUGH
8860
8861 :*****
8862 :REREAD (PREVIOUS),IE,ACK,OPP=1 CMD.
8863 :*****
8864
8865
8866
8867 071412 012737 161001 071720 MOV #161001,T26PK3 ;REREAD (PREVIOUS),IE,ACK,OPP=1 CMD.
8868 071420 000403 BR 178$ ;SKIP NEXT COMMAND
8869
8870 :*****
8871 :REREAD ,ACK,OPP=1 COMMAND
8872 :*****
8873
8874
8875
8876 071422 012737 141001 071720 176$: MOV #141001,T26PK3 ;REREAD ,ACK,OPP=1 COMMAND
8877 071430 178$:
8878 071430 012704 071720 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8879 071434 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
8880 071440 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
8881 071444 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8882 071450 012702 100204 MOV #SSR!SC!BIT2,R2 ;SET UP EXPECTED
8883 071454 020102 CMP R1,R2 ;ARE THEY EQUAL
8884 071456 001406 BEQ 180$ ;BR, IF OK
8885 071460 005237 002212 INC FATFLG ;BUMP COUNT
8889 071464 ERRHRD ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
071464 104456 TRAP C$ERHRD
```


8918
8919
8920
8922 071600
8924 071600 071600
8925 071600 014004
8926 071602 071610
8927 071604 000000
8928 071606 000012
8929 071610
8930 071610 071622
8931 071612 000000
8932 071614 000024
8933 071616 000000
8934 071620 000000
8935 071622
8936
8937
8938
8940 071710
8942 071710
8943 071710 100006
8944 071712 071730
8945 071714 000000
8946 071716 000006
8947
8951 071720
8952 071720 140005
8953 071722
8954 071722 003114
8955 071724 000000
8956 071726 000000
8957
8958
8959
8960
8961 071730
8962 071730 010
8963 071731 200
8964 071732 000000
8965 071734 000000
8966
8967
8968
8969
8970
8971 071736 140001
8972 071740 141401
8973 071742 161401
8974 071744 177777
8975
8976
8977 071746 000000
8978 071750 000000
8979
8980 071752 000000
8981

```

:+
: LOCAL STORAGE FOR THIS TEST
:-
      .=<.+10>&177770
T26PACKET:
      .WORD 14004
      .WORD T26DATA
      .WORD 0
      .WORD 10.
T26DATA:
      .WORD T26BFR
      .WORD 0
      .WORD 20.
      .WORD 0
T26DSW: .WORD 0
T26BFR: .BLKW 25.
: WRITE SUBSYSTEM MEMORY COMMAND PACKET
      .=<.+10>&177770
T26PK2:
      .WORD 100006
      .WORD T26BF2
      .WORD 0
      .WORD 6.
T26PK3:
      .WORD 140005
T26RB:
T26WB: .WORD FREE
      .WORD 0
T26SZ: .WORD 0
      .EVEN
:
:
T26BF2:
T26BS0: .BYTE 10
T26BS1: .BYTE 200
T26S2: .WORD 0
T26S3: .WORD 0
:
      .EVEN
: TAPE MOTION PACKET COMMAND VALUES
T26DRN: .WORD 140001
      .WORD 141401
      .WORD 161401
      .WORD 177777
:
T26CNT: .WORD 0
T26CNU: .WORD 0
T26RSZ: .WORD 0

```

: COMMAND PACKET FOR TEST
: WRITE CHARACTERISTICS COMMAND, WITH CVC=1, 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
: WRITE SUB SYS MEM COMMAND, AND ACK
: ADDRESS OF SELECT BLOCK DATA
: SIZE OF DATA PACKET
: REREAD COMMAND, CVC=1 AND ACK
: ADDRESS OF WRITE BUFFER
: SIZE OF BUFFER (EXTENT)
: BSEL0 AREA
: BSEL1 AREA
: SEL 2 AREA
: DATA AREA
: READ DATA
: REREAD NEXT OPP=0
: REREAD NEXT OPP=1
: END OF DATA
: TAPE RECORD COUNTER STORAGE AREA
: TAPE RECORD COUNTER STORAGE AREA
: RECORD STORAGE SIZE AREA

TSV7 - HARDWARE TESTS 1-8
TEST 6: REREADS

MACRO M1113 25-MAY-82 09:19 PAGE 141-1

M 9

SEQ 0322

8982 071754 000000
8983

T26DLY: .WORD 0

;DELAY COUNTER AREA

```

8985
8986
8987
8988
8989
8990
8991 071756      124      141      160  T26WNG: .ASCIZ  'Tape Position Incorrect After REREAD Previous (OPP=1)'
8992 072044      122      105      122  T26NEF: .ASCIZ  'REREAD PREVIOUS, At BOT, Failed To Set NEF (XST0)'
8993 072126      124      123      123  T26RDF: .ASCIZ  'TSSR Incorrect After READ DATA Command'
8994 072175      122      105      122  T26RRF: .ASCIZ  'REREAD Previous (Space Reverse, Read Forward) Command Failed'
8995 072272      122      105      122  T26RRG: .ASCIZ  'REREAD Previous (Read Reverse, Space Forward) Command Failed'
8996 072367      120      117      123  T26SC: .ASCIZ   'POSITION (Space Command) Failed, TSSR Not Correct'
8997 072451      122      111      102  T26LOR: .ASCIZ  'RIB NOT SET AFTER READ REVERSE INTO BOT'
8998 072521      124      123      123  T26WDF: .ASCIZ  'TSSR Not Correct After Illegal Mode Bits Set'
8999 072576      111      154      154  T26LOQ: .ASCIZ  'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
9000 072657      122      105      122  T26SSR: .ASCIZ  'REREAD COMMAND Not Accepted'
9001 072713      124      123      123  T26WDE: .ASCIZ  'TSSR Not Correct After WRITE DATA Command'
9002 072765      124      141      160  T26BOT: .ASCIZ  'Tape Not At BOT After REWIND Command'
9003 073032      104      141      164  T26DTA: .ASCIZ  'Data Written To Tape Not Equal To Data Read From Tape'
9004 073120      122      105      122  T26EOT: .ASCIZ  'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
9005 073177      124      123      123  T26TM: .ASCIZ   'TSSR Not Correct After REREAD COMMAND Reject'
9006 073254      122      145      167  T26RWN: .ASCIZ  'Rewind (POSITION) Command Not Accepted'
9007 073323      122      101      115  T26RNC: .ASCIZ  'RAM Error, Correct Data Pattern Not In Ram'
9008 073376      124      123      123  T26AM3: .ASCIZ  'TSSR Init. Failed After REREAD COMMAND'
9009 073445      104      162      151  T26OFL: .ASCIZ  'Drive 7 Select Failed To Set 'OFL' In TSSR'
9010 073520      124      123      123  T26WDD: .ASCIZ  'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
9011 073610      124      123      123  T26WDC: .ASCIZ  'TSSR Not Correct After REREAD DATA Command'
9012 073663      103      126      103  T26VCK: .ASCIZ  'CVC Set, Didn't Reset VCK In Message Buffer'
9013 073736      124      123      102  T26BA: .ASCIZ   'TSBA Not Correct After REREAD DATA Command'
9014 074011      127      122      111  T26WSS: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
9015 074100      122      145      141  T26LON: .ASCIZ  'Reading Long Record Failed To Set RLL Bit In XST0'
9016 074162      122      145      141  T26LOP: .ASCIZ  'Reading Long Record Failed To Set RLS Bit In XST0'
9017 074244      122      145      163  T26PBP: .ASCIZ  'Residual Byte Count Incorrect After Short Record Read'
9018 074332      122      145      141  T26TRL: .ASCIZ  'Reading Long Record Failed To Give Tape Status Alert'
9019 074420      104      141      164  T26NEQ: .ASCIZ  'Data REREAD From Tape Not Correct, After SWB=1'
9020 074477      122      145      162  TST26ID: .ASCIZ  'Rereads'

```

```

: +
: LOCAL TEXT MESSAGES FOR TEST
: -

```

```

9021
9022
9023
9024
9025
9026
9027
9028
9029 074510
9030 074510
9031 074514      012701      071600
9032 074520      012721      140004
9033 074524      012721      071610
9034 074530      005021
9035 074532      012721      000012
9036 074536      012721      071622
9037 074542      005021
9038 074544      012721      000024
9039 074550      005021
9040 074552      012711      000000
9041 074556      012702      000030

```

```

: +
: ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
: WRITE SUBSYSTEM MEMORY COMMAND
: -

```

```

T26REST:
      SAVREG
      MOV      #T26PACKET,R1
      MOV      #140004,(R1)+
      MOV      #T26DATA,(R1)+
      CLR      (R1)+
      MOV      #10,(R1)+
      MOV      #T26BFR,(R1)+
      CLR      (R1)+
      MOV      #20,(R1)+
      CLR      (R1)+
      MOV      #0,(R1)
      MOV      #24.,R2
      ;SAVE THE REGISTERS
      ;START OF THE PACKET
      ;WRITE SUBSYSTEM MEM. WITH ACK, CVC=1
      ;ADDRESS OF CHARAISTICS DATA BLOCK
      ;EXTENDED ADDRESS
      ;SIZE OF DATA BLOCK IN BYTES
      ;ADDRESS OF MESSAGE BUFFER
      ;LENGTH OF MESSAGE BUFFER
      ;SELECT DRIVE ZERO (0)
      ;NUMBER OF LOCATIONS TO BE CLEARED

```

```

9042 074562 012762 177777 071622 64$: MOV #177777,T26BFR(R2) ;ALL ONES TO MESSAGE BUFFER
9043 074570 005742 TST -(R2) ;NEXT LOCATION
9044 074572 020227 000000 CMP R2,#0 ;CHECK FOR END OF LOOP
9045 074576 001371 BNE 64$ ;KEEP GOING UNTIL DONE
9046 074600 000207 RTS PC ;RETURN
9047
9048
9049 074602 T26RT2:
9050 074602 SAVREG ;SAVE THE REGISTERS
9051 074606 012701 071710 MOV #T26PK2,R1 ;START OF THE PACKET
9052 074612 012721 140006 MOV #140006,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1,
9053 074616 012721 071730 MOV #T26BF2,(R1)+ ;ADDRESS OF DATA BLOCK
9054 074622 005021 CLR (R1)+ ;EXTENDED ADDRESS
9055 074624 012721 000006 MOV #6,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
9056 074630 005021 CLR (R1)+
9057 074632 012701 071730 MOV #T26BF2,R1 ;POINT TO DATA SEL AREA
9058 074636 005021 CLR (R1)+
9059 074640 005011 CLR (R1)
9060 074642 000207 RTS PC ;RETURN
9061 074644 T26RT3:
9062 074644 SAVREG ;SAVE THE REGISTERS
9063 074650 012701 071720 MOV #T26PK3,R1 ;START OF THE PACKET
9064 074654 012721 000000 MOV #0,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
9065 074660 012721 000000 MOV #0,(R1)+ ;ADDRESS OF DATA BLOCK
9066 074664 005021 CLR (R1)+ ;EXTENDED ADDRESS
9067 074666 012711 000000 MOV #0,(R1) ;SIZE OF DATA BLOCK IN BYTES
9068 074672 000207 RTS PC ;RETURN
9069 074674 ENDTST
074674
074674 104401 L10102: TRAP CSETST
  
```



```
075010 005367 177772
075014 001375
075016 005367 177756
075022 001367
9125 075024 005337 101562
9126 075030 001356
9127 075032 005237 002212
9131 075036 010001
9132 075040
075040 104455
075042 001275
075044 003650
075046 012124
9133 075050 013737 002172 101430 20$: MOV UNITN,T27DSW ;SET UP DRIVE NUMBER
9134 075056 012704 101410 MOV #T27PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
9135
9136
9137
9138
9139
9140
9141
9142 075062 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
9143 075066 103407 BCS 25$ ;BR, IF COMMAND ISSUED OK
9144 075070 005237 002212 INC FATFLG ;BUMP COUNT
9148 075074 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
9149 075076
075076 104456 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
075100 001276 TRAP C$ERHRD
075102 005054 .WORD 702
075104 012124 .WORD WRTMSG
9150 075106 25$: CKLOOP ;LOOP IF SELECTED .WORD SFIMSG
075106 104406 TRAP C$CLP1
9151
9152
9153
9154
9155
9156
9157
9158 075110 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
9159 075114 103407 BCS 30$ ;BR, IF NO PROBLEM
9160 075116 010004 MOV RO,R4 ;SET UP REWIND PACKET ADDRESS
9161 075120 005237 002212 INC FATFLG ;BUMP COUNT
9165 075124
075124 104456 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
075126 001277 TRAP C$ERHRD
075130 102735 .WORD 703
075132 012136 .WORD T27RWN
9166 075134 30$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
075134 104406 TRAP C$CLP1
9167
9168
9169
9170
9171
9172
```

```

9173
9174 075136 013701 101440      MOV      T27BFR+6,R1      ;PICK UP XSTO
9175 075142 010102      MOV      R1,R2           ;SET UP EXPECTED
9176 075144 052702 000002      BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
9177 075150 020102      CMP      R1,R2           ;DOES EXP = REC'D
9178 075152 001406      BEQ      40$             ;BR, IF EQUAL (OK)
9179 075154 005237 002212      INC      FATFLG          ;BUMP COUNT
9183 075160      ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      075160 104456      TRAP      C$SERHRD
      075162 001300      .WORD    704
      075164 102431      .WORD    T27BOT
      075166 015564      .WORD    EXPREC
9184 075170      40$:  CKLOOP           ;LOOP IF SELECTED      TRAP      C$CLP1
      075170 104406
9185 075172 012737 000400 101536      MOV      #256.,T27SZ     ;SET UP RECORD SIZE
9186 075200 013737 003114 101532      MOV      FREE,T27WB     ;ADDRESS OF WRITE BUFFER
9187
9188      ;*****
9189      ;WRITE DATA RETRY,ACK,CVC=1 COMMAND
9190      ;*****
9191
9192
9193
9194 075206 012737 141005 101530      MOV      #141005,T27PK3 ;WRITE DATA RETRY,ACK,CVC=1 COMMAND
9195 075214 012704 101530      MOV      #T27PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
9196 075220 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
9197 075224 004737 016340      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
9198 075230 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
9199 075234 012702 100206      MOV      #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
9200 075240 020102      CMP      R1,R2           ;ARE THEY EQUAL
9201 075242 001406      BEQ      75$             ;BR, IF OK
9202 075244 005237 002212      INC      FATFLG          ;BUMP COUNT
9206 075250      ERRHRD  ERRNO,T27WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      075250 104456      TRAP      C$SERHRD
      075252 001300      .WORD    705
      075254 102342      .WORD    T27WDE
      075256 012136      .WORD    PKTSSR
9207 075260      75$:  CKLOOP           ;LOOP IF SELECTED      TRAP      C$CLP1
      075260 104406
9208
9209      ;*****
9210      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9211      ;*****
9212
9213
9214
9215 075262 013701 101440      MOV      T27BFR+6,R1    ;GET XSTO STATUS WORD
9216 075266 010102      MOV      R1,R2           ;SET UP EXPECTED
9217 075270 052702 002000      BIS      #BIT10,R2      ;SET THE NEF BIT
9218 075274 020102      CMP      R1,R2           ;ARE THEY EQUAL
9219 075276 001406      BEQ      170$           ;BR, IF EQUAL (GOOD)
9220 075300 005237 002212      INC      FATFLG          ;BUMP COUNT
9224 075304      ERRHRD  ERRNO,T27NEF,EXPREC ;NEF SHOULD BE SET
      075304 104456      TRAP      C$SERHRD
      075306 001302      .WORD    706
      075310 104101      .WORD    T27NEF
      075312 015564      .WORD    EXPREC

```

TSV7 - HARDWARE TESTS 1-8
TEST 7: WRITE DATA RETRY

MACRO M1113 25-MAY-82 09:19 PAGE 143-3 F 10

SEQ 0328

9225	075314				170\$:	CKLOOP			
	075314	104406						TRAP	C\$CLP1
9226	075316					ENDSUB			
	075316							L10123:	
	075316	104403						TRAP	C\$ESUB
9227	075320	023727	002212	000017		CMP	FATFLG,#15.	:IS ERROR COUNT AT 25	
9228	075326	103402				BLO	999\$:BR, IF LESS THAN 25	
9229	075330	004737	017272			JSR	PC,CKDROP	:TRY TO DROP THE UNIT	
9230	075334				999\$:				


