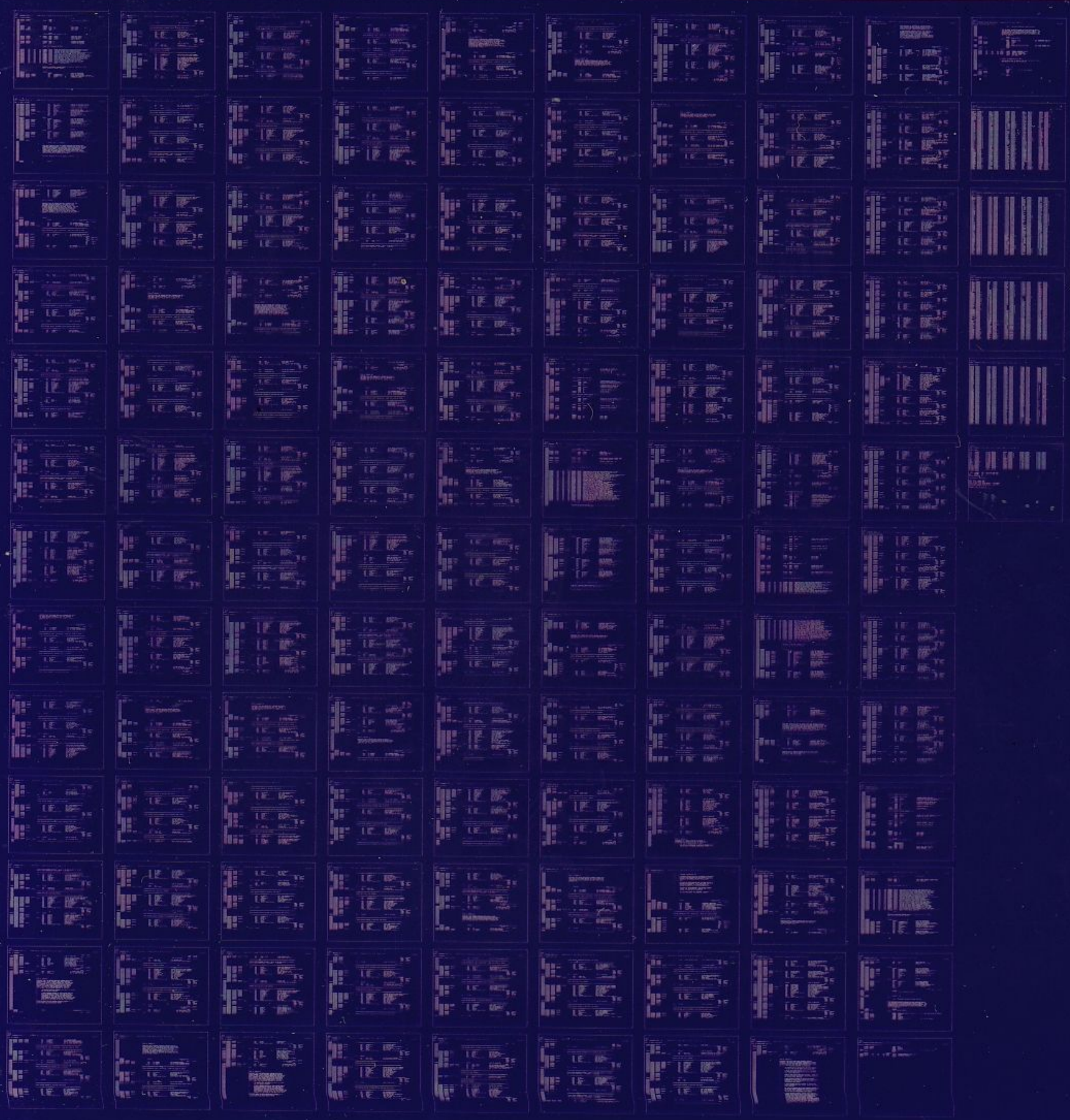


The main body of the document is a large grid of approximately 15 columns and 15 rows of small, illegible tables or diagrams. Each cell in the grid appears to contain a small table with multiple columns and rows of text, which is too small to read. The overall layout is a dense matrix of these small tables, likely representing a data matrix or control points for a system.



B1

D C W W
A [C

1
USER DOCUMENTATION

MACRO V05.03 Tuesday 28-Apr-87 08:48 Page 2

SEQ 000

.REM_
IDENTIFICATION

PRODUCT ID: AC-T097D-MC
PRODUCT TITLE: CVTSCDO TSV05 CTRL PART 3
DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PGG
DATE: JUNE 04, 1987

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) 1983, 1987 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 LSI-11 RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSV05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A LSI-11/23 SYSTEM (QBUS). 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

LSI-11 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 (HSAAS.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 HIERARCHY PREREQUISITES

FUNCTIONAL LSI-11 CENTRAL PROCESSOR AND MEMORY
 FUNCTIONAL CONSOLE TERMINAL
 FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR
 FUNCTIONAL DIAGNOSTIC LOADER/MONITOR (XXDP*)

1.5 ASSUMPTIONS

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

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.1.1 OPERATOR COMMANDS

THE TSV05 DIAGNOSTIC IS A LSI-11 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-D-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 M7455 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 LSI-11 AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

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

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

UNITS (D) ? 8<CR>

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

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

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

Q-FACTOR (0) 0 ? <CR>

UNIT 4

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 3<CR>

Q-FACTOR (0) 0 ? <CR>

UNIT 5

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 4<CR>

Q-FACTOR (0) 0 ? <CR>

UNIT 6

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 5<CR>

Q-FACTOR (0) 0 ? <CR>

UNIT 7

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 6<CR>

Q-FACTOR (0) 0 ? 1<CR>

UNIT 8

CSR ADDRESS (0) 160000<CR>

SUB-DEVICE # (0) ? 7<CR>

Q-FACTOR (0) 1 ? <CR>

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

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

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

```
# UNITS (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 AN 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 (LSI-11)

```
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 LSI-11 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 M7455 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 LSI-11 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 LSI-11 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 LSI-11 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 - JUNE 1983

REVISION B - JUNE 1984

UPDATED FOR NEW ORION CPU PROBLEMS, RELATING TO TIMEOUT
 ERRORS (#311 & #320) ON REWINDS.
 ELIMINATED CPU ID MESSAGE.

REVISION C - JUNE 1985

UPDATED TO PERFORM CORRECTLY WITH XXDP V2.1 EXTENDED
 MONITOR (DRSXM).

REVISION D - APRIL 1987

CHANGES MADE TO ALLOW DIAGNOSTICS TO WORK WITH
 THE NEW TSV05 MICROCODE (REVISION 2). THE NEW
 TSV05 MICROCODE ALWAYS IN EXTENDED FEATURE MODE.

```

845          .TITLE  TSV2 - PROGRAM HEADER
846          .SBTTL  PROGRAM HEADER
847 000000    .PSECT  ABS
848
854          .MCALL  SVC
855 000000    SVC          ; INITIALIZE SUPERVISOR MACROS
856          .ENABLE LC
857          .NLIST  BEX,CND
863 000000    .ENABL  ABS,AMA
864          .=2000
865 002000    BGNMOD  TSV2
      002000
866
867          TSV2::
868          ;**
869          ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
870          ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
871          ;--
872
873          POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
874 002000    HEADER  CVTSC,D,0,655.,0
      002000
      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  /D/
      002011          L$DEPO::          ;0
      002011          .ASCII  /0/
      002012          L$UNIT::          ;NUMBER OF UNITS
      002012          .WORD   0
      000000

```

PROGRAM HEADER

002014		L\$TIML::		;LONGEST TEST TIME
002014	001217	L\$HPCP::	.WORD 655.	;POINTER TO H.W. QUES.
002016		L\$SPCP::	.WORD L\$HARD	;POINTER TO S.W. QUES.
002016	113670	L\$HPTP::	.WORD L\$SOFT	;PTR. TO DEF. H.W. PTABLE
002020		L\$SPTP::	.WORD L\$HW	;PTR. TO S.W. PTABLE
002020	114022	L\$LADP::	.WORD L\$SW	;DIAG. END ADDRESS
002022		L\$STA::	.WORD L\$LAST	;RESERVED FOR APT STATS
002022	002146	L\$CO::	.WORD 0	
002024		L\$DTYP::	.WORD 0	;DIAGNOSTIC TYPE
002024	002156	L\$APT::	.WORD 0	;APT EXPANSION
002026		L\$DTP::	.WORD 0	;PTR. TO DISPATCH TABLE
002026	114404	L\$PRIO::	.WORD L\$DISPATCH	;DIAGNOSTIC RUN PRIORITY
002030		L\$ENVI::	.WORD 0	;FLAGS DESCRIBE HOW IT WAS SETUP
002030	000000	L\$EXP1::	.WORD 0	;EXPANSION WORD
002032		L\$MREV::	.WORD 0	;SVC REV AND EDIT #
002032	000000		.BYTE C\$REVISION	
002034		L\$EF::	.BYTE C\$EDIT	;DIAG. EVENT FLAGS
002034	000000	L\$SPC::	.WORD 0	
002036		L\$DEVP::	.WORD 0	; POINTER TO DEVICE TYPE LIST
002036	000000	L\$REPP::	.WORD L\$DVTYP	;PTR. TO REPORT CODE
002040		L\$EXP4::	.WORD L\$RPT	
002040	002124	L\$EXP5::	.WORD 0	
002042		L\$AUT::	.WORD 0	;PTR. TO ADD UNIT CODE
002042	000000	L\$DUT::	.WORD L\$AU	;PTR. TO DROP UNIT CODE
002044		L\$LUN::	.WORD L\$DU	;LUN FOR EXERCISERS TO FILL
002044	000000	L\$DESP::	.WORD 0	;POINTER TO DIAG. DESCRIPTION
002046		L\$LOAD::	.WORD L\$DESC	;GENERATE SPECIAL AUTOLOAD EMT
002046	000000	L\$ETP::	EMT E\$LOAD	;POINTER TO ERR_TBL
002050		L\$ICP::	.WORD 0	;PTR. TO INIT CODE
002050	003			
002051	003			
002052				
002052	000000			
002054				
002054	000000			
002056				
002056	000000			
002060				
002060	003374			
002062				
002062	023624			
002064				
002064	000000			
002066				
002066	000000			
002070				
002070	023312			
002072				
002072	023410			
002074				
002074	000000			
002076				
002076	003402			
002100				
002100	104035			
002102				
002102	000000			
002104				

I2

PROGRAM HEADER

002104	022516			
002106		L\$CCP::	.WORD	L\$INIT
002106	023576			;PTR. TO CLEAN-UP CODE
002110		L\$ACP::	.WORD	L\$CLEAN
002110	023516			;PTR. TO AUTO CODE
002112		L\$PRT::	.WORD	L\$AUTO
002112	022506			;PTR. TO PROTECT TABLE
002114		L\$TEST::	.WORD	L\$PROT
002114	000000			;TEST NUMBER
002116		L\$DLY::	.WORD	0
002116	000000			;DELAY COUNT
002120		L\$HIME::	.WORD	0
002120	000000			;PTR. TO HIGH MEM

875

DISPATCH TABLE

```

877
878
879
880
881
882 002122
      002122 000010
      002124
      002124 024406
      002126 025522
      002130 030202
      002132 035412
      002134 047656
      002136 056604
      002140 076166
      002142 106224
883
884
885
886
887
888
889
890 002144
      002144 000003
      002146
      002146
891 002146 172520
892 002150 000224
893 002152 000200
894 002154
      002154
895
896
897
898
899
900
901 002154
      002154 000004
      002156
      002156
902 002156 000000
903 002160 000000
904
905
906 002162 000017
907 002164 000310
908 002166
      002166
909 002166
910
919
920
925
931
932 002166

```

```

      .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

```

```

      .SBTTL DEFAULT HARDWARE P-TABLE
      ;**
      ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
      ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
      ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
      ;--
      BGNHW DFPTBL ;DEFAULT HARD-P-TABLE
      .WORD L10000-L$HW/2
L$HW::
DFPTBL::
      .WORD 172520 ; 1ST (OF 2) REGISTERS.
      .WORD 224 ; INTERRUPT VECTOR
      .WORD PRI04 ; INTERRUPT PRIORITY.
      ENDSW
L10000:

```

```

      .SBTTL SOFTWARE P-TABLE
      ;**
      ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
      ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
      ;--
      BGNSW SFPTBL
      .WORD L10001-L$SW/2
L$SW::
SFPTBL::
TRANSTST:: .WORD 0 ; ENABLE TEST OF TRANSPORT(S) IF =1
NOITS:: .WORD 0 ; INHIBIT ITERATION OPTION.
      ; ... 0 = ITERATE.
      ; ...NZ = INHIBIT ITERATE.
LERRMAX:: .WORD 15. ; LOCAL (PER TEST) ERROR LIMIT
GERRMAX:: .WORD 200. ; GLOBAL (PER UNIT) ERROR LIMIT
      ENDSW
L10001:
      ENDMOD
      .TITLE TSV3 - GLOBAL AREAS
      .SBTTL GLOBAL EQUATES SECTION
      BGNMOD TSV3

```


K2

GLOBAL EQUATES SECTION

002166
933
934
935
936
937
938
939
940
944 002166

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

001000	BIT9== BIT09
000400	BIT8== BIT08
000200	BIT7== BIT07
000100	BIT6== BIT06
000040	BIT5== BIT05
000020	BIT4== BIT04
000010	BIT3== BIT03
000004	BIT2== BIT02
000002	BIT1== BIT01
000001	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

L2

GLOBAL EQUATES SECTION

000140
000100
000040
000000

PRI03== 140
PRI02== 100
PRI01== 40
PRI00== 0

; OPERATOR FLAG BITS

000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000

EVL== 4
LOT== 10
ADR== 20
IDU== 40
ISR== 100
UAM== 200
BOE== 400
PNT== 1000
PRI== 2000
IXE== 4000
IBE== 10000
IER== 20000
LOE== 40000
HOE== 100000

945
946 002166

000250
177572
177574
177576
172516

KT11
.SBTTL MEMORY MANAGEMENT DEFINITIONS
;*KT11 VECTOR ADDRESS
MMVEC= 250
;*KT11 STATUS REGISTER ADDRESSES
SR0= 177572
SR1= 177574
SR2= 177576
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

; DEFINE MEMORY MANAGEMENT REGISTERS

MEMORY MANAGEMENT DEFINITIONS

```
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
```

MEMORY MANAGEMENT DEFINITIONS

```
172300      .ENDC
172302      ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
172304      KIPDR0= 172300
172306      KIPDR1= 172302
172310      KIPDR2= 172304
172312      KIPDR3= 172306
172314      KIPDR4= 172310
172316      KIPDR5= 172312
            KIPDR6= 172314
            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
172340      ;*KERNEL "I" PAGE ADDRESS REGISTERS
172342      KIPAR0= 172340
172344      KIPAR1= 172342
172346      KIPAR2= 172344
172350      KIPAR3= 172346
172352      KIPAR4= 172350
172354      KIPAR5= 172352
172356      KIPAR6= 172354
            KIPAR7= 172356
            .IF NB
            ;*KERNEL "D" PAGE ADDRESS REGISTERS
            KDPAR0= 172360
            KDPAR1= 172362
            KDPAR2= 172364
            KDPAR3= 172366
            KDPAR4= 172370
            KDPAR5= 172372
            KDPAR6= 172374
            KDPAR7= 172376
            .ENDC
```

TSV05 REGISTER AND PACKET DEFINITIONS

```

951                      .SBTTL TSV05 REGISTER AND PACKET DEFINITIONS
952
953                      ;
954                      ; SOME GENERAL EQUATES.
955                      ;
956
957                      000004 ERRVEC==      4           ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.
958                      000060 TTIVEC==     60           ; INTERRUPT VECTOR FOR CONSOLE INPUT
959                      177560 TTICSR==    177560        ; BUS ADDRESS OF CONSOLE INPUT
960                      177562 TTIBFR==    177562        ; CONSOLE INPUT DATA BUFFER
961                      177520 BDVPCR==    177520        ; BDV11 PAGE CONTROL REGISTER
962
963                      ;+
964                      ;BIT DEFINITIONS FOR TSSR REGISTER
965                      ;-
966
967                      100000 SC=          BIT15        ;SPECIAL CONDITION
968                      040000 BIE=         BIT14        ;BUS INTERFACE ERROR
969                      020000 SCE=         BIT13        ;SANITY CHECK ERROR
970                      010000 RMR=         BIT12        ;MODIFICATION REFUSED
971                      004000 NXM=         BIT11        ;NONEXISTANT MEMORY ERROR
972                      002000 NBA=         BIT10        ;NEED BUFFER ADDRESS
973                      001400 HIADDR=     BIT9!BIT8     ;EXTENDED ADDRESS BITS
974                      000200 SSR=         BIT7         ;SUB SYSTEM READY
975                      000100 OFL=         BIT6         ;OFF LINE BIT
976                      000060 FATERR=     BIT4!BIT5    ;FATAL TERMINATION ERROR CODES
977                      000016 TERCLS=     BIT3!BIT2!BIT1 ;TERMINATION CODES
978
979                      ;+
980                      ;
981                      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
982                      ;(XST0)
983                      ;
984                      ;-
985
986                      100000 XSOTMK=     BIT15        ;TAPE MARK DETECTED
987                      040000 XSORLS=     BIT14        ;RECORD LENGTH SHORT
988                      020000 XSOLET=     BIT13        ;LOGICAL END OF TAPE
989                      010000 XSORLL=     BIT12        ;RECORD LENGTH LONG
990                      004000 XSOWLE=     BIT11        ;WRITE LOCK ERROR
991                      002000 XSONEF=     BIT10        ;NON EXECUTABLE FUNCTION
992                      001000 XSOILC=     BIT9         ;ILLEGAL COMMAND
993                      000400 XSOILA=     BIT8         ;ILLEGAL ADDRESS
994                      000200 XSOMOT=     BIT7         ;TAPE IN MOTION
995                      00010C XSOONL=     BIT6         ;TRANSPORT ON LINE
996                      000040 XSOIE=      BIT5         ;INTERRUPT ENABLE
997                      000020 XSOVCK=     BIT4         ;VOLUME CHECK BIT
998                      000010 XSOPED=     BIT3         ;PHASE ENCODED DRIVE
999                      000004 XSOWLK=     BIT2         ;WRITE LOCKED
1000                     000002 XSOBOT=     BIT1         ;BEGINNING OF TAPE
1001                     000001 XSOEOT=     BIT0         ;END OF TAPE

```

C3

TSV05 REGISTER AND PACKET DEFINITIONS

```

1003      ;*
1004      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
1005      ;(XST1)
1006      ;-
1007      100000 X1.DLT = BIT15      ;DATA LATE
1008      040000 X1.SPARE= BIT14      ;NOT USED
1009      020000 X1.COR  = BIT13      ;CORRECTABLE DATA ERROR
1010      017375 X1.MBZ  = BIT12·BIT11·BIT10·BIT9·BIT7·BIT6·BIT5·BIT4·BIT3·BIT2·BIT0 ;ALWAYS 0
1011      000400 X1.RBP  = BIT8      ;READ BUS PARITY ERROR
1012      000002 X1.UNC  = BIT1      ;UNCORRECTABLE DATA OR HARD ERROR
1013
1014      ;*
1015      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
1016      ;(XST2)
1017      ;-
1018      100000 X2.OPM  = BIT15      ;OPERATION IN PROGRESS (TAPE MOVING)
1019      040000 X2.RCE  = BIT14      ;RAM CHECKSUM ERROR
1020      035400 X2.SPARE= BIT13·BIT12·BIT11·BIT9·BIT8 ;NOT USED BY TSV05 (ALWAYS=0)
1021      002000 X2.WCF  = BIT10      ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
1022      000200 X2.EXTF = BIT7      ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
1023      000100 X2.BUFE = BIT6      ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
1024      000077 X2.REV  = 000077 ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
1025      000007 X2.UNIT = BIT2·BIT1·BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
1026
1027      ;*
1028      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
1029      ;(XST3)
1030      ;-
1031      177400 X3.MDE  = 177400 ;MICRO-DIAGNOSTIC ERROR CODE
1032      000200 X3.SPARE= BIT7      ;NOT USED BY TSV05
1033      000100 X3.OPI  = BIT6      ;OPERATION INCOMPLETE
1034      000040 X3.REV  = BIT5      ;REVERSE
1035      000020 X3.TRF  = BIT4      ;TRANSPORT RESPONSE FAILURE
1036      000010 X3.DCK  = BIT3      ;DENSITY CHECK
1037      000006 X3.MBZ  =BIT2·BIT1 ;NOT USED ALWAYS 0
1038      000001 X3.RIB  = BIT0      ;REVERSE INTO BOT
1039
1040      ;*
1041      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
1042      ;(XST4)
1043      ;-
1044      100000 X4.HSP  = BIT15      ;HIGH SPEED
1045      040000 X4.RCE  = BIT14      ;RETRY COUNT EXCEEDED
1046      020000 X4.TSM  = BIT13      ;TRANSPORT SPECIAL MODE
1047      017400 X4.MBZ  = BIT12·BIT11·BIT10·BIT9·BIT8 ;NOT USED ALWAYS 0
1048      000377 X4.WRC  = 000377 ;WRITE RETRY COUNT FIELD
1049
1050      ;*
1051      ;
1052      ;TSSR TERMINATION CODES (BIT 0-2)
1053      ;
1054      ;-
1055
1056      000006 TSREJ= 3*2      ;COMMAND REJECTED
1057      000006 UNREC= 6      ;UNRECOVERABLE ERROR

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1059      ;*
1060      ;
1061      ;DEVICE REGISTER OFFSETS
1062      ;
1063      ;-
1064
1065      000000      TSBA== 0
1066      000000      TSDB== 0      ;TSDB/TSBA REGISTER
1067      000001      TSBAH== 1
1068      000001      TSDBH== 1      ;TSDB/TSBA REGISTER HIGH BYTE
1069      000002      TSSR== 2      ;TSSR REGISTER
1070      000003      TSSRH== 3      ;TSSR REGISTER HIGH BYTE
1071
1072      ;*
1073      ; TSDB ADDRESS BIT DEFINITIONS
1074      ;-
1075      000003      A1716 = BIT1:BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
1076
1077      ;*
1078      ; COMMAND DEFINITIONS
1079      ;-
1080      000017      P.GETSTAT      = 17      ;GET STATUS
1081      000013      P.INIT        = 13      ;INITIALIZE
1082      000012      P.CONTROL     = 12      ;CONTROL COMMANDS
1083      000011      P.FORMAT      = 11      ;FORMAT
1084      000010      P.POSITION    = 10      ;POSITION
1085      000006      P.WRTSUB      = 6       ;SUBSYSTEM WRITE
1086      000005      P.WRITE       = 5       ;WRITE
1087      000004      P.WRTCHAR     = 4       ;WRITE CHARACTERISTICS
1088      000001      P.READ        = 1       ;READ
1089
1090      ;*
1091      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
1092      ;-
1093      100000      P.ACK          = BIT15      ;BUFFER AVAIL FOR CONTROLLER
1094      040000      P.CVC          = BIT14      ;CLEAR VOLUME CHECK
1095      020000      P.OPP          = BIT13      ;REVERSE SEQUENCE OF DATA BITS
1096      010000      P.SWB         = BIT12      ;SWAP BYTES IN MEMORY
1097      007400      P.MODE        = BIT11:BIT10:BIT9:BIT8 ;EXTENDED COMMAND MODE FIELD
1098      000200      P.IE          = BIT7       ;INTERRUPT ENABLE
1099      000140      P.FMT         = BIT6:BIT5   ;PACKET HEADER TYPE (ALWAYS=0)
1100      000037      P.CMD         = 37        ;MAJOR COMMAND FIELD
1101
1102      ;*
1103      ; CONTROL COMMAND MODE CODES
1104      ;-
1104      000000      PC.RELEASE     = 0*256.    ;RELEASE BUFFER
1105      000400      PC.REWIND      = 1*256.    ;REWIND
1106      001000      PC.NOOP        = 2*256.    ;NO-OP
1107      002000      PC.IEREW       = 4*256.    ;REWIND IMMEDIATE INTERRUPT
1108      002400      PC.ERASE       = 5*256.    ;SECURITY ERASE

```

E3

TSV05 REGISTER AND PACKET DEFINITIONS

```

1110
1111      ;*
1112      ; CONTROLLER RAM DEFINITIONS
1113      ;-
1113      000167 RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
1114      000200 RMCHEND = 200     ;CHARACTERISTICS IO DATA END RAM ADDRESS
1115      000201 RMPKTBEG= 201     ;COMMAND PACKET BEGIN RAM ADDRESS
1116      000210 RMPKTEND= 210     ;COMMAND PACKET END RAM ADDRESS
1117      000215 RMMGBEG= 215     ;MESSAGE BUFFER BEGIN RAM ADDRESS
1118      000234 RMMGEND= 234     ;MESSAGE BUFFER END RAM ADDRESS
1119      ;*
1120      ;
1121      ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
1122      ;
1123      ;-
1124
1125      000006 XST0== 6          ;EXTENDED STATUS REGISTER 0 (WORD 4)
1126      000010 XST1== 8.        ;EXTENDED STATUS REGISTER 1 (WORD 5)
1127      000012 XST2== 10.       ;EXTENDED STATUS REGISTER 2 (WORD 6)
1128      000014 XST3== 12.       ;EXTENDED STATUS REGISTER 3 (WORD 7)
1129      000016 XST4== 14.       ;EXTENDED STATUS REGISTER 4 (WORD 8)
1130
1131      ;*
1132      ;
1133      ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
1134      ;
1135      ;-
1136
1137      000002      PKLOW  = 2          ;LOW ORDER CHARACTERISTIC DATA POINTER
1138      000004      PKHI   = 4          ;HIGH ORDER CHARACTERISTIC DATA POINTER
1139      000006      PKBCNT = 6          ;NUMBER OF BYTES IN DATA PACKET
1140
1141      000010      EXBCNT=10          ;NUMBER OF BYTES IN EXTENDED DATA PACKET
1142
1143      ;*
1144      ;DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
1145      ;-
1146      000000      BSELO  = 0          ;BYTE 0
1147      000001      BSEL1  = 1          ;BYTE 1
1148      000002      SEL2   = 2          ;WORD 2
1149      000004      SELDATA = 4          ;WORD 3

```


TSV05 REGISTER AND PACKET DEFINITIONS

```

1151
1152      ;*
1153      ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
1154      ;-
1155      000000      PW.NOP          = 0          ;NO-OP
1156      000001      PW.RDRAM       = 1          ;READ RAM
1157      000002      PW.WTRAM       = 2          ;WRITE RAM
1158      000003      PW.RFIFO       = 3          ;READ FIFO
1159      000004      PW.WFIFO       = 4          ;WRITE FIFO
1160      000005      PW.RDSTAT      = 5          ;READ STATUS
1161      000006      PW.WCTL        = 6          ;WRITE TAPE CONTROL
1162      000007      PW.WFMT        = 7          ;WRITE TAPE FORMAT
1163      000010      PW.WMISC       = 10         ;WRITE MISCELLANEOUS
1164      000011      PW.WNPR       = 11         ;WRITE NPR CONTROL
1165      000020      PW.D22        = 20         ;DO MICROTEST 22
1166      000021      PW.D11        = 21         ;DO MICROTEST 11
1167      000022      PW.D13        = 22         ;DO MICROTEST 13
1168      000023      PW.NO1311     = 23         ;DISABLE MICROTEST 11 AND 13
1169      000024      PW.RDXT        = 24         ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
1170
1171      ;*
1172      ;BSEL1 CODES FOR WRITE TAPE CONTROL
1173      ;-
1174      000200      WC.IFAD        = BIT7       ;IFAD - FORMATTER ADDRESS
1175      000100      WC.IOTAD       = BIT6       ;ITADO - TRANSPORT ADDRESS BIT 0
1176      000040      WC.I1TAD       = BIT5       ;ITAD1 - TRANSPORT ADDRESS BIT 1
1177      000020      WC.ISRESV     = BIT4       ;IRESV5 - RESERVED #5
1178      000010      WC.IREW       = BIT3       ;IREW - REWIND
1179      000004      WC.IRWU       = BIT2       ;IRWU - REWIND AND UNLOAD
1180      000002      WC.IFEN       = BIT1       ;IFEN - FORMATTER ENABLE
1181      000001      WC.IGO        = BIT0
1182
1183      ;*
1184      ;BSEL1 CODES FOR WRITE FORMAT
1185      ;-
1186      000200      WF.IHISP       = BIT7       ;IHISP - HIGH SPEED
1187      000100      WF.IWRT       = BIT6       ;IWRT - WRITE
1188      000040      WF.IREV       = BIT5       ;IREV - REVERSE
1189      000020      WF.IWFM       = BIT4       ;IWFM - WRITE FILE MARK
1190      000010      WF.IEDIT     = BIT3       ;IEDIT - EDIT
1191      000004      WF.IERASE     = BIT2       ;IERASE - ERASE
1192      000002      WF.I3RESV    = BIT1       ;IRESV3 - RESERVED #3
1193      000001      WF.I4RESV    = BIT0       ;IRESV4 - RESERVED #4
1194
1195      ;*
1196      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
1197      ;-
1198      000200      MS.EXT         = BIT7       ;INVERT SENSE OF EXTENDED FEATURES SWITCH
1199      000020      MS.RSFIFO     = BIT4       ;RESET FIFO AND INPUT PARITY ERRORR
1200      000010      MS.RSTAPE    = BIT3       ;RESET TAPE STATUS IN 2 FLIP-FLOPS
1201      000006      MS.ATTN      = BIT2!BIT1  ;ATTENTION TRIGGER FIELD
1201      000001      MS.RSD       = BIT0       ;RESET TIMER A,B THEN DELAY TIMES IN SEL2

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1203
1204      ;*
1205      ; MS.ATTN SUBCODES
1206      ;-
1206      000000      MSA.NOP = 0*2      ;NO-OP (NOTHING TRIGGERED)
1207      000002      MSA.VOL = 1*2      ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
1208      000004      MSA.NRAM= 2*2      ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
1209      000006      MSA.FRAME= 3*2     ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
1210
1211      ;*
1212      ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
1213      ;-
1213      000200      NP.IR      = BIT7      ;INTERRUPT REQUEST (0-1 TRANSITION)
1214      000100      NP.OUT     = BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
1215      000040      NP.LOOP    = BITS      ;ENABLE TRANSPORT LOOPBACK
1216      000020      NP.WRP     = BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
1217
1218      ;*
1219      ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
1220      ;-
1221      000200      S2.DIM      = BIT7      ;WORD #9 BYTE 2 DATA IN MISS
1222      000100      S2.ILW     = BIT6      ; ILW H
1223      000040      S2.OUTRDY   = BITS      ; OUT RDY H
1224      000020      S2.INRDY   = BIT4      ; IN RDY H
1225      000010      S2.ATIMR   = BIT3      ; TIMER A FLAG H
1226      000004      S2.BTIMR   = BIT2      ; TIMER B FLAG H
1227      000003      S2.UNDEF    = BIT1-BIT0 ;(UNDEFINED)
1228      100000      S1.PARIN    = BIT15     ;WORD #8 BYTE 1 PARIN H
1229      040000      S1.I2RESV  = BIT14     ; IRESV2
1230      020000      S1.I1RESV  = BIT13     ; IRESV1
1231      010000      S1.IEOT     = BIT12     ; IEOT L
1232      004000      S1.IIDENT   = BIT11     ; IIDENT H
1233      002000      S1.ICER     = BIT10     ; ICER H
1234      001000      S1.IFMK     = BIT9      ; IFMK H
1235      000400      S1.IHER     = BIT8      ; IHER H
1236      000200      S0.ISPEED   = BIT7      ;WORD #8 BYTE 0 ISPEED H
1237      000100      S0.IRDY     = BIT6      ; IRDY L
1238      000040      S0.IONL     = BITS      ; IONL L
1239      000020      S0.ILDP     = BIT4      ; ILDP L
1240      000010      S0.IDBY     = BIT3      ; IDBY L
1241      000004      S0.IRWD     = BIT2      ; IRWD L
1242      000002      S0.IFBY     = BIT1      ; IFBY L
1243      000001      S0.IFPT     = BIT0      ; IFPT L

```

SPECIAL MACROS AND OPDEFS.

```

1245             .SBTTL SPECIAL MACROS AND OPDEFS.
1246
1247             ;*
1248             ;SAVE GENERAL REGS 1 TO 5
1249             ;-
1250
1251             .MACRO SAVREG
1252             JSR   R5,REGSAV
1253             .ENDM
1254
1255             ;*
1256             ; MACRO TO FORCE AN ERROR
1257             ;-
1258             .MACRO FORCERROR TAG,NOTSSR
1259             .NLIST
1260             .IIF NDF LISTALL, .NLIST
1261             .LIST
1262             .IF B NOTSSR
1263             MOV   TSSR(R5),R1 ;READ TSSR
1264             .ENDC
1265             MOV   FORCER,FORCER ;IS FORCER SET? (LEAVE C BIT ALONE)
1266             BNE  TAG           ;BR IF YES
1267             .NLIST
1268             .IIF NDF LISTALL, .LIST
1269             .LIST
1270             .ENDM
1271
1272             ;*
1273             ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
1274             ; WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
1275             ; SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
1276             ; FORCER TO 177777
1277             ; TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
1278             ;-
1279             .MACRO FORCEEXIT TAG
1280             .NLIST
1281             .IIF NDF LISTALL, .NLIST
1282             .LIST
1283             MOV   FORCER,FORCER ;IS FORCER NEGATIVE?
1284             BMI  TAG           ;BR IF YES
1285             .NLIST
1286             .IIF NDF LISTALL, .LIST
1287             .LIST
1288             .ENDM
1289             ;*
1290             ; MACRO TO INCREMENT ERROR COUNTS
1291             ;-
1292             .MACRO NEXT.ERRNO
1293             .NLIST
1294             ;;;.IIF NDF LISTALL, .NLIST
1295             ERRNO=ERRNO+1
1296             ;;;.IIF NDF LISTALL, .LIST
1297             .LIST
1298             .ENDM

```

SPECIAL MACROS AND OPDEFS.

```

1300
1301      ;*
1302      ;MACRO TO PERFORM XOR
1303      ;-
1304      .MACRO XOR A,B
1305      MOV A,-(SP)
1306      BIC B,(SP)
1307      BIC A,B
1308      BIS (SP),B
1309      .ENDM
1310
1311      000000      EN=0 ; INITIALIZE ERROR NUMBER
1312      .SBTTL FORCER - FORCE ERROR FLAG
1313
1314      ;
1315      ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
1316      ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
1317      ;
1318
1319      002166 000000 FORCER:: 0 ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
1320      ; - BY THE MACRO "IFERROR"). AN ERROR NEED NOT -
1321      ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.

```

GLOBAL DATA SECTION

```

1323                                     .SBTTL GLOBAL DATA SECTION
1324
1325                                     ;**
1326                                     ;THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1327                                     ;IN MORE THAN ONE TEST.
1328                                     ;--
1329
1330                                     ;
1331                                     ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
1332                                     ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
1333                                     ;
1334 002170 000000 EPRTSW::          .WORD 0          ;PRINT SWITCH
1335 002172 000000 UNITN::          .WORD 0          ;UNIT # UNDER TEST.
1336 002174 000000 QVP::            .WORD 0          ;QUICK VERIFY FLAG.
1337 002176 000000 CSRADDR::        .WORD 0          ;ADDRESS OF CSR FOR CURRENT DEVICE
1338 002200 000224 IVEC::            .WORD 224        ;INTERRUPT VECTOR
1339 002202 000200 IPRI::            .WORD PRI04      ;INTERRUPT PRIORITY.
1340 002204 000000 TSTCNT::          .WORD 0          ;NUMBER OF TESTS RUN IN THIS PASS
1341 002206 000000 LOOPCNT::        .WORD 0          ;REMAINING ITERATION COUNT FOR TEST
1342 002210 000000 DEVCNT::          .WORD 0          ;NUMBER OF DEVICE UNDER TEST
1343 002212 000000 FATFLG::          .WORD 0          ;SET IF FATAL ERROR IS DETECTED IN TEST
1344 002214 000000 INTRECV::        .WORD 0          ;SET IF TAPE INTERRUPT WAS RECEIVED
1345 002216 000000 EXTFEA::          .WORD 0          ;EXTENDED FEATURES SOFTWARE SW 0=OFF;1=ON
1346 002220 000000 REV::            .WORD 0          ;REV = ( 1 = old microcode, 2= newmicrocode)
1347 002222 000000 BENBSW::          .WORD 0          ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
1348 002224 000000 EXPD::            .WORD 0          ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
1349 002226 000000 RECV::            .WORD 0          ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
1350 002230 000000 ERRHI::          .WORD 0          ;HIGH ADDRESS MEMORY ERROR
1351 002232 000000 ERRLO::          .WORD 0          ;LOW ADDRESS MEMORY ERROR
1352 002234 000000 RAMDATA::         .BLKW 16.        ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
1353 002274 000000 RAMSIZ::          .WORD 0          ;RAM DATA SIZE FOR PRAMPKT ROUTINE
1354 002276 000000 RCVHIADD::        .WORD 0          ;RECEIVED BUFFER HIGH ADDRESS
1355 002300 000000 RCVLOADD::        .WORD 0          ;RECEIVED BUFFER LOW ADDRESS
1356 002302 000000 COUNT::          .WORD 0          ;TEST COUNT PATTERN
1357 002304 000000 DATA::          .WORD 0          ;TEST DATA
1358 002306 000000 TSTFLAG::         .WORD 0          ;TEST FLAG WORD
1359 002310 000000 TSTPTR::          .WORD 0          ;TSTBLK POINTER
1360 002312 000000 PRMNO::          .WORD 0          ;PRINT ROUTINE TEMP
1361 002314 000000 EXPMSG::          .BLKB 100.       ;EXPECTED MESSAGE BUFFER DATA
1362 002460 000000 RECMSG::          .BLKB 100.       ;RECEIVED MESSAGE BUFFER DATA
1363 002624 000000 TMPBFR::          .BLKB 80.        ;TEMPORARY STORAGE FOR PRINT

```

K3

TSTBLK - TEST DATA TABLE

```

1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381 002744
1382 002744 000000
1383 002746 177777
1384 002750 000001
1385 002752 000002
1386 002754 000004
1387 002756 000010
1388 002760 000020
1389 002762 000040
1390 002764 000100
1391 002766 000200
1392 002770 000400
1393 002772 001000
1394 002774 002000
1395 002776 004000
1396 003000 010000
1397 003002 020000
1398 003004 040000
1399 003006 100000
1400 003010 177776
1401 003012 177775
1402 003014 177773
1403 003016 177767
1404 003020 177757
1405 003022 177737
1406 003024 177677
1407 003026 177577
1408 003030 177377
1409 003032 176777
1410 003034 175777
1411 003036 173777
1412 003040 167777
1413 003042 157777
1414 003044 137777
1415 003046 077777
1416 003050 125252
1417 003052 052525
1418 003054

```

.SBTTL TSTBLK - TEST DATA TABLE

```

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

```

```

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

TBLEND==.

```

L3

GLOBAL ENVIRONMENT STORAGE

```

1420          .SBTTL GLOBAL ENVIRONMENT STORAGE
1421          ;
1422          ;STORAGE FOR DEVICE REGISTERS
1423          ;
1424 003054 000000 100000 000000 DUMMY: 0,100000,0,0 ;DUMMY DEVICE REGISTERS...
1425 003064 000000 000000 000000      0,0,0,0,0,0,0,0 ;...FOR MULTI-UNIT CHECKOUT.
1426          ;
1427          ;
1428 003104 000000 DUFLG::          .WORD 0          ;"DROPPED UNIT" FLAG.
1429          ;INHIBITS CODE IN "CLEAN-UP".
1430 003106 000000 NODEV::          .WORD 0          ;FLAG TO SAY NO DEVICE.
1431          ;
1432 003110 000000 TEMP1::          .WORD 0          ;SOME TEMP LOCATIONS.
1433 003112 000000 TEMP2::          .WORD 0
1434 003114 000000 XXCOMM::          .WORD 0          ;XXDP+ COMM BLOCK POINTER.
1435 003116 000000 FREE::          .WORD 0          ;1ST FREE MEMORY ADDRESS...
1436 003120 000000 FRESIZ::          .WORD 0          ;...AND SIZE (IN WORDS).
1437 003122 000000 FREEHI: .WORD 0          ;LAST WORD IN FREE SPACE
1438 003124 000000 KTFLG::          .WORD 0          ;KT11, MEM AVAIL FLAG -
1439          ;- .WORD 0 = <24K OR NO KT -
1440          ;- NZ = >24K AND KT.
1441 003126 000000 KTENABLE::          .WORD 0          ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
1442 003130 000000 NXMFLG::          .WORD 0          ;SET IF WE CAN TEST CLEARED OTHERWISE
1443 003132 000000 NXML0::          .WORD 0          ;NXM LO ADDRESS BITS
1444 003134 000000 NXMHI::          .WORD 0          ;NXM HI ADDRESS BITS FOR DAL'S 16-21
1445 003136 000000 T23A::          .WORD 0          ;11/23A FLAG
1446 003140 000000 T23B::          .WORD 0          ;11/23B FLAG
1447 003142 000000 T3BFLG::          .WORD 0          ;TEST 3B FLAG +0
1448 003144 002000 PST32W::          .WORD 2000          ;32W BLOCK ADDRESS FOR 32K START
1449 003146 000000 SIFLAG::          .WORD 0
1450 003150 000000 BADDAT::          .WORD 0          ;ACTUAL DATA
1451 003152 000000 GDDAT::          .WORD 0          ;EXPECTED DATA
1452 003154 000000 LOOPFL::          .WORD 0
1453 003156          CTAB::          ;CONFIGURATION TABLES.
1454 003156 000000 CTABM::          .WORD 0          ;CONFIG WORK.
1455 003160          .WORD 0
1456 003162          .WORD 0
1457 003164          .WORD 0
1458 003166 177777          .WORD -1          ;END OF MEM TABLE.
1459 003170          CTABE::          ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
1460          ;
1461          ; 0 = UNIT NOT TESTED
1462          ; 100000 = UNIT ONLINE, NO ERRORS
1463          ; 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
1464          ; 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
1465          ; 160001 = UNIT DROPPED, NOT IDLE AT START
1466          ; 14XXXX = UNIT DROPPED, ENCOUNTERED XXXX ERRORS
1467          ;
1468          ;
1469 003170          ERTABL:          .BLKW 64.
1470 003370 000000          ERTABE:          .WORD 0
1471          ;
1472 003372 000000          SKIPT:          .WORD 0          ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST

```

GLOBAL TEXT MESSAGES

```

1474                                     .SBTTL GLOBAL TEXT MESSAGES
1475                                     ;**
1476                                     ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1477                                     ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1478                                     ; MORE THAN ONE TEST.
1479                                     ;--
1480                                     ;*
1481                                     ;NAMES OF DEVICES SUPPORTED
1482                                     ;-
1483 003374                                DEVTYP <TSV05>
      003374                                L$DVTYP::
      003374                                .ASCIZ /TSV05/
      124 123 126                          .EVEN

1484
1499
1500                                     ;*
1501                                     ;TEST DESCRIPTION
1502                                     ;-
1502 003402                                DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - CHK CABLES-TRANSPORT IF ERR ****>
      003402                                L$DESC::
      003402                                .ASCIZ /**** TSV05 LOGIC DIAGNOSTIC - CHK CABLES-TRANSPORT IF ERR ****/
      052 052 052                          .EVEN

1510
1511                                     ;*
1512                                     ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
1513                                     ;-
1514 003502 003542 003545 003551          TSSRBIT::
1515 003522 003603 003607 003613          .WORD 1$,2$,3$,4$,5$,6$,7$,8$
1516 003542 123 103 000 1$: .ASCIZ 'SC'
1517 003545 102 111 105 2$: .ASCIZ 'BIE'
1518 003551 123 103 105 3$: .ASCIZ 'SCE'
1519 003555 122 115 122 4$: .ASCIZ 'RMR'
1520 003561 116 130 115 5$: .ASCIZ 'NXM'
1521 003565 116 102 101 6$: .ASCIZ 'NBA'
1522 003571 102 111 124 7$: .ASCIZ 'PIT9'
1523 003576 102 111 124 8$: .ASCIZ 'BIT8'
1524 003603 123 123 122 9$: .ASCIZ 'SSR'
1525 003607 117 106 114 10$: .ASCIZ 'OFL'
1526 003613 102 111 124 11$: .ASCIZ 'BIT5'
1527 003620 102 111 124 12$: .ASCIZ 'BIT4'
1528 003625 102 111 124 13$: .ASCIZ 'BIT3'
1529 003632 102 111 124 14$: .ASCIZ 'BIT2'
1530 003637 102 111 124 15$: .ASCIZ 'BIT1'
1531 003644 102 111 124 16$: .ASCIZ 'BIT0'
1532                                     .EVEN
1533 003652 124 123 123 SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
1534 003705 124 123 123 SFHERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
1535 003740 040 040 116 NXR: .ASCIZ / NON-EXISTANT DEVICE REGISTER/
1536 003777 045 101 040 NXR: .ASCIZ /%A ADDRESS: %06/
1537 004020 045 101 040 TSSX: .ASCII /%A TSBA,TSSR EXP'D: %06%A,%06%N/
1538 004060 045 101 040 TSSX: .ASCII /%A TSBA,TSSR REC'D: %06%A,%06/
1539 004117 045 116 045 FUSI: .ASCII /%N%A/
1540 004123 040 040 125 USI: .ASCIZ / UNEXPECTED INTERRUPT/
1541 004152 040 040 111 NSI: .ASCIZ / INTERRUPT EXPECTED, NOT RECEIVED/
1542 004215 045 116 045 FNOINTR: .ASCII /%N%A/
1543 004221 040 040 116 NOINTR: .ASCIZ / NO INTERRUPT WAS GENERATED/
1544 004256 040 040 111 IFAULT: .ASCIZ / INTERRUPT FAULT/
1545 004300 045 101 040 INTX: .ASCIZ /%A CPU PC: %06%A TSBA: %06/

```


GLOBAL TEXT MESSAGES

```

1546 004335      040      040      042 NOINIT: .ASCIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
1547 004407      040      040      042 NSINIT: .ASCIZ / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
1548 004457      040      040      042 BRINIT: .ASCIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
1549 004527      000
1550 004530      045      116      000 NULCR: .ASCIZ //
1551 004533      045      101      040 EXPGOT: .ASCIZ /%A EXP'D: %06%A, REC'D: %06/
1552 004567      045      116      045 EXPGT2: .ASCIZ /%N%A EXP'D: %06%A, %06%N%A REC'D: %0%A, %06/
1553 004643      045      101      040 DUAD12: .ASCIZ /%A REG(W) WRITTEN TO: %06%A REG(R) READ; EXP'D: %06%A, REC'D: %06/
1554 004745      122      101      115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
1555 005013      040      040      103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
1556 005056      127      122      111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
1557 005113      124      123      123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
1558 005206      124      123      123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
1559 005300      106      101      124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
1560 005372      105      122      122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
1561 005460      045      116      045 NOMEM: .ASCIZ '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****N'
1562 005554      045      116      045 M8186: .ASCIZ '%N%A ***** 11/23A SYSTEM *****N'
1563 005645      045      116      045 M8189: .ASCIZ '%N%A ***** 11/23B SYSTEM *****N'
1564
1565
1566
1567
1568
1569
1570
1571
1572 005736
1573 005736
1574 005762
1575 005766
1576
1577
1578
1579
1580 005770
1581 005772
1582 005774
1583 005776
1584 006002
1585 006022

```

.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.
NXRERR:
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 ; 0 = NO EXTENSION.
BEQ 1$
JSR PC,@EXTA ; APPEND EXTENSION TEXT.
1$: PRINTX #NULCR ; PRINT A BLANK LINE
MOV #NULCR,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #4,SP
RTS PC

```

```

005736 013746 003106
005742 012746 003777
005746 012746 000002
005752 010600
005754 104415
005756 062706 000006
004737 005770
104423
005727
000000
001402
004777 177770
012746 004530
012746 000001
010600
104415
062706 000004
000207

```

PRITSSR - PRINT TSSR CONTENTS

```

1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605 006024
1606 006024
1607 006030 010104
1608 006032
      006032 010446
      006034 012746 006507
      006040 012746 000002
      006044 010600
      006046 104414
      006050 062706 000006
1609 006054 010400
1610 006056 004737 016164
1611 006062 103410
1612 006064
      006064 012746 006727
      006070 012746 000001
      006074 010600
      006076 104415
      006100 062706 000004
1613 006104 010403
1614 006106 042703 001476
1615 006112 001434
1616 006114 012702 002624
1617 006120 012701 003502
1618 006124 005703
1619 006126 001413
1620 006130 000241
1621 006132 006103
1622 006134 103006
1623 006136 011100
1624 006140 112022
1625 006142 001376
1626 006144 112762 000054 177777
1627 006152 005721
1628 006154 000763
1629 006156 105042
1630 006160
      006160 012746 002624
      006164 012746 006700
    
```

```

.SBTTL PRITSSR - PRINT TSSR CONTENTS

;*
;ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
;THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
;BY A MESSAGE PRINTING ROUTINE
;
;INPUTS:
;
;   R1      CONTENTS OF TSSR
;
;SUBORDINATE ROUTINES:
;
;   CHKAMB  CHECK FOR AMBIGUOUS CONTENTS
;
;-

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

PRITSSR - PRINT TSSR CONTENTS

```

006170 012746 000002      MOV      #2,-(SP)
006174 010600      MOV      SP,R0
006176 104415      TRAP    C#PNTX
006200 062706 000006      ADD     #6,SP

1631
1632 006204 010403      20$:   MOV      R4,R3          ;GET THE TSSR CONTENTS
1633 006206 042703 177761      BIC     #1#CTERCLS,R3   ;CLEAR ALL BUT TERMINATION
1634 006212 016303 006770      MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
1635 006216      PRINTX #T#COASC,R3     ;PRINT THE TERMINATION CODE
      006216 010346      MOV     R3,-(SP)
      006220 012746 006570      MOV     #T#COASC,-(SP)
      006224 012746 000002      MOV     #2,-(SP)
      006230 010600      MOV     SP,R0
      006232 104415      TRAP    C#PNTX
      006234 062706 000006      ADD     #6,SP

1636 006240 010403      MOV     R4,R3          ;TSSR CONTENTS AGAIN
1637 006242 042703 177717      BIC     #1#CFATERR,R3   ;CLEAR ALL BUT FATAL TERMINATION
1638 006246 001416      BEQ     25$            ;DON'T PRINT IF ZERO
1639 006250 006203      ASR     R3
1640 006252 006203      ASR     R3
1641 006254 006203      ASR     R3
1642 006256 016303 007330      MOV     TSFCOD(R3),R3   ;ALINE TERMINATION CODE FOR INDEX
1643 006262      PRINTX #T#FCASC,R3     ;GET THE FATAL TERMINATION CODE
      006262 010346      MOV     R3,-(SP)
      006264 012746 006631      MOV     #T#FCASC,-(SP)
      006270 012746 000002      MOV     #2,-(SP)
      006274 010600      MOV     SP,R0
      006276 104415      TRAP    C#PNTX
      006300 062706 000006      ADD     #6,SP

1644 006304 042704 176377      25$:   BIC     #1#CHIADDR,R4   ;CLEAR ALL BUT EXTENDED ADDRESS
1645 006310 001411      BEQ     30$            ;DON'T PRINT IF ZERO
1646 006312      PRINTX #T#EXASC,R4     ;PRINT THE EXTENDED ADDRESS BITS
      006312 010446      MOV     R4,-(SP)
      006314 012746 006527      MOV     #T#EXASC,-(SP)
      006320 012746 000002      MOV     #2,-(SP)
      006324 010600      MOV     SP,R0
      006326 104415      TRAP    C#PNTX
      006330 062706 000006      ADD     #6,SP

1647 006334 013703 002170      30$:   MOV     EPRTSW,R3       ;PRINT MEASGE BUFFER ADDRESS
1648 006340      PRINTX R3              ;PRINT PROPER MESSAGE
      006340 010346      MOV     R3,-(SP)
      006342 012746 000001      MOV     #1,-(SP)
      006346 010600      MOV     SP,R0
      006350 104415      TRAP    C#PNTX
      006352 062706 000004      ADD     #4,SP
1649 006356 000207      RTS     PC              ;RETURN TO CALLER

```

PRITSSR - PRINT TSSR CONTENTS

```

1661 006360 045 116 045 EPRT1: .ASCIZ 'N%#A *****CHECK CABLES BETWEEN M7196 AND TRANSPORT*****'
1662 006450 045 116 045 EPRT2: .ASCIZ 'N%#A *****CHECK TRANSPORT*****'
1668 006507 045 116 045 TSSRFOR: .ASCIZ 'N%#A TSSR = #06'
1669 006527 045 116 045 TEXASC: .ASCIZ 'N%#A Extended Address Bits = #06'
1670 006570 045 116 045 TCOASC: .ASCIZ 'N%#A Termination Class Code = #T'
1671 006631 045 116 045 TFCASC: .ASCIZ 'N%#A Fatal Termination Class Code = #T'
1672 006700 045 116 045 TSSDEF: .ASCIZ 'N%#A TSSR Bits Set: #T'
1673 006727 045 116 045 AMBTSSR: .ASCIZ 'N%#A TSSR Contents Are Ambiguous'
1674 .EVEN
1675 006770 007010 007033 007061 TCOCOD: .WORD 1$,2$,3$,4$,5$,6$,7$,8$
1676 007010 116 157 162 1$: .ASCIZ 'Normal Termination'
1677 007033 124 145 162 2$: .ASCIZ 'Termination Condition'
1678 007061 124 141 160 3$: .ASCIZ 'Tape Status Alert'
1679 007103 106 165 156 4$: .ASCIZ 'Function Reject'
1680 007123 122 145 143 5$: .ASCIZ 'Recoverable Error - Tape Position One Record Down'
1681 007205 122 145 143 6$: .ASCIZ 'Recoverable Error - Tape Was Not Moved'
1682 007254 125 156 162 7$: .ASCIZ 'Unrecoverable Error'
1683 007300 106 141 164 8$: .ASCIZ 'Fatal Controller Error'
1684 .EVEN
1685
1686 007330 007340 007374 007405 TSFCOD: .WORD 1$,2$,3$,4$
1687 007340 111 156 164 1$: .ASCIZ 'Internal Diagnostic Failure'
1688 007374 122 145 163 2$: .ASCIZ 'Reserved'
1689 007405 102 165 163 3$: .ASCIZ 'Bus Interface or Sanity Check Error'
1690 007451 122 145 163 4$: .ASCIZ 'Reserved'
1691 .EVEN

```

PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

```

1693 .SBTTL PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
1694
1695
1696 ;*
1697 ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
1698 ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
1699 ;INPUT:
1700 ;
1701 ; R0 NUMBER OF WORDS IN PACKET
1702 ; R3 HIGH ORDER COMMAND PACKET ADDRESS
1703 ; R4 ADDRESS OF COMMAND PACKET
1704 ;
1705 ; NOTE: R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
1706 ;-
1707
1708 007462 PRIPKT:: SAVREG ;SAVE THE REGISTERS
1709 007462 MOV R0,R5 ;SAVE NO. OF WORDS IN PACKET
1710 007466 010005 TST KTENABLE ;ABOVE 28K UNDER TEST?
1711 007470 005737 003126 BNE 10$ ;BR IF YES
1712 007474 001001 CLR R3 ;SET HIGH ORDER ADDRESS TO 0
1713 007476 005003 10$: MOV R3,R1 ;COPY HIGH ORDER ADDRESS
1714 007500 010301 MOV R4,R0 ;GET LOWER ADDRESS
1715 007502 010400 ROL R0 ;SHIFT BIT 15 INTO C BIT
1716 007504 006100 ROL R1 ;AND INTO HIGH ORDER.
1717 007506 006101 PRINTB #PKTADD,R1,R4 ;PRINT PACKET ADDRESS
1718 007510 MOV R4,-(SP)
1719 007512 010146 MOV R1,-(SP)
1720 007514 012746 007646 MOV #PKTADD,-(SP)
1721 007520 012746 000003 MOV #3,-(SP)
1722 007524 010600 MOV SP,R0
1723 007526 104414 TRAP C#PNTB
1724 007530 062706 000010 ADD #10,SP
1725 007534 010300 15$: MOV R3,R0 ;GET HIGH ORDER ADDRESS
1726 007536 001404 BEQ 20$ ;BR IF NOT ABOVE 28K.
1727 007540 010401 MOV R4,R1 ;GET LOW ORDER ADDRESS
1728 007542 004737 017436 JSR PC,SETMAP ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
1729 007546 010004 MOV R0,R4 ;GET RETURNED PAR6 ADDRESS BIAS
1730 007550 005001 20$: CLR R1 ;SAVE WORD NUMBER
1731 007552 012402 25$: MOV (R4)+,R2 ;GET PACKET CONTENTS
1732 007554 PRINTB #PKTFRM,R1,R2 ;PRINT THE DATA
1733 007554 010246 MOV R2,-(SP)
1734 007556 010146 MOV R1,-(SP)
1735 007560 012746 007610 MOV #PKTFRM,-(SP)
1736 007564 012746 000003 MOV #3,-(SP)
1737 007570 010600 MOV SP,R0
1738 007572 104414 TRAP C#PNTB
1739 007574 062706 000010 ADD #10,SP
1740 007600 005201 INC R1 ;NEXT WORD NUMBER
1741 007602 020105 CMP R1,R5 ;DONE ALL PACKET WORDS?
1742 007604 002762 BLT 25$ ;LOOP TILL ALL DONE
1743 007606 000207 RTS ;RETURN
1744
1745 007610 045 116 045 PKTFRM: .ASCIZ '#N#A Packet Word #D1#A = #06'
1746 007646 045 116 045 PKTADD: .ASCIZ '#N#A Packet Address = #01#05'
1747 .EVEN

```

F4

PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

```

1736 .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR BYTE
1737
1738
1739 ;*
1740 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
1741 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
1742
1743 ;INPUTS:
1744 ; R1 RECEIVED DATA
1745 ; R2 EXPECTED DATA
1746
1747 ;OUTPUT:
1748 ;
1749 ; R0 XOR OF EXPECTED/RECEIVED DATA
1750 ;-
1751 PRIBXOR::
1752 SAVREG ;SAVE THE REGISTERS
1753 MOV R2,R3 ;EXPECTED DATA
1754 XOR R1,R3 ;FORM THE EXCLUSIVE OR
1755 MOV #1C<377>,R0 ;BYTE MASK
1756 BIC R0,R1 ;SAVE LOW BYTE RECV
1757 BIC R0,R2 ;SAVE LOW BYTE EXPD
1758 BIC R0,R3 ;SAVE LOW BYTE XOR
1759 PRINTB #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
007734 MOV R3,-(SP)
007736 MOV R1,-(SP)
007740 MOV R2,-(SP)
007742 MOV #XORBFOR,-(SP)
007746 MOV #4,-(SP)
007752 MOV SP,R0
007754 TRAP C$PNTB
007756 ADD #12,SP
1760 MOV R3,R0 ;R0 HAS XOR ON RETURN
1761 RTS PC ;RETURN TO CALLER
1762
1763 007766 045 116 045 XORBFOR: .ASCIZ '#N#A EXPD: #03#A RECV: #03#A XOR: #03'
1764 .EVEN
1765 .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR
1766
1767 ;*
1768 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
1769 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
1770
1771 ;INPUTS:
1772 ; R1 RECEIVED DATA
1773 ; R2 EXPECTED DATA
1774
1775 ;OUTPUT:
1776 ;
1777 ; R0 XOR OF EXPECTED/RECEIVED DATA
1778 ;-
1779 PRIBXOR::
1780 SAVREG ;SAVE THE REGISTERS
1781 MOV R2,R3 ;EXPECTED DATA
1782 XOR R1,R3 ;FORM THE EXCLUSIVE OR
1783 PRINTB #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
1784 010034
010034
010040 010203
010042
010052

```

G4

PRIXOR - PRINT EXPD, RECV AND XOR

010052	010346				MOV	R3,-(SP)		
010054	010146				MOV	R1,-(SP)		
010056	010246				MOV	R2,-(SP)		
010060	012746	010104			MOV	#XORFOR,-(SP)		
010064	012746	000004			MOV	#4,-(SP)		
010070	010600				MOV	SP,RO		
010072	104414				TRAP	C:PNTB		
010074	062706	000012			ADD	#12,SP		
1785	010100	010300			MOV	R3,RO		;RO HAS XOR ON RETURN
1786	010102	000207			RTS	PC		;RETURN TO CALLER
1787								
1788	010104	045	116	045	XORFOR:	.ASCIZ	'#N#A EXPD: #06#A RECV: #06#A XOR: #06'	
1789						.EVEN		

PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT

```

1791 .SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
1792
1793
1794 ;*
1795 ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
1796 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
1797
1798 ;INPUTS:
1799
1800 ; R0 OCTAL VALUE TO CONVERT
1801 ; R1 TABLE OF POINTERS TO ASCII EQUIVALENT
1802
1803 ;-
1804
1805 010152 PRIEQU: SAVREG ;SAVE THE REGISTERS
1806 010152 RTS PC ;RETURN TO CALLER
1807 010156 000207
1808
1809 .SBTTL PRIRAM - PRINT RAM ADDRESS
1810
1811 ;*
1812 ;PRINT CONTROLLER RAM ADDRESS.
1813 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
1814
1815 ;INPUTS:
1816
1817 ; R4 RAM ADDRESS
1818
1819 ;-
1820 010160 PRIRAM: SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1821 010160 PRINTB #RAMFOR,R4 ;PRINT RAM ADDRESS IN ERROR
1822 010164 010446 MOV R4,-(SP)
010166 012746 010210 MOV #RAMFOR,-(SP)
010172 012746 000002 MOV #2,-(SP)
010176 010600 MOV SP,R0
010200 104414 TRAP C#PNTB
010202 062706 000006 ADD #6,SP
1823 010206 000207 RTS PC ;RETURN
1824
1825 010210 045 116 045 RAMFOR: .ASCIZ '#N#A CONTROLLER RAM ADDRESS = #06'
1826 .EVEN