```
9284
9285
9286
9287
9288
9289 075436 004737 011104      JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
9290 075442 103411      BCS    26$            ;BR, IF NO PROBLEM
9291 075444 010004      MOV    R0,R4         ;SET UP REWIND PACKET ADDRESS
9292 075446 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
9293 075457 005237 002212      INC    FATFLG        ;BUMP COUNT
9297 075456      ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
      075456 104456      TRAP   C$ERHRD
      075460 001305      .WORD  709
      075462 102735      .WORD  T27RWN
      075464 012136      .WORD  PKTSSR
9298 075466      26$:  CKLOOP      ;LOOP IF SELECTED
      075466 104406      TRAP   C$CLP1
9299 075470 012703 000400      MOV    #256,R3       ;STARTING RECORD SIZE
9300 075474 013737 003114 101532      MOV    FREE,T27WB    ;STARTING WRITE BUFFER ADDRESS
9301
9302
9303
9304
9305
9306
9307
9308 075502 012737 140005 101530      MOV    #140005,T27PK3 ;WRITE DATA,CVC=1,ACK COMMAND
9309 075510 012704 101530      MOV    #T27PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
9310 075514 010337 101536      MOV    R3,T27SZ      ;SET UP RECORD SIZE IN PACKET
9311 075520 010465 000000      MOV    R4,TSDB(R5)   ;ISSUE COMMAND
9312 075524 004737 016340      JSR    PC,WAITF      ;WAIT FOR SSR TO SET
9313 075530 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
9314 075534 012702 000200      MOV    #SSR,R2       ;SET UP EXPECTED
9315 075540 020102      CMP    R1,R2         ;ARE THEY EQUAL
9316 075542 001406      BEQ    28$           ;BR, IF OK
9317 075544 005237 002212      INC    FATFLG        ;BUMP COUNT
9321 075550      ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      075550 104456      TRAP   C$ERHRD
      075552 001306      .WORD  710
      075554 005111      .WORD  WRERR
      075556 012136      .WORD  PKTSSR
9322 075560      28$:  CKLOOP      ;LOOP IF SELECTED
      075560 104406      TRAP   C$CLP1
9323
9324
9325
9326
9327
9328
9329
9330 075562 004737 011104      JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
9331 075566 103411      BCS    30$            ;BR, IF NO PROBLEM
9332 075570 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
9333 075574 010004      MOV    R0,R4         ;SET UP REWIND PACKET ADDRESS
9334 075576 005237 002212      INC    FATFLG        ;BUMP COUNT
9338 075602      ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
      075602 104456      TRAP   C$ERHRD
```

```
075604 0G1307 .WORD 711
075606 102735 .WORD T27RWN
075610 012136 .WORD PKTSSR
9339 075612 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
075612 104406

9340
9341 :*****
9342 :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9343 :*****
9344
9345
9346
9347 075614 013701 101440 MOV T27BFR+6,R1 ;PICK UP XSTO
9348 075620 010102 MOV R1,R2 ;SET UP EXPECTED
9349 075622 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
9350 075626 020102 CMP R1,R2 ;DOES EXP = REC'D
9351 075630 001406 BEQ 40$ ;BR, IF EQUAL (OK)
9352 075632 005237 002212 INC FATFLG ;BUMP COUNT
9356 075636 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
075636 104456 TRAP C$SERHRD
075640 001310 .WORD 712
075642 102431 .WORD T27BOT
075644 015564 .WORD EXPREC
9357 075646 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
075646 104406

9358
9359 :*****
9360 :ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
9361 :BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
9362 :*****
9363
9364
9365
9366 075650 012703 000001 MOV #1,R3 ;PARAMETER SPACE FORWARD 1 RECORD
9367 075654 004737 010556 JSR PC,SPACE ;CALL SPACE RECORDS ROUTINE
9368 075660 103413 BCS 50$ ;BR, IF NO ERRORS
9369 075662 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9370 075666 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9371 075672 010004 MOV R0,R4 ;SET UP REWIND PACKET ADDRESS
9372 075674 005237 002212 INC FATFLG ;BUMP COUNT
9376 075700 ERRHRD ERRNO,T27SCF,PKTSSR ;SPACE RECORDS COMMAND FAILED
075700 104456 TRAP C$SERHRD
075702 001311 .WORD 713
075704 104177 .WORD T27SCF
075706 012136 .WORD PKTSSR
9377 075710 50$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
075710 104406

9378
9379 :*****
9380 :ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
9381 :BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
9382 :*****
9383
9384
9385
9386 075712 012703 100001 MOV #100001,R3 ;PARAMETER SPACE REVERSE 1 RECORD
9387 075716 004737 010556 JSR PC,SPACE ;CALL SPACE RECORDS ROUTINE
```



```

9490                                     :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9491                                     :
9492                                     :*****
9493
9494 076220 004737 011104                JSR    PC,REWIND                ;CALL TAPE REWIND COMMAND
9495 076224 103407                        BCS    30$                      ;BR, IF NO PROBLEM
9496 076226 010004                        MOV    R0,R4                    ;SET UP REWIND PACKET ADDRESS
9497 076230 005237 002212                INC    FATFLG                   ;BUMP COUNT
9501 076234                                ERRHRD ERRNO,T27RWN,PKTSSR      ;REWIND NOT ACCEPTED
                                076234 104456                                TRAP  CSERHRD
                                076236 001317                                .WORD 719
                                076240 102735                                .WORD T27RWN
                                076242 012136                                .WORD PKTSSR
9502 076244 30$: CKLOOP                    ;LOOP IF SELECTED                TRAP  CSCLP1
                                076244 104406
9503
9504                                     :*****
9505                                     :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9506                                     :
9507                                     :*****
9508
9509
9510 076246 013701 101440                MOV    T27BFR+6,R1              ;PICK UP XSTO
9511 076252 010102                        MOV    R1,R2                    ;SET UP EXPECTED
9512 076254 052702 000002                BIS    #BIT1,R2                 ;SET BOT BIT IN EXPECTED
9513 076260 020102                        CMP    R1,R2                    ;DOES EXP = REC'D
9514 076262 001406                        BEQ    40$                      ;BR, IF EQUAL (OK)
9515 076264 005237 002212                INC    FATFLG                   ;BUMP COUNT
9519 076270                                ERRHRD ERRNO,T27BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
                                076270 104456                                TRAP  CSERHRD
                                076272 001320                                .WORD 720
                                076274 102431                                .WORD T27BOT
                                076276 015564                                .WORD EXPREC
9520 076300 40$: CKLOOP                    ;LOOP IF SELECTED                TRAP  CSCLP1
                                076300 104406
9521 076302 012703 000024                MOV    #20.,R3                  ;STARTING RECORD SIZE
9522 076306 013737 003114 101532        MOV    FREE,T27WB               ;STARTING WRITE BUFFER ADDRESS
9523
9524                                     :*****
9525                                     :WRITE DATA,CVC=1,ACK COMMAND
9526                                     :
9527                                     :*****
9528
9529
9530 076314 012737 140005 101530 65$: MOV    #140005,T27PK3            ;WRITE DATA,CVC=1,ACK COMMAND
9531 076322 012704 101530                MOV    #T27PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
9532 076326 010300                        MOV    R3,R0                    ;SET PATTERN IN CORRECT REGISTER
9533 076330 004737 017512                JSR    PC,FILLMEM               ;FILL MEMORY WITH RECORD SIZE
9534 076334 010337 101536                MOV    R3,T27SZ                ;SET UP RECORD SIZE IN PACKET
9535 076340 010465 000000                MOV    R4,TSDB(R5)             ;ISSUE COMMAND
9536 076344 004737 016340                JSR    PC,WAITF                 ;WAIT FOR SSR TO SET
9537 076350 010501 000002                MOV    TSSR(R5),R1             ;GET TSSR CONTENTS
9538 076354 012702 000200                MOV    #SSR,R2                 ;SET UP EXPECTED
9539 076360 020102                        CMP    R1,R2                    ;ARE THEY EQUAL
9540 076362 001406                        BEQ    80$                      ;BR, IF OK
9541 076364 005237 002212                INC    FATFLG                   ;BUMP COUNT
9545 076370                                ERRHRD ERRNO,WRERR,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA

```

```
076370 104456 TRAP C$ERHRD
076372 001321 .WORD 721
076374 005111 .WORD WRTERR
076376 012136 .WORD PKTSSR
9546 076400 80$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
076400 104406
9547
9548
9549
9550
9551
9552
9553
9554 076402 012737 141005 101530 MOV #141005,T27PK3 ;WRITE DATA RETRY,CVC=1,ACK COMMAND
9555 076410 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
9556 076414 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
9557 076420 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9558 076424 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9559 076430 020102 CMP R1,R2 ;ARE THEY EQUAL
9560 076432 001406 BEQ 90$ ;BR, IF OK
9561 076434 005237 002212 INC FATFLG ;BUMP COUNT
9565 076440 ERRHRD ERRNO,T27WRF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA RETRY
076440 104456 TRAP C$ERHRD
076442 001322 .WORD 722
076444 104336 .WORD T27WRF
076446 012136 .WORD PKTSSR
9566 076450 90$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
076450 104406
9567 076452 005723 TST (R3)+ ;BUMP RECORD SIZE COUNTER
9568 076454 020327 000050 CMP R3,#40. ;AT 40 SIZE YET
9569 076460 001315 BNE 65$ ;BR, IF MORE RECORDS TO WRITE
9570
9571
9572
9573
9574
9575
9576
9577 076462 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
9578 076466 103407 BCS 230$ ;BR, IF NO PROBLEM
9579 076470 010001 MOV R0,R1 ;SAVE TSSR
9580 076472 005237 002212 INC FATFLG ;BUMP COUNT
9584 076476 ERRHRD ERRNO,T27RWN,EXPREC ;REWIND NOT ACCEPTED
076476 104456 TRAP C$ERHRD
076500 001323 .WORD 723
076502 102735 .WORD T27RWN
076504 015564 .WORD EXPREC
9585 076506 230$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
076506 104406
9586
9587
9588
9589
9590
9591
9592
9593 076510 013701 101440 MOV T27BFR+6,R1 ;PICK UP XST0
```

```

9594 076514 010102          MOV      R1,R2          ;SET UP EXPECTED
9595 076516 052702 000002  BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
9596 076522 020102          CMP      R1,R2          ;DOES EXP = REC'D
9597 076524 001406          BEQ      240$          ;BR, IF EQUAL (OK)
9598 076526 005237 002212  INC      FATFLG        ;BUMP COUNT
9602 076532          ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          076532 104456          TRAP      C$ERHRD
          076534 001324          .WORD    724
          076536 102431          .WORD    T27BOT
          076540 015564          .WORD    EXPREC
9603 076542          240$:  CKLOOP          ;LOOP IF SELECTED
          076542 104406          TRAP      C$CLP1
9604 076544 012703 000024  MOV      #20.,R3        ;STARTING RECORD SIZE
9605 076550 013737 003114 101532  MOV      FREE,T27RB     ;STARTING READ BUFFER ADDRESS
9606
9607          ;*****
9608          ;READ DATA,ACK COMMAND
9609          ;*****
9610
9611
9612
9613 076556 012737 100001 101530 265$:  MOV      #100001,T27PK3 ;READ DATA,ACK COMMAND
9614 076564 012704 101530  MOV      #T27PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
9615 076570 010337 101536  MOV      R3,T27SZ       ;SET UP RECORD SIZE IN PACKET
9616 076574 010465 000000  MOV      R4,TSDB(R5)    ;ISSUE COMMAND
9617 076600 004737 016340  JSR      PC,WAITF       ;WAIT FOR SSR TO SET
9618 076604 016501 000002  MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
9619 076610 012702 000200  MOV      #SSR,R2        ;SET UP EXPECTED
9620 076614 020102          CMP      R1,R2          ;ARE THEY EQUAL
9621 076616 001406          BEQ      280$          ;BR, IF OK
9622 076620 005237 002212  INC      FATFLG        ;BUMP COUNT
9626 076624          ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
          076624 104456          TRAP      C$ERHRD
          076626 001325          .WORD    725
          076630 005204          .WORD    RDERR
          076632 012136          .WORD    PKTSSR
9627 076634          280$:  CKLOOP          ;LOOP IF SELECTED
          076634 104406          TRAP      C$CLP1
9628 076636 013702 003114  MOV      FREE,R2        ;GET BUFFER ADDRESS
9629 076642 010304          MOV      R3,R4          ;GET RECCRD SIZE
9630 076644 162704 000024  SUB      #20.,R4        ;POINT BACK TO 1ST RECORD
9631 076650 060204          285$:  ADD      R2,R4          ;POINT TO 1ST LOC IN BUFFER
9632 076652 021403          CMP      (R4),R3        ;DATA WRITTEN = READ
9633 076654 001410          BEQ      290$          ;BR, IF DATA OK (GOOD)
9634 076656 011401          MOV      (R4),R1        ;PICK UP BAD DATA
9635 076660 010302          MOV      R3,R2          ;SET UP EXPECTED
9636 076662 005237 002212  INC      FATFLG        ;BUMP COUNT
9640 076666          ERRHRD  ERRNO,T27DTA,EXPREC ;DATA IN BUFFER NOT CORRECT
          076666 104456          TRAP      C$ERHRD
          076670 001326          .WORD    726
          076672 104416          .WORD    T27DTA
          076674 015564          .WORD    EXPREC
9641 076676          290$:  CKLOOP          ;LOOP IF SELECTED
          076676 104406          TRAP      C$CLP1
9642 076700 005724          TST      (R4)+          ;BUMP TO NEXT ADDRESS
9643 076702 160204          SUB      R2,R4          ;BACK TO RECORD SIZE
9644 076704 020403          CMP      R4,R3          ;AT END OF RECORD YET

```

9645	076706	001360		BNE	285\$:BR, IF MORE DATA TO CHECK
9646	076710	005723		TST	(R3)+	:BUMP RECORD SIZE
9647	076712	020327	000050	CMP	R3,#40.	:DONE YET
9648	076716	001317		BNE	265\$:BR, IF NOT DONE YET (MORE READS)
9649	076720		300\$:	CKLOOP		:LOOP IF SELECTED
	076720	104406				
9650	076722		330\$:			TRAP C\$CLP1
9651	076722			ENDSUB		:>>>>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>>>
	076722					L10125:
	076722	104403				TRAP C\$ESUB
9652	076724	023727	002212 000017	CMP	FATFLG,#15.	:IS ERROR COUNT AT 25
9653	076732	103402		BLO	999\$:BR, IF LESS THAN 25
9654	076734	004737	017272	JSR	PC,CKDROP	:TRY TO DROP THE UNIT
9655	076740		999\$:			

9657
9658
9659
9660
9661
9662
9663
9664
9665
9666
9667
9668

:+
:TEST 7, SUBTEST 4
:VERIFIES THAT A WRITE DATA RETRY COMMAND WITH SWB=1
:TERMINATES PROPERLY AND WRITES CORRECT DATA ON TAPE
:(THE WRITTEN RECORD IS READ AND CHECKED). VARIOUS
:BYTE COUNTS AND DATA PATTERNS ARE USED.

9669 076740

BGNSUB

:>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>>
T7.4:

076740 104402
076740

TRAP CSB SUB

9670 076742 004737 104534

JSR PC,T27REST

:SET COMMAND PACKET

9671 076746 004737 104626

JSR PC,T27RT2

:SET UP OTHER COMMAND PACKET

9672 076752 004737 104670

JSR PC,T27RT3

:SET UP OTHER COMMAND PACKET

9673 076756 012737 176750 101562

MOV #65000.,T27DLY

:SET UP DELAY COUNTER

9674

9675

9676

9677

9678

9679

9680

:*****
:ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
:*****

9681 076764 004737 016064

10\$: JSR PC,SOFINIT

:DO INITIALIZE ON CONTROLLER

9682 076770 103426

BCS 20\$

:BR IF INIT WAS OK

9683 076772 012727 000250

DELAY 250

:DELAY ABOUT .25 SEC

076776 000000

MOV #250,(PC)+

077000 013727 002116

.WORD 0

077004 000000

MOV LSDLY,(PC)+

077006 005367 177772

.WORD 0

077012 001375

DEC -6(PC)

077014 005367 177756

BNE -4

077020 001367

DEC -22(PC)

9684 077022 005337 101562

DEC T27DLY

:BUMP COUNTER

9685 077026 001356

BNE 10\$

:BR, IF COUNTER NOT DONE

9686 077030 005237 002212

INC FATFLG

:BUMP COUNT

9690 077034 010001

MOV RO,R1

:CONTENTS OF TSSR REGISTER

9691 077036 104455

ERRDF ERNO,SFIERR,SFIMSG

:FATAL ERROR TSSR WAS NOT OK

077040 001327

TRAP CSERDF

077042 003650

.WORD 727

077044 012124

.WORD SFIERR

9692 077046 013737 002172 101430

20\$: MOV UNITN,T27DSW

:SET UP UNIT (DRIVE) NUMBER

9693 077054 012704 101410

MOV #T27PACKET,R4

:SUBROUTINE NEEDS PACKET ADDRESS

9694

9695

9696

9697

9698

9699

9700

:*****
:WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
:*****

9701 077060 004737 010752

JSR PC,WRTCHR

:ISSUE WRITE CHARACTERISTICS

9702 077064 103407

BCS 23\$

:BR, IF COMMAND ISSUED OK

```

9703 077066 005237 002212          INC    FATFLG          ;BUMP COUNT
9707 077072 010001                  MOV    R0,R1          ;SAVE CONTENTS OF TSSR
9708 077074                  ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP   CSERHRD
                                .WORD  728
                                .WORD  WRTMSG
                                .WORD  SFIMSG
                                077074 104456
                                077076 001330
                                077100 005054
                                077102 012124
9709 077104          23$:  CKLOOP          ;LOOP IF SELECTED          TRAP   CSCLP1
                                077104 104406
9710
9711          :*****
9712          :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9713          :*****
9714
9715
9716
9717 077106 004737 011104          JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
9718 077112 103411                  BCS    30$            ;BR, IF NO PROBLEM
9719 077114 016501 000002          MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
9720 077120 010004                  MOV    R0,R4          ;GET PACKET ADDRESS
9721 077122 005237 002212          INC    FATFLG          ;BUMP COUNT
9725 077126                  ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP   CSERHRD
                                .WORD  729
                                .WORD  T27RWN
                                .WORD  PKTSSR
                                077126 104456
                                077130 001331
                                077132 102735
                                077134 012136
9726 077136          30$:  CKLOOP          ;LOOP IF SELECTED          TRAP   CSCLP1
                                077136 104406
9727
9728          :*****
9729          :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9730          :*****
9731
9732
9733
9734 077140 013701 101440          MOV    T27BFR+6,R1   ;PICK UP XSTO
9735 077144 010102                  MOV    R1,R2          ;SET UP EXPECTED
9736 077146 052702 000002          BIS    #BIT1,R2      ;SET BOT BIT IN EXPECTED
9737 077152 020102                  CMP    R1,R2          ;DOES EXP = REC'D
9738 077154 001406                  BEQ    40$            ;BR, IF EQUAL (OK)
9739 077156 005237 002212          INC    FATFLG          ;BUMP COUNT
9743 077162                  ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP   CSERHRD
                                .WORD  730
                                .WORD  T27BOT
                                .WORD  EXPREC
                                077162 104456
                                077164 001332
                                077166 102431
                                077170 015564
9744 077172          40$:  CKLOOP          ;LOOP IF SELECTED          TRAP   CSCLP1
                                077172 104406
9745 077174 012703 000024          MOV    #20.,R3       ;STARTING RECORD SIZE
9746 077200 013737 003114 101532    MOV    FREE,T27WB    ;STARTING WRITE BUFFER ADDRESS
9747
9748          :*****
9749          :WRITE DATA,CVC=1,ACK COMMAND
9750          :*****
9751
9752
9753

```

```

9754 077206 012737 140005 101530 65$:  MOV      #140005,T27PK3      ;WRITE DATA,CVC=1,ACK COMMAND
9755 077214 012704 101530      MOV      #T27PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
9756 077220 010300      MOV      R3,R0          ;SET PATTERN IN CORRECT REGISTER
9757 077222 004737 017512      JSR      PC,FILLMEM     ;FILL MEMORY WITH RECORD SIZE
9758 077226 010337 101536      MOV      R3,T27SZ      ;SET UP RECORD SIZE IN PACKET
9759 077232 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
9760 077236 004737 016340      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
9761 077242 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
9762 077246 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
9763 077252 020102      CMP      R1,R2        ;ARE THEY EQUAL
9764 077254 001406      BEQ      80$          ;BR, IF OK
9765 077256 005237 002212      INC      FATFLG       ;BUMP COUNT
9769 077262      ERRHRD  ERRNO,WRterr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP  CSERHRD
                                .WORD 731
                                .WORD WRterr
                                .WORD PKTSSR
9770 077272      80$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP  C$CLP1
                                .WORD 104406
9771 077272 104406
9772
9773 :*****
9774 :WRITE DATA RETRY,ACK,SWB=1 COMMAND
9775 :*****
9776
9777
9778 077274 012737 111005 101530      MOV      #111005,T27PK3 ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
9779 077302 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
9780 077306 004737 016340      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
9781 077312 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
9782 077316 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
9783 077322 020102      CMP      R1,R2        ;ARE THEY EQUAL
9784 077324 001406      BEQ      90$          ;BR, IF OK
9785 077326 005237 002212      INC      FATFLG       ;BUMP COUNT
9789 077332      ERRHRD  ERRNO,T27WRF,EXPREC ;TSSR INCORRECT AFTER WRITE DATA RETRY
                                TRAP  CSERHRD
                                .WORD 732
                                .WORD T27WRF
                                .WORD EXPREC
9790 077342      90$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP  C$CLP1
                                .WORD 104406
9791 077344 005723      TST      (R3)+        ;BUMP RECORD SIZE COUNTER
9792 077346 020327 000050      CMP      R3,#40.     ;AT 40 SIZE YET
9793 077352 001315      BNE      65$          ;BR, IF MORE RECORDS TO WRITE
9794
9795 :*****
9796 :ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9797 :*****
9798
9799
9800
9801 077354 004737 011104      JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
9802 077360 103411      BCS      230$        ;BR, IF NO PROBLEM
9803 077362 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
9804 077366 010004      MOV      R0,R4        ;GET PACKET ADDRESS
9805 077370 005237 002212      INC      FATFLG       ;BUMP COUNT
9809 077374      ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED

```

```

077374 104456 TRAP C$ERHRD
077376 001335 .WORD 733
077400 102735 .WORD T27RWN
077402 012136 .WORD PKTSSR
9810 077404 230$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
077404 104406
9811
9812 ;*****
9813 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9814 ;*****
9815
9816
9817
9818 077406 013701 101440 MOV T27BFR+6,R1 ;PICK UP XSTO
9819 077412 010102 MOV R1,R2 ;SET UP EXPECTED
9820 077414 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
9821 077420 020102 CMP R1,R2 ;DOES EXP = REC'D
9822 077422 001406 BEQ 240$ ;BR, IF EQUAL (OK)
9823 077424 005237 002212 INC FATFLG ;BUMP COUNT
9827 077430 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
077430 104456 TRAP C$ERHRD
077432 001336 .WORD 734
077434 102431 .WORD T27BOT
077436 015564 .WORD EXPREC
9828 077440 240$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
077440 104406
9829 077442 012703 000024 MOV #20.,R3 ;STARTING RECORD SIZE
9830 077446 013737 003114 101532 MOV FREE,T27RB ;STARTING READ BUFFER ADDRESS
9831
9832 ;*****
9833 ;READ DATA,ACK COMMAND
9834 ;*****
9835
9836
9837
9838 077454 012737 100001 101530 265$: MOV #100001,T27PK3 ;READ DATA,ACK COMMAND
9839 077462 012704 101530 MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
9840 077466 010337 101536 MOV R3,T27SZ ;SET UP RECORD SIZE IN PACKET
9841 077472 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
9842 077476 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
9843 077502 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9844 077506 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9845 077512 020102 CMP R1,R2 ;ARE THEY EQUAL
9846 077514 001406 BEQ 280$ ;BR, IF OK
9847 077516 005237 002212 INC FATFLG ;BUMP COUNT
9851 077522 ERRHRD ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
077522 104456 TRAP C$ERHRD
077524 001337 .WORD 735
077526 005204 .WORD RDERR
077530 012136 .WORD PKTSSR
9852 077532 280$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
077532 104406
9853 077534 013702 003114 MOV FREE,R2 ;GET BUFFER ADDRESS
9854 077540 010304 MOV R3,R4 ;GET RECORD SIZE
9855 077542 162704 000024 SUB #20.,R4 ;POINT BACK TO 1ST RECORD
9856 077546 060204 285$: ADD R2,R4 ;POINT TO 1ST LOC IN BUFFER
9857 077550 000303 SWAB R3 ;SWAP BYTES SWB=1 ETC.

```



```

077714 000000
077716 013727 002116
077722 000000
077724 005367 177772
077730 001375
077732 005367 177756
077736 001367
9937 077740 005337 101562
9938 077744 001356
9939 077746 005237 002212
9943 077752 010001
9944 077754
077754 104455
077756 001341
077760 003650
077762 012124
9945 077764 013737 002172 101430 20$: MOV UNITN,T27DSW ;SET UP UNIT NUMBER
9946
9947 077772 012704 101410 MOV #T27PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
9948
9949
9950
9951
9952
9953
9954
9955 077776 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
9956 100002 103407 BCS 23$ ;BR, IF COMMAND ISSUED OK
9957 100004 005237 002212 INC FATFLG ;BUMP COUNT
9961 100010 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
9962 100012
100012 104456 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
100014 001342 TRAP C$ERHRD
100016 005054 .WORD 738
100020 012124 .WORD WRTMSG
9963 100022 23$: CKLOOP ;LOOP IF SELECTED .WORD SFIMSG
100022 104406 TRAP C$CLP1
9964
9965
9966
9967
9968
9969
9970
9971 100024 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
9972 100030 103411 BCS 30$ ;BR, IF NO PROBLEM
9973 100032 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9974 100036 010004 MOV R0,R4 ;GET PACKET ADDRESS
9975 100040 005237 002212 INC FATFLG ;BUMP COUNT
9979 100044
100044 104456 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
100046 001343 TRAP C$ERHRD
100050 102735 .WORD 739
100052 012136 .WORD T27RWN
9980 100054 30$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
100054 104406 TRAP C$CLP1
9981

```

```

9982
9983
9984
9985
9986
9987
9988 100056 013701 101440      MOV      T27BFR+6,R1      :PICK UP XSTO
9989 100062 010102      MOV      R1,R2           :SET UP EXPECTED
9990 100064 052702 000002      BIS      #BIT1,R2        :SET BOT BIT IN EXPECTED
9991 100070 020102      CMP      R1,R2           :DOES EXP = REC'D
9992 100072 001406      BEQ      40$             :BR, IF EQUAL (OK)
9993 100074 005237 002212      INC      FATFLG          :BUMP COUNT
9997 100100      ERRHRD  ERRNO,T27BOT,EXPREC :TAPE NOT AT BOT AFTER REWIND
      100100 104456      TRAP     C$ERHRD
      100102 001344      .WORD   740
      100104 102431      .WORD   T27BOT
      100106 015564      .WORD   EXPREC
9998 100110      40$:   CKLOOP           :LOOP IF SELECTED
      100110 104406      TRAP     C$CLP1
9999 100112 012703 000144      MOV      #100.,R3        :NUMBER OF RECORDS TO BE WRITTEN
10000 100116 013737 003114 101532      MOV      FREE,T27WB      :STARTING WRITE BUFFER ADDRESS
10001
10002
10003
10004
10005
10006
10007
10008 100124 012737 140005 101530 65$:   MOV      #140005,T27PK3   :WRITE DATA,ACK,CVC=1 COMMAND
10009 100132 012704 101530      MOV      #T27PK3,R4      :SET UP R4 WITH PACKET ADDRESS
10010 100136 012737 000024 101536      MOV      #20.,T27SZ      :SET UP RECORD SIZE IN PACKET
10011 100144 010465 000000      MOV      R4,T$DB(R5)     :ISSUE COMMAND
10012 100150 004737 016340      JSR      PC,WAITF        :WAIT FOR SSR TO SET
10013 100154 016501 000002      MOV      T$SR(R5),R1     :GET T$SR CONTENTS
10014 100160 012702 000200      MOV      #SSR,R2         :SET UP EXPECTED
10015 100164 020102      CMP      R1,R2           :ARE THEY EQUAL
10016 100166 001406      BEQ      70$             :BR, IF OK
10017 100170 005237 002212      INC      FATFLG          :BUMP COUNT
10021 100174      ERRHRD  ERRNO,WRERR,PKTSSR :T$SR INCORRECT AFTER WRITE DATA
      100174 104456      TRAP     C$ERHRD
      100176 001345      .WORD   741
      100200 005111      .WORD   WRERR
      100202 012136      .WORD   PKTSSR
10022 100204      70$:   CKLOOP           :LOOP IF SELECTED
      100204 104406      TRAP     C$CLP1
10023 100206 005303      DEC      R3              :DEC RECORD COUNTER
10024 100210 001345      BNE     65$             :BR, IF MORE RECORDS TO WRITE
10025
10026
10027
10028
10029
10030
10031
10032 100212 004737 011104      JSR      PC,REWIND       :CALL TAPE REWIND COMMAND
10033 100216 103411      BCS     130$            :BR, IF NO PROBLEM
10034 100220 016501 000002      MOV      T$SR(R5),R1     :GET T$SR CONTENTS

```

```

10035 100224 010004          MOV    R0,R4          ;GET PACKET ADDRESS
10036 100226 005237 002212  INC    FATFLG        ;BUMP COUNT
10040 100232          ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
10041 100242          130$:  CKLOOP          ;LOOP IF SELECTED
10042 100242 104406          TRAP   C$SERHRD
10043          .WORD   742
10044          .WORD   T27RWN
10045          .WORD   PKTSSR
10046          .WORD   C$CLP1
10047          :*****
10048          :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
10049          :*****
10049 100244 013701 101440          MOV    T27BFR+6,R1   ;PICK UP XST0
10050 100250 010102          MOV    R1,R2         ;SET UP EXPECTED
10051 100252 052702 000002          BIS    #BIT1,R2      ;SET BOT BIT IN EXPECTED
10052 100256 020102          CMP    R1,R2         ;DOES EXP = REC'D
10053 100260 001406          BEQ    140$          ;BR, IF EQUAL (OK)
10054 100262 005237 002212  INC    FATFLG        ;BUMP COUNT
10058 100266          ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
10059 100276          140$:  CKLOOP          ;LOOP IF SELECTED
10060 100300 012704 101530          TRAP   C$CLP1
10061 100304 012737 000010 101532  MOV    #T27PK3,R4    ;SET UP PACKET ADDRESS
10062          MOV    #10,T27RB   ;SET UP RECORDS TO SPACE OVER
10063          :*****
10064          :ACK,CVC=1,SPACE FORWARD COMMAND
10065          :*****
10069 100312 012737 140010 101530  MOV    #140010,T27PK3 ;ACK,CVC=1,SPACE FORWARD COMMAND
10070 100320 010465 000000          MOV    R4,TSDB(R5)  ;ISSUE COMMAND
10071 100324 005237 101556          152$:  INC    T27CNT   ;BUMP TIMER
10072 100330          DELAY  1             ;DELAY ABOUT 100US
10073 100330 012727 000001          MOV    #1,(PC)+
10074 100334 000000          .WORD  0
10075 100336 013727 002116          MOV    L$DLY,(PC)+
10076 100342 000000          .WORD  0
10077 100344 005367 177772          DEC    -6(PC)
10078 100350 001375          BNE    -4
10079 100352 005367 177756          DEC    -22(PC)
10080 100356 001367          BNE    -20
10073 100360 016501 000002          MOV    TSSR(R5),R1  ;GET TSSR
10074 100364 032701 000200          BIT    #BIT7,R1     ;CHECK FOR TSSR'S SSR SET
10075 100370 001755          BEQ    152$         ;KEEP COUNTING UNTIL SET
10076 100372 016501 000002          MOV    TSSR(R5),R1  ;GET STATUS FROM TSSR
10077 100376 012702 000200          MOV    #SSR,R2      ;SET UP EXPECTED
10078 100402 020201          CMP    R2,R1        ;WAS EVERYTHING OK
10079 100404 001406          BEQ    160$         ;BR, IF ALL IS WELL

```

```
10080 100406 005237 002212          INC    FATFLG          ;BUMP COUNT
10084 100412          ERRHRD  ERRNO,T27SCF,PKTSSR ;SPACE FORWARD DIDN'T WORK OUT
      100412 104456          TRAP    C$ERHRD
      100414 001350          .WORD  744
      100416 104177          .WORD  T27SCF
      100420 012136          .WORD  PKTSSR
10085 100422          160$:  CKLOOP          ;LOOP IF SELECTED
      100422 104406          TRAP    C$CLP1
10086
10087
10088
10089
10090
10091
10092
10093 100424 004737 011104          JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
10094 100430 004737 016426          JSR    PC,CHKTSSR    ;SEE HOW TSSR IS
10095 100434 103407          BCS    170$          ;BR, IF NO PROBLEM
10096 100436 010001          MOV    R0,R1         ;SAVE TSSR
10097 100440 005237 002212          INC    FATFLG          ;BUMP COUNT
10101 100444          ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
      100444 104456          TRAP    C$ERHRD
      100446 001351          .WORD  745
      100450 102735          .WORD  T27RWN
      100452 012136          .WORD  PKTSSR
10102 100454          170$:  CKLOOP          ;LOOP IF SELECTED
      100454 104406          TRAP    C$CLP1
10103
10104
10105
10106
10107
10108
10109
10110 100456 013701 101440          MOV    T27BFR+6,R1   ;PICK UP XST0
10111 100462 010102          MOV    R1,R2         ;SET UP EXPECTED
10112 100464 052702 000002          BIS    #BIT1,R2     ;SET BOT BIT IN EXPECTED
10113 100470 020102          CMP    R1,R2         ;DOES EXP = REC'D
10114 100472 001406          BEQ    175$          ;BR, IF EQUAL (OK)
10115 100474 005237 002212          INC    FATFLG          ;BUMP COUNT
10119 100500          ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      100500 104456          TRAP    C$ERHRD
      100502 001352          .WORD  746
      100504 102431          .WORD  T27BOT
      100506 015564          .WORD  EXPREC
10120 100510          175$:  CKLOOP          ;LOOP IF SELECTED
      100510 104406          TRAP    C$CLP1
10121 100512 012703 000144          MOV    #100.,R3     ;STARTING RECORD SIZE
10122 100516 013737 003114 101532 177$:  MOV    FREE,T27WB    ;STARTING WRITE BUFFER ADDRESS
10123
10124
10125
10126
10127
10128
10129
10130 100524 012737 140005 101530          MOV    #140005,T27PK3 ;WRITE DATA,CVC=1,ACK COMMAND
```