```


PRIADD - PRINT MEMORY ERROR ADDRESS

```

1828 .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
1829
1830 ;*
1831 ;PRINT MEMORY ADDRESS
1832 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
1833 ;
1834 ; IMPLICIT INPUTS
1835 ;
1836 ; ERRHI - HIGH ORDER ADDRESS
1837 ; ERRLO - LOW ORDER ADDRESS
1838 ;
1839 ;-
1840 PRIADD: SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1841 MOV ERRHI,R0 ;GET HIGH ADDRESS
1842 MOV ERRLO,R1 ;GET LOW ADDRESS
1843 MOV R1,R2 ;COPY LOW ADDRESS
1844 ROL R1 ;SHIFT BIT 15 TO C BIT
1845 ROL R0 ;SHIFT INTO HIGH ORDER
1846 PRINTB #PRIA0,R0,R2 ;PRINT MEMORY ADDRESS IN ERROR
1847 MOV R2,-(SP)
1848 MOV R0,-(SP)
1849 MOV #PRIA0,-(SP)
1850 MOV #3,-(SP)
1851 MOV SP,R0
1852 TRAP C#PNTB
1853 ADD #10,SP
1854 RTS PC ;RETURN

```

```

1848 010252
1849 010252
1850 010256 013700 002230
1851 010262 013701 002232
1852 010266 010102
1853 010270 006101
1854 010272 006100
1855 010274
1856 010274 010246
1857 010276 010046
1858 010300 012746 010322
1859 010304 012746 000003
1860 010310 010600
1861 010312 104414
1862 010314 062706 000010
1863 010320 000207
1864
1865 010322 045 116 045 PRIA0: .ASCIZ '#N#A MEMORY ERROR ADDRESS = #01#05'
1866 .EVEN

```

```

1852 .SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
1853
1854 ;*
1855 ;PRINT MEMORY ADDRESS
1856 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
1857 ;
1858 ; IMPLICIT INPUTS
1859 ;
1860 ; ERRHI - HIGH ORDER ADDRESS
1861 ; ERRLO - LOW ORDER ADDRESS
1862 ;
1863 ;-
1864 PRITADD: SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1865 MOV ERRHI,R2 ;GET HIGH ADDRESS
1866 MOV ERRLO,R1 ;GET LOW ADDRESS
1867 MOV R1,R2 ;COPY LOW ADDRESS
1868 ROL R1 ;SHIFT BIT 15 TO C BIT
1869 ROL R0 ;SHIFT INTO HIGH ORDER
1870 PRINTB #PRIT0,R1 ;PRINT MEMORY ADDRESS LOW IN ERROR
1871 MOV R1,-(SP)
1872 MOV #PRIT0,-(SP)
1873 MOV #2,-(SP)
1874 MOV SP,R0
1875 TRAP C#PNTB

```

J4

PRITADD - PRINT MEMORY TEST ADDRESS

```

1873 010420 062706 000006      ADD    #6,SP
      010424      PRINTB #PRIT1,R2      ;PRINT MEMORY ADDRESS HIGH IN ERROR
      010424 010246      MOV    R2,-(SP)
      010426 012746 010513      MOV    #PRIT1,-(SP)
      010432 012746 000002      MOV    #2,-(SP)
      010436 010600      MOV    SP,R0
      010440 104414      TRAP  C$PNTB
1874 010442 062706 000006      ADD    #6,SP
      010446 000207      RTS    PC      ;RETURN
1875
1876 010450      045      116      045 PRIT0: .ASCIZ 'N#A MEMORY TEST ADDRESS LOW = #06'
1877 010513      045      116      045 PRIT1: .ASCIZ 'N#A MEMORY TEST ADDRESS HIGH = #06'
1878 .EVEN

```

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

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

1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914

```

;+
;ROUTINE TO ISSUE A SPACE RECORDS
;COMMAND (FORWARD OR REVERSE)
;INPUT:
;
;   R3      NUMBER OF RECORDS TO BE SPACED OVER
;           BIT15 CONTROLS DIRECTION
;           BIT15 = 0 IS FORWARD
;           BIT15 = 1 IS REVERSE
;   R5      FIRST DEVICE UNIBUS ADDRESS
;
;   REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
;OUTPUT:
;
;   CARRY   SET - SPACE RECORDS COMMAND OK
;           CLR - SPACE RECORDS FAILED
;
;   R0      THE CONTENTS OF R4 IS MOVED TO R0
;
;IMPLICIT OUTPUT:
;
;   TAPE HAS BEEN MOVED
;SIDE EFFECTS:
;
;-

```

```

1915 010560
1916 010560
1917 010564 012737 000764 010750
1918 010572 012737 140010 010740
1919 010600 005703
1920 010602 100403
1921 010604 010337 010742
1922 010610 000407
1923 010612 042703 100000
1924 010616 010337 010742
1925 010622 052737 000400 010740
1926 010630 012704 010740
1927 010634 010465 000000
1928 010640 004737 016370
1929 010644 103420
1930 010646
    010646 012727 000250
    010652 000000
    010654 013727 002116
    010660 000000
    010662 005367 177772
    010666 001375

```

```

SPACE::
    SAVREG                                ;SAVE THE GENERAL REGISTERS
    MOV #500.,SDELAY                       ;SET UP DELAY
    MOV #140010,80$                         ;SET UP COMMAND, SPACE FORWARD
    TST R3                                  ;CHECK FOR DIRECTION
    BMI 5$                                  ;BR, IF REVERSE INDICATED
    MOV R3,90$                              ;LOAD UP NUMBER OF RECORDS TO SPACE
    BR 10$                                  ;GO DO COMMAND
    BIC #BIT15,R3                          ;CLEAR DIRECTION BIT
    MOV R3,90$                              ;LOAD UP NUMBER OF RECORDS TO SPACE
    BIS #BIT8,80$                          ;SET REVERSE BIT IN COMMAND PACKET
    MOV #80$,R4                             ;SET UP R4 WITH PACKET ADDRESS
    MOV R4,TSDB(R5)                        ;SEND OUT COMMAND
    JSR PC,WAITF                            ;WAIT FOR SSR
    BCS 20$                                  ;BR, IF SSR IS SET AND OK
    DELAY 250                               ;DELAY ABOUT .25 SECONDS
    MOV #250,(PC)
    .WORD 0
    MOV L$DLY,(PC)
    .WORD 0
    DEC -6(PC)
    BNE .-4

```

L4

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```

010670 005367 177756          DEC    -22(PC)
010674 001367                BNE    .-20
1931 010676 005337 010750     DEC    SDELAY          ;BUMP DELAY COUNTER DOWN
1932 010702 001356                BNE    15$            ;BR, IF MORE DELAY
1933 010704 000411                BR     60$            ;BR IF TROUBLE CARRY = CLEAR
1934 010706 016501 000002     20$:  MOV    TSSR(R5),R1 ;READ TSSR
1935 010712 012702 000200     MOV    #SSR,R2        ;SET UP EXPECTED
1936 010716 020201     25$:  CMP    R2,R1        ;ARE THEY OK
1937 010720 001401                BEQ    40$            ;BR, IF EQUAL = OK
1938 010722 000402                BR     60$            ;TROUBLE EXIT
1939 010724 000261     40$:  SEC                    ;SET CARRY NO TROUBLE
1940 010726 000401                BR     70$            ;EXIT
1941 010730 000241     60$:  CLC                    ;CARRY CLEAR = ERROR
1942 010732 70$:                MOV    R4,R0          ;PASS PACKET ADDRESS
1943 010732 010400                RTS    PC              ;RETURN
1944 010734 000207
1945
1946
1947
1948
1949
1951          010740
1953
1954
1955 010740 000000     ;COMMAND WORD
1956     80$:  .WORD
1957          000000     ;NUMBER OF RECORDS TO BE SPACED OVER WORD
1958 010742 000000     90$:  .WORD
1959 010744 000000                .WORD
1960 010746 000000                .WORD
1961          000000     SDELAY: .WORD    0          ;DELAY COUNTER
1962                .EVEN
                .SBTTL  WRTCHR - WRITE CHARACTERISTICS COMMAND

```

WRTCHR - WRITE CHARACTERISTICS COMMAND

```

1964
1965 ;*
1966 ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1967 ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1968
1969 ;INPUT:
1970 ; R4 ADDRESS OF PACKET FROM TEST
1971 ; R5 FIRST DEVICE UNIBUS ADDRESS
1972 ; REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1973
1974 ;OUTPUT:
1975 ; R0 TSSR CONTENTS
1976 ; CARRY SET - WRITE CHARACTERISTICS COMMAND OK
1977 ; CLR - WRITE CHARACTERISTICS FAILED
1978
1979 ;IMPLICIT OUTPUT:
1980 ;
1981 ; MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
1982 ; SOFTWARE SWITCHES SET AS FOLLOWS:
1983 ; EXTFEA = EXTENDED FEATURES PRESENT
1984 ; BENBSW = BUFFER ENABLE SWITCH ON OR OFF
1985
1986 ;SIDE EFFECTS:
1987 ;-
1988 WRTCHR::
1989 ; SAVREG ;SAVE THE GENERAL REGISTERS
1990 CLR BENBSW ;CLEAR BUFFER ENABLE SWITCH
1991 CLR EXTFEA ;CLEAR EXTENDED FEATURES SW SWITCH
1992 10$: MOV R4,TSDB(R5) ;SEND OUT COMMAND
1993 JSR PC,CHKTSSR ;WAIT FOR SSR
1994 BCS 20$ ;BR, IF SSR IS SET AND OK
1995 BR 60$ ;BR IF TROUBLE CARRY = CLEAR
1996 20$: MOV TSSR(R5),R1 ;READ TSSR
1997 MOV #SSR,R2 ;SET UP EXPECTED
1998 BIT #OFL,R1 ;WAS OFF LINE SET IN TSSR
1999 BEQ 25$ ;BR, IF NO OFL SET
2000 25$: BIS #OFL,R2 ;MAKE THEM LOOK ALIKE
2001 CMP R2,R1 ;ARE THEY OK
2002 BEQ 40$ ;BR, IF EQUAL = OK
2003 BR 60$ ;TROUBLE EXIT
2004 40$: ADD #8,R4 ;POINT TO WRT CHARA DATA PACKET
2005 MOV (R4),R3 ;GET ADDRESS OF MESSAGE BUFFER
2006 BIT #X2.EXTF,XST2(R3) ;EXTENDED FEATURES BIT SET?
2007 BEQ 45$ ;BR IF NO
2008 INC EXTFEA ;SET EXTENDED FEATURES SW SWITCH
2009 45$: BIT #X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
2010 BEQ 50$ ;BR, IF SWITCH NOT SET
2011 INC BENBSW ;SET SOFTWARE SWITCH FOR ENABLED
2012 50$: MOV XST2(R3),REV ;GET REVISION LEVEL OF MICROCODE
2013 BIC #17700,REV ;REVISION LEVEL IN REV
2014 CMP #1,REV ;IS IT A NEW REV
2015 BEQ 55$ ;NO BR.
2016 MOV #1,EXTFEA ;FOR NEW REV ALWAYS EXTFEA SET
2017 55$: SEC ;SET CARRY NO TROUBLE
2018 BR 70$ ;EXIT
2019 60$: CLC ;CARRY CLEAR = ERROR
2020

```

N4

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 08:48 Page 39-1

SEQ 0052

WRTCHR - WRITE CHARACTERISTICS COMMAND

2021 011130 016500 000002
2022 011134 000207

704:

MOV
RTS

TSSR(R5),R0
PC

;RETURN TSSR CONTENTS
;RETURN

REWIND - POSITION TAPE (REWIND) COMMAND

2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2066
2068
2069
2070

011136
011136
011142 012704 011230
011146 010465 000000
011152 012703 000550
011156 004737 016370
103417
011164 012727 000372
011170 000000
011172 013727 002116
011176 000000
011200 005367 177772
011204 001375
011206 005367 177756
011212 001367
011214 005303
011216 001357
011220 000241
011222 010400
011224 000207
011230
011230 102010
011232 000000

```

.SBTTL REWIND - POSITION TAPE (REWIND) COMMAND
; *
; THIS ROUTINE WILL REWIND THE SELECTED TAPE.
; CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
; TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
; SSR TO SET IN THE TSSR
; CALLING SEQUENCE:
; DO A SOFT INIT
; DO A WRITE CHARACTERISTICS
; JSR PC,REWIND
; INPUT:
; R5 FIRST DEVICE UNIBUS ADDRESS
; OUTPUT
; R0 THE CONTENTS OF R4 IS PASSED TO R0
; -
REWIND:
; SAVREG ;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 ;POSTION COMMAND (REWIND)
.WORD 0 ;NOT USED

```

CKRAM - COMPARE RAM TO I/O PACKET

2072
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

```
.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
;-
```

2100 011234
2101 011234
2102 011240 012701 002234
2103 011244 012702 000201
2104 011250 005003
2105 011252 004737 016456
2106 011256 112765 000000 000000
2107 011264 004737 016456 10\$:
2108 011270 010265 000000
2109 011274 004737 016456
2110 011300 116511 000000
2111 011304 122124
2112 011306 001401
2113 011310 005203
2114 011312 005202 20\$:
2115 011314 020227 000210
2116 011320 003761
2117 011322 005703
2118 011324 001402
2119 011326 000241
2120 011330 000401
2121 011332 000261 30\$:
2122 011334 012737 000010 002274 50\$:
2123 011342 000207

```
CKRAM::
SAVREG
MOV #RAMDATA,R1 ;SAVE THE GENERAL REGISTERS
MOV #RMPKTBEGR,R2 ;ADDRESS TO SAVE THE RAM DATA
CLR R3 ;BYTE ADDRESS OF FIRST RAM DATA
;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
```


CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

```

2125          .SBTTL CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA
2126          ;*
2127          ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
2128          ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
2129
2130          ;INPUT:
2131          ;
2132          ;       R4      ADDRESS OF THE CHARACTERISTICS DATA
2133          ;       R5      FIRST DEVICE UNIBUS ADDRESS
2134
2135          ;OUTPUT:
2136          ;
2137          ;       CARRY   SET - RAM MATCHES PACKET
2138          ;              CLR - RAM DOES NOT MATCH PACKET
2139
2140          ;IMPLICIT OUTPUT:
2141          ;
2142          ;       THE TABLE RAMDATA IS FILLED WITH THE
2143          ;       DATA HELD IN RAM.
2144          ;       RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
2145
2146          ;SIDE EFFECTS:
2147          ;
2148          ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
2149
2150          ;-
2151          CKRAM2::
2152          SAVREG          ;SAVE THE GENERAL REGISTERS
2153          MOV             #RAMDATA,R1      ;ADDRESS TO SAVE THE RAM DATA
2154          MOV             #RMCHBEG,R2     ;BYTE ADDRESS OF FIRST RAM DATA
2155          CLR            R3              ;CLEAR THE ERROR FLAG
2156          JSR            PC,CHKTSSR      ;WAIT FOR SSR
2157          MOV            #0,TSDB(R5)     ;SET MAINTENANCE MODE
2158          JSR            PC,CHKTSSR      ;WAIT FOR SSR TO SET
2159          MOV            R2,TSDB(R5)     ;SELECT NEXT RAM ADDRESS
2160          JSR            PC,CHKTSSR      ;WAIT FOR SSR TO SET
2161          MOV            TSBA(R5),(R1)   ;READ THE RAM DATA
2162          CMPB           (R1)+,(R4)+    ;COMPARE TO EXPECTED
2163          BEQ            20$            ;BRANCH IF OK
2164          INC            R3              ;SET ERROR FLAG
2165          INC            R2              ;ADDRESS OF NEXT RAM LOCATION
2166          MOV            #8.,RAMSIZ     ;ASSUME EXTFEA NOT SET
2167          TST            EXTFEA         ;IS THE SOFTWARE EXTENDED FEATURES SET
2168          BEQ            25$            ;BR, IF NOT SET
2169          MOV            #10.,RAMSIZ    ;SET RAMSIZ FOR EXTEND FEATURES
2170          CMP            R2,#RMCHEND    ;AT END OF EXTENDED BUFFER
2171          BLE            10$            ;BR, IF NOT AT END YET
2172          BR             27$            ;AT END BRANCH
2173          CMP            R2,#RMCHEND-2  ;REACHED END YET ?
2174          BLE            10$            ;BRANCH TILL ALL READ
2175          TST            R3              ;WAS AN ERROR FOUND ?
2176          BEQ            30$            ;BRANCH IF NOT
2177          CLC              ;CLEAR CARRY TO SHOW ERROR
2178          BR             50$            ;AND EXIT
2179          SEC              ;SHOW GOOD COMPARE
2180          RTS             PC            ;RETURN

```

CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

```

2182          .SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS
2183          ;*
2184          ;
2185          ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
2186          ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
2187          ;ERROR PRINT ROUTINES.
2188          ;
2189          ;INPUT:
2190          ;
2191          ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
2192          ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
2193          ;      R2      EXPD MESSAGE BUFFER ADDRESS
2194          ;OUTPUT:
2195          ;
2196          ;      CARRY   SET - MESSAGE BUFFERS MATCH
2197          ;             CLR -MESSAGE BUFFERS DON'T MATCH
2198          ;
2199          ;IMPLICIT OUTPUT:
2200          ;
2201          ;      EXPMSG   BUFFER IS SET TO EXPD DATA
2202          ;      RECVMSG  BUFFER IS SET TO RECV DATA
2203          ;      RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
2204          ;      RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
2205          ;
2206          ;-
2207          CKMSG::
2208          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
2209          MOV      R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
2210          MOV      R1,RCVLOAD  ;SAVE RECV LOW ADDRESS
2211          TST     KTENABLE    ;TESTING ABOVE 28K?
2212          BEQ     10$         ;BR IF NO
2213          JSR     PC,SETMAP    ;RETURN ADDRESS BIASED TO PAR6 IN R0
2214          MOV     R0,R1       ;GET RETURNED ADDRESS BIASED TO PAR6
2215          10$:  CLR     R4      ;WORD IN BUFFER
2216          CLR     R3         ;CLEAR ERROR SEEN FLAG
2217          MOV     R2,R5       ;GET EXPD BUFFER ADDRESS
2218          15$:  MOV     (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
2219          MOV     (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
2220          CMP     (R2)+,(R1)+  ;EXPD EQUAL RECV?
2221          BEQ     25$         ;BR IF YES
2222          INC     R3         ;SET ERROR SEEN FLAG
2223          25$:  ADD     #2,R4   ;POINT TO NEXT WORD ADDRESS
2224          CMP     R4,#14     ;DONE FIRST 7 WORDS?
2225          BLE     15$         ;BR IF NO
2226          BIT     #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
2227          BEQ     50$         ;BR IF NO
2228          CMP     R4,#16     ;DONE EXTENDED FEATURES WORD?
2229          BLE     15$         ;BR IF NO
2230          50$:  TST     R3      ;ANY ERRORS SEEN?
2231          BEQ     55$         ;BR IF NO
2232          CLC          ;SET FAILURE
2233          BR      60$         ;
2234          55$:  SEC          ;SET SUCCESS
2235          60$:  RTS     PC    ;RETURN

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2237 .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
2238
2239 ;*
2240 ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
2241 ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
2242 ;ERROR PRINT ROUTINES.
2243
2244 ;INPUT:
2245 ; R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
2246 ; R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
2247 ; R2 EXPD MESSAGE BUFFER ADDRESS
2248 ; R3 NUMBER OF BYTES TO COMPARE
2249
2250 ;OUTPUT:
2251 ; CARRY SET - MESSAGE BUFFERS MATCH
2252 ; CLR - MESSAGE BUFFERS DON'T MATCH
2253
2254 ;IMPLICIT OUTPUT:
2255 ; EXPMSG BUFFER IS SET TO EXPD DATA
2256 ; RECMMSG BUFFER IS SET TO RECV DATA
2257 ; RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
2258 ; RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
2259
2260 CKMSG2::
2261 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2262 CMP R3,#RECMMSG-EXPMSG;@AD IS COUNT ABOVE MAX ALLOWED?
2263 BLE 5$ ;@AD BR IF NO
2264 MOV #RECMMSG-EXPMSG,R3;@AD
2265 PRINTF #DEBUGMSG ;@AD
2266 MOV #DEBUGMSG,-(SP)
2267 MOV #1,-(SP)
2268 MOV SP,R0
2269 TRAP C$PNTF
2270 ADD #4,SP
2271 5$: MOV R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
2272 MOV R1,RCVLOAD ;SAVE RECV LOW ADDRESS
2273 TST KTENABLE ;TESTING ABOVE 28K?
2274 BEQ 10$ ;BR IF NO
2275 JSR PC,SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN R0
2276 MOV R0,R1 ;GET RETURNED ADDRESS BIASED TO PAR6
2277 10$: CLR R4 ;WORD IN BUFFER
2278 CLR R5 ;CLEAR ERROR SEEN FLAG
2279 15$: MOVB (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
2280 MOVB (R1),RECMMSG(R4) ;SAVE RECV FOR ERROR REPORT
2281 CMPB (R2)+,(R1)+ ;EXPD EQUAL RECV?
2282 BEQ 25$ ;BR IF YES
2283 INC R5 ;SET ERROR SEEN FLAG
2284 25$: ADD #1,R4 ;POINT TO NEXT BYTE
2285 CMP R4,R3 ;DONE ALL BYTES?
2286 BGE 50$ ;BR IF YES
2287 BR 15$ ;DO NEXT BYTE
2288 50$: TST R5 ;ANY ERRORS SEEN?
2289 BEQ 55$ ;BR IF NO
2290 CLC ;SET FAILURE
2291 BR 60$ ;
2292 55$: SEC ;SET SUCCESS
2293 60$: RTS PC ;RETURN

```

G5

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2289 011752      120      122      117  DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';@@D
2290 012042      045      116      045  FERCM:  .ASCII /%N%A ***/
2291 012053      040      040      124  ERCM:   .ASCIZ / TSSR ERROR CODE REC'D = /
2292 012106      056      056      056  SIMSG:  .ASCIZ /... AFTER DOING SOFT INIT/
2293 012141      124      105      123  TINERR: .ASCIZ /TEST: .../
2294                                     .EVEN
2295                                     ;*
2296                                     ;
2297                                     ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
2298                                     ;
2299                                     ;INPUT:
2300                                     ;
2301                                     ;      R1      CONTENTS OF TSSR AT ERROR
2302                                     ;
2303                                     ;SIDE EFFECTS:
2304                                     ;
2305                                     ;      EXECUTES DROP UNIT TO CEASE TESTING
2306                                     ;
2307                                     ;-
2308
2309 012154      BGNMSG  SFIMSG
2310 012154      SFIMSG:: JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
2311 012160      004737  006024 JSR      PC,CKDROP      ;DROP UNIT, IF ALLOWED
2312 012164      ENDMSG
2313 012164      104423  L10003: TRAP      C$MSG
2314
2315                                     ;*
2316                                     ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2317                                     ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
2318                                     ;
2319                                     ;INPUTS:
2320                                     ;
2321                                     ;      R1      TSSR CONTENTS
2322                                     ;      R4      ADDRESS OF COMMAND PACKET
2323                                     ;
2324                                     ;-
2325 012166      BGNMSG  PKTSSR
2326 012166      PKTSSR:: JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
2327 012172      004737  006024 MOV      #4,R0          ;NO. OF WORDS IN PACKET
2328 012176      004737  007462 JSR      PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
2329 012202      ENDMSG
2330 012202      L10004: TRAP      C$MSG
2331 012202      104423

```

H5

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2331
2332 ;*
2333 ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2334 ;TSSR AND A GET STATUS COMMAND PACKET.
2335 ;
2336 ;INPUTS:
2337 ;
2338 ; R1 TSSR CONTENTS
2339 ; R4 ADDRESS OF COMMAND PACKET
2340 ;-
2340 012204 BGNMSG PKTGETS
2341 012204 PKTGETS::
2341 012204 004737 006024 JSR PC,PRITSSR ;PRINT THE CONTENTS OF TSSR REGISTER
2342 012210 012700 000002 MOV #2,R0 ;NO. OF WORDS IN GET STATUS PACKET
2343 012214 004737 007462 JSR PC,PRIPKT ;PRINT THE CONTENTS OF COMMAND PACKET
2344 012220 ENDMSG
2345 012220 L10005:
2346 012220 104423 TRAP C$MSG
2347 ;*
2348 ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
2349 ;
2350 ;INPUTS:
2351 ; R1 TSSR CONTENTS
2352 ; R4 ADDRESS OF COMMAND PACKET
2352 012222 BGNMSG SFFMSG
2353 012222 004737 006024 SFFMSG::
2354 012226 JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR REGISTER
2355 012226 ENDMSG
2356 012226 L10006:
2357 012226 104423 TRAP C$MSG
2358 ;.SBTTL PKTMES - PRINT TSSR AND MESSAGE BUFFER
2359 ;*
2360 ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
2361 ;BUFFER FOR ERROR REPORTS
2362 ;
2363 ;INPUTS:
2364 ; R1 CONTENTS OF TSSR
2365 ; R2 LOW ORDER MESSAGE BUFFER
2366 ; R3 HIGH ORDER MESSAGE BUFFER ADDRESS
2367 ; NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
2367 012230 BGNMSG PKTMES
2368 012230 PKTMES::
2368 012230 004737 006024 JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR
2369 012234 010200 MOV R2,R0 ;LOW ORDER ADDRESS
2370 012236 010301 MOV R3,R1 ;HIGH ORDER ADDRESS
2371 012240 004737 014362 JSR PC,PRMESS ;PRINT THE MESSAGE BUFFER
2372 012244 ENDMSG
2372 012244 L10007:
2372 012244 104423 TRAP C$MSG

```

ADDSSR - PRINT TEST ADDRESS AND TSSR

```

2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386 012246
      012246
2387 012246 004737 010366
2388 012252 016501 000002
2389 012256 004737 006024
2390 012262
      012262
      012262 104423
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404 012264
      012264
2405 012264 012700 000007
2406 012270 005737 002216
2407 012274 001402
2408 012276 012700 000010
2409 012302 004737 014672
2410 012306
      012306
      012306 104423

```

```

      .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
;      RECMMSG - RECEIVED MESSAGE BUFFER
;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
;-
      BGNMSG MSGEXP
MSGEXP::
      MOV      #7,R0          ;ASSUME NO EXT FEATURES
      TST      EXTFEA        ;EXT FEATURES SET?
      BEQ      5$            ;BR IF NO
      MOV      #8.,R0        ;EXT FEATURE BUFFER IS 8 WORDS
      JSR      PC,PRMSGEXP   ;PRINT EXPD/RCV MESSAGE BUFFERS
      ENDMSG
5$:
L10011:
      TRAP     C$MSG

```

FIFEXP - PRINT FIFO EXP/RECV DATA

```

2412 .SBTTL FIFEXP - PRINT FIFO EXP/RECV DATA
2413
2414 ;*
2415 ;PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
2416 ;
2417 ; R1 - BYTE COUNT
2418 ;
2419 ;IMPLICIT INPUTS:
2420 ;
2421 ; EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
2422 ; RECVMSG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
2423 ;
2424 ;-
012310 BGNMSG FIFEXP
2425 FIFEXP::
012310 PRINTX #FIF1MSG,R1 ;PRINT BYTES TRANSFERRED
012310 MOV R1,-(SP)
012312 010146 012746 012362 MOV #FIF1MSG,-(SP)
012316 012746 000002 MOV #2,-(SP)
012322 010600 MOV SP,R0
012324 104415 TRAP C#PNTX
012326 062706 000006 ADD #6,SP
2426 012332 PRINTX #FIF2MSG ;PRINT HEADER MSG
012332 012746 012431 MOV #FIF2MSG,-(SP)
012336 012746 000001 MOV #1,-(SP)
012342 010600 MOV SP,R0
012344 104415 TRAP C#PNTX
012346 062706 000004 ADD #4,SP
2427 012352 010100 MOV R1,R0 ;GET BYTE COUNT
2428 012354 004737 015242 JSR PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
2429 012360 ENDMSG
012360 L10012:
012360 104423 TRAP C#MSG
2430 012362 045 116 045 FIF1MSG: .ASCIZ '%N#A NUMBER OF BYTES TRANSFERRED = #D2'
2431 012431 045 116 045 FIF2MSG: .ASCIZ '%N#A FIFO DATA BYTES IN ERROR:'
2432 .EVEN

```

MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS

```

2434 .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
2435 ;*
2436 ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
2437 ;
2438 ;
2439 ;IMPLICIT INPUTS:
2440 ;
2441 ;
2442 ; EXPMSG - EXPECTED MESSAGE BUFFER
2443 ; RECMSG - RECEIVED MESSAGE BUFFER
2444 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2445 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2446 ;-
2447 012470 BGNMSG MSGSTAT
012470 MSGSTAT:
2448 012470 012701 012532 MOV #STATCOD,R1 ;ASCII ADDRESS TABLE
2449 012474 012100 10$: MOV (R1)+,RO ;DONE ALL MSG LINES?
2450 012476 001410 BEQ 20$ ;BR IF YES
2451 012500 PRINTX RO ;PRINT STATUS BIT NAMES
012500 010046 MOV RO,-(SP)
012502 012746 000001 MOV #1,-(SP)
012506 010600 MOV SP,RO
012510 104415 TRAP C$PNTX
012512 062706 000004 ADD #4,SP
2452 012516 000766 BR 10$ ;DO ANOTHER MSG LINE
2453 012520 012700 000012 20$: MOV #10,RO ;NUMBER OF WORDS IN A READ STATUS BUFFER
2454 012524 004737 014672 JSR PC,PRMSGEXP ;PRINT EXPD/RECV MESSAGE BUFFERS
2455 012530 ENDMMSG
012530 L10013:
012530 104423 TRAP C$MSG
2456 2457 012532 012550 012612 012703 STATCOD: .WORD 1$,2$,3$,4$,5$,6$,0
2458 012550 045 116 045 1$: .ASCIZ '%N%A Tape Bus Signals in Word #8:'
2459 012612 045 116 045 2$: .ASCIZ '%N%A PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
2460 012703 045 116 045 3$: .ASCIZ '%N%A IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
2461 012774 045 116 045 4$: .ASCIZ '%N%A IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
2462 013065 045 116 045 5$: .ASCIZ '%N%A Tape Bus Signals in Word #9:'
2463 013127 045 116 045 6$: .ASCIZ '%N%A DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
2464 .EVEN
2465

```


L5

MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

```

2467          .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
2468          ;*
2469          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
2470          ;
2471          ;IMPLICIT INPUTS:
2472          ;
2473          ;   EXPMSG - EXPECTED MESSAGE BUFFER
2474          ;   RECMSG - RECEIVED MESSAGE BUFFER
2475          ;   RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2476          ;   RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2477          ;
2478          ;-
2479 013204      BGNMSG MSGLOOP
013204      MSGLOOP:
2480 013204 012701 013246      MOV     #LOOPCOD,R1      ;ASCII ADDRESS TABLE
2481 013210 012100             MOV     (R1)+,R0        ;DONE ALL MSG LINES?
2482 013212 001410             BEQ     20$              ;BR IF YES
2483 013214             PRINTX  R0              ;PRINT STATUS BIT NAMES
013214 010046             MOV     R0,-(SP)
013216 012746 000001         MOV     #1,-(SP)
013222 010600             MOV     SP,R0
013224 104415             TRAP   C#PNTX
013226 062706 000004         ADD     #4,SP
2484 013232 000766             BR     10$              ;DO ANOTHER MSG LINE
2485 013234 012700 000012     20$:  MOV     #10,R0          ;NUMBER OF WORDS IN A READ STATUS BUFFER
2486 013240 004737 014672     JSR     PC,PRMSGEXP    ;PRINT EXPD/RECV MESSAGE BUFFERS
2487 013244             ENDMSG
013244             L10014:
013244 104423             TRAP   C#MSG
2488
2489 013246 013266 013341 013440 LOOPCOD:      .WORD  1$,2$,3$,4$,5$,6$,7$,0
2490 013266           045 116 045 1$: .ASCIZ  '%N%A Tape Bus Loopback Signals in Word #8:'
2491 013341           045 116 045 2$: .ASCIZ  '%N%A PARERR<15> IRESV2<14> IRESV1<13>'
2492 013440           045 116 045 3$: .ASCIZ  '%N%A IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
2493 013537           045 116 045 4$: .ASCIZ  '%N%A IWFM =>IFMK<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
2494 013636           045 116 045 5$: .ASCIZ  '%N%A ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDP <04>'
2495 013735           045 116 045 6$: .ASCIZ  '%N%A IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
2496 014034           045 116 045 7$: .ASCIZ  '%N%A IGO =>IFPT<00>'
2497             .EVEN

```

M5

MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```

2499          .SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
2500          ;*
2501          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
2502          ;
2503          ;
2504          ;IMPLICIT INPUTS:
2505          ;
2506          ;     EXPMSG - EXPECTED MESSAGE BUFFER
2507          ;     RECMSG - RECEIVED MESSAGE BUFFER
2508          ;     RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2509          ;     RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2510          ;
2511          ;-
2512 014062      BGNMSG  MSGSUB
014062
2513 014062    012700  000012
2514 014066    004737  014672
2515 014072
014072
014072    104423

2516
2517          .SBTTL MEMADD - PRINT MEMORY ADDRESS DATA ERROR
2518          ;*
2519          ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
2520          ;
2521          ;IMPLICIT INPUTS:
2522          ;
2523          ;     ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
2524          ;     ERRLO - MEMORY ERROR LOW ORDER ADDRESS
2525          ;     EXP   - EXPECTED DATA
2526          ;     RECV  - RECEIVED DATA
2527          ;
2528          ;-
2529 014074      BGNMSG  MEMADD
014074
2530 014074    004737  010252
2531 014100    013701  002224
2532 014104    013702  002226
2533 014110    004737  010034
2534 014114
014114
014114    104423

MSGSUB::
      MOV     #10,R0          ;SIZE OF WRITE SUBSYSTEM BUFFER
      JSR    PC,PRMSGEXP     ;PRINT EXPD/RCV MESSAGE BUFFERS
      ENDMSG

L10015:
      TRAP   C$MSG

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   C$MSG

```

PRAMPKT - PRINT RAM AND PACKET DATA

```

2536 .SBTTL PRAMPKT - PRINT RAM AND PACKET DATA
2537 ;*
2538 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
2539 ;WHEN THE RAM DATA DOES NOT MATCH.
2540 ;
2541 ;INPUTS:
2542 ;
2543 ; R4 POINTER TO COMMAND PACKET
2544 ;IMPLICIT INPUTS:
2545 ; RAMDATA DATA AS READ FROM THE RAM
2546 ; RAMSIZ NUMBER OF BYTES IN PACKET
2547 ; IF RAMSIZ=0 THEN DEFAULT TO 8.
2548 ;
2549 ;IMPLICIT OUTPUTS:
2550 ; RAMSIZ SET TO 0
2551 ;-
2552 PRAMPKT:
2553 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2554 MOV #RAMDATA,R1 ;DATA FROM THE RAM
2555 CLR R2 ;INIT BYTE NUMBER
2556 5$: CMPB (R1)+,(R4)+ ;COMPARE EXPECTED, RECEIVED
2557 BNE 7$ ;BR IF NO MATCH
2558 FORCERROR 7$,NOTSSR
2559 BR 10$ ;@@D
2560 7$: MOVB -1(R1),R5 ;GET RECV RAM DATA
2561 MOVB -1(R4),R3 ;GET EXPD PACKET DATA
2562 XOR R5,R3 ;XOR EXPD/RECV
2563 BIC #177400,R3 ;LOW BYTE ONLY
2564 MOVB -1(R1),RECV ;GET RECEIVED RAM DATA
2565 MOVB -1(R4),EXPD ;GET EXPECTED RAM DATA
2566 PRINTB #RAMASC,R2,RECV,EXPD,R3
2567 MOV R3,-(SP)
2568 MOV EXPD,-(SP)
2569 MOV RECV,-(SP)
2570 MOV R2,-(SP)
2571 MOV #RAMASC,-(SP)
2572 MOV #5,-(SP)
2573 MOV SP,R0
2574 TRAP C$PNTB
2575 ADD #14,SP
2576 10$: INC R2 ;UPDATE BYTE COUNT
2577 TST RAMSIZ ;DEFAULT TO 8.?
2578 BEQ 15$ ;BR IF YES
2579 CMP R2,RAMSIZ ;DONE ALL BYTES?
2580 BLE 5$ ;BR IF NO
2581 BR 25$ ;
2582 15$: CMP R2,#8. ;DONE DEFAULT NUMBER OF BYTES?
2583 BLT 5$ ;BR IF NO
2584 25$: CLR RAMSIZ ;SET DEFAULT RAMSIZ
2585 RTS PC ;RETURN
2586 2578 014276 045 116 045 RAMASC: .ASCIZ 'N#A BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03'
2579 .EVEN

```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

2581 .SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
2582 ;*
2583 ;THIS ROUTINE PRINTS THE CONTENTS OF
2584 ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE TSV-05.
2585 ;
2586 ;INPUT:
2587 ; R0 LOW ORDER ADDRESS OF MESSAGE BUFFER
2588 ; R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER
2589 ; NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
2590 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
2591 ;-
2592 PRMESS: SAVREG ;SAVE THE REGISTERS
2593 MOV R0,R5 ;SAVE LOW ORDER ADDRESS
2594 TST KTENABLE ;ADDRESS ABOVE 28K?
2595 BNE 10$ ;BR IF YES
2596 CLR R1 ;SET HIGH ORDER ADDRESS TO 0
2597 10$: MOV R1,R3 ;SAVE HIGH ORDER ADDRESS
2598 ROL R0 ;SHIFT BIT15 TO C BIT
2599 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
2600 PRINTX #PROASC,R1,R5 ;PRINT MESSAGE BUFFER ADDRESS
      MOV R5,-(SP)
      MOV R1,-(SP)
      MOV #PROASC,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C$PNTX
2601 10$: PRINTX #PRIASC ;PRINT HEADER FOR CONTENTS
      MOV #PRIASC,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
      TRAP C$PNTX
      ADD #4,SP
2602 CLR R4 ;NUMBER OF THE NEXT WORD
2603 MOV R5,R1 ;COPY LOW ORDER ADDRESS
2604 MOV R3,R0 ;COPY HIGH ORDER ADDRESS
2605 BEQ 20$ ;BR IF NOT ABOVE 28K
2606 JSR PC,SETMAP ;SETUP PAR ADDRESS IN R0
2607 MOV R0,R5 ;GET PAR FORMAT ADDRESS ABOVE 28K
2608 20$: PRINTX #PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
      MOV (R5)+,-(SP)
      MOV R4,-(SP)
      MOV #PRASC,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C$PNTX
      ADD #10,SP
2609 INC R4 ;NUMBER OF THE NEXT
2610 CMP R4,#7 ;DONE ALL YET ?
2611 BGT 50$ ;BRANCH IF ALL DONE
2612 BLT 20$ ;PRINT FIRST 7 WORDS
2613 BIT #X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
2614 BNE 20$ ;PRINT EXTENDED STATUS WORD
2615 RTS PC ;RETURN
2616 PROASC: .ASCIZ '%N%A Message Buffer Address = %01%05'
2617 PR1ASC: .ASCIZ '%N%A Message Buffer Contents:'
2618 PRASC: .ASCIZ '%N%A Word%D1%A: %0'

```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

2620 .EVEN
2621 .SBTTL PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
2622 ;*
2623 ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
2624 ; RO - NUMBER OF WORDS IN BUFFER
2625 ;IMPLICIT INPUTS:
2626 ; EXPMSG - EXPECTED MESSAGE BUFFER
2627 ; RECMMSG - RECEIVED MESSAGE BUFFER
2628 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2629 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2630 ;-
2631 PRMSGEXP::
2632 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2633 MOV RO,R5 ;SAVE NUMBER OF WORDS
2634 MOV RCVLOADD,RO ;GET RECV LOW ADDRESS
2635 MOV RO,R4 ;COPY LOW ADDRESS
2636 MOV RCVHIADD,R1 ;GET RECV HIGH ADDRESS
2637 ROL RO ;SHIFT BIT15 TO C BIT
2638 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
2639 PRINTX #PRMSG0,R1,R4 ;PRINT MESSAGE BUFFER ADDRESS
    014716 010446 MOV R4,-(SP)
    014720 010146 MOV R1,-(SP)
    014722 012746 015052 MOV #PRMSG0,-(SP)
    014726 012746 000003 MOV #3,-(SP)
    014732 010600 MOV SP,RO
    014734 104415 TRAP C:PNTX
    014736 062706 000010 ADD #10,SP
2640 PRINTX #PRMSG1 ;PRINT HEADER FOR CONTENTS
    014742 012746 015117 MOV #PRMSG1,-(SP)
    014746 012746 000001 MOV #1,-(SP)
    014752 010600 MOV SP,RO
    014754 104415 TRAP C:PNTX
    014756 062706 000004 ADD #4,SP
2641 CLR R4 ;NUMBER OF THE CURRENT WORD
2642 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
2643 MOV #RECMMSG,R2 ;GET RECV BUFFER ADDRESS
20$: MOV (R1),RO ;GET EXPD
    MOV (R2),R3 ;GET RECV
    XOR RO,R3 ;XOR EXPD/RECV
2646 PRINTX #PRMSG2,R4,(R1)*,(R2)*,R3
    015010 010346 MOV R3,-(SP)
    015012 012246 MOV (R2)*,-(SP)
    015014 012146 MOV (R1)*,-(SP)
    015016 010446 MOV R4,-(SP)
    015020 012746 015155 MOV #PRMSG2,-(SP)
    015024 012746 000005 MOV #5,-(SP)
    015030 010600 MOV SP,RO
    015032 104415 TRAP C:PNTX
    015034 062706 000014 ADD #14,SP
2648 INC R4 ;NUMBER OF THE NEXT
2649 CMP R4,R5 ;DONE ALL YET?
2650 BGE 50$ ;BR IF YES
2651 BR 20$ ;DO ANOTHER
2652 RTS PC ;RETURN
000207
50$: PRMSG0: .ASCIZ '#N#A Message Buffer Address = #01#05'
2653 015052 045 116 045 PRMSG1: .ASCIZ '#N#A Message Buffer Contents:'
2654 015117 045 116 045 PRMSG2: .ASCIZ '#N#A WORD #D2#A EXPD: #06#A RECV: #06#A XOR: #06#'
2655 015155 045 116 045

```

PRMSGEXP - PRINT EXPD/RCV MESSAGE BUFFERS

```

2657 .EVEN
2658 .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
2659
2660 ;*
2661 ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
2662 ; ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
2663 ;
2664 ; RO - NUMBER OF BYTES IN BUFFER
2665 ;
2666 ;IMPLICIT INPUTS:
2667 ;
2668 ; EXPMSG - EXPECTED MESSAGE BUFFER
2669 ; RECMMSG - RECEIVED MESSAGE BUFFER
2670 ;-
2671 PRBYTEXP::
2672 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2673 MOV R0,R5 ;SAVE NUMBER OF BYTES
2674 CLR PRMNO ;INIT ERROR COUNT
2675 CLR R4 ;NUMBER OF THE CURRENT BYTE
2676 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
2677 MOV #RECMMSG,R2 ;GET RECV BUFFER ADDRESS
2678 20$: MOV (R1),R0 ;GET EXPD BYTE
2679 BIC #1C<377>,R0 ;CLEAR UPPER BYTE
2680 MOV R0,PRBEXP ;SAVE FOR ERROR REPORT
2681 MOV (R2),R3 ;GET RECV BYTE
2682 BIC #1C<377>,R3 ;CLEAR UPPER BYTE
2683 MOV R3,PRBREC ;FOR ERROR REPORT
2684 XOR R0,R3 ;XOR EXPD/RCV
2685 CMPB (R1)+,(R2)+ ;EXPD = RECV?
2686 BEQ 30$ ;BR IF YES
2687 INC PRMNO ;UPDATE ERROR COUNT
2688 CMP PRMNO,#8. ;PRINTED 8?
2689 BHI 30$ ;BR IF YES
2690 27$: PRINTX #PRBMSG,R4,PRBEXP,PRBREC,R3
2691 MOV R3,-(SP)
2692 MOV PRBREC,-(SP)
2693 MOV PRBEXP,-(SP)
2694 MOV R4,-(SP)
2695 MOV #PRBMSG,-(SP)
2696 MOV #5,-(SP)
2697 MOV SP,R0
2698 TRAP C#PNTX
2699 ADD #14,SP
2700 FORCEEXIT 50$ ;@ad
2701 BR 35$ ;@ad
2702 30$: FORCERROR 27$,NOTSSR ;@ad
2703 35$: ;@ad
2704 INC R4 ;NUMBER OF THE NEXT
2705 CMP R4,R5 ;DONE ALL YET?
2706 BGE 50$ ;BR IF YES
2707 BR 20$ ;DO ANOTHER
2708 50$: PRINTX #PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
2709 MOV PRMNO,-(SP)
2710 MOV #PRBTOT,-(SP)
2711 MOV #2,-(SP)
2712 MOV SP,R0

```

E6

PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

015446 104415 TRAP C$PNTX
015450 062706 000006 ADD #6,SP
2701 015454 000207 RTS PC ;RETURN
2702
2703 015456 045 116 045 PRBMSG: .ASCIZ '%N%A BYTE #D2%A EXPD: %03%A RECV: %03%A XOR: %03'
2704 015543 045 116 045 PRBTOT: .ASCIZ '%N%A NUMBER OF BYTES IN ERROR = %D2'
2705 .EVEN
2706 015610 000000 PRBEXP: .WORD 0 ;EXPD
2707 015612 000000 PRBREC: .WORD 0 ;RECV
2708 .SBTTL EXPREC - PRINT EXPD/RECV WORD DATA
2709
2710 ;*
2711 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2712 ;
2713 ;INPUTS:
2714 ;
2715 ; R1 RECEIVED DATA
2716 ; R2 EXPECTED DATA
2717 ;
2718 ;-
2719
2720 015614 BGNMSG EXPREC
015614 EXPREC::
2721 015614 004737 010034 JSR PC,PRIXOR ;PRINT THE DATA
2722 015620 ENDMSG
015620 L10017:
015620 104423 TRAP C$MSG
.SBTTL EXPBREC - PRINT EXPD/RECV BYTE DATA
2723
2724 ;*
2725 ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
2726 ;
2727 ;INPUTS:
2728 ;
2729 ; R1 RECEIVED DATA BYTE
2730 ; R2 EXPECTED DATA BYTE
2731 ;
2732 ;-
2733
2734
2735
2736 015622 BGNMSG EXPBREC
015622 EXPBREC::
2737 015622 004737 007704 JSR PC,PRIBXOR ;PRINT THE DATA
2738 015626 ENDMSG
015626 L10020:
015626 104423 TRAP C$MSG
.SBTTL RAMERR - PRINT RAM AND PACKET DATA
2739
2740 ;*
2741 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
2742 ;
2743 ;INPUTS:
2744 ;
2745 ; R4 POINTER TO COMMAND PACKET
2746 ;
2747 ;
2748 ;
2749 ;

```

F6

RAMERR - PRINT RAM AND PACKET DATA

```

2750      ;IMPLICIT INPUTS:
2751      ;
2752      ;      RAMDATA      DATA AS READ FROM THE RAM
2753      ;      RAMSIZ      NUMBER OF BYTES IN PACKET
2754      ;                      IF RAMSIZ=0 THEN DEFAULT TO 8.
2755      ;
2756      ;IMPLICIT OUTPUTS:
2757      ;
2758      ;      RAMSIZ  SET TO 0
2759      ;
2760      ;
2761 015630      BGNMSG  RAMERR
015630
2762 015630 004737 014116      RAMERR:: JSR  PC,PRAMPKT      ;PRINT RAM/PACKET DATA
2763 015634      ENDMSG
015634
2764 015634 104423      L10021: TRAP  C$MSG
2765      .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
2766      ;*
2767      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
2768      ;
2769      ;INPUTS:
2770      ;
2771      ;      R4      POINTER TO COMMAND PACKET
2772      ;
2773      ;IMPLICIT INPUTS:
2774      ;
2775      ;      RAMDATA      DATA AS READ FROM THE RAM
2776      ;      RAMSIZ      NUMBER OF BYTES IN PACKET
2777      ;                      IF RAMSIZ=0 THEN DEFAULT TO 8.
2778      ;      ERRHI      HIGH ORDER TEST ADDRESS
2779      ;      ERRLO      LOW ORDER TEST ADDRESS
2780      ;
2781      ;IMPLICIT OUTPUTS:
2782      ;
2783      ;      RAMSIZ  SET TO 0
2784      ;
2785      ;
2786      ;
2787 015636      BGNMSG  RAMTADD
015636
2788 015636 004737 010366      RAMTADD:: JSR  PC,PRITADD      ;PRINT TEST ADDRESS
2789 015642 004737 014116      JSR  PC,PRAMPKT      ;PRINT RAM/PACKET DATA
2790 015646      ENDMSG
015646
2791 015646 104423      L10022: TRAP  C$MSG
2792      .SBTTL  RAMEXP - PRINT RAM EXPD/RECV DATA
2793      ;*
2794      ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2795      ;
2796      ;INPUTS:
2797      ;
2798      ;      R1      RECEIVED DATA
2799      ;      R2      EXPECTED DATA
2800      ;

```


G6

RAMEXP - PRINT RAM EXPD/RECV DATA

```

2801          ;      R4      CONTROLLER RAM ADDRESS
2802          ; -
2803
2804 015650    BGNMSG  RAMEXP
      015650
2805 015650    042701  177400
2806 015654    042702  177400
2807 015660    004737  010160
2808 015664    004737  010034
2809 015670
      015670
      015670    104423
2810
2811
2812          ;
2813          ; *
2814          ; PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2815          ; AND TIMER A,B HEADER MESSAGE
2816          ;
2817          ; INPUTS:
2818          ;
2819          ;      R1      RECEIVED DATA
2820          ;      R2      EXPECTED DATA
2821          ; -
2822
2823 015672    BGNMSG  TIMEXP
      015672
2824 015672    TIMEXP:  PRINTX  #TIMSGO          ;PRINT HEADER
      015672    012746  015720
      015676    012746  000001
      015702    010600
      015704    104415
      015706    062706  000004
2825 015712    004737  010034
2826 015716
      015716
      015716    104423
2827
2828 015720    045      116      045  TIMSGO: .ASCIZ 'N#A TIMER A STATUS IS IN BIT 3#N#A TIMER B STATUS IS IN BIT 2'
2829          .EVEN
2830          .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
2831
2832          ;
2833          ; *
2834          ; PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
2835          ;
2836          ; INPUTS:
2837          ;
2838          ;      R1      CONTENTS OF TSSR
2839          ;      R2      DATA WRITTEN (8 BITS)
2840          ;
2841          ; -
2842
2843 016020    BGNMSG  BADSSR
      016020
2844 016020    010246
2845 016022    042702  177400
      BADSSR:  MOV      R2, -(SP)          ;SAVE DATA TRANSFERRED
              BIC      #177400,R2      ;GET JUST ONE BYTE

```

H6

BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

```

2846 016026          PRINTB #XFERASC,R2
      016026 010246      MOV R2,-(SP)
      016030 012746 016060 MOV #XFERASC,-(SP)
      016034 012746 000002 MOV #2,-(SP)
      016040 010600      MOV SP,R0
      016042 104414      TRAP C#PNTB
      016044 062706 000006 ADD #6,SP
2847 016050 012602      MOV (SP),R2          ;RESTORE R2
2848 016052 004737 006024 JSR PC,PRITSSR      ;DECODE TSSR CONTENTS
2849 016056          ENDMSG
      016056          L10025:
      016056 104423      TRAP C#MSG
2850 016060 045 116 045 XFERASC: .ASCIZ '#N#A Data Transferred = #03'

```

GLOBAL SUBROUTINES SECTION

```

2852      .SBTTL GLOBAL SUBROUTINES SECTION
2853
2854      ;**
2855      ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2856      ; THAT ARE USED IN MORE THAN ONE TEST.
2857      ;--
2858      .SBTTL SOFINIT - SOFT INITIALIZE OF CONTROLLER
2859
2860      ;*
2861      ; ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
2862      ; BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
2863      ; THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
2864      ; DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
2865      ;
2866      ; INPUTS:
2867      ;
2868      ;       R5      ADDRESS OF FIRST REGISTER
2869      ;
2870      ; OUTPUTS:
2871      ;
2872      ;       R0      CONTENTS OF TSSR, IF ERROR
2873      ;       CARRY   SET IF INIT WAS OKAY
2874      ;               CLEAR IF FATAL ERROR
2875      ;
2876      ; CALLING SEQUENCE:
2877      ;
2878      ;       MOV     #ADDRESS,R5
2879      ;       JSR     PC,SOFINIT
2880      ;       BCS    CONTINUE
2881      ;       ERRDF
2882      ;               ;REPORT FATAL ERROR
2883      ;
2884      ;-
2885
2886 016114 SOFINIT::
2887 016114 SAVREG
2888 016120 012765 000000 000002 MOV #0,TSSR(R5) ; SAVE THE REGISTERS
2889 016126 004737 016370 JSR PC,WAITF ; DO THE INIT.
2890 016132 016500 000002 MOV TSSR(R5),R0 ; WAIT FOR SSR
2891 016136 010004 MOV R0,R4 ; GET THE TSSR REGISTER
2892 016140 042704 176277 BIC #1<HIADDR!OFL>,R4 ; TSSR CONTENTS
2893 016144 052704 002200 BIS #SSR!NBA,R4 ; R4 HAS EXPECTED CONTENTS
2894 016150 020400 CMP R4,R0 ; ONLY EXPECTED BITS SET ?
2895 016152 001402 BEQ 5$ ; BRANCH IF OKAY
2896 016154 000241 CLC ; CLEAR THE CARRY FOR ERROR
2897 016156 000401 BR 10$ ; GO TO EXIT
2898 016160 000261 5$: SEC ; SET THE CARRY BIT
2899 016162 000207 10$: RTS PC ; RETURN TO CALLER

```

CHKAMB - CHECK TSSR FOR AMBIGUITY

```

2901 .SBTTL CHKAMB - CHECK TSSR FOR AMBIGUITY
2902
2903
2904 ;*
2905 ;THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
2906 ;FOR AMBIGUITY
2907
2908 ;INPUT:
2909
2910 ;      RO      CONTENTS OF TSSR
2911
2912 ;OUTPUT:
2913
2914 ;      RO      CONTENTS OF TSSR
2915
2916 ;      CARRY   SET - NO AMBIGUITY
2917 ;              CLR - AMBIGUOUS CONTENTS
2918
2919 ;-
2920
2921 CHKAMB:
2922 SAVREG
2923 MOV      RO,R4          ;SAVE THE GENERAL REGISTERS
2924 BIT     #SC,RO         ;CONTENTS OF TSSR
2925 BNE     5$            ;IS BIT 15 SET ?
2926 BIT     #C<NBA!OFL!SSR!HIADDR>,RO ;ANY OTHER BITS SET ?
2927 BNE     40$          ;MUST BE AN ERROR
2928 BR      45$          ;RETURN WITH SUCCESS
2929 5$:    BIT     #SSR,RO ;IS READY BIT SET ?
2930 BNE     10$          ;BRANCH IF READY BIT IS SET.
2931 BIT     #BITS,RO     ;IS FATAL ERROR BIT SET ?
2932 BEQ     40$          ;ERROR IF NOT
2933 BIC     #TERCLS,R4   ;CLEAR ALL BUT TERMINATION CODE
2934 CMP     R4,#16       ;ALL THREE BITS MUST BE SET
2935 BNE     40$          ;ERROR IF NOT SET
2936 BR      45$          ;OK IF ALL ARE SET
2937 10$:   BIT     #BITS,RO ;IS FATAL ERROR BIT SET ?
2938 BEQ     45$          ;ERROR IF BIT IS SET WITH SSR
2939 BIT     #BIT2:BIT1,RO ;IS THIS A FUNCTION REJECT
2940 BNE     45$          ;BR, IF TSSR IS OK
2941 40$:   CLC
2942 BR      50$          ;AMBIGUOUS CONTENTS
2943 45$:   SEC
2944 50$:   RTS          ;SHOW SUCCESS - NO AMBIGUITY
                ;RETURN TO CALLER

```

ENAINIT,DSBINT - ENABLE/DISABLE INTERRUPTS

```

2946      .SBTTL ENAINIT,DSBINT - ENABLE/DISABLE INTERRUPTS
2947      ;
2948      ; DEFAULT DISPLAY INTERRUPT HANDLERS.
2949      ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2950      ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
2951      ;
2952      ;
2953      ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2954      ;
2955      ;           IOKCKIN=BIT7      ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
2956      ;           IOKSTP=BIT0      ; EXPECT "STOP" INTERRUPT.
2957      ;
2958      ; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
2959      INTMASK:      .BYTE 0
2960      ; INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2961      INTFLAG:     .BYTE 0
2962      ;
2963      ; SAVED INTERRUPT VECTOR:
2964      INTVEC:      .WORD 0
2965      ; SAVE CPU PC
2966      INTCPC:     .WORD 0
2967      ;
2968      ; SUBROUTINE TO ENABLE INTERRUPTS:
2969      ENAINIT:    MOV     RO,-(SP)      ;SAVE RO
2970      ;           MOV     IVEC,RO      ;GET POINTER TO VECTORS
2971      ;           MOV     #INTR,(RO)+  ;SET UP INTERRUPT VECTOR
2972      ;           MOV     #PRI07,(RO)+
2973      ;           MOV     (SP)+,RO    ;RESTORE RO
2974      ;           MOV     (SP),-(SP)
2975      ;           MOV     #0,2(SP)    ;SET CPU TO LEVEL 0
2976      ;           RTI
2977      ;
2978      ; SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2979      DSBINT:    MOV     (SP),-(SP)
2980      ;           MOV     #PRI07,2(SP)
2981      ;           RTI
2982      ;           .SBTTL INTR      - INTERRUPT HANDLERS
2983      ;
2984      ;           BGNSRV INTR      ;DEFINE INTERRUPT ENTRY
2985      INTR::    MOV     #1,INTRECV    ;SET FLAG TO SHOW INTERRUPT RECEIVED
2986      ;           CLR    INTFLAG    ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2987      ;           BIT    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2988      ;           BNE    1$         ;BR IF YES
2989      ;           BIS    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2990      ;
2991      ; SAVE REGISTERS, MSG BUFFER, ETC.
2992      1$:
2993      ENDSRV
L10026:    RTI
016366    000002

```

L6

WAITF - WAIT FOR SUBSYSTEM READY

```

2995          .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
2996          ;
2997          ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2998          ;
2999          ; INPUTS:
3000          ;
3001          ; R5 ADDRESS OF FIRST DEVICE REGISTER
3002          ;
3003          ; OUTPUTS:
3004          ;
3005          ; R0 CONTENTS OF LAST TSSR READ
3006          ; CARRY SET - READY BIT SET
3007          ; CLR - TIMEOUT WAITING FOR READY
3008          ;
3009 016370 000401 WAITF:: BR 1$ ;NOP WHEN SUPER FIXED
3010 016372          BREAK          ; DO A SUPVSR BREAK FIRST.
          016372 104422 TRAP C$BRK
3011 016374 012746 011000 1$: MOV #11000,-(SP) ;25-APRIL-83 REV B - 1100 MSEC TIMER
3012 016400 016500 000002 2$: MOV TSSR(R5),R0 ;READ THE TSSR REGISTER
3013 016404 105700          TSTB R0 ;TEST FOR READY BIT SET
3014
3015 016406 100420          BMI 3$ ; EXIT ON STOP FLAG.
3016 016410          DELAY 1 ; WAIT 100 USEC
          016410 012727 000001 MOV #1,(PC)+
          016414 000000 .WORD 0
          016416 013727 002116 MOV L$DLY,(PC)+
          016422 000000 .WORD 0
          016424 005367 177772 DEC -6(PC)
          016430 001375 BNE -.4
          016432 005367 177756 DEC -22(PC)
          016436 001367 BNE -.20
3017 016440 005316 DEC (SP) ;REDUCE DELAY COUNT
3018 016442 001356 BNE 2$ ;RETRY UNTIL TIMER EXPIRES
3019 016444 000241 CLC ; C = 0, CONTROLLER STILL RUNNING...
3020 016446 000401 BR 4$ ;...OR HUNG-UP AFTER 300 MSEC.
3021 016450 000261 3$: SEC ; C = 1, CONTROLLER IS STOPPED.
3022 016452 005326 4$: DEC (SP)+ ;RESTORE STACK WITHOUT CHANGING CARRY BIT
3023 016454 000207 RTS PC

```

CHKTSSR - CHECK TSSR FOR READY