```
10131 100532 012704 101530      MOV      #T27PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
10132 100536 012737 000024 101536      MOV      #20.,T27SZ     ;SET UP RECORD SIZE IN PACKET
10133 100544 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
10134 100550 004737 016340      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
10135 100554 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
10136 100560 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED
10137 100564 020102      CMP      R1,R2         ;ARE THEY EQUAL
10138 100566 001406      BEQ      180$          ;BR, IF OK
10139 100570 005237 002212      INC      FATFLG        ;BUMP COUNT
10143 100574      ERRHRD  ERRNO,WRterr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      100574 104456      TRAP    C$ERHRD
      100576 001353      .WORD  747
      100600 005111      .WORD  WRterr
      100602 012136      .WORD  PKTSSR
10144 100604      180$:  CKLOOP          ;LOOP IF SELECTED      TRAP    C$CLP1
      100604 104406
10145 100606 005303      DEC      R3            ;COUNT NUMBER OF RECORDS
10146 100610 001342      BNE      177$          ;BR, IF MORE RECORDS TO WRITE
10147
10148      ;*****
10149      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10150      ;*****
10151
10152
10153
10154 100612 004737 011104      JSR      PC,REWIND      ;ISSUE REWIND
10155 100616 103411      BCS      182$          ;BR, IF ALL IS WELL
10156 100620 010004      MOV      R0,R4         ;GET PACKET ADDRESS
10157 100622 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
10158 100626 005237 002212      INC      FATFLG        ;BUMP COUNT
10162 100632      ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND FAILED
      100632 104456      TRAP    C$ERHRD
      100634 001354      .WORD  748
      100636 102735      .WORD  T27RWN
      100640 012136      .WORD  PKTSSR
10163 100642      182$:  CKLOOP          ;SELECT LOOP MAYBE    TRAP    C$CLP1
      100642 104406
10164
10165      ;*****
10166      ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
10167      ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
10168      ;*****
10169
10170
10171
10172 100644 012703 000001      MOV      #1.,R3        ;SPACE 1 RECORD FORWARD
10173 100650 004737 010556      JSR      PC,SPACE      ;ISSUE SPACE COMMAND
10174 100654 103411      BCS      185$          ;BR, IF COMMAND OK
10175 100656 010004      MOV      R0,R4         ;GET PACKET ADDRESS
10176 100660 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR STATUS
10177 100664 005237 002212      INC      FATFLG        ;BUMP COUNT
10181 100670      ERRHRD  ERRNO,T27SCF,PKTSSR ;SPACE FORWARD COMMAND FAILED
      100670 104456      TRAP    C$ERHRD
      100672 001355      .WORD  749
      100674 104177      .WORD  T27SCF
      100676 012136      .WORD  PKTSSR
10182 100700      185$:  CKLOOP          ;LOOP IF SELECTED
```

```
10183 100700 104406
10184 100702 012703 000144
10185 100706 013737 003114 101532
10186
10187
10188
10189
10190
10191
10192 100714 012737 101005 101530 190$: MOV #101005,T27PK3 ;WRITE DATA RETRY,ACK COMMAND
10193 100722 012704 101530 MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
10194 100726 012737 000024 101536 MOV #20,T27SZ ;SET UP RECORD SIZE IN PACKET
10195 100734 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
10196 100740 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
10197 100744 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10198 100750 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
10199 100754 020102 CMP R1,R2 ;ARE THEY EQUAL
10200 100756 001406 BEQ 200$ ;BR, IF OK
10201 100760 005237 002212 INC FATFLG ;BUMP COUNT
10205 100764 ERRHRD ERRNO,T27WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
100764 104456 TRAP C$ERHRD
100766 001356 .WORD 750
100770 103271 .WORD T27WDC
100772 012136 .WORD PKTSSR
10206 100774 200$: CKLOOP ;LOOP IF SELECTED
100774 104406 TRAP C$CLP1
10207 100776 013737 003114 101532 MOV FREE,T27WB ;STARTING WRITE BUFFER ADDRESS
10208
10209
10210
10211
10212
10213
10214
10215 101004 012737 140005 101530 MOV #140005,T27PK3 ;WRITE DATA,CVC=1,ACK COMMAND
10216 101012 012704 101530 MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
10217 101016 012737 000024 101536 MOV #20,T27SZ ;SET UP RECORD SIZE IN PACKET
10218 101024 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
10219 101030 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
10220 101034 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10221 101040 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
10222 101044 020102 CMP R1,R2 ;ARE THEY EQUAL
10223 101046 001406 BEQ 210$ ;BR, IF OK
10224 101050 005237 002212 INC FATFLG ;BUMP COUNT
10228 101054 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
101054 104456 TRAP C$ERHRD
101056 001357 .WORD 751
101060 005111 .WORD WRERR
101062 012136 .WORD PKTSSR
10229 101064 210$: CKLOOP ;LOOP IF SELECTED
101064 104406 TRAP C$CLP1
10230 101066 005303 DEC R3 ;BUMP DOWN RECORD COUNTER
10231 101070 001311 BNE 190$ ;BR, IF MORE RECORDS TO WRITE RETRY
10232
10233
10234
```



```
10235      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10236      :
10237      :*****
10238      :
10239 101072 004737 011104      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
10240 101076 103411      BCS      230$           ;BR, IF NO PROBLEM
10241 101100 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
10242 101104 010004      MOV      R0,R4          ;GET PACKET ADDRESS
10243 101106 005237 002212      INC      FATFLG         ;BUMP COUNT
10247 101112      ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
10247 101112 104456      TRAP    C$ERHRD
10247 101114 001360      .WORD  752
10247 101116 102735      .WORD  T27RWN
10247 101120 012136      .WORD  PKTSSR
10248 101122      230$:  CKLOOP           ;LOOP IF SELECTED
10248 101122 104406      TRAP    C$CLP1
10249      :
10250      :*****
10251      :
10252      :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
10253      :
10254      :*****
10255      :
10256 101124 013701 101440      MOV      T27BFR+6,R1    ;PICK UP XST0
10257 101130 010102      MOV      R1,R2          ;SET UP EXPECTED
10258 101132 052702 000002      BIS      #BIT1,R2       ;SET BOT BIT IN EXPECTED
10259 101136 020102      CMP      R1,R2          ;DOES EXP = REC'D
10260 101140 001406      BEQ      240$           ;BR, IF EQUAL (OK)
10261 101142 005237 002212      INC      FATFLG         ;BUMP COUNT
10265 101146      ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
10265 101146 104456      TRAP    C$ERHRD
10265 101150 001361      .WORD  753
10265 101152 102431      .WORD  T27BOT
10265 101154 015564      .WORD  EXPREC
10266 101156      240$:  CKLOOP           ;LOOP IF SELECTED
10266 101156 104406      TRAP    C$CLP1
10267 101160 012704 101530      MOV      #T27PK3,R4     ;SET UP PACKET ADDRESS
10268 101164 012737 000010 101532  MOV      #10,T27RB      ;SET UP RECORDS TO SPACE OVER
10269      :
10270      :*****
10271      :
10272      :ACK,CVC=1,SPACE FORWARD COMMAND
10273      :
10274      :*****
10275      :
10276 101172 012737 140010 101530  MOV      #140010,T27PK3 ;ACK,CVC=1,SPACE FORWARD COMMAND
10277 101200 010465 000000      250$:  MOV      R4,TSDB(R5) ;ISSUE COMMAND
10278 101204 005237 101560      252$:  INC      T27CNU    ;BUMP TIMER
10279 101210      DELAY  1              ;DELAY ABOUT 100US
10279 101210 012727 000001      MOV      #1,(PC)+
10279 101214 000000      .WORD  0
10279 101216 013727 002116      MOV      L$DLY,(PC)+
10279 101222 000000      .WORD  0
10279 101224 005367 177772      DEC      -6(PC)
10279 101230 001375      BNE     #-4
10279 101232 005367 177756      DEC      -22(PC)
10279 101236 001367      BNE     -20
```


10322
 10323
 10324
 10328 101410
 10329 101410 100004
 10330 101412 101420
 10331 101414 000000
 10332 101416 000012
 10333 101420
 10334 101420 101432
 10335 101422 000000
 10336 101424 000024
 10337 101426 000000
 10338 101430 000000
 10339 101432
 10340
 10341
 10342
 10344 101520
 10346 101520 100006
 10347 101520 101540
 10348 101522 000000
 10349 101524 000006
 10351
 10355 101530
 10356 101530 100005
 10357 101532
 10358 101532 003114
 10359 101534 000000
 10360 101536 000000
 10361
 10362
 10363
 10364
 10365 101540
 10366 101540 010
 10367 101541 200
 10368 101542 000000
 10369 101544 000000
 10370
 10371
 10372
 10373
 10374
 10375 101546 100205
 10376 101550 100605
 10377 101552 102205
 10378 101554 177777
 10379
 10380
 10381 101556 000000
 10382 101560 000000
 10383 101562 000000
 10384

```

;+
;LOCAL STORAGE FOR THIS TEST
;
T27PACKET:
      .WORD 100004 ;COMMAND PACKET FOR TEST
      .WORD T27DATA ;WRITE CHARACTERISTICS COMMAND, WITH , ACK
      .WORD 0 ;ADDRESS OF CHARACTERISTICS BLOCK
      .WORD 10. ;STARTING VALUE OF BLOCK SIZE
T27DATA:
      .WORD T27BFR ;CHARACTERISTICS DATA BLOCK
      .WORD 0 ;ADDRESS OF MESSAGE BUFFER
      .WORD 20. ;LENGTH OF MESSAGE BUFFER
      .WORD 0
T27DSW: .WORD 0 ;SELECT DRIVE 0
T27BFR: .BLKW 25. ;MESSAGE BUFFER
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
      .=<.+10>&177770
T27PK2:
      .WORD 100006 ;WRITE SUB SYS MEM COMMAND, AND ACK
      .WORD T27BF2 ;ADDRESS OF SELECT BLOCK DATA
      .WORD 0
      .WORD 6. ;SIZE OF DATA PACKET
T27PK3:
      .WORD 100005 ;REREAD COMMAND, AND ACK
T27RB:
T27WB: .WORD FREE ;ADDRESS OF WRITE BUFFER
      .WORD 0
T27SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
      .EVEN
;
;
T27BF2:
T27BS0: .BYTE 10 ;BSELO AREA
T27BS1: .BYTE 200 ;BSEL1 AREA
T27S2: .WORD 0 ;SEL 2 AREA
T27S3: .WORD 0 ;DATA AREA
;
;
      .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T27RN: .WORD 100205 ;REREAD DATA (NEXT)
T27WDR: .WORD 100605 ;REREAD DATA RETRY
T27CON: .WORD 102205 ;WRITE CONTINOUS
      .WORD 177777 ;END OF DATA
;
T27CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T27CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T27DLY: .WORD 0 ;DELAY COUNTER

```

```

10386
10387
10388
10389
10390
10391
10392 101564 124 141 160 T27WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
10393 101652 124 123 123 T27RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
10394 101721 122 105 122 T27RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
10395 102016 120 117 123 T27SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
10396 102100 122 111 102 T27LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
10397 102150 124 123 123 T27WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
10398 102225 111 154 154 T27LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
10399 102306 122 105 122 T27SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
10400 102342 124 123 123 T27WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command, At BOT'
10401 102431 124 141 160 T27BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XSTO)'
10402 102524 127 122 111 T27TIM: .ASCIZ 'WRITE DATA RETRY'S Erase Tape Not Long Enough'
10403 102601 122 105 122 T27EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
10404 102660 124 123 123 T27TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
10405 102735 122 145 167 T27RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
10406 103004 122 101 115 T27RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
10407 103057 124 123 123 T27AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
10408 103126 104 162 151 T27OFL: .ASCIZ 'Drive 7 Select Failed To Set 'OFL' In TSSR'
10409 103201 124 123 123 T27WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
10410 103271 124 123 123 T27WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
10411 103344 103 126 103 T27VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
10412 103417 124 123 102 T27BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
10413 103472 127 122 111 T27WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
10414 103561 122 145 141 T27LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XSTO'
10415 103643 122 145 141 T27LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XSTO'
10416 103725 122 145 163 T27PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
10417 104013 122 145 141 T27TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
10418 104101 127 122 111 T27NEF: .ASCIZ 'WRITE DATA RETRY, At First Record, failed To Set RIB Bit XST3'
10419 104177 124 123 123 T27SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
10420 104254 124 123 123 T27TSA: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
10421 104336 124 123 123 T27WRF: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command'
10422 104416 104 141 164 T27DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
10423 104513 127 162 151 TST27ID: .ASCIZ 'Write Data Retry'
10424
10425
10426
10427
10428
10429
10430
10431
10432 104534
10433 104534
10434 104540 012701 101410
10435 104544 012721 100004
10436 104550 012721 101420
10437 104554 005021
10438 104556 012721 000012
10439 104562 012721 101432
10440 104566 005021
10441 104570 012721 000024
10442 104574 005021

```

```

: +
: LOCAL TEXT MESSAGES FOR TEST
: -

```

```

: +
: ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
: WRITE SUBSYSTEM MEMORY COMMAND
: -

```

```

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)+

```

10443	104576	012711	000000		MOV	#0,(R1)		:SELECT DRIVE ZERO
10444	104602	012702	000030		MOV	#24,R2		:NUMBER OF LOCATIONS TO BE CLEARED
10445	104606	012762	177777	101432 64\$:	MOV	#177777,T27BFR(R2)		:ALL ONES TO MESSAGE BUFFER
10446	104614	005742			TST	-(R2)		:NEXT LOCATION
10447	104616	022702	000000		CMP	#0,R2		:AT END OF LOOP YET
10448	104622	001371			BNE	64\$:KEEP GOING UNTIL DONE
10449	104624	000207			RTS	PC		:RETURN
10450								
10451								
10452	104626				T27RT2:			
10453	104626				SAVREG			:SAVE THE REGISTERS
10454	104632	012701	101520		MOV	#T27PK2,R1		:START OF THE PACKET
10455	104636	012721	100006		MOV	#100006,(R1)+		:WRITE SUBSYSTEM MEM. WITH ACK.
10456	104642	012721	101540		MOV	#T27BF2,(R1)+		:ADDRESS OF DATA BLOCK
10457	104646	005021			CLR	(R1)+		:EXTENDED ADDRESS
10458	104650	012721	000006		MOV	#6,(R1)+		:SIZE OF DATA BLOCK IN BYTES
10459	104654	005021			CLR	(R1)+		
10460	104656	012701	101540		MOV	#T27BF2,R1		:POINT TO DATA SEL AREA
10461	104662	005021			CLR	(R1)+		
10462	104664	005011			CLR	(R1)		
10463	104666	000207			RTS	PC		:RETURN
10464	104670				T27RT3:			
10465	104670				SAVREG			:SAVE REGISTERS
10466	104674	012701	101530		MOV	#T27PK3,R1		:SET UP POINTER ADDRESS
10467	104700	005021			CLR	(R1)+		:COMMAND SPACE
10468	104702	005021			CLR	(R1)+		:ADDRESS OF DATA BLOCK
10469	104704	005021			CLR	(R1)+		:EXTENDED ADDRESS
10470	104706	005011			CLR	(R1)		:SIZE OF DATA TRANSFER BLOCK
10471	104710	000207			RTS	PC		:RETURN
10472	104712				ENDTST			
	104712							
	104712	104401						

L10122: TRAP C\$ETST

10531	105030	103407			BCS	24\$:BR, IF COMMAND ISSUED OK
10532	105032	005237	002212		INC	FATFLG		:BUMP COUNT
10536	105036	010001			MOV	R0,R1		:SAVE CONTENTS OF TSSR
10537	105040				ERRHRD	ERRNO,WRTMSG,SFIMSG		:WRITE CHARACTERISTIC FAILED
	105040	104456						TRAP C\$ERHRD
	105042	001442						.WORD 802
	105044	005054						.WORD WRTMSG
	105046	012124						.WORD SFIMSG
10538	105050				24\$:	CKLOOP		
	105050	104406						TRAP C\$CLP1
10539	105052	005737	002216		TST	EXTFEA		:CHECK FOR EXTENDED FEATURES SW SWITCH
10540	105056	001044			BNE	50\$:BR IF SWITCH IS ON
10541								
10542	105060	112737	000200	110231	MOVB	#200,T28BS1		:WRITE MISCELLANEOUS CONT/READ STATUS
10543	105066	112737	000010	110230	MOVB	#10,T28BS0		:FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
10544	105074	012704	110210		MOV	#T28PK2,R4		:WRITE SUBSYS MEM PACKET
10545	105100	010465	000000		MOV	R4,TSDB(R5)		:ISSUE COMMAND
10546	105104	004737	016426		JSR	PC,CHKTSSR		:WAIT FOR SSR
10547	105110	103407			BCS	30\$:BR, IF NO ERROR
10548	105112	010001			MOV	R0,R1		:ERROR, SAVE TSSR
10549	105114	005237	002212		INC	FATFLG		:BUMP COUNT
10553	105120				ERRHRD	ERRNO,T28SSR,PKTSSR		:TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
	105120	104456						TRAP C\$ERHRD
	105122	001443						.WORD 803
	105124	110725						.WORD T28SSR
	105126	012136						.WORD PKTSSR
10554	105130				30\$:	CKLOOP		:LOOP IF SELECTED
	105130	104406						TRAP C\$CLP1
10555	105132	012704	110100		MOV	#T28PACKET,R4		:SUBROUTINE NEEDS PACKET ADDRESS
10556	105136	012737	000007	110120	MOV	#7,T28DSW		:SELECT DRIVE 7
10557	105144	004737	010752		JSR	PC,WRTCHR		:ISSUE WRITE CHARACTERISTICS
10558	105150	103407			BCS	50\$:BR, IF COMMAND ISSUED OK
10559	105152	005237	002212		INC	FATFLG		:BUMP COUNT
10563	105156	010001			MOV	R0,R1		:SAVE CONTENTS OF TSSR
10564	105160				ERRHRD	ERRNO,WRTMSG,SFIMSG		:WRITE CHARACTERISTIC FAILED
	105160	104456						TRAP C\$ERHRD
	105162	001444						.WORD 804
	105164	005054						.WORD WRTMSG
	105166	012124						.WORD SFIMSG
10565	105170				50\$:	CKLOOP		:SCOPE LOOP
	105170	104406						TRAP C\$CLP1
10566	105172	016501	000002		MOV	TSSR(R5),R1		:GET TSSR CONTENTS
10567	105176	032701	000100		BIT	#OFL,R1		:CHECK FOR THE OFFLINE BIT SET
10568	105202	001006			BNE	60\$:BR, IF OFFLINE (GOOD)
10569	105204	005237	002212		INC	FATFLG		:BUMP COUNT
10573	105210				ERRDF	ERRNO,T28OFL,SFIMSG		:OFF LINE SHOULD HAVE BEEN SET (BAD)
	105210	104455						TRAP C\$ERDF
	105212	001445						.WORD 805
	105214	111260						.WORD T28OFL
	105216	012124						.WORD SFIMSG
10574	105220				60\$:	CKLOOP		:LOOP IF SELECTED
	105220	104406						TRAP C\$CLP1
10575	105222	012703	110246		MOV	#T28RN,R3		:POINTER FOR COMMANDS
10576	105226	011337	110220		MOV	(R3),T28PK3		:TAPE READ COMMAND IN PLACE
10577	105232	012704	110220		MOV	#T28PK3,R4		:R4 = POINTER TO PACKET
10578	105236	010465	000000		MOV	R4,TSDB(R5)		:ISSUE COMMAND
10579	105242	004737	016340		JSR	PC,WAITF		:WAIT FOR SSR TO SET

10674
10675
10676
10677
10678
10679
10680
10681
10682
10683
10684
10685
10686
10687
10688
10689
10690
10691
10692
10693
10694
10695
10696
10697
10698
10699
10700
10701
10702
10703
10704
10705
10706
10707
10708
10709
10710
10711
10712
10713
10714
10715
10716
10717
10718
10719
10720
10721
10722
10723
10724
10725
10726
10727
10728
10729
10730

:+
:TEST 8, SUBTEST 3
:VERIFIES THAT WRITE TAPE MARK COMMANDS OPERATE
:PROPERLY, AND THAT READ COMMANDS SUBSEQUENTLY ISSUED
:TO DETECT THE WRITTEN TAPE MARKS TERMINATE WITH TAPE
:STATUS ALERT WITH THE TAPE MARK DETECTED (TMK) STATUS
:BIT SET. THE FOLLOWING SEQUENCE IS EXECUTED.
:1. THE CONTROLLER IS INITIALIZED AND TAPE REWOUND.
: THIS SETS THE VOLUME CHECK (VCK) STATUS BIT.
:2. A WRITE TAPE MARK COMMAND WITH CVC=1 IS ISSUED
: AND PROPER TERMINATION AND STATUS IS VERIFIED
: (I.E. VCK=0 AND TMK=1).
:3. SEVERAL MORE WRITE TAPE MARK COMMANDS, THESE WITH
: CVC=0 ARE ISSUED AND PROPER TERMINATION (NORMAL)
: AND STATUS (TMK) VERIFIED.
:4. A READ REVERSE COMMAND IS ISSUED AND PROPER
: TERMINATION (TAPE STATUS ALERT) AND STATUS (TMK)
: VERIFIED. IT IS ALSO VERIFIED THAT NO DATA IS
: TRANSFERRED INTO MEMORY.
:5. A SPACE RECORDS REVERSE COMMAND IS ISSUED AND
: PROPER TERMINATION (TAPE STATUS ALERT) AND STATUS
: (TMK) VERIFIED.
:6. THE TAPE IS REWOUND AND A READ FORWARD COMMAND IS
: ISSUED AND PROPER TERMINATION (TAPE STATUS ALERT)
: AND STATUS (TMK) VERIFIED. IT IS ALSO VERIFIED
: THAT NO DATA IS TRANSFERRED INTO MEMORY.
:7. A SPACE RECORDS REVERSE COMMAND THAT CONTAINS A
: RECORD COUNT GREATER THAN 1 IS ISSUED AND IT IS
: VERIFIED THAT TAPE STATUS ALERT TERMINATION
: OCCURED, TMK=1 AND THAT RBPCR (RESIDUAL
: BYTE/RECORD COUNTER) CONTAINS THE PROPER NONZERO
: VALUE. THIS OPERATION VERIFIES THAT DETECTION OF
: THE TAPE MARK CAUSES THE SPACE RECORDS OPERATION
: TO BE PREMATURELY TERMINATED. THIS SHOULD LEAVE
: THE POSITION JUST BEFORE THE FIRST RECORD ON
: TAPE.
:8. TAPE POSITION IS VERIFIED BY ISSUING ANOTHER
: SPACE RECORDS REVERSE COMMAND AND VERIFYING THAT
: TAPE STATUS ALERT TERMINATION OCCURS, WITH THE
: REVERSE INTO BOT (RIB) STATUS ERROR BIT SET.
:9. A SPACE RECORDS FORWARD COMMAND THAT CONTAINS A
: RECORD COUNT GREATER THAN 1 IS ISSUED AND IT IS
: VERIFIED THAT TAPE STATUS ALERT TERMINATION
: OCCURED, TMK=1, AND THAT RBPCR (RESIDUAL
: BYTE/RECORD COUNTER) CONTAINS THE PROPER NONZERO
: VALUE. THIS OPERATION VERIFIES THAT DETECTION OF