```

3025 .SBTTL CHKTSSR - CHECK TSSR FOR READY
3026 ;*
3027 ;THIS ROUTINE WAITS FOR READY IN THE TSSR
3028 ;AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
3029 ;
3030 ;INPUT:
3031 ;      R5      ADDRESS OF CSR REGISTERS
3032 ;
3033 ;OUTPUT:
3034 ;      R0      CONTENTS OF TSSR
3035 ;      CARRY   SET - OKAY
3036 ;            CLR - NOT READY AMBIGUOUS, OR SC SET
3037 ;
3038 CHKTSSR:
3039 016456 004737 016370 JSR PC, WAITF ;WAIT FOR READY
3040 016462 103014 BCC 20$ ;BRANCH IF TIME OUT
3041 016464 004737 016164 JSR PC, CHKAMB ;TSSR AMBIGUOUS?
3042 016470 103006 BCC 10$ ;BR IF YES
3043 016472 032700 100000 BIT #SC, R0 ;SPECIAL CONDITION SET?
3044 016476 001405 BEQ 15$ ;BR IF NO
3045 016500 032700 074000 BIT #<SCE!BIE!RMR!NXM>, R0 ;ANY ERROR BITS SET?
3046 016504 001402 BEQ 15$ ;BR IF NO
3047 016506 000241 10$: CLC ;SET FAILURE
3048 016510 000401 BR 20$ ;
3049 016512 000261 15$: SEC ;SET SUCCESS
3050 016514 000207 20$: RTS PC ;RETURN TO CALLER
3051 .SBTTL XNXM - CHECK FOR NONEXISTENT MEMORY
3052 ;*
3053 ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
3054 ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
3055 ; "C" = 0, ALL ADDRESSES OK.
3056 ;
3057 ;CALL: MOV ADR1, R1
3058 ;      MOV ADR2, R2
3059 ;      JSR PC, NXM
3060 ;      RETURN ;TEST "C" AND PROCEED.
3061 016516 012737 016550 000004 XNXM: MOV #2$, @#4 ; SET BUSERR VECTOR.
3062 016524 012737 000200 000006 MOV #PRI04, @#6
3063 016532 005003 CLR R3 ;FLAG.
3064 016534 005711 1$: TST (R1) ;TEST THE ADDRESS(ES).
3065 ;IF ANY TRAP, CONTINUE AT 2$.
3066 016536 020102 CMP R1, R2 ;OTHERWISE, CONTINUE HERE.
3067 016540 001407 BEQ 3$ ;BR IF FINISHED (NO NEXM'S).
3068 016542 062701 000002 ADD #2, R1 ;SET NEXT ADDRESS...
3069 016546 000772 BR 1$ ;...AND CONTINUE.
3070 016550 005103 2$: COM R3 ;GOT ONE, SET FLAG...
3071 016552 012716 016560 MOV #3$, (SP)
3072 016556 000002 RTI ;...AND DISMISS INTERRUPT...
3073 016560 3$: CLRVEC #4 ;...AND GIVE BACK THE VECTOR.
3074 016560 012700 000004 MOV #4, R0
3075 016564 104436 TRAP C$CVEC
3076 016566 005703 TST R3 ;DID WE CATCH ONE ??
3077 016570 001401 BEQ .+4 ;NO, "C" = 0, SKIP NEXT.
3078 016572 000261 SEC ;YES, "C" = 1, (R1) = NEXM ADDR.
3079 016574 000207 RTS PC

```

TSTLOOP - CHECK ITERATION COUNT

```

3079
3080
3081
3082
3083
3084
3085
3086
3087 016576
3088 016576 005737 002160
3089 016602 001006
3090 016604 005737 002174
3091 016610 100403
3092 016612 005337 002206
3093 016616 001002
3094 016620 000241
3095 016622 000401
3096 016624 000261
3097 016626 000207
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125 016630
3126 016630 010046
3127 016632 005037 003146
3128 016636 005037 017076
3129 016642 005037 005772
3130 016646 105037 016264
3131 016652 013700 002172
3132 016656 006300
3133 016660 005737 003106
3134 016664 001430
3135 016666 100010
    
```

```

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

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


TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```

3136 016670 052760 160000 003170      BIS      #160000,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
3137 016676      ERRDF    1,NXR,NXRERR ; NO DEVICE HERE -- PRINT IT
      016676 104455      TRAP    C$ERDF
      016700 000001      .WORD   1
      016702 003740      .WORD   NXR
      016704 005736      .WORD   NXRERR
3138 016706 000407      BR      2$
3139 016710 052760 160001 003170 3$:  BIS      #160001,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
3140 016716      ERRDF    2,NOINIT ; DEVICE NOT IDLE
      016716 104455      TRAP    C$ERDF
      016720 000002      .WORD   2
      016722 004335      .WORD   NOINIT
      016724 000000      .WORD   0
3141 016726 012737 177777 003104 2$:  MOV      #-1,DUFLG ; DROP THE UNIT
3142 016734      DODU     UNITN
      016734 013700 002172      MOV      UNITN,RO
      016740 104451      TRAP    C$DODU
3143 016742      DOCLN   ; ABORT THE PASS
      016742 104444      TRAP    C$DCLN
3144 016744 000423      BR      5$
3145
3146 016746      RFLAGS   RO ; GET THE OPERATOR FLAGS.
      016746 104421      TRAP    C$RFLA
3147 016750 032700 001000      BIT      #PNT,RO ; PRINT THE TEST NUMBERS?
3148 016754 001412      BEQ     1$ ; BR IF NO
3149 016756 011600      MOV      (SP),RO ;GET THE ID MESSAGE
3150 016760      PRINTF   #TNAM,RO ;DISPLAY THE TEST ID
      016760 010046      MOV      RO,-(SP)
      016762 012746 017024      MOV      #TNAM,-(SP)
      016766 012746 000002      MOV      #2,-(SP)
      016772 010600      MOV      SP,RO
      016774 104417      TRAP    C$PNTF
      016776 062706 000006      ADD      #6,SP
3151 017002 005237 002204      1$:  INC      TSTCNT ; BUMP TEST COUNTER.
3152 017006      SETPRI   IPRI ;PRIORITY THAT OF DEVICE
      017006 013700 002202      MOV      IPRI,RO
      017012 104441      TRAP    C$SPRI
3153 017014 005726      5$:  TST      (SP)+ ;FIX UP THE STACK
3154 017016 013705 002176      MOV      CSRADDR,R5 ; ADDRESS OF TSV REGISTERS ON UNIBUS
3155 017022 000207      RTS     PC
3156 017024 045 123 045 TNAM: .ASCIZ '#S#T#A Test'
3157 .EVEN
3158 .SBTTL TSTEND - PRINT ERRORS RECEIVED
3159
3160 ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
3161 ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
3162
3163 TSTEND: RFLAGS RO
      017040 104421      TRAP    C$RFLA
3164 017042 030027 020000      BIT      RO,#IER
3165 017046 001412      BEQ     1$ ; BR IF "IER" NOT SET.
3166 017050      PRINTF   #ESUM,ERRK ; PRINT ERROR COUNT.
      017050 013746 017076      MOV      ERRK,-(SP)
      017054 012746 017100      MOV      #ESUM,-(SP)
      017060 012746 000002      MOV      #2,-(SP)
      017064 010600      MOV      SP,RO
      017066 104417      TRAP    C$PNTF

```

C7

TSTEND - PRINT ERRORS RECEIVED

3167	017070	062706	000006							
3168	017074	000207			1\$:	ADD	#6.SP			
3169	017076	000000				RTS	PC			
3170	017100	045	101	040	ERRK:	0				; LOCAL ERROR COUNT.
3171	017117	105	122	122	ESUM:	.ASCIZ	/%A %D%A ERRORS/			
3172					EMAXDU:	.ASCIZ	/ERROR LIMIT REACHED -- DROPPING UNIT/			
						.EVEN				

INCERK - INCREMENT LOCAL ERROR COUNT

```

3174                                     .SBTTL INCERK - INCREMENT LOCAL ERROR COUNT
3175
3176 ;* ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
3177 ;
3178 INCERK: INC ERRK ; INCREMENT LOCAL ERROR COUNT
3179 MOV RO,-(SP) ; SAVE RO
3180 MOV UNITN,RO ; GET UNIT NUMBER,
3181 ASL RO ; ... AND MAKE IT A WORD OFFSET.
3182 ADD #ERTABL,RO ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
3183 INC (RO) ; INCREMENT THE DEVICE ERROR COUNT
3184 BIT #7777,(RO) ; DID WE OVERFLOW THE FIELD?
3185 BNE 1$ ; BR IF NO.
3186 DEC (RO) ; YES -- BACK IT UP TO 7777.
3187 1$: MOV (SP)+,RO ; RESTORE RO
3188 RTS PC ; RETURN TO CALLER.
3189
3190 CKEMAX: MOV RO,-(SP) ; SAVE RO
3191 MOV UNITN,RO ; GET UNIT NUMBER
3192 ASL RO ; ... AND MAKE IT A WORD OFFSET
3193 MOV ERTABL(RO),RO ; GET ERROR TABLE ENTRY
3194 BIC #17000,RO ; EXTRACT ERROR COUNT FIELD
3195 CMP RO,GERRMAX ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
3196 BHIS 1$ ; BR IF YES
3197 CMP ERRK,LERRMAX ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
3198 BLO 2$ ; BR IF NO
3199 1$: RFLAGS RO ; GET OPERATOR FLAGS
3200 TRAP C$RFLA
3201 BIT #IDU,RO ; IS DROPPING INHIBITED?
3202 BNE 2$ ; BR IF YES.
3203 MOV #-1,DUFLG ; NO -- DROP THE UNIT
3204 ERRDF 4,EMAXDU
3205 TRAP C$ERDF
3206 .WORD 4
3207 .WORD EMAXDU
3208 .WORD 0
3209 DODU UNITN
3210 MOV UNITN,RO
3211 TRAP C$DODU
3212 DOCLN
3213 TRAP C$DCLN
3214 2$: MOV (SP)+,RO ; RESTORE RO
3215 RTS PC ; RETURN TO CALLER

```

E7

CKDROP - CHECK IF UNIT SHOULD BE DROPPED

```

3209 .SBTTL CKDROP - CHECK IF UNIT SHOULD BE DROPPED
3210 ;*
3211 ; CHECK IF UNIT SHOULD BE DROPPED
3212 ;-
3213 017322 010046 CKDROP: MOV RO, -(SP)
3214 017324 FORCERROR 1$,NOTSSR
3215 017334 RFLAGS RO
3216 017336 104421 TRAP C$RFLA
3217 017342 032700 000040 BIT #IDU,RO
3218 017344 001010 BNE 1$
3219 017346 011600 MOV (SP),RO
3220 017354 012737 177777 003104 MOV #-1,DUFLG
3221 017354 DODU UNITN
3222 017354 013700 002172 MOV UNITN,RO
3223 017360 104451 TRAP C$DODU
3224 017362 DOCLN ;ABORT THE PASS
3225 017362 104444 TRAP C$DCLN
3226 017364 012600 1$: MOV (SP)+,RO
3227 017366 000207 RTS PC

.SBTTL CONFIG - DETERMINE CONFIGURATION OF SYSTEM
; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
; CONFIG:
3230 017370 JSR PC,SOFINIT
3231 017370 004737 016114 RTS PC
3232 017374 000207 .SBTTL KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
3233 ;
3234 ; SUBROUTINE - ENABLE MEM MGT.
3235 ;
3236 ;
3237 017376 005737 003124 KTON: TST KTF LG ; GOT KT?
3238 017402 001403 BEQ 1$ ; NO.
3239 017404 012737 000001 177572 MOV #1,SRO ; YES. ENABLE KT11.
3240 017412 000207 1$: RTS PC
3241 ;
3242 ; SUBROUTINE - DISABLE MEM MGT.
3243 ;
3244 ;
3245 017414 005737 003124 KTOFF: TST KTF LG ; GOT KT11?
3246 017420 001405 BEQ 1$ ; NO.
3247 017422 000240 NOP
3248 017424 000240 NOP
3249 017426 012737 000000 177572 MOV #0,SRO ; DISABLE KT.
3250 017434 000207 1$: RTS PC

```

F7

SETMAP - SETUP PAR6 MAPPING

```

3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271 017436
3272 017436
3273 017442 005737 003124
3274 017446 001433
3275 017450 010102
3276 000006
3277
3278
3279
3280 017502 042701 000177
3281 017506 020137 003124
3282 017512 103011
3283 017514 010137 172352
3284 017520 042702 160000
3285 017524 062702 120000
3286 017530 010200
3287 017532 000261
3288 017534 000401
3289 017536 000241
3290 017540 000207
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306 017542
3307 017542
3308 017546 004737 017414

```

```

.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:
;
;      RO      HIGH ORDER ADDRESS BITS
;      R1      LOW ORDER ADDRESS BITS
;
;OUTPUTS:
;
;      RO      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
  ROR  R1                                ;CONVERT WORD ADDRESS TO 32W BLOCKS
  .ENDR                                  ;MAKE IT DOUBLE PRECISION
  BIC  #177,R1                           ;ALINE FOR LOWER 4K BOUNDARY
  CMP  R1,KTFLG                          ;HIGHER THAN EXISTING MEMORY?
  BHS  10$                                ;BR IF YES
  MOV  R1,#KIPARS                         ;SETUP MAPPING REGISTER PARS
  BIC  #160000,R2                        ;SETUP DISPLACEMENT IN PAGE
  ADD  #120000,R2                          ;ADD IN PARS BIAS
  MOV  R2,R0                              ;RETURN IN R0
  SEC                                     ;SET SUCCESS
  BR   15$
10$:  CLC                                 ;SET FAILURE
15$:  RTS  PC                             ;RETURN
.SBTTL FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
;*
; FILL MEMORY WITH A BACKGROUND PATTERN
;
; INPUTS:
;
;      RO = BACKGROUND PATTERN
;      FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
;      KTFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
;
; OUTPUTS:
;
;      NONE
;
;-
FILLMEM:
  SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
  JSR  PC,KTOFF                          ;DISABLE KT.

```

G7

FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN

```

3309 017552 010003      MOV      R0,R3      ;COPY TEST PATTERN
3310 017554 013701 003116  MOV      FREE,R1    ;GET FIRST FREE LOCATION
3311 017560 013702 003120  MOV      FRESIZ,R2  ;SIZE OF FREE SPACE BELOW 28K.
3312 017564 010321      MOV      R3,(R1)+   ;STORE A BACKGROUND WORD
3313 017566 005302      DEC      R2         ;DONE ALL MEMORY IN FREE SPACE?
3314 017570 003375      BGT     10$        ;BR IF NO
3315 017572 005737 003124  TST     KTFLG      ; GOT KT?
3316 017576 001477      BEQ     55$        ; NO. GET OUT.
3317 017600 004737 017376  JSR     PC,KTON    ; YES. ENABLE KT.
3318 017604 005000      CLR     R0         ;HIGH ORDER ADDRESS START
3319 017606 013701 003144  MOV     PST32W,R1  ;GET >28K START ADDRESS (IN 32W BLOCKS)
3320                .REPT 6
3321                CLC
3322                ROL  R1
3323                ROL  R0
3324                .ENDR
3325 017656 004737 017436  JSR     PC,SETMAP  ;SETUP PARS MAPPING REGISTER
3326 017662 010320      MOV     R3,(R0)+  ;STORE TEST PATTERN IN >28K ADDRESS
3327 017664 020027 140000  CMP     R0,#140000 ;END OF PARS MAPPING AREA?
3328 017670 103774      BLO    30$        ;BR IF NO
3329 017672 162700 020000  SUB     #20000,R0  ;BACKUP INTO PARS MAPPING BEGIN
3330 017676 062737 000200 172352  ADD     #200,@#KIPAR5 ;POINT TO NEXT 4K BLOCK >28K.
3331 017704 023727 172352 006000  CMP     @#KIPAR5,#6000 ;END OF MEMORY BELOW XMON AREA?
3332                :::
3333 017712 001427      BEQ     50$        ;BR IF YES
3334 017714 005737 003136  TST     T23A      ;11/23A?
3335 017720 001407      BEQ     35$        ;NO KEEP GOING
3336 017722 013704 177572  MOV     SRO,R4    ;GET SRO CONTENTS
3337 017726 042704 177761  BIC     #177761,R4 ;CLEAR ALL BUT PAGE NUMBER
3338 017732 022704 000016  CMP     #16,R4    ;SEE IF PAGE 7
3339 017736 001415      BEQ     50$        ;EXIT IF THERE
3340 017740 005737 003140  TST     T23B      ;11/23B?
3341 017744 001410      BEQ     45$        ;NO KEEP GOING
3342 017746 023727 172352 007600  CMP     @#KIPAR5,#7600 ;REACHED 18 BITS?
3343 017754 103001      BHIS   40$        ;YES
3344 017756 000403      BR     45$        ;NO KEEP GOING
3345 017760 012737 000020 172516 40$:  MOV     #20,S3    ;SET 22 BIT RELOCATION
3346 017766 000137 017662      45$:  JMP     30$        ;KEEP GOING ON ETC.
3347 017772 004737 017414      50$:  JSR     PC,KTOFF  ; DISABLE KT.
3348 017776 000207      55$:  RTS     PC

```

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```

3350          .SBTTL  CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
3351
3352          ;*
3353          ; COMPARE MEMORY WITH A BACKGROUND PATTERN
3354          ;
3355          ; INPUTS:
3356          ;
3357          ;     RO = BACKGROUND PATTERN
3358          ;     FREE  = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
3359          ;     KTFLG  = SET TO HIGHEST MEMORY LOCATION IF > 28K.
3360          ;
3361          ; OUTPUTS:
3362          ;
3363          ;     CARRY  - SET IF NO ERROR
3364          ;     CARRY  - CLR IF ERROR
3365          ;
3366          ; IMPLICIT OUTPUTS:
3367          ;
3368          ;     ERRHI  - ERROR HIGH ADDRESS
3369          ;     ERRLO  - ERROR LOW ADDRESS
3370          ;     EXPD   - EXPECTED DATA
3371          ;     RECV   - RECEIVED DATA
3372          ;-
3372 020000  CMPMEM:
3373 020000          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
3374 020004 010003  MOV          R0,R3      ;COPY TEST PATTERN
3375 020006 004737 017414  JSR          PC,KTOFF    ;DISABLE KT.
3376 020012 013701 003116  MOV          FREE,R1      ;GET FIRST FREE LOCATION
3377 020016 013702 003120  MOV          FRESIZ,R2    ;SIZE OF FREE SPACE BELOW 28K.
3378 020022 020311          10$:  CMP          R3,(R1)    ;FREE SPACE LOCATION EQUAL TO EXPD?
3379 020024 001411          BEQ          15$      ;BR IF YES
3380 020026 010137 002232  MOV          R1,ERRLO    ;SAVE ADDRESS IN ERROR
3381 020032 005037 002230  CLR          ERRHI      ;NO HIGH ADDRESS
3382 020036 010337 002224  MOV          R3,EXPD     ;SAVE EXPD FOR ERROR REPORT
3383 020042 011137 002226  MOV          (R1),RECV   ;SAVE RECV FOR ERROR REPORT
3384 020046 000474          BR          50$      ;
3385 020050 005721          15$:  TST          (R1)+    ;POINT TO NEXT ADDRESS
3386 020052 005302          DEC          R2      ;DONE ALL MEMORY IN FREE SPACE?
3387 020054 003362          BGT          10$      ;BR IF NO
3388 020056 005737 003124  TST          KTFLG      ; GOT KT?
3389 020062 001472          BEQ          55$      ; NO. GET OUT.
3390 020064 004737 017376  JSR          PC,KTON     ; YES. ENABLE KT.
3391 020070 005000          CLR          RO        ;HIGH ORDER ADDRESS START
3392 020072 013701 003144  MOV          PST32W,R1   ;GET >28K START ADDRESS (IN 32W BLOCKS)
3393          .REPT          6
3394          ROL          R1      ;CONVERT BLOCKS TO WORDS
3395          ROL          RO      ;MAKE IT DOUBLE PRECISION
3396          .ENDR
3397 020126 042701 000177  BIC          #177,R1    ;ALINE 4K BOUNDARY
3398 020132 010046          MOV          RO,-(SP)   ;SAVE HIGH ORDER
3399 020134 010146          MOV          R1,-(SP)   ;SAVE LOW ORDER
3400 020136 004737 017436  JSR          PC,SETMAP   ;SETUP PAR6 MAPPING REGISTER
3401 020142 010004          MOV          RO,R4      ;COPY ADDRESS BIASED TO PAR6
3402 020144 012601          MOV          (SP)+,R1   ;RESTORE LOW ORDER IN NON PAR6 FORMAT
3403 020146 012600          MOV          (SP)+,RO   ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
3404 020150 020314          30$:  CMP          R3,(R4)    ;ABOVE 28K LOCATION EQUAL EXPD?
3405 020152 001411          BEQ          32$      ;BR IF YES
3406 020154 010037 002230  MOV          RO,ERRHI   ;SAVE HIGH ORDER IN ERROR

```

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```

3407 020160 010137 002232      MOV      R1,ERRLO      ;SAVE LOW ORDER IN ERROR
3408 020164 010337 002224      MOV      R3,EXPD      ;SAVE EXPD FOR ERROR REPORT
3409 020170 011437 002226      MOV      (R4),RECV    ;SAVE RECV FOR ERROR REPORT
3410 020174 000421              BR       50$          ;
3411 020176 062701 000002      32$:    ADD      #2,R1      ;UPDATE NON PAR6 ADDRESS
3412 020202 005500              ADC      R0           ;MAKE IT DOUBLE PRECISION ADD
3413 020204 062704 000002      ADD      #2,R4      ;UPDATE PAR FORMAT ADDRESS
3414 020210 020427 140000      CMP      R4,#140000  ;END OF PAR5 MAPPING AREA?
3415 020214 103755              BLO     30$          ;BR IF NO
3416 020216 162704 020000      SUB      #20000,R4   ;BACKUP INTO PAR6 MAPPING BEGIN
3417 020222 062737 000200 172352  ADD      #200,@#KIPARS ;POINT TO NEXT 4K BLOCK >28K.
3418 020230 023737 172352 003124  CMP      @#KIPARS,KTFLG ;END OF MEMORY?
3419 020236 101744              BLOS    30$          ;BR IF NO
3420 020240 004737 017414      50$:    JSR      PC,KTOFF   ;TURN OFF MEMORY MAPPING
3421 020244 000241              CLC                    ;SET FAILURE
3422 020246 000403              BR       60$          ;
3423 020250 004737 017414      55$:    JSR      PC,KTOFF   ;TURN OFF MEMORY MAPPING
3424 020254 000261              SEC                    ;SET SUCCESS
3425 020256 000207      60$:    RTS      PC
3426              .SBTTL  REGSAV - SAVE R1-R5 ON STACK
3427              ;*
3428              ;
3429              ;ROUTINE TO
3430              ;SAVE R1 THROUGH R5 ON THE STACK
3431              ;
3432              ;CALLING SEQUENCE:
3433              ;
3434              ;      JSR      R5,REGSAV
3435              ;
3436              ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
3437              ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
3438              ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
3439              ;REGISTERS.
3440              ;
3441              ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
3442              ;CALLED VIA A JSR PC INSTRUCTION
3443              ;
3444              ;-
3445              ;
3446 020260      REGSAV:    MOV      R4,-(SP)
3447 020260      MOV      R3,-(SP)
3448 020262      MOV      R2,-(SP)
3449 020264      MOV      R1,-(SP)
3450 020266      MOV      R5,-(SP)
3451 020270      MOV      R5,-(SP)
3452 020272      MOV      10.(SP),R5
3453 020276      JSR      PC,@(SP)+
3454 020300      MOV      (SP)+,R1
3455 020302      MOV      (SP)+,R2
3456 020304      MOV      (SP)+,R3
3457 020306      MOV      (SP)+,R4
3458 020310      MOV      (SP)+,R5
3459 020312      RTS      PC

```


J7

GETPAT - GET 8 BIT PATTERN FROM OPERATOR

```

3461 .SBTTL GETPAT - GET 8 BIT PATTERN FROM OPERATOR
3462 ;+
3463 ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
3464 ;
3465 ;INPUTS: NONE.
3466 ;
3467 ;OUTPUTS:
3468 ; RO OCTAL NUMBER FROM THE OPERATOR
3469 ;
3470 ;CALLING SEQUENCE:
3471 ; JSR PC,GETPAT
3472 ;-
3473 GETPAT::
3474 SAVREG ;SAVE THE GENERAL REGISTERS
3475 1$: GMANID DATASC,PATDAT,0,377,0,377,NO
      TRAP C$GMAN
      BR 10000$
      .WORD PATDAT
      .WORD T$CODE
      .WORD DATASC
      .WORD 377
      .WORD T$LOLIM
      .WORD T$HILIM
3476 10000$: BNCOMPLETE 1$ ;RETRY IF ERROR
          BCC 1$
3477 MOV PATDAT,RO ;DATA PATTERN FROM OPERATOR
3478 RTS PC ;RETURN TO CALLER
3479
3480 ;+
3481 ;LOCAL DATA AREA
3482 ;-
3483
3484 PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
3485 DATASC: .ASCIZ 'ENTER DATA PATTERN'
3486 .EVEN

```

```

020314
020314
020320 104443
020322 000406
020324 020350
020326 000022
020330 020352
020332 000377
020334 000000
020336 000377
020340
020340 103367
020342 013700 020350
020346 000207
020350 000000
020352 105 116 124

```

GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

```

3488 .SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
3489 ;*
3490 ;ROUTINE TO ISSUE A MENU AND GET THE OPERATOR'S RESPONSE.
3491 ;
3492 ;INPUTS:
3493 ; R0 ADDRESS OF ASCIZ STRING OF MENU
3494 ; R1 MAXIMUM ALLOWABLE OPERATOR RESPONSE
3495 ;
3496 ;OUTPUTS:
3497 ; R0 NUMBER OF THE OPERATOR'S SELECTION
3498 ;-
3499 020376 GETSEL::
3500 020376 SAVREG ;SAVE GENERAL REGISTERS
3501 020402 010002 MOV R0,R2 ;SAVE THE MENU ADDRESS
3502 020404 010203 MOV R2,R3 ;START OF MENU STRING
3503 020406 005713 TST (R3) ;END OF ASCII ?
3504 020410 001412 BEQ 3$ ;BRANCH IF ALL LINES DISPLAYED
3505 020412 PRINTF #SELASC,(R3)+ ;DISPLAY THE MENU
020412 012346 MOV (R3)+,-(SP)
020414 012746 020562 MOV #SELASC,-(SP)
020420 012746 000002 MOV #2,-(SP)
020424 010600 MOV SP,R0
020426 104417 TRAP C$PNTF
020430 062706 000006 ADD #6,SP
3506 020434 000764 BR 2$
3507 020436 3$: GMANID MENASC,MENRES,D,-1,0,-1,NO
020436 104443 TRAP C$GMAN
020440 000406 BR 10001$
020442 020616 .WORD MENRES
020444 000042 .WORD T$CODE
020446 020567 .WORD MENASC
020450 177777 .WORD -1
020452 000000 .WORD T$LOLIM
020454 177777 .WORD T$HILIM
020456 10001$: BNCOMPLETE 1$ ;RETRY IF ERROR
3508 020456 BCC 1$
020456 103352 MOV MENRES,R0 ;GET THE OPERATOR'S REPLY
3509 020460 013700 020616 CMP R0,R1 ;COMPARE TO MAXIMUM ALLOWED
3510 020464 020001 BLOS 5$ ;BRANCH IF OK
3511 020466 101411 PRINTF #MENERR ;DISPLAY ERROR MESSAGE
3512 020470 MOV #MENERR,-(SP)
020470 012746 020514 MOV #1,-(SP)
020474 012746 000001 MOV SP,R0
020500 010600 TRAP C$PNTF
020502 104417 ADD #4,SP
020504 062706 000004 BR 1$ ;RETRY
3513 020510 000735 RTS PC ;RETURN TO CALLER
3514 020512 000207 5$: MENERR: .ASCIZ '#N#A *** Menu Selection Too Large ***'
3515 020514 045 116 045 SELASC: .ASCIZ '#N#T'
3516 020562 045 116 164 MENASC: .ASCIZ 'Enter Menu Selection: '
3517 020567 105 156 .EVEN
3518 .EVEN
3519 020616 000000 MENRES: .WORD 0

```

L7

CHKMAN - CHECK MANUAL INTERVENTION LEGALITY

```

3521          .SBTTL  CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
3522          ;*
3523          ;
3524          ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
3525          ;
3526          ;INPUT:
3527          ;
3528          ;     NONE.
3529          ;
3530          ;OUTPUT:
3531          ;
3532          ;     CARRY  0      MANUAL INTERVENTION NOT ALLOWED
3533          ;     CARRY  1      MANUAL INTERVENTION IS OK
3534          ;
3535          ;SIDE EFFECTS:
3536          ;
3537          ;     A MESSAGE IS DISPLAYED WARNING THAT TEST IS
3538          ;     NOT EXECUTED IF MANUAL INTERVENTION IS NOT
3539          ;     ALLOWED.
3540          ;
3541          ;-
3542
3543          020620          CHKMAN::
3544          020620          SAVREG          ;SAVE THE REGISTERS
3545          020624          MANUAL          ;SEE IF MANUAL INTERVENTION OK
3546          020624 104450          TRAP    C$MANI
3547          020626          BCOMPLETE 1$    ;BRANCH IF ALLOWED
3548          020626 103411          BCS    1$
3549          020630          PRINTF  #NOMAN   ;PRINT THE WARNING MESSAGE
3550          020630 012746 020654          MOV    #NOMAN,-(SP)
3551          020634 012746 000001          MOV    #1,-(SP)
3552          020640 010600          MOV    SP,R0
3553          020642 104417          TRAP    C$PNTF
3554          020644 062706 000004          ADD    #4,SP
3555          020650 000241          CLC          ;CLEAR CARRY FOR ERROR
3556          020652 000207          1$:      RTS    PC          ;RETURN
3557          020654 045 116 045 NOMAN: .ASCIZ  'N%A *** Manual Intervention not Allowed - Test Aborted ***'
3558          .even

```


KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3596                                     .SBTTL  KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS
3597                                     ;*
3598                                     ;
3599                                     ;ROUTINE TO INIT KT-11
3600                                     ;
3601                                     ;-
3602
3603 021130                               KTINIT:
3604 021130 005037 003124                 CLR    KTFLG           ; INIT >28K MEMORY FLAG
3605 021134 005037 003126                 CLR    KTENABLE      ; INIT TEST >28K FLAG
3606 021140 023727 002120 001577         CMP    L$HIME,#1577   ; GOT ENOUGH MEMORY (>28K)?
3607 021146 101454                       BLOS   9$            ; NO.
3608 021150 013700 000004                 MOV    @#ERRVEC,R0   ; SAVE OLD ERR VEC PTR.
3609 021154 012737 021266 000004         MOV    #2$,@#ERRVEC ; SET ERR VEC PTR.
3610 021162 005737 177572                 TST    @#SRO         ; GOT KT11?
3611 021166 000240                       NOP                    ; (TRAP IF NO).
3612 021170 013737 002120 003124         MOV    L$HIME,KTFLG ; YES. SET KT FLAG.
3613 021176 022737 007777 003124         CMP    #7777,KTFLG  ; >256K ?
3614 021204 100404                       BMI    4$            ; NO
3615 021206 042737 003777 003124         BIC    #3777,KTFLG  ; ALIGN ON BOUNDARY
3616 021214 000403                       BR     5$
3617 021216 042737 000177 003124 4$:   BIC    #177,KTFLG   ;
3618 021224 010037 000004 5$:         MOV    R0,@#ERRVEC  ; RESTORE OLD ERR VEC PTR.
3619 021230 005000                       CLR    R0            ; R0 = AR DATA.
3620 021232 012701 172340                 MOV    #KIPAR0,R1   ; R1 = KI REGS PTR.
3621 021236 012761 077406 177740 1$:   MOV    #77406,-40(R1); SET DESCRIPTOR REG.
3622 021244 010021                       MOV    R0,(R1)+     ; SET KIPAR REG.
3623 021246 062700 000200                 ADD    #200,R0      ; BUMP AR DATA BY "4K".
3624 021252 020027 002000                 CMP    R0,#2000     ; AT "I/O"?
3625 021256 001367                       BNE    1$           ; NO.
3626 021260 012741 177600                 MOV    #177600,-(R1); YES. SET KTPAR7 FOR I/O.
3627 021264 000405                       BR     9$
3628
3629 021266 012716 021274 2$:         MOV    #6$, (SP)    ; SET UP RETURN
3630 021272 000002                       RTI                    ; RTI TO NEXT LOCATION
3631
3632 021274 010037 000004 6$:         MOV    R0,@#ERRVEC  ; RESTORE OLD ERR VEC PTR.
3633
3634 021300 000207 9$:         RTS    PC

```

B8

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 08:48 Page 71

SEQ 0092

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

3636

C8

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```
3638      ;*
3639      ;      SUBROUTINE TO SET EXTENDED FEATURES SWITCH
3640      ;
3641      ;      Requires that SOFINIT and WRTCHR have been done previous to call.
3642      ;
3643      ;
3644      ;INPUTS:
3645      ;      R5      CURRENT UNIT NUMBER
3646      ;OUTPUTS:
3647      ;      The Extended Features Switch is set.
3648      ;
3649      ;-
3650
3651
```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666 021302
3667 021302 013705 002176
3668 021306 004737 016114
3669 021312 103406
3670 021314 010001
3671 021316
      021316 104455
      021320 001513
      021322 003652
      021324 012154
3672 021326 104455
3673 021330
      021330 104406
3674 021332 013737 002172 022170
3675 021340 012704 022150
3676 021344 004737 010752
3677 021350 103406
3678 021352 010001
3679 021354
      021354 104456
      021356 001513
      021360 005056
      021362 012154
3680 021364 104456
3681 021366
      021366 104406
3682 021370 013701 021500
3683 021374 032701 000200
3684 021400 001020
3685 021402 012737 100206 022220
3686 021410 012737 022230 022222
3687 021416 012737 000006 022226
3688 021424 012737 100010 022230
3689 021432 012704 022220
3690 021436 004737 010752
3691 021442 000207
3692
3693 021444 000000
3695 021450
3697 021450
3698 021450 140006
3699 021452 021460
3700 021454 000000
3701 021456 000012

;*
; 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::
      mov     csraddr,r5
      jsr     pc,sofinit
      bcs    25$
      mov     r0,r1
      errdf  errno,sfierr,sfimsg
      TRAP   C$ERDF
      .WORD  843
      .WORD  sfierr
      .WORD  sfimsg
      trap   c$erdf
25$:  ckloop
      TRAP   C$CLP1
      mov     unitn,t39dsw
      mov     #t39pk2,r4
      jsr     pc,wrtchr
      bcs    50$
      mov     r0,r1
      errhrd errno,wrtmsg,sfimsg
      TRAP   C$ERHRD
      .WORD  843
      .WORD  wrtmsg
      .WORD  sfimsg
      trap   c$erhrd
50$:  ckloop
      TRAP   C$CLP1
      mov     t39bfr+12,r1
      bit    #bit7,r1
      bne    1$
10$:  MOV     #100206,CMDPKT
      MOV     #WSMBK,CMDPKT+2
      MOV     #6,CMDPKT+6
      MOV     #100010,WSMBK
      MOV     #CMDPKT,R4
      JSR     PC,WRTCHR
1$:  RTS
      PC

; WRT SUB-SYS MEM CMD
; MSG BUF ADDR
; BYTE COUNT
; INVERT THE SWITCH
; SET CMDPKT INTO R4
; DO IT
; RETURN

T39DLY: .WORD 0
; DELAY COUNTER FOR TEST
; = <..10> &177770
T39PACKET:
      .WORD 140006
      .WORD T39TAD
      .WORD 0
      .WORD 10.
; COMMAND PACKET FOR TEST
; WRITE SUBSYSTEM MEM. CMD, ACK,CVC=1
; ADDRESS OF CHARACTERISTICS BLOCK
; STARTING VALUE OF BLOCK SIZE

```


KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3702 021460          T39TAD:          ;CHARACTERISTICS DATA BLOCK
3703 021460          T39BS0: .BYTE 0      ;BSELO BYTE
3704 021461          T39BS1: .BYTE 0      ;BSEL1 BYTE
3705 021462 000000   T39BS2: .WORD 0      ;BSEL1 WORD
3706 021464 000000   T39BFR: .BLKW 150.    ;DATA
3707 021466          ;MESSAGE BUFFER
3708
3709
3711          022150
3713 022150          T39PK2: .=<.+10>&177770 ;COMMAND PACKET FOR TEST
3714 022150 140004   .WORD 140004    ;WRITE CHARA. MEM. CMND., ACK,CVC=1
3715 022152 022160   .WORD T39DTA    ;ADDRESS OF SELECT DATA BLOCK
3716 022154 000000   .WORD 0
3717 022156 000012   .WORD 10.      ;STARTING VALUE OF BLOCK SIZE
3718
3719
3720 022160          T39DTA:          ;SELECT DATA BLOCK
3721 022160 021466   .WORD T39BFR    ;ADDRESS OF MESSAGE BUFFER
3722 022162 000000   .WORD 0
3723 022164 000400   .WORD 256.     ;LENGTH OF MESSAGE BUFFER
3724 022166 000000   T39EAI: .WORD 0 ;EAI BIT WORD
3725 022170 000000   T39DSW: .WORD 0 ;DRIVE SELECT WORD ETC
3727          022200   .=<.+10>&177770
3729 022200 140012   T39PK3: .WORD 140012 ;MESSAGE BUFFER RELEASE COMMAND
3730 022202 000000   .WORD 0        ;NOT USED
3731
3732          ;WRITE TAPE PACKET
3733
3735          022210
3737 022210 140005   T39PK4: .WORD 140005 ;WRITE, ACK, CVC=1 COMMAND
3738 022212 000000   T39WR: .WORD 0    ;ADDRESS OF WRITE BUFFER
3739 022214 000000   .WORD 0        ;MORE ADDRESS OF WRITE BUFFER
3740 022216 000400   T39SIZ: .WORD 256. ;SIZE OF RECORD
3741
3742          ; COMMAND PACKET.
3743
3744          022220   . = <.+3>&177774 ;MUST BE ON MOD 4 BOUNDRY.
3745
3746 022220 000000   CMDPKT:: 0      ;1ST WORD IS TS05 COMMAND.
3747 022222 000000   0          ;2ND WORD IS THE BUFFER LOW ADDRESS.
3748 022224 000000   0          ;3RD WORD IS THE BUFFER HIGH ADDRESS.
3749 022226 000000   0          ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
3750
3751          ; WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
3752
3753 022230 000000   WSMBK:: 0      ;1ST WORD:: SEL 0
3754 022232 000000   0          ;2ND WORD:: SEL 2
3755 022234 000000   0          ;3RD WORD:: SEL 4
3756          .EVEN
3757
3758          ;*
3759          ; SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
3760          ;
3761          ; INPUTS:
3762          ; OUTPUTS:
3763          ; The NXMFLG is set if we can test.
3764          ; The NXML0 and NXMH1 addresses are setup.

```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3765      ;-
3766
3767 022236      MEMCK::
3768
3769 022236      SAVREG      ;SAVE THE REGISTERS
3770 022242 005037 003130  CLR      NXMFLG      ;CLEAR THE FLAG
3771 022246 005037 003132  CLR      NXML0      ;CLEAR THE TEST ADDRESS LO
3772 022252 005037 003134  CLR      NXMHI      ;CLEAR THE TEST ADDRESS HI
3773 022256 005737 003140  TST      T23B      ;IS IT A 11/23B?
3774 022262 001407          BEQ      1$          ;NO
3775 022264 023727 002120 007777  CMP      L$HIME,#7777 ; GREATER THAN 128K
3776 022272 103406          BLO      2$          ; NO
3777 022274 004737 022412  JSR      PC,NXMTST  ;SETUP THE ADDRESS
3778 022300 000427          BR       13$         ;SET THE FLAG AND EXIT
3779 022302 005737 003136 1$:      TST      T23A      ;IS IT A 11/23A?
3780 022306 001413          BEQ      4$          ;NO
3781 022310 023727 002120 005777 2$:      CMP      L$HIME,#5777 ;GREATER THAN 96K
3782 022316 101023          BHI      14$         ;YES, 23A/23B WITH 128K MEMORY
3783 022320 023727 002120 003777  CMP      L$HIME,#3777 ;GREATER THAN 64K BUT LESS THAN 92K?
3784 022326 103403          BLO      4$          ;NO, CHECK 24K
3785 022330 004737 022412  JSR      PC,NXMTST  ;SETUP THE ADDRESS
3786 022334 000411          BR       13$         ;SET THE FLAG AND EXIT
3787 022336 023727 002120 001577 4$:      CMP      L$HIME,#1577 ;GREATER THAN 24K BUT LESS THAN 64K?
3788 022344 103410          BLO      14$         ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
3789 022346 004737 022412  JSR      PC,NXMTST  ;SETUP THE ADDRESS
3790 022352 062737 000077 003134  ADD      #77,NXMHI  ;FOOL THE 11/02 & 11/03
3791 022360 005237 003130 13$:     INC      NXMFLG      ;SET THE FLAG
3792 022364 000411          BR       15$         ;EXIT
3793 022366 000410          BR       14$         ;NOP FOR PRINTOUT
3794 022370          PRINTF   #NOMEM      ;TELL THEM & EXIT ***NO PRINT*****
      022370 012746 005460  MOV      #NOMEM,-(SP)
      022374 012746 000001  MOV      #1,-(SP)
      022400 010600          MOV      SP,RO
      022402 104417          TRAP     C$PNTF
      022404 062706 000004  ADD      #4,SP
3795 022410 000207          15$:     RTS      PC          ;RETURN
3796
3797      ;*
3798      ; SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
3799      ;
3800      ; OUTPUTS: NXML0, NXMHI      ; SETUP WITH NXM ADDRESS
3801      ;
3802      ;-
3803
3804 022412 013701 002120  NXMTST: MOV      L$HIME,R1      ;GET TOP OF MEMORY
3805 022416 062701 000200  ADD      #200,R1      ;MAKE IT I/O BLOCK OR OTHER NXM
3806 022422 042701 000177  BIC      #177,R1
3807 022426 010102          MOV      R1,R2
3808          000006          ;RESAVE RESULTS
3809          .REPT      6
3810          ASL      R1          ;PUT IN PLACE FOR XFER
3811          .ENDR
3811 022444 010137 003132  MOV      R1,NXML0      ;SAVE TEST ADDRESS LOW
3812          000012          .REPT      10
3813          ASR      R2          ;PUT IN PLACE FOR XFER
3814          .ENDR
3815 022474 042702 177700  BIC      #177700,R2    ;DON'T WANT ILA!
3816 022500 010237 003134  MOV      R2,NXMHI      ;SAVE TEST ADDRESS HIGH

```

G8

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```
3817 022504 000207                    RTS    PC                    ;RETURN
3818
3819 022506                    ENDMOD
3828                    .TITLE    TSV4 - MISCELLANEOUS SECTIONS
3829
3830 022506                    BGNMOD    TSV4
      022506                    TSV4::
3831
3837
3838
3839
3840                    .SBTTL    PROTECTION TABLE
3841 022506                    BGNPROT
      022506                    L$PROT::
3842 022506                    .WORD    -1, -1, -1, -1                    ;NO DEVICE PROTECTION REQUIRED.
3843 022516                    ENDPROT
      177777 177777 177777
```

INITIALIZE SECTION

```

3845          .SBTTL INITIALIZE SECTION
3846
3847          ;**
3848          ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3849          ;AT THE BEGINNING OF EACH PASS.
3850
3851          ;
3852          ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
3853          ;IF "CONTINUE", NOTHING IS REQUIRED.
3854          ;
3855          ;--
3856          ;*
3857          ;INSERT TEMPORARY JUMP TO ODT
3858          ;-
          BGNINIT
3859 022516    022516
          L$INIT::
3860 022522    005037    002216    002170
          40$: CLR      EXTFEA
          CLR      NXMFLG
          MOV      #EPRT1,EPRTSW      ;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
          READDEF #EF.CONTINUE
          MOV      #EF.CONTINUE,RO
          TRAP     C$REFG
3861 022526    012737    006360    002170
          BNCOMplete 1$
          BCC      1$
          CMP      UNITN,L$UNIT        ;UNIT IN RANGE?
          BHIS     4$                  ;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$EXIT
          .WORD    L10030-
3862 022534    005037    003146
          1$: READDEF #EF.NEW
          MOV      #EF.NEW,RO
          TRAP     C$REFG
          BNCOMplete NXTU              ;TAKE NEXT UNIT IF NOT NEW PASS.
          BCC      NXTU
          READDEF #EF.START
          MOV      #EF.START,RO
          TRAP     C$REFG
3863 022540    005037    003126
          BCOMplete 2$
          BCS      2$
          READDEF #EF.RESTART
          MOV      #EF.RESTART,RO
          TRAP     C$REFG
3864 022544    005037    002274
          BNCOMplete 31$
          BCC      31$
          BRESET
          TRAP     C$RESET
          2$:
3865 022550    012700    000036
          022554    104447
3866 022556
          022556    103023
3867 022560    023737    002172    002012
          022566    103070
3868 022566
          022570    005737    003104
3869 022570
          022574    100472
3870 022574
          022576    013701    002172
3871 022576
          022602    006301
3872 022602
          022604    005761    003170
3873 022604
          022610    001516
3874 022610
          022612    032761    040000    003170
3875 022612
          022620    001060
3876 022620
          022622
          022622    104432
          022624    000416
3877 022622
          022626    012700    000035
          022632    104447
3878 022626
          022634    103052
3879 022634
          022636    012700    000040
          022642    104447
3880 022636
          022644    103404
3881 022644
          022646
          022646    012700    000037
          022652    104447
3882 022646
          022654    103031
3883 022654
          022656
3884 022656
          022656    104433
3885 022656

```

INITIALIZE SECTION

```

3886 022660 005037 002204      CLR      TSTCNT      ;NUMBER OF TESTS RUN IN PASS
3887 022664 005037 002212      CLR      FATFLG     ;CLEAR FATAL ERROR COUNT
3888 022670 005037 003136      CLR      T23A      ;CLEAR 11/23A FLAG
3889 022674 005037 003140      CLR      T23B      ;CLEAR 11/23B FLAG
3890      :      MOV      #340,-(SP)
3891      :      MOV      #20,-(SP)      ;RETURN TO DEBUGGER
3892      :      JMP      0.ODT      ;:ENTER THE DEBUGGER
3893 022700 005037 003372      CLR      SKIPT      ;CLEAR THE SUBTEST "SKIPPER"
3894 022704      :      :
3895 022704 012737 177777 002174 20$:      MOV      #-1,QVP      ;...QUICK VERIFY...
3896 022712 004737 020750      JSR      PC,ENVIRN   ;SET ENVIRONMENT.
3897 022716 004737 021130      JSR      PC,KTINIT  ;INITIALIZE KT MEMORY MANAGEMENT
3898 022722 012700 003170      MOV      #ERTABL,RO
3899 022726 005020 30$:      CLR      (RO)+      ;CLEAR THE ERROR TABLE
3900 022730 020027 003370      CMP      RO,#ERTABE
3901 022734 103774      BLO     30$
3902 022736 000404      BR      4$
3903 022740 005037 002174      CLR      QVP
3904 022744 000137 023014      JMP     PASRPT      ;GO REPORT THE STATUS
3905
3906 022750      4$:
3907 022750 012737 177777 002172 NEWPAS: MOV     #-1,UNITN      ;INIT UNIT NUMBER...
3908 022756 005037 002210      CLR     DEVCNT     ;CLEAR COUNT OF DEVICES RUNNING
3909 022762      NXTU:
3910 022764 104422      BREAK  C$BRK
3911 022770 005237 002172 002012      TRAP   UNITN
3912 022776 023737 002172      INC    UNITN,L$UNIT
3913 023000 103423      CMP    UNITN,L$UNIT
3914 023006 012737 177777 003104      BLO   SETU
3915 023010 000401      MOV    #-1,DUFLG
3916 023012 104444      BR    11$
3917 023014 000240      DOCLN C$DCLN      ;ABORT, NO MORE UNITS.
3918 023014 023727 002012 000001 PASRPT: CMP    L$UNIT,#1      ;HOW MANY UNITS SELECTED?
3919 023022 101752      BLOS  NEWPAS      ;BR IF ONLY 1
3920 023024 005737 002210      TST   DEVCNT      ;ARE ANY STILL RUNNING?
3921 023030 001747      BEQ   NEWPAS      ;BR IF NO
3922 023032 104421      RFLAGS RO
3923 023034 032700 000100      TRAP  C$RFLA
3924 023040 001343      BIT   #ISR,RO
3925      BNE  NEWPAS      ;SHOULD WE PRINT STATISTICS
3926 023042      DORPT
3927 023042 104424      TRAP  C$DRPT      ;BR IF NO
3928 023044 000741      BR    NEWPAS
3929      10$:
3930 023046      SETU:
3931 023046 013700 002172      GPHARD UNITN,RO      ;GET UNIT N P-TABLE POINTER.
3932 023052 104442      MOV   UNITN,RO
3933 023054 103342      TRAP  C$GPHRD
3934 023056 005037 003104      BNCOMPLETE NXTU      ;BR IF UNIT NOT AVAILABLE.
3935 023062 005237 002210      BCC   NXTU
3936 023066 012001 002176      CLR   DUFLG      ;CLEAR "DROPPED" FLAG.
3937 023070 010137 002176      INC   DEVCNT
3938      MOV  (RO)+,R1      ;GET 1ST REGISTER ADDRESS.
3939      MOV  R1,CSRADDR   ;ADDRESS OF REGISTERS OF UNIT UNDER TEST

```

INITIALIZE SECTION

```

3936
3937 023074 012001      MOV      (R0),R1      ;GET VECTOR ADDRESS.
3938                  ;MOV      (R0),R2      ;GET INTERRUPT PRIORITY
3939                  ;MOV      R2,IPRI    ;SET INTERRUPT PRIORITY.
3940 023076 010137 002200 MOV      R1,IVEC     ;SET INTERRUPT VECTOR POINTER...
3941 023102 012721 016336 MOV      #INTR,(R1)+ ;...VECTOR...
3942 023106 013721 002202 MOV      IPRI,(R1)+  ;...AND PRIORITY.
3943
3944 023112              1$:
3945                  ;      TST      QVP          ;1ST PASS ??
3946                  ;      BEQ      5$          ;NO, SKIP THE PASS 1 STUFF.
3947
3948
3949                  ;
3950                  ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
3951                  ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
3952                  ;
3952 023112 013701 002172      MOV      UNITN,R1
3953 023116 006301              ASL      R1
3954 023120 052761 100000 003170 BIS      #BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
3955 023126 005037 005772      CLR      EXTA        ;CLEAR ERROR EXTENSION FLAG.
3956 023132 023727 002012 000001 CMP      L$UNIT,#1    ;ARE WE TESTING MULTIPLE UNITS?
3957 023140 101416              BLOS    10$          ;BR IF NO.
3958 023142              RFLAGS    RO        ;YES -- GET OPERATOR FLAGS.
3959 023144 032700 001000      TRAP    C$RFLA
3960 023150 001412              BIT      #PNT,RO      ;SHOULD WE PRINT UNIT #?
3961 023152              BEQ      10$          ;BR IF NOT.
3962 023152 013746 002172      PRINTF #PUNIT,UNITN ;PRINT THE UNIT #
3963 023156 012746 023244      MOV      UNITN,-(SP)
3964 023162 012746 000002      MOV      #PUNIT,-(SP)
3965 023166 010600              MOV      #2,-(SP)
3966 023170 104417              MOV      SP,RO
3967 023172 062706 000006      TRAP    C$PNTF
3968 023176              ADD      #6,SP
3969 023176 005037 003106      10$: CLR      NODEV
3970 023202 013701 002176      MOV      CSRADDR,R1 ;ADDRESS OF FIRST REGISTER
3971 023206 010102              MOV      R1,R2      ;START OF REGISTERS
3972 023210 062702 000002      ADD      #TSSR,R2  ;ADDRESS OF TSSR REGISTER
3973 023214 004737 016516      JSR      PC,XNXM    ;TEST BOTH CONTROLLER REGISTERS...
3974 023220 103005              BCC     2$          ;...AND BR IF ALL OK.
3975 023222 010137 003106      MOV      R1,NODEV  ;FLAG DEVICE AS NON-EXISTENT
3976 023226 012737 177777 003104 MOV      #-1,DUFLG ;DROP THIS UNIT.
3977 023234              2$:
3978                  ;
3979                  ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
3980                  ;
3981 023234 012700 000000      5$: SETPRI #PRI00      ;ENABLE INTERRUPTS.
3982 023240 104441              MOV      #PRI00,RO
3983 023242              TRAP    C$SPRI
3984 023242              ENDINIT
3985 023242 104411      L10030: TRAP    C$INIT
3986 023244 045 116 045 PUNIT: .ASCIZ /#N#N#A***** TESTING UNIT #D2#A *****/
3987                  .EVEN

```

ADD AND DROP UNITS SECTIONS

.SBTTL ADD AND DROP UNITS SECTIONS

```

3981
3982
3983
3984
3985
3986
3987
3988 023312
023312
3989 023312 010001
3990 023314 006301
3991 023316 052761 100000 003170
3992 023324 042761 040000 003170
3993 023332
023332 010046
023334 012746 023360
023340 012746 000002
023344 010600
023346 104417
023350 062706 000006
3994 023354
023354 000167
023356 000026
3995 023360 045 116 045 1$:
3996
3997
3998 023406
023406
023406 104452
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010 023410
023410
4011 023410 012737 177777 003104
4012 023416 010001
4013 023420 006301
4014 023422 052761 140000 003170
4015 023430 000240 000240 000240
4016 023436
023436 010046
023440 012746 023464
023444 012746 000002
023450 010600
023452 104417
023454 062706 000006
4017 023460
023460 000167
023462 000030

```

```

; **
; 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.
; --
      BGNU
L$AU::
      MOV      RO,R1          ; GET UNIT TO BE ADDED (RO)
      ASL      R1             ; MAKE IT A WORD INDEX
      BIS      #100000,ERTABL(R1) ; SET THE "ACTIVE" BIT
      BIC      #40000,ERTABL(R1) ; CLEAR THE "DROPPED" BIT
      PRINTF   #1$,RO
      MOV      RO,-(SP)
      MOV      #1$,-(SP)
      MOV      #2,-(SP)
      MOV      SP,RO
      TRAP     C$PNTF
      ADD      #6,SP
      EXIT     AU
      .WORD    J$JMP
      .WORD    L10031-2-.
      .ASCIZ   /#N#A UNIT #D#A 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, RO CONTAINS THE UNIT TO BE DROPPED.
      BGNU
L$DU::
      MOV      #-1,DUFLG
      MOV      RO,R1
      ASL      R1
      BIS      #140000,ERTABL(R1) ; SAY DROPPED
      240,240,240 ; ??????????
      PRINTF   #1$,RO
      MOV      RO,-(SP)
      MOV      #1$,-(SP)
      MOV      #2,-(SP)
      MOV      SP,RO
      TRAP     C$PNTF
      ADD      #6,SP
      EXIT     DU
      .WORD    J$JMP
      .WORD    L10032-2-.

```

L8

ADD AND DROP UNITS SECTIONS

```

4018 023464 045 116 045 1$: .ASCIZ /%N%A UNIT %D%A DROPPED/
4019 .EVEN
4020 023514 ENDDU
      023514 L10032: TRAP C$DU
      023514 104453
4021
4022 ;**
4023 ; AUTO-DROP CODE SECTION.
4024 ;--
      023516 BGNAUTO
      023516 L$AUTO::
4025 023516 013705 002176 MOV CSRADDR,R5 ;POINT TO DEVICE REGISTER
4026 023522 012703 000550 MOV #360.,R3 ;ENOUGH TIME FOR 2400' REEL TO REWIND
4027 023526 004737 016370 10$: JSR PC,WAITF ;WAIT FOR SSR TO SET
4028 023532 103420 BCS 20$ ;LEAVE WHEN SSR IS SET
4029 023534 DELAY 250. ;WAIT FOR .25 SECONDS
      023534 012727 000372 MOV #250.,(PC)+
      023540 000000 .WORD 0
      023542 013727 002116 MOV L$DLY,(PC)+
      023546 000000 .WORD 0
      023550 005367 177772 DEC -6(PC)
      023554 001375 BNE .-4
      023556 005367 177756 DEC -22(PC)
      023562 001367 BNE .-20
4030 023564 005303 DEC R3 ;BUMP COUNTER DOWN
4031 023566 001357 BNE 10$ ;KEEP GOING
4032 023570 004737 017322 JSR PC,CKDROP ;TRY AND DROP UNIT
4033 023574 20$: ENDAUTO ; UNUSED.
4034 023574 L10033: TRAP C$AUTO
      023574 104461

```


CLEAN-UP AND REPORT CODING SECTIONS

.SBTTL CLEAN-UP AND REPORT CODING SECTIONS

```

4036
4037
4038
4039
4040
4041
4042
4043 023576
      023576
4044 023576 013705 002176
4045 023602 005737 003104
4046 023606 100405
4047
4048
4049 023610 012765 000000 000002
4050 023616 004737 016370
4051 023622
4052 023622
      023622
      023622 104412
4053
4054
4055
4056
4057 023624
      023624
4058 023624
      023624 012746 024066
      023630 012746 000001
      023634 010600
      023636 104416
      023640 062706 000004
4059 023644 010246
4060 023646 010346
4061 023650 010446
4062 023652 012704 003170
4063 023656 005003
4064 023660 011402
4065 023662 001467
4066 023664 100066
4067 023666 032702 040000
4068 023672 001015
4069 023674 042702 170000
4070 023700
      023700 010246
      023702 010346
      023704 012746 024123
      023710 012746 000003
      023714 010600
      023716 104416
      023720 062706 000010
4071 023724 000446
4072 023726 020227 160000
4073 023732 001012
4074 023734
      023734 010346
      023736 012746 024173

```

```

      BGNCLN
L$CLEAN::
      MOV      CSRADDR,R5          ;POINT TO DEVICE REGISTER
      TST      DUFLG              ;"DROPPED" FLAG IS SET ON...
      BMI      1$                 ;...AND GROSS CONTROLLER FAULT...
                                   ;...DON'T TRY TO XCT CLEANUP CODE.
      MOV      #0,TSSR(R5)        ;DO SOFT INIT
      JSR      PC,WAITF
1$:
2$:
L10034:
      ENDCLN
      TRAP     C$CLEAN

```

```

      BGNRPT
L$RPT::
      PRINTS  #DEVSUM
      MOV      #DEVSUM,-(SP)
      MOV      #1,-(SP)
      MOV      SP,R0
      TRAP     C$PNTS
      ADD      #4,SP
      MOV      R2,-(SP)
      MOV      R3,-(SP)
      MOV      R4,-(SP)
      MOV      #ERTABL,R4        ; GET START OF ERROR TABLE.
      CLR      R3                ; CLEAR UNIT NUMBER
1$:
      MOV      (R4),R2           ; GET ERROR TABLE ENTRY & TEST IT.
      BEQ      4$                ; ZERO IF UNIT NOT RUN
      BPL      4$
      BIT      #BIT14,R2        ; WAS UNIT DROPPED?
      BNE      2$                ; BR IF YES
      BIC      #C7777,R2        ; GET ERROR COUNT FIELD
      PRINTS  #DEVONL,R3,R2     ; PRINT
      MOV      R2,-(SP)
      MOV      R3,-(SP)
      MOV      #DEVONL,-(SP)
      MOV      #3,-(SP)
      MOV      SP,R0
      TRAP     C$PNTS
      ADD      #10,SP
      BR       4$
2$:
      CMP      R2,#160000        ; WAS UNIT NON-EXISTENT?
      BNE      3$                ; BR IF NO
      PRINTS  #DEVNXR,R3
      MOV      R3,-(SP)
      MOV      #DEVNXR,-(SP)

```

CLEAN-UP AND REPORT CODING SECTIONS

```

023742 012746 000002      MOV    #2,-(SP)
023746 010600      MOV    SP,R0
023750 104416      TRAP  C#PNTS
023752 062706 000006      ADD    #6,SP
4075 023756 000431      BR     4#
4076 023760 020227 160001      3# :  CMP    R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
4077 023764 001012      BNE   30#                ; BR IF NO.
4078 023766      PRINTS #DEVNRD,R3
023766 010346      MOV    R3,-(SP)
023770 012746 024255      MOV    #DEVNRD,-(SP)
023774 012746 000002      MOV    #2,-(SP)
024000 010600      MOV    SP,R0
024002 104416      TRAP  C#PNTS
024004 062706 000006      ADD    #6,SP
4079 024010 000414      BR     4#
4080 024012 042702 170000      30# : BIC    #1C7777,R2
4081 024016      PRINTS #DEVDR0,R3,R2
024016 010246      MOV    R2,-(SP)
024020 010346      MOV    R3,-(SP)
024022 012746 024336      MOV    #DEVDR0,-(SP)
024026 012746 000003      MOV    #3,-(SP)
024032 010600      MOV    SP,R0
024034 104416      TRAP  C#PNTS
024036 062706 000010      ADD    #10,SP
4082 024042 062704 000002      4# :  ADD    #2,R4
4083 024046 005203      INC    R3
4084 024050 020427 003370      CMP    R4,#ERTABE
4085 024054 103701      BLO   1#
4086 024056 012604      MOV    (SP)+,R4
4087 024060 012603      MOV    (SP)+,R3
4088 024062 012602      MOV    (SP)+,R2
4089 024064      ENDRPT ; UNUSED.
024064      L10035:
024064 104425      TRAP  C#RPT
4090
4091 024066      045      116      045  DEVSUM: .ASCIZ /#N#ADEVICE STATUS SUMMARY:#N/
4092 024123      045      101      040  DEVONL: .ASCIZ /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
4093 024173      045      101      040  DEVNXR: .ASCIZ /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
4094 024255      045      101      040  DEVNRD: .ASCIZ /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
4095 024336      045      101      040  DEVDR0: .ASCIZ /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
4096      .EVEN
4097
4098 024406      ENDMOD
4099

```

CLEAN-UP AND REPORT CODING SECTIONS

```

4103          .TITLE  TSV7 - HARDWARE TESTS 1-8
4104
4111
4112 024406    BGNMOD  TSV7
         024406    TSV7::
4118
4126          .SBTTL  TEST 1: INITIALIZE #4 TEST
4127
4128          ;+
4129          ;
4130          ;THIS TEST VERIFIES THAT WRITING INTO THE TSSR RETURNS THE
4131          ;CONTROLLER TO ITS INITIALIZED STATE FROM VARIOUS CONDITIONS
4132          ;(I.E. LOOPBACK ENABLED, FORCING WRONG PARITY, INVERTING SENSE OF
4133          ;EXTENDED FEATURES SWITCH, ETC.)
4134          ;
4135          ;-
4136          BGNTST
4137 024406    012737  006360  002170    MOV      #EPRT1,EPRTSW          ;SET UP PRIMARY ERROR MESSAGE
         024406
4138
4139          ;
4140          ;
4141          ;TEST 1
4142          ;
4143          ;
4144          ;-
4145
4150 024414    004737  016324
4151 024420    012700  025344
4152 024424    004737  016630
4153 024430    012737  000005  002206    JSR      PC,DSBINT             ;DISABLE INTERRUPTS
         024436    T21LOOP:      MOV      #TST21ID,R0          ;ASCII MESSAGE TO IDENTIFY TEST
4154 024436    004737  025366    JSR      PC,TSTSETUP          ;DO INITIAL TEST SETUP
4155 024436    004737  025366    MOV      #5,LOOPCNT          ;PERFORM 5 ITERATIONS
4156 024442    004737  025456    JSR      PC,T21REST          ;SET COMMAND PACKET
         024442    JSR      PC,T21RT2          ;SET UP OTHER COMMAND PACKET
4157
4158          ;*****
4159          ;
4160          ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
4161          ;
4162          ;*****
4163
4164 024446    012737  176750  025022    MOV      #65000.,T21DLY      ;SET DELAY ROUTINE
4165 024454    004737  016114    11$:   JSR      PC,SOFINIT          ;DO INITIALIZE ON CONTROLLER
4166 024460    103426
         024462    DELAY      250          ;BR IF INIT WAS OK
         024462    012727  000250          ;DELAY FOR A REWIND TO FINISH
         024466    000000
         024470    013727  002116          MOV      #250,(PC)+
         024474    000000
         024476    005367  177772          .WORD   0
         024502    001375
         024504    005367  177756          MOV      L#DLY,(PC)+
         024510    001367
         024512    005337  025022          .WORD   0
         024516    001356          DEC      -6(PC)
         024520    005237  002212          BNE     -.4
         024524    010001          DEC      -22(PC)
         024512    005337  025022          BNE     .-20
4168 024512    005337  025022    DEC      T21DLY              ;BUMP COUNTER DOWN
4169 024516    001356          BNE     11$                  ;BR, IF MORE TIME TO GO
4170 024520    005237  002212    INC      FATFLG              ;BUMP COUNT
4174 024524    010001          MOV      R0,R1                ;CONTENTS OF TSSR REGISTER

```

TEST 1: INITIALIZE #4 TEST

```

4175 024526          ERRDF  ERRNO,SFIERR,SFIMSG      ;FATAL ERROR TSSR WAS NOT OK
      024526 104455          TRAP  C$ERDF
      024530 000145          .WORD 101
      024532 003652          .WORD SFIERR
      024534 012154          .WORD SFIMSG
4176          20$:          MOV  #T21PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
4177 024536 012704 025000
4178
4179          ;*****
4180          ;
4181          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
4182          ;
4183          ;*****
4184
4185 024542 013737 002172 025020      MOV  UNITN,T21DSW          ;SET UP DRIVE NUMBER
4186 024550 004737 010752          JSR  PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
4187 024554 103407          BCS  23$              ;BR, IF COMMAND ISSUED OK
4188 024556 005237 002212          INC  FATFLG          ;BUMP COUNT
4192 024562 010001          MOV  R0,R1          ;SAVE CONTENTS OF TSSR
4193 024564          ERRHRD ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTICS FAILED
      024564 104456          TRAP  C$ERHRD
      024566 000146          .WORD 102
      024570 005056          .WORD WRTMSG
      024572 012154          .WORD SFIMSG
4194 024574          23$:          CKLOOP          TRAP  C$CLP1
      024574 104406
4195 024576 112737 000200 025120      MOV  #200,T21BS0          ;WRITE MISCELLANEOUS CONT/READ STATUS
4196 024604 112737 000010 025121      MOV  #10,T21BS1          ;FUNCTION SELECTION BIT
4197 024612          25$:          MOV  #T21PK2,R4          ;WRITE SUBSYS MEM PACKET
4198 024612 012704 025110          MOV  R4,TSDB(R5)          ;ISSUE COMMAND
4199 024616 010465 000000          JSR  PC,CHKTSSR          ;WAIT FOR SSR
4200 024622 004737 016456          BCS  30$              ;BR, IF NO ERROR
4201 024626 103407          MOV  R0,R1          ;ERROR, SAVE TSSR
4202 024630 010001          INC  FATFLG          ;BUMP COUNT
4203 024632 005237 002212          ERRHRD ERRNO,T21SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
4207 024636          TRAP  C$ERHRD
      024636 104456          .WORD 103
      024640 000147          .WORD T21SSR
      024642 025126          .WORD PKTSSR
      024644 012166
4208 024646          30$:          CKLOOP          ;SCOPE LOOP          TRAP  C$CLP1
      024646 104406
4209 024650 012765 000000 000002      MOV  #0,TSSR(R5)          ;ISSUE A SOFT INITIALIZE
4210 024656 004737 016370          JSR  PC,WAITF          ;WAIT FOR JUST THE SSR BIT TO SET
4211 024662 016501 000002          MOV  TSSR(R5),R1          ;READ THE TSSR BACK
4212 024666 010102          MOV  R1,R2          ;WORK REGISTER
4213 024670 042702 176377          BIC  #!C<HIADDR>,R2          ;CLEAR OUT OTHER BITS
4214 024674 052702 002200          BIS  #SSR!NBA,R2          ;SOME OF THE BITS THAT SHOULD BE SET
4215 024700 032701 000100          BIT  #OFL,R1          ;IS OFF LINE BIT SET
4216 024704 001012          BNE  38$              ;BR, IF DRIVE IS OFF LINE
4217 024706 020102          35$:          CMP  R1,R2          ;EXPECTED (R2) = RECEIVED (R1)
4218 024710 001406          BEQ  37$              ;BR, IF THEY ARE EQUAL (OK)
4219 024712 005237 002212          INC  FATFLG          ;BUMP COUNT
4223 024716          ERRHRD ERRNO,T21AM3,EXPREC ;"ERROR TRYING TO INIT AFTER WRITE MISC.
      024716 104456          TRAP  C$ERHRD
      024720 000150          .WORD 104
      024722 025223          .WORD T21AM3

```

TEST 1: INITIALIZE #4 TEST

```

4224 024724 015614
4224 024726 104406
4225 024730 000406
4226 024732
4230 024732 104455
024734 000151
024736 025323
024740 015614
4231 024742 004737 017322
4232 024746 000241
4233 024750 106037 025121
4234 024754 001316
4235 024756
024756 104406
4236 024760 004737 016576
4237 024764 103002
4238 024766 000137 024436
4239 024772
024772 104432
024774 000524
4240
4241
4242
4243
4245 024776
4247 025000
4248 025000 100004
4249 025002 025010
4250 025004 000000
4251 025006 000012
4252 025010
4253 025010 025024
4254 025012 000000
4255 025014 000024
4256 025016 000000
4257 025020 000000
4258 025022 000000
4259 025024
4260
4261
4262
4264 025106
4266 025110
4267 025110 100206
4268 025112 025120
4269 025114 000000
4270 025116 000006
4271
4272
4273 025120
4274 025120 000
4275 025121 000
4276 025122 000000
4277 025124 000000
4278

37$: CKLOOP ;LOOP IF SELECTED .WORD EXPREC
;TRAP C$CLP1
38$: BR 40$ ;SKIP OVER OFF-LINE STUFF
ERRDF ERRNO,T21OFL,EXPREC ;DRIVE IS OFF LINE
;TRAP C$ERDF
;WORD 105
;WORD T21OFL
;WORD EXPREC
40$: JSR PC,CKDROP ;TRY AND DROP UNIT
CLC ;DON'T LET CARRY SNEAK IN
RORB T21BS1 ;TRY NEXT "LOWEST" BIT POSITION
BNE 25$ ;LOOP UNTIL ALL EIGHT BITS TESTED
50$: CKLOOP ;SCOPE LOOP
;TRAP C$CLP1
40$: JSR PC,TSTLOOP ;DO WE NEED TO ITERATE TEST
BCC 63$ ;BR, IF NO LOOP REQUIRED
JMP T21LOOP ;EXECUTE AGAIN
63$: EXIT TST ;ALL DONE THIS TEST
;TRAP C$EXIT
;WORD L10036-.