```
10823 106212 104406 002172 110120 MOV UNITN,T28DSW ;SET UP DRIVE NUMBER TRAP C$CLP1
10824 106214 013737 110100 MOV #T28PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
10825 106222 012704 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
10826 106226 004737 002212 BCS 68$ ;BR, IF COMMAND ISSUED OK
10827 106232 103407 002212 INC FATFLG ;BUMP COUNT
10831 106234 005237 002212 MOV R0,R1 ;SAVE CONTENTS OF TSSR
10832 106240 010001 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
106242 104456 TRAP C$SERHRD
106244 001462 .WORD 818
106246 005054 .WORD WRTMSG
106250 012124 .WORD SFIMSG
10833 106252 68$: CKLOOP ;LOOP IF SELECTED
106252 104406 TRAP C$CLP1
10834 106254 012737 140011 110220 MOV #140011,T28PK3 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
10835 106262 012704 110220 MOV #T28PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
10836 106266 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
10837 106272 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
10838 106276 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10839 106302 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
10840 106306 020102 CMP R1,R2 ;ARE THEY EQUAL
10841 106310 001406 BEQ 70$ ;BR, IF OK
10842 106312 005237 002212 INC FATFLG ;BUMP COUNT
10846 106316 005237 002212 ERRHRD ERRNO,T28WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE TAPE MARK
106316 104456 TRAP C$SERHRD
106320 001463 .WORD 819
106322 111333 .WORD T28WDC
106324 012136 .WORD PKTSSR
10847 106326 70$: CKLOOP ;LOOP IF SELECTED
106326 104406 TRAP C$CLP1
10848 106330 013701 110130 MOV T28BFR+6,R1 ;PICK UP XSTO (VCK CHECK)
10849 106334 010102 MOV R1,R2 ;SET UP EXPECTED
10850 106336 042702 000020 BIC #BIT4,R2 ;VCK SHOULD BE 0
10851 106342 020102 CMP R1,R2 ;IS VCK SET CORRECTLY
10852 106344 001406 BEQ 80$ ;BR, IF VCK IS CLEAR
10853 106346 005237 002212 INC FATFLG ;BUMP COUNT
10857 106352 005237 002212 ERRHRD ERRNO,T28VCK,EXPREC ;VCK WAS NOT CLEAR AFTER CVC=1
106352 104456 TRAP C$SERHRD
106354 001464 .WORD 820
106356 111412 .WORD T28VCK
106360 015564 .WORD EXPREC
10858 106362 80$: CKLOOP ;LOOP IF SELECTED
106362 104406 TRAP C$CLP1
10859 106364 013701 110130 MOV T28BFR+6,R1 ;PICK UP XSTO (CHECK TMK)
10860 106370 010102 MOV R1,R2 ;SET UP EXPECTED
10861 106372 052702 100000 BIS #BIT15,R2 ;TMK SHOULD BE SET
10862 106376 020102 CMP R1,R2 ;WAS TMK SET
10863 106400 001406 BEQ 90$ ;BR, IF TMK WAS SET
10864 106402 005237 002212 INC FATFLG ;BUMP COUNT
10868 106406 005237 002212 ERRHRD ERRNO,T28TMK,EXPREC ;TMK WAS NOT SET AFTER WRT TAPE MARK
106406 104456 TRAP C$SERHRD
106410 001465 .WORD 821
106412 111465 .WORD T28TMK
106414 015564 .WORD EXPREC
10869 106416 90$: CKLOOP ;LOOP IF SELECTED
10870 106420 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND TRAP C$CLP1
```

10871	106424	103411			BCS	130\$:BR, IF NO PROBLEM		
10872	106426	010004			MOV	R0,R4		:SAVE PACKET ADDRESS		
10873	106430	016501	000002		MOV	TSSR(R5),R1		:GET TSSR STATUS		
10874	106434	005237	002212		INC	FATFLG		:BUMP COUNT		
10878	106440				ERRHRD	ERRNO,T28RWN,PKTSSR		:REWIND NOT ACCEPTED		
	106440	104456							TRAP	C\$ERHRD
	106442	001466							.WORD	822
	106444	111211							.WORD	T28RWN
	106446	012136							.WORD	PKTSSR
10879	106450				130\$:	CKLOOP		:LOOP IF SELECTED		
	106450	104406							TRAP	C\$CLP1
10880	106452	013701	110130		MOV	T28BFR+6,R1		:PICK UP XSTO		
10881	106456	010102			MOV	R1,R2		:SET UP EXPECTED		
10882	106460	052702	000002		BIS	#BIT1,R2		:SET BOT BIT IN EXPECTED		
10883	106464	020102			CMP	R1,R2		:DOES EXP = REC'D		
10884	106466	001406			BEQ	140\$:BR, IF EQUAL (OK)		
10885	106470	005237	002212		INC	FATFLG		:BUMP COUNT		
10889	106474				ERRHRD	ERRNO,T28BOT,EXPREC		:TAPE NOT AT BOT AFTER REWIND		
	106474	104456							TRAP	C\$ERHRD
	106476	001467							.WORD	823
	106500	111067							.WORD	T28BOT
	106502	015564							.WORD	EXPREC
10890	106504				140\$:	CKLOOP		:LOOP IF SELECTED		
	106504	104406							TRAP	C\$CLP1
10891	106506	012703	000012		MOV	#10,R3		:NUMBER OF RECORDS TO WRITE TM		
10892	106512	012737	140011	110220	MOV	#140011,T28PK3		:WRITE TAPE MARK,ACK,CVC=1 COMMAND		
10893	106520	012704	110220		MOV	#T28PK3,R4		:SET UP R4 WITH PACKET ADDRESS		
10894	106524	010465	000000		155\$:	MOV	R4,TSDB(R5)	:ISSUE COMMAND		
10895	106530	004737	016340		JSR	PC,WAITF		:WAIT FOR SSR TO SET		
10896	106534	016501	000002		MOV	TSSR(R5),R1		:PICK UP TSSR		
10897	106540	012702	000200		MOV	#SSR,R2		:SET UP EXPECTED (SSR ONLY)		
10898	106544	020102			CMP	R1,R2		:WAS STATUS GOOD		
10899	106546	001406			BEQ	165\$:BR, IF TERMINATION WAS GOOD		
10900	106550	005237	002212		INC	FATFLG		:BUMP COUNT		
10904	106554				ERRHRD	ERRNO,T28WDC,PKTSSR		:TSSR NOT CORRECT AFTER WRT TAPE M.		
	105554	104456							TRAP	C\$ERHRD
	106556	001470							.WORD	824
	106560	111333							.WORD	T28WDC
	106562	012136							.WORD	PKTSSR
10905	106564				165\$:	CKLOOP		:LOOP IF SELECTED		
	106564	104406							TRAP	C\$CLP1
10906	106566	013701	110130		MOV	T28BFR+6,R1		:PICK UP XSTO		
10907	106572	010102			MOV	R1,R2		:SET UP EXPECTED		
10908	106574	052702	100000		BIS	#BIT15,R2		:SET TMK BIT IN EXPECTED		
10909	106600	020102			CMP	R1,R2		:DOES EXP = REC'D		
10910	106602	001406			BEQ	180\$:BR, IF EQUAL (OK)		
10911	106604	005237	002212		INC	FATFLG		:BUMP COUNT		
10915	106610				ERRHRD	ERRNO,T28TMK,EXPREC		:TMK NOT SET AFTER WRT TAPE MARK		
	106610	104456							TRAP	C\$ERHRD
	106612	001471							.WORD	825
	106614	111465							.WORD	T28TMK
	106616	015564							.WORD	EXPREC
10916	106620				180\$:	CKLOOP		:LOOP IF SELECTED		
	106620	104406							TRAP	C\$CLP1
10917	106622	005303			DEC	R3		:BUMP COUNTER DOWN		
10918	106624	001337			BNE	155\$:BR, IF LESS THAN 10 TAPE MARKS		
10919	106626	012700	177777		MOV	#177777,R0		:VALUE TO WRITTEN TO MEMORY		

10920	106632	004737	017512		JSR	PC,FILLMEM	:FILL MEM WITH ALL ONES
10921	106636	013737	003114	110222	MOV	FREE,T28WB	:STARTING READ BUFFER ADDRESS
10922	106644	012737	140401	110220	MOV	#140401,T28PK3	:READ REVERSE,ACK, COMMAND
10923	106652	012704	110220		MOV	#T28PK3,R4	:SET UP R4 WITH PACKET ADDRESS
10924	106656	013737	000024	110226	MOV	20.,T28SZ	:SET UP RECORD SIZE IN PACKET
10925	106664	010465	000000		MOV	R4,TSDB(R5)	:ISSUE COMMAND
10926	106670	004737	016340		JSR	PC,WAITF	:WAIT FOR SSR TO SET
10927	106674	016501	000002		MOV	TSSR(R5),R1	:GET TSSR CONTENTS
10928	106700	012702	100204		MOV	#SSR!SC!BIT2,R2	:SET UP EXPECTED
10929	106704	020102			CMP	R1,R2	:ARE THEY EQUAL
10930	106706	001406			BEQ	200\$:BR, IF OK
10931	106710	005237	002212		INC	FATFLG	:BUMP COUNT
10935	106714				ERRHRD	ERRNO,T28RDF,PKTSSR	:TSSR INCORRECT AFTER WRITE DATA
	106714	104456					TRAP CSERHRD
	106716	001472					.WORD 826
	106720	110424					.WORD T28RDF
	106722	012136					.WORD PKTSSR
10936	106724			200\$:	CKLOOP		:LOOP IF SELECTED
	106724	104406					TRAP CSCLP1
10937	106726	013701	110130		MOV	T28BFR+6,R1	:PICK UP XSTO
10938	106732	010102			MOV	R1,R2	:SET UP EXPECTED
10939	106734	052702	100000		BIS	#BIT15,R2	:TMK SHOULD BE SET
10940	106740	020102			CMP	R1,R2	:IS TMK SET
10941	106742	001406			BEQ	210\$:BR, IF TMK WAS SET (GOOD)
10942	106744	005237	002212		INC	FATFLG	:BUMP COUNT
10946	106750				ERRHRD	ERRNO,T28RRM,EXPREC	:TMK NOT SET AFTER READ REV
	106750	104456					TRAP CSERHRD
	106752	001473					.WORD 827
	106754	111537					.WORD T28RRM
	106756	015564					.WORD EXPREC
10947	106760			210\$:	CKLOOP		:LOOP IF SELECTED
	106760	104406					TRAP CSCLP1
10948	106762	017701	074126		MOV	@FREE,R1	:FIRST LOC IN READ BUFFER
10949	106766	012702	177777		MOV	#177777,R2	:EXPECTED IF NO DATA TRANS.
10950	106772	020102			CMP	R1,R2	:DID ANY DATA GET TRANSFERRED
10951	106774	001406			BEQ	220\$:BR, IF NO DATA TRANS (GOOD)
10952	106776	005237	002212		INC	FATFLG	:BUMP COUNT
10956	107002				ERRHRD	ERRNO,T28DTR,EXPREC	:DATA TRANSFERRED ON READ TAPE MARK
	107002	104456					TRAP CSERHRD
	107004	001474					.WORD 828
	107006	111752					.WORD T28DTR
	107010	015564					.WORD EXPREC
10957	107012			220\$:	CKLOOP		:LOOP IF SELECTED
	107012	104406					TRAP CSCLP1
10958	107014	012737	100410	110220	MOV	#100410,T28PK3	:SPACE REVERSE,ACK, COMMAND
10959	107022	012737	000001	110222	MOV	#1,T28RB	:NUMBER OF RECORDS TO SPACE BACK
10960	107030	012704	110220		MOV	#T28PK3,R4	:SET UP R4 WITH PACKET ADDRESS
10961	107034	010465	000000		MOV	R4,TSDB(R5)	:ISSUE COMMAND
10962	107040	004737	016340		JSR	PC,WAITF	:WAIT FOR SSR TO SET
10963	107044	016501	000002		MOV	TSSR(R5),R1	:GET TSSR CONTENTS
10964	107050	012702	100204		MOV	#SSR!SC!BIT2,R2	:SET UP EXPECTED
10965	107054	020102			CMP	R1,R2	:ARE THEY EQUAL
10966	107056	001406			BEQ	222\$:BR, IF OK
10967	107060	005237	002212		INC	FATFLG	:BUMP COUNT
10971	107064				ERRHRD	ERRNO,T28RDG,PKTSSR	:TSSR INCORRECT AFTER SPACE CMD.
	107064	104456					TRAP CSERHRD
	107066	001475					.WORD 829

	107070	110505					.WORD	T28RDG
	107072	012136					.WORD	PKTSSR
10972	107074			222\$:	CKLOOP	:LOOP IF SELECTED		
	107074	104406					TRAP	C\$CLP1
10973	107076	013701	110130		MOV T28BFR+6,R1	:PICK UP XSTO		
10974	107102	010102			MOV R1,R2	:SET UP EXPECTED		
10975	107104	052702	100000		BIS #BIT15,R2	:TMK SHOULD BE SET		
10976	107110	020102			CMP R1,R2	:IS TMK SET		
10977	107112	001406			BEQ 226\$:BR, IF TMK WAS SET (GOOD)		
10978	107114	005237	002212		INC FATFLG	:BUMP COUNT		
10982	107120				ERRHRD ERRNO,T28RRN,EXPREC	:TMK NOT SET AFTER SPACE REV		
	107120	104456					TRAP	C\$ERHRD
	107122	001476					.WORD	830
	107124	111615					.WORD	T28RRN
	107126	015564					.WORD	EXPREC
10983	107130			226\$:	CKLOOP	:LOOP IF SELECTED		
	107130	104406					TRAP	C\$CLP1
10984	107132	004737	011104		JSR PC,REWIND	:CALL TAPE REWIND COMMAND		
10985	107136	103411			BCS 230\$:BR, IF NO PROBLEM		
10986	107140	010004			MOV R0,R4	:SAVE PACKET ADDRESS		
10987	107142	016501	000002		MOV TSSR(R5),R1	:GET TSSR		
10988	107146	005237	002212		INC FATFLG	:BUMP COUNT		
10992	107152				ERRHRD ERRNO,T28RWN,PKTSSR	:REWIND NOT ACCEPTED		
	107152	104456					TRAP	C\$ERHRD
	107154	001477					.WORD	831
	107156	111211					.WORD	T28RWN
	107160	012136					.WORD	PKTSSR
10993	107162			230\$:	CKLOOP	:LOOP IF SELECTED		
	107162	104406					TRAP	C\$CLP1
10994	107164	013701	110130		MOV T28BFR+6,R1	:PICK UP XSTO		
10995	107170	010102			MOV R1,R2	:SET UP EXPECTED		
10996	107172	052702	000002		BIS #BIT1,R2	:SET BOT BIT IN EXPECTED		
10997	107176	020102			CMP R1,R2	:DOES EXP = REC'D		
10998	107200	001406			BEQ 240\$:BR, IF EQUAL (OK)		
10999	107202	005237	002212		INC FATFLG	:BUMP COUNT		
11003	107206				ERRHRD ERRNO,T28BOT,EXPREC	:TAPE NOT AT BOT AFTER REWIND		
	107206	104456					TRAP	C\$ERHRD
	107210	001500					.WORD	832
	107212	111067					.WORD	T28BOT
	107214	015564					.WORD	EXPREC
11004	107216			240\$:	CKLOOP	:LOOP IF SELECTED		
	107216	104406					TRAP	C\$CLP1
11005	107220	012700	177777		MOV #177777,R0	:VALUE TO WRITTEN TO MEMORY		
11006	107224	004737	017512		JSR PC,FILLMEM	:FILL MEM WITH ALL ONES		
11007	107230	013737	003114	110222	MOV FREE,T28RB	:STARTING READ BUFFER ADDRESS		
11008	107236	012737	100001	110220	MOV #100001,T28PK3	:READ FORWARD,ACK, COMMAND		
11009	107244	012704	110220		MOV #T28PK3,R4	:SET UP R4 WITH PACKET ADDRESS		
11010	107250	013737	000024	110226	MOV 20.,T28SZ	:SET UP RECORD SIZE IN PACKET		
11011	107256	010465	000000		MOV R4,TSDB(R5)	:ISSUE COMMAND		
11012	107262	004737	016340		JSR PC,WAITF	:WAIT FOR SSR TO SET		
11013	107266	016501	000002		MOV TSSR(R5),R1	:GET TSSR CONTENTS		
11014	107272	012702	100204		MOV #SSR!SC!BIT2,R2	:SET JP EXPECTED		
11015	107276	020102			CMP R1,R2	:ARE THEY EQUAL		
11016	107300	001406			BEQ 245\$:BR, IF OK		
11017	107302	005237	002212		INC FATFLG	:BUMP COUNT		
11021	107306				ERRHRD ERRNO,T28WDE,PKTSSR	:TSSR INCORRECT AFTER WRITE DATA		
	107306	104456					TRAP	C\$ERHRD

	107310	001501										.WORD	833
	107312	110776										.WORD	T28WDE
	107314	012136										.WORD	PKTSSR
11022	107316				245\$:	CKLOOP							:LOOP IF SELECTED
	107316	104406										TRAP	C\$CLP1
11023	107320	013701	110130			MOV	T28BFR+6,R1						:PICK UP XSTO
11024	107324	010102				MOV	R1,R2						:SET UP EXPECTED
11025	107326	052702	100000			BIS	#BIT15,R2						:TMK SHOULD BE SET
11026	107332	020102				CMP	R1,R2						:IS TMK SET
11027	107334	001406				BEQ	247\$:BR, IF TMK WAS SET (GOOD)
11028	107336	005237	002212			INC	FATFLG						:BUMP COUNT
11032	107342					ERRHRD	ERRNO,T28RRP,EXPREC						:TMK NOT SET AFTER READ REV
	107342	104456										TRAP	C\$ERHRD
	107344	001502										.WORD	834
	107346	111674										.WORD	T28RRP
	107350	015564										.WORD	EXPREC
11033	107352					247\$:	CKLOOP						:LOOP IF SELECTED
	107352	104406										TRAP	C\$CLP1
11034	107354	017701	073534			MOV	@FREE,R1						:FIRST LOC IN READ BUFFER
11035	107360	012702	177777			MOV	#177777,R2						:EXPECTED IF NO DATA TRANS.
11036	107364	020102				CMP	R1,R2						:DID ANY DATA GET TRANSFERRED
11037	107366	001406				BEQ	250\$:BR, IF NO DATA TRANS (GOOD)
11038	107370	005237	002212			INC	FATFLG						:BUMP COUNT
11042	107374					ERRHRD	ERRNO,T28DTR,EXPREC						:DATA TRANSFERRED ON READ TAPE MARK
	107374	104456										TRAP	C\$ERHRD
	107376	001503										.WORD	835
	107400	111752										.WORD	T28DTR
	107402	015564										.WORD	EXPREC
11043	107404					250\$:	CKLOOP						:LOOP IF SELECTED
	107404	104406										TRAP	C\$CLP1
11044	107406	012737	100410	110220		MOV	#100410,T28PK3						:SPACE REVERSE,ACK, COMMAND
11045	107414	012737	000005	110222		MOV	#5,T28RB						:NUMBER OF RECORDS TO SPACE BACK
11046	107422	012704	110220			MOV	#T28PK3,R4						:SET UP R4 WITH PACKET ADDRESS
11047	107426	010465	000000			MOV	R4,TSDB(R5)						:ISSUE COMMAND
11048	107432	004737	016340			JSR	PC,WAITF						:WAIT FOR SSR TO SET
11049	107436	016501	000002			MOV	TSSR(R5),R1						:GET TSSR CONTENTS
11050	107442	012702	100204			MOV	#SSR!SC!BIT2,R2						:SET UP EXPECTED
11051	107446	020102				CMP	R1,R2						:ARE THEY EQUAL
11052	107450	001406				BEQ	260\$:BR, IF OK
11053	107452	005237	002212			INC	FATFLG						:BUMP COUNT
11057	107456					ERRHRD	ERRNO,T28RDG,PKTSSR						:TSSR INCORRECT AFTER SPACE REV CMD.
	107456	104456										TRAP	C\$ERHRD
	107460	001504										.WORD	836
	107462	110505										.WORD	T28RDG
	107464	012136										.WORD	PKTSSR
11058	107466					260\$:	CKLOOP						:LOOP IF SELECTED
	107466	104406										TRAP	C\$CLP1
11059	107470	013701	110130			MOV	T28BFR+6,R1						:PICK UP XSTO
11060	107474	010102				MOV	R1,R2						:SET UP EXPECTED
11061	107476	052702	100000			BIS	#BIT15,R2						:TMK SHOULD BE SET
11062	107502	020102				CMP	R1,R2						:IS TMK SET
11063	107504	001406				BEQ	270\$:BR, IF TMK WAS SET (GOOD)
11064	107506	005237	002212			INC	FATFLG						:BUMP COUNT
11068	107512					ERRHRD	ERRNO,T28RRN,EXPREC						:TMK NOT SET AFTER READ REV
	107512	104456										TRAP	C\$ERHRD
	107514	001505										.WORD	837
	107516	111615										.WORD	T28RRN


```
107746 0G1511
107750 110424
107752 015564
11120 107754 310$: CKLOOP ;LOOP IF SELECTED .WORD 841
107754 104406 ;PICK UP XSTO TRAP C$CLP1
11121 107756 013701 110130 MOV T28BFR+6,R1 ;SET UP EXPECTED
11122 107762 010102 MOV R1,R2 ;TMK SHOULD BE SET
11123 107764 052702 100000 BIS #BIT15,R2 ;IS TMK SET
11124 107770 020102 CMP R1,R2 ;BR, IF TMK WAS SET (GOOD)
11125 107772 001406 BEQ 320$ ;BUMP COUNT
11126 107774 005237 002212 INC FATFLG ;TMK NOT SET AFTER READ REV
11130 110000 ERRHRD ERRNO,T28RRP,EXPREC TRAP C$ERHRD
110000 104456 .WORD 842
110002 001512 .WORD T28RRP
110004 111674 .WORD EXPREC
110006 015564
11131 110010 320$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
110010 104406 ;PICK UP RESIDUAL BYTE COUNTER
11132 110012 013701 110126 MOV T28BFR+4,R1 ;SHOULD BE THE DIFFERENCE
11133 110016 012702 000004 MOV #4.,R2 ;IS COUNTER CORRECT
11134 110022 020102 CMP R1,R2 ;BR, IF COUNTER CORRECT
11135 110024 001406 BEQ 330$ ;BUMP COUNT
11136 110026 005237 002212 INC FATFLG ;RESIDUAL BYTE COUNTER NOT CORRECT
11140 110032 ERRHRD ERRNO,T28PBP,EXPREC TRAP C$ERHRD
110032 104456 .WORD 843
110034 001513 .WORD T28PBP
110036 110341 .WORD EXPREC
110040 015564
11141 110042 330$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
110042 104406 ;<<<<<<<<<<<<< END SUBTEST >>>>>>>>>>
11142 110044 ENDSUB L10133: TRAP C$ESUB
110044 104403 ;IS ERROR COUNT AT 25
11143 110046 023727 002212 000017 CMP FATFLG,#15. ;BR, IF LESS THAN 25
11144 110054 103402 BLO 999$ ;TRY TO DROP THE UNIT
11145 110056 004737 017272 JSR PC,CKDROP
11146 110062 999$:
11147 :
11148 :
11149 :
11150 110062 004737 016546 JSR PC,TSTLOOP ;DO WE NEED TO ITERATE TEST
11151 110066 103002 BCC 163$ ;BR, IF NO LOOP REQUIRED
11152 110070 000137 104750 JMP T28LOOP ;EXECUTE AGAIN
11153 110074 163$:
11154 110074 EXIT TST ;ALL DONE THIS TEST
110074 104432 TRAP C$EXIT
110076 002236 .WORD L10130-
```

11156			:+		
11157			:LOCAL STORAGE FOR THIS TEST		
11158			:		
11162 110100			T28PACKET:		:COMMAND PACKET FOR TEST
11163 110100 100004			.WORD 100004		:WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
11164 110102 110110			.WORD T28DATA		:ADDRESS OF CHARACTERISTICS BLOCK
11165 110104 000000			.WORD 0		
11166 110106 000012			.WORD 10.		:STARTING VALUE OF BLOCK SIZE
11167 110110			T28DATA:		:CHARACTERISTICS DATA BLOCK
11168 110110 110122			.WORD T28BFR		:ADDRESS OF MESSAGE BUFFER
11169 110112 000000			.WORD 0		
11170 110114 000024			.WORD 20.		:LENGTH OF MESSAGE BUFFER
11171 110116 000000			.WORD 0		
11172 110120 000000			T28DSW: .WORD 0		:SELECT DRIVE 0
11173 110122			T28BFR: .BLKW 25.		:MESSAGE BUFFER
11174			:		
11175			:WRITE SUBSYSTEM MEMORY COMMAND PACKET		
11176			:		
11178 110210			.=<. +10>&177770		
11180 110210			T28PK2:		
11181 110210 100006			.WORD 100006		:WRITE SUB SYS MEM COMMAND, IE AND ACK
11182 110212 110230			.WORD T28BF2		:ADDRESS OF SELECT BLOCK DATA
11183 110214 000000			.WORD 0		
11184 110216 000006			.WORD 6.		:SIZE OF DATA PACKET
11185					
11189 110220			T28PK3:		
11190 110220 100005			.WORD 100005		:REREAD COMMAND, AND ACK
11191 110222			T28RB:		
11192 110222 003114			T28WB: .WORD FREE		:ADDRESS OF WRITE BUFFER
11193 110224 000000			.WORD 0		
11194 110226 000000			T28SZ: .WORD 0		:SIZE OF BUFFER (EXTENT)
11195			.EVEN		
11196			:		
11197			:		
11198			:		
11199 110230			T28BF2:		
11200 110230 010			T28BS0: .BYTE 10		:BSELO AREA
11201 110231 200			T28BS1: .BYTE 200		:BSEL1 AREA
11202 110232 000000			T28S2: .WORD 0		:SEL 2 AREA
11203 110234 000000			T28S3: .WORD 0		:DATA AREA
11204			:		
11205			:		
11206			.EVEN		
11207			:TAPE MOTION PACKET COMMAND VALUES		
11208					
11209 110236			T28IMV:		
11210 110236 101411			.WORD 101411		:ILLEGAL MODE BITS TEST DATA
11211 110240 102011			.WORD 102011		
11212 110242 103411			.WORD 103411		
11213 110244 177777			.WORD 177777		
11214 110246 100011			T28RN: .WORD 100011		:WRITE TAPE MARK COMMAND
11215 110250 100411			T28WDR: .WORD 100411		:ERASE COMMAND
11216 110252 101011			T28CON: .WORD 101011		:WRITE TAPE MARK RETRY
11217 110254 177777			.WORD 177777		:END OF DATA
11218			:		
11219			:		
11220 110256 000000			T28CNT: .WORD 0		:TAPE TIMER COUNTER STORAGE AREA

TSV7 - HARDWARE TESTS 1-8
TEST 8: WRITE/READ TAPE MARK

MACRO M1113 25-MAY-82 09:19 PAGE 153-1

J 13

SEQ 0371

11221 110260 000000
11222 110262 000000
11223
11224

T28CNU: .WORD 0
T28DLY: .WORD 0
.EVEN

;TAPE TIMER COUNTER STORAGE AREA
;DELAY COUNTER

11226
 11227
 11228
 11229
 11230
 11231
 11232 110264
 11233 110341
 11234 110424
 11235 110505
 11236 110567
 11237 110644
 11238 110725
 11239 110776
 11240 111067
 11241 111134
 11242 111211
 11243 111260
 11244 111333
 11245 111412
 11246 111465
 11247 111537
 11248 111615
 11249 111674
 11250 111752
 11251 112034
 11252 112131
 11253
 11254
 11255
 11256
 11257
 11258
 11259
 11260
 11261 112156
 11262 112156
 11263 112162
 11264 112166
 11265 112172
 11266 112176
 11267 112200
 11268 112204
 11269 112210
 11270 112212
 11271 112216
 11272 112220
 11273 112224
 11274 112230
 11275 112236
 11276 112240
 11277 112244
 11278 112246
 11279
 11280
 11281 112250
 11282 112250

```

: +
: LOCAL TEXT MESSAGES FOR TEST
: -

```

```

160 T28RIB: .ASCIZ 'Tape Position Not Correct, RIB Should Be Set'
163 T28PBP: .ASCIZ 'Residual Byte Counter Register (RBPCR) Not Correct'
123 T28RDF: .ASCIZ 'TSSR Incorrect After READ REVERSE Into TAPE MARK'
123 T28RDG: .ASCIZ 'TSSR Incorrect After SPACE Command Into TAPE MARK'
123 T28WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
154 T28LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
111 T28SSR: .ASCIZ 'WRITE MISCELLANEOUS Command Not Accepted'
123 T28WDE: .ASCIZ 'TSSR Not Correct After READ DATA Command, Into TAPE MARK'
160 T28BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
123 T28TM: .ASCIZ 'TSSR Not Correct After FORMAT Command Reject'
167 T28RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
151 T28OFL: .ASCIZ 'Drive 7 Select Failed To Set 'OFL' In TSSR'
123 T28WDC: .ASCIZ 'TSSR Not Correct After WRITE TAPE MARK Command'
103 T28VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
113 T28TMK: .ASCIZ 'TMK Not Set After WRITE TAPE MARK Command'
113 T28RRM: .ASCIZ 'TMK Not Set After READ REVERSE Into TAPE MARK'
113 T28RRN: .ASCIZ 'TMK Not Set After SPACE REVERSE Into TAPE MARK'
113 T28RRP: .ASCIZ 'TMK Not Set After READ FORWARD Into TAPE MARK'
164 T28DTR: .ASCIZ 'Data Transferred On READ REVERSE Into A TAPE MARK'
164 T28DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
151 T28ID: .ASCIZ 'Write/Read Tape Mark'
      .EVEN

```

```

: +
: ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
: WRITE SUBSYSTEM MEMORY COMMAND
: -

```