;+
;LOCAL STORAGE FOR THIS TEST
;-
T21PACKET: .BLKB 10-<.-TSV2&7>
;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH, ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;WORD 100004
;WORD T21DATA
;WORD 0
;WORD 10.
;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
T21DATA: .WORD T21BFR
;WORD 0
;WORD 20.
;LENGTH OF MESSAGE BUFFER
;WORD 0
T21DSW: .WORD 0 ;DRIVE SELECT WORD
T21DLY: .WORD 0 ;DELAY COUNTER
T21BFR: .BLKW 25. ;MESSAGE BUFFER

;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
T21PK2: .BLKB 10-<.-TSV2&7>
;WRITE SUB SYS MEM COMMAND, IE AND ACK
;ADDRESS OF SELECT BLOCK DATA
;WORD 100206
;WORD T21BF2
;WORD 0
;WORD 6.
;SIZE OF DATA PACKET
;EVEN
T21BF2:
T21BS0: .BYTE 0 ;BSELO AREA --- "COMMAND" BYTE
T21BS1: .BYTE 0 ;BSEL1 AREA
T21S2: .WORD 0 ;SEL 2 AREA
T21S3: .WORD 0 ;DATA AREA

```

TEST 1: INITIALIZE #4 TEST

```

4279
4280
4281      ;*
4282      ;LOCAL TEXT MESSAGES FOR TEST
4283      ;-
4284 025126      127      122      111  T21SSR: .ASCIZ 'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
4285 025223      124      123      123  T21AM3: .ASCIZ 'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
4286 025323      104      162      151  T21OFL: .ASCIZ 'Drive is OFFLINE'
4287 025344      111      156      151  T21ID:  .ASCIZ 'Initialization #4'
4288
4289      ;*
4290      ;
4291      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4292      ;WRITE SUBSYSTEM MEMORY COMMAND
4293      ;
4294      ;-
4295
4296 025366      T21REST:
4297 025366      SAVREG
4298 025372      012701  025000      MOV      #T21PACKET,R1      ;SAVE THE REGISTERS
4299 025376      012721  100004      MOV      #100004,(R1)+      ;START OF THE PACKET
4300 025402      012721  025010      MOV      #T21DATA,(R1)+    ;WRITE SUBSYSTEM MEM. WITH ACK,
4301 025406      005021      CLR      (R1)+              ;ADDRESS OF CHARAISTICS DATA BLOCK
4302 025410      012721  000010      MOV      #8,(R1)+          ;EXTENDED ADDRESS
4303 025414      012721  025024      MOV      #T21BFR,(R1)+     ;SIZE OF DATA BLOCK IN BYTES
4304 025420      005021      CLR      (R1)+              ;ADDRESS OF MESSAGE BUFFER
4305 025422      012721  000024      MOV      #20,(R1)+         ;LENGTH OF MESSAGE BUFFER
4306 025426      005021      CLR      (R1)+
4307 025430      005011      CLR      (R1)
4308 025432      012702  000020      MOV      #20,R2            ;NUMBER OF LOCATIONS TO BE CLEARED
4309 025436      012762  177777  025024  64$:  MOV      #177777,T21BFR(R2) ;ALL ONES TO MESSAGE BUFFER
4310 025444      005742      TST      -(R2)              ;NEXT LOCATION
4311 025446      020227  000000      CMP      R2,#0             ;CHECK R2 FOR ZERO
4312 025452      001371      BNE      64$               ;BR, IF NOT AT ZERO YET
4313 025454      000207      RTS      PC                ;RETURN
4314
4315
4316 025456      T21RT2:
4317 025456      SAVREG
4318 025462      012701  025110      MOV      #T21PK2,R1        ;SAVE THE REGISTERS
4319 025466      012721  100206      MOV      #100206,(R1)+     ;START OF THE PACKET
4320 025472      012721  025120      MOV      #T21BF2,(R1)+    ;WRITE SUBSYSTEM MEM. WITH ACK, IE
4321 025476      005021      CLR      (R1)+              ;ADDRESS OF DATA BLOCK
4322 025500      012721  000006      MOV      #6,(R1)+         ;EXTENDED ADDRESS
4323 025504      005021      CLR      (R1)+              ;SIZE OF DATA BLOCK IN BYTES
4324 025506      012701  025120      MOV      #T21BF2,R1        ;ADDRESS OF DATA FOR WRT SUB SYS MEM
4325 025512      005021      CLR      (R1)+
4326 025514      005011      CLR      (R1)
4327 025516      000207      RTS      PC                ;RETURN
4328 025520      ENDTST
4329      025520      104401      L10036: TRAP      C4ETST
4330
4331      ;
4332      ;
4333      ;SBTTL TEST 2: OFF-LINE AND REJECT REWIND
4334
4335      ;*
4336      ;
4337      ;THIS TEST VERIFIES BASIC TAPE-MOTION COMMAND DECODING AND BASIC

```


TEST 2: OFF-LINE AND REJECT REWIND

```

4391
4392
4393
4394 025614 004737 010752
4395 025620 103407
4396 025622 005237 002212
4400 025626 010001
4401 025630
      025630 104456
      025632 000312
      025634 005056
      025636 012154
4402 025640
      025640 104406
4403 025642 013701 027120
4404 025646 032701 000004
4405 025652 001407
4406 025654 005237 002212
4410 025660
      025660 104455
      025662 000313
      025664 027722
      025666 012154
4411 025670
      025670 104444
4412 025672
      025672 104406
4413 025674 005737 002216
4414 025700 001041
4415 025702 112737 000200 027211
4416 025710 112737 000010 027210
4417 025716 012704 027200
4418 025722 010465 000000
4419 025726 004737 016456
4420 025732 103407
4421 025734 010001
4422 025736 005237 002212
4426 025742
      025742 104456
      025744 000314
      025746 027230
      025750 012166
4427 025752
      025752 104406
4428 025754 012704 027070
4429
4430
4431
4432
4433
4434
4435
4436 025760 004737 010752
4437 025764 103407
4438 025766 005237 002212
4442 025772 010001
4443 025774

```

```

;*****
;
;ISSUE WRITE CHARACTERISTICS
;BR, IF COMMAND ISSUED OK
;BUMP COUNT
;SAVE CONTENTS OF TSSR
;WRITE CHARACTERISTIC FAILED
TRAP C$ERHRD
.WORD 202
.WORD WRTMSG
.WORD SFIMSG
23$: CKLOOP
TRAP C$CLP1
MOV T22BFR+6,R1 ;PICK UP XTSO
BIT #4,R1 ;IS UNIT WRITE-LOCKED?
BEQ 24$ ;NO, PROCEED WITH TESTING
INC FATFLG ;BUMP COUNT
ERRDF ERRNO,T22WLK,SFIMSG ;TAPE IS WRITE LOCKED
TRAP C$ERDF
.WORD 203
.WORD T22WLK
.WORD SFIMSG
DOCLN
TRAP C$DCLN
24$: CKLOOP
TRAP C$CLP1
TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
BNE 50$ ;BR IF SWITCH IS ON
MOVB #200,T22BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
MOVB #10,T22BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
MOV #T22PK2,R4 ;WRITE SUBSYS MEM PACKET
MOV R4,TSD(R5) ;ISSUE COMMAND
JSR PC,CHKTSSR ;WAIT FOR SSR
BCS 30$ ;BR, IF NO ERROR
MOV RO,R1 ;ERROR, SAVE TSSR
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T22SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
TRAP C$ERHRD
.WORD 204
.WORD T22SSR
.WORD PKTSSR
30$: CKLOOP ;LOOP IF SELECTED
TRAP C$CLP1
MOV #T22PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
;*****
;
;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
;
;*****
;
;ISSUE WRITE CHARACTERISTICS
;BR, IF COMMAND ISSUED OK
;BUMP COUNT
;SAVE CONTENTS OF TSSR
;WRITE CHARACTERISTIC FAILED

```


TEST 2: OFF-LINE AND REJECT REWIND

```

026622 104456 TRAP C$ERHRD
026624 000330 .WORD 216
026626 027230 .WORD T22SSR
026630 012166 .WORD PKTSSR
4646 026632 30$: CKLOOP ;LOOP IF SELECTED
026632 104406 TRAP C$CLP1
4647 026634 012704 027070 MOV #T22PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4648
4649 ;*****
4650 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
4651 ;
4652 ;*****
4653
4654
4655 026640 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
4656 026644 103407 BCS 50$ ;BR, IF COMMAND ISSUED OK
4657 026646 005237 002212 INC FATFLG ;BUMP COUNT
4661 026652 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
4662 026654 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
026654 104456 TRAP C$ERHRD
026656 000331 .WORD 217
026660 005056 .WORD WRTMSG
026662 012154 .WORD SFIMSG
4663 026664 50$: CKLOOP ;SCOPE LOOP TRAP C$CLP1
026664 104406
4664 026666 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4665 026672 032701 000100 BIT #OFL,R1 ;CHECK FOR THE OFFLINE BIT SET
4666 026676 001006 BNE 60$ ;BR, IF OFFLINE (GOOD)
4667 026700 005237 002212 INC FATFLG ;BUMP COUNT
4671 026704 ERRDF ERRNO,T22OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
026704 104455 TRAP C$ERDF
026706 000332 .WORD 218
026710 027425 .WORD T22OFL
026712 012154 .WORD SFIMSG
4672 026714 60$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
026714 104406
4673 026716 012737 142010 027200 65$: MOV #142010,T22PK2 ;POSITION COMMAND (REWIND MODE) CVC=1
4674 026724 012704 027200 MOV #T22PK2,R4 ;R4 = POINTER TO PACKET
4675 026730 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
4676 026734 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
4677 026740 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4678 026744 012702 100306 MOV #SSR!OFL!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
4679 026750 020102 CMP R1,R2 ;ARE THEY EQUAL
4680 026752 001406 BEQ 80$ ;BR, IF OK ESP. FUNCTION REJECT
4681 026754 005237 002212 INC FATFLG ;BUMP COUNT
4685 026760 ERRHRD ERRNO,T22RWJ,EXPREC ;TSSR INCORRECT AFTER TAPE MOTION CMD
026760 104456 TRAP C$ERHRD
026762 000333 .WORD 219
026764 027574 .WORD T22RWJ
026766 015614 .WORD EXPREC
4686 026770 80$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
026770 104406
4687 026772 012703 027112 MOV #T22BFR,R3 ;POINTER TO MESSAGE BUFFER
4688 026776 016301 000006 MOV XSTO(R3),R1 ;PICK UP XSTO FROM MESSAGE BUFFER
4689 027002 010102 MOV R1,R2 ;SET UP EXPECTED
4690 027004 042702 000020 BIC #BIT4,R2 ;VCK SHOULD BE CLEAR
4691 027010 020102 CMP R1,R2 ;ARE THEY EQUAL

```


TEST 2: OFF-LINE AND REJECT REWIND

```

4748 ;TAPE MOTION PACKET COMMAND VALUES
4749 027216 100201 T22RD: .WORD 100201 ;READ TAPE FORWARD
4750 027220 100205 T22WRT: .WORD 100205 ;WRITE TAPE FORWARD
4751 027222 100210 T22POS: .WORD 100210 ;POSITION TAPE
4752 027224 100211 T22FOR: .WORD 100211 ;FORMAT TAPE
4753 027226 177777 .WORD 177777 ;END OF DATA
4754
4755
4756 ;*
4757 ;LOCAL TEXT MESSAGES FOR TEST
4758 ;-
4759
4760 027230 127 122 111 T22SSR: .ASCIZ 'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
4761 027325 124 123 123 T22AM3: .ASCIZ 'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
4762 027425 104 162 151 T22OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
4763 027500 124 123 123 T22TM: .ASCIZ 'TSSR Incorrect After Tape Motion Command To Off-Line Device'
4764 027574 124 123 123 T22RWJ: .ASCIZ 'TSSR Not Correct After REWIND With VCK Set'
4765 027647 103 126 103 T22VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
4766 027722 052 052 052 T22WLK: .ASCIZ '*****TAPE IS WRITE-LOCKED AND WILL CAUSE ERRORS*****'
4767 030007 117 146 146 TST22ID: .ASCIZ 'Off-Line And Reject Rewind'
4768 .EVEN
4769 ;*
4770 ;
4771 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4772 ;WRITE SUBSYSTEM MEMORY COMMAND
4773 ;
4774 ;-
4775
4776 030042 T22REST:
4777 030042 SAVREG ;SAVE THE REGISTERS
4778 030046 012701 027070 MOV #T22PACKET,R1 ;START OF THE PACKET
4779 030052 012721 100204 MOV #100204,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK, IE
4780 030056 012721 027100 MOV #T22DATA,(R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
4781 030062 005021 CLR (R1)+ ;EXTENDED ADDRESS
4782 030064 012721 000012 MOV #10,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
4783 030070 012721 027112 MOV #T22BFR,(R1)+ ;ADDRESS OF MESSAGE BUFFER
4784 030074 005021 CLR (R1)+
4785 030076 012721 000024 MOV #20,(R1)+ ;LENGTH OF MESSAGE BUFFER
4786 030102 005021 CLR (R1)+
4787 030104 012711 000007 MOV #7,(R1) ;SELECT DRIVE SEVEN
4788 030110 012702 000020 MOV #20,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
4789 030114 012762 177777 027112 64$: MOV #177777,T22BFR(R2) ;ALL ONES TO MESSAGE BUFFER
4790 030122 005742 TST -(R2) ;BUMP R2 DOWN
4791 030124 020227 000000 CMP R2,#0 ;IS R2 AT ZERO YET
4792 030130 001371 BNE 64$ ;KEEP GOING UNTIL DONE
4793 030132 000207 RTS PC ;RETURN
4794
4795
4796 030134 T22RT2:
4797 030134 SAVREG ;SAVE THE REGISTERS
4798 030140 012701 027200 MOV #T22PK2,R1 ;START OF THE PACKET
4799 030144 012721 100206 MOV #100206,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK, IE
4800 030150 012721 027210 MOV #T22BF2,(R1)+ ;ADDRESS OF DATA BLOCK
4801 030154 005021 CLR (R1)+ ;EXTENDED ADDRESS
4802 030156 012721 000006 MOV #6,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
4803 030162 005021 CLR (R1)+
4804 030164 012701 027210 MOV #T22BF2,R1 ;POINT TO DATA SEL AREA

```


TEST 3: BASIC WRITE DATA

```

4861 030264 004737 016114      JSR    PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
4862 030270 103407              BCS    20$             ;BR IF INIT WAS OK
4863 030272 005237 002212      INC    FATFLG          ;BUMP COUNT
4867 030276 010001              MOV    R0,R1           ;CONTENTS OF TSSR REGISTER
4868 030300              ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                TRAP    C$ERDF
                                .WORD   301
                                .WORD   SFIERR
                                .WORD   SFIMSG
                                030300 104455
                                030302 000455
                                030304 003652
                                030306 012154
4869 030310              20$:
4870 030310 012737 000007 033540  MOV    #7,T23DSW      ;SET DRIVE NUMBER IN PACKET
4871 030316 012704 033520      MOV    #T23PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4872
4873
4874
4875 ;*****
4876 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
4877 ;*****
4878
4879 030322 004737 010752      JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
4880 030326 103407              BCS    23$             ;BR, IF COMMAND ISSUED OK
4881 030330 005237 002212      INC    FATFLG          ;BUMP COUNT
4885 030334 010001              MOV    R0,R1           ;SAVE CONTENTS OF TSSR
4886 030336              ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP    C$ERHRD
                                .WORD   302
                                .WORD   WRTMSG
                                .WORD   SFIMSG
                                030336 104456
                                030340 000456
                                030342 005056
                                030344 012154
4887 030346 005737 002216      23$: TST    EXTFEA        ;CHECK FOR EXTENDED FEATURES SW SWITCH
4888 030352 001044              BNE    50$             ;BR IF SWITCH IS ON
4889
4890 030354 112737 000200 033663  MOVB   #200,T23BS1    ;WRITE MISCELLANEOUS CONT/READ STATUS
4891 030362 112737 000010 033662  MOVB   #10,T23BS0     ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4892 030370 012704 033630      MOV    #T23PK2,R4     ;WRITE SUBSYS MEM PACKET
4893 030374 010465 000000      MOV    R4,TSDB(R5)    ;ISSUE COMMAND
4894 030400 004737 016456      JSR    PC,CHKTSSR     ;WAIT FOR SSR
4895 030404 103407              BCS    30$             ;BR, IF NO ERROR
4896 030406 010001              MOV    R0,R1           ;ERROR, SAVE TSSR
4897 030410 005237 002212      INC    FATFLG          ;BUMP COUNT
4901 030414              ERRHRD ERRNO,T23SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
                                TRAP    C$ERHRD
                                .WORD   303
                                .WORD   T23SSR
                                .WORD   PKTSSR
                                030414 104456
                                030416 000457
                                030420 033704
                                030422 012166
4902 030424              30$: CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
4903 030426 012737 000007 033540  MOV    #7,T23DSW      ;SET DRIVE NUMBER IN PACKET
4904 030434 012704 033520      MOV    #T23PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4905
4906 ;*****
4907 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
4908 ;*****
4909
4910
4911
4912 030440 004737 010752      JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
4913 030444 103407              BCS    50$             ;BR, IF COMMAND ISSUED OK

```


TEST 3: BASIC WRITE DATA

```

5020 030744          ERRHRD  ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTIC FAILED
      030744 104456          TRAP                  C$ERHRD
      030746 000464          .WORD                  308
      030750 005056          .WORD                  WRTMSG
      030752 012154          .WORD                  SFIMSG
5021 030754          23$:   CKLOOP                    ;LOOP IF SELECTED
      030754 104406          TRAP                  C$CLP1
5022
5023 ;*****
5024 ;
5025 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5026 ;
5027 ;*****
5028
5029 030756 004737 011136          JSR      PC,REWIND          ;CALL THE TAPE REWIND
5030 030762 012703 000024          MOV      #20.,R3          ;STARTING RECORD SIZE
5031 030766 013737 003116 033652 65$:   MOV      FREE,T23WB      ;STARTING WRITE BUFFER ADDRESS
5032
5033 ;*****
5034 ;
5035 ;WRITE DATA,CVC=1,ACK COMMAND
5036 ;
5037 ;*****
5038
5039 030774 012737 140005 033650          MOV      #140005,T23PK3   ;WRITE DATA,CVC=1,ACK COMMAND
5040 031002 012737 140005 033672          MOV      #140005,T23WRT   ;SETUP FOR RETRY COMMAND
5041 031010 052737 004000 033672          BIS      #4000,T23WRT     ;MAKE IT A RETRY
5042 031016 012704 033650          MOV      #T23PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
5043 031022 010300          MOV      R3,R0           ;SET PATTERN IN CORRECT REGISTER
5044 031024 004737 017542          JSR      PC,FILLMEM       ;FILL MEMORY WITH RECORD SIZE
5045 031030 010337 033656          MOV      R3,T23SZ        ;SET UP RECORD SIZE IN PACKET
5046 031034 010465 000000          MOV      R4,TSDB(R5)     ;ISSUE COMMAND
5047 031040 004737 016370          JSR      PC,WAITF        ;WAIT FOR SSR TO SET
5048 031044 016501 000002          MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
5049 031050 012702 000200          MOV      #SSR,R2        ;SET UP EXPECTED
5050 031054 020102          CMP      R1,R2           ;ARE THEY EQUAL
5051 031056 001402          BEQ      80$             ;BR, IF OK
5052 031060 004737 035222          JSR      PC,T23CHK       ;CHECK SPECIAL CONDITION
5053 031064          80$:   CKLOOP                    ;LOOP IF SELECTED
      031064 104406          TRAP                  C$CLP1
5054 031066 016501 000000          MOV      TSBA(R5),R1     ;GET TSBA CONTENTS
5055 031072 012702 033542          MOV      #T23BFR,R2     ;SET UP EXPECTED
5056 031076 062702 000016          ADD      #16,R2         ;SET TO END OF MESSAGE BUFFER
5057 031102 005737 002216          TST      EXTFEA         ;CHECK FOR EXTENDED FEATURES SW SET
5058 031106 001402          BEQ      85$             ;BR, IF IT NOT SET
5059 031110 062702 000002          ADD      #2,R2          ;BUMP R2 FOR EXTRA DATA
5060 031114 020102          85$:   CMP      R1,R2        ;ARE THEY EQUAL
5061 031116 001406          BEQ      90$             ;BR, IF TSBA IS CORRECT
5062 031120 005237 002212          INC      FATFLG         ;BUMP COUNT
5066 031124          ERRHRD  ERRNO,T23BA,EXPREC ;TSBA WAS NOT CORRECT AFTER WRITE DATA
      031124 104456          TRAP                  C$ERHRD
      031126 000465          .WORD                  309
      031130 034665          .WORD                  T23BA
      031132 015614          .WORD                  EXPREC
5067 031134          90$:   CKLOOP                    ;LOOP IF SELECTED
      031134 104406          TRAP                  C$CLP1
5068 031136 020327 007376          CMP      R3,#7376        ;ONLY CHECK RAM UNTIL ITS FULL

```

TEST 3: BASIC WRITE DATA

```

5069 031142 002114          BGE      115$          ;IT WRAPS AROUND ETC.
5070 031144 004737 035134    JSR      PC,T23RT2    ;MAKE SURE PACKET AND DATA ARE CLEAN
5071 031150 012737 000400 033664    MOV      #256.,T23S2 ;STARTING RAM ADDRESS
5072 031156 112737 000000 033662    MOV      #0,T23BS0   ;STOP INTERNAL TSV05 DIAGNOSTICS
5073 031164 112737 000000 033663    MOV      #0,T23BS1   ;SIZE OF RAM READ
5074 031172 012704 033630    MOV      #T23PK2,R4  ;SET R4 WITH PACKET ADDRESS
5075 031176 010465 000000    MOV      R4,TSDB(R5) ;ISSUE WRITE SUB SYS MEM COMMAND
5076 031202 004737 016456    JSR      PC,CHKTSSR  ;CHECK TSSR AND WAIT FOR SSR TO SET
5077 031206 103407          BCS      92$         ;BR, IF NO ERRORS IN TSSR
5078 031210 010001          MOV      R0,R1       ;SAVE TSSR
5079 031212 005237 002212    INC      FATFLG      ;BUMP COUNT
5083 031216          ERRHRD  ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
      031216 104456          TRAP    C$ERHRD
      031220 000466          .WORD  310
      031222 034737          .WORD  T23WSS
      031224 012166          .WORD  PKTSSR
5084 031226          92$:  CKLOOP          ;LOOP IF SELECTED
      031226 104406          TRAP    C$CLP1
5085 031230 004737 035134    JSR      PC,T23RT2    ;MAKE SURE PACKET AND DATA ARE CLEAN
5086 031234 012737 000400 033664    MOV      #256.,T23S2 ;STARTING RAM ADDRESS
5087 031242 112737 000001 033662    MOV      #1,T23BS0   ;READ RAM COMMAND FOR WRITE SUB SYS M.
5088 031250 112737 000002 033663    MOV      #2,T23BS1   ;SIZE OF RAM READ
5089 031256 012704 033630    MOV      #T23PK2,R4  ;SET R4 WITH PACKET ADDRESS
5090 031262 010465 000000 95$:  MOV      R4,TSDB(R5) ;ISSUE WRITE SUB SYS MEM COMMAND
5091 031266 004737 016456    JSR      PC,CHKTSSR  ;CHECK TSSR AND WAIT FOR SSR TO SET
5092 031272 103407          BCS      100$        ;BR, IF NO ERRORS IN TSSR
5093 031274 010001          MOV      R0,R1       ;SAVE TSSR
5094 031276 005237 002212    INC      FATFLG      ;BUMP COUNT
5098 031302          ERRHRD  ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
      031302 104456          TRAP    C$ERHRD
      031304 000467          .WORD  311
      031306 034737          .WORD  T23WSS
      031310 012166          .WORD  PKTSSR
5099 031312          100$: CKLOOP          ;LOOP IF SELECTED
      031312 104406          TRAP    C$CLP1
5100 031314 005001          CLR      R1           ;CLEAR REGISTER
5101 031316 005002          CLR      R2           ;CLEAR REGISTER
5102 031320 013701 033562    MOV      T23BFR+20,R1 ;PICK UP BYTE READ FROM RAM
5103 031324 010302          MOV      R3,R2       ;SET UP EXPECTED
5104 031326 020102          CMP      R1,R2       ;IS RAM DATA CORRECT
5105 031330 001406          BEQ      110$        ;BR, IF OK (EQUAL)
5106 031332 005237 002212    INC      FATFLG      ;BUMP COUNT
5110 031336          ERRHRD  ERRNO,T23RNC,EXPREC ;RNC=RAM NOT CORRECT
      031336 104456          TRAP    C$ERHRD
      031340 000470          .WORD  312
      031342 034225          .WORD  T23RNC
      031344 015614          .WORD  EXPREC
5111 031346          110$: CKLOOP          ;LOOP IF SELECTED
      031346 104406          TRAP    C$CLP1
5112 031350 005237 033664    INC      T23S2       ;BUMP RAM ADDRESS TO BE CHECKED
5113 031354 005237 033664    INC      T23S2       ;BUMP RAM ADDRESS TO BE CHECKED
5114 031360 010301          MOV      R3,R1       ;GET SIZE OF RECORD
5115 031362 062701 000400    ADD      #256.,R1    ;FIGURE OUT END RECORD ADDRESS
5116 031366 023701 033664    CMP      T23S2,R1    ;AT END OF RAM CHECK YET
5117 031372 001333          BNE      95$         ;BR, IF MORE TO CHECK
5118 031374 062703 001750    ADD      #1000.,R3   ;NEXT RECORD SIZE/DATA PATTERN
5119 031400 020337 033660    CMP      R3,T23RSZ   ;IS R3 OVER MAX RECORD SIZE

```


TEST 3: BASIC WRITE DATA

```

5172 031542 103407          BCS      20$          ;BR IF INIT WAS OK
5173 031544 005237 002212  INC      FATFLG      ;BUMP COUNT
5177 031550 010001          MOV      RO,R1       ;CONTENTS OF TSSR REGISTER
5178 031552          ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      031552 104455          TRAP    C$ERDF
      031554 000472          .WORD  314
      031556 003652          .WORD  SFIERR
      031560 012154          .WORD  SFIMSG

5179 031562          20$:
5180 031562 013737 002172 033540  MOV      UNITN,T23DSW ;SET UP UNIT NUMBER
5181 031570 012704 033520  MOV      #T23PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
5182
5183 ;*****
5184 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
5185 ;
5186 ;*****
5187
5188
5189 031574 004737 010752          JSR      PC,WRTCHR    ;ISSUE WRITE CHARACTERISTICS
5190 031600 103407          BCS      23$          ;BR, IF COMMAND ISSUED OK
5191 031602 005237 002212  INC      FATFLG      ;BUMP COUNT
5195 031606 010001          MOV      RO,R1       ;SAVE CONTENTS OF TSSR
5196 031610          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      031610 104456          TRAP    C$ERHRD
      031612 000473          .WORD  315
      031614 005056          .WORD  WRTMSG
      031616 012154          .WORD  SFIMSG

5197 031620          23$:
5198 031620 012703 000024          MOV      #20.,R3     ;STARTING RECORD SIZE
5199 031624 013737 003116 033652 65$:  MOV      FREE,T23WB  ;STARTING WRITE BUFFER ADDRESS
5200
5201 ;*****
5202 ;WRITE DATA,CVC=1,ACK,SWB COMMAND
5203 ;
5204 ;*****
5205
5206
5207 031632 012737 150005 033650  MOV      #150005,T23PK3 ;WRITE DATA,CVC=1,ACK,SWB COMMAND
5208 031640 012737 150005 033672  MOV      #150005,T23WRT ;SETUP FOR RETRY COMMAND
5209 031646 052737 004000 033672  BIS      #4000,T23WRT ;MAKE IT A RETRY
5210 031654 012704 033650  MOV      #T23PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
5211 031660 010300  MOV      R3,R0        ;SET PATTERN IN CORRECT REGISTER
5212 031662 004737 017542  JSR      PC,FILLMEM   ;FILL MEMORY WITH RECORD SIZE
5213 031666 010337 033656  MOV      R3,T23SZ     ;SET UP RECORD SIZE IN PACKET
5214 031672 010465 000000  MOV      R4,TSDB(R5)  ;ISSUE COMMAND
5215 031676 004737 016370  JSR      PC,WAITF     ;WAIT FOR SSR TO SET
5216 031702 016501 000002  MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
5217 031706 012702 000200  MOV      #SSR,R2     ;SET UP EXPECTED
5218 031712 020102  CMP      R1,R2        ;ARE THEY EQUAL
5219 031714 001402  BEQ      80$          ;BR, IF OK
5220 031716 004737 035222  JSR      PC,T23CHK   ;CHECK SPECIAL CONDITION
5221 031722          80$:  CKLOOP ;LOOP IF SELECTED
      031722 104406          TRAP    C$CLP1
5222 031724 016501 000000  MOV      TSBA(R5),R1  ;GET TSBA CONTENTS
5223 031730 012702 033542  MOV      #T23BFR,R2  ;SET UP EXPECTED
5224 031734 062702 000016  ADD      #16,R2       ;SET TO END OF MESSAGE BUFFER
5225 031740 005737 002216  TST     EXTFEA       ;CHECK FOR EXTENDED FEATURES SW SET

```

TEST 3: BASIC WRITE DATA

```

5226 031744 001402          BEQ      85$          ;BR, IF IT NOT SET
5227 031746 062702 000002  ADD      #2,R2       ;BUMP R2 FOR EXTRA DATA
5228 031752 020102          CMP      R1,R2       ;ARE THEY EQUAL
5229 031754 001406          BEQ      90$          ;BR, IF TSBA IS CORRECT
5230 031756 005237 002212  INC      FATFLG      ;BUMP COUNT
5234 031762          ERRHRD  ERRNO,T23BA,EXPREC ;TSBA WAS NOT CORRECT AFTER WRITE DATA
                    TRAP      C$ERHRD
                    .WORD    316
                    .WORD    T23BA
                    .WORD    EXPREC
                    TRAP      C$CLP1
5235 031772          90$:  CKLOOP      ;LOOP IF SELECTED
                    TRAP      C$CLP1
5236 031774 104406          CMP      R3,#7376    ;ONLY CHECK RAM UNTIL ITS FULL
5237 032000 020327 007376  BGE     115$         ;IT WRAPS AROUND ETC.
5238 032002 004737 035134  JSR     PC,T23RT2    ;MAKE SURE PACKET AND DATA ARE CLEAN
5239 032006 012737 000400 033664  MOV     #256.,T23S2 ;STARTING RAM ADDRESS
5240 032014 112737 000000 033662  MOVB   #0,T23BS0    ;STOP INTERNAL TSV05 DIAGNOSTICS
5241 032022 112737 000000 033663  MOVB   #0,T23BS1    ;SIZE OF RAM READ
5242 032030 012704 033630  MOV     #T23PK2,R4   ;SET R4 WITH PACKET ADDRESS
5243 032034 010465 000000  MOV     R4,TSDB(R5) ;ISSUE WRITE SUB SYS MEM COMMAND
5244 032040 004737 016456  JSR     PC,CHKTSSR   ;CHECK TSSR AND WAIT FOR SSR TO SET
5245 032044 103407          BCS     92$          ;BR, IF NO ERRORS IN TSSR
5246 032046 010001          MOV     R0,R1        ;SAVE TSSR
5247 032050 005237 002212  INC     FATFLG      ;BUMP COUNT
5251 032054          ERRHRD  ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
                    TRAP      C$ERHRD
                    .WORD    317
                    .WORD    T23WSS
                    .WORD    PKTSSR
5252 032064          92$:  CKLOOP      ;LOOP IF SELECTED
                    TRAP      C$CLP1
5253 032066 104406          JSR     PC,T23RT2    ;MAKE SURE PACKET AND DATA ARE CLEAN
5254 032072 012737 000400 033664  MOV     #256.,T23S2 ;STARTING RAM ADDRESS
5255 032100 112737 000001 033662  MOVB   #1,T23BS0    ;READ RAM COMMAND FOR WRITE SUB SYS M.
5256 032106 112737 000002 033663  MOVB   #2,T23BS1    ;SIZE OF RAM READ
5257 032114 012704 033630  MOV     #T23PK2,R4   ;SET R4 WITH PACKET ADDRESS
5258 032120 010465 000000  95$:  MOV     R4,TSDB(R5) ;ISSUE WRITE SUB SYS MEM CMD (READ RAM)
5259 032124 004737 016456  JSR     PC,CHKTSSR   ;CHECK TSSR AND WAIT FOR SSR TO SET
5260 032130 103407          BCS     100$         ;BR, IF NO ERRORS IN TSSR
5261 032132 010001          MOV     R0,R1        ;SAVE TSSR
5262 032134 005237 002212  INC     FATFLG      ;BUMP COUNT
5266 032140          ERRHRD  ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
                    TRAP      C$ERHRD
                    .WORD    318
                    .WORD    T23WSS
                    .WORD    PKTSSR
5267 032150          100$: CKLOOP      ;LOOP IF SELECTED
                    TRAP      C$CLP1
5268 032152 104406          CLR     R1           ;CLEAR REGISTERS
5269 032154 005001          CLR     R2           ;CLEAR REGISTERS
5270 032156 013701 033562  MOV     T23BFR+20,R1 ;PICK UP BYTE READ FROM RAM
5271 032162 010302          MOV     R3,R2       ;SET UP EXPECTED
5272 032164 000302          SWAB   R2           ;SWAP BYTES
5273 032166 020102          CMP     R1,R2       ;IS RAM DATA CORRECT
5274 032170 001406          BEQ     110$         ;BR, IF OK (EQUAL)
5275 032172 005237 002212  INC     FATFLG      ;BUMP COUNT
5279 032176          ERRHRD  ERRNO,T23RNC,EXPREC ;RNC=RAM NOT CORRECT

```


TEST 3: BASIC WRITE DATA

```

5432
5433 032646 004737 010752      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
5434 032652 103407              BCS      123$           ;BR, IF COMMAND ISSUED OK
5435 032654 005237 002212      INC      FATFLG        ;BUMP COUNT
5439 032660 010001              MOV      R0,R1         ;SAVE CONTENTS OF TSSR
5440 032662              ERRHRD  ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICSC FAILED
      032662 104456              TRAP     C$ERHRD
      032664 000505              .WORD   325
      032666 005056              .WORD   WRTMSG
      032670 012154              .WORD   SFMSG

5441
5442
5443
5444
5445
5446
5447
5448 032672
5449 032672 005737 002216      123$:  TST      EXTFEA      ;CHECK FOR EXTENDED FEATURES SW SWITCH
5450 032676 001026              BNE      130$           ;BR IF SWITCH IS ON
5451 032700 005237 002216      INC      EXTFEA        ;ONLY ONE TIME
5452 032704 112737 000200 033663  MOVB     #200,T23BS1    ;WRITE MISCELLANEOUS CONT/READ STATUS
5453 032712 112737 000010 033662  MOVB     #10,T23BS0     ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
5454 032720 012704 033630      MOV      #T23PK2,R4    ;WRITE SUBSYS MEM PACKET
5455 032724 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
5456 032730 004737 016456      JSR      PC,CHKTSSR    ;WAIT FOR SSR
5457 032734 103407              BCS      130$           ;BR, IF NO ERROR
5458 032736 010001              MOV      R0,R1         ;ERROR, SAVE TSSR
5459 032740 005237 002212      INC      FATFLG        ;BUMP COUNT
5463 032744              ERRHRD  ERRNO,T22SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      032744 104456              TRAP     C$ERHRD
      032746 000506              .WORD   326
      032750 027230              .WORD   T22SSR
      032752 012166              .WORD   PKTSSR

5464 032754
5465 032754 104406      130$:  CKLOOP          ;LOOP IF SELECTED
      TRAP     C$CLP1

5466 032756 012701 160000      MOV      #160000,R1    ;NXM LOW ADDRESS START
5467 032762 012702 177776      MOV      #177776,R2    ;LIMIT CHECK FOR NXM (HIGHEST)
5468 032766 004737 016516      JSR      PC,NXM        ;LOOK FOR NXM ADDRESS
5469 032772 103045              BCC      80$           ;BR, IF NON FOUND
5470 032774 010137 003132      MOV      R1,NXML0     ;SET ADDRESS UP FOR TEST
5471
5472
5473 033000 005037 033654      CLR      T23WB+2       ;CLEAR OUT THE HIGH BITS AREA
5474 033004
5475 033004 012737 140005 033650 24$:  MOV      #140005,T23PK3 ;WRITE DATA, ACK, CVC=1
5476 033012 013737 003132 033652  MOV      NXML0,T23WB    ;SET UP WRITE BUFFER ADDRESS
5477 033020 012737 000100 033656  MOV      #64.,T23SZ     ;SET UP BUFFER SIZE
5478 033026 012704 033650      MOV      #T23PK3,R4    ;R4 = POINTER TO PACKET
5479 033032 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
5480 033036 004737 016370      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
5481 033042 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
5482 033046 012702 104210      MOV      #SC!NXM!SSR!BIT3,R2 ;SET UP EXPECTED
5483 033052 020102              CMP      R1,R2         ;ARE THEY EQUAL
5484 033054 001414              BEQ      80$           ;BR, IF OK ESP. FUNCTION REJECT
5485 033056 005237 033654      INC      T23WB+2       ;BUMP TO NEXT ADDRESS BIT

```


TEST 3: BASIC WRITE DATA

```

5536 033210 012704 033520          MOV      #T23PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
5537
5538                               ;*****
5539                               ;
5540                               ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
5541                               ;
5542                               ;*****
5543
5544 033214 004737 010752          JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
5545 033220 103407                BCS     23$                ;BR, IF COMMAND ISSUED OK
5546 033222 005237 002212          INC     FATFLG            ;BUMP COUNT
5550 033226 010001                MOV     R0,R1             ;SAVE CONTENTS OF TSSR
5551 033230                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
                                TRAP     C$ERHRD
                                .WORD    329
                                .WORD    WRTMSG
                                .WORD    SFIMSG
033230 104456
033232 000511
033234 005056
033236 012154

5552
5553                               ;*****
5554                               ;
5555                               ;WRITE DATA, ACK,CVC=1
5556                               ;
5557                               ;*****
5558
5559 033240                23$:
5560 033240 012701 160000          MOV     #160000,R1        ;NXM LOW ADDRESS START
5561 033244 012702 177776          MOV     #177776,R2        ;LIMIT CHECK FOR NXM (HIGHEST)
5562 033250 004737 016516          JSR     PC, NXNM          ;LOOK FOR NXM ADDRESS
5563 033254 103051                BCC     80$                ;BR, IF NON FOUND
5564 033256 010137 003132          MOV     R1,NXML0         ;SET ADDRESS UP FOR TEST
5565 033262 012737 000000 033654  MOV     #0,T23WB+2        ;SET TO 16 BIT ADDRESS

5566 033270                24$:
5567 033270 012737 140005 033650  MOV     #140005,T23PK3    ;WRITE DATA, ACK,CVC=1
5568 033276 013701 003132          MOV     NXML0,R1          ;HIGHEST MEMORY ADDRESS LOW BITS
5569 033302 162701 000500          SUB     #500,R1           ;SET ADDRESS A LITTLE LOWER
5570 033306 010137 033652          MOV     R1,T23WB         ;LOAD INTO THE PACKET
5571 033312 012737 000000 033656  MOV     #0,T23SZ          ;SET UP BUFFER SIZE (64K BYTES)
5572 033320 012704 033650          MOV     #T23PK3,R4        ;R4 = POINTER TO PACKET
5573 033324 010465 000000          MOV     R4,TSDB(R5)       ;ISSUE COMMAND
5574 033330 004737 016370          JSR     PC,WAITF          ;WAIT FOR SSR TO SET
5575 033334 016501 000002          MOV     TSSR(R5),R1       ;GET TSSR CONTENTS
5576 033340 012702 104210          MOV     #SC!NXM!SSR!BIT3,R2 ;SET UP EXPECTED
5577 033344 020102                CMP     R1,R2             ;ARE THEY EQUAL
5578 033346 001414                BEQ     80$                ;BR, IF OK ESP. FUNCTION REJECT
5579 033350 005237 033654          INC     T23WB+2          ;BUMP TO NEXT ADDRESS RANGE
5580 033354 023727 033654 000004  CMP     T23WB+2,#4        ;CHECK TO SEE IF WE WENT TO HIGH
5581 033362 001342                BNE     24$                ;BR, IF NO OVER FLOW
5582 033364 005237 002212          25$: INC     FATFLG          ;BUMP COUNT
5586 033370                ERRHRD  ERRNO,T23TM,PKTSSR ;TSSR INCORRECT AFTER WRITE COMMAND
                                TRAP     C$ERHRD
                                .WORD    330
                                .WORD    T23TM
                                .WORD    PKTSSR
                                033370 104456
                                033372 000512
                                033374 034102
                                033376 012166

5587 033400                80$: CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP1
                                033400 104406
5588 033402 004737 035134          JSR     PC,T23RT2        ;CLEAN UP PACKET
5589 033406 004737 035176          JSR     PC,T23RT3        ;RESTORE PACKET

```


E11

TEST 3: BASIC WRITE DATA

```

5648 033652 000000      T23WB:  .WORD  0      ;ADDRESS OF WRITE BUFFER
5649 033654 000000      .WORD  0
5650 033656 000000      T23SZ:  .WORD  0      ;SIZE OF BUFFER (EXTENT)
5651                      .EVEN
5652                      ;
5653 033660 000000      †T23RSZ: .WORD  0      ;LARGEST TAPE RECORD IN BYTES
5654                      ;
5655                      ;
5656 033662              †T23BF2:
5657 033662              T23BS0: .BYTE  10      ;BSELO AREA
5658 033663              T23BS1: .BYTE 200      ;BSEL1 AREA
5659 033664 000000      T23S2:  .WORD  0      ;SEL 2 AREA
5660 033666 000000      T23S3:  .WORD  0      ;DATA AREA
5661                      ;
5662                      ;
5663 033670 000000      T23TMP: .WORD  0      ;TEMPORARY REGISTER
5664 033672 000000      T23WRT: .WORD  0      ;RETRY COMMAND
5665                      ;
5666                      .EVEN
5667                      ;TAPE MOTION PACKET COMMAND VALUES
5668
5669 033674 100005      T23WD:  .WORD 100005    ;WRITE DATA (NEXT)
5670 033676 100405      T23WDR: .WORD 100405    ;WRITE DATA RETRY
5671 033700 102005      T23CON: .WORD 102005    ;WRITE CONTINOUS
5672 033702 177777      .WORD  177777    ;END OF DATA
5673
5674                      ;*
5675                      ;LOCAL TEXT MESSAGES FOR TEST
5676                      ;-
5677
5678 033704          127      122      111  T23SSR: .ASCIZ 'WRITE Command Not Accepted'
5679 033737          105      117      124  T23ET:  .ASCIZ 'EOT Not Found In 12000 4k Writes, (Use Shorter Tape)'
5680 034024          127      122      111  T23EOT: .ASCIZ 'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
5681 034102          124      123      123  T23TM:  .ASCIZ 'TSSR Not Correct After WRITE Command Reject'
5682 034156          122      145      167  T23RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
5683 034225          122      101      115  T23RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
5684 034300          124      123      123  T23AM3: .ASCIZ 'TSSR Init. Failed After WRITE Command'
5685 034346          104      162      151  T23OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
5686 034421          124      123      123  T23WDD: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, SWB Bit Set'
5687 034510          124      123      123  T23WDC: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
5688 034612          103      126      103  T23VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
5689 034665          124      123      102  T23BA:  .ASCIZ 'TSBA Not Correct After WRITE DATA Command'
5690 034737          127      122      111  T23WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
5691 035026          102      141      163  TST23ID: .ASCIZ 'Basic Write'
5692                      .EVEN
5693                      ;*
5694                      ;
5695                      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5696                      ;WRITE SUBSYSTEM MEMORY COMMAND
5697                      ;
5698                      ;-
5699
5700 035042              T23REST:
5701 035042              SAVREG          ;SAVE THE REGISTERS
5702 035046 012701 033520      MOV      #T23PACKET,R1      ;START OF THE PACKET
5703 035052 012721 100004      MOV      #100004,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK
5704 035056 012721 033530      MOV      #T23DATA,(R1)+    ;ADDRESS OF CHARAISTICS DATA BLOCK

```


TEST 3: BASIC WRITE DATA

```

5705 035062 005021          CLR      (R1)+          ;EXTENDED ADDRESS
5706 035064 012721 000012  MOV      #10.,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
5707 035070 012721 033542  MOV      #T23BFR,(R1)+   ;ADDRESS OF MESSAGE BUFFER
5708 035074 005021          CLR      (R1)+          ;
5709 035076 012721 000024  MOV      #20.,(R1)+      ;LENGTH OF MESSAGE BUFFER
5710 035102 005021          CLR      (R1)+          ;
5711 035104 012711 000000  MOV      #0,(R1)         ;SELECT DRIVE ZERO
5712 035110 012702 000030  MOV      #24.,R2         ;NUMBER OF LOCATIONS TO BE CLEARED
5713 035114 012762 177777 033542 64$: MOV      #177777,T23BFR(R2) ;ALL ONES TO MESSAGE BUFFER
5714 035122 005742          TST      -(R2)           ;BUMP DOWN TO NEXT LOCATION
5715 035124 020227 000000  CMP      R2,#0           ;R2 AT ZERO YET
5716 035130 001371          BNE      64$             ;KEEP GOING UNTIL DONE
5717 035132 000207          RTS      PC              ;RETURN
5718
5719
5720 035134          T23RT2:
5721 035134          SAVREG          ;SAVE THE REGISTERS
5722 035140 012701 033630  MOV      #T23PK2,R1      ;START OF THE PACKET
5723 035144 012721 100006  MOV      #100006,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK
5724 035150 012721 033662  MOV      #T23BF2,(R1)+   ;ADDRESS OF DATA BLOCK
5725 035154 005021          CLR      (R1)+          ;EXTENDED ADDRESS
5726 035156 012721 000006  MOV      #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
5727 035162 012701 033662  MOV      #T23BF2,R1      ;POINT TO DATA SEL AREA
5728 035166 005021          CLR      (R1)+          ;
5729 035170 005021          CLR      (R1)+          ;
5730 035172 005011          CLR      (R1)           ;
5731 035174 000207          RTS      PC              ;RETURN
5732 035176          T23RT3:
5733 035176          SAVREG          ;SAVE THE REGISTERS
5734 035202 012701 033650  MOV      #T23PK3,R1      ;START OF THE PACKET
5735 035206 012721 100005  MOV      #100005,(R1)+   ;WRITE TAPE. WITH ACK
5736 035212 005021          CLR      (R1)+          ;ADDRESS OF DATA BLOCK
5737 035214 005021          CLR      (R1)+          ;EXTENDED ADDRESS
5738 035216 005011          CLR      (R1)           ;SIZE OF DATA BLOCK
5739 035220 000207          RTS      PC              ;RETURN
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749 035222          ;+
5750 035222          ;ROUTINE TO RETRY WRITE DATA IN CASE OF BAD TAPE FOR TEST
5751 035226 005037 033670  ;3,SUBTEST 2 & 3
5752 035232 032701 100000  ;
5753 035236 001452          ;INPUTS:          R1=TSSR
5754 035240 013702 033552  ;SUBROUTINE SETS UP T23WRT FOR RETRY
5755 035244 032702 000002  ;
5756 035250 001401          ;-
5757 035252 000405          T23CHK:
5758 035254 032702 020000  SAVREG          ;SAVE THE REGISTERS
5759 035260 001002          CLR      T23TMP         ;CLEAR LOCAL REGISTER
5760 035262 000440          BIT      #SC,R1         ;IS SC SET IN TSSR?
5761 035264 000207          BEQ      FATAL         ;NO, YOU GOT PROBLEMS!
                          MOV      T23BFR+10,R2      ;YES,GET XSTAT1
                          BIT      #X1.UNC,R2         ;IS UNC SET IN XSTAT1?
                          BEQ      1$                 ;NO, CHECK COR
                          BR       RETRY              ;YES,DO WRITE DATA RETRY
                          BIT      #X1.COR,R2         ;IS COR SET IN XSTAT1 THEN?
                          BNE      RETRY              ;YES SO RETRY
                          BR       FATAL              ;NO, YOU GOT PROBLEMS
EXIT:  RTS      PC              ;RETURN

```

G11

TEST 3: BASIC WRITE DATA

```

5762
5763 035266          RETRY:
5764 035266 012703 000024 2$: MOV #20.,R3 ;STARTING RECORD SIZE
5765 035272 013737 003116 033652 MOV FREE,T23WB ;STARTING WRITE BUFFER ADDRESS
5766 035300 012737 033672 033650 MOV #T23WRT,T23PK3 ;WRITE DATA RETRY COMMAND SETUP BY SUBROUTINE
5767 035306 012704 033650 MOV #T23PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
5768 035312 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
5769 035314 004737 017542 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
5770 035320 010337 033656 MOV R3,T23SZ ;SET UP RECORD SIZE IN PACKET
5771 035324 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
5772 035330 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
5773 035334 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
5774 035340 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
5775 035344 020102 CMP R1,R2 ;ARE THEY EQUAL
5776 035346 001746 BEQ EXIT ;BR, IF OK
5777 035350 005237 033670 INC T23TMP ;TRY FIVE TIMES THEN EXIT
5778 035354 022737 000005 033670 CMP #5,T23TMP ;DONE FIVE YET?
5779 035362 001341 BNE 2$ ;NO GO AGAIN
5780 035364 005237 002212 FATAL: INC FATFLG ;BUMP COUNT
5784 035370 013702 033542 MOV T23BFR,R2 ;LOW ORDER MSGBUF
5785 035374 ERRHRD ERRNO,SCHERR,PKTMES ;TSSR INCORRECT AFTER WRITE DATA
      035374 104456 TRAP C$ERHRD
      035376 000514 .WORD 332
      035400 005300 .WORD SCHERR
      035402 012230 .WORD PKTMES
5786 035404 004737 017322 JSR PC,CKDROP ;DROP THE UNIT
5787 035410 ENDTST
      035410 104401 L10043: TRAP C$ETST
      035410

```

.SBTTL TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

5789
5790
5791 ;*
5792 ;THIS TEST VERIFIES THAT THE READ FORWARD AND READ REVERSE
5793 ;COMMANDS OPERATE PROPERLY. VARIOUS COMBINATIONS OF ODD AND EVEN
5794 ;DATA BUFFER BOUNDARIES, RECORD SIZES (UP TO 64K BYTES IF MEMORY
5795 ;SPACE IS AVAILIABLE), AND BYTE-SWAP CONTROL ARE USED. THIS TEST
5796 ;OF COURSE, FURTHER VERIFIES THE WRITE DATA COMMAND BY ACTUALLY
5797 ;READING AND VERIFYING WRITTEN DATA. ALSO TESTED ARE PROPER
5798 ;TERMINATIONS ON EXCEPTIONAL OR ERROR CONDITIONS: RECORD LENGTH
5799 ;LONG, RECORD LENGTH SHORT, READ REVERSE AT BOT, ILLEGAL DATA
5800 ;BUFFER ADDRESSES, ILLEGAL CODES IN THE MODE FIELD OF THE BASIC
5801 ;READ COMMAND, AND DATA BUFFERS IN NON-EXISTANT MEMORY. THE TEST
5802 ;
5803 ;
5804 ;THE TEST CONSISTS OF THE FOLLOWING 14 SUBTESTS
5805 ;
5806 ;
5807 ;
5808 ;-