```

T28REST:
      SAVREG
      MOV #T28PACKET,R1 ;SAVE THE REGISTERS
      MOV #100004,(R1)+ ;START OF THE PACKET
      MOV #T28DATA,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
      CLR (R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
      MOV #10,(R1)+ ;EXTENDED ADDRESS
      MOV #T28BFR,(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,T28BFR(R2) ;ALL ONES TO MESSAGE BUFFER
      TST -(R2) ;NEXT LOCATION
      CMP R2,#0 ;CHECK FOR END
      BNE 64$ ;KEEP GOING UNTIL DONE
      RTS PC ;RETURN

```

```

T28RT2:
      SAVREG
      ;SAVE THE REGISTERS

```

11283	112254	012701	110210
11284	112260	012721	100006
11285	112264	012721	110230
11286	112270	005021	
11287	112272	012721	000006
11288	112276	005021	
11289	112300	012701	110230
11290	112304	005021	
11291	112306	005011	
11292	112310	000207	
11293	112312		
11294	112312		
11295	112316	012701	110220
11296	112322	005021	
11297	112324	005021	
11298	112326	005021	
11299	112330	005011	
11300	112332	000207	
11301	112334		
	112334		
	112334	104401	
11302	112336		

```

MOV #T28PK2,R1 ;START OF THE PACKET
MOV #100006,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK.
MOV #T28BF2,(R1)+ ;ADDRESS OF DATA BLOCK
CLR (R1)+ ;EXTENDED ADDRESS
MOV #6,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
CLR (R1)+
MOV #T28BF2,R1 ;POINT TO DATA SEL AREA
CLR (R1)+
CLR (R1)
RTS PC ;RETURN

T28RT3:
SAVREG
MOV #T28PK3,R1 ;GET PACKET ADDRESS
CLR (R1)+ ;CLEAR COMMAND AREA
CLR (R1)+ ;CLEAR ADDRESS AREA
CLR (R1)+ ;CLEAR EXTENDED ADDRESS AREA
CLR (R1) ;SIZE OF DATA TRANSFER
RTS PC ;RETURN
ENDTST

L10130:
TRAP C$ETST

ENDMOD
  
```



```

1          .TITLE  TSV6 - PARAMETER CODING
7
12
18
19 112336   BGNMOD  TSV6
112336   TSV6::
20
21
22          .SBTTL  HARDWARE PARAMETER CODING SECTION
23
24          :++
25          : THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
26          : THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
27          : MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
28          : INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
29          : MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
30          : WI 4 THE OPERATOR.
31          :--
32 112336   BGNHRD
112336   .WORD  L10134-L$HARD/2
112340   L$HARD::
33
34 112340   GPRMA  HPM1,0,0,160010,177776,YES      :GET TSBA/TSDB REGISTER ADDRESS.
112340   .WORD  T$CODE
112342   .WORD  HPM1
112344   .WORD  T$LOLIM
112346   .WORD  T$HILIM
35 112350   GPRMA  HPM2,2,0,0,776,YES              :GET VECTOR ADDRESS.
112350   .WORD  T$CODE
112352   .WORD  HPM2
112354   .WORD  T$LOLIM
112356   .WORD  T$HILIM
36          ;GPRMD  HPM3,4,0,340,0,7,YES          :GET INTERRUPT PRIORITY.
37 112360   ENDHRD
          .EVEN
          L10134:
38 112360   104    105    126  HPM1:  .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
39 112414   111    116    124  HPM2:  .ASCIZ  'INTERRUPT VECTOR '
40 112440   111    116    124  HPM3:  .ASCIZ  'INTERRUPT PRIORITY '
41          .EVEN
42

```

```

44          .SBTTL SOFTWARE PARAMETER CODING SECTION
45
46          :++
47          : THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
48          : THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
49          : MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
50          : INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
51          : MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
52          : WITH THE OPERATOR.
53          :--
54 112470          BGNSFT
55 112470          .WORD L10135-L$SOFT/2
56 112472          000003
57          L$SOFT::
58          : GPRML SPM1,0,-1,YES ; GET TRANSPORT TEST FLAG.
59 112472          GPRML SPM4,2,-1,YES ; GET ITERATION CONTROL.
60 112472          .WORD T$CODE
61 112474          .WORD SPM4
62 112476          .WORD -1
63          : GPRMD SPM6,4,D,7777,0,7777,YES ; GET LOCAL ERROR LIMIT
64          : GPRMD SPM7,6,D,7777,0,7777,YES ; GET GLOBAL ERROR LIMIT
65 112500          ENDSFT
66          .EVEN
67          L10135:
68          SPM1: .ASCIZ 'ENABLE TRANSPORT TESTS '
69          SPM4: .ASCIZ 'INHIBIT ITERATIONS '
70          SPM6: .ASCIZ 'PER TEST ERROR LIMIT '
71          SPM7: .ASCIZ 'PER UNIT ERROR LIMIT '
72          .SBTTL PATCH AREA
73          :
74          : FINALLY A GENEROUS PATCH AREA.
75          :
76          : AND AN ADJUSTMENT TO ACCOUNT FOR THE 'LASTAD BIT7' HACK
77          : DESCRIBED IN 'SUPPRG.MEM' (FOR REV C).
78          :
79          PATCH::
80          .BLKW 32.
81          .=.!377+1
82 113000          LASTAD ;SET LAST USED ADDRESS.
83          .EVEN
84          .WORD 0
85          .WORD 0
86          L$LAST::
87          ENDMOD
88          .END
    
```

ADSSR	012216	G	C\$AU	=	000052	DEVDR0	023466	FREE	003114	G	INCERK	017134					
ADR	=	000020	G	C\$AUTO	=	000061	DEVNRD	023405	FREEHI	003120		INTCPC	016240				
AMBTSS	006725		C\$BRK	=	000022	DEVNXR	023323	FRESIZ	003116	G	INTFLA	016235					
ASSEMB	=	000010		C\$BSEG	=	000004	DEVONL	023253	FUSI	004115		INTMAS	016234				
A1716	=	000003		C\$BSUB	=	000002	DEVSUM	023216	F\$AU	=	000015	INTR	016306	G			
BADDAT	003146	G		C\$CEFG	=	000045	DFPTBL	002146	G	F\$AUTO	=	000020	INTREC	002214	G		
BADSSR	015770	G		C\$CLCK	=	000062	DIAGMC	=	000000	F\$BGN	=	000040	INTVEC	016236			
BDVPCR	=	177520	G	C\$CLEA	=	000012	DICEC	=	000001	F\$CLEA	=	000007	INTX	004276			
BENBSW	002220	G		C\$CLOS	=	000035	DSBINT	016274		F\$DU	=	000016	INVERT	021276	G		
BIE	=	040000		C\$CLP1	=	000006	DUAD12	004641		F\$END	=	000041	IOKCKI	=	000200		
BIT0	=	000001	G	C\$CVEC	=	000036	DUFLG	003102	G	F\$HARD	=	000004	IOKSTP	=	000001		
BIT00	=	000001	G	C\$DCLN	=	000044	DUMMY	003052		F\$HW	=	000013	IPRI	002202	G		
BIT01	=	000002	G	C\$DODU	=	000051	EF.CON	=	000036	G	F\$INIT	=	000006	ISR	=	000100	G
BIT02	=	000004	G	C\$DRPT	=	000024	EF.NEW	=	000035	G	F\$JMP	=	000050	IVEC	002200	G	
BIT03	=	000010	G	C\$DU	=	000053	EF.PWR	=	000034	G	F\$MOD	=	000000	IXE	=	004000	G
BIT04	=	000020	G	C\$EDIT	=	000003	EF.RES	=	000037	G	F\$MSG	=	000011	ISAU	=	000041	
BIT05	=	000040	G	C\$ERDF	=	000055	EF.STA	=	000040	G	F\$PROT	=	000021	ISAUTO	=	000041	
BIT06	=	000100	G	C\$ERHR	=	000056	EMAXDU	017067		F\$PWR	=	000017	ISCLN	=	000041		
BIT07	=	000200	G	C\$ERRO	=	000060	EN	=	000000	F\$RPT	=	000012	ISDU	=	000041		
BIT08	=	000400	G	C\$ERSF	=	000054	ENAIN	016242		F\$SEG	=	000003	ISHRD	=	000041		
BIT09	=	001000	G	C\$ERSO	=	000057	ENVIRN	020720		F\$SOFT	=	000005	ISINIT	=	000041		
BIT1	=	000002	G	C\$ESCA	=	000010	EPRTSW	002170	G	F\$SRV	=	000010	ISMOD	=	000041		
BIT10	=	002000	G	C\$ESEG	=	000005	EPRT1	006356		F\$SUB	=	000002	ISMSG	=	000041		
BIT11	=	004000	G	C\$ESUB	=	000003	EPRT2	006446		F\$SW	=	000014	ISPROT	=	000040		
BIT12	=	010000	G	C\$ETST	=	000001	ERCM	012023		F\$TEST	=	000001	ISPTAB	=	000041		
BIT13	=	020000	G	C\$EXIT	=	000032	ERRHI	002226	G	GDDAT	003150	G	ISPWR	=	000041		
BIT14	=	040000	G	C\$GETB	=	000026	ERRK	017046		GERRMA	002164	G	ISRPT	=	000041		
BIT15	=	100000	G	C\$GETW	=	000027	ERRLO	002230	G	GETPAT	020264	G	ISSEG	=	000041		
BIT2	=	000004	G	C\$GMAN	=	000043	ERRNO	=	001513	GETSEL	020346	G	ISSETU	=	000041		
BIT3	=	000010	G	C\$GPHR	=	000042	ERRVEC	=	000004	G	G\$CNT0	=	000200	ISSFT	=	000041	
BIT4	=	000020	G	C\$GPLO	=	000030	ERTABE	003366		G\$DELM	=	000372	ISSRV	=	000041		
BIT5	=	000040	G	C\$GPRI	=	000040	ERTABL	003166		G\$DISP	=	000003	ISSUB	=	000041		
BIT6	=	000100	G	C\$INIT	=	000011	ESUM	017050		G\$EXCP	=	000400	ISTST	=	000041		
BIT7	=	000200	G	C\$INLP	=	000020	EVL	=	000004	G	G\$HILI	=	000002	JSJMP	=	000167	
BIT8	=	000400	G	C\$MANI	=	000050	EXBCNT	=	000010		G\$LOLI	=	000001	KIPAR0	=	172340	
BIT9	=	001000	G	C\$MEM	=	000031	EXIT	034174		G\$NO	=	000000	KIPAR1	=	172342		
BOE	=	000400	G	C\$MSG	=	000023	EXPBRE	015572	G	G\$OFFS	=	000400	KIPAR2	=	172344		
BRINIT	004455		C\$OPEN	=	000034	EXPD	002222	G	G\$OFSI	=	000376	KIPAR3	=	172346			
BSELO	=	000000		C\$PNTB	=	000014	EXPGOT	004531		G\$PRMA	=	000001	KIPAR4	=	172350		
BSEL1	=	000001		C\$PNTF	=	000017	EXPGT2	004565		G\$PRMD	=	000002	KIPAR5	=	172352		
CHKAMB	016134		C\$PNTS	=	000016	EXPMSC	002312	G	G\$PRML	=	000000	KIPAR6	=	172354			
CHKMAN	020570	G	C\$PNTX	=	000015	EXPREC	015564	G	G\$RADA	=	000140	KIPAR7	=	172356			
CHKTSS	016426		C\$QIO	=	000377	EXTA	005770		G\$RADB	=	000000	KIPDR0	=	172300			
CKDROP	017272		C\$RDBU	=	000007	EXTEND	005766		G\$RADD	=	000040	KIPDR1	=	172302			
CKEMAX	017172		C\$REFG	=	000047	EXTFEA	002216	G	G\$RADL	=	000120	KIPDR2	=	172304			
CKMSG	011450	G	C\$RESE	=	000033	ESEND	=	002100	G\$RADO	=	000020	KIPDR3	=	172306			
CKMSG2	011570	G	C\$REVI	=	000003	ESLOAD	=	000035	G\$XFER	=	000004	KIPDR4	=	172310			
CKRAM	011204	G	C\$RFLA	=	000021	FATAL	034274		G\$YES	=	000010	KIPDR5	=	172312			
CKRAM2	011314	G	C\$RPT	=	000025	FATERR	=	000060	HIADDR	=	001400	KIPDR6	=	172314			
CMDPKT	021350	G	C\$SEFG	=	000046	FATFLG	002212	G	HOE	=	100000	G	KIPDR7	=	172316		
CMEM	017750		C\$SPRI	=	000041	FERCM	012012		HPI1	112360		KTENAB	003124	G			
CONF IG	017340		C\$SVEC	=	000037	FIFEXP	012260	G	HPI2	112414		KTLG	003122	G			
COUNT	002300	G	C\$TPRI	=	000013	FIF1MS	012332		HPI3	112440		KTINIT	021144				
CSRADD	002176	G	DATA	002302	G	FIF2MS	012401		IBE	=	010000	G	KTOFF	017364			
CTAB	003154	G	DATASC	020322		FILLME	017512		IDU	=	000040	G	KTON	017346			
CTABE	003166	G	DEBUGM	011722		FNOINT	004213		IER	=	020000	G	LERRMA	002162	G		
CTABM	003154	G	DEVCNT	002210	G	FORCER	002166	G	IFAU	004254		LISTAL	=	000001			

LOE = 040000 G	L\$UNIT 002012 G	L10071 055352	M8189 005643	PRBEXP 015560
LOOPCN 002206 G	L10000 002154	L10072 047424	NBA = 002000	PRBMSG 015426
LOOPCO 013216	L10001 002166	L10073 050024	NEWPAS 022100	PRBREC 015562
LOOPFL 003152 G	L10002 005764	L10074 050500	NODEV 003104 G	PRBTOT 015513
LOT = 000010 G	L10003 012134	L10075 051144	NOINIT 004333	PRBYTE 015212 G
L\$ACP 002110 G	L10004 012152	L10076 051704	NOINTR 004217	PRI = 002000 G
L\$APT 002036 G	L10005 012170	L10077 052644	NOITS 002160 G	PRIADD 010250
L\$AU 022442 G	L10006 012176	L10100 053164	NOMAN 020624	PRIAO 010320
L\$AUT 002070 G	L10007 012214	L10101 053566	NOMEM 005456	PRIBXO 007702 G
L\$AUTO 022646 G	L10010 012232	L10102 074674	NP.IR = 000200	PRIEQU 010150
L\$CCP 002106 G	L10011 012256	L10103 056310	NP.LOO= 000040	PRIPKT 007460 G
L\$CLEA 022726 G	L10012 012330	L10104 057156	NP.OUT= 000100	PRIRAM 010156
L\$CO 002032 G	L10013 012500	L10105 060050	NP.WRP= 000020	PRITAD 010364
L\$DEPO 002011 G	L10014 013214	L10106 060776	NSI 004150	PRITSS 006022
L\$DESC 003400 G	L10015 014042	L10107 061554	NSINIT 004405	PRITO 010446
L\$DESP 002076 G	L10016 014064	L10110 062416	NUL 004525	PRITI 010511
L\$DEVP 002060 G	L10017 015570	L10111 063270	NULCR 004526	PRIXOR 010032 G
L\$DISP 002124 G	L10020 015576	L10112 064142	NXM = 004000	PRI00 = 000000 G
L\$DLY 002116 G	L10021 015604	L10113 065016	NXMFLG 003126 G	PRI01 = 001040 G
L\$DTP 002040 G	L10022 015616	L10114 065672	NXMHI 003132 G	PRI02 = 000100 G
L\$DTYP 002034 G	L10023 015640	L10115 066542	NXMLO 003130 G	PRI03 = 000140 G
L\$DU 022540 G	L10024 015666	L10116 067474	NXMTST 021542	PRI04 = 000200 G
L\$DUT 002072 G	L10025 016026	L10117 070504	NXR 003736	PRI05 = 000240 G
L\$DVTY 003372 G	L10026 016336	L10120 071064	NXRERR 005734 G	PRI06 = 000300 G
L\$EF 002052 G	L10030 022372	L10121 071540	NXRX 003775	PRI07 = 000340 G
L\$ENVI 002044 G	L10031 022536	L10122 104712	NXTU 022112	PRMESS 014332
L\$ETP 002102 G	L10032 022644	L10123 075316	OFL = 000100	PRMNO 002310 G
L\$EXP1 002046 G	L10033 022724	L10124 076100	ONEFIL= 000000	PRMSGE 014642 G
L\$EXP4 002064 G	L10034 022752	L10125 076722	OSAPTS= 000000	PRMSG0 015022
L\$EXP5 002066 G	L10035 023214	L10126 077624	OSAU = 000001	PRMSG1 015067
L\$HARD 112340 G	L10036 024650	L10127 101354	OSBGNR= 000001	PRMSG2 015125
L\$HIME 002120 G	L10037 027330	L10130 112334	OSBGNS= 000001	PROASC 014510
L\$HPCP 002016 G	L10040 025256	L10131 105312	OSDU = 000001	PRASC 014555
L\$HPTP 002022 G	L10041 025600	L10132 105572	OSERRT= 000000	PST32W 003142 G
L\$HW 002146 G	L10042 026160	L10133 110044	OSGNSW= 000001	PUNIT 022374
L\$ICP 002104 G	L10043 034320	L10134 112360	OSPOIN= 000001	PW.D11= 000021
L\$INIT 021646 G	L10044 027732	L10135 112500	OSSETU= 000000	PW.D13= 000022
L\$LADP 002026 G	L10045 030602	MEMADD 014044 G	PASRPT 022144	PW.D22= 000020
L\$LAST 113004 G	L10046 031422	MEMCK 021366 G	PATCH 112640 G	PW.NOP= 000000
L\$LOAD 002100 G	L10047 031636	MENASC 020537	PATDAT 020320	PW.NO1= 000023
L\$LUN 002074 G	L10050 032070	MENERR 020464	PC.ERA= 002400	PW.RDE= 000024
L\$MREV 002050 G	L10051 032402	MENRES 020566	PC.IER= 002000	PW.RDR= 000001
L\$NAME 002000 G	L10052 046434	MMVEC = 000250	PC.NOO= 001000	PW.RDS= 000005
L\$PRIO 002042 G	L10053 034766	MSA.FR= 000006	PC.REL= 000000	PW.RFI= 000003
L\$PROT 021636 G	L10054 035546	MSA.NO= 000000	PC.REW= 000400	PW.WCT= 000006
L\$PRT 002112 G	L10055 036322	MSA.NR= 000004	PKBCNT= 000006	PW.WFI= 000004
L\$REPP 002062 G	L10056 037024	MSA.VO= 000002	PKHI = 000004	PW.WFM= 000007
L\$REV 002010 G	L10057 037470	MSGEXP 012234 G	PKLOW = 000002	PW.WMI= 000010
L\$RPT 022754 G	L10060 040124	MSGLOO 013154 G	PKTADD 007644	PW.WNP= 000011
L\$SOFT 112472 G	L10061 040560	MSGSTA 012440 G	PKTFRM 007606	PW.WTR= 000002
L\$SPC 002056 G	L10062 041152	MSGSUB 014032 G	PKTGET 012154 G	P.ACK = 100000
L\$SPCP 002020 G	L10063 041570	MS.ATT= 000006	PKTMES 012200 G	P.CMD = 000037
L\$SPTP 002024 G	L10064 042034	MS.EXT= 000200	PKTRAM 004743 G	P.CONT= 000012
L\$STA 002030 G	L10065 042306	MS.RSD= 000001	PKTSSR 012136 G	P.CVC = 040000
L\$SW 002156 G	L10066 042542	MS.RSF= 000020	PNT = 001000 G	P.FMT = 000140
L\$TEST 002114 G	L10067 043042	MS.RST= 000010	PRAMPK 014066	P.FORM= 000011
L\$TIML 002014 G	L10070 043526	M8186 005552	PRASC 014613	P.GETS= 000017

P.IE = 000200	SPM6 112560	TSREJ = 000006	TSSCLE= 010034	T22WRT 026350
P.INIT= 000013	SPM7 112610	TSSDEF 006676	TSSDU = 010032	T23A 003134 G
P.MODE= 007400	SR0 = 177572	TSSR = 000002 G	TSSHAR= 010134	T23AM3 033210
P.OPP = 020000	SR1 = 177574	TSSRBI 003500 G	TSSHW = 010000	T23B 003136 G
P.POSI= 000010	SR2 = 177576	TSSRFO 006505	TSSINI= 010030	T23BA 033575
P.READ= 000001	SR3 = 172516	TSSRH = 000003 G	TSSMSG= 010025	T23BFR 032462
P.SWB = 010000	SSR = 000200	TSSX 004016	TSSPRO= 010027	T23BF2 032572
P.WRIT= 000005	STATCO 012502	TSTBLK 002742 G	TSSRPT= 010035	T23BS0 032572
P.WRTC= 000004	SVCGBL= 000000	TSTCNT 002204 G	TSSSOF= 010135	T23BS1 032573
P.WRTS= 000006	SVCINS= 000000	TSTEND 017010	TSSSRV= 010026	T23CHK 034132
QVP 002174 G	SVCSUB= 000001	TSTFLA 002304 G	TSSSUB= 010133	T23CON 032610
RAMASC 014246	SVCTAG= 000000	TSTLOO 016546 G	TSSSW = 010001	T23DAT 032450
RAMDAT 002232 G	SVCTST= 000001	TSTPTR 002306 G	TSTES= 010130	T23DSW 032460
RAMERR 015600 G	SLSYM= 010000	TSTSET 016600 G	T1 023536 G	T23EOT 032734
RAMEXP 015620 G	SO.IDB= 000010	TST21I 024474	T2 024652 G	T23ET 032647
RAMFOR 010206	SO.IFB= 000002	TST22I 027137	T2.1 024702	T23LOO 027376
RAMSIZ 002272 G	SO.IFP= 000001	TST23I 033736	T2.2 025274	T23OFL 033256
RAMTAD 015606 G	SO.ILD= 000020	TST24I 046202	T2.3 025616	T23PAC 032440
RCVHIA 002274 G	SO.ION= 000040	TST25I 055150	T21AM3 024353	T23PK2 032550
RCVLOA 002276 G	SO.IRD= 000100	TST26I 074477	T21BFR 024154	T23PK3 032560
RDERR 005204	SO.IRW= 000004	TST27I 104513	T21BF2 024250	T23RES 033752
RECMG 002456 G	SO.ISP= 000200	TST28I 112131	T21BS0 024250	T23RNC 033135
RECV 002224 G	S1.ICE= 002000	TSV2 002000 G	T21BS1 024251	T23RSZ 032570
REGSAV 020230	S1.IEO= 010000	TSV3 002166 G	T21DAT 024140	T23RT2 034044
RETERR 005370	S1.IFM= 001000	TSV4 021636 G	T21DLY 024152	T23RT3 034106
RETRY 034176	S1.IHE= 000400	TSV6 112336 G	T21DSW 024150	T23RWN 033066
REWIND 011104 G	S1.IID= 004000	TSV7 023536 G	T21LOO 023566	T23SSR 032614
RMCHBE= 000167	S1.IIR= 020000	TTIBFR= 177562 G	T21OFL 024453	T23SZ 032566
RMCHEN= 000200	S1.I2R= 040000	TTICSR= 177560 G	T21PAC 024130	T23S2 032574
RMMSGB= 000215	S1.PAR= 100000	TTIVC= 000060 G	T21PK2 024240	T23S3 032576
RMMSGE= 000234	S2.ATI= 000010	T\$ARGC= 000003	T21RES 024516	T23TM 033012
RMPKTB= 000201	S2.BTI= 000004	T\$CODE= 001130	T21RT2 024606	T23TMP 032600
RMPKTE= 000210	S2.DIM= 000200	T\$ERRN= 001513	T21SSR 024256	T23VCK 033522
RMR = 010000	S2.ILW= 000100	T\$EXCP= 000000	T21S2 024252	T23WB 032562
RWPACK 011200	S2.INR= 000020	T\$FLAG= 000040	T21S3 024254	T23WD 032604
SC = 100000	S2.OUT= 000040	T\$GMAN= 000000	T22AM3 026455	T23WDC 033420
SCE = 020000	S2.UND= 000003	T\$HILI= 000776	T22BFR 026242	T23WDD 033331
SCHERR 005276	TBLEND= 003052 G	T\$LAST= 000001	T22BF2 026340	T23WDR 032606
SCME 005011	TCOASC 006566	T\$LOLI= 000000	T22BS0 026340	T23WRT 032602
SDELAY 010750	TCOCOD 006766	T\$LSYM= 010000	T22BS1 026341	T23WSS 033647
SELASC 020532	TEMP1 003106 G	T\$LTNO= 000010	T22DAT 026230	T24AM3 045170
SELDAT= 000004	TEMP2 003110 G	T\$NEST= 177777	T22FOR 026354	T24BA 045522
SEL2 = 000002	TERCLS= 000016	T\$NSO = 000000	T22LOO 024702	T24BFR 043612
SETMAP 017406	TESTNO= 000010	T\$NS1 = 000005	T22OFL 026555	T24BF2 043720
SETU 022176	TEXASC 006525	T\$NS2 = 000002	T22PAC 026220	T24BOT 044563
SFFMSG 012172 G	TFCASC 006627	T\$PTNU= 000000	T22PK2 026330	T24BS0 043720
SFHERR 003703	TIMEXP 015642 G	T\$SAVL= 177777	T22POS 026352	T24BS1 043721
SFIERR 003650	TIMSGO 015670	T\$SEGL= 177777	T22RD 026346	T24CON 043732
SFIMSG 012124 G	TINERR 012111	T\$SUBN= 000003	T22RES 027172	T24DAT 043600
SFPTBL 002156 G	TMPBFR 002622 G	T\$TAGL= 177777	T22RT2 027264	T24DLY 043736
SIFLAG 003144 G	TNAM 016774	T\$TAGN= 010136	T22RWJ 026724	T24DSW 043610
SIMSG 012056	TRANST 002156 G	T\$TEMP= 000000	T22SSR 026360	T24DTA 044630
SKIPT 003370	TSBA = 000000 G	T\$TEST= 000010	T22S2 026342	T24EOT 044716
SOFINI 016064 G	TSBAH = 000001 G	T\$TSTM= 177777	T22S3 026344	T24ILA 044312
SPACE 010556 G	TSDB = 000000 G	T\$TSTS= 000001	T22TM 026630	T24LON 045662
SPM1 112500	TSDBH = 000001 G	T\$SAU = 010031	T22VCK 026777	T24LOO 034362
SPM4 112530	TSFCOD 007326	T\$SAUT= 010033	T22WLK 027052	T24LOP 045744

T24LOQ	044376	T25SSR	054004	T26WDC	073610	T27WDR	101550	T4	034322	G
T241OR	044012	T25SZ	053756	T26WDD	073520	T27WNG	101564	T4.1	034362	
T24NEF	043740	T25S2	053762	T26WDE	072713	T27WRF	104336	T4.10	041606	
T24NXM	044151	T25S3	053764	T26WDF	072521	T27WSS	103472	T4.11	042052	
T24OFL	045235	T25TM	054212	T26WNG	071756	T28BFR	110122	T4.12	042324	
T24PAC	043570	T25WB	053752	T26WSS	074011	T28BF2	110230	T4.13	042560	
T24PBP	046026	T25WDC	055077	T27AM3	103057	T28BOT	111067	T4.14	043060	
T24PK2	043700	T25WDE	054065	T27BA	103417	T28BS0	110230	T4.2	035004	
T24PK3	043710	T25WDR	053770	T27BFR	101432	T28BS1	110231	T4.3	035564	
T24RB	043712	T25WNG	054355	T27BF2	101540	T28CNT	110256	T4.4	036340	
T24RES	046250	T25WNH	054530	T27BOT	102431	T28CNU	110260	T4.5	037042	
T24RN	043726	T26AM3	073376	T27BS0	101540	T28CON	110252	T4.6	037506	
T24RNC	045115	T26BA	073736	T27BS1	101541	T28DAT	110110	T4.7	040142	
T24RT2	046342	T26BFR	071622	T27CNT	101556	T28DLY	110262	T4.8	040576	
T24RT3	046404	T26BF2	071730	T27CNU	101560	T28DSW	110120	T4.9	041170	
T24RWN	045046	T26BOT	072765	T27CON	101552	T28DTA	112034	T5	046436	G
T24SSR	044457	T26BS0	071730	T27DAT	101420	T28DTR	111752	T5.1	046466	
T24SZ	043716	T26BS1	071731	T27DLY	101562	T28IMV	110236	T5.2	047442	
T24S2	043722	T26CNT	071746	T27DSW	101430	T28LOO	104750	T5.3	050042	
T24S3	043724	T26CNU	071750	T27DTA	104416	T28LOQ	110644	T5.4	050516	
T24TM	044773	T26DAT	071610	T27EOT	102601	T28OFL	111260	T5.5	051162	
T24TRL	046114	T26DLY	071754	T27LON	103561	T28PAC	110100	T5.6	051722	
T24VCK	045447	T26DSW	071620	T27LOO	074736	T28PBP	110341	T5.7	052662	
T24WB	043712	T26DTA	073032	T27LOP	103643	T28PK2	110210	T5.8	053202	
T24WDC	045376	T26EOT	073120	T27LOQ	102225	T28PK3	110220	T6	055354	G
T24WDD	045310	T26LON	074100	T27LOR	102100	T28RB	110222	T6.1	055414	
T24WDE	044511	T26LOO	055414	T27NEF	104101	T28RDF	110424	T6.10	065034	
T24WDF	044235	T26LOP	074162	T27OFL	103126	T28RDG	110505	T6.11	065710	
T24WDG	044062	T26LOQ	072576	T27PAC	101410	T28RES	112156	T6.12	066560	
T24WDR	043730	T26LOR	072451	T27PBP	103725	T28RIB	110264	T6.13	067512	
T24WSS	045573	T26NEF	072044	T27PK2	101520	T28RN	110246	T6.14	070522	
T25BFR	053652	T26NEQ	074420	T27PK3	101530	T28RRM	111537	T6.15	071102	
T25BF2	053760	T26OFL	073445	T27RB	101532	T28RRN	111615	T6.2	056326	
T25BNC	054440	T26PAC	071600	T27RDF	101652	T28RRP	111674	T6.3	057174	
T25BOT	054145	T26PBP	074244	T27RES	104534	T28RT2	112250	T6.4	060066	
T25BS0	053760	T26PK2	071710	T27RN	101546	T28RT3	112312	T6.5	061014	
T25BS1	053761	T26PK3	071720	T27RNC	103004	T28RWN	111211	T6.6	061572	
T25CNT	054000	T26RB	071722	T27RRF	101721	T28SSR	110725	T6.7	062434	
T25CH2	053776	T26RDF	072126	T27RT2	104626	T28SZ	110226	T6.8	063306	
T25CON	053772	T26RES	074510	T27RT3	104670	T28S2	110232	T6.8	063306	
T25DAT	053640	T26RN	071736	T27RWN	102735	T28S3	110234	T6.9	064160	
T25DLY	054002	T26RNC	073323	T27SC	102016	T28TM	111134	T7	074676	G
T25DSW	053650	T26RRF	072175	T27SCF	104177	T28TMK	111465	T7.1	074736	
T25LOO	046466	T26RRG	072272	T27SSR	102306	T28VCK	111112	T7.2	075334	
T25NEF	054613	T26RSZ	071752	T27SZ	101536	T28WB	110222	T7.3	076116	
T25NET	054301	T26RT2	074602	T27S2	101542	T28WDC	111333	T7.4	076740	
T25OFL	055024	T26RT3	074644	T27S3	101544	T28WDE	110776	T7.5	077642	
T25PAC	053630	T26RWN	073254	T27TIM	102524	T28WDF	110567	T8	104714	G
T25PK2	053740	T26SC	072367	T27TM	102660	T28WDR	110250	T8.1	104750	
T25PK3	053750	T26SSR	072657	T27TRL	104013	T3	027332	T8.2	105330	
T25RB	053752	T26SZ	071726	T27TSA	104254	T3BFLG	003140	T8.3	105610	
T25RES	055166	T26S2	071732	T27VCK	103344	T3.1	027376	UAM =	000200	G
T25RIB	054673	T26S3	071734	T27WB	101532	T3.2	027750	UNITM	002172	G
T25RN	053766	T26TM	073177	T27WDC	103271	T3.3	030620	UNREC =	000006	
T25RT2	055260	T26TRL	074332	T27WDD	103201	T3.4	031440	USI	004121	
T25RT3	055322	T26VCK	073663	T27WDE	102342	T3.5	031654	WAITF	016340	G
T25RWN	054755	T26WB	071722	T27WDF	102150	T3.6	032106	WC.IFA=	000200	
								WC.IFE=	000002	

WC.IGO= 000001	WRTERR 005111	XSOILA= 000400	XSFALS= 000040	X2.UNI= 000007
WC.IRE= 000010	WRTMSG 005054	XSOILC= 001000	XSOFFS= 000400	X2.WCF= 002000
WC.IRW= 000004	WSMBK 021360 G	XSOLET= 020000	XSTRUE= 000020	X3.DCK= 000010
WC.IOT= 000100	XFERAS 016030	XSOMOT= 000200	X1.COR= 020000	X3.MBZ= 000006
WC.IIT= 000040	XNXM 016466	XSONEF= 002000	X1.DLT= 100000	X3.MDE= 177400
WC.ISR= 000020	XORBFO 007764	XSOONL= 000100	X1.MBZ= 017375	X3.OPI= 000100
WF.IED= 000010	XORFOR 010102	XSOPED= 000010	X1.RBP= 000400	X3.REV= 000040
WF.IER= 000004	XST0 = 000006 G	XSORLL= 010000	X1.SPA= 040000	X3.RIB= 000001
WF.IHI= 000200	XST1 = 000010 G	XSORLS= 040000	X1.UNC= 000002	X3.SPA= 000200
WF.IRF= 000040	XST2 = 000012 G	XSOTMK= 100000	X2.BUF= 000100	X3.TRF= 000020
WF.IWF= 000020	XST3 = 000014 G	XSOVCK= 000020	X2.EXT= 000200	X4.HSP= 100000
WF.IWR= 000100	XST4 = 000016 G	XSOWLE= 004000	X2.OPM= 100000	X4.MBZ= 017400
WF.I3R= 000002	XSOBOT= 000002	XSOWLK= 000004	X2.RCE= 040000	X4.RCE= 040000
WF.I4R= 000001	XSOEOT= 000001	XXCOMM 003112 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: 31544 WORDS (124 PAGES)

DYNAMIC MEMORY: 20346 WORDS (78 PAGES)

ELAPSED TIME: 00:49:06

CVTSCAO, CVTSCAO/-SP=SVC/ML, TSV1C, TSV22C, TSV3B, TSV4, TSV7A, TSV6