```

```

5809 035412          BGNTST
5810 035412          T4::
5810 035412 012737 006360 002170 MOV #EPRT1,EPRTSW ;SET UP PRIMARY ERROR MESSAGE
5811 035420 005037 003126 CLR KTENABLE ;TURN OFF KT11
5812 035424 004737 017414 JSR PC,KTOFF ;TURN KT11 OFF
5817 035430 012700 047422 MOV #TST24ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
5818 035434 004737 016630 JSR PC,TSTSETUP ;DO INITIAL TEST SETUP

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

5865
5866
5867
5868
5869
5870
5871
5872 035576 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
5873 035602 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
5874 035604 005237 002212 INC FATFLG ;BUMP COUNT
5878 035610 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
5879 035612 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
035612 104456 TRAP C$ERHRD
035614 000622 .WORD 402
035616 005056 .WORD WRTMSG
035620 012154 .WORD SFIMSG
5880 035622 005737 002216 24$: TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
5881 035626 001044 BNE 50$ ;BR IF SWITCH IS ON
5882
5883 035630 112737 000200 045141 MOVB #200,T24BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
5884 035636 112737 000010 045140 MOVB #10,T24BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
5885 035644 012704 045110 MOV #T24PK2,R4 ;WRITE SUBSYS MEM PACKET
5886 035650 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
5887 035654 004737 016456 JSR PC,CHKTSSR ;WAIT FOR SSR
5888 035660 103407 BCS 30$ ;BR, IF NO ERROR
5889 035662 010001 MOV RO,R1 ;ERROR, SAVE TSSR
5890 035664 005237 002212 INC FATFLG ;BUMP COUNT
5894 035670 ERRHRD ERRNO,T24SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
035670 104456 TRAP C$ERHRD
035672 000623 .WORD 403
035674 045677 .WORD T24SSR
035676 012166 .WORD PKTSSR
5895 035700 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
035700 104406
5896 035702 012737 000007 045020 MOV #7,T24DSW ;SET DRIVE NUMBER IN PACKET
5897 035710 012704 045000 MOV #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
5898
5899
5900
5901
5902
5903
5904
5905 035714 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
5906 035720 103407 BCS 50$ ;BR, IF COMMAND ISSUED OK
5907 035722 005237 002212 INC FATFLG ;BUMP COUNT
5911 035726 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
5912 035730 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
035730 104456 TRAP C$ERHRD
035732 000624 .WORD 404
035734 005056 .WORD WRTMSG
035736 012154 .WORD SFIMSG
5913 035740 50$: CKLOOP ;SCOPE LOOP TRAP C$CLP1
035740 104406
5914 035742 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
5915 035746 032701 000100 BIT #OFL,R1 ;CHECK FOR THE OFFLINE BIT SET
5916 035752 001006 BNE 60$ ;BR, IF OFFLINE (GOOD)

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

6020 036210 010001          MOV    R0,R1          ;SAVE TSSR
6021 036212 005237 002212  INC    FATFLG        ;BUMP COUNT
6025 036216          ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
        036216 104456          TRAP  C$ERHRD
        036220 000631          .WORD 409
        036222 046266          .WORD T24RWN
        036224 012166          .WORD PKTSSR
6026 036226          30$:  CKLOOP          ;LOOP IF SELECTED
        036226 104406          TRAP  C$CLP1
6027
6028
6029
6030
6031
6032
6033
        ;*****
        ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
        ;*****
6034 036230 013701 045030          MOV    T24BFR+6,R1    ;PICK UP XSTO
6035 036234 010102          MOV    R1,R2          ;SET UP EXPECTED
6036 036236 052702 000002          BIS    #BIT1,R2       ;SET BOT BIT IN EXPECTED
6037 036242 020102          CMP    R1,R2          ;DOES EXP = REC'D
6038 036244 001406          BEQ   40$             ;BR, IF EQUAL (OK)
6039 036246 005237 002212  INC    FATFLG        ;BUMP COUNT
6043 036252          ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
        036252 104456          TRAP  C$ERHRD
        036254 000632          .WORD 410
        036256 046003          .WORD T24BOT
        036260 015614          .WORD EXPREC
6044 036262          40$:  CKLOOP          ;LOOP IF SELECTED
        036262 104406          TRAP  C$CLP1
6045 036264 012703 000400          MOV    #256.,R3       ;RECORD SIZE
6046 036270 013737 003116 045132  MOV    FREE,T24RB     ;STARTING WRITE BUFFER ADDRESS
6047
6048
6049
6050
6051
6052
6053
        ;*****
        ;WRITE DATA,CVC=1,ACK COMMAND
        ;*****
6054 036276 012737 140005 045130  MOV    #140005,T24PK3 ;WRITE DATA,CVC=1,ACK COMMAND
6055 036304 012704 045130          MOV    #T24PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
6056 036310          65$:
6057 036310 010300          MOV    R3,R0          ;SET PATTERN IN CORRECT REGISTER
6058 036312 004737 017542          JSR   PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
6059 036316 010337 045136          MOV    R3,T24SZ       ;SET UP RECORD SIZE IN PACKET
6060 036322 010465 000000          MOV    R4,TSDB(R5)    ;ISSUE COMMAND
6061 036326 004737 016370          JSR   PC,WAITF        ;WAIT FOR SSR TO SET
6062 036332 016501 000002          MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
6063 036336 012702 000200          MOV    #SSR,R2        ;SET UP EXPECTED
6064 036342 020102          CMP    R1,R2          ;ARE THEY EQUAL
6065 036344 001406          BEQ   75$             ;BR, IF OK
6066 036346 005237 002212  INC    FATFLG        ;BUMP COUNT
6070 036352          ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
        036352 104456          TRAP  C$ERHRD
        036354 000633          .WORD 411
        036356 005113          .WORD WRTErr
        036360 012166          .WORD PKTSSR
6071 036362          75$:  CKLOOP          ;LOOP IF SELECTED
    
```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

6072 036362 104406                                TRAP      C$CLP1
6072 036364 005723                                TST      (R3)+      ;BUMP RECORD SIZE
6073 036366 022703 000414                        CMP      #268.,R3  ;END OF RECORD YET
6074 036372 001346                                BNE     65$        ;BR, IF MORE RECORDS TO WRITE
6075 036374 104406                                80$:    CKLOOP     ;LOOP IF SELECTED
6076 036376 104406                                TRAP     C$CLP1
6077 036376 104406                                120$:
6078 ;*****
6079 ;
6080 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6081 ;
6082 ;*****
6083 ;
6084 036376 004737 011136                        JSR     PC,REWIND   ;CALL TAPE REWIND COMMAND
6085 036402 004737 016456                        JSR     PC,CHKTSSR ;SEE HOW TSSR IS
6086 036406 103407                                BCS    130$        ;BR, IF NO PROBLEM
6087 036410 010001                                MOV     R0,R1      ;SAVE TSSR
6088 036412 005237 002212                        INC     FATFLG     ;BUMP COUNT
6092 036416 104456                                ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
6092 036416 104456                                TRAP     C$ERHRD
6092 036420 000634                                .WORD   412
6092 036422 046266                                .WORD   T24RWN
6092 036424 012166                                .WORD   PKTSSR
6093 036426 104406                                130$:   CKLOOP     ;LOOP IF SELECTED
6093 036426 104406                                TRAP     C$CLP1
6094 ;*****
6095 ;
6096 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
6097 ;
6098 ;*****
6099 ;
6100 ;
6101 036430 013701 045030                        MOV     T24BFR+6,R1 ;PICK UP XST0
6102 036434 010102                                MOV     R1,R2      ;SET UP EXPECTED
6103 036436 052702 000002                        BIS     #BIT1,R2   ;SET BOT BIT IN EXPECTED
6104 036442 020102                                CMP     R1,R2      ;DOES EXP = REC'D
6105 036444 001406                                BEQ    140$        ;BR, IF EQUAL (OK)
6106 036446 005237 002212                        INC     FATFLG     ;BUMP COUNT
6110 036452 104456                                ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
6110 036452 104456                                TRAP     C$ERHRD
6110 036454 000635                                .WORD   413
6110 036456 046003                                .WORD   T24BOT
6110 036460 015614                                .WORD   EXPREC
6111 036462 104406                                140$:   CKLOOP     ;LOOP IF SELECTED
6111 036462 104406                                TRAP     C$CLP1
6112 036464 012703 000400                        MOV     #256.,R3   ;RECORD SIZE
6113 036470 013737 003116 045132                MOV     FREE,T24RB ;STARTING READ BUFFER ADDRESS
6114 ;*****
6115 ;
6116 ;READ DATA,CVC=1,ACK COMMAND
6117 ;
6118 ;*****
6119 ;
6120 ;
6121 036476 012737 140001 045130                MOV     #140001,T24PK3 ;READ DATA,CVC=1,ACK COMMAND
6122 036504 012704 045130                165$:   MOV     #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

6228 036776          ERRHRD  ERRNO,T24RWN,PKTSSR      ;REWIND NOT ACCEPTED
      036776 104456
      037000 000642
      037002 046266
      037004 012166
6229 037006          30$:   CKLOOP                    ;LOOP IF SELECTED
      037006 104406
6230
6231
6232
6233
6234
6235
6236
6237 037010 013701 045030
6238 037014 010102
6239 037016 052702 000002
6240 037022 020102
6241 037024 001406
6242 037026 005237 002212
6246 037032
      037032 104456
      037034 000643
      037036 046003
      037040 015614
6247 037042          40$:   CKLOOP                    ;LOOP IF SELECTED
      037042 104406
6248 037044 012703 000400
6249 037050 013737 003116 045132
6250
6251
6252
6253
6254
6255
6256
6257 037056 012737 150005 045130
6258 037064 012704 045130
6259 037070
6260 037070 010300
6261 037072 004737 017542
6262 037076 010337 045136
6263 037102 010465 000000
6264 037106 004737 016370
6265 037112 016501 000002
6266 037116 012702 000200
6267 037122 020102
6268 037124 001406
6269 037126 005237 002212
6273 037132
      037132 104456
      037134 000644
      037136 005113
      037140 012166
6274 037142          75$:   CKLOOP                    ;LOOP IF SELECTED
      037142 104406
6275 037144 005723
      TST      (R3)+

```

```

;*****
;
;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$ERHRD
      .WORD  419
      .WORD  T24BOT
      .WORD  EXPREC
      TRAP    C$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
      INC      FATFLG          ;BUMP COUNT
      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      TRAP    C$ERHRD
      .WORD  420
      .WORD  WRTErr
      .WORD  PKTSSR
      TRAP    C$CLP1
      ;BUMP RECORD SIZE

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

6276 037146 022703 000414          CMP      #268.,R3          ;END OF RECORD YET
6277 037152 001346                    BNE      65$              ;BR, IF MORE RECORDS TO WRITE
6278 037154          104406          80$:   CKLOOP              ;LOOP IF SELECTED
6279 037156          104406                    TRAP      C$CLP1
6280
6281          ;*****
6282          ;
6283          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6284          ;
6285          ;*****
6286
6287 037156 004737 011136          JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
6288 037162 103407                    BCS      130$              ;BR, IF NO PROBLEM
6289 037164 010001                    MOV      R0,R1              ;SAVE TSSR
6290 037166 005237 002212          INC      FATFLG            ;BUMP COUNT
6294 037172          104456          ERRHRD  ERRNO,T24RWN,EXPREC ;REWIND NOT ACCEPTED
6295 037174 000645                    TRAP      C$ERHRD
6296 037176 046266                    .WORD    421
6297 037200 015614                    .WORD    T24RWN
6298 037202          104406          130$:   CKLOOP              ;LOOP IF SELECTED
6299 037202          104406                    TRAP      C$CLP1
6300
6301          ;*****
6302          ;
6303          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
6304          ;
6305          ;*****
6306
6307 037204 013701 045030          MOV      T24BFR+6,R1        ;PICK UP XST0
6308 037210 010102                    MOV      R1,R2              ;SET UP EXPECTED
6309 037212 052702 000002          BIS      #BIT1,R2          ;SET BOT BIT IN EXPECTED
6310 037216 020102                    CMP      R1,R2              ;DOES EXP = REC'D
6311 037220 001406                    BEQ      140$              ;BR, IF EQUAL (OK)
6312 037222 005237 002212          INC      FATFLG            ;BUMP COUNT
6313 037226          104456          ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
6314 037230 000646                    TRAP      C$ERHRD
6315 037232 046003                    .WORD    422
6316 037234 015614                    .WORD    T24BOT
6317 037236          104406          140$:   CKLOOP              ;LOOP IF SELECTED
6318 037236          104406                    TRAP      C$CLP1
6319 037240 012703 000400          MOV      #256.,R3          ;RECORD SIZE
6320 037244 013737 003116 045132    MOV      FREE,T24RB        ;STARTING READ BUFFER ADDRESS
6321
6322          ;*****
6323          ;
6324          ;READ DATA,IE,ACK,SWB COMMAND
6325          ;
6326          ;*****
6327
6328 037252 012737 110001 045130    165$:   MOV      #110001,T24PK3 ;READ DATA,IE,ACK,SWB COMMAND
6329 037260 012704 045130          MOV      #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
6330 037264 010337 045136          MOV      R3,T24SZ          ;SET UP RECORD SIZE IN PACKET
6331 037270 010465 000000          MOV      R4,TSDB(R5)      ;ISSUE COMMAND
6332 037274 004737 016370          JSR      PC,WAITF          ;WAIT FOR SSR TO SET

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

037556 046266 .WORD T24RWN
037560 012166 .WORD PKTSSR
6431 037562 104406 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
037562 104406
6432
6433 ;*****
6434 ;
6435 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6436 ;
6437 ;*****
6438
6439 037564 013701 045030 MOV T24BFR+6,R1 ;PICK UP XSTO
6440 037570 010102 MOV R1,R2 ;SET UP EXPECTED
6441 037572 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6442 037576 020102 CMP R1,R2 ;DOES EXP = REC'D
6443 037600 001406 BEQ 40$ ;BR, IF EQUAL (OK)
6444 037602 005237 002212 INC FATFLG ;BUMP COUNT
6448 037606 ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
037606 104456 TRAP C$ERHRD
037610 000654 .WORD 428
037612 046003 .WORD T24BOT
037614 015614 .WORD EXPREC
6449 037616 104406 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
037616 104406
6450 037620 012703 001000 MOV #512,R3 ;RECORD SIZE
6451 037624 013737 003116 045132 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
6452
6453 ;*****
6454 ;
6455 ;WRITE DATA,ACK,CVC=1 COMMAND
6456 ;
6457 ;*****
6458
6459 037632 012737 140005 045130 MOV #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
6460 037640 012704 045130 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6461 037644
6462 037644 010337 045136 65$: MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
6463 037650 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6464 037654 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
6465 037660 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6466 037664 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6467 037670 020102 CMP R1,R2 ;ARE THEY EQUAL
6468 037672 001406 BEQ 75$ ;BR, IF OK
6469 037674 005237 002212 INC FATFLG ;BUMP COUNT
6473 037700 ERRHRD ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
037700 104456 TRAP C$ERHRD
037702 000655 .WORD 429
037704 005113 .WORD WRTErr
037706 012166 .WORD PKTSSR
6474 037710 104406 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
037710 104406
6475 037712 120$:
6476
6477 ;*****
6478 ;
6479 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6480 ;

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

6481 ;*****
6482 ;
6483 037712 004737 011136 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6484 037716 103407 BCS 130$ ;BR, IF NO PROBLEM
6485 037720 010001 MOV R0,R1 ;SAVE TSSR
6486 037722 005237 002212 INC FATFLG ;BUMP COUNT
6490 037726 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
        037726 104456 TRAP C$ERHRD
        037730 000656 .WORD 430
        037732 046266 .WORD T24RWN
        037734 012166 .WORD PKTSSR
6491 037736 130$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
        037736 104406
6492 ;*****
6493 ;
6494 ;
6495 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
6496 ;
6497 ;
6498 ;*****
6499 037740 013701 045030 MOV T24BFR+6,R1 ;PICK UP XST0
6500 037744 010102 MOV R1,R2 ;SET UP EXPECTED
6501 037746 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6502 037752 020102 CMP R1,R2 ;DOES EXP = REC'D
6503 037754 001406 BEQ 140$ ;BR, IF EQUAL (OK)
6504 037756 005237 002212 INC FATFLG ;BUMP COUNT
6508 037762 ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
        037762 104456 TRAP C$ERHRD
        037764 000657 .WORD 431
        037766 046003 .WORD T24BOT
        037770 015614 .WORD EXPREC
6509 037772 140$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
        037772 104406
6510 037774 012703 000400 MOV #256,R3 ;RECORD SIZE
6511 040000 013737 003116 045132 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
6512 ;*****
6513 ;
6514 ;
6515 ;READ DATA,ACK,CVC=1 COMMAND
6516 ;
6517 ;
6518 ;*****
6519 040006 012737 140001 045130 MOV #140001,T24PK3 ;READ DATA,ACK,CVC=1 COMMAND
6520 040014 012704 045130 165$: MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6521 040020 010337 045136 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
6522 040024 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6523 040030 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
6524 040034 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6525 040040 012702 100204 MOV #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6526 040044 020102 CMP R1,R2 ;ARE THEY EQUAL
6527 040046 001406 BEQ 170$ ;BR, IF OK
6528 040050 005237 002212 INC FATFLG ;BUMP COUNT
6532 040054 ERRHRD ERRNO,T24TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
        040054 104456 TRAP C$ERHRD
        040056 000660 .WORD 432
        040060 047334 .WORD T24TRL
        040062 012166 .WORD PKTSSR

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

6584
6585 ;*****
6586 040154 004737 016114 JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
6587 040160 103407 BCS 20$ ;BR IF INIT WAS OK
6588 040162 005237 002212 INC FATFLG ;BUMP COUNT
6592 040166 010001 MOV RO,R1 ;CONTENTS OF TSSR REGISTER
6593 040170 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
        040170 104455 TRAP C$ERDF
        040172 000662 .WORD 434
        040174 003652 .WORD SFIERR
        040176 012154 .WORD SFIMSG
6594 040200
6595 040200 013737 002172 045020 20$: MOV UNITN,T24DSW ;SET DRIVE NUMBER IN PACKET
6596 040206 012704 045000 MOV #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
6597
6598 ;*****
6599 ;
6600 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
6601 ;
6602 ;*****
6603
6604 040212 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
6605 040216 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
6606 040220 005237 002212 INC FATFLG ;BUMP COUNT
6610 040224 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
6611 040226 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
        040226 104456 TRAP C$ERHRD
        040230 000663 .WORD 435
        040232 005056 .WORD WRTMSG
        040234 012154 .WORD SFIMSG
6612 040236 24$: CKLOOP ;LOOP IF SELECTED
        040236 104406 TRAP C$CLP1
6613
6614 ;*****
6615 ;
6616 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6617 ;
6618 ;*****
6619
6620 040240 004737 011136 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6621 040244 103407 BCS 30$ ;BR, IF NO PROBLEM
6622 040246 010001 MOV RO,R1 ;SAVE TSSR
6623 040250 005237 002212 INC FATFLG ;BUMP COUNT
6627 040254 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
        040254 104456 TRAP C$ERHRD
        040256 000664 .WORD 436
        040260 046266 .WORD T24RWN
        040262 012166 .WORD PKTSSR
6628 040264 30$: CKLOOP ;LOOP IF SELECTED
        040264 104406 TRAP C$CLP1
6629 040266 012703 000400 MOV #256.,R3 ;RECORD SIZE
6630 040272 013737 003116 045132 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
6631
6632 ;*****
6633 ;
6634 ;WRITE DATA,ACK,CVC=1 COMMAND
6635 ;

```

K12

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

6636 ;*****
6637
6638 040300 012737 140005 045130      MOV    #140005,T24PK3      ;WRITE DATA,ACK,CVC=1 COMMAND
6639 040306 012704 045130      MOV    #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
6640 040312      65$:
6641 040312 010337 045136      MOV    R3,T24SZ          ;SET UP RECORD SIZE IN PACKET
6642 040316 010465 000000      MOV    R4,TSDB(R5)       ;ISSUE COMMAND
6643 040322 004737 016370      JSR    PC,WAITF          ;WAIT FOR SSR TO SET
6644 040326 016501 000002      MOV    TSSR(R5),R1       ;GET TSSR CONTENTS
6645 040332 012702 000200      MOV    #SSR,R2           ;SET UP EXPECTED
6646 040336 020102      CMP    R1,R2             ;ARE THEY EQUAL
6647 040340 001406      BEQ    75$               ;BR, IF OK
6648 040342 005237 002212      INC    FATFLG            ;BUMP COUNT
6652 040346      ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
        040346 104456      TRAP  C$ERHRD
        040350 000665      .WORD 437
        040352 005113      .WORD WRERR
        040354 012166      .WORD PKTSSR
6653 040356      75$: CKLOOP              ;LOOP IF SELECTED      TRAP  C$CLP1
        040356 104406
6654 040360      120$:
6655
6656 ;*****
6657 ;
6658 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6659 ;
6660 ;*****
6661
6662 040360 004737 011136      JSR    PC,REWIND         ;CALL TAPE REWIND COMMAND
6663 040364 103407      BCS    130$              ;BR, IF NO PROBLEM
6664 040366 010001      MOV    R0,R1             ;SAVE TSSR
6665 040370 005237 002212      INC    FATFLG            ;BUMP COUNT
6669 040374      ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
        040374 104456      TRAP  C$ERHRD
        040376 000666      .WORD 438
        040400 046266      .WORD T24RWN
        040402 012166      .WORD PKTSSR
6670 040404      130$: CKLOOP              ;LOOP IF SELECTED      TRAP  C$CLP1
        040404 104406
6671 040406 012703 001000      MOV    #512,R3           ;RECORD SIZE
6672 040412 013737 003116 045132      MOV    FREE,T24RB        ;STARTING READ BUFFER ADDRESS
6673
6674 ;*****
6675 ;
6676 ;READ DATA,ACK,CVC=1 COMMAND
6677 ;
6678 ;*****
6679
6680 040420 012737 140001 045130      MOV    #140001,T24PK3    ;READ DATA,ACK,CVC=1 COMMAND
6681 040426 012704 045130      MOV    #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
6682 040432 010337 045136      MOV    R3,T24SZ          ;SET UP RECORD SIZE IN PACKET
6683 040436 010465 000000      MOV    R4,TSDB(R5)       ;ISSUE COMMAND
6684 040442 004737 016370      JSR    PC,WAITF          ;WAIT FOR SSR TO SET
6685 040446 016501 000002      MOV    TSSR(R5),R1       ;GET TSSR CONTENTS
6686 040452 012702 100204      MOV    #SSR!SC!BIT2,R2  ;SET UP EXPECTED
6687 040456 020102      CMP    R1,R2             ;ARE THEY EQUAL
6688 040460 001406      BEQ    170$              ;BR, IF OK

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

6791 040704 004737 011136      JSR    PC,REWIND          ;CALL TAPE REWIND COMMAND
6792 040710 103407             BCS    30$                ;BR, IF NO PROBLEM
6793 040712 010001             MOV    R0,R1              ;SAVE TSSR
6794 040714 005237 002212      INC    FATFLG             ;BUMP COUNT
6798 040720             ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD   444
                                .WORD   T24RWN
                                .WORD   PKTSSR
                                TRAP    C$CLP1
                                .WORD   445
                                .WORD   WRERR
                                .WORD   PKTSSR
                                TRAP    C$CLP1
6799 040730             30$:   CKLOOP            ;LOOP IF SELECTED
                                TRAP    C$CLP1
6800 040730 104406             MOV    #256.,R3           ;RECORD SIZE
6801 040732 012703 000400      MOV    FREE,T24RB        ;STARTING WRITE BUFFER ADDRESS
6801 040736 013737 003116 045132
6802
6803 ;*****
6804 ;
6805 ;WRITE DATA,ACK,CVC=1 COMMAND
6806 ;
6807 ;*****
6808
6809 040744 012737 140005 045130      MOV    #140005,T24PK3    ;WRITE DATA,ACK,CVC=1 COMMAND
6810 040752 012704 045130      MOV    #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
6811 040756             65$:   MOV    R3,R0          ;SET PATTERN IN CORRECT REGISTER
6812 040756 010300             JSR    PC,FILLMEM        ;FILL MEMORY WITH RECORD SIZE
6813 040760 004737 017542      MOV    R3,T24SZ          ;SET UP RECORD SIZE IN PACKET
6814 040764 010337 045136      MOV    R4,TSDB(R5)       ;ISSUE COMMAND
6815 040770 010465 000000      JSR    PC,WAITF          ;WAIT FOR SSR TO SET
6816 040774 004737 016370      MOV    TSSR(R5),R1       ;GET TSSR CONTENTS
6817 041000 016501 000002      MOV    #SSR,R2           ;SET UP EXPECTED
6818 041004 012702 000200      CMP    R1,R2             ;ARE THEY EQUAL
6819 041010 020102             BEQ    75$                ;BR, IF OK
6820 041012 001406             INC    FATFLG             ;BUMP COUNT
6821 041014 005237 002212      ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
6825 041020             TRAP    C$ERHRD
                                .WORD   445
                                .WORD   WRERR
                                .WORD   PKTSSR
                                TRAP    C$CLP1
6826 041030             75$:   CKLOOP            ;LOOP IF SELECTED
                                TRAP    C$CLP1
6827 041030 104406             TST    (R3)               ;BUMP RECORD SIZE
6828 041032 005723             CMP    #268.,R3          ;END OF RECORD YET
6829 041034 022703 000414      BNE    65$                ;BR, IF MORE RECORDS TO WRITE
6830 041042             80$:   CKLOOP            ;LOOP IF SELECTED
                                TRAP    C$CLP1
6831 041042 104406             TST    -(R3)              ;SET BACK TO 512.
6832 041044 005743             MOV    FREE,T24RB        ;STARTING READ BUFFER ADDRESS
6832 041046 013737 003116 045132
6833
6834 ;*****
6835 ;
6836 ;READ REVERSE DATA,ACK COMMAND
6837 ;
6838 ;*****
6839
6840 041054 012737 100401 045130      MOV    #100401,T24PK3    ;READ REVERSE DATA,ACK COMMAND
6841 041062 012704 045130      MOV    #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
6842 041066 010337 045136      MOV    R3,T24SZ          ;SET UP RECORD SIZE IN PACKET

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

041354 104456 TRAP C$ERHRD
041356 000702 .WORD 450
041360 046266 .WORD T24RWN
041362 012166 .WORD PKTSSR
6948 041364 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
041364 104406 ;RECORD SIZE
6949 041366 012703 000400 MOV #256.,R3 ;STARTING WRITE BUFFER ADDRESS
6950 041372 013737 003116 045132 MOV FREE,T24RB
6951
6952 ;*****
6953 ;
6954 ;WRITE DATA,ACK,CVC=1,SWB COMMAND
6955 ;
6956 ;*****
6957
6958 041400 012737 150005 045130 MOV #150005,T24PK3 ;WRITE DATA,ACK,CVC=1,SWB COMMAND
6959 041406 012704 045130 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6960 041412
6961 041412 010300 65$: MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
6962 041414 004737 017542 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
6963 041420 010337 045136 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
6964 041424 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6965 041430 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
6966 041434 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6967 041440 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6968 041444 020102 CMP R1,R2 ;ARE THEY EQUAL
6969 041446 001406 BEQ 75$ ;BR, IF OK
6970 041450 005237 002212 INC FATFLG ;BUMP COUNT
6974 041454 ERRHRD ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
041454 104456 TRAP C$ERHRD
041456 000703 .WORD 451
041460 005113 .WORD WRTERR
041462 012166 .WORD PKTSSR
6975 041464 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
041464 104406 ;RECORD SIZE
6976 041466 005723 TST (R3)+ ;BUMP RECORD SIZE
6977 041470 022703 000414 CMP #268.,R3 ;END OF RECORD YET
6978 041474 001346 BNE 65$ ;BR, IF MORE RECORDS TO WRITE
6979 041476 80$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
041476 104406 ;RECORD SIZE
6980 041500 005743 TST -(R3) ;SET RECORD SIZE BACK TO 512.
6981 041502 013737 003116 045132 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
6982
6983 ;*****
6984 ;
6985 ;READ REVERSE DATA,ACK,SWB COMMAND
6986 ;
6987 ;*****
6988
6989 041510 012737 110401 045130 MOV #110401,T24PK3 ;READ REVERSE DATA,ACK,SWB COMMAND
6990 041516 012704 045130 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6991 041522 010337 045136 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
6992 041526 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6993 041532 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
6994 041536 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6995 041542 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6996 041546 020102 CMP R1,R2 ;ARE THEY EQUAL

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

7045 041700 004737 047470      JSR      PC,T24REST      ;SET COMMAND PACKET
7046 041704 004737 047562      JSR      PC,T24RT2      ;SET UP OTHER COMMAND PACKET
7047
7048      ;*****
7049      ;
7050      ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
7051      ;
7052      ;*****
7053
7054 041710 004737 016114      JSR      PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
7055 041714 103407              BCS      20$             ;BR IF INIT WAS OK
7056 041716 005237 002212      INC      FATFLG          ;BUMP COUNT
7060 041722 010001              MOV      R0,R1           ;CONTENTS OF TSSR REGISTER
7061 041724              ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                TRAP      C$ERDF
                                .WORD    454
                                .WORD    SFIERR
                                .WORD    SFIMSG
7062 041734              20$:
7063 041734 013737 002172 045020 MOV      UNITN,T24DSW    ;SET DRIVE NUMBER IN PACKET
7064 041742 012704 045000      MOV      #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
7065
7066      ;*****
7067      ;
7068      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
7069      ;
7070      ;*****
7071
7072 041746 004737 010752      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
7073 041752 103407              BCS      24$             ;BR, IF COMMAND ISSUED OK
7074 041754 005237 002212      INC      FATFLG          ;BUMP COUNT
7078 041760 010001              MOV      R0,R1           ;SAVE CONTENTS OF TSSR
7079 041762              ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP      C$ERHRD
                                .WORD    455
                                .WORD    WRTMSG
                                .WORD    SFIMSG
7080 041772              24$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
7081 041772 104406
7082      ;*****
7083      ;
7084      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7085      ;
7086      ;*****
7087
7088 041774 004737 011136      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
7089 042000 004737 016456      JSR      PC,CHKTSSR     ;SEE HOW TSSR IS
7090 042004 103407              BCS      30$             ;BR, IF NO PROBLEM
7091 042006 010001              MOV      R0,R1           ;SAVE TSSR
7092 042010 005237 002212      INC      FATFLG          ;BUMP COUNT
7096 042014              ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    456
                                .WORD    T24RWN
                                .WORD    PKTSSR
7097 042024              30$:   CKLOOP          ;LOOP IF SELECTED

```

G13

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

7098 042024 104406                                TRAP    C$CLP1
7099 042026 012703 001000                        MOV     #512.,R3          ;RECORD SIZE
7100 042032 013737 003116 045132                 MOV     FREE,T24RB       ;STARTING WRITE BUFFER ADDRESS
7101                                     ;*****
7102                                     ;
7103                                     ;WRITE DATA,ACK,CVC=1 COMMAND
7104                                     ;
7105                                     ;*****
7106
7107 042040 012737 140005 045130                 MOV     #140005,T24PK3   ;WRITE DATA,ACK,CVC=1 COMMAND
7108 042046 012704 045130                        MOV     #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
7109 042052                                     65$:
7110 042052 010337 045136                        MOV     R3,T24SZ        ;SET UP RECORD SIZE IN PACKET
7111 042056 010465 000000                        MOV     R4,TSDB(R5)     ;ISSUE COMMAND
7112 042062 004737 016370                        JSR     PC,WAITF        ;WAIT FOR SSR TO SET
7113 042066 016501 000002                        MOV     TSSR(R5),R1    ;GET TSSR CONTENTS
7114 042072 012702 000200                        MOV     #SSR,R2        ;SET UP EXPECTED
7115 042076 020102                                CMP     R1,R2          ;ARE THEY EQUAL
7116 042100 001406                                BEQ     75$            ;BR, IF OK
7117 042102 005237 002212                        INC     FATFLG         ;BUMP COUNT
7121 042106                                ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
7122 042106 104456                                TRAP    C$ERHRD
7123 042110 000711                                .WORD  457
7124 042112 005113                                .WORD  WRTErr
7125 042114 012166                                .WORD  PKTSSR
7126
7127                                     75$:  CKLOOP          ;LOOP IF SELECTED
7128                                     TRAP    C$CLP1
7129
7130 042120 012703 000400                        MOV     #256.,R3        ;SIZE OF RECORD
7131 042124 013737 003116 045132                 MOV     FREE,T24RB     ;STARTING READ BUFFER ADDRESS
7132                                     ;*****
7133                                     ;
7134                                     ;READ DATA,ACK COMMAND
7135                                     ;
7136                                     ;*****
7137
7138 042132 012737 100401 045130                 MOV     #100401,T24PK3  ;READ DATA,ACK COMMAND
7139 042140 012704 045130                        MOV     #T24PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
7140 042144 010337 045136                        MOV     R3,T24SZ      ;SET UP RECORD SIZE IN PACKET
7141 042150 010465 000000                        MOV     R4,TSDB(R5)   ;ISSUE COMMAND
7142 042154 004737 016370                        JSR     PC,WAITF      ;WAIT FOR SSR TO SET
7143 042160 016501 000002                        MOV     TSSR(R5),R1  ;GET TSSR CONTENTS
7144 042164 012702 100204                        MOV     #SSR!SC!BIT2,R2 ;SET UP EXPECTED
7145 042170 020102                                CMP     R1,R2        ;ARE THEY EQUAL
7146 042172 001406                                BEQ     170$         ;BR, IF OK
7147 042174 005237 002212                        INC     FATFLG        ;BUMP COUNT
7148 042200                                ERRHRD  ERRNO,T24TRL,EXPREC ;TSSR INCORRECT AFTER READ DATA
7149 042200 104456                                TRAP    C$ERHRD
7150 042202 000712                                .WORD  458
7151 042204 047334                                .WORD  T24TRL
7152 042206 015614                                .WORD  EXPREC
7153
7154                                     170$:  CKLOOP          ;LOOP IF SELECTED
7155                                     TRAP    C$CLP1
7156
7157                                     ;*****
7158                                     ;
7159                                     ;
7160                                     ;

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

042334 003652
042336 012154
7202 042340
7203 042340 013737 002172 045020 20$: MOV UNITN,T24DSW ;SET DRIVE NUMBER IN PACKET
7204 042346 012704 045000 MOV #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
7205
7206 ;*****
7207 ;
7208 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
7209 ;
7210 ;*****
7211
7212 042352 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
7213 042356 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
7214 042360 005237 002212 INC FATFLG ;BUMP COUNT
7218 042364 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
7219 042366 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
042366 104456 TRAP C$ERHRD
042370 000715 .WORD 461
042372 005056 .WORD WRTMSG
042374 012154 .WORD SFIMSG
7220 042376 24$: CKLOOP ;LOOP IF SELECTED
042376 104406 TRAP C$CLP1
7221
7222 ;*****
7223 ;
7224 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7225 ;
7226 ;*****
7227
7228
7229 042400 005737 002216 TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
7230 042404 001024 BNE 27$ ;BR IF SWITCH IS ON
7231 042406 112737 000200 045141 MOVB #200,T24BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
7232 042414 112737 000010 045140 MOVB #10,T24BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
7233 042422 012704 045110 MOV #T24PK2,R4 ;WRITE SUBSYS MEM PACKET
7234 042426 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
7235 042432 004737 016456 JSR PC,CHKTSSR ;WAIT FOR SSK
7236 042436 103407 BCS 28$ ;BR, IF NO ERROR
7237 042440 010001 MOV R0,R1 ;ERROR, SAVE TSSR
7238 042442 005237 002212 INC FATFLG ;BUMP COUNT
7242 042446 ERRHRD ERRNO,T22SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
042446 104456 TRAP C$ERHRD
042450 000716 .WORD 462
042452 027230 .WORD T22SSR
042454 012166 .WORD PKTSSR
7243 042456 27$:
7244 042456 28$: CKLOOP ;LOOP IF SELECTED
042456 104406 TRAP C$CLP1
7245
7246
7247
7248 042460 004737 011136 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
7249 042464 004737 016456 JSR PC,CHKTSSR ;SEE HOW TSSR IS
7250 042470 103407 BCS 30$ ;BR, IF NO PROBLEM
7251 042472 010001 MOV R0,R1 ;SAVE TSSR
7252 042474 005237 002212 INC FATFLG ;BUMP COUNT

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

7256 042500          ERRHRD  ERRNO,T24RWN,PKTSSR  ;REWIND NOT ACCEPTED
      042500 104456          TRAP  C$ERHRD
      042502 000717          .WORD 463
      042504 046266          .WORD T24RWN
      042506 012166          .WORD PKTSSR
7257 042510          30$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      042510 104406
7258 042512 012703 000005          MOV  #5.,R3          ;NUMBER OF RECORDS
7259 042516 013737 003116 045132  MOV  FREE,T24RB      ;STARTING WRITE BUFFER ADDRESS
7260          ;*****
7261          ;
7262          ;WRITE DATA,ACK,CVC=1 COMMAND
7263          ;
7264          ;*****
7265
7266 042524 012737 140005 045130  MOV  #140005,T24PK3  ;WRITE DATA,ACK,CVC=1 COMMAND
7267 042532 012704 045130          MOV  #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
7268 042536          65$:
7269 042536 012737 000256 045136  MOV  #256,T24SZ      ;SET UP RECORD SIZE IN PACKET
7270 042544 010465 000000          MOV  R4,TSDB(R5)     ;ISSUE COMMAND
7271 042550 004737 016370          JSR  PC,WAITF        ;WAIT FOR SSR TO SET
7272 042554 016501 000002          MOV  TSSR(R5),R1     ;GET TSSR CONTENTS
7273 042560 012702 000200          MOV  #SSR,R2        ;SET UP EXPECTED
7274 042564 020102          CMP  R1,R2          ;ARE THEY EQUAL
7275 042566 001406          BEQ  75$            ;BR, IF OK
7276 042570 005237 002212          INC  FATFLG         ;BUMP COUNT
7280 042574          ERRHRD  ERRNO,WRTErr,PKTSSR  ;TSSR INCORRECT AFTER WRITE DATA
      042574 104456          TRAP  C$ERHRD
      042576 000720          .WORD 464
      042600 005113          .WORD WRTErr
      042602 012166          .WORD PKTSSR
7281 042604          75$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      042604 104406
7282 042606 005303          DEC  R3              ;BUMP DOWN RECORD COUNTER
7283 042610 001352          BNE  65$            ;BR, IF NOT AT 5 RECORDS YET
7284 042612 012703 000400          MOV  #256.,R3       ;RECORD SIZE
7285 042616 012701 160000          MOV  #160000,R1     ;NXM LOW ADDRESS START
7286 042622 012702 177776          MOV  #177776,R2     ;LIMIT CHECK FOR NXM (HIGHEST)
7287 042626 004737 016516          JSR  PC,NXM         ;LOOK FOR NXM ADDRESS
7288 042632 103046          BCC  180$          ;BR, IF NON FOUND
7289 042634 010137 003132          MOV  R1,NXML0       ;SET ADDRESS UP FOR TEST
7290 042640 013737 003132 045132  MOV  NXML0,T24RB     ;STARTING READ BUFFER ADDRESS
7291 042646 005037 045134          CLR  T24RB-2        ;SET TO 16 BIT ADDRESSING
7292          ;*****
7293          ;
7294          ;READ DATA,ACK COMMAND
7295          ;
7296          ;*****
7297
7298
7299 042652 012737 100001 045130  MOV  #100001,T24PK3  ;READ DATA,ACK COMMAND
7300 042660 012704 045130          MOV  #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
7301 042664 012737 000400 045136  MOV  #256.,T24SZ     ;SET UP RECORD SIZE IN PACKET
7302 042672 010465 000000          MOV  R4,TSDB(R5)     ;ISSUE COMMAND
7303 042676 004737 016370          JSR  PC,WAITF        ;WAIT FOR SSR TO SET
7304 042702 016501 000002          MOV  TSSR(R5),R1     ;GET TSSR CONTENTS
7305 042706 012702 104210          MOV  #SSR!NXM!SC!BIT3,R2 ;SET UP EXPECTED

```


L13

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

7356 043030 013737 002172 045020      MOV    UNITN,T24DSW      ;SET UP DRIVE NUMBER
7357 043036 012704 045000              MOV    #T24PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
7358
7359      ;*****
7360      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
7361      ;
7362      ;*****
7363
7364
7365 043042 004737 010752              JSR    PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
7366 043046 103407                      BCS    24$              ;BR, IF COMMAND ISSUED OK
7367 043050 005237 002212              INC    FATFLG           ;BUMP COUNT
7371 043054 010001                      MOV    R0,R1            ;SAVE CONTENTS OF TSSR
7372 043056                      ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP    C$ERHRD
                                .WORD   467
                                .WORD   WRTMSG
                                .WORD   SFIMSG
7373 043066                      24$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP    C$CLP1
7374 043070 013737 003116 045132      MOV    FREE,T24RB      ;STARTING WRITE BUFFER ADDRESS
7375
7376      ;*****
7377      ;ILLEGAL MODE DATA,ACK COMMAND
7378      ;
7379      ;*****
7380
7381
7382 043076 012737 104001 045130      MOV    #104001,T24PK3   ;ILLEGAL MODE DATA,ACK COMMAND
7383 043104 012704 045130              MOV    #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
7384 043110 012737 000400 045136      MOV    #256.,T24SZ     ;SET UP RECORD SIZE IN PACKET
7385 043116 010465 000000              MOV    R4,TSDB(R5)     ;ISSUE COMMAND
7386 043122 004737 016370              JSR    PC,WAITF        ;WAIT FOR SSR!BIT1!BIT2 TO SET
7387 043126 016501 000002              MOV    TSSR(R5),R1     ;GET TSSR CONTENTS
7388 043132 012702 100206              MOV    #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
7389 043136 020102                      CMP    R1,R2           ;ARE THEY EQUAL
7390 043140 001406                      BEQ    75$             ;BR, IF OK
7391 043142 005237 002212              INC    FATFLG           ;BUMP COUNT
7395 043146                      ERRHRD ERRNO,T24WDF,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP    C$ERHRD
                                .WORD   468
                                .WORD   T24WDF
                                .WORD   PKTSSR
7396 043156                      75$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP    C$CLP1
7397
7398      ;*****
7399      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7400      ;
7401      ;*****
7402
7403
7404 043160 013701 045030              MOV    T24BFR+6,R1     ;GET MESSAGE BUFFER
7405 043164 010102                      MOV    R1,R2           ;SET UP EXPECTED
7406 043166 052702 001000              BIS    #BIT9,R2        ;SET THE ILC BIT IN EXPECTED
7407 043172 020102                      CMP    R1,R2           ;ARE THEY EQUAL
7408 043174 001406                      BEQ    180$            ;BR, IF EQUAL (ALL IS WELL)

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

7459
7460
7461
7462 043306 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
7463 043312 103407 BCS 24; ;BR, IF COMMAND ISSUED OK
7464 043314 005237 002212 INC FATFLG ;BUMP COUNT
7468 043320 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
7469 043322 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
043322 104456 TRAP C#ERHRD
043324 000727 .WORD 471
043326 005056 .WORD WRTMSG
043330 012154 .WORD SFIMSG
7470 043332 24;: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
043332 104406 ;ILLEGAL STARTING READ BUFFER ADDRESS
7471 043334 013737 003116 045132 MOV FREE,T24RB ;CREATE ILLEGAL ADDRESS
7472 043342 012737 177700 045134 MOV #177700,T24RB+2
7473
7474 ;*****
7475 ;
7476 ;LEGAL MODE,ACK,CVC=1,READ COMMAND
7477 ;
7478 ;*****
7479
7480 043350 012737 140001 045130 MOV #140001,T24PK3 ;LEGAL MODE,ACK,CVC=1,READ COMMAND
7481 043356 012704 045130 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
7482 043362 012737 000400 045136 MOV #256.,T24SZ ;SET UP RECORD SIZE IN PACKET
7483 043370 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
7484 043374 004737 016370 JSR PC,WAITF ;WAIT FOR SSR!BIT1!BIT2 TO SET
7485 043400 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
7486 043404 012702 100206 MOV #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
7487 043410 020102 CMP R1,R2 ;ARE THEY EQUAL
7488 043412 001406 BEQ 75; ;BR, IF OK
7489 043414 005237 002212 INC FATFLG ;BUMP COUNT
7493 043420 ERRHRD ERRNO,T24WDG,PKTSSR ;TSSR INCORRECT AFTER READ DATA
043420 104456 TRAP C#ERHRD
043422 000730 .WORD 472
043424 045302 .WORD T24WDG
043426 012166 .WORD PKTSSR
7494 043430 75;: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
043430 104406
7495
7496 ;*****
7497 ;
7498 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7499 ;
7500 ;*****
7501
7502 043432 013701 045030 MOV T24BFR+6,R1 ;GET MESSAGE BUFFER
7503 043436 010102 MOV R1,R2 ;SET UP EXPECTED
7504 043440 052702 000400 BIS #BIT8,R2 ;SET THE ILA BIT IN EXPECTED
7505 043444 020102 CMP R1,R2 ;ARE THEY EQUAL
7506 043446 001406 BEQ 180; ;BR, IF EQUAL (ALL IS WELL)
7507 043450 005237 002212 INC FATFLG ;BUMP COUNT
7511 043454 ERRHRD ERRNO,T24ILA,EXPREC ;THE ILA BIT WAS NOT SET IN XSTO
043454 104456 TRAP C#ERHRD
043456 000731 .WORD 473
043460 045532 .WORD T24ILA
    
```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

7665
7666 044072 004737 011136      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
7667 044076 004737 016456      JSR      PC,CHKTSSR        ;SEE HOW TSSR IS
7668 044102 103407              BCS      30$               ;BR, IF NO PROBLEM
7669 044104 010001              MOV      R0,R1             ;SAVE TSSR
7670 044106 005237 002212      INC      FATFLG           ;BUMP COUNT
7674 044112              ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    479
                                .WORD    T24RWN
                                .WORD    PKTSSR
                                044112 104456
                                044114 000737
                                044116 046266
                                044120 012166
7675 044122              30$:   CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                044122 104406
7676 044124 012703 000400      MOV      #256,R3          ;RECORD SIZE
7677 044130 013737 003116 045132  MOV      FREE,T24RB       ;STARTING WRITE BUFFER ADDRESS
7678
7679 ;*****
7680 ;
7681 ;READ REVERSE DATA,ACK COMMAND
7682 ;
7683 ;*****
7684
7685 044136 012737 100401 045130      MOV      #100401,T24PK3   ;READ REVERSE DATA,ACK COMMAND
7686 044144 012704 045130      MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
7687 044150
7688 044150 010337 045136      65$:   MOV      R3,T24SZ    ;SET UP RECORD SIZE IN PACKET
7689 044154 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
7690 044160 004737 016370      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
7691 044164 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
7692 044170 012702 100206      MOV      #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
7693 044174 020102              CMP      R1,R2           ;ARE THEY EQUAL
7694 044176 001406              BEQ      75$             ;BR, IF OK
7695 044200 005237 002212      INC      FATFLG           ;BUMP COUNT
7699 044204              ERRHRD  ERRNO,T24WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    480
                                .WORD    T24WDE
                                .WORD    PKTSSR
                                044204 104456
                                044206 000740
                                044210 045731
                                044212 012166
7700 044214              75$:   CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                044214 104406
7701
7702 ;*****
7703 ;
7704 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7705 ;
7706 ;*****
7707
7708 044216 013701 045030      MOV      T24BFR+6,R1     ;GET MESSAGE BUFFER
7709 044222 010102              MOV      R1,R2           ;SET UP EXPECTED
7710 044224 052702 002000      BIS      #BIT10,R2       ;SET THE NEF BIT IN EXPECTED
7711 044230 020102              CMP      R1,R2           ;ARE THEY EQUAL
7712 044232 001406              BEQ      180$            ;BR, IF EQUAL (ALL IS WELL)
7713 044234 005237 002212      INC      FATFLG           ;BUMP COUNT
7717 044240              ERRHRD  ERRNO,T24NEF,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
                                TRAP      C$ERHRD
                                .WORD    481
                                .WORD    T24NEF
                                044240 104456
                                044242 000741
                                044244 045160

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

7768 044352 005237 002212      INC      FATFLG      ;BUMP COUNT
7772 044356 010001      MOV      RO,R1      ;SAVE CONTENTS OF TSSR
7773 044360      ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      044360 104456      TRAP      C$ERHRD
      044362 000743      .WORD    483
      044364 005056      .WORD    WRTMSG
      044366 012154      .WORD    SFIMSG
7774 044370      24$:   CKLOOP      ;LOOP IF SELECTED
      044370 104406      TRAP      C$CLP1
7775
7776      ;*****
7777      ;
7778      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7779      ;
7780      ;*****
7781
7782 044372 004737 011136      JSR      PC,REWIND  ;CALL TAPE REWIND COMMAND
7783 044376 004737 016456      JSR      PC,CHKTSSR ;SEE HOW TSSR IS
7784 044402 103407      BCS     30$        ;BR, IF NO PROBLEM
7785 044404 010001      MOV      RO,R1      ;SAVE TSSR
7786 044406 005237 002212      INC      FATFLG      ;BUMP COUNT
7790 044412      ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      044412 104456      TRAP      C$ERHRD
      044414 000744      .WORD    484
      044416 046266      .WORD    T24RWN
      044420 012166      .WORD    PKTSSR
7791 044422      30$:   CKLOOP      ;LOOP IF SELECTED
      044422 104406      TRAP      C$CLP1
7792 044424 012703 000400      MOV      #256.,R3   ;RECORD SIZE
7793 044430 013737 003116 045132      MOV      FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
7794
7795      ;*****
7796      ;
7797      ;WRITE DATA,ACK,CVC=1 COMMAND
7798      ;
7799      ;*****
7800
7801 044436 012737 140005 045130      MOV      #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
7802 044444 012704 045130      MOV      #T24PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
7803 044450      65$:
7804 044450 010337 045136      MOV      R3,T24SZ    ;SET UP RECORD SIZE IN PACKET
7805 044454 010465 000000      MOV      R4,TSDB(R5) ;ISSUE COMMAND
7806 044460 004737 016370      JSR      PC,WAITF    ;WAIT FOR SSR TO SET
7807 044464 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
7808 044470 012702 000200      MOV      #SSR,R2    ;SET UP EXPECTED
7809 044474 020102      CMP      R1,R2      ;ARE THEY EQUAL
7810 044476 001406      BEQ     75$        ;BR, IF OK
7811 044500 005237 002212      INC      FATFLG      ;BUMP COUNT
7815 044504      ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      044504 104456      TRAP      C$ERHRD
      044506 000745      .WORD    485
      044510 005113      .WORD    WRTERR
      044512 012166      .WORD    PKTSSR
7816 044514      75$:   CKLOOP      ;LOOP IF SELECTED
      044514 104406      TRAP      C$CLP1
7817 044516 012703 000400      MOV      #256.,R3   ;RECORD SIZE
7818 044522 013737 003116 045132      MOV      FREE,T24RB ;STARTING READ BUFFER ADDRESS

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

7819
7820 ;*****
7821 ;
7822 ;READ REVERSE DATA,ACK COMMAND
7823 ;
7824 ;*****
7825
7826 044530 012737 100401 045130      165$:  MOV     #100401,T24PK3      ;READ REVERSE DATA,ACK COMMAND
7827 044536 012704 045130      MOV     #T24PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
7828 044542 010337 045136      MOV     R3,T24SZ           ;SET UP RECORD SIZE IN PACKET
7829 044546 010465 000000      MOV     R4,TSDB(R5)        ;ISSUE COMMAND
7830 044552 004737 016370      JSR     PC,WAITF           ;WAIT FOR SSR TO SET
7831 044556 016501 000002      MOV     TSSR(R5),R1        ;GET TSSR CONTENTS
7832 044562 012702 000200      MOV     #SSR,R2           ;SET UP EXPECTED
7833 044566 020102      CMP     R1,R2             ;ARE THEY EQUAL
7834 044570 001406      BEQ     170$              ;BR, IF OK
7835 044572 005237 002212      INC     FATFLG            ;BUMP COUNT
7839 044576      ERRHRD  ERRNO,T24TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP     C$ERHRD
                                .WORD   486
                                .WORD   T24TRL
                                .WORD   PKTSSR
                                044576 104456
                                044600 000746
                                044602 047334
                                044604 012166
7840 044606      170$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP     C$CLP1
                                044606 104406
7841 044610 012703 000400      MOV     #256.,R3          ;RECORD SIZE
7842 044614 013737 003116 045132      MOV     FREE,T24RB        ;STARTING READ BUFFER ADDRESS
7843
7844 ;*****
7845 ;
7846 ;READ REVERSE DATA,ACK COMMAND
7847 ;
7848 ;*****
7849
7850 044622 012737 100401 045130      195$:  MOV     #100401,T24PK3      ;READ REVERSE DATA,ACK COMMAND
7851 044630 012704 045130      MOV     #T24PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
7852 044634 010337 045136      MOV     R3,T24SZ           ;SET UP RECORD SIZE IN PACKET
7853 044640 010465 000000      MOV     R4,TSDB(R5)        ;ISSUE COMMAND
7854 044644 004737 016370      JSR     PC,WAITF           ;WAIT FOR SSR TO SET
7855 044650 016501 000002      MOV     TSSR(R5),R1        ;GET TSSR CONTENTS
7856 044654 012702 100204      MOV     #SSR!SC!BIT2,R2    ;SET UP EXPECTED
7857 044660 020102      CMP     R1,R2             ;ARE THEY EQUAL
7858 044662 001406      BEQ     200$              ;BR, IF OK
7859 044664 005237 002212      INC     FATFLG            ;BUMP COUNT
7863 044670      ERRHRD  ERRNO,T24TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP     C$ERHRD
                                .WORD   487
                                .WORD   T24TRL
                                .WORD   PKTSSR
                                044670 104456
                                044672 000747
                                044674 047334
                                044676 012166
7864 044700      200$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP     C$CLP1
                                044700 104406
7865 044702 013701 045036      MOV     T24BFR+14,R1       ;GET MESSAGE BUFFER (XST3)
7866 044706 010102      MOV     R1,R2             ;SET UP EXPECTED
7867 044710 052702 000001      BIS     #BIT0,R2          ;SET THE RIB BIT IN EXPECTED
7868 044714 020102      CMP     R1,R2             ;ARE THEY EQUAL
7869 044716 001406      BEQ     210$              ;BR, IF EQUAL (ALL IS WELL)
7870 044720 005237 002212      INC     FATFLG            ;BUMP COUNT
7874 044724      ERRHRD  ERRNO,T24LOR,EXPREC ;THE RIB BIT WAS NOT SET IN XST0

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

7986      ;
7987      :-
7988
7989      T24REST:
7990      SAVREG      ;SAVE THE REGISTERS
7991      MOV #T24PACKET,R1 ;START OF THE PACKET
7992      MOV #100004,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
7993      MOV #T24DATA,(R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
7994      CLR (R1)+ ;EXTENDED ADDRESS
7995      MOV #10,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
7996      MOV #T24BFR,(R1)+ ;ADDRESS OF MESSAGE BUFFER
7997      CLR (R1)+
7998      MOV #20,(R1)+ ;LENGTH OF MESSAGE BUFFER
7999      CLR (R1)+
8000      MOV #0,(R1) ;SELECT DRIVE ZERO
8001      MOV #24,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
8002      MOV #177777,T24BFR(R2) ;ALL ONES TO MESSAGE BUFFER
8003      TST -(R2) ;NEXT LOCATION
8004      CMP #0,R2 ;CHECK FOR END OF LOOP
8005      BNE 64$ ;KEEP GOING UNTIL DONE
8006      RTS PC ;RETURN
8007
8008
8009      T24RT2:
8010      SAVREG      ;SAVE THE REGISTERS
8011      MOV #T24PK2,R1 ;START OF THE PACKET
8012      MOV #100206,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK, IE
8013      MOV #T24BF2,(R1)+ ;ADDRESS OF DATA BLOCK
8014      CLR (R1)+ ;EXTENDED ADDRESS
8015      MOV #6,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
8016      CLR (R1)+
8017      MOV #T24BF2,R1 ;POINT TO DATA SEL AREA
8018      CLR (R1)+
8019      CLR (R1)
8020      RTS PC ;RETURN
8021
8022      T24RT3:
8023      SAVREG      ;SAVE THE REGISTERS
8024      MOV #T24PK3,R1 ;START OF THE PACKET
8025      MOV #0,(R1)+ ;CLEAR AREA OUT
8026      MOV #0,(R1)+ ;ADDRESS OF DATA BLOCK
8027      CLR (R1)+ ;EXTENDED ADDRESS
8028      MOV #0,(R1) ;SIZE OF DATA BLOCK IN BYTES
8029      RTS PC ;RETURN
8030
8031      L10052: TRAP C$ETST
8032
8033      .SBTTL TEST 5: SPACE RECORDS
8034      ;*
8035      ;THIS TEST VERIFIES THAT THE SPACE RECORDS FORWARD AND SPACE
8036      ;RECORDS REVERSE POSITION COMMANDS OPERATE PROPERLY WHEN SPACING
8037      ;OVER NORMAL DATA RECORDS. OPERATION WHEN SPACING OVER TAPE MARKS
8038      ;IS VERIFIED IN A SUBSEQUENT TEST. THE BASIC WRITE DATA TEST
8039      ;SHOULD HAVE BEEN RUN SUCCESSFULLY FOR THIS TEST TO PRODUCE MEANINGFUL
8040      ;RESULTS. THIS TEST CONSISTS OF A SERIES OF SUBTESTS. IN EACH
      ;OF THE SUBTESTS, THE TAPE IS ENTIRELY WRITTEN WITH RECORDS

```


M14

TEST 5: SPACE RECORDS

```

8092 047772 001367          DEC      T25DLY          ;DEC DELAY COUNTER          BNE      .-20
8093 047774 005337 055232  BNE      5$          ;BR, IF LOOP IS REQUIRED
8094 050002 001356          INC      FATFLG      ;BUMP COUNT
8098 050006 005237 002212  MOV      TSSR(R5),R1 ;CONTENTS OF TSSR REGISTER
8099 050012 016501 000002  ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      050012 104455          TRAP                                C$ERDF
      050014 000765          .WORD                                501
      050016 003652          .WORD                                SFIERR
      050020 012154          .WORD                                SFIMSG
8100 050022
8101 050022 013737 002172 055070 10$:  MOV      UNITN,T25DSW      ;SET UP DRIVE NUMBER
8102 050030 012704 055050  MOV      #T25PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
8103
8104 ;*****
8105 ;
8106 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
8107 ;
8108 ;*****
8109
8110 050034 004737 010752  JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
8111 050040 103407          BCS      15$          ;BR, IF COMMAND ISSUED OK
8112 050042 005237 002212  INC      FATFLG      ;BUMP COUNT
8116 050046 010001          MOV      R0,R1        ;SAVE CONTENTS OF TSSR
8117 050050 010001          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      050050 104456          TRAP                                C$ERHRD
      050052 000766          .WORD                                502
      050054 005056          .WORD                                WRTMSG
      050056 012154          .WORD                                SFIMSG
8118
8119 ;*****
8120 ;
8121 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8122 ;
8123 ;*****
8124
8125 050060 15$:  CKLOOP
      050060 104406          TRAP                                C$CLP1
8126 050062 004737 011136  JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
8127 050066 103407          BCS      30$          ;BR, IF NO PROBLEM
8128 050070 010001          MOV      R0,R1        ;SAVE TSSR
8129 050072 005237 002212  INC      FATFLG      ;BUMP COUNT
8133 050076 010001          ERRHRD  ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      050076 104456          TRAP                                C$ERHRD
      050100 000767          .WORD                                503
      050102 056205          .WORD                                T25RWN
      050104 012166          .WORD                                PKTSSR
8134 050106 30$:  CKLOOP          ;LOOP IF SELECTED          TRAP      C$CLP1
      050106 104406
8135
8136 ;*****
8137 ;
8138 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8139 ;
8140 ;*****
8141
8142 050110 013701 055100  MOV      T25BFR+6,R1    ;PICK UP XSTO

```

TEST 5: SPACE RECORDS

```

8143 050114 010102          MOV      R1,R2          ;SET UP EXPECTED
8144 050116 052702 000002  BIS      #BIT1,R2       ;SET BOT BIT IN EXPECTED
8145 050122 020102          CMP      R1,R2          ;DOES EXP = REC'D
8146 050124 001406          BEQ      40$            ;BR, IF EQUAL (OK)
8147 050126 005237 002212  INC      FATFLG         ;BUMP COUNT
8151 050132          ERRHRD  ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      050132 104456          TRAP      C$ERHRD
      050134 000770          .WORD    504
      050136 055375          .WORD    T25BOT
      050140 015614          .WORD    EXPREC
8152 050142          40$:  CKLOOP          ;LOOP IF SELECTED
      050142 104406          TRAP      C$CLP1
8153 050144 012703 000400  MOV      #256.,R3       ;RECORD SIZE
8154 050150 013737 003116 055202  MOV      FREE,T25RB     ;STARTING WRITE BUFFER ADDRESS
8155
8156          ;*****
8157          ;
8158          ;WRITE DATA,ACK,CVC=1 COMMAND
8159          ;
8160          ;*****
8161
8162 050156 012737 140005 055200  MOV      #140005,T25PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
8163 050164 012704 055200          MOV      #T25PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
8164 050170          65$:  MOV      R3,T25SZ       ;SET UP RECORD SIZE IN PACKET
8165 050170 010337 055206          MOV      T25CNT,@FREE  ;LOAD UP RECORD COUNTER IN WRT BUFFER
8166 050174 013777 055230 132714  ADD      #1,T25CNT      ;GET READY FOR NEXT RECORD
8167 050202 062737 000001 055230  MOV      R4,TSDB(R5)    ;ISSUE COMMAND
8168 050210 010465 000000          JSR      PC,WAITF      ;WAIT FOR SSR TO SET
8169 050214 004737 016370          MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
8170 050220 016501 000002          MOV      #SSR,R2       ;SET UP EXPECTED
8171 050224 012702 000200          CMP      R1,R2         ;ARE THEY EQUAL
8172 050230 020102          BEQ      75$            ;BR, IF OK
8173 050232 001411          BIT      #BIT2,R1      ;CHECK FOR TAPE STATUS ALERT
8174 050234 032701 000004          BNE     120$           ;BR, IF TSA IS SET (SUSPECT IS EOT)
8175 050240 001014          INC      FATFLG         ;BUMP COUNT
8176 050242 005237 002212  ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
8180 050246          TRAP      C$ERHRD
      050246 104456          .WORD    505
      050250 000771          .WORD    WRERR
      050252 005113          .WORD    PKTSSR
      050254 012166
8181 050256          75$:  CKLOOP          ;LOOP IF SELECTED
      050256 104406          TRAP      C$CLP1
8182 050260 005203          INC      R3             ;BUMP RECORD SIZE
8183 050262 022703 001000  CMP      #512.,R3      ;END OF RECORD YET
8184 050266 001340          BNE     65$            ;BR, IF MORE RECORDS TO WRITE
8185 050270 000415          BR      125$           ;ENOUGH RECORDS
8186 050272
8187          120$:
8188          ;*****
8189          ;
8190          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
8191          ;
8192          ;*****
8193
8194 050272 013701 055100  MOV      T25BFR+6,R1    ;QUICK CHECK FOR EOT SET
8195 050276 010102          MOV      R1,R2         ;SET UP EXPECTED

```


TEST 5: SPACE RECORDS

```

8196 050300 052702 000001      BIS      #BIT0,R2      ;SET THE EOT BIT XSTO
8197 050304 020102      CMP      R1,R2      ;IS THE EOT BIT SET IN XSTO
8198 050306 001406      BEQ     125$      ;BR, IF SET (GOOD)
8199 050310 005237 002212      INC     FATFLG      ;BUMP COUNT
8203 050314      ERRDF  ERRNO,T25NET,EXPREC ;DEVICE FATAL NOT EOT FOUND ETC.
      050314 104455      TRAP   C$ERDF
      050316 000772      .WORD 506
      050320 055531      .WORD T25NET
      050322 015614      .WORD EXPREC
8204 050324
8205
8206
8207
8208
8209
8210
8211
8212 050324 004737 011136      JSR     PC,REWIND    ;CALL TAPE REWIND COMMAND
8213 050330 103407      BCS    130$      ;BR, IF NO PROBLEM
8214 050332 010001      MOV    R0,R1      ;SAVE TSSR
8215 050334 005237 002212      INC    FATFLG      ;BUMP COUNT
8219 050340      ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      050340 104456      TRAP   C$ERHRD
      050342 000773      .WORD 507
      050344 056205      .WORD T25RWN
      050346 012166      .WORD PKTSSR
8220 050350 104406      130$: CKLOOP      ;LOOP IF SELECTED      TRAP   C$CLP1
8221 050352 012737 000007 055070      MOV    #7,T25DSW   ;SET UP DRIVE NUMBER
8222 050360 012704 055050      MOV    #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
8223
8224
8225
8226
8227
8228
8229
8230 050364 004737 010752      JSR    PC,WRTCHR    ;ISSUE WRITE CHARACTERISTICS
8231 050370 103407      BCS    140$      ;BR, IF COMMAND ISSUED OK
8232 050372 005237 002212      INC    FATFLG      ;BUMP COUNT
8236 050376 010001      MOV    R0,R1      ;SAVE CONTENTS OF TSSR
8237 050400      ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      050400 104456      TRAP   C$ERHRD
      050402 000774      .WORD 508
      050404 005056      .WORD WRTMSG
      050406 012154      .WORD SFIMSG
8238 050410      140$: CKLOOP      ;SCOPE LOOP      TRAP   C$CLP1
      050410 104406
8239 050412 005737 002216      TST    EXTFEA      ;CHECK FOR EXTENDED FEATURES
8240 050416 001044      BNE    160$      ;BR IF SWITCH IS ON
8241
8242 050420 112737 000200 055211      MOV    #200,T25BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
8243 050426 112737 000010 055210      MOV    #10,T25BS0  ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
8244 050434 012704 055160      MOV    #T25PK2,R4 ;WRITE SUBSYS MEM PACKET
8245 050440 010465 000000      MOV    R4,TSDB(R5) ;ISSUE COMMAND
8246 050444 004737 016456      JSR    PC,CHKTSSR  ;WAIT FOR SSR
8247 050450 103407      BCS    150$      ;BR, IF NO ERROR

```

TEST 5: SPACE RECORDS

```

8248 050452 010001          MOV    R0,R1          ;ERROR, SAVE TSSR
8249 050454 005237 002212  INC    FATFLG        ;BUMP COUNT
8253 050460          ERRHRD  ERRNO,T25SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      050460 104456          TRAP   C4ERHRD
      050462 000775          .WORD  509
      050464 055234          .WORD  T25SSR
      050466 012166          .WORD  PKTSSR
8254 050470          150$:  CKLOOP          ;LOOP IF SELECTED
      050470 104406          TRAP   C4CLP1
8255 050472 012737 000007 055070  MOV    #7,T25DSW      ;SET UP DRIVE NUMBER
8256 050500 012704 055050  MOV    #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
8257
8258 ;*****
8259 ;
8260 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
8261 ;
8262 ;*****
8263
8264 050504 004737 010752          JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
8265 050510 103407          BCS    160$          ;BR, IF COMMAND ISSUED OK
8266 050512 005237 002212  INC    FATFLG        ;BUMP COUNT
8270 050516 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
8271 050520          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      050520 104456          TRAP   C4ERHRD
      050522 000776          .WORD  510
      050524 005056          .WORD  WRTMSG
      050526 012154          .WORD  SFIMSG
8272 050530          160$:  CKLOOP          ;SCOPE LOOP
      050530 104406          TRAP   C4CLP1
8273 050532 016501 000002  MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
8274 050536 032701 000100  BIT    #OFL,R1        ;CHECK FOR THE OFFLINE BIT SET
8275 050542 001006          BNE    170$          ;BR, IF OFFLINE (GOOD)
8276 050544 005237 002212  INC    FATFLG        ;BUMP COUNT
8280 050550          ERRDF  ERRNO,T25OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
      050550 104455          TRAP   C4ERDF
      050552 000777          .WORD  511
      050554 056254          .WORD  T25OFL
      050556 012154          .WORD  SFIMSG
8281 050560          170$:  CKLOOP          ;LOOP IF SELECTED
      050560 104406          TRAP   C4CLP1
8282
8283 ;*****
8284 ;
8285 ;SPACE FORWARD COMMAND IN PLACE
8286 ;
8287 ;*****
8288
8289 050562 012737 140010 055200 180$:  MOV    #140010,T25PK3 ;SPACE FORWARD COMMAND IN PLACE
8290 050570 012737 000001 055202  MOV    #1,T25RB      ;NUMBER OF RECORDS TO SPACE
8291 050576 012704 055200          MOV    #T25PK3,R4    ;R4 = POINTER TO PACKET
8292 050602 010465 000000          MOV    R4,TSD8(R5)   ;ISSUE COMMAND
8293 050606 004737 016370          JSR    PC,WAITF      ;WAIT FOR SSR TO SET
8294 050612 016501 000002          MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
8295 050616 012702 100306          MOV    #SSR!SC!OFL!BIT1!BIT2,R2 ;SET UP EXPECTED
8296 050622 020102          CMP    R1,R2         ;ARE THEY EQUAL
8297 050624 001406          BEQ    190$          ;BR, IF OK ESP. FUNCTION REJECT
8298 050626 005237 002212  INC    FATFLG        ;BUMP COUNT

```


TEST 5: SPACE RECORDS

```

8349
8350 050736 004737 010752          JSR    PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
8351 050742 103407                BCS    15$                ;BR, IF COMMAND ISSUED OK
8352 050744 005237 002212          INC    FATFLG            ;BUMP COUNT
8356 050750 010001                MOV    R0,R1             ;SAVE CONTENTS OF TSSR
8357 050752                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP    C$ERHRD
                                .WORD   514
                                .WORD   WRTMSG
                                .WORD   SFIMSG
050752 104456
050754 001002
050756 005056
050760 012154

8358
8359
8360
8361
8362
8363
8364
8365 050762                15$:   CKLOOP
                                TRAP    C$CLP1
                                .WORD   515
                                .WORD   T25RWN
                                .WORD   PKTSSR
050762 104406
8366 050764 004737 011136          JSR    PC,REWIND        ;CALL TAPE REWIND COMMAND
8367 050770 103407                BCS    30$                ;BR, IF NO PROBLEM
8368 050772 010001                MOV    R0,R1             ;SAVE TSSR
8369 050774 005237 002212          INC    FATFLG            ;BUMP COUNT
8373 051000                ERRHRD  ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD   515
                                .WORD   T25RWN
                                .WORD   PKTSSR
051000 104456
051002 001003
051004 056205
051006 012166

8374 051010                30$:   CKLOOP
                                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   516
                                .WORD   T25SSR
                                .WORD   PKTSSR
051010 104406
8375 051012 005737 002216          140$: TST    EXTFEA        ;CHECK FOR EXTENDED FEATURES SW SWITCH
8376 051016 001044                BNE    160$                ;BR IF SWITCH IS ON
8377
8378 051020 112737 000200 055211          MOV    #200,T25BS1      ;WRITE MISCELLANEOUS CONT/READ STATUS
8379 051026 112737 000010 055210          MOV    #10,T25BS0      ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
8380 051034 012704 055160          MOV    #T25PK2,R4      ;WRITE SUBSYS MEM PACKET
8381 051040 010465 000000          MOV    R4,T5DB(R5)     ;ISSUE COMMAND
8382 051044 004737 016456          JSR    PC,CHKTSSR      ;WAIT FOR SSR
8383 051050 103407                BCS    150$                ;BR, IF NO ERROR
8384 051052 010001                MOV    R0,R1             ;ERROR, SAVE TSSR
8385 051054 005237 002212          INC    FATFLG            ;BUMP COUNT
8389 051060                ERRHRD  ERRNO,T25SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
                                TRAP    C$ERHRD
                                .WORD   516
                                .WORD   T25SSR
                                .WORD   PKTSSR
051060 104456
051062 001004
051064 055234
051066 012166

8390 051070                150$:  CKLOOP
                                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   517
                                .WORD   T25DSW
                                .WORD   #T25PACKET,R4
051070 104406
8391 051072 012737 000007 055070          MOV    #7,T25DSW       ;SET UP DRIVE NUMBER
8392 051100 012704 055050          MOV    #T25PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
8393
8394
8395
8396
8397
8398
8399
;*****
;
;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
;
;*****

```


H15

TEST 5: SPACE RECORDS

```

8501 051364 004737 011136      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
8502 051370 103407              BCS      30$            ;BR, IF NO PROBLEM
8503 051372 010001              MOV      R0,R1         ;SAVE TSSR
8504 051374 005237 002212      INC      FATFLG        ;BUMP COUNT
8508 051400              ERRHRD   ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      051400 104456              TRAP     C$ERHRD
      051402 001012              .WORD   522
      051404 056205              .WORD   T25RWN
      051406 012166              .WORD   PKTSSR
8509 051410              30$:    CKLOOP        ;LOOP IF SELECTED
      051410 104406              TRAP     C$CLP1

8510
8511      ;*****
8512      ;
8513      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8514      ;
8515      ;*****
8516
8517 051412 013701 055100      MOV      T25BFR+6,R1   ;PICK UP XSTO
8518 051416 010102              MOV      R1,R2         ;SET UP EXPECTED
8519 051420 052702 000002      BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
8520 051424 020102              CMP      R1,R2         ;DOES EXP = REC'D
8521 051426 001406              BEQ      40$            ;BR, IF EQUAL (OK)
8522 051430 005237 002212      INC      FATFLG        ;BUMP COUNT
8526 051434              ERRHRD   ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      051434 104456              TRAP     C$ERHRD
      051436 001013              .WORD   523
      051440 055375              .WORD   T25BOT
      051442 015614              .WORD   EXPREC
8527 051444              40$:    CKLOOP        ;LOOP IF SELECTED
      051444 104406              TRAP     C$CLP1
8528 051446 012737 000001 055202 MOV      #000001,T25RB ;NUMBER OF RECORDS TO SPACE OVER
8529
8530      ;*****
8531      ;
8532      ;SPACE FORWARD,ACK,CVC=1 COMMAND
8533      ;
8534      ;*****
8535
8536 051454 012737 140010 055200 MOV      #140010,T25PK3 ;SPACE FORWARD,ACK,CVC=1 COMMAND
8537 051462 012704 055200      MOV      #T25PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
8538 051466              65$:
8539 051466 010465 000000      MOV      R4,TSDB(R5)  ;ISSUE COMMAND
8540 051472 004737 016370      JSR      PC,WAITF     ;WAIT FOR SSR TO SET
8541 051476 016501 000002      MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
8542 051502 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
8543 051506 020102              CMP      R1,R2         ;ARE THEY EQUAL
8544 051510 001411              BEQ      75$            ;BR, IF OK
8545 051512 032701 000004      BIT      #BIT2,R1     ;CHECK FOR TAPE STATUS ALERT
8546 051516 001006              BNE     75$            ;BR, IF TSA IS SET (SUSPECT IS EOT)
8547 051520 005237 002212      INC      FATFLG        ;BUMP COUNT
8551 051524              ERRHRD   ERRNO,T25WDE,EXPREC ;TSSR INCORRECT AFTER READ DATA
      051524 104456              TRAP     C$ERHRD
      051526 001014              .WORD   524
      051530 055315              .WORD   T25WDE
      051532 015614              .WORD   EXPREC
8552 051534              75$:    CKLOOP        ;LOOP IF SELECTED

```

TEST 5: SPACE RECORDS

```

8553 051534 104406                                TRAP    C$CLP1
8554 051536                                120$:
8555 ;*****
8556 ;
8557 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8558 ;
8559 ;*****
8560
8561 051536 013701 055100                        MOV     T25BFR+6,R1      ;QUICK CHECK FOR BOT SET
8562 051542 010102                                MOV     R1,R2           ;SET UP EXPECTED
8563 051544 042702 000002                        BIC     #BIT1,R2        ;CLEAR THE BOT BIT (XSTO)
8564 051550 020102                                CMP     R1,R2           ;IS THE EOT BIT SET IN XSTO
8565 051552 001406                                BEQ     125$            ;BR, IF SET (GOOD)
8566 051554 005237 002212                        INC     FATFLG          ;BUMP COUNT
8570 051560                                ERRHRD  ERRNO,T25BNC,EXPREC ;BOT NOT CLEARED AFTER SPACE FROM BOT
                                TRAP    C$ERHRD
                                .WORD   525
                                .WORD   T25BNC
                                .WORD   EXPREC
8571 051570                                125$:  CKLOOP
8572 051572 004737 056552                        JSR     PC,T25RT3       ;CLEAN UP PACKET
8573 051576 012737 000401 055206                MOV     #257.,T25SZ     ;SET THE CORRECT SIZE UP
                                TRAP    C$CLP1
8574 ;*****
8575 ;READ DATA COMMAND IN PLACE
8576 ;
8577 ;*****
8578
8579
8580
8581 051604 012737 140001 055200                MOV     #140001,T25PK3  ;READ DATA COMMAND IN PLACE
8582 051612 013737 003116 055202                MOV     FREE,T25RB      ;READ BUFFER ADDRESS TO PACKET
8583 051620 012704 055200                        MOV     #T25PK3,R4      ;R4 = POINTER TO PACKET
8584 051624 010465 000000                        MOV     R4,TSDB(R5)     ;ISSUE COMMAND
8585 051630 004737 016370                        JSR     PC,WAITF        ;WAIT FOR SSR TO SET
8586 051634 016501 000002                        MOV     TSSR(R5),R1     ;GET TSSR CONTENTS
8587 051640 012702 000200                        MOV     #SSR,R2        ;SET UP EXPECTED
8588 051644 020102                                CMP     R1,R2           ;ARE THEY EQUAL
8589 051646 001406                                BEQ     190$            ;BR, IF OK ESP. FUNCTION REJECT
8590 051650 005237 002212                        INC     FATFLG          ;BUMP COUNT
8594 051654                                ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP    C$ERHRD
                                .WORD   526
                                .WORD   RDERR
                                .WORD   PKTSSR
8595 051664                                190$:  CKLOOP
                                ;LOOP IF SELECTED
                                TRAP    C$CLP1
8596 051666 017701 131224                        MOV     @FREE,R1        ;GET FIRST WORD FROM BUFFER
8597 051672 012702 000001                        MOV     #1,R2          ;SET UP EXPECTED
8598 051676 020102                                CMP     R1,R2           ;WAS RECORD NUMBERED 1
8599 051700 001406                                BEQ     200$            ;BR, IF CORRECT RECORD
8600 051702 005237 002212                        INC     FATFLG          ;BUMP COUNT
8604 051706                                ERRHRD  ERRNO,T25WNG,EXPREC ;SHOULD HAVE BEEN RECORD NUMBER 1
                                TRAP    C$ERHRD
                                .WORD   527
                                .WORD   T25WNG

```


TEST 5: SPACE RECORDS

```

8658 052026          ERRHRD  ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTISC FAILED
      052026 104456          TRAP                   C$ERHRD
      052030 001021          .WORD                   529
      052032 005056          .WORD                   WRTMSG
      052034 012154          .WORD                   SFIMSG
8659 052036          25$:   CKLOOP                    ;LOOP IF SELECTED
      052036 104406          TRAP                   C$CLP1
8660
8661 ;*****
8662 ;
8663 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8664 ;
8665 ;*****
8666
8667 052040 004737 011136      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
8668 052044 103407          BCS      30$                ;BR, IF NO PROBLEM
8669 052046 010001          MOV      R0,R1              ;SAVE TSSR
8670 052050 005237 002212      INC      FATFLG            ;BUMP COUNT
8674 052054          ERRHRD  ERRNO,T25RWN,PKTSSR      ;REWIND NOT ACCEPTED
      052054 104456          TRAP                   C$ERHRD
      052056 001022          .WORD                   530
      052060 056205          .WORD                   T25RWN
      052062 012166          .WORD                   PKTSSR
8675 052064          30$:   CKLOOP                    ;LOOP IF SELECTED
      052064 104406          TRAP                   C$CLP1
8676
8677 ;*****
8678 ;
8679 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8680 ;
8681 ;*****
8682
8683 052066 013701 055100      MOV      T25BFR+6,R1       ;PICK UP XSTO
8684 052072 010102          MOV      R1,R2              ;SET UP EXPECTED
8685 052074 052702 000002      BIS      #BIT1,R2         ;SET BOT BIT IN EXPECTED
8686 052100 020102          CMP      R1,R2              ;DOES EXP = REC'D
8687 052102 001406          BEQ      40$                ;BR, IF EQUAL (OK)
8688 052104 005237 002212      INC      FATFLG            ;BUMP COUNT
8692 052110          ERRHRD  ERRNO,T25BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      052110 104456          TRAP                   C$ERHRD
      052112 001023          .WORD                   531
      052114 055375          .WORD                   T25BOT
      052116 015614          .WORD                   EXPREC
8693 052120          40$:   CKLOOP                    ;LOOP IF SELECTED
      052120 104406          TRAP                   C$CLP1
8694
8695 ;*****
8696 ;
8697 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
8698 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
8699 ;
8700 ;*****
8701
8702 052122 012703 000001      MOV      #000001,R3       ;NUMBER OF RECORDS TO SPACE FORWARD
8703 052126 004737 010560      JSR      PC,SPACE         ;CALL SPACE COMMAND
8704 052132 103410          BCS      50$                ;CHECK FOR ERROR
8705 052134 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS

```

L15

TEST 5: SPACE RECORDS

```

8706 052140 005237 002212          INC    FATFLG          ;BUMP COUNT
8710 052144          ERRHRD  ERRNO,T2SWDE,SFFMSG ;SPACE FORWARD FAILED
      052144 104456          TRAP    C$ERHRD
      052146 001024          .WORD  532
      052150 055315          .WORD  T2SWDE
      052152 012222          .WORD  SFFMSG
8711 052154          50$:   CKLOOP          ;LOOP IF SELECTED
      052154 104406          TRAP    C$CLP1
8712 052156 012737 000001 055202    MOV    #1,T25RB        ;NUMBER OF RECORDS TO SPACE OVER
8713
8714          ;*****
8715          ;
8716          ;SPACE REVERSE,ACK,CVC=1 COMMAND
8717          ;
8718          ;*****
8719
8720 052164 012737 140410 055200    MOV    #140410,T25PK3  ;SPACE REVERSE,ACK,CVC=1 COMMAND
8721 052172 012704 055200          MOV    #T25PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
8722 052176          65$:
8723 052176 010465 000000          MOV    R4,TSDB(R5)     ;ISSUE COMMAND
8724 052202 004737 016370          JSR    PC,WAITF        ;WAIT FOR SSR TO SET
8725 052206 016501 000002          MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
8726 052212 012702 000200          MOV    #SSR,R2        ;SET UP EXPECTED
8727 052216 020102          CMP    R1,R2          ;ARE THEY EQUAL
8728 052220 001406          BEQ    75$            ;BR, IF OK
8729 052222 005237 002212          INC    FATFLG          ;BUMP COUNT
8733 052226          ERRHRD  ERRNO,T2SWDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      052226 104456          TRAP    C$ERHRD
      052230 001025          .WORD  533
      052232 055315          .WORD  T2SWDE
      052234 012166          .WORD  PKTSSR
8734 052236          75$:   CKLOOP          ;LOOP IF SELECTED
      052236 104406          TRAP    C$CLP1
8735 052240          120$:
8736 052240 012703 000400          MOV    #256.,R3       ;RECORD SIZE
8737 052244 013737 003116 055202    MOV    FREE,T25RB     ;STARTING READ BUFFER ADDRESS
8738
8739          ;*****
8740          ;
8741          ;READ DATA,ACK,CVC=1 COMMAND
8742          ;
8743          ;*****
8744
8745 052252 012737 140001 055200    MOV    #140001,T25PK3  ;READ DATA,ACK,CVC=1 COMMAND
8746 052260 012704 055200          MOV    #T25PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
8747 052264 010337 055206          MOV    R3,T25SZ        ;SET UP RECORD SIZE IN PACKET
8748 052270 010465 000000          MOV    R4,TSDB(R5)     ;ISSUE COMMAND
8749 052274 004737 016370          JSR    PC,WAITF        ;WAIT FOR SSR TO SET
8750 052300 016501 000002          MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
8751 052304 012702 000200          MOV    #SSR,R2        ;SET UP EXPECTED
8752 052310 020102          CMP    R1,R2          ;ARE THEY EQUAL
8753 052312 001406          BEQ    170$           ;BR, IF OK
8754 052314 005237 002212          INC    FATFLG          ;BUMP COUNT
8758 052320          ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      052320 104456          TRAP    C$ERHRD
      052322 001026          .WORD  534
      052324 005206          .WORD  RDERR

```


TEST 5: SPACE RECORDS

```

052456 013727 002116                                MOV     L$DLY,(PC)+
052462 000000                                .WORD  0
052464 005367 177772                                DEC     -6(PC)
052470 001375                                BNE     -4
052472 005367 177756                                DEC     -22(PC)
052476 001367                                BNE     -20
8806 052500 005337 055232          DEC     T25DLY          ;DEC COUNTER
8807 052504 001356          BNE     10$            ;BR, IF MORE LOOPS REQUIRED
8808 052506 016501 000002          MOV     TSSR(R5),R1    ;CONTENTS OF TSSR REGISTER
8809 052512 005237 002212          INC     FATFLG          ;BUMP COUNT
8813 052516          ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                TRAP   C$ERDF
                                .WORD  536
                                .WORD  SFIERR
                                .WORD  SFIMSG
052516 104455
052520 001030
052522 003652
052524 012154
8814 052526
8815 052526 013737 002172 055070 20$:  MOV     UNITN,T25DSW    ;SET UP UNIT NUMBER
8816 052534 012704 055050          MOV     @T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
8817
8818          ;*****
8819          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
8820          ;
8821          ;*****
8822
8823
8824 052540 004737 010752          JSR     PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
8825 052544 103407          BCS     25$            ;BR, IF COMMAND ISSUED OK
8826 052546 005237 002212          INC     FATFLG          ;BUMP COUNT
8830 052552 010001          MOV     R0,R1          ;SAVE CONTENTS OF TSSR
8831 052554          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP   C$ERHRD
                                .WORD  537
                                .WORD  WRTMSG
                                .WORD  SFIMSG
052554 104456
052556 001031
052560 005056
052562 012154
8832 052564          CKLOOP          ;LOOP IF SELECTED
052564 104406                                TRAP   C$CLP1
8833
8834          ;*****
8835          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8836          ;
8837          ;*****
8838
8839
8840 052566 004737 011136          JSR     PC,REWIND      ;CALL TAPE REWIND COMMAND
8841 052572 103407          BCS     30$            ;BR, IF NO PROBLEM
8842 052574 010001          MOV     R0,R1          ;SAVE TSSR
8843 052576 005237 002212          INC     FATFLG          ;BUMP COUNT
8847 052602          ERRHRD  ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP   C$ERHRD
                                .WORD  538
                                .WORD  T25RWN
                                .WORD  PKTSSR
052602 104456
052604 001032
052606 056205
052610 012166
8848 052612          CKLOOP          ;LOOP IF SELECTED
052612 104406                                TRAP   C$CLP1
8849 052614 013701 055226          MOV     T25CN2,R1      ;NUMBER OF RECORDS ON TAPE
8850 052620 012702 177776          MOV     @65534.,R2    ;MAX IT CAN SPACE OVER
8851 052624 020201          CMP     R2,R1          ;WHICH VALUE CAN WE USE

```

TEST 5: SPACE RECORDS

```

8852 052626 003002          BGT      46$          ;BR, IF # WRITTEN > 64K
8853 052630 010103          MOV      R1,R3       ;# WRITTEN CAN BE USED
8854 052632 000401          BR       47$         ;MOVE ON
8855 052634 010203          46$:    MOV      R2,R3       ;USE MAX NUMBER
8856 052636 162703 000001  47$:    SUB      #1,R3      ;DON'T GO ALL THE WAY YET
8857 052642 010337 055202          MOV      R3,T25RB    ;NUMBER OF RECORDS TO SPACE OVER
8858
8859          ;*****
8860          ;
8861          ;SPACE FORWARD,ACK,CVC=1 COMMAND
8862          ;
8863          ;*****
8864
8865 052646 012737 140010 055200          MOV      #140010,T25PK3 ;SPACE FORWARD,ACK,CVC=1 COMMAND
8866 052654 012704 055200          MOV      #T25PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
8867 052660
8868 052660 013737 055226 055232 65$:    MOV      T25CN2,T25DLY ;NUMBER OF RECORDS USED AS DELAY COUNTER
8869 052666 010465 000000          MOV      R4,TSDB(R5)  ;ISSUE COMMAND
8870 052672 004737 016370          67$:    JSR      PC,WAITF    ;WAIT FOR SSR TO SET
8871 052676 016501 000002          MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
8872 052702 012702 000200          MOV      #SSR,R2     ;SET UP EXPECTED
8873 052706 020102          CMP      R1,R2       ;ARE THEY EQUAL
8874 052710 001425          BEQ     75$         ;BR, IF OK
8875 052712          DELAY   250        ;DELAY .25 SECONDS
      MOV      #250,(PC)+
      .WORD   0
      MOV      L$DLY,(PC)+
      .WORD   0
      DEC     -6(PC)
      BNE     -4
      DEC     -22(PC)
      BNE     -20
8876 052742 005337 055232          DEC     T25DLY       ;BUMP DOWN COUNTER
8877 052746 001351          BNE     67$         ;BR, IF NOT AT END OF DELAY
8878 052750 005237 002212          INC     FATFLG      ;BUMP COUNT
8882 052754          ERRHRD  ERRNO,T25WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      TRAP   C$ERHRD
      .WORD  539
      .WORD  T25WDE
      .WORD  PKTSSR
      8883 052764          75$:    CKLOOP          ;LOOP IF SELECTED
      TRAP   C$CLP1
8884 052766 012703 010000          MOV      #4096,R3    ;RECORD SIZE
8885 052772 013737 003116 055202          MOV      FREE,T25RB  ;STARTING READ BUFFER ADDRESS
8886
8887          ;*****
8888          ;
8889          ;READ DATA,ACK COMMAND
8890          ;
8891          ;*****
8892
8893 053000 012737 100001 055200          MOV      #100001,T25PK3 ;READ DATA,ACK COMMAND
8894 053006 012704 055200          165$:  MOV      #T25PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
8895 053012 010337 055206          MOV      R3,T25SZ    ;SET UP RECORD SIZE IN PACKET
8896 053016 010465 000000          MOV      R4,TSDB(R5) ;ISSUE COMMAND
8897 053022 004737 016370          JSR      PC,WAITF    ;WAIT FOR SSR TO SET
8898 053026 016501 000002          MOV      TSSR(R5),R1 ;GET TSSR CONTENTS

```


TEST 5: SPACE RECORDS

```

8948 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
8949 ;
8950 ;*****
8951 ;
8952 053174 004737 016114 10$: JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
8953 053200 103427 BCS 20$ ;BR IF INIT WAS OK
8954 053202 DELAY 250 ;WAIT ABOUT .25 SECONDS
      053202 012727 000250 MOV #250,(PC)+
      053206 000000 .WORD 0
      053210 013727 002116 MOV L$DLY,(PC)+
      053214 000000 .WORD 0
      053216 005367 177772 DEC -6(PC)
      053222 001375 BNE -.4
      053224 005367 177756 DEC -22(PC)
      053230 001367 BNE .-20
8955 053232 005337 055232 DEC T25DLY ;DEC COUNTER
8956 053236 001356 BNE 10$ ;BR, IF MORE LOOPS REQUIRED
8957 053240 016501 000002 MOV TSSR(R5),R1 ;CONTENTS OF TSSR REGISTER
8958 053244 005237 002212 INC FATFLG ;BUMP COUNT
8962 053250 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      053250 104455 TRAP C$ERDF
      053252 001036 .WORD 542
      053254 003652 .WORD SFIERR
      053256 012154 .WORD SFIMSG
8963 053260 013737 002172 055070 20$: MOV UNITN,T25DSW ;SET UP UNIT NUMBER
8964
8965 053266 012704 055050 MOV #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
8966
8967 ;*****
8968 ;
8969 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
8970 ;
8971 ;*****
8972 ;
8973 053272 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
8974 053276 103407 BCS 25$ ;BR, IF COMMAND ISSUED OK
8975 053300 005237 002212 INC FATFLG ;BUMP COUNT
8979 053304 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
8980 053306 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
      053306 104456 TRAP C$ERHRD
      053310 001037 .WORD 543
      053312 005056 .WORD WRTMSG
      053314 012154 .WORD SFIMSG
8981 053316 25$: CKLOOP ;LOOP IF SELECTED
      053316 104406 TRAP C$CLP1
8982
8983 ;*****
8984 ;
8985 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8986 ;
8987 ;*****
8988 ;
8989 053320 004737 011136 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
8990 053324 103407 BCS 30$ ;BR, IF NO PROBLEM
8991 053326 010001 MOV R0,R1 ;SAVE TSSR
8992 053330 005237 002212 INC FATFLG ;BUMP COUNT
8996 053334 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED

```


TEST 5: SPACE RECORDS

```

053334 104456
053336 001040
053340 056205
053342 012166
8997 053344 104406
8998
8999
9000
9001
9002
9003
9004
9005 053346 013701 055100
9006 053352 010102
9007 053354 052702 000002
9008 053360 020102
9009 053362 001406
9010 053364 005237 002212
9014 053370
053370 104456
053372 001041
053374 055375
053376 015614
9015 053400
053400 104406
9016 053402 013701 055226
9017 053406 012702 177776
9018 053412 020201
9019 053414 003002
9020 053416 010103
9021 053420 000401
9022 053422 010203
9023 053424
9024 053424 010337 055202
9025
9026
9027
9028
9029
9030
9031
9032 053430 012737 140010 055200
9033 053436 012704 055200
9034 053442 010465 000000
9035 053446 013737 055226 055232
9036 053454 004737 016370
9037 053460 016501 000002
9038 053464 012702 000200
9039 053470 020102
9040 053472 001425
9041 053474
053474 012727 000250
053500 000000
053502 013727 002116
053506 000000
053510 005367 177772

```

```

30$: CKLOOP ;LOOP IF SELECTED TRAP C$ERHRD
;*****
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;*****
MOV T25BFR-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,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
TRAP C$ERHRD
;*****
40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
MOV T25CN2,R1 ;NUMBER OF RECORDS ON TAPE
MOV #65534.,R2 ;MAX IT CAN SPACE OVER
CMP R2,R1 ;WHICH VALUE CAN WE USE
BGT 46$ ;BR, IF # WRITTEN > 64K
MOV R1,R3 ;# WRITTEN CAN BE USED
BR 47$ ;MOVE ON
46$: MOV R2,R3 ;USE MAX NUMBER
47$: MOV R3,T25RB ;NUMBER OF RECORDS TO SPACE OVER
;*****
;SPACE FORWARD,ACK,CVC=1 COMMAND
;*****
MOV #140010,T25PK3 ;SPACE FORWARD,ACK,CVC=1 COMMAND
MOV #T25PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
MOV R4,TSDB(R5) ;ISSUE COMMAND
MOV T25CN2,T25DLY ;SET UP DELAY COUNTER
48$: JSR PC,WAITF ;WAIT FOR SSR TO SET
MOV TSSR(R5),R1 ;GET TSSR CONTENTS
MOV #SSR,R2 ;SET UP EXPECTED
CMP R1,R2 ;ARE THEY EQUAL
BEQ 50$ ;BR, IF OK
DELAY 250 ;WAIT .25 SECONDS
MOV #250,(PC)+
;*****
;*****
;*****
MOV 0
MOV L$DLY,(PC)+
;*****
MOV 0
DEC -6(PC)

```

TEST 5: SPACE RECORDS

```

053514 001375
053516 005367 177756
053522 001367
9042 053524 005337 055232
9043 053530 001351
9044 053532 005237 002212
9048 053536
053536 104456
053540 001042
053542 055315
053544 015614
9049 053546
053546 104406
9050 053550 013701 055226
9051 053554 012702 177776
9052 053560 020201
9053 053562 003002
9054 053564 010103
9055 053566 000401
9056 053570 010203
9057 053572 162703 000001
9058 053576 010337 055202
9059
9060
9061
9062
9063
9064
9065
9065 053602 012737 140410 055200
9067 053610 012704 055200
9068 053614 010465 000000
9069 053620 013737 055226 055232
9070 053626 004737 016370
9071 053632 016501 000002
9072 053636 012702 000200
9073 053642 020102
9074 053644 001425
9075 053646
053646 012727 000250
053652 000000
053654 013727 002116
053660 000000
053662 005367 177772
053666 001375
053670 005367 177756
053674 001367
9076 053676 005337 055232
9077 053702 001351
9078 053704 005237 002212
9082 053710
053710 104456
053712 001043
053714 055315
053716 015614
9083 053720
053720 104406

```

```

;*****
;SPACE REVERSE,ACK,CVC=1 COMMAND
;*****
MOV #140410,T25PK3 ;SPACE REVERSE,ACK,CVC=1 COMMAND
MOV #T25PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
MOV R4,TSDB(R5) ;ISSUE COMMAND
MOV T25CN2,T25DLY ;SET UP COUNTER
70$: JSR PC,WAITF ;WAIT FOR SSR TO SET
MOV TSSR(R5),R1 ;GET TSSR CONTENTS
MOV #SSR,R2 ;SET UP EXPECTED
CMP R1,R2 ;ARE THEY EQUAL
BEQ 75$ ;BR, IF OK
DELAY 250 ;WAIT ABOUT .25 SECONDS
MOV #250,(PC)+
WORD 0
MOV L$DLY,(PC)+
WORD 0
DEC -6(PC)
BNE -4
DEC -22(PC)
BNE -20
DEC T25DLY ;BUMP COUNTER DOWN
BNE 70$ ;BR, IF COUNTER HASN'T EXPIRED
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T25WDE,EXPREC ;TSSR INCORRECT AFTER READ DATA
TRAP C$ERHRD
WORD 547
WORD T25WDE
WORD EXPREC
75$: CKLOOP ;LOOP IF SELECTED
TRAP C$CLP1

```


TEST 5: SPACE RECORDS

```

9135
9136
9137
9138
9139 054102          ;
      054102          ;
      054102 104402   ;
9140 054104 004737 056416   JSR   PC,T25REST      ;SET COMMAND PACKET
9141 054110 004737 056510   JSR   PC,T25RT2       ;SET UP OTHER COMMAND PACKET
9142 054114 004737 056552   JSR   PC,T25RT3       ;SET UP OTHER COMMAND PACKET
9143
9144
9145
9146
9147
9148
9149
9150 054120 004737 016114   JSR   PC,SOFINIT     ;DO INITIALIZE ON CONTROLLER
9151 054124 103407          BCS   20$            ;BR IF INIT WAS OK
9152 054126 005237 002212   INC   FATFLG         ;BUMP COUNT
9156 054132 010001          MOV   R0,R1          ;CONTENTS OF TSSR REGISTER
9157 054134          ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      054134 104455          TRAP   C$ERDF
      054136 001046          .WORD  550
      054140 003652          .WORD  SFIERR
      054142 012154          .WORD  SFIMSG
9158 054144 013737 002172 055070 20$:  MOV   UNITN,T25DSW    ;SET UP UNIT NUMBER
9159
9160 054152 012704 055050   MOV   #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
9161
9162
9163
9164
9165
9166
9167
9168 054156 004737 010752   JSR   PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
9169 054162 103407          BCS   25$            ;BR, IF COMMAND ISSUED OK
9170 054164 005237 002212   INC   FATFLG         ;BUMP COUNT
9174 054170 010001          MOV   R0,R1          ;SAVE CONTENTS OF TSSR
9175 054172          ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      054172 104456          TRAP   C$ERHRD
      054174 001047          .WORD  551
      054176 005056          .WORD  WRTMSG
      054200 012154          .WORD  SFIMSG
9176 054202          25$:  CKLOOP          ;LOOP IF SELECTED
      054202 104406          TRAP   C$CLP1
9177
9178
9179
9180
9181
9182
9183
9184 054204 004737 011136   JSR   PC,REWIND      ;CALL TAPE REWIND COMMAND
9185 054210 103407          BCS   30$            ;BR, IF NO PROBLEM
9186 054212 010001          MOV   R0,R1          ;SAVE TSSR

```

TEST 5: SPACE RECORDS

```

9187 054214 005237 002212          INC    FATFLG          ;BUMP COUNT
9191 054220          ERRHRD  ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      054220 104456          TRAP    C$ERHRD
      054222 001050          .WORD  552
      054224 056205          .WORD  T25RWN
      054226 012166          .WORD  PKTSSR
9192 054230          30$:   CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      054230 104406
9193
9194
9195          ;*****
9196          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9197          ;
9198          ;*****
9199
9200 054232 013701 055100          MOV    T25BFR+6,R1      ;PICK UP XSTO
9201 054236 010102          MOV    R1,R2           ;SET UP EXPECTED
9202 054240 052702 000002          BIS    #BIT1,R2        ;SET BOT BIT IN EXPECTED
9203 054244 020102          CMP    R1,R2           ;DOES EXP = REC'D
9204 054246 001406          BEQ    40$             ;BR, IF EQUAL (OK)
9205 054250 005237 002212          INC    FATFLG          ;BUMP COUNT
9209 054254          ERRHRD  ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      054254 104456          TRAP    C$ERHRD
      054256 001051          .WORD  553
      054260 055375          .WORD  T25BOT
      054262 015614          .WORD  EXPREC
9210 054264          40$:   CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      054264 104406
9211 054266 012737 000001 055202          MOV    #1,T25RB        ;NUMBER OF RECORDS TO SPACE OVER
9212
9213          ;*****
9214          ;SPACE REVERSE,ACK COMMAND
9215          ;
9216          ;*****
9217
9218
9219 054274 012737 100410 055200          MOV    #100410,T25PK3  ;SPACE REVERSE,ACK COMMAND
9220 054302 012704 055200          MOV    #T25PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
9221 054306          65$:   MOV    R4,TSDB(R5)    ;ISSUE COMMAND
9222 054306 010465 000000          JSR    PC,WAITF        ;WAIT FOR SSR TO SET
9223 054312 004737 016370          MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
9224 054316 016501 000002          MOV    #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
9225 054322 012702 100206          CMP    R1,R2           ;ARE THEY EQUAL
9226 054326 020102          BEQ    75$             ;BR, IF OK
9227 054330 001406          INC    FATFLG          ;BUMP COUNT
9228 054332 005237 002212          ERRHRD  ERRNO,T25WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
9232 054336          TRAP    C$ERHRD
      054336 104456          .WORD  554
      054340 001052          .WORD  T25WDE
      054342 055315          .WORD  PKTSSR
      054344 012166
9233 054346          75$:   CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      054346 104406
9234
9235          ;*****
9236          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9237

```


K16

TEST 5: SPACE RECORDS

```

054460 003652 .WORD SFIERR
054462 012154 .WORD SFIMSG
9290 054464 013737 002172 055070 20$: MOV UNITN,T25DSW ;SET UP UNIT NUMBER
9291
9292 054472 012704 055050 MOV #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
9293
9294 ;*****
9295 ;
9296 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
9297 ;
9298 ;*****
9299
9300 054476 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
9301 054502 103407 BCS 25$ ;BR, IF COMMAND ISSUED OK
9302 054504 005237 002212 INC FATFLG ;BUMP COUNT
9306 054510 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
9307 054512 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
054512 104456 TRAP C$ERHRD
054514 001055 .WORD 557
054516 005056 .WORD WRTMSG
054520 012154 .WORD SFIMSG
9308 054522 25$: CKLOOP ;LOOP IF SELECTED
054522 104406 TRAP C$CLP1
9309
9310 ;*****
9311 ;
9312 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9313 ;
9314 ;*****
9315
9316 054524 004737 011136 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
9317 054530 103407 BCS 30$ ;BR, IF NO PROBLEM
9318 054532 010001 MOV RO,R1 ;SAVE TSSR
9319 054534 005237 002212 INC FATFLG ;BUMP COUNT
9323 054540 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
054540 104456 TRAP C$ERHRD
054542 001056 .WORD 558
054544 056205 .WORD T25RWN
054546 012166 .WORD PKTSSR
9324 054550 30$: CKLOOP ;LOOP IF SELECTED
054550 104406 TRAP C$CLP1
9325
9326 ;*****
9327 ;
9328 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9329 ;
9330 ;*****
9331
9332 054552 013701 055100 MOV T25BFR+6,R1 ;PICK UP XSTO
9333 054556 010102 MOV R1,R2 ;SET UP EXPECTED
9334 054560 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
9335 054564 020102 CMP R1,R2 ;DOES EXP = REC'D
9336 054566 001406 BEQ 40$ ;BR, IF EQUAL (OK)
9337 054570 005237 002212 INC FATFLG ;BUMP COUNT
9341 054574 ERRHRD ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
054574 104456 TRAP C$ERHRD
054576 001057 .WORD 559

```

L16

TEST 5: SPACE RECORDS

```

054600 055375 .WORD T25BOT
054602 015614 .WORD EXPREC
9342 054604 40$: CKLOOP TRAP C$CLP1
054604 104406 ;NUMBER OF RECORDS TO SPACE OVER
9343 054606 012737 000001 055202 MOV #1,T25RB
9344
9345 ;*****
9346 ;SPACE FORWARD,IE,ACK,CVC=1 COMMAND
9347 ;
9348 ;*****
9349
9350
9351 054614 012737 140210 055200 MOV #140210,T25PK3 ;SPACE FORWARD,IE,ACK,CVC=1 COMMAND
9352 054622 012704 055200 MOV #T25PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
9353 054626 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
9354 054632 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
9355 054636 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9356 054642 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9357 054646 020102 CMP R1,R2 ;ARE THEY EQUAL
9358 054650 001406 BEQ 75$ ;BR, IF OK
9359 054652 005237 002212 INC FATFLG ;BUMP COUNT
9363 054656 ERRHRD ERRNO,T25WDE,EXPREC ;TSSR INCORRECT AFTER READ DATA
054656 104456 TRAP C$ERHRD
054660 001060 .WORD 560
054662 055315 .WORD T25WDE
054664 015614 .WORD EXPREC
9364 054666 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
054666 104406 ;NUMBER OF RECORDS TO SPACE OVER
9365 054670 012737 000020 055202 MOV #20,T25RB
9366
9367 ;*****
9368 ;SPACE REVERSE,IE,ACK COMMAND
9369 ;
9370 ;*****
9371
9372
9373 054676 012737 100610 055200 MOV #100610,T25PK3 ;SPACE REVERSE,IE,ACK COMMAND
9374 054704 012704 055200 MOV #T25PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
9375 054710 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
9376 054714 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
9377 054720 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9378 054724 012702 100204 MOV #SSR!BIT2!SC,R2 ;SET UP EXPECTED
9379 054730 020102 CMP R1,R2 ;ARE THEY EQUAL
9380 054732 001406 BEQ 175$ ;BR, IF OK
9381 054734 005237 002212 INC FATFLG ;BUMP COUNT
9385 054740 ERRHRD ERRNO,T25WDE,EXPREC ;TSSR INCORRECT AFTER READ DATA
054740 104456 TRAP C$ERHRD
054742 001061 .WORD 561
054744 055315 .WORD T25WDE
054746 015614 .WORD EXPREC
9386 054750 175$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
054750 104406
9387 054752 013701 055106 MOV T25BFR+14,R1 ;GET XST3 STATUS WORD
9388 054756 010102 MOV R1,R2 ;SET UP EXPECTED
9389 054760 052702 000001 BIS #BIT0,R2 ;SET THE RIB BIT
9390 054764 020102 CMP R1,R2 ;ARE THEY EQUAL
9391 054766 001406 BEQ 180$ ;BR, IF EQUAL (GOOD)

```


B1

TEST 5: SPACE RECORDS

```

9449 055204 000000
9450 055206 000000
9451
9452
9453
9454
9455 055210
9456 055210 010
9457 055211 200
9458 055212 000000
9459 055214 000000
9460
9461
9462
9463
9464
9465 055216 100005
9466 055220 100405
9467 055222 102005
9468 055224 177777
9469
9470 055226 000000
9471 055230 000000
9472 055232 000000
9473
9474
9475
9476
9477
9478 055234 127 122 111
9479 055315 124 123 123
9480 055375 124 141 160
9481 055442 124 123 123
9482 055531 127 162 151
9483 055605 123 160 141
9484 055670 123 160 141
9485 055760 123 160 141
9486 056043 123 160 141
9487 056123 123 160 141
9488 056205 122 145 167
9489 056254 104 162 151
9490 056327 124 123 123
9491 056400 123 160 141
9492
9493
9494
9495
9496
9497
9498
9499
9500 056416
9501 056416
9502 056422 012701 055050
9503 056426 012721 100004
9504 056432 012721 055060
9505 056436 005021

T25SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
        .WORD 0
        .EVEN
;
;
T25BF2:
T25BS0: .BYTE 10 ;BSELO AREA
T25BS1: .BYTE 200 ;BSEL1 AREA
T25S2: .WORD 0 ;SEL 2 AREA
T25S3: .WORD 0 ;DATA AREA
;
;
        .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T25RN: .WORD 100005 ;READ DATA (NEXT)
T25WDR: .WORD 100405 ;READ DATA RETRY
T25CON: .WORD 102005 ;WRITE CONTINOUS
        .WORD 177777 ;END OF DATA
;
T25CN2: .WORD 0 ;COUNTER FOR RECORDS
T25CNT: .WORD 0 ;COUNTER FOR RECORDS
T25DLY: .WORD 0 ;COUNTER FOR RECORDS
;
;*
;LOCAL TEXT MESSAGES FOR TEST
;-
T25SSR: .ASCIZ 'WRITE SUBSYSTEM Miscellaneous Read Status Failed'
T25WDE: .ASCIZ 'TSSR Not Correct After POSITION (SPACE) Command'
T25BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
T25TM: .ASCIZ 'TSSR Not Correct After POSITION (Space Command) Reject'
T25NET: .ASCIZ 'Write Tape, Status Alert, But No EOT Sensed'
T25WNG: .ASCIZ 'Space Forward Failed To Position On Correct Record'
T25BNC: .ASCIZ 'Space Forward, From BOT, Failed To Clear BOT Indication'
T25WNH: .ASCIZ 'Space Reverse Failed To Position On Correct Record'
T25NEF: .ASCIZ 'Space Reverse, At BOT, Failed To Set NEF (XST0)'
T25RIB: .ASCIZ 'Space Reverse, Into BOT, Failed To Set RIB (XST3)'
T25RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
T25OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
T25WDC: .ASCIZ 'TSSR Not Correct After READ DATA Command'
T25ID: .ASCIZ 'Space Records'
        .EVEN
;
;
;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
;WRITE SUBSYSTEM MEMORY COMMAND
;
;-
T25REST:
        SAVREG ;SAVE THE REGISTERS
        MOV #T25PACKET,R1 ;START OF THE PACKET
        MOV #100004,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK
        MOV #T25DATA,(R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
        CLR (R1)+ ;EXTENDED ADDRESS

```


E1

TEST 6: REREADS

```

9614 056744 010001          MOV    R0,R1          ;CONTENTS OF TSSR REGISTER
9615 056746          ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                TRAP    C#ERDF
                                .WORD  601
                                .WORD  SFIERR
                                .WORD  SFIMSG
9616 056756          013737 002172 073100 20$:
9617 056756          MOV    UNITN,T26DSW      ;SET UP UNIT NUMBER
9618 056764          MOV    #T26PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
9619
9620          ;*****
9621          ;
9622          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
9623          ;
9624          ;*****
9625
9626 056770          004737 010752          JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
9627 056774          103407          BCS    26$           ;BR, IF COMMAND ISSUED OK
9628 056776          005237 002212          INC    FATFLG        ;BUMP COUNT
9632 057002          010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
9633 057004          ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP    C#ERHRD
                                .WORD  602
                                .WORD  WRTMSG
                                .WORD  SFIMSG
9634 057014          057014 104406          26$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C#CLP1
9635
9636          ;*****
9637          ;
9638          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9639          ;
9640          ;*****
9641
9642 057016          004737 011136          JSR    PC,REWIND     ;CALL TAPE REWIND COMMAND
9643 057022          103413          BCS    30$           ;BR, IF NO PROBLEM
9644 057024          016501 000002          MOV    TSSR(R5),R1   ;GET TSSR
9645 057030          012702 000200          MOV    #SSR,R2      ;SET UP EXPECTED TSSR
9646 057034          010004          MOV    R0,R4        ;PACKET ADDRESS SET UP
9647 057036          005237 002212          INC    FATFLG        ;BUMP COUNT
9651 057042          ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C#ERHRD
                                .WORD  603
                                .WORD  T26RWN
                                .WORD  PKTSSR
9652 057052          057052 104406          30$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C#CLP1
9653
9654          ;*****
9655          ;
9656          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
9657          ;
9658          ;*****
9659
9660 057054          013701 073110          MOV    T26BFR+6,R1   ;PICK UP XST0
9661 057060          010102          MOV    R1,R2        ;SET UP EXPECTED
9662 057062          052702 000002          BIS    #BIT1,R2     ;SET BOT BIT IN EXPECTED

```

F1

TEST 6: REREADS

```

9663 057066 020102          CMP      R1,R2          ;DOES EXP = REC'D
9664 057070 001406          BEQ      40$           ;BR, IF EQUAL (OK)
9665 057072 005237 002212  INC      FATFLG        ;BUMP COUNT
9669 057076          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    604
                                .WORD    T26BOT
                                .WORD    EXPREC
9670 057106          40$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
9671 057110 012703 000400  MOV      #256.,R3      ;RECORD SIZE
9672 057114 013737 003116 073212  MOV      FREE,T26RB    ;STARTING WRITE BUFFER ADDRESS
9673
9674 ;*****
9675 ;
9676 ;WRITE DATA,ACK,CVC=1 COMMAND
9677 ;
9678 ;*****
9679
9680 057122 012737 140005 073210  MOV      #140005,T26PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
9681 057130 012704 073210  MOV      #T26PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
9682 057134          65$:   MOV      R3,R0          ;SET PATTERN IN CORRECT REGISTER
9683 057134 010300          JSR      PC,FILLMEM    ;FILL MEMORY WITH RECORD SIZE
9684 057136 004737 017542  MOV      R3,T26SZ     ;SET UP RECORD SIZE IN PACKET
9685 057142 010337 073216  MOV      R4,TSDB(R5)  ;ISSUE COMMAND
9686 057146 010465 000000  JSR      PC,WAITF     ;WAIT FOR SSR TO SET
9687 057152 004737 016370  MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
9688 057156 016501 000002  MOV      #SSR,R2      ;SET UP EXPECTED
9689 057162 012702 000200  CMP      R1,R2        ;ARE THEY EQUAL
9690 057166 020102          BEQ      75$           ;BR, IF OK
9691 057170 001406          INC      FATFLG        ;BUMP COUNT
9692 057172 005237 002212  ERRHRD  ERRNO,WRERR,EXPREC ;TSSR INCORRECT AFTER WRITE DATA
9696 057176          TRAP      C$ERHRD
                                .WORD    605
                                .WORD    WRERR
                                .WORD    EXPREC
9697 057206          75$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
9698 057210 005723          TST      (R3)+        ;BUMP RECORD SIZE
9699 057212 022703 000414  CMP      #268.,R3    ;END OF RECORD YET
9700 057216 001346          BNE      65$         ;BR, IF MORE RECORDS TO WRITE
9701 057220          80$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
9702 057222          120$:
9703
9704 ;*****
9705 ;
9706 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9707 ;
9708 ;*****
9709
9710 057222 004737 011136  JSR      PC,REWIND    ;CALL TAPE REWIND COMMAND
9711 057226 103413          BCS     130$         ;BR, IF NO PROBLEM
9712 057230 016501 000002  MOV      TSSR(R5),R1  ;GET TSSR
9713 057234 012702 000200  MOV      #SSR,R2      ;SET UP EXPECTED TSSR
9714 057240 010004          MOV      R0,R4       ;PACKET ADDRESS SET UP

```

G1

TEST 6: REREADS

```

9715 057242 005237 002212          INC    FATFLG          ;BUMP COUNT
9719 057246          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
          057246 104456          TRAP    C$ERHRD
          057250 001136          .WORD  606
          057252 074544          .WORD  T26RWN
          057254 012166          .WORD  PKTSSR
9720 057256          130$:  CKLOOP          ;LOOP IF SELECTED
          057256 104406          TRAP    C$CLP1
9721
9722          ;*****
9723          ;
9724          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9725          ;
9726          ;*****
9727
9728 057260 013701 073110          MOV    T26BFR+6,R1      ;PICK UP XSTO
9729 057264 010102          MOV    R1,R2           ;SET UP EXPECTED
9730 057266 052702 000002          BIS    #BIT1,R2        ;SET BOT BIT IN EXPECTED
9731 057272 020102          CMP    R1,R2           ;DOES EXP = REC'D
9732 057274 001406          BEQ   140$             ;BR, IF EQUAL (OK)
9733 057276 005237 002212          INC    FATFLG          ;BUMP COUNT
9737 057302          ERRHRD  ERRNO,T26BOT,PKTSSR ;TAPE NOT AT BOT AFTER REWIND
          057302 104456          TRAP    C$ERHRD
          057304 001137          .WORD  607
          057306 074255          .WORD  T26BOT
          057310 012166          .WORD  PKTSSR
9738 057312          140$:  CKLOOP          ;LOOP IF SELECTED
          057312 104406          TRAP    C$CLP1
9739 057314 012737 000400 073242      MOV    #256.,T26RSZ    ;SET RECORD SIZE
9740
9741          ;*****
9742          ;
9743          ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
9744          ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
9745          ;
9746          ;*****
9747
9748 057322 012703 000001          145$:  MOV    #1,R3      ;SPACE ONE RECORD PARAMETER
9749 057326 004737 010560          JSR    PC,SPACE        ;CALL SPACE ROUTINE
9750 057332 103412          BCS   150$             ;BR, IF NO PROBLEM WITH SPACE COMMAND
9751 057334 016501 000002          MOV    TSSR(R5),R1     ;GET TSSR
9752 057340 012702 000200          MOV    #SSR,R2        ;SET UP EXPECTED TSSR
9753 057344 005237 002212          INC    FATFLG          ;BUMP COUNT
9757 057350          ERRHRD  ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
          057350 104456          TRAP    C$ERHRD
          057352 001140          .WORD  608
          057354 073657          .WORD  T26SC
          057356 015614          .WORD  EXPREC
9758 057360          150$:  CKLOOP          ;LOOP IF SELECTED
          057360 104406          TRAP    C$CLP1
9759 057362 013703 073242          MOV    T26RSZ,R3      ;RECORD SIZE
9760 057366 013737 003116 073212      MOV    FREE,T26RB     ;STARTING READ BUFFER ADDRESS
9761
9762          ;*****
9763          ;
9764          ;REREREAD DATA,CVC=1,ACK COMMAND
9765          ;

```


J1

TEST 6: REREADS

```

9868
9869
9870 057664 004737 011136
9871 057670 103413
9872 057672 016501 000002
9873 057676 012702 000200
9874 057702 010004
9875 057704 005237 002212
9879 057710
      057710 104456
      057712 001145
      057714 074544
      057716 012166
9880 057720
      057720 104406
9881
9882
9883
9884
9885
9886
9887
9888 057722 013701 073110
9889 057726 010102
9890 057730 052702 000002
9891 057734 020102
9892 057736 001406
9893 057740 005237 002212
9897 057744
      057744 104456
      057746 001146
      057750 074255
      057752 015614
9898 057754
      057754 104406
9899 057756 012703 000400
9900 057762 013737 003116 073212
9901
9902
9903
9904
9905
9906
9907
9908 057770 012737 110005 073210
9909 057776 012704 073210
9910 060002
9911 060002 010300
9912 060004 004737 017542
9913 060010 010337 073216
9914 060014 010465 000000
9915 060020 004737 016370
9916 060024 016501 000002
9917 060030 012702 000200
9918 060034 020102
9919 060036 001406
9920 060040 005237 002212

;*****
;
;CALL TAPE REWIND COMMAND
;BR, IF NO PROBLEM
;GET TSSR
;SET UP EXPECTED TSSR
;PACKET ADDRESS SET UP
;BUMP COUNT
;REWIND NOT ACCEPTED
      TRAP C4ERHRD
      .WORD 613
      .WORD T26RWN
      .WORD PKTSSR
30$: CKLOOP ;LOOP IF SELECTED
      TRAP C4CLP1
;*****
;
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
;
;*****
      MOV T26BFR+6,R1 ;PICK UP XST0
      MOV R1,R2 ;SET UP EXPECTED
      BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
      CMP R1,R2 ;DOES EXP = REC'D
      BEQ 40$ ;BR, IF EQUAL (OK)
      INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      TRAP C4ERHRD
      .WORD 614
      .WORD T26BOT
      .WORD EXPREC
40$: CKLOOP ;LOOP IF SELECTED
      TRAP C4CLP1
      MOV #256,R3 ;RECORD SIZE
      MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
;*****
;
;WRITE DATA,ACK,SWB COMMAND
;
;*****
      MOV #110005,T26PK3 ;WRITE DATA,ACK,SWB COMMAND
      MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
65$:
      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
      MOV #SSR,R2 ;SET UP EXPECTED
      CMP R1,R2 ;ARE THEY EQUAL
      BEQ 75$ ;BR, IF OK
      INC FATFLG ;BUMP COUNT

```

K1

TEST 6: REREADS

```

9924 060044          ERRHRD  ERRNO,WRERR,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA
      060044 104456          TRAP C$ERHRD
      060046 001147          .WORD 615
      060050 005113          .WORD WRERR
      060052 012166          .WORD PKTSSR
9925 060054          75$: CKLOOP                      ;LOOP IF SELECTED
      060054 104406          TRAP C$CLP1
9926 060056 005723          TST (R3)+          ;BUMP RECORD SIZE
9927 060060 022703 000414  CMP #268.,R3      ;END OF RECORD YET
9928 060064 001346          BNE 65$          ;BR, IF MORE RECORDS TO WRITE
9929 060066          80$: CKLOOP                      ;LOOP IF SELECTED
      060066 104406          TRAP C$CLP1
9930 060070          120$:
9931
9932 ;*****
9933 ;
9934 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9935 ;
9936 ;*****
9937
9938 060070 004737 011136    JSR PC,REWIND      ;CALL TAPE REWIND COMMAND
9939 060074 103413          BCS 130$          ;BR, IF NO PROBLEM
9940 060076 016501 000002    MOV TSSR(R5),R1   ;GET TSSR
9941 060102 012702 000200    MOV #SSR,R2       ;SET UP EXPECTED TSSR
9942 060106 010004          MOV R0,R4          ;PACKET ADDRESS SET UP
9943 060110 005237 002212    INC FATFLG        ;BUMP COUNT
9947 060114          ERRHRD  ERRNO,T26RWN,PKTSSR      ;REWIND NOT ACCEPTED
      060114 104456          TRAP C$ERHRD
      060116 001150          .WORD 616
      060120 074544          .WORD T26RWN
      060122 012166          .WORD PKTSSR
9948 060124          130$: CKLOOP                      ;LOOP IF SELECTED
      060124 104406          TRAP C$CLP1
9949
9950 ;*****
9951 ;
9952 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
9953 ;
9954 ;*****
9955
9956 060126 013701 073110    MOV T26BFR+6,R1   ;PICK UP XST0
9957 060132 010102          MOV R1,R2          ;SET UP EXPECTED
9958 060134 052702 000002    BIS #BIT1,R2      ;SET BOT BIT IN EXPECTED
9959 060140 020102          CMP R1,R2          ;DOES EXP = REC'D
9960 060142 001406          BEQ 140$          ;BR, IF EQUAL (OK)
9961 060144 005237 002212    INC FATFLG        ;BUMP COUNT
9965 060150          ERRHRD  ERRNO,T26BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      060150 104456          TRAP C$ERHRD
      060152 001151          .WORD 617
      060154 074255          .WORD T26BOT
      060156 015614          .WORD EXPREC
9966 060160          140$: CKLOOP                      ;LOOP IF SELECTED
      060160 104406          TRAP C$CLP1
9967 060162 012737 000400 073242  MOV #256.,T26RSZ  ;SET UP RECORD SIZE
9968
9969 ;*****
9970 ;

```

L1

TEST 6: REREADS

```

9971 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
9972 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
9973 ;
9974 ;*****
9975
9976 060170 012703 000001 145$: MOV #1,R3 ;SPACE ONE RECORD PARAMETER
9977 060174 004737 010560 JSR PC,SPACE ;CALL SPACE ROUTINE
9978 060200 103412 BCS 150$ ;BR, IF NO PROBLEM WITH SPACE COMMAND
9979 060202 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
9980 060206 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
9981 060212 005237 002212 INC FATFLG ;BUMP COUNT
9985 060216 ERRHRD ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
060216 104456 TRAP C#ERHRD
060220 001152 .WORD 618
060222 073657 .WORD T26SC
060224 015614 .WORD EXPREC
9986 060226 150$: CKLOOP TRAP C#CLP1
060226 104406
9987 060230 013703 073242 MOV T26RSZ,R3 ;RECORD SIZE
9988 060234 013737 003116 073212 MOV FREE,T26RB ;STARTING READ BUFFER ADDRESS
9989
9990 ;*****
9991 ;
9992 ;REREAD DATA,CVC=1,ACK,SWB COMMAND
9993 ;
9994 ;*****
9995
9996 060242 012737 151001 073210 165$: MOV #151001,T26PK3 ;REREAD DATA,CVC=1,ACK,SWB COMMAND
9997 060250 012704 073210 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
9998 060254 010337 073216 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
9999 060260 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
10000 060264 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
10001 060270 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10002 060274 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
10003 060300 020102 CMP R1,R2 ;ARE THEY EQUAL
10004 060302 001406 BEQ 170$ ;BR, IF OK
10005 060304 005237 002212 INC FATFLG ;BUMP COUNT
10009 060310 ERRHRD ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
060310 104456 TRAP C#ERHRD
060312 001153 .WORD 619
060314 075100 .WORD T26WDC
060316 012166 .WORD PKTSSR
10010 060320 170$: CKLOOP ;LOOP IF SELECTED TRAP C#CLP1
060320 104406
10011 060322 013702 003116 MOV FREE,R2 ;CURRENT BUFFER ADDRESS TO R2
10012 060326 010304 MOV R3,R4 ;CURRENT RECORD SIZE
10013 060330 162704 000400 SUB #256.,R4 ;FIRST LOCATION IN BUFFER
10014 060334 060204 173$: ADD R2,R4 ;SET UP POINTER
10015 060336 021403 CMP (R4),R3 ;CHECK DATA READ (R3=DATA ALSO)
10016 060340 001410 BEQ 180$ ;BR, IF ALL IS WELL
10017 060342 011401 MOV (R4),R1 ;RECD DATA
10018 060344 010302 MOV R3,R2 ;EXPECTED DATA
10019 060346 005237 002212 INC FATFLG ;BUMP COUNT
10023 060352 ERRHRD ERRNO,T26DTA,EXPREC ;DATA READ NOT = WRITTEN
060352 104456 TRAP C#ERHRD
060354 001154 .WORD 620
060356 074322 .WORD T26DTA

```


B2

TEST 6: REREADS

```

060566 074544
060570 012166
10129 060572 104406 30$: CKLOOP ;LOOP IF SELECTED .WORD T26RWN
060572 104406 TRAP C$CLP1 .WORD PKTSSR
10130
10131 ;*****
10132 ;
10133 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
10134 ;
10135 ;*****
10136
10137 060574 013701 073110 MOV T26BFR+6,R1 ;PICK UP XSTO
10138 060600 010102 MOV R1,R2 ;SET UP EXPECTED
10139 060602 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
10140 060606 020102 CMP R1,R2 ;DOES EXP = REC'D
10141 060610 001406 BEQ 40$ ;BR, IF EQUAL (OK)
10142 060612 005237 002212 INC FATFLG ;BUMP COUNT
10146 060616 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
060616 104456 TRAP C$ERHRD
060620 001160 .WORD 624
060622 074255 .WORD T26BOT
060624 015614 .WORD EXPREC
10147 060626 104406 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
060626 104406 MOV #256,R3 ;RECORD SIZE
10148 060630 012703 000400 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
10149 060634 013737 003116 073212
10150
10151 ;*****
10152 ;
10153 ;WRITE DATA,CVC=1,ACK COMMAND
10154 ;
10155 ;*****
10156
10157 060642 012737 140005 073210 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
10158 060650 012704 073210 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
10159 060654
10160 060654 010300 65$: MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
10161 060656 004737 017542 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
10162 060662 010337 073216 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
10163 060666 013777 073236 122222 MOV T26CNT,#FREE ;MOVE TAPE RECORD NUMBER TO BUFFER
10164 060674 062737 000001 073236 ADD #1,T26CNT ;NUMBER READY FOR NEXT RECORD
10165 060702 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
10166 060706 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
10167 060712 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10168 060716 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
10169 060722 020102 CMP R1,R2 ;ARE THEY EQUAL
10170 060724 001406 BEQ 75$ ;BR, IF OK
10171 060726 005237 002212 INC FATFLG ;BUMP COUNT
10175 060732 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
060732 104456 TRAP C$ERHRD
060734 001161 .WORD 625
060736 005113 .WORD WRERR
060740 012166 .WORD PKTSSR
10176 060742 104406 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
060742 104406 TST (R3) ;BUMP THE RECORD SIZE
10177 060744 005723 CMP #268.,R3 ;MAXIMUM SIZE YET
10178 060746 022703 000414

```

TEST 6: REREADS

```

10179 060752 001401          BEQ      120$          ;BR, IF AT END OF WRITE SEQUENCE
10180 060754 000737          BR       65$          ;WRITE MORE RECORDS
10181 060756                120$:          CLR      T26CNT        ;SET RECORD COUNTER BACK TO ZERO
10182 060756 005037 073236
10183
10184 ;*****
10185 ;
10186 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10187 ;
10188 ;*****
10189
10190 060762 004737 011136      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
10191 060766 103413          BCS     130$          ;BR, IF NO PROBLEM
10192 060770 016501 000002      MOV     TSSR(R5),R1    ;GET TSSR
10193 060774 012702 000200      MOV     #SSR,R2        ;SET UP EXPECTED TSSR
10194 061000 010004          MOV     R0,R4          ;PACKET ADDRESS SET UP
10195 061002 005237 002212      INC     FATFLG         ;BUMP COUNT
10199 061006          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
          061006 104456          TRAP   C#ERHRD
          061010 001162          .WORD 626
          061012 074544          .WORD T26RWN
          061014 012166          .WORD PKTSSR
10200 061016          130$:  CKLOOP        ;LOOP IF SELECTED
          061016 104406          TRAP   C#CLP1
10201
10202 ;*****
10203 ;
10204 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
10205 ;
10206 ;*****
10207
10208 061020 013701 073110      MOV     T26BFR+6,R1    ;PICK UP XST0
10209 061024 010102          MOV     R1,R2          ;SET UP EXPECTED
10210 061026 052702 000002      BIS     #BIT1,R2       ;SET BOT BIT IN EXPECTED
10211 061032 020102          CMP     R1,R2          ;DOES EXP = REC'D
10212 061034 001406          BEQ     140$          ;BR, IF EQUAL (OK)
10213 061036 005237 002212      INC     FATFLG         ;BUMP COUNT
10217 061042          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          061042 104456          TRAP   C#ERHRD
          061044 001163          .WORD 627
          061046 074255          .WORD T26BOT
          061050 015614          .WORD EXPREC
10218 061052          140$:  CKLOOP        ;LOOP IF SELECTED
          061052 104406          TRAP   C#CLP1
10219
10220 ;*****
10221 ;
10222 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
10223 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
10224 ;
10225 ;*****
10226
10227 061054 012703 000001      MOV     #1,R3          ;SPACE 1 RECORD FORWARD
10228 061060 004737 010560      JSR     PC,SPACE       ;SPACE CALL
10229 061064 012703 000400      MOV     #256,R3        ;RECORD SIZE
10230 061070 013737 003116 073212 150$:  MOV     FREE,T26RB     ;STARTING READ BUFFER ADDRESS
10231

```


TEST 6: REREADS

```

10232 ;*****
10233 ;
10234 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
10235 ;
10236 ;*****
10237
10238 061076 012737 161001 073210      MOV      #161001,T26PK3      ;REREAD DATA,CVC=1,ACK, OPP COMMAND
10239 061104 012704 073210      MOV      #T26PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
10240 061110 010337 073216      MOV      R3,T26SZ          ;SET UP RECORD SIZE IN PACKET
10241 061114 010465 000000      MOV      R4,TSDB(R5)       ;ISSUE COMMAND
10242 061120 004737 016370      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
10243 061124 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
10244 061130 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
10245 061134 020102              CMP      R1,R2             ;ARE THEY EQUAL
10246 061136 001406              BEQ      170$              ;BR, IF OK
10247 061140 005237 002212      INC      FATFLG            ;BUMP COUNT
10251 061144              ERRHRD  ERRNO,T26RRG,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                                TRAP      C$ERHRD
                                .WORD    628
                                .WORD    T26RRG
                                .WORD    PKTSSR
                                TRAP      C$CLP1
10252 061154              170$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104456
                                .WORD    001164
                                .WORD    073562
                                .WORD    012166
10253 061154              104406
10254 061156 005723
10255 061160 062737 000001 073236      TST      (R3)+             ;BUMP RECORD SIZE
10256 ;*****
10257 ;
10258 ;READ DATA, CVC=1, ACK COMMAND
10259 ;
10260 ;*****
10261
10262 061166 012737 140001 073210      MOV      #140001,T26PK3    ;READ DATA, CVC=1, ACK COMMAND
10263 061174 010337 073216      MOV      R3,T26SZ          ;SET SIZE INTO PACKET
10264 061200 010465 000000      MOV      R4,TSDB(R5)       ;ISSUE READ DATA COMMAND
10265 061204 004737 016370      JSR      PC,WAITF          ;WAIT FOR SSR
10266 061210 016501 000002      MOV      TSSR(R5),R1       ;PICK UP THE TSSR
10267 061214 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
10268 061220 020102              CMP      R1,R2             ;IS THE TSSR OK
10269 061222 001406              BEQ      195$              ;BR, IF TSSR OK (GOOD)
10270 061224 005237 002212      INC      FATFLG            ;BUMP COUNT
10274 061230              ERRHRD  ERRNO,RDERR,PKTSSR  ;READ DATA COMMAND FAILED
                                TRAP      C$ERHRD
                                .WORD    629
                                .WORD    RDERR
                                .WORD    PKTSSR
                                TRAP      C$CLP1
10275 061240              195$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406
                                .WORD    017701
                                .WORD    121650
                                .WORD    013702
                                .WORD    073236
10276 061242              017701
10277 061246              013702
10278 061252 020102
10279 061254 001406
10280 061256 005237 002212      INC      FATFLG            ;BUMP COUNT
10284 061262              ERRHRD  ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
                                TRAP      C$ERHRD
                                .WORD    630
                                .WORD    T26WNG
                                .WORD    104456
                                .WORD    001166
                                .WORD    073246

```


TEST 6: REREADS

```

10335
10336
10337
10338
10339
10340
10341 061402 004737 010752
10342 061406 103407
10343 061410 005237 002212
10347 061414 010001
10348 061416
      061416 104456
      061420 001170
      061422 005056
      061424 012154
10349 061426
      061426 104406
10350
10351
10352
10353
10354
10355
10356
10357 061430 004737 011136
10358 061434 103413
10359 061436 016501 000002
10360 061442 012702 000200
10361 061446 010004
10362 061450 005237 002212
10366 061454
      061454 104456
      061456 001171
      061460 074544
      061462 012166
10367 061464
      061464 104406
10368
10369
10370
10371
10372
10373
10374
10375 061466 013701 073110
10376 061472 010102
10377 061474 052702 000002
10378 061500 020102
10379 061502 001406
10380 061504 005237 002212
10384 061510
      061510 104456
      061512 001172
      061514 074255
      061516 015614
10385 061520
      061520 104406

;*****
;
;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,SFMSG ;WRITE CHARACTERISTICS FAILED
                                  TRAP      C;ERHRD
                                  .WORD     632
                                  .WORD     WRTMSG
                                  .WORD     SFMSG
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
      MOV     R0,R4              ;PACKET ADDRESS SET UP
      INC     FATFLG             ;BUMP COUNT
      ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                  TRAP      C;ERHRD
                                  .WORD     633
                                  .WORD     T26RWN
                                  .WORD     PKTSSR
30;:  CKLOOP                    ;LOOP IF SELECTED
                                  TRAP      C;CLP1

;*****
;
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;
;*****
      MOV     T26BFR-6,R1        ;PICK UP XSTO
      MOV     R1,R2              ;SET UP EXPECTED
      BIS     #BIT1,R2           ;SET BOT BIT IN EXPECTED
      CMP     R1,R2              ;DOES EXP = REC'D
      BEQ     40;                ;BR, IF EQUAL (OK)
      INC     FATFLG             ;BUMP COUNT
      ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                  TRAP      C;ERHRD
                                  .WORD     634
                                  .WORD     T26BOT
                                  .WORD     EXPREC
40;:  CKLOOP                    ;LOOP IF SELECTED
                                  TRAP      C;CLP1

```

TEST 6: REREADS

```

10386 061522 012703 000400          MOV    #256.,R3          ;RECORD SIZE
10387 061526 013737 003116 073212  MOV    FREE,T26RB      ;STARTING WRITE BUFFER ADDRESS
10388
10389 ;*****
10390 ;
10391 ;WRITE DATA,CVC=1,ACK COMMAND
10392 ;
10393 ;*****
10394
10395 061534 012737 140005 073210          MOV    #140005,T26PK3  ;WRITE DATA,CVC=1,ACK COMMAND
10396 061542 012704 073210          MOV    #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
10397 061546
10398 061546 010300 65$:          MOV    R3,R0          ;SET PATTERN IN CORRECT REGISTER
10399 061550 004737 017542          JSR    PC,FILLMEM     ;FILL MEMORY WITH RECORD SIZE
10400 061554 010337 073216          MOV    R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
10401 061560 013777 073236 121330  MOV    T26CNT,@FREE  ;MOVE TAPE RECORD NUMBER TO BUFFER
10402 061566 062737 000001 073236  ADD    #1,T26CNT     ;NUMBER READY FOR NEXT RECORD
10403 061574 010465 000000          MOV    R4,TSDB(R5)   ;ISSUE COMMAND
10404 061600 004737 016370          JSR    PC,WAITF      ;WAIT FOR SSR TO SET
10405 061604 016501 000002          MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
10406 061610 012702 000200          MOV    #SSR,R2      ;SET UP EXPECTED
10407 061614 020102          CMP    R1,R2        ;ARE THEY EQUAL
10408 061616 001406          BEQ    75$          ;BR, IF OK
10409 061620 005237 002212          INC    FATFLG       ;BUMP COUNT
10413 061624          ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
          TRAP  C#ERHRD
          .WORD 635
          .WORD WRERR
          .WORD PKTSSR
10414 061634 104406 75$:          CKLOOP          ;LOOP IF SELECTED
          TRAP  C#CLP1
10415 061636 005723          TST    (R3)+        ;BUMP THE RECORD SIZE
10416 061640 022703 000412          CMP    #266.,R3    ;MAXIMUM SIZE YET
10417 061644 001401          BEQ    120$        ;BR, IF AT END OF WRITE SEQUENCE
10418 061646 000737          BR     65$         ;WRITE MORE RECORDS
10419 061650
10420 061650 005037 073236 120$:          CLR    T26CNT       ;SET RECORD COUNTER BACK TO ZERO
10421
10422 ;*****
10423 ;
10424 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10425 ;
10426 ;*****
10427
10428 061654 004737 011136          JSR    PC,REWIND     ;CALL TAPE REWIND COMMAND
10429 061660 103413          BCS    130$        ;BR, IF NO PROBLEM
10430 061662 016501 000002          MOV    TSSR(R5),R1  ;GET TSSR
10431 061666 012702 000200          MOV    #SSR,R2     ;SET UP EXPECTED TSSR
10432 061672 010004          MOV    R0,R4       ;PACKET ADDRESS SET UP
10433 061674 005237 002212          INC    FATFLG      ;BUMP COUNT
10437 061700          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
          TRAP  C#ERHRD
          .WORD 636
          .WORD T26RWN
          .WORD PKTSSR
          061700 104456
          061702 001174
          061704 074544
          061706 012166
10438 061710 104406 130$:          CKLOOP          ;LOOP IF SELECTED
          TRAP  C#CLP1
          061710 104406

```

H2

TEST 6: REREADS

```

10439
10440
10441
10442
10443
10444
10445
10446 061712 013701 073110
10447 061716 010102
10448 061720 052702 000002
10449 061724 020102
10450 061726 001406
10451 061730 005237 002212
10455 061734
      061734 104456
      061736 001175
      061740 074255
      061742 015614
10456 061744
      061744 104406
10457
10458
10459
10460
10461
10462
10463
10464
10465 061746 012703 000001
10466 061752 004737 010560
10467 061756 012703 000400
10468 061762 013737 003116 073212
10469
10470
10471
10472
10473
10474
10475
10476 061770 012737 171001 073210
10477 061776 012704 073210
10478 062002 010337 073216
10479 062006 010465 000000
10480 062012 004737 016370
10481 062016 016501 000002
10482 062022 012702 000200
10483 062026 020102
10484 062030 001406
10485 062032 005237 002212
10489 062036
      062036 104456
      062040 001176
      062042 073465
      062044 012166
10490 062046
      062046 104406
10491 062050 017701 121042

```

```

;*****
;
;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
      ERRHRD   ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP   C$ERHRD
                                .WORD  637
                                .WORD  T26BOT
                                .WORD  EXPREC
140$:  CKLOOP                  ;LOOP IF SELECTED
                                TRAP   C$CLP1
;*****
;
;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
;
;*****
      MOV      #1,R3           ;SET UP SPACE FORWARD 1
      JSR      PC,SPACE       ;ISSUE SPACE COMMAND
      MOV      #256,R3        ;RECORD SIZE
150$:  MOV      FREE,T26R8     ;STARTING READ BUFFER ADDRESS
;*****
;
;REREAD DATA,CVC=1,ACK, OPP COMMAND
;
;*****
165$:  MOV      #171001,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
      MOV      #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
      MOV      R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
      MOV      #SSR,R2        ;SET UP EXPECTED
      CMP      R1,R2         ;ARE THEY EQUAL
      BEQ      170$         ;BR, IF OK
      INC      FATFLG         ;BUMP COUNT
      ERRHRD   ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                                TRAP   C$ERHRD
                                .WORD  638
                                .WORD  T26RRF
                                .WORD  PKTSSR
170$:  CKLOOP                  ;LOOP IF SELECTED
                                TRAP   C$CLP1
      MOV      @FREE,R1       ;FIRST WORD FROM READ BUFFER

```


K2

TEST 6: REREADS

```

062342 001203 .WORD 643
062344 005056 .WORD WRTMSG
062346 012154 .WORD SFIMSG
10597 062350 264: CKLOOP ;LOOP IF SELECTED TRAP C4CLP1
062350 104406
10598
10599 ;*****
10600 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10601 ;
10602 ;*****
10603
10604
10605 062352 004737 011136 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
10606 062356 103413 BCS 304 ;BR, IF NO PROBLEM
10607 062360 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
10608 062364 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
10609 062370 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
10610 062372 005237 002212 INC FATFLG ;BUMP COUNT
10614 062376 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
062376 104456 TRAP C4ERHRD
062400 001204 .WORD 644
062402 074544 .WORD T26RWN
062404 012166 .WORD PKTSSR
10615 062406 304: CKLOOP ;LOOP IF SELECTED TRAP C4CLP1
062406 104406
10616
10617 ;*****
10618 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
10619 ;
10620 ;*****
10621
10622
10623 062410 013701 073110 MOV T26BFR+6,R1 ;PICK UP XST0
10624 062414 010102 MOV R1,R2 ;SET UP EXPECTED
10625 062416 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
10626 062422 020102 CMP R1,R2 ;DOES EXP = REC'D
10627 062424 001406 BEQ 404 ;BR, IF EQUAL (OK)
10628 062426 005237 002212 INC FATFLG ;BUMP COUNT
10632 062432 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
062432 104456 TRAP C4ERHRD
062434 001205 .WORD 645
062436 074255 .WORD T26BOT
062440 015614 .WORD EXPREC
10633 062442 404: CKLOOP ;LOOP IF SELECTED TRAP C4CLP1
062442 104406
10634 062444 012703 001000 MOV #512,R3 ;RECORD SIZE
10635 062450 013737 003116 073212 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
10636
10637 ;*****
10638 ;WRITE DATA,CVC=1,ACK COMMAND
10639 ;
10640 ;*****
10641
10642
10643 062456 012737 140005 073210 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
10644 062464 012704 073210 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
10645 062470 654:

```


L2

TEST 6: REREADS

```

10646 062470 010337 073216      MOV      R3,T26SZ          ;SET UP RECORD SIZE IN PACKET
10647 062474 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
10648 062500 004737 016370      JSR      PC,WAITF         ;WAIT FOR SSR TO SET
10649 062504 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
10650 062510 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED
10651 062514 020102                CMP      R1,R2           ;ARE THEY EQUAL
10652 062516 001406                BEQ      75$             ;BR, IF OK
10653 062520 005237 002212      INC      FATFLG          ;BUMP COUNT
10657 062524                ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    646
                                .WORD    WRERR
                                .WORD    PKTSSR
                                TRAP      C$CLP1
                                .WORD    647
                                .WORD    T26TRL
                                .WORD    PKTSSR
                                TRAP      C$CLP1
                                .WORD    648
                                .WORD    648
062524 104456
062526 001206
062530 005113
062532 012166
10658 062534                75$:   CKLOOP            ;LOOP IF SELECTED
062534 104406
10659 062536 005303                DEC      R3              ;SET RECORD SIZE TO 511.
10660 062540 013737 003116 073212      MOV      FREE,T26RB      ;STARTING READ BUFFER ADDRESS
10661
10662                ;*****
10663                ;
10664                ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
10665                ;
10666                ;*****
10667
10668 062546 012737 161001 073210      MOV      #161001,T26PK3  ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
10669 062554 012704 073210      165$:  MOV      #T26PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
10670 062560 010337 073216      MOV      R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
10671 062564 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
10672 062570 004737 016370      JSR      PC,WAITF         ;WAIT FOR SSR TO SET
10673 062574 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
10674 062600 012702 100204      MOV      #SSR!SC!BIT2,R2 ;SET UP EXPECTED
10675 062604 020102                CMP      R1,R2           ;ARE THEY EQUAL
10676 062606 001406                BEQ      170$            ;BR, IF OK
10677 062610 005237 002212      INC      FATFLG          ;BUMP COUNT
10681 062614                ERRHRD  ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                                TRAP      C$ERHRD
                                .WORD    647
                                .WORD    T26TRL
                                .WORD    PKTSSR
                                TRAP      C$CLP1
                                .WORD    648
                                .WORD    648
062614 104456
062616 001207
062620 075622
062622 012166
10682 062624                170$:  CKLOOP            ;LOOP IF SELECTED
062624 104406
                                TRAP      C$CLP1
                                .WORD    648
                                .WORD    648
10683
10684                ;*****
10685                ;
10686                ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
10687                ;
10688                ;*****
10689
10690 062626 013701 073110      MOV      T26BFR+6,R1     ;GET MESSAGE BUFFER
10691 062632 010102                MOV      R1,R2           ;SET UP EXPECTED
10692 062634 052702 010000      BIS      #BIT12,R2       ;SET THE RLL BIT IN EXPECTED
10693 062640 020102                CMP      R1,R2           ;ARE THEY EQUAL
10694 062642 001406                BEQ      180$            ;BR, IF EQUAL (ALL IS WELL)
10695 062644 005237 002212      INC      FATFLG          ;BUMP COUNT
10699 062650                ERRHRD  ERRNO,T26LON,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
                                TRAP      C$ERHRD
                                .WORD    648
                                .WORD    648
062650 104456
062652 001210

```


B3

TEST 6: REREADS

```

063122 005056
063124 012154
10803 063126 104406      264:  CKLOOP                ;LOOP IF SELECTED      .WORD  WRTMSG
                                ;                               .WORD  SFIMSG
                                TRAP    C4CLP1
10804
10805
10806      ;*****
10807      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10808      ;
10809      ;*****
10810
10811 063130 004737 011136      JSR    PC,REWIND        ;CALL TAPE REWIND COMMAND
10812 063134 103413              BCS    304              ;BR, IF NO PROBLEM
10813 063136 016501 000002      MOV    TSSR(R5),R1     ;GET TSSR
10814 063142 012702 000200      MOV    #SSR,R2        ;SET UP EXPECTED TSSR
10815 063146 010004              MOV    R0,R4          ;PACKET ADDRESS SET UP
10816 063150 005237 002212      INC    FATFLG         ;BUMP COUNT
10820 063154              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C4ERHRD
                                .WORD  653
                                .WORD  T26RWN
                                .WORD  PKTSSR
063154 104456
063156 001215
063160 074544
063162 012166
10821 063164 104406      304:  CKLOOP                ;LOOP IF SELECTED      .WORD  WRTMSG
                                ;                               .WORD  SFIMSG
                                TRAP    C4CLP1
10822
10823      ;*****
10824      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
10825      ;
10826      ;*****
10827
10828
10829 063166 013701 073110      MOV    T26BFR+6,R1    ;PICK UP XST0
10830 063172 010102              MOV    R1,R2          ;SET UP EXPECTED
10831 063174 052702 000002      BIS    #BIT1,R2       ;SET BOT BIT IN EXPECTED
10832 063200 020102              CMP    R1,R2          ;DOES EXP = REC'D
10833 063202 001406              BEQ    404            ;BR, IF EQUAL (OK)
10834 063204 005237 002212      INC    FATFLG         ;BUMP COUNT
10838 063210              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C4ERHRD
                                .WORD  654
                                .WORD  T26BOT
                                .WORD  EXPREC
063210 104456
063212 001216
063214 074255
063216 015614
10839 063220 104406      404:  CKLOOP                ;LOOP IF SELECTED      .WORD  WRTMSG
                                ;                               .WORD  SFIMSG
                                TRAP    C4CLP1
063220 104406
10840 063222 012703 000400      MOV    #256,R3        ;RECORD SIZE
10841 063226 013737 003116 073212  MOV    FREE,T26RB     ;STARTING WRITE BUFFER ADDRESS
10842
10843      ;*****
10844      ;WRITE DATA,CVC=1,ACK COMMAND
10845      ;
10846      ;*****
10847
10848
10849 063234 012737 140005 073210      MOV    #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
10850 063242 012704 073210      MOV    #T26PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
10851 063246
10852 063246 010337 073216      654:  MOV    R3,T26SZ     ;SET UP RECORD SIZE IN PACKET

```

C3

TEST 6: REREADS

```

10853 063252 010465 000000      MOV      R4,TSDB(R5)          ;ISSUE COMMAND
10854 063256 004737 016370      JSR      PC,WAITF           ;WAIT FOR SSR TO SET
10855 063262 016501 000002      MOV      TSSR(R5),R1        ;GET TSSR CONTENTS
10856 063266 012702 000200      MOV      #SSR,R2           ;SET UP EXPECTED
10857 063272 020102                CMP      R1,R2              ;ARE THEY EQUAL
10858 063274 001406                BEQ      75$                ;BR, IF OK
10859 063276 005237 002212      INC      FATFLG             ;BUMP COUNT
10863 063302                ERRHRD  ERRNO,WRERR,PKTSSR  ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    655
                                .WORD    WRERR
                                .WORD    PKTSSR
                                TRAP      C$CLP1
                                .WORD    656
                                .WORD    T26TRL
                                .WORD    PKTSSR
10864 063312 104456      75$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    657
                                .WORD    T26LOP
10865 063314 012703 001000      MOV      #512,R3           ;RECORD SIZE
10866 063320 013737 003116 073212  MOV      FREE,T26RB        ;STARTING READ BUFFER ADDRESS
10867
10868 ;*****
10869 ;
10870 ;REREAD PREVIOUS,ACK,CVC=1,OPP=1
10871 ;
10872 ;*****
10873
10874 063326 012737 161001 073210 165$:  MOV      #161001,T26PK3    ;REREAD PREVIOUS,ACK,CVC=1,OPP=1
10875 063334 012704 073210      MOV      #T26PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
10876 063340 010337 073216      MOV      R3,T26S2          ;SET UP RECORD SIZE IN PACKET
10877 063344 010465 000000      MOV      R4,TSDB(R5)        ;ISSUE COMMAND
10878 063350 004737 016370      JSR      PC,WAITF           ;WAIT FOR SSR TO SET
10879 063354 016501 000002      MOV      TSSR(R5),R1        ;GET TSSR CONTENTS
10880 063360 012702 100204      MOV      #SSR!SC!BIT2,R2   ;SET UP EXPECTED
10881 063364 020102                CMP      R1,R2              ;ARE THEY EQUAL
10882 063366 001406                BEQ      170$                ;BR, IF OK
10883 063370 005237 002212      INC      FATFLG             ;BUMP COUNT
10887 063374                ERRHRD  ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    656
                                .WORD    T26TRL
                                .WORD    PKTSSR
                                TRAP      C$CLP1
                                .WORD    657
                                .WORD    T26LOP
10888 063404 104406      170$: CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    657
                                .WORD    T26LOP
10889
10890 ;*****
10891 ;
10892 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
10893 ;
10894 ;*****
10895
10896 063406 013701 073110      MOV      T268FR-6,R1        ;GET MESSAGE BUFFER
10897 063412 010102      MOV      R1,R2             ;SET UP EXPECTED
10898 063414 052702 040000      BIS      #BIT14,R2          ;SET THE RLS BIT IN EXPECTED
10899 063420 020102                CMP      R1,R2              ;ARE THEY EQUAL
10900 063422 001406                BEQ      180$                ;BR, IF EQUAL (ALL IS WELL)
10901 063424 005237 002212      INC      FATFLG             ;BUMP COUNT
10905 063430                ERRHRD  ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
                                TRAP      C$ERHRD
                                .WORD    657
                                .WORD    T26LOP
10905 063430 104456
10905 063432 001221
10905 063434 075452

```

TEST 6: REREADS

```

10906 063436 015614                                .WORD  EXPREC
10906 063440                                1804:  CKLOOP                                TRAP  C4CLP1
10907 063440 104406                                MOV    T26BFR-4,R1      ;PICK UP RESIDUAL BYTE COUNTER
10908 063442 013701 073106                        MOV    #256.,R2        ;THIS SHOULD BE THE DIFFERENCE
10909 063446 012702 000400                        CMP    R1,R2           ;IS THE DIFFERENCE CORRECT
10910 063452 020102                                BEQ    1904             ;BR, IF CORRECT
10911 063454 001406                                INC    FATFLG           ;BUMP COUNT
10915 063456 005237 002212                        ERRHRD ERRNO,T26PBP,EXPREC ;RBPCT NOT CORRECT
10915 063462 104456                                TRAP  C4ERHRD
10915 063462 104456                                .WORD 658
10915 063464 001222                                .WORD T26PBP
10915 063466 075534                                .WORD EXPREC
10916 063470 015614
10916 063472                                1904:  CKLOOP                                ;LOOP IF SELECTED
10917 063472 104406                                TRAP  C4CLP1
10917 063474 012703 001000                        MOV    #512.,R3        ;RECORD SIZE
10918 063500 013737 003116 073212                MOV    FREE,T26RB      ;STARTING READ BUFFER ADDRESS
10919
10920 ;*****
10921 ;
10922 ;REREAD PREVIOUS,ACK,CVC=1,OPP=0
10923 ;
10924 ;*****
10925
10926 063506 012737 141001 073210                MOV    #141001,T26PK3  ;REREAD PREVIOUS,ACK,CVC=1,OPP=0
10927 063514 012704 073210                        MOV    #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
10928 063520 010337 073216                        MOV    R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
10929 063524 010465 000000                        MOV    R4,TSDB(R5)    ;ISSUE COMMAND
10930 063530 004737 016370                        JSR    PC,WAITF       ;WAIT FOR SSR TO SET
10931 063534 016501 000002                        MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
10932 063540 012702 100204                        MOV    #SSR!SC!BIT2,R2 ;SET UP EXPECTED
10933 063544 020102                                CMP    R1,R2           ;ARE THEY EQUAL
10934 063546 001406                                BEQ    2704             ;BR, IF OK
10935 063550 005237 002212                        INC    FATFLG           ;BUMP COUNT
10939 063554                                ERRHRD ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
10939 063554 104456                                TRAP  C4ERHRD
10939 063556 001223                                .WORD 659
10939 063560 075622                                .WORD T26TRL
10939 063562 012166                                .WORD PKTSSR
10940 063564                                2704:  CKLOOP                                ;LOOP IF SELECTED
10940 063564 104406                                TRAP  C4CLP1
10941
10942 ;*****
10943 ;
10944 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
10945 ;
10946 ;*****
10947
10948 063566 013701 073110                        MOV    T26BFR-6,R1    ;GET MESSAGE BUFFER
10949 063572 010102                                MOV    R1,R2           ;SET UP EXPECTED
10950 063574 052702 040000                        BIS    #BIT14,R2      ;SET THE RLS BIT IN EXPECTED
10951 063600 020102                                CMP    R1,R2           ;ARE THEY EQUAL
10952 063602 001406                                BEQ    2804             ;BR, IF EQUAL (ALL IS WELL)
10953 063604 005237 002212                        INC    FATFLG           ;BUMP COUNT
10957 063610                                ERRHRD ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
10957 063610 104456                                TRAP  C4ERHRD
10957 063612 001224                                .WORD 660

```


F3

TEST 6: REREADS

```

11006 063712 103407          BCS      20$          ;BR IF INIT WAS OK
11007 063714 005237 002212  INC      FATFLG      ;BUMP COUNT
11011 063720 010001          MOV      R0,R1       ;CONTENTS OF TSSR REGISTER
11012 063722          ERRDF    ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                TRAP      C$ERDF
                                .WORD    661
                                .WORD    SFIERR
                                .WORD    SFIMSG
11013 063732 013737 002172 073100 20$:  MOV      UNITN,T26DSW ;SET UP UNIT NUMBER
11014
11015 063740 012704 073060          MOV      @T26PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
11016
11017 ;*****
11018 ;
11019 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
11020 ;
11021 ;*****
11022
11023 063744 004737 010752          JSR      PC,WRTCHR    ;ISSUE WRITE CHARACTERISTICS
11024 063750 103407          BCS      26$          ;BR, IF COMMAND ISSUED OK
11025 063752 005237 002212  INC      FATFLG      ;BUMP COUNT
11029 063756 010001          MOV      R0,R1       ;SAVE CONTENTS OF TSSR
11030 063760          ERRHRD   ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP      C$ERHRD
                                .WORD    662
                                .WORD    WRTMSG
                                .WORD    SFIMSG
11031 063770          26$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
11032
11033 ;*****
11034 ;
11035 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
11036 ;
11037 ;*****
11038
11039 063772 004737 011136          JSR      PC,REWIND    ;CALL TAPE REWIND COMMAND
11040 063776 103413          BCS      30$          ;BR, IF NO PROBLEM
11041 064000 016501 000002  MOV      TSSR(R5),R1 ;GET TSSR
11042 064004 012702 000200  MOV      @SSR,R2     ;SET UP EXPECTED TSSR
11043 064010 010004          MOV      R0,R4       ;PACKET ADDRESS SET UP
11044 064012 005237 002212  INC      FATFLG      ;BUMP COUNT
11048 064016          ERRHRD   ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    663
                                .WORD    T26RWN
                                .WORD    PKTSSR
11049 064026          30$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
11050
11051 ;*****
11052 ;
11053 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
11054 ;
11055 ;*****
11056
11057 064030 013701 073110          MOV      T26BFR-6,R1 ;PICK UP XSTO

```


TEST 6: REREADS

```

11058 064034 010102          MOV    R1,R2          ;SET UP EXPECTED
11059 064036 052702 000002  BIS    #BIT1,R2       ;SET BOT BIT IN EXPECTED
11060 064042 020102          CMP    R1,R2         ;DOES EXP = REC'D
11061 064044 001406          BEQ    40$           ;BR, IF EQUAL (OK)
11062 064046 005237 002212  INC    FATFLG        ;BUMP COUNT
11066 064052          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD   664
                                .WORD   T26BOT
                                .WORD   EXPREC
                                TRAP    C$CLP1
                                .WORD   104456
                                .WORD   001230
                                .WORD   074255
                                .WORD   015614
11067 064062          40$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
11068 064064 012703 000400  MOV    #256.,R3      ;RECORD SIZE
11069 064070 013737 003116 073212  MOV    FREE,T26RB    ;STARTING WRITE BUFFER ADDRESS
11070
11071 ;*****
11072 ;
11073 ;WRITE DATA,CVC=1,ACK COMMAND
11074 ;
11075 ;*****
11076
11077 064076 012737 140005 073210  MOV    #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
11078 064104 012704 073210  MOV    #T26PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
11079 064110          65$:
11080 064110 010300          MOV    R3,R0         ;SET PATTERN IN CORRECT REGISTER
11081 064112 004737 017542  JSR    PC,FILLMEM    ;FILL MEMORY WITH RECORD SIZE
11082 064116 010337 073216  MOV    R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
11083 064122 010465 000000  MOV    R4,TSDB(R5)   ;ISSUE COMMAND
11084 064126 004737 016370  JSR    PC,WAITF      ;WAIT FOR SSR TO SET
11085 064132 016501 000002  MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
11086 064136 012702 000200  MOV    #SSR,R2       ;SET UP EXPECTED
11087 064142 020102          CMP    R1,R2         ;ARE THEY EQUAL
11088 064144 001406          BEQ    75$           ;BR, IF OK
11089 064146 005237 002212  INC    FATFLG        ;BUMP COUNT
11093 064152          ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERHRD
                                .WORD   665
                                .WORD   WRERR
                                .WORD   PKTSSR
                                TRAP    C$CLP1
                                .WORD   104456
                                .WORD   001231
                                .WORD   005113
                                .WORD   012166
11094 064162          75$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
11095 064164 005723          TST    (R3)+         ;BUMP RECORD SIZE
11096 064166 022703 000414  CMP    #268.,R3     ;END OF RECORD YET
11097 064172 001346          BNE    65$           ;BR, IF MORE RECORDS TO WRITE
11098 064174          80$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
11099 064176          120$:
11100 ;*****
11101 ;
11102 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
11103 ;
11104 ;*****
11105
11106
11107 064176 004737 011136  JSR    PC,REWIND     ;CALL TAPE REWIND COMMAND
11108 064202 103413          BCS    130$          ;BR, IF NO PROBLEM
11109 064204 016501 000002  MOV    TSSR(R5),R1   ;GET TSSR

```

H3

TEST 6: REREADS

```

11110 064210 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED TSSR
11111 064214 010004              MOV      R0,R4      ;PACKET ADDRESS SET UP
11112 064216 005237 002212      INC      FATFLG      ;BUMP COUNT
11116 064222              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C#ERHRD
                                .WORD    666
                                .WORD    T26RWN
                                .WORD    PKTSSR
11117 064232 104406      130$:  CKLOOP      ;LOOP IF SELECTED
                                TRAP      C#CLP1
11118
11119      ;*****
11120      ;
11121      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
11122      ;
11123      ;*****
11124
11125 064234 013701 073110      MOV      T26BFR+6,R1 ;PICK UP XSTO
11126 064240 010102              MOV      R1,R2      ;SET UP EXPECTED
11127 064242 052702 000002      BIS      #BIT1,R2    ;SET BOT BIT IN EXPECTED
11128 064246 020102              CMP      R1,R2      ;DOES EXP = REC'D
11129 064250 001406              BEQ     140$        ;BR, IF EQUAL (OK)
11130 064252 005237 002212      INC      FATFLG      ;BUMP COUNT
11134 064256              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C#ERHRD
                                .WORD    667
                                .WORD    T26BOT
                                .WORD    EXPREC
11135 064266              140$:  CKLOOP      ;LOOP IF SELECTED
                                TRAP      C#CLP1
11136 064270 012737 000400 073242  MOV      #256.,T26RSZ ;STORE START RECORD SIZE
11137 064276 000420              BR      150$        ;SKIP THE SAPCE THIS TIME
11138
11139      ;*****
11140      ;
11141      ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
11142      ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
11143      ;
11144      ;*****
11145
11146 064300 012703 000001      145$:  MOV      #1,R3    ;SPACE ONE RECORD PARAMETER
11147 064304 004737 010560      JSR     PC,SPACE    ;CALL SPACE ROUTINE
11148 064310 103413              BCS     150$        ;BR, IF NO PROBLEM WITH SPACE COMMAND
11149 064312 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR
11150 064316 012702 000200      MOV      #SSR,R2    ;SET UP EXPECTED TSSR
11151 064322 010004              MOV      R0,R4      ;PACKET ADDRESS SET UP
11152 064324 005237 002212      INC      FATFLG      ;BUMP COUNT
11156 064330              ERRHRD  ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
                                TRAP      C#ERHRD
                                .WORD    668
                                .WORD    T26SC
                                .WORD    EXPREC
11157 064340              150$:  CKLOOP
                                TRAP      C#CLP1
11158 064342 013703 073242      MOV      T26RSZ,R3   ;RECORD SIZE
11159 064346 013737 003116 073212  MOV      FREE,T26RB  ;STARTING READ BUFFER ADDRESS
11160

```


K3

TEST 6: REREADS

```

11263 ;*****
11264 ;
11265 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
11266 ;
11267 ;*****
11268
11269 064644 004737 011136          JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
11270 064650 103413          BCS      30$                ;BR, IF NO PROBLEM
11271 064652 016501 000002        MOV      TSSR(R5),R1        ;GET TSSR
11272 064656 012702 000200        MOV      #SSR,R2           ;SET UP EXPECTED TSSR
11273 064662 010004          MOV      R0,R4              ;PACKET ADDRESS SET UP
11274 064664 005237 002212        INC      FATFLG             ;BUMP COUNT
11278 064670          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C#ERHRD
                                .WORD    673
                                .WORD    T26RWN
                                .WORD    PKTSSR
                                TRAP      C#CLP1
064670 104456
064672 001241
064674 074544
064676 012166
11279 064700          30$:   CKLOOP                ;LOOP IF SELECTED
064700 104406          TRAP      C#CLP1
11280
11281 ;*****
11282 ;
11283 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
11284 ;
11285 ;*****
11286
11287 064702 013701 073110          MOV      T26BFR+6,R1        ;PICK UP XSTO
11288 064706 010102          MOV      R1,R2              ;SET UP EXPECTED
11289 064710 052702 000002        BIS      #BIT1,R2           ;SET BOT BIT IN EXPECTED
11290 064714 020102          CMP      R1,R2              ;DOES EXP = REC'D
11291 064716 001406          BEQ      40$                ;BR, IF EQUAL (OK)
11292 064720 005237 002212        INC      FATFLG             ;BUMP COUNT
11296 064724          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C#ERHRD
                                .WORD    674
                                .WORD    T26BOT
                                .WORD    EXPREC
                                TRAP      C#CLP1
064724 104456
064726 001242
064730 074255
064732 015614
11297 064734          40$:   CKLOOP                ;LOOP IF SELECTED
064734 104406          TRAP      C#CLP1
11298 064736 012703 000400          MOV      #256.,R3           ;RECORD SIZE
11299 064742 013737 003116 073212    MOV      FREE,T26RB         ;STARTING WRITE BUFFER ADDRESS
11300
11301 ;*****
11302 ;
11303 ;WRITE DATA,CVC-1,ACK,SWB COMMAND
11304 ;
11305 ;*****
11306
11307 064750 012737 150005 073210    MOV      #150005,T26PK3     ;WRITE DATA,CVC-1,ACK,SWB COMMAND
11308 064756 012704 073210          MOV      #T26PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
11309 064762
11310 064762 010300          65$:   MOV      R3,R0         ;SET PATTERN IN CORRECT REGISTER
11311 064764 004737 017542          JSR      PC,FILLMEM         ;FILL MEMORY WITH RECORD SIZE
11312 064770 010337 073216          MOV      R3,T26SZ          ;SET UP RECORD SIZE IN PACKET
11313 064774 010465 000000          MOV      R4,TSDB(R5)       ;ISSUE COMMAND
11314 065000 004737 016370          JSR      PC,WAITF          ;WAIT FOR SSR TO SET
11315 065004 016501 000002          MOV      TSSR(R5),R1       ;GET TSSR CONTENTS

```

L3

TEST 6: REREADS

```

11316 065010 012702 000200      MOV    #SSR,R2          ;SET UP EXPECTED
11317 065014 020102             CMP    R1,R2           ;ARE THEY EQUAL
11318 065016 001406             BEQ    75$             ;BR, IF OK
11319 065020 005237 002212      INC    FATFLG          ;BUMP COUNT
11323 065024             ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERHRD
                                .WORD  675
                                .WORD  WRERR
                                .WORD  PKTSSR
                                104456
                                065026 001243
                                065030 005113
                                065032 012166
11324 065034             75$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
11325 065036 104406             TST    (R3)+          ;BUMP RECORD SIZE
11326 065040 005723             CMP    #268.,R3      ;END OF RECORD YET
11327 065044 001346             BNE    65$           ;BR, IF MORE RECORDS TO WRITE
11328 065046             80$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                065046 104406
11329 065050             120$:
11330
11331 ;*****
11332 ;
11333 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
11334 ;
11335 ;*****
11336
11337 065050 004737 011136      JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
11338 065054 103413             BCS    130$          ;BR, IF NO PROBLEM
11339 065056 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR
11340 065062 012702 000200      MOV    #SSR,R2      ;SET UP EXPECTED TSSR
11341 065066 010004             MOV    R0,R4         ;PACKET ADDRESS SET UP
11342 065070 005237 002212      INC    FATFLG        ;BUMP COUNT
11346 065074             ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD  676
                                .WORD  T26RWN
                                .WORD  PKTSSR
                                065074 104456
                                065076 001244
                                065100 074544
                                065102 012166
11347 065104             130$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                065104 104406
11348
11349 ;*****
11350 ;
11351 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
11352 ;
11353 ;*****
11354
11355 065106 013701 073110      MOV    T26BFR+6,R1   ;PICK UP XSTO
11356 065112 010102             MOV    R1,R2         ;SET UP EXPECTED
11357 065114 052702 000002      BIS    #BIT1,R2      ;SET BOT BIT IN EXPECTED
11358 065120 020102             CMP    R1,R2         ;DOES EXP = REC'D
11359 065122 001406             BEQ    140$          ;BR, IF EQUAL (OK)
11360 065124 005237 002212      INC    FATFLG        ;BUMP COUNT
11364 065130             ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD  677
                                .WORD  T26BOT
                                .WORD  EXPREC
                                065130 104456
                                065132 001245
                                065134 074255
                                065136 015614
11365 065140             140$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                065140 104406

```


C4

TEST 6: REREADS

```

11523 065534 012702 000200          MOV    #SSR,R2          ;SET UP EXPECTED TSSR
11524 065540 010004          MOV    R0,R4           ;PACKET ADDRESS SET UP
11525 065542 005237 002212          INC    FATFLG          ;BUMP COUNT
11529 065546          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      065546 104456          TRAP   C#ERHRD
      065550 001253          .WORD 683
      065552 074544          .WORD T26RWN
      065554 012166          .WORD PKTSSR
11530 065556          304:  CKLOOP          ;LOOP IF SELECTED
      065556 104406          TRAP   C#CLP1

11531          ;*****
11532          ;
11533          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
11534          ;
11535          ;*****
11536          ;
11537          ;
11538 065560 013701 073110          MOV    T26BFR+6,R1     ;PICK UP XSTO
11539 065564 010102          MOV    R1,R2           ;SET UP EXPECTED
11540 065566 052702 000002          BIS    #BIT1,R2        ;SET BOT BIT IN EXPECTED
11541 065572 020102          CMP    R1,R2           ;DOES EXP = REC'D
11542 065574 001406          BEQ    40$             ;BR, IF EQUAL (OK)
11543 065576 005237 002212          INC    FATFLG          ;BUMP COUNT
11547 065602          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      065602 104456          TRAP   C#ERHRD
      065604 001254          .WORD 684
      065606 074255          .WORD T26BOT
      065610 015614          .WORD EXPREC
11548 065612          404:  CKLOOP          ;LOOP IF SELECTED
      065612 104406          TRAP   C#CLP1
11549 065614 012703 000400          MOV    #256.,R3        ;RECORD SIZE
11550 065620 013737 003116 073212          MOV    FREE,T26R8      ;STARTING WRITE BUFFER ADDRESS
11551          ;*****
11552          ;
11553          ;WRITE DATA,CVC=1,ACK COMMAND
11554          ;
11555          ;*****
11556          ;
11557          ;
11558 065626 012737 140005 073210          MOV    #140005,T26PK3  ;WRITE DATA,CVC=1,ACK COMMAND
11559 065634 012704 073210          MOV    #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
11560 065640          654:
11561 065640 010337 073216          MOV    R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
11562 065644 013777 073236 115244          MOV    T26CNT,#FREE    ;MOVE TAPE RECORD NUMBER TO BUFFER
11563 065652 062737 000001 073236          ADD    #1,T26CNT       ;NUMBER READY FOR NEXT RECORD
11564 065660 010465 000000          MOV    R4,TSDB(R5)     ;ISSUE COMMAND
11565 065664 004737 016370          JSR    PC,WAITF        ;WAIT FOR SSR TO SET
11566 065670 016501 000002          MOV    TSSR(R5),R1     ;GET TSSR CONTENTS
11567 065674 012702 000200          MOV    #SSR,R2        ;SET UP EXPECTED
11568 065700 020102          CMP    R1,R2           ;ARE THEY EQUAL
11569 065702 001406          BEQ    75$             ;BR, IF OK
11570 065704 005237 002212          INC    FATFLG          ;BUMP COUNT
11574 065710          ERRHRD  EPRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      065710 104456          TRAP   C#ERHRD
      065712 001255          .WORD 685
      065714 005113          .WORD WRERR
      065716 012166          .WORD PKTSSR

```

D4

TEST 6: REREADS

```

11575 065720          75$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C4CLP1
      065720 104406
11576 065722 005723          TST      (R3)+          ;BUMP THE RECORD SIZE
11577 065724 022703 000414    CMP      #268.,R3      ;MAXIMUM SIZE YET
11578 065730 001401          BEQ      120$          ;BR, IF AT END OF WRITE SEQUENCE
11579 065732 000742          BR       65$          ;WRITE MORE RECORDS
11580 065734          120$:  CLR      T26CNT          ;SET RECORD COUNTER BACK TO ZERO
11581 065734 005037 073236
11582
11583          ;*****
11584          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
11585          ;
11586          ;*****
11587
11588          JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
11589 065740 004737 011136    BCS      130$          ;BR, IF NO PROBLEM
11590 065744 103411          MOV      TSSR(R5),R1  ;GET TSSR
11591 065746 016501 000002    MOV      R0,R4        ;PACKET ADDRESS SET UP
11592 065752 010004          INC     FATFLG        ;BUMP COUNT
11593 065754 005237 002212    ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
11597 065760          .WORD  686
      065760 104456          .WORD  T26RWN
      065762 001256          .WORD  PKTSSR
      065764 074544
      065766 012166
11598 065770          130$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C4CLP1
      065770 104406
11599
11600          ;*****
11601          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
11602          ;
11603          ;*****
11604
11605          MOV      T268FR-6,R1        ;PICK UP XST0
11606 065772 013701 073110    MOV      R1,R2        ;SET UP EXPECTED
11607 065776 010102          BIS     #BIT1,R2      ;SET BOT BIT IN EXPECTED
11608 066000 052702 000002    CMP     R1,R2        ;DOES EXP = REC'D
11609 066004 020102          BEQ     135$          ;BR, IF EQUAL (OK)
11610 066006 001406          INC     FATFLG        ;BUMP COUNT
11611 066010 005237 002212    ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
11615 066014          .WORD  687
      066014 104456          .WORD  T26BOT
      066016 001257          .WORD  EXPREC
      066020 074255
      066022 015614
11616 066024          135$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C4CLP1
      066024 104406
11617 066026 012737 000400 073242 MOV     #256.,T26RSZ  ;STARTING RECORD SIZE
11618 066034 000420          BR      140$          ;SKIP OVER THE SAPCE THIS TIME
11619
11620          ;*****
11621          ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
11622          ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
11623          ;
11624          ;*****
11625
11626

```

E4

TEST 6: REREADS

```

11627 066036 012703 000001      1324:  MOV    #000001,R3      ;SET UP SPACE COMMAND (1 FORWARD)
11628 066042 004737 010560      JSR    PC,SPACE          ;CALL SPACE ROUTINE
11629 066046 103413                BCS    1404              ;BR, IF NO TROUBLE
11630 066050 016501 000002      MOV    TSSR(R5),R1      ;GET TSSR
11631 066054 012702 000200      MOV    #SSR,R2          ;SET UP EXPECTED TSSR
11632 066060 010004                MOV    R0,R4            ;PACKET ADDRESS SET UP
11633 066062 005237 002212      INC    FATFLG           ;BUMP COUNT
11637 066066                ERRHRD  ERRNO,T26SC,PKTSSR ;SPACE (FORWARD) FAILED
                                TRAP    C4ERHRD
                                .WORD   688
                                .WORD   T26SC
                                .WORD   PKTSSR
                                TRAP    C4CLP1
11638 066076                1404:  CKLOOP              ;LOOP IF SELECTED
                                TRAP    C4CLP1
11639 066100 013703 073242      MOV    T26RSZ,R3        ;RECORD SIZE
11640 066104 013737 003116 073212 1504:  MOV    FREE,T26RB       ;STARTING READ BUFFER ADDRESS
11641
11642 ;*****
11643 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
11644 ;
11645 ;*****
11646
11647
11648 066112 012737 161401 073210 1654:  MOV    #161401,T26PK3   ;REREAD DATA,CVC=1,ACK, OPP COMMAND
11649 066120 012704 073210      MOV    #T26PK3,R4       ;SET UP R4 WITH PACKET ADDRESS
11650 066124 010337 073216      MOV    R3,T26SZ         ;SET UP RECORD SIZE IN PACKET
11651 066130 010465 000000      MOV    R4,TSDB(R5)      ;ISSUE COMMAND
11652 066134 004737 016370      JSR    PC,WAITF         ;WAIT FOR SSR TO SET
11653 066140 016501 000002      MOV    TSSR(R5),R1      ;GET TSSR CONTENTS
11654 066144 012702 000200      MOV    #SSR,R2          ;SET UP EXPECTED
11655 066150 020102                CMP    R1,R2            ;ARE THEY EQUAL
11656 066152 001406                BEQ    1704              ;BR, IF OK
11657 066154 005237 002212      INC    FATFLG           ;BUMP COUNT
11661 066160                ERRHRD  ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                                TRAP    C4ERHRD
                                .WORD   689
                                .WORD   T26RRF
                                .WORD   PKTSSR
11662 066170                1704:  CKLOOP              ;LOOP IF SELECTED
                                TRAP    C4CLP1
11663 066170 104406                MOV    #FREE,R1         ;FIRST WORD FROM READ BUFFER
11664 066176 017701 114720      MOV    T26CNT,R2        ;SET UP EXPECTED
11665 066202 020102                CMP    R1,R2            ;IS TAPE POSITION CORRECT
11666 066204 001406                BEQ    1904              ;KEEP GOING POSITION OK
11667 066206 005237 002212      INC    FATFLG           ;BUMP COUNT
11671 066212                ERRHRD  ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
                                TRAP    C4ERHRD
                                .WORD   690
                                .WORD   T26WNG
                                .WORD   EXPREC
11672 066222                1904:  CKLOOP              ;LOOP IF SELECTED
                                TRAP    C4CLP1
11673 066224 104406                ADD    #1,T26CNT        ;BUMP TAPE RECORD COUNTER
11674 066232 062737 000001 073236      TST    (R3)             ;NEXT RECORD SIZE
11675 066234 010337 073242      MOV    R3,T26RSZ       ;STORE RECORD SIZE
11676 066240 022703 000412      CMP    #266.,R3        ;AT MAX SIZE YET
11677 066244 001406                BEQ    2004              ;BR, IF AT END OF THE SUBTEST

```


G4

TEST 6: REREADS

```

11730
11731 066350 004737 010752          JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
11732 066354 103407                    BCS      26$                ;BR, IF COMMAND ISSUED OK
11733 066356 005237 002212          INC      FATFLG            ;BUMP COUNT
11737 066362 010001                    MOV      RO,R1             ;SAVE CONTENTS OF TSSR
11738 066364                    ERRHRD  ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICSC FAILED
                                TRAP      C$ERHRD
                                .WORD    692
                                .WORD    WRTMSG
                                .WORD    SFMSG
                                066364 104456
                                066366 001264
                                066370 005056
                                066372 012154
11739 066374                    26$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                066374 104406
11740
11741 ;*****
11742 ;
11743 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
11744 ;
11745 ;*****
11746
11747 066376 004737 011136          JSR      PC,REWIND         ;CALL TAPE REWIND COMMAND
11748 066402 016501 000002          MOV      TSSR(R5),R1      ;GET TSSR
11749 066406 012702 000200          MOV      #SSR,R2         ;SET UP EXPECTED TSSR
11750 066412 103407                    BCS      30$                ;BR, IF NO PROBLEM
11751 066414 010004                    MOV      RO,R4            ;PACKET ADDRESS SET UP
11752 066416 005237 002212          INC      FATFLG            ;BUMP COUNT
11756 066422                    ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    693
                                .WORD    T26RWN
                                .WORD    PKTSSR
                                066422 104456
                                066424 001265
                                066426 074544
11757 066430 012166                    30$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                066432 104406
11758
11759 ;*****
11760 ;
11761 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
11762 ;
11763 ;*****
11764
11765 066434 013701 073110          MOV      T26BFR+6,R1      ;PICK UP XST0
11766 066440 010102                    MOV      R1,R2            ;SET UP EXPECTED
11767 066442 052702 000002          BIS      #BIT1,R2         ;SET BOT BIT IN EXPECTED
11768 066446 020102                    CMP      R1,R2            ;DOES EXP = REC'D
11769 066450 001406                    BEQ      40$                ;BR, IF EQUAL (OK)
11770 066452 005237 002212          INC      FATFLG            ;BUMP COUNT
11774 066456                    ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    694
                                .WORD    T26BOT
                                .WORD    EXPREC
                                066456 104456
                                066460 001266
                                066462 074255
                                066464 015614
11775 066466                    40$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                066466 104406
11776 066470 012703 000400          MOV      #256.,R3         ;RECORD SIZE
11777 066474 013737 003116 073212  MOV      FREE,T26RB       ;STARTING WRITE BUFFER ADDRESS
11778
11779 ;*****
11780 ;

```

TEST 6: REREADS

```

11781 ;WRITE DATA,CVC=1,ACK COMMAND
11782 ;
11783 ;*****
11784 ;
11785 066502 012737 140005 073210      MOV      #140005,T26PK3      ;WRITE DATA,CVC=1,ACK COMMAND
11786 066510 012704 073210      MOV      #T26PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
11787 066514      65$:
11788 066514 010337 073216      MOV      R3,T26SZ          ;SET UP RECORD SIZE IN PACKET
11789 066520 013777 073236 114370  MOV      T26CNT,@FREE      ;MOVE TAPE RECORD NUMBER TO BUFFER
11790 066526 062737 000001 073236  ADD      #1,T26CNT        ;NUMBER READY FOR NEXT RECORD
11791 066534 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
11792 066540 004737 016370      JSR      PC,WAITF         ;WAIT FOR SSR TO SET
11793 066544 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
11794 066550 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED
11795 066554 020102      CMP      R1,R2           ;ARE THEY EQUAL
11796 066556 001406      BEQ      75$             ;BR, IF OK
11797 066560 005237 002212      INC      FATFLG          ;BUMP COUNT
11801 066564      ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    695
                                .WORD    WRTERR
                                .WORD    PKTSSR
                                TRAP      C$CLP1
11802 066574      75$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406
11803 066576 005723      TST      (R3)+           ;BUMP THE RECORD SIZE
11804 066600 022703 000414      CMP      #268.,R3       ;MAXIMUM SIZE YET
11805 066604 001401      BEQ      120$           ;BR, IF AT END OF WRITE SEQUENCE
11806 066606 000742      BR       65$            ;WRITE MORE RECORDS
11807 066610      120$:
11808 066610 005037 073236      CLR      T26CNT          ;SET RECORD COUNTER BACK TO ZERO
11809
11810 ;*****
11811 ;
11812 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
11813 ;
11814 ;*****
11815 ;
11816 066614 004737 011136      JSR      PC,REWIND        ;CALL TAPE REWIND COMMAND
11817 066620 103411      BCS     130$            ;BR, IF NO PROBLEM
11818 066622 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR
11819 066626 010004      MOV      R0,R4          ;PACKET ADDRESS SET UP
11820 066630 005237 002212      INC      FATFLG          ;BUMP COUNT
11824 066634      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    696
                                .WORD    T26RWN
                                .WORD    PKTSSR
                                TRAP      C$CLP1
11825 066644      130$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406
11826
11827 ;*****
11828 ;
11829 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
11830 ;
11831 ;*****
11832 ;
11833 066646 013701 073110      MOV      T26BFR+6,R1     ;PICK UP XST0

```

TEST 6: REREADS

```

11834 066652 010102          MOV    R1,R2          ;SET UP EXPECTED
11835 066654 052702 000002    BIS    #BIT1,R2       ;SET BOT BIT IN EXPECTED
11836 066660 020102          CMP    R1,R2          ;DOES EXP = REC'D
11837 066662 001406          BEQ    135$           ;BR, IF EQUAL (OK)
11838 066664 005237 002212    INC    FATFLG         ;BUMP COUNT
11842 066670          ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD  697
                                .WORD  T26BOT
                                .WORD  EXPREC
                                TRAP    C$CLP1
                                .WORD  104456
                                .WORD  001271
                                .WORD  074255
                                .WORD  015614
11843 066700          135$: CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD  066700
                                .WORD  104406
11844 066702 012737 000400 073242    MOV    #256.,T26RSZ   ;START RECORD SIZE
11845 066710 000420          BR     140$           ;SKIP OVER SPACE
11846
11847 ;*****
11848 ;
11849 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
11850 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
11851 ;
11852 ;*****
11853
11854 066712 012703 000001    136$: MOV    #000001,R3 ;SET UP SPACE COMMAND (1 FORWARD)
11855 066716 004737 010560    JSR    PC,SPACE       ;CALL SPACE ROUTINE
11856 066722 103413          BCS    140$           ;BR, IF NO TROUBLE
11857 066724 016501 000002    MOV    TSSR(R5),R1    ;GET TSSR
11858 066730 012702 000200    MOV    #SSR,R2        ;SET UP EXPECTED TSSR
11859 066734 010004          MOV    R0,R4          ;PACKET ADDRESS SET UP
11860 066736 005237 002212    INC    FATFLG         ;BUMP COUNT
11864 066742          ERRHRD ERRNO,T26SC,PKTSSR ;SPACE (FORWARD) FAILED
                                TRAP    C$ERHRD
                                .WORD  698
                                .WORD  T26SC
                                .WORD  PKTSSR
                                TRAP    C$CLP1
                                .WORD  066742
                                .WORD  104456
                                .WORD  001272
                                .WORD  073657
                                .WORD  012166
11865 066752          140$: CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD  066752
                                .WORD  104406
11866 066754 013703 073242          150$: MOV    T26RSZ,R3 ;RECORD SIZE
11867 066760 013737 003116 073212    MOV    FREE,T26RB     ;STARTING READ BUFFER ADDRESS
11868
11869 ;*****
11870 ;
11871 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
11872 ;
11873 ;*****
11874
11875 066766 012737 161401 073210          165$: MOV    #161401,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
11876 066774 012704 073210          MOV    #T26PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
11877 067000 010337 073216          MOV    R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
11878 067004 010465 000000          MOV    R4,TSD8(R5)   ;ISSUE COMMAND
11879 067010 004737 016370          JSR    PC,WAITF       ;WAIT FOR SSR TO SET
11880 067014 016501 000002          MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
11881 067020 012702 000200          MOV    #SSR,R2        ;SET UP EXPECTED
11882 067024 020102          CMP    R1,R2          ;ARE THEY EQUAL
11883 067026 001406          BEQ    170$           ;BR, IF OK
11884 067030 005237 002212          INC    FATFLG         ;BUMP COUNT
11888 067034          ERRHRD ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                                TRAP    C$ERHRD
                                .WORD  067034
                                .WORD  104456

```


TEST 6: REREADS

```

11936
11937
11938
11939 067162 004737 016114
11940 067166 103407
11941 067170 005237 002212
11945 067174 010001
11946 067176
      067176 104455
      067200 001275
      067202 003652
      067204 012154
11947 067206 013737 002172 073100 20+: MOV UNITN,T26DSW ;SET UP UNIT NUMBER
11948
11949 067214 012704 073060 MOV #T26PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
11950
11951
11952
11953 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
11954
11955
11956
11957 067220 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
11958 067224 103407 BCS 26: ;BR, IF COMMAND ISSUED OK
11959 067226 005237 002212 INC FATFLG ;BUMP COUNT
11963 067232 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
11964 067234 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      067234 104456 TRAP C#ERHRD
      067236 001276 .WORD 702
      067240 005056 .WORD WRTMSG
      067242 012154 .WORD SFIMSG
11965 067244 26+: CKLOOP ;LOOP IF SELECTED
      067244 104406 TRAP C#CLP1
11966
11967
11968
11969 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
11970
11971
11972
11973 067246 004737 011136 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
11974 067252 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
11975 067256 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
11976 067262 103407 BCS 30: ;BR, IF NO PROBLEM
11977 067264 010004 MOV RO,R4 ;PACKET ADDRESS SET UP
11978 067266 005237 002212 INC FATFLG ;BUMP COUNT
11982 067272 ERRHRD ERRNO,T26RMN,PKTSSR ;REWIND NOT ACCEPTED
      067272 104456 TRAP C#ERHRD
      067274 001277 .WORD 703
      067276 074544 .WORD T26RMN
      067300 012166 .WORD PKTSSR
11983 067302 30+: CKLOOP ;LOOP IF SELECTED
      067302 104406 TRAP C#CLP1
11984
11985
11986
11987 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)

```

TEST 6: REREADS

```

11988
11989
11990
11991 067304 013701 07311C      MOV      T26BFR+6,R1      ;PICK UP XSTO
11992 067310 010102      MOV      R1,R2           ;SET UP EXPECTED
11993 067312 052702 000002      BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
11994 067316 020102      CMP      R1,R2           ;DOES EXP = REC'D
11995 067320 001406      BEQ      40$             ;BR, IF EQUAL (OK)
11996 067322 005237 002212      INC      FATFLG          ;BUMP COUNT
12000 067326      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    704
                                .WORD    T26BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
12001 067336      40$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
12002 067340 012703 001000      MOV      @512.,R3        ;RECORD SIZE
12003 067344 013737 003116 073212      MOV      FREE,T26RB      ;STARTING WRITE BUFFER ADDRESS
12004
12005
12006
12007
12008
12009
12010
12011 067352 012737 140005 073210      MOV      @140005,T26PK3  ;WRITE DATA,CVC=1,ACK COMMAND
12012 067360 012704 073210      MOV      @T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
12013 067364
12014 067364 010337 073216      65$:  MOV      R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
12015 067370 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
12016 067374 004737 016370      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
12017 067400 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
12018 067404 012702 000200      MOV      @SSR,R2        ;SET UP EXPECTED
12019 067410 020102      CMP      R1,R2           ;ARE THEY EQUAL
12020 067412 001406      BEQ      75$             ;BR, IF OK
12021 067414 005237 002212      INC      FATFLG          ;BUMP COUNT
12025 067420      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    705
                                .WORD    WRTErr
                                .WORD    PKTSSR
12026 067430      75$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
12027
12028
12029
12030
12031
12032
12033
12034 067432 004737 011136      JSR      PC,REWIND       ;CALL TAPE REWIND COMMAND
12035 067436 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR
12036 067442 012702 000200      MOV      @SSR,R2        ;SET UP EXPECTED TSSR
12037 067446 103407      BCS      130$           ;BR, IF NO PROBLEM
12038 067450 010004      MOV      R0,R4           ;PACKET ADDRESS SET UP
12039 067452 005237 002212      INC      FATFLG          ;BUMP COUNT
12043 067456      ERRHRD  ERRNO,T26RMN,PKTSSR ;REWIND NOT ACCEPTED

```

M4

TEST 6: REREADS

```

067456 104456
067460 001302
067462 074544
067464 012166
12044 067466 104406 1304: CKLOOP ;LOOP IF SELECTED TRAP C4ERHRD
067466 104406 ;***** .WORD 706
;***** .WORD T26RWN
;***** .WORD PKTSSR
12045 ;*****
12046 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
12047 ;*****
12048 ;*****
12049 ;*****
12050 ;*****
12051 ;*****
12052 067470 013701 073110 MOV T26BFR+6,R1 ;PICK UP XSTO
12053 067474 010102 MOV R1,R2 ;SET UP EXPECTED
12054 067476 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
12055 067502 020102 CMP R1,R2 ;DOES EXP = REC'D
12056 067504 001406 BEQ 1404 ;BR, IF EQUAL (OK)
12057 067506 005237 002212 INC FATFLG ;BUMP COUNT
12061 067512 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
067512 104456 TRAP C4ERHRD
067514 001303 .WORD 707
067516 074255 .WORD T26BOT
067520 015614 .WORD EXPREC
12062 067522 1404: CKLOOP ;LOOP IF SELECTED TRAP C4CLP1
067522 104406 ;*****
12063 067524 005303 DEC R3 ;SET RECORD SIZE TO 511.
12064 067526 013737 003116 073212 MOV FREE,T26RB ;STARTING READ BUFFER ADDRESS
12065 ;*****
12066 ;*****
12067 ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
12068 ;*****
12069 ;*****
12070 ;*****
12071 ;*****
12072 067534 012737 161401 073210 MOV #161401,T26PK3 ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
12073 067542 012704 073210 1654: MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
12074 067546 010337 073216 MOV R3,T26S2 ;SET UP RECORD SIZE IN PACKET
12075 067552 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
12076 067556 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
12077 067562 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
12078 067566 012702 100204 MOV #SSR!SC!BIT2,R2 ;SET UP EXPECTED
12079 067572 020102 CMP R1,R2 ;ARE THEY EQUAL
12080 067574 001406 BEQ 1704 ;BR, IF OK
12081 067576 005237 002212 INC FATFLG ;BUMP COUNT
12085 067602 ERRHRD ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
067602 104456 TRAP C4ERHRD
067604 001304 .WORD 708
067606 075622 .WORD T26TRL
067610 012166 .WORD PKTSSR
12086 067612 1704: CKLOOP ;LOOP IF SELECTED TRAP C4CLP1
067612 104406 ;*****
12087 ;*****
12088 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
12089 ;*****
12090 ;*****
12091 ;*****

```


C5

TEST 6: REREADS

```

12195 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
12196 ;
12197 ;*****
12198 ;
12199 070070 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
12200 070074 103407 BCS 26$ ;BR, IF COMMAND ISSUED OK
12201 070076 005237 002212 INC FATFLG ;BUMP COUNT
12205 070102 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
12206 070104 ERRHRD ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTIC FAILED
070104 104456 TRAP C$ERHRD
070106 001311 .WORD 713
070110 005056 .WORD WRTMSG
070112 012154 .WORD SFMSG
12207 070114 26$: CKLOOP ;LOOP IF SELECTED
070114 104406 TRAP C$CLP1
12208 ;*****
12209 ;
12210 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
12211 ;
12212 ;*****
12213 ;
12214 ;
12215 070116 004737 011136 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
12216 070122 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
12217 070126 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
12218 070132 103407 BCS 30$ ;BR, IF NO PROBLEM
12219 070134 010004 MOV RO,R4 ;PACKET ADDRESS SET UP
12220 070136 005237 002212 INC FATFLG ;BUMP COUNT
12224 070142 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
070142 104456 TRAP C$ERHRD
070144 001312 .WORD 714
070146 074544 .WORD T26RWN
070150 012166 .WORD PKTSSR
12225 070152 30$: CKLOOP ;LOOP IF SELECTED
070152 104406 TRAP C$CLP1
12226 ;*****
12227 ;
12228 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
12229 ;
12230 ;*****
12231 ;
12232 ;
12233 070154 013701 073110 MOV T26BFR+6,R1 ;PICK UP XSTO
12234 070160 010102 MOV R1,R2 ;SET UP EXPECTED
12235 070162 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
12236 070166 020102 CMP R1,R2 ;DOES EXP = REC'D
12237 070170 001406 BEQ 40$ ;BR, IF EQUAL (OK)
12238 070172 005237 002212 INC FATFLG ;BUMP COUNT
12242 070176 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
070176 104456 TRAP C$ERHRD
070200 001313 .WORD 715
070202 074255 .WORD T26BOT
070204 015614 .WORD EXPREC
12243 070206 40$: CKLOOP ;LOOP IF SELECTED
070206 104406 TRAP C$CLP1
12244 070210 012703 000400 MOV #256.,R3 ;RECORD SIZE
12245 070214 013737 003116 073212 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS

```

TEST 6: REREADS

```

12246
12247
12248
12249
12250
12251
12252
12253 070222 012737 140005 073210      MOV      #140005,T26PK3      ;WRITE DATA,CVC=1,ACK COMMAND
12254 070230 012704 073210      MOV      #T26PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
12255 070234      65$:
12256 070234 010337 073216      MOV      R3,T26SZ          ;SET UP RECORD SIZE IN PACKET
12257 070240 010465 000000      MOV      R4,TSDB(R5)       ;ISSUE COMMAND
12258 070244 004737 016370      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
12259 070250 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
12260 070254 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
12261 070260 020102      CMP      R1,R2            ;ARE THEY EQUAL
12262 070262 001406      BEQ      75$              ;BR, IF OK
12263 070264 005237 002212      INC      FATFLG           ;BUMP COUNT
12267 070270      ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      070270 104456
      070272 001314
      070274 005113
      070276 012166
12268 070300      75$: CKLOOP              ;LOOP IF SELECTED
      070300 104406
12269 070302      120$:
12270
12271
12272
12273
12274
12275
12276
12277 070302 004737 011136      JSR      PC,REWIND         ;CALL TAPE REWIND COMMAND
12278 070306 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR
12279 070312 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED TSSR
12280 070316 103407      BCS     130$              ;BR, IF NO PROBLEM
12281 070320 010004      MOV      R0,R4            ;PACKET ADDRESS SET UP
12282 070322 005237 002212      INC      FATFLG           ;BUMP COUNT
12286 070326      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      070326 104456
      070330 001315
      070332 074544
      070334 012166
12287 070336      130$: CKLOOP            ;LOOP IF SELECTED
      070336 104406
12288
12289
12290
12291
12292
12293
12294
12295 070340 013701 073110      MOV      T26BFR-6,R1       ;PICK UP XSTO
12296 070344 010102      MOV      R1,R2            ;SET UP EXPECTED
12297 070346 052702 000002      BIS     #BIT1,R2          ;SET BOT BIT IN EXPECTED
12298 070352 020102      CMP      R1,R2            ;DOES EXP = REC'D

```


E5

TEST 6: REREADS

```

12299 070354 001406          BEQ      1354          ;BR, IF EQUAL (OK)
12300 070356 005237 002212  INC      FATFLG      ;BUMP COUNT
12304 070362          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C4ERHRD
                                .WORD    718
                                .WORD    T26BOT
                                .WORD    EXPREC
12305 070372          1354:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C4CLP1
12306 070372 104406          MOV      #512.,R3      ;RECORD SIZE
12307 070374 012703 001000  MOV      FREE,T26RB   ;STARTING READ BUFFER ADDRESS
12308 070400 013737 003116 073212
12309          ;*****
12310          ;
12311          ;REREAD NEXT,ACK,CVC=1,OPP=1
12312          ;
12313          ;*****
12314          ;
12315 070406 012737 161401 073210 1654:  MOV      #161401,T26PK3 ;REREAD NEXT,ACK,CVC=1,OPP=1
12316 070414 012704 073210  MOV      #T26PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
12317 070420 010337 073216  MOV      R3,T26SZ     ;SET UP RECORD SIZE IN PACKET
12318 070424 010465 000000  MOV      R4,TSD8(R5) ;ISSUE COMMAND
12319 070430 004737 016370  JSR      PC,WAITF     ;WAIT FOR SSR TO SET
12320 070434 016501 000002  MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
12321 070440 012702 100204  MOV      #SSR:SC:BIT2,R2 ;SET UP EXPECTED
12322 070444 020102          CMP      R1,R2        ;ARE THEY EQUAL
12323 070446 001406          BEQ      1704        ;BR, IF OK
12324 070450 005237 002212  INC      FATFLG      ;BUMP COUNT
12328 070454          ERRHRD  ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C4ERHRD
                                .WORD    719
                                .WORD    T26TRL
                                .WORD    PKTSSR
12329 070464          1704:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C4CLP1
12330 070464 104406
12331          ;*****
12332          ;
12333          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
12334          ;
12335          ;*****
12336          ;
12337 070466 013701 073110          MOV      T26BFR+6,R1 ;GET MESSAGE BUFFER
12338 070472 010102          MOV      R1,R2        ;SET UP EXPECTED
12339 070474 052702 040000  BIS      #BIT14,R2   ;SET THE RLS BIT IN EXPECTED
12340 070500 020102          CMP      R1,R2        ;ARE THEY EQUAL
12341 070502 001406          BEQ      1804        ;BR, IF EQUAL (ALL IS WELL)
12342 070504 005237 002212  INC      FATFLG      ;BUMP COUNT
12346 070510          ERRHRD  ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
                                TRAP      C4ERHRD
                                .WORD    720
                                .WORD    T26LOP
                                .WORD    EXPREC
12347 070510 104456          1804:  CKLOOP
12348 070522 013701 073106          MOV      T26BFR+4,R1 ;PICK UP RESIDUAL BYTE COUNTER
12349 070526 012702 000400          MOV      #256.,R2   ;THIS SHOULD BE THE DIFFERENCE

```

F5

TEST 6: REREADS

```

12350 070532 020102          CMP      R1,R2          ;IS THE DIFFERENCE CORRECT
12351 070534 001405          BEQ      190$           ;BR, IF CORRECT
12355 070540          ERRHRD  ERRNO,T26PBP,EXPREC ;RBPGR NOT CORRECT
                                TRAP      C$ERHRD
                                .WORD    720
                                .WORD    T26PBP
                                .WORD    EXPREC
12356 070550          190$:  CKLOOP          ;LOOP IF SELECTED          TRAP      C$CLP1
                                .WORD    104406
12357 070552 012703 001000    MOV      #512.,R3       ;RECORD SIZE
12358 070556 013737 003116 073212  MOV      FREE,T26RB     ;STARTING READ BUFFER ADDRESS
12359
12360          ;*****
12361          ;
12362          ;REREAD NEXT,ACK,CVC=1,OPP=0
12363          ;
12364          ;*****
12365
12366 070564 012737 141401 073210  MOV      #141401,T26PK3 ;REREAD NEXT,ACK,CVC=1,OPP=0
12367 070572 012704 073210    MOV      #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
12368 070576 010337 073216    MOV      R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
12369 070602 010465 000000    MOV      R4,TSDB(R5)   ;ISSUE COMMAND
12370 070606 004737 016370    JSR      PC,WAITF      ;WAIT FOR SSR TO SET
12371 070612 016501 000002    MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
12372 070616 012702 100204    MOV      #SSR!SC!BIT2,R2 ;SET UP EXPECTED
12373 070622 020102          CMP      R1,R2          ;ARE THEY EQUAL
12374 070624 001406          BEQ      270$           ;BR, IF OK
12375 070626 005237 002212    INC      FATFLG        ;BUMP COUNT
12379 070632          ERRHRD  ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    721
                                .WORD    T26TRL
                                .WORD    PKTSSR
12380 070642          270$:  CKLOOP          ;LOOP IF SELECTED          TRAP      C$CLP1
                                .WORD    104406
12381
12382          ;*****
12383          ;
12384          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
12385          ;
12386          ;*****
12387
12388 070644 013701 073110    MOV      T26BFR-6,R1   ;GET MESSAGE BUFFER
12389 070650 010102          MOV      R1,R2          ;SET UP EXPECTED
12390 070652 052702 040000    BIS      #BIT14,R2     ;SET THE RLS BIT IN EXPECTED
12391 070656 020102          CMP      R1,R2          ;ARE THEY EQUAL
12392 070660 001406          BEQ      280$           ;BR, IF EQUAL (ALL IS WELL)
12393 070662 005237 002212    INC      FATFLG        ;BUMP COUNT
12397 070666          ERRHRD  ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XST0
                                TRAP      C$ERHRD
                                .WORD    722
                                .WORD    T26LOP
                                .WORD    EXPREC
12398 070676          280$:  CKLOOP          ;LOOP IF SELECTED          TRAP      C$CLP1
                                .WORD    104406
12399 070700 013701 073106    MOV      T26BFR-4,R1   ;PICK UP RESIDUAL BYTE COUNTER
12400 070704 012702 000400    MOV      #256.,R2     ;THIS SHOULD BE THE DIFFERENCE

```


TEST 6: REREADS

```

12451
12452
12453
12454
12455
12456
12457
12458 071040 004737 010752
12459 071044 103407
12460 071046 005237 002212
12464 071052 010001
12465 071054
      071054 104456
      071056 001324
      071060 005056
      071062 012154
12466 071064
      071064 104406
12467
12468
12469
12470
12471
12472
12473
12474 071066 004737 021302
12475 071072 004737 011136
12476 071076 103411
12477 071100 016501 000002
12478 071104 010004
12479 071106 005237 002212
12483 071112
      071112 104456
      071114 001325
      071116 074544
      071120 012166
12484 071122
      071122 104406
12485
12486
12487
12488
12489
12490
12491
12492 071124 013701 073110
12493 071130 010102
12494 071132 052702 000002
12495 071136 020102
12496 071140 001406
12497 071142 005237 002212
12501 071146
      071146 104456
      071150 001326
      071152 074255
      071154 015614
12502 071156

```

```

;*****
;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 CHARACTERISTIC FAILED
                                TRAP      C$ERHRD
                                .WORD     724
                                .WORD     WRTMSG
                                .WORD     SFIMSG
26$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
;*****
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;*****
      JSR      PC,INVERT     ;INVERT THE EXTENDED FEATURES SWITCH
      JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
      BCS      30$           ;BR, IF NO PROBLEM
      MOV      TSSR(R5),R1   ;GET TSSR
      MOV      R0,R4         ;PACKET ADDRESS SET UP
      INC      FATFLG        ;BUMP COUNT
      ERRHRD   ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     725
                                .WORD     T26RWN
                                .WORD     PKTSSR
30$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
;*****
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;*****
      MOV      T26BFR-6,R1   ;PICK UP XSTO
      MOV      R1,R2         ;SET UP EXPECTED
      BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
      CMP      R1,R2         ;DOES EXP = REC'D
      BEQ      40$           ;BR, IF EQUAL (OK)
      INC      FATFLG        ;BUMP COUNT
      ERRHRD   ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD     726
                                .WORD     T26BOT
                                .WORD     EXPREC
40$:   CKLOOP                ;LOOP IF SELECTED

```

TEST 6: REREADS

```

12503 071156 104406 003116 073212      MOV      FREE,T26RB          ;STARTING WRITE BUFFER ADDRESS      TRAP      C#CLP1
12504 071160 013737
12505 ;*****
12506 ;WRITE DATA,CVC=1,ACK COMMAND
12507 ;
12508 ;*****
12509 ;
12510 ;
12511 071166 012737 140005 073210      MOV      #140005,T26PK3      ;WRITE DATA,CVC=1,ACK COMMAND
12512 071174 012704 073210      MOV      #T26PK3,R4         ;SET UP R4 WITH PACKET ADDRESS
12513 071200 012737 000400 073216 65$:  MOV      #256.,T26SZ        ;SET UP RECORD SIZE IN PACKET
12514 071206 013777 073236 111702  MOV      T26CNT,#FREE       ;MOVE TAPE RECORD NUMBER TO BUFFER
12515 071214 062737 000001 073236  ADD      #1,T26CNT          ;NUMBER READY FOR NEXT RECORD
12516 071222 010465 000000      MOV      R4,TSDB(R5)        ;ISSUE COMMAND
12517 071226 004737 016370      JSR      PC,WAITF           ;WAIT FOR SSR TO SET
12518 071232 016501 000002      MOV      TSSR(R5),R1        ;GET TSSR CONTENTS
12519 071236 012702 000200      MOV      #SSR,R2           ;SET UP EXPECTED
12520 071242 020102      CMP      R1,R2              ;ARE THEY EQUAL
12521 071244 001406      BEQ      75$                ;BR, IF OK
12522 071246 005237 002212      INC      FATFLG             ;BUMP COUNT
12526 071252      ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C#ERHRD
                                .WORD      727
                                .WORD      WRERR
                                .WORD      PKTSSR
12527 071262      75$:  CKLOOP                    ;LOOP IF SELECTED
                                TRAP      C#CLP1
12528 071264 022737 000013 073236  CMP      #11.,T26CNT        ;CHECK NUMBER OF RECORDS WRITTEN
12529 071272 001401      BEQ      120$              ;BR, IF AT END OF WRITE SEQUENCE
12530 071274 000741      BR      65$                ;WRITE MORE RECORDS
12531 071276      120$:
12532 071276 005037 003134      CLR      NXMH1              ;SET TO 16 BIT ADDRESS
12533 071302      125$:
12534 071302 005037 073236      CLR      T26CNT            ;SET RECORD COUNTER BACK TO ZERO
12535 ;*****
12536 ;
12537 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
12538 ;
12539 ;*****
12540 ;
12541 ;
12542 071306 004737 011136      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
12543 071312 103411      BCS      130$              ;BR, IF NO PROBLEM
12544 071314 016501 000002      MOV      TSSR(R5),R1        ;GET TSSR
12545 071320 010004      MOV      R0,R4             ;PACKET ADDRESS SET UP
12546 071322 005237 002212      INC      FATFLG            ;BUMP COUNT
12550 071326      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C#ERHRD
                                .WORD      728
                                .WORD      T26RWN
                                .WORD      PKTSSR
12551 071336 104406      130$:  CKLOOP                    ;LOOP IF SELECTED
                                TRAP      C#CLP1
12552 ;*****
12553 ;
12554 ;

```

TEST 6: REREADS

```

12555 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
12556 ;
12557 ;*****
12558 ;
12559 071340 013701 073110          MOV      T268FR+6,R1      ;PICK UP XSTO
12560 071344 010102                MOV      R1,R2           ;SET UP EXPECTED
12561 071346 052702 000002        BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
12562 071352 020102                CMP      R1,R2           ;DOES EXP = REC'D
12563 071354 001406                BEQ     140$             ;BR, IF EQUAL (OK)
12564 071356 005237 002212        INC      FATFLG          ;BUMP COUNT
12568 071362                ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    729
                                .WORD    T26BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
071362 104456
071364 001331
071366 074255
071370 015614
12569 071372                140$:  CKLOOP          ;LOOP IF SELECTED
071372 104406
12570 071374 012703 073226        MOV      @T26RN,R3       ;COMMAND BUFFER ADDRESS
12571 071400 013737 003132 073212 150$:  MOV      NXML0,T26RB     ;STARTING READ BUFFER ADDRESS
12572 071406 013737 003134 073214  MOV      NXMHI,T26RB+2   ;SET UP HIGH ORDER ADDRESS BITS
12573 ;*****
12574 ;
12575 ;REREAD DATA,IE,ACK, OPP COMMAND
12576 ;
12577 ;*****
12578 ;
12579 ;
12580 071414 011337 073210          MOV      (R3),T26PK3     ;REREAD DATA,IE,ACK, OPP COMMAND
12581 071420 012704 073210          MOV      @T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
12582 071424 012737 000400 073216 165$:  MOV      @256.,T26SZ     ;SET UP RECORD SIZE IN PACKET
12583 071432 010465 000000          MOV      R4,TSDB(R5)    ;ISSUE COMMAND
12584 071436 004737 016370          JSR     PC,WAITF        ;WAIT FOR SSR TO SET
12585 071442 016501 000002          MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
12586 071446 012702 104210          MOV      @SSR!NXM!SC!BIT3,R2 ;SET UP EXPECTED
12587 071452 020102                CMP      R1,R2           ;ARE THEY EQUAL
12588 071454 001422                BEQ     170$             ;BR, IF OK
12589 071456 031327 001000          BIT     (R3),@BIT9     ;CHECK FOR A READ COMMAND
12590 071462 001403                BEQ     168$             ;BR, IF IT WAS A READ COMMAND
12591 071464 030127 000002          BIT     R1,@BIT1       ;WAS BIT1 SET
12592 071470 001014                BNE     170$             ;BR, IF REREAD AND BIT1 SET
12593 071472                168$:
12594 071472 005237 003134          INC      NXMHI          ;BUMP TO NEXT ADDRESS RANGE
12595 071476 023727 003134 000004  CMP      NXMHI,@4       ;CHECK FOR OVERFLOW
12596 071504 001276                BNE     125$             ;BR, IF MORE BITS TO GO
12597 071506 005237 002212        INC      FATFLG          ;BUMP COUNT
12601 071512                ERRHRD  ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                                TRAP      C$ERHRD
                                .WORD    730
                                .WORD    T26RRF
                                .WORD    PKTSSR
                                TRAP      C$CLP1
071512 104456
071514 001332
071516 073465
071520 012166
12602 071522                170$:  CKLOOP          ;LOOP IF SELECTED
071522 104406
12603 ;*****
12604 ;
12605 ;READ DATA, ACK,CVC=1 COMMAND
12606 ;
12607 ;

```

TEST 6: REREADS

```

12608
12609
12610 071524 012737 140001 073210      MOV      #140001,T26PK3      ;READ DATA, ACK,CVC=1 COMMAND
12611 071532 012737 000400 073216      MOV      #256.,T26SZ        ;SET SIZE INTO PACKET
12612 071540 005037 073214      CLR      T26RB+2           ;CLEAR OUT HIGH ADDRESS BITS
12613 071544 013737 003116 073212      MOV      FREE,T26RB        ;GIVE READ A GOOD BUFFER
12614 071552 010465 000000      MOV      R4,TSDB(R5)       ;ISSUE READ DATA COMMAND
12615 071556 004737 016370      JSR      PC,WAITF          ;WAIT FOR SSR
12616 071562 016501 000002      MOV      TSSR(R5),R1       ;PICK UP THE TSSR
12617 071566 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
12618 071572 020102      CMP      R1,R2             ;IS THE TSSR OK
12619 071574 001406      BEQ      180$              ;BR, IF TSSR OK (GOOD)
12620 071576 005237 002212      INC      FATFLG            ;BUMP COUNT
12624 071602      ERRHRD  ERRNO,RDERR,PKTSSR ;READ DATA COMMAND FAILED
                                TRAP      C$ERHRD
                                .WORD     731
                                .WORD     RDERR
                                .WORD     PKTSSR
                                TRAP      C$CLP1
12625 071612      180$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
12626 071614 017701 111276      MOV      @FREE,R1          ;FIRST WORD FROM READ BUFFER
12627 071620 012702 000001      MOV      #1,R2            ;SET UP EXPECTED
12628 071624 020102      CMP      R1,R2            ;IS TAPE POSITION CORRECT
12629 071626 001406      BEQ      190$              ;KEEP GOING POSITION OK
12630 071630 005237 002212      INC      FATFLG            ;BUMP COUNT
12634 071634      ERRHRD  ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
                                TRAP      C$ERHRD
                                .WORD     732
                                .WORD     T26WNG
                                .WORD     EXPREC
                                TRAP      C$CLP1
12635 071644      190$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
12636
12637
12638
12639
12640
12641
12642
12643 071646 004737 011136      JSR      PC,REWIND         ;CALL TAPE REWIND COMMAND
12644 071652 103411      BCS     194$              ;BR, IF NO PROBLEM
12645 071654 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR
12646 071660 010004      MOV      R0,R4            ;PACKET ADDRESS SET UP
12647 071662 005237 002212      INC      FATFLG            ;BUMP COUNT
12651 071666      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     733
                                .WORD     T26RWN
                                .WORD     PKTSSR
                                TRAP      C$CLP1
12652 071676      194$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
12653
12654
12655
12656
12657
12658
;*****
;
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;
;*****

```


M5

TEST 6: REREADS

```

12710 072012 004737 076134          JSR      PC,T26RT3          ;SET UP OTHER COMMAND PACKET
12711
12712          ;*****
12713          ;
12714          ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
12715          ;
12716          ;*****
12717
12718 072016 004737 016114          JSR      PC,SOFINIT        ;DO INITIALIZE ON CONTROLLER
12719 072022 103407                BCS      20$              ;BR IF INIT WAS OK
12720 072024 005237 002212          INC      FATFLG          ;BUMP COUNT
12724 072030 010001                MOV      RO,R1           ;CONTENTS OF TSSR REGISTER
12725 072032                ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                TRAP      C4ERDF
                                .WORD    735
                                .WORD    SFIERR
                                .WORD    SFIMSG
                                072032 104455
                                072034 001337
                                072036 003652
                                072040 012154
12726 072042 013737 002172 073100 20$:  MOV      UNITN,T26DSW      ;SET UP UNIT NUMBER
12727
12728 072050 012704 073060          MOV      @T26PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
12729
12730          ;*****
12731          ;
12732          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
12733          ;
12734          ;*****
12735
12736 072054 004737 010752          JSR      PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
12737 072060 103407                BCS      25$              ;BR, IF COMMAND ISSUED OK
12738 072062 005237 002212          INC      FATFLG          ;BUMP COUNT
12742 072066 010001                MOV      RO,R1           ;SAVE CONTENTS OF TSSR
12743 072070                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP      C4ERHRD
                                .WORD    736
                                .WORD    WRTMSG
                                .WORD    SFIMSG
                                072070 104456
                                072072 001340
                                072074 005056
                                072076 012154
12744 072100                25$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C4CLP1
                                072100 104406
12745
12746          ;*****
12747          ;
12748          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
12749          ;
12750          ;*****
12751
12752 072102 004737 011136          26$:  JSR      PC,REWIND    ;CALL TAPE REWIND COMMAND
12753 072106 016501 000002          MOV      TSSR(R5),R1     ;GET TSSR
12754 072112 012702 000200          MOV      @SSR,R2        ;SET UP EXPECTED TSSR
12755 072116 103407                BCS      30$              ;BR, IF NO PROBLEM
12756 072120 010004                MOV      RO,R4           ;PACKET ADDRESS SET UP
12757 072122 005237 002212          INC      FATFLG          ;BUMP COUNT
12761 072126                ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C4ERHRD
                                .WORD    737
                                .WORD    T26RWN
                                .WORD    PKTSSR
                                072126 104456
                                072130 001341
                                072132 074544
                                072134 012166
12762 072136                30$:  CKLOOP          ;LOOP IF SELECTED

```

TEST 6: REREADS

```

072136 104406
12763 TRAP C#CLP1
12764
12765 ;*****
12766 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
12767 ;
12768 ;*****
12769
12770 072140 013701 073110 MOV T26BFR+6,R1 ;PICK UP XSTO
12771 072144 010102 MOV R1,R2 ;SET UP EXPECTED
12772 072146 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
12773 072152 020102 CMP R1,R2 ;DOES EXP = REC'D
12774 072154 001406 BEQ 40$ ;BR, IF EQUAL (OK)
12775 072156 005237 002212 INC FATFLG ;BUMP COUNT
12779 072162 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
072162 104456 TRAP C#ERHRD
072164 001342 .WORD 738
072166 074255 .WORD T26BOT
072170 015614 .WORD EXPREC
12780 072172 40$: CKLOOP ;LOOP IF SELECTED
072172 104406 TRAP C#CLP1
12781 072174 012737 000400 073216 MOV #256.,T26SZ ;SET UP RECORD SIZE IN PACKET
12782 072202 013737 003116 073212 MOV FREE,T26RB ;ADDRESS OF READ BUFFER
12783 072210 005703 TST R3 ;CHECK NUMBER OF TIMES THROUGH HERE
12784 072212 001404 BEQ 50$ ;BR, IF FIRST TIME THROUGH HERE
12785
12786 ;*****
12787 ;REREAD,CVC-1,ACK COMMAND
12788 ;
12789 ;*****
12790
12791
12792 072214 012737 161001 073210 MOV #161001,T26PK3 ;REREAD,CVC-1,ACK COMMAND
12793 072222 000403 BR 55$ ;SKIP NEXT COMMAND
12794
12795 ;*****
12796 ;REREAD,ACK COMMAND
12797 ;
12798 ;*****
12799
12800
12801 072224 012737 141001 073210 50$: MOV #141001,T26PK3 ;REREAD,ACK COMMAND
12802 072232 55$:
12803 072232 012704 073210 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
12804 072236 65$:
12805 072236 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
12806 072242 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
12807 072246 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
12808 072252 012702 100206 MOV #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
12809 072256 020102 CMP R1,R2 ;ARE THEY EQUAL
12810 072260 001406 BEQ 75$ ;BR, IF OK
12811 072262 005237 002212 INC FATFLG ;BUMP COUNT
12815 072266 ERRHRD ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
072266 104456 TRAP C#ERHRD
072270 001343 .WORD 739
072272 074203 .WORD T26WDE
072274 012166 .WORD PKTSSR

```


C6

TEST 6: REREADS

```

12866
12867
12868
12869 072400 004737 016114
12870 072404 103407
12871 072406 005237 002212
12875 072412 010001
12876 072414
      072414 104455
      072416 001345
      072420 003652
      072422 012154
12877 072424 013737 002172 073100 20$:  MOV    UNITN,T26DSW      ;SET UP UNIT NUMBER
12878
12879 072432 012704 073060      MOV    #T26PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
12880
12881
12882
12883
12884
12885
12886
12887 072436 004737 010752      JSR    PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
12888 072442 103407              BCS    25$                ;BR, IF COMMAND ISSUED OK
12889 072444 005237 002212      INC    FATFLG             ;BUMP COUNT
12893 072450 010001              MOV    R0,R1              ;SAVE CONTENTS OF TSSR
12894 072452
      072452 104456              ERRHRD ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICSC FAILED
      072454 001346
      072456 005056
      072460 012154
12895 072462
      072462 104406      25$:  CKLOOP              ;LOOP IF SELECTED
12896
12897
12898
12899
12900
12901
12902
12903 072464 004737 011136      26$:  JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
12904 072470 016501 000002      MOV    TSSR(R5),R1      ;GET TSSR
12905 072474 012702 000200      MOV    #SSR,R2          ;SET UP EXPECTED TSSR
12906 072500 103407              BCS    30$                ;BR, IF NO PROBLEM
12907 072502 010004              MOV    R0,R4             ;PACKET ADDRESS SET UP
12908 072504 005237 002212      INC    FATFLG             ;BUMP COUNT
12912 072510
      072510 104456              ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      072512 001347
      072514 074544
      072516 012166
12913 072520
      072520 104406      30$:  CKLOOP              ;LOOP IF SELECTED
12914
12915
12916
12917

```

D6

TEST 6: REREADS

```

12918
12919
12920
12921 072522 013701 073110
12922 072526 010102
12923 072530 052702 000002
12924 072534 020102
12925 072536 001406
12926 072540 005237 002212
12930 072544
      072544 104456
      072546 001350
      072550 074255
      072552 015614
12931 072554
      072554 104406
12932
12933
12934
12935
12936
12937
12938
12939
12940 072556 012703 000001
12941 072562 004737 010560
12942 072566 103411
12943 072570 016501 000002
12944 072574 010004
12945 072576 005237 002212
12949 072602
      072602 104456
      072604 001351
      072606 074203
      072610 012166
12950 072612
      072612 104406
12951
12952
12953
12954
12955
12956
12957
12958
12959 072614 012703 100001
12960 072620 004737 010560
12961 072624 103411
12962 072626 016501 000002
12963 072632 010004
12964 072634 005237 002212
12968 072640
      072640 104456
      072642 001352
      072644 074203
      072646 012166
12969 072650

```

```

;
;*****
      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   C4ERHRD
                                .WORD  744
                                .WORD  T26BOT
                                .WORD  EXPREC
40$:   CKLOOP
                                TRAP   C4CLP1
;*****
;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
;*****
      MOV      #000001,R3      ;SET UP SPACE FORWARD 1 RECORD
      JSR      PC,SPACE       ;ISSUE SPACE COMMAND
      BCS     75$            ;BR, IF OK
      MOV      TSSR(R5),R1    ;GET STATUS DATA
      MOV      R0,R4         ;GET PACKET ADDRESS
      INC      FATFLG         ;BUMP COUNT
      ERRHRD   ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP   C4ERHRD
                                .WORD  745
                                .WORD  T26WDE
                                .WORD  PKTSSR
75$:   CKLOOP?                ;LOOP IF SELECTED
                                TRAP   C4CLP1
;*****
;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
;*****
      MOV      #100001,R3     ;SET SPACE REVERSE 1 RECORD
      JSR      PC,SPACE       ;ISSUE COMMAND
      BCS     175$           ;GO ON IF ALL IS WELL
      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
      MOV      R0,R4         ;SET UP EXPECTED (PACKET CONTENTS)
      INC      FATFLG         ;BUMP COUNT
      ERRHRD   ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP   C4ERHRD
                                .WORD  746
                                .WORD  T26WDE
                                .WORD  PKTSSR
175$:  CKLOOP                ;LOOP IF SELECTED

```


F6

TEST 6: REREADS

```

13020 073026 004737 017322          JSR      PC,CKDROP          ;TRY TO DROP THE UNIT
13021 073032          9994:
13022          ;
13023          ;
13024          ;
13025 073032 004737 016576          JSR      PC,TSTLOOP        ;DO WE NEED TO ITERATE TEST
13026 073036 103002          BCC     1634               ;BR, IF NO LOOP REQUIRED
13027 073040 000137 056650          JMP     T26LOOP           ;EXECUTE AGAIN
13028 073044          1634:
13029 073044          EXIT     TST          ;ALL DONE THIS TEST
          073044          TRAP     C4EXIT
          073046 003116          .WORD   L10102-.

13030
13031
13032          ;*
13033          ;LOCAL STORAGE FOR THIS TEST
13034          ;-
13035 073050          .BLKB   10-<.-TSV2&7>
13036 073060          T26PACKET:
          .WORD   14004          ;COMMAND PACKET FOR TEST
          .WORD   T26DATA       ;WRITE CHARACTERISTICS COMMAND, WITH CVC=1, ACK
          .WORD   0              ;ADDRESS OF CHARACTERISTICS BLOCK
          .WORD   10.           ;STARTING VALUE OF BLOCK SIZE
13037 073066          T26DATA:
          .WORD   T26BFR        ;CHARACTERISTICS DATA BLOCK
          .WORD   0              ;ADDRESS OF MESSAGE BUFFER
          .WORD   20.          ;LENGTH OF MESSAGE BUFFER
13038 073072          .WORD   0
13039 073074          .WORD   0
13040 073076          T26DSW: .WORD   0          ;SELECT DRIVE 0
13041 073100          T26BFR: .BLKW  25.         ;MESSAGE BUFFER
13042 073102          ;
13043 073102          ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
13044 073102          ;
13045 073164          .BLKB   10-<.-TSV2&7>
13046 073170          T26PK2:
          .WORD   100006        ;WRITE SUB SYS MEM COMMAND, AND ACK
          .WORD   T26BF2       ;ADDRESS OF SELECT BLOCK DATA
          .WORD   0              ;SIZE OF DATA PACKET
13047 073176          .WORD   6.
13048 073200          .BLKB   10-<.-TSV2&7>
13049 073210          T26PK3:
          .WORD   140005        ;REREAD COMMAND, CVC=1 AND ACK
13050 073212          T26RB:
          .WORD   FREE          ;ADDRESS OF WRITE BUFFER
13051 073214          T26MB: .WORD   0
13052 073216          T26SZ: .WORD   0          ;SIZE OF BUFFER (EXTENT)
13053 073216          .EVEN
13054
13055
13056          ;
13057          ;
13058          ;
13059          ;
13060          ;
13061          ;
13062          ;
13063          ;
13064          ;
13065          ;
13066          ;
13067          ;
13068          ;
13069          ;
13070          ;
13071          ;
13072          ;
13073          ;
13074 073220          T26BF2:
13075 073220          T26BS0: .BYTE   10          ;BSELO AREA
          010
13076 073221          T26BS1: .BYTE   200         ;BSEL1 AREA
          200
13077 073222          T26S2: .WORD   0          ;SEL 2 AREA
          000000
13078 073224          T26S3: .WORD   0          ;DATA AREA
          000000
13079
13080

```

TEST 6: REREADS

```

13081
13082
13083
13084 073226 140001
13085 073230 141401
13086 073232 161401
13087 073234 177777
13088
13089
13090 073236 000000
13091 073240 000000
13092
13093 073242 000000
13094
13095 073244 000000
13096
13097
13098
13099
13100
13101
13102
13103 073246 124 141 160
13104 073334 122 105 122
13105 073416 124 123 123
13106 073465 122 105 122
13107 073562 122 105 122
13108 073657 120 117 123
13109 073741 122 111 102
13110 074011 124 123 123
13111 074066 111 154 154
13112 074147 122 105 122
13113 074203 124 123 123
13114 074255 124 141 160
13115 074322 104 141 164
13116 074410 122 105 122
13117 074467 124 123 123
13118 074544 122 145 167
13119 074613 122 101 115
13120 074666 124 123 123
13121 074735 104 162 151
13122 075010 124 123 123
13123 075100 124 123 123
13124 075153 103 126 103
13125 075226 124 123 102
13126 075301 127 122 111
13127 075370 122 145 141
13128 075452 122 145 141
13129 075534 122 145 163
13130 075622 122 145 141
13131 075710 104 141 164
13132 075767 122 145 162
13133
13134
13135
13136
13137

;TAPES MOTION PACKET COMMAND VALUES
T26RN: .WORD 140001 ;READ DATA
        .WORD 141401 ;REREAD NEXT OPP=0
        .WORD 161401 ;REREAD NEXT OPP=1
        .WORD 177777 ;END OF DATA

T26CNT: .WORD 0 ;TAPE RECORD COUNTER STORAGE AREA
T26CNU: .WORD 0 ;TAPE RECORD COUNTER STORAGE AREA

T26RSZ: .WORD 0 ;RECORD STORAGE SIZE AREA

T26DLY: .WORD 0 ;DELAY COUNTER AREA

;*
;LOCAL TEXT MESSAGES FOR TEST
;-

T26WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
T26NEF: .ASCIZ 'REREAD PREVIOUS, At BOT, Failed To Set NEF (XST0)'
T26RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
T26RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
T26RRG: .ASCIZ 'REREAD Previous (Read Reverse, Space Forward) Command Failed'
T26SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
T26LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
T26WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
T26LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
T26SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
T26WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
T26BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
T26DTA: .ASCIZ 'Data Written To Tape Not Equal To Data Read From Tape'
T26EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
T26TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
T26RW: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
T26RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
T26AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
T26OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
T26WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
T26WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
T26VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
T26BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
T26WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
T26LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
T26LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
T26PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
T26TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
T26NEQ: .ASCIZ 'Data REREAD From Tape Not Correct, After SWB=1'
TST26ID: .ASCIZ 'Rereads'

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

```


H6

TEST 6: REREADS

```

13138
13139
13140
13141 076000
13142 076000
13143 076004 012701 073060
13144 076010 012721 140004
13145 076014 012721 073070
13146 076020 005021
13147 076022 012721 000012
13148 076026 012721 073102
13149 076032 005021
13150 076034 012721 000024
13151 076040 005021
13152 076042 012711 000000
13153 076046 012702 000030
13154 076052 012762 177777
13155 076060 005742
13156 076062 020227 000000
13157 076066 001371
13158 076070 000207
13159
13160
13161 076072
13162 076072
13163 076076 012701 073170
13164 076102 012721 140006
13165 076106 012721 073220
13166 076112 005021
13167 076114 012721 000006
13168 076120 005021
13169 076122 012701 073220
13170 076126 005021
13171 076130 005011
13172 076132 000207
13173 076134
13174 076134
13175 076140 012701 073210
13176 076144 012721 000000
13177 076150 012721 000000
13178 076154 005021
13179 076156 012711 000000
13180 076162 000207
13181 076164
076164 104401

```

```

;
;-
T26REST:
  SAVREG
  MOV #T26PACKET,R1 ;SAVE THE REGISTERS
  MOV #140004,(R1)+ ;START OF THE PACKET
  MOV #T26DATA,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK, CVC=1
  CLR (R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
  MOV #10,(R1)+ ;EXTENDED ADDRESS
  MOV #T26BFR,(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 (0)
  MOV #24,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
  MOV #177777,T26BFR(R2) ;ALL ONES TO MESSAGE BUFFER
  TST -(R2) ;NEXT LOCATION
  CMP R2,#0 ;CHECK FOR END OF LOOP
  BNE 644 ;KEEP GOING UNTIL DONE
  RTS PC ;RETURN

T26RT2:
  SAVREG
  MOV #T26PK2,R1 ;SAVE THE REGISTERS
  MOV #140006,(R1)+ ;START OF THE PACKET
  MOV #T26BF2,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1.
  CLR (R1)+ ;ADDRESS OF DATA BLOCK
  MOV #6,(R1)+ ;EXTENDED ADDRESS
  CLR (R1)+ ;SIZE OF DATA BLOCK IN BYTES
  MOV #T26BF2,R1 ;POINT TO DATA SEL AREA
  CLR (R1)+
  CLR (R1)
  RTS PC ;RETURN

T26RT3:
  SAVREG
  MOV #T26PK3,R1 ;SAVE THE REGISTERS
  MOV #0,(R1)+ ;START OF THE PACKET
  MOV #0,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
  CLR (R1)+ ;ADDRESS OF DATA BLOCK
  MOV #0,(R1) ;EXTENDED ADDRESS
  RTS PC ;SIZE OF DATA BLOCK IN BYTES
  ENDTST ;RETURN

L10102: TRAP C4ETST

```

```

.SBTTL TEST 7: WRITE DATA RETRY
;
; THIS TEST VERIFIES PROPER OPERATION OF THE WRITE DATA RETRY
; COMMAND (SPACE REVERSE, ERASE, WRITE DATA)
;
; THE TEST CONSISTS OF THE FOLLOWING 5 SUBTESTS
;

```


TEST 7: WRITE DATA RETRY

```

076332 001275
076334 003652
076336 012154
13245 076340 013737 002172 102730 204: MOV UNITN,T27DSW ;SET UP DRIVE NUMBER
13246 076346 012704 102710 MOV #T27PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
13247
13248 ;*****
13249 ;
13250 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
13251 ;
13252 ;*****
13253
13254 076352 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
13255 076356 103407 BCS 254 ;BR, IF COMMAND ISSUED OK
13256 076360 005237 002212 INC FATFLG ;BUMP COUNT
13260 076364 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
13261 076366 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
076366 104456 TRAP C#ERHRD
076370 001276 .WORD 702
076372 005056 .WORD WRTMSG
076374 012154 .WORD SFIMSG
13262 076376 104406 254: CKLOOP ;LOOP IF SELECTED
076376 104406 TRAP C#CLP1
13263
13264 ;*****
13265 ;
13266 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
13267 ;
13268 ;*****
13269
13270 076400 004737 011136 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
13271 076404 103407 BCS 304 ;BR, IF NO PROBLEM
13272 076406 010004 MOV RO,R4 ;SET UP REWIND PACKET ADDRESS
13273 076410 005237 002212 INC FATFLG ;BUMP COUNT
13277 076414 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
076414 104456 TRAP C#ERHRD
076416 001277 .WORD 703
076420 104245 .WORD T27RWN
076422 012166 .WORD PKTSSR
13278 076424 104406 304: CKLOOP ;LOOP IF SELECTED
076424 104406 TRAP C#CLP1
13279
13280 ;*****
13281 ;
13282 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
13283 ;
13284 ;*****
13285
13286 076426 013701 102740 MOV T27BFR+6,R1 ;PICK UP XSTO
13287 076432 010102 MOV R1,R2 ;SET UP EXPECTED
13288 076434 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
13289 076440 020102 CMP R1,R2 ;DOES EXP = REC'D
13290 076442 001406 BEQ 404 ;BR, IF EQUAL (OK)
13291 076444 005237 002212 INC FATFLG ;BUMP COUNT
13295 076450 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
076450 104456 TRAP C#ERHRD
076452 001300 .WORD 704

```

TEST 7: WRITE DATA RETRY

```

076454 103741 .WORD T27BOT
076456 015614 .WORD EXPREC
13296 076460 104406 40$: CKLOOP ;LOOP IF SELECTED TRAP C4CLP1
076462 012737 000400 103046 MOV #256.,T27SZ ;SET UP RECORD SIZE
076470 013737 003116 103042 MOV FREE,T27WB ;ADDRESS OF WRITE BUFFER
13297
13298
13299
13300 ;*****
13301 ;
13302 ;WRITE DATA RETRY,ACK,CVC=1 COMMAND
13303 ;
13304 ;*****
13305
13306 076476 012737 141005 103040 MOV #141005,T27PK3 ;WRITE DATA RETRY,ACK,CVC=1 COMMAND
13307 076504 012704 103040 MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
13308 076510 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
13309 076514 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
13310 076520 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
13311 076524 012702 100206 MOV #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
13312 076530 020102 CMP R1,R2 ;ARE THEY EQUAL
13313 076532 001406 BEQ 75$ ;BR, IF OK
13314 076534 005237 002212 INC FATFLG ;BUMP COUNT
13318 076540 ERRHRD ERRNO,T27WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
076540 104456 TRAP C4ERHRD
076542 001301 .WORD 705
076544 103652 .WORD T27WDE
076546 012166 .WORD PKTSSR
13319 076550 104406 75$: CKLOOP ;LOOP IF SELECTED TRAP C4CLP1
076550
13320
13321 ;*****
13322 ;
13323 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
13324 ;
13325 ;*****
13326
13327 076552 013701 102740 MOV T27BFR-6,R1 ;GET XST0 STATUS WORD
13328 076556 010102 MOV R1,R2 ;SET UP EXPECTED
13329 076560 052702 002000 BIS #BIT10,R2 ;SET THE NEF BIT
13330 076564 020102 CMP R1,R2 ;ARE THEY EQUAL
13331 076566 001406 BEQ 170$ ;BR, IF EQUAL (GOOD)
13332 076570 005237 002212 INC FATFLG ;BUMP COUNT
13336 076574 ERRHRD ERRNO,T27NEF,EXPREC ;NEF SHOULD BE SET
076574 104456 TRAP C4ERHRD
076576 001302 .WORD 706
076600 105411 .WORD T27NEF
076602 015614 .WORD EXPREC
13337 076604 104406 170$: CKLOOP TRAP C4CLP1
076604
13338 076606 ENDSUB
076606
13339 076610 023727 002212 000017 L10123: CMP FATFLG,#15. ;IS ERROR COUNT AT 25
13340 076616 103402 BLO 999$ ;BR, IF LESS THAN 25
13341 076620 004737 017322 JSR PC,CKDROP ;TRY TO DROP THE UNIT
13342 076624
13343

```


TEST 7: WRITE DATA RETRY

```

13396
13397
13398
13399
13400
13401 076726 004737 011136
13402 076732 103411
13403 076734 010004
13404 076736 016501 000002
13405 076742 005237 002212
13409 076746
      076746 104456
      076750 001305
      076752 104245
      076754 012166
13410 076756
      076756 104406
13411 076760 012703 000400
13412 076764 013737 003116 103042
13413
13414
13415
13416
13417
13418
13419
13420 076772 012737 140005 103040
13421 077000 012704 103040
13422 077004 010337 103046
13423 077010 010465 000000
13424 077014 004737 016370
13425 077020 016501 000002
13426 077024 012702 000200
13427 077030 020102
13428 077032 001406
13429 077034 005237 002212
13433 077040
      077040 104456
      077042 001306
      077044 005113
      077046 012166
13434 077050
      077050 104406
13435
13436
13437
13438
13439
13440
13441
13442 077052 004737 011136
13443 077056 103411
13444 077060 016501 000002
13445 077064 010004
13446 077066 005237 002212
13450 077072
      077072 104456

```

```

;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
      BCS     26$                ;BR, IF NO PROBLEM
      MOV     R0,R4              ;SET UP REWIND PACKET ADDRESS
      MOV     TSSR(R5),R1        ;GET TSSR CONTENTS
      INC     FATFLG            ;BUMP COUNT
      ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP  C$ERHRD
                                .WORD 709
                                .WORD T27RWN
                                .WORD PKTSSR
26$:  CKLOOP                    ;LOOP IF SELECTED
                                TRAP  C$CLP1
      MOV     #256.,R3          ;STARTING RECORD SIZE
      MOV     FREE,T27WB        ;STARTING WRITE BUFFER ADDRESS
;*****
;WRITE DATA,CVC=1,ACK COMMAND
;
;*****
      MOV     #140005,T27PK3    ;WRITE DATA,CVC=1,ACK COMMAND
      MOV     #T27PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
      MOV     R3,T27SZ          ;SET UP RECORD SIZE IN PACKET
      MOV     R4,TSDB(R5)       ;ISSUE COMMAND
      JSR     PC,WAITF          ;WAIT FOR SSR TO SET
      MOV     TSSR(R5),R1        ;GET TSSR CONTENTS
      MOV     #SSR,R2           ;SET UP EXPECTED
      CMP     R1,R2             ;ARE THEY EQUAL
      BEQ     28$                ;BR, IF OK
      INC     FATFLG            ;BUMP COUNT
      ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP  C$ERHRD
                                .WORD 710
                                .WORD WRERR
                                .WORD PKTSSR
28$:  CKLOOP                    ;LOOP IF SELECTED
                                TRAP  C$CLP1
;*****
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
      JSR     PC,REWIND          ;CALL TAPE REWIND COMMAND
      BCS     30$                ;BR, IF NO PROBLEM
      MOV     TSSR(R5),R1        ;GET TSSR CONTENTS
      MOV     R0,R4              ;SET UP REWIND PACKET ADDRESS
      INC     FATFLG            ;BUMP COUNT
      ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP  C$ERHRD

```

TEST 7: WRITE DATA RETRY

```

077074 001307
077076 104245
077100 012166
13451 077102 104406
077102 104406
13452
13453
13454
13455
13456
13457
13458
13459 077104 013701 102740
13460 077110 010102
13461 077112 052702 000002
13462 077116 020102
13463 077120 001406
13464 077122 005237 002212
13468 077126
077126 104456
077130 001310
077132 103741
077134 015614
13469 077136
077136 104406
13470
13471
13472
13473
13474
13475
13476
13477
13478 077140 012703 000001
13479 077144 004737 010560
13480 077150 103413
13481 077152 016501 000002
13482 077156 012702 000200
13483 077162 010004
13484 077164 005237 002212
13488 077170
077170 104456
077172 001311
077174 105507
077176 012166
13489 077200
077200 104406
13490
13491
13492
13493
13494
13495
13496
13497
13498 077202 012703 100001
13499 077206 004737 010560

```

```

304: CKLOOP ;LOOP IF SELECTED
;*****
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
;*****
MOV T27BFR+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 404 ;BR, IF EQUAL (OK)
INC 404 ;BUMP COUNT
ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
TRAP C#ERHRD
.WORD 712
.WORD T27BOT
.WORD EXPREC

404: CKLOOP ;LOOP IF SELECTED
;*****
;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
;*****
MOV #1,R3 ;PARAMETER SPACE FORWARD 1 RECORD
JSR PC,SPACE ;CALL SPACE RECORDS ROUTINE
BCS 504 ;BR, IF NO ERRORS
MOV TSSR(R5),R1 ;GET TSSR CONTENTS
MOV #SSR,R2 ;SET UP EXPECTED
MOV R0,R4 ;SET UP REWIND PACKET ADDRESS
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T27SCF,PKTSSR ;SPACE RECORDS COMMAND FAILED
TRAP C#ERHRD
.WORD 713
.WORD T27SCF
.WORD PKTSSR

504: CKLOOP ;LOOP IF SELECTED
;*****
;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
;*****
MOV #100001,R3 ;PARAMETER SPACE REVERSE 1 RECORD
JSR PC,SPACE ;CALL SPACE RECORDS ROUTINE

```


TEST 7: WRITE DATA RETRY

```

13601
13602
13603
13604
13605
13606 077510 004737 011136
13607 077514 103407
13608 077516 010004
13609 077520 005237 002212
13613 077524
      077524 104456
      077526 001317
      077530 104245
      077532 012166
13614 077534
      077534 104406
13615
13616
13617
13618
13619
13620
13621
13622 077536 013701 102740
13623 077542 010102
13624 077544 052702 000002
13625 077550 020102
13626 077552 001406
13627 077554 005237 002212
13631 077560
      077560 104456
      077562 001320
      077564 103741
      077566 015614
13632 077570
      077570 104406
13633 077572 012703 000024
13634 077576 013737 003116 103042
13635
13636
13637
13638
13639
13640
13641
13642 077604 012737 140005 103040 65:
13643 077612 012704 103040
13644 077616 010300
13645 077620 004737 017542
13646 077624 010337 103046
13647 077630 010465 000000
13648 077634 004737 016370
13649 077640 016501 000002
13650 077644 012702 000200
13651 077650 020102
13652 077652 001406
13653 077654 005237 002212

;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
      BCS     30:                ;BR, IF NO PROBLEM
      MOV     R0,R4              ;SET UP REWIND PACKET ADDRESS
      INC     FATFLG             ;BUMP COUNT
      ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
;
;*****
      TRAP   C:ERHRD
      .WORD 719
      .WORD T27RWN
      .WORD PKTSSR
30:   CKLOOP                    ;LOOP IF SELECTED
      TRAP   C:CLP1
;*****
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;
;*****
      MOV     T27BFR-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,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
;
;*****
      TRAP   C:ERHRD
      .WORD 720
      .WORD T27BOT
      .WORD EXPREC
40:   CKLOOP                    ;LOOP IF SELECTED
      TRAP   C:CLP1
      MOV     #20.,R3            ;STARTING RECORD SIZE
      MOV     FREE,T27WB        ;STARTING WRITE BUFFER ADDRESS
;*****
;WRITE DATA,CVC=1,ACK COMMAND
;
;*****
65:   MOV     #140005,T27PK3     ;WRITE DATA,CVC=1,ACK COMMAND
      MOV     #T27PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
      MOV     R3,R0              ;SET PATTERN IN CORRECT REGISTER
      JSR     PC,FILLMEM         ;FILL MEMORY WITH RECORD SIZE
      MOV     R3,T27SZ          ;SET UP RECORD SIZE IN PACKET
      MOV     R4,TSSDB(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     80:                ;BR, IF OK
      INC     FATFLG             ;BUMP COUNT

```

E7

TEST 7: WRITE DATA RETRY

```

13657 077660          ERRHRD  ERRNO,WRTErr,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA
      077660 104456          TRAP          C#ERHRD
      077662 001321          .WORD        721
      077664 005113          .WORD        WRTErr
      077666 012166          .WORD        PKTSSR
13658 077670          80$:   CKLOOP                    ;LOOP IF SELECTED
      077670 104406          TRAP          C#CLP1
13659
13660          ;*****
13661          ;
13662          ;WRITE DATA RETRY,CVC=1,ACK COMMAND
13663          ;
13664          ;*****
13665
13666 077672 012737 141005 103040      MOV      #141005,T27PK3      ;WRITE DATA RETRY,CVC=1,ACK COMMAND
13667 077700 010465 000000          MOV      R4,TSDB(R5)        ;ISSUE COMMAND
13668 077704 004737 016370          JSR      PC,WAITF          ;WAIT FOR SSR TO SET
13669 077710 016501 000002          MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
13670 077714 012702 000200          MOV      #SSR,R2          ;SET UP EXPECTED
13671 077720 020102          CMP      R1,R2            ;ARE THEY EQUAL
13672 077722 001406          BEQ     90$              ;BR, IF OK
13673 077724 005237 002212          INC     FATFLG           ;BUMP COUNT
13677 077730          ERRHRD  ERRNO,T27WRF,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA RETRY
      077730 104456          TRAP          C#ERHRD
      077732 001322          .WORD        722
      077734 105646          .WORD        T27WRF
      077736 012166          .WORD        PKTSSR
13678 077740          90$:   CKLOOP                    ;LOOP IF SELECTED
      077740 104406          TRAP          C#CLP1
13679 077742 005723          TST     (R3)+            ;BUMP RECORD SIZE COUNTER
13680 077744 020327 000050          CMP     R3,#40          ;AT 40 SIZE YET
13681 077750 001315          BNE    65$              ;BR, IF MORE RECORDS TO WRITE
13682
13683          ;*****
13684          ;
13685          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
13686          ;
13687          ;*****
13688
13689 077752 004737 011136          JSR      PC,REWIND        ;CALL TAPE REWIND COMMAND
13690 077756 103407          BCS    230$            ;BR, IF NO PROBLEM
13691 077760 010001          MOV     R0,R1           ;SAVE TSSR
13692 077762 005237 002212          INC     FATFLG           ;BUMP COUNT
13696 077766          ERRHRD  ERRNO,T27RWN,EXPREC      ;REWIND NOT ACCEPTED
      077766 104456          TRAP          C#ERHRD
      077770 001323          .WORD        723
      077772 104245          .WORD        T27RWN
      077774 015614          .WORD        EXPREC
13697 077776          230$:  CKLOOP                    ;LOOP IF SELECTED
      077776 104406          TRAP          C#CLP1
13698
13699          ;*****
13700          ;
13701          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
13702          ;
13703          ;*****
13704

```

F7

TEST 7: WRITE DATA RETRY

```

13705 100000 013701 102740      MOV      T27BFR+6,R1      ;PICK UP XSTO
13706 100004 010102      MOV      R1,R2           ;SET UP EXPECTED
13707 100006 052702 000002      BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
13708 100012 020102      CMP      R1,R2           ;DOES EXP = REC'D
13709 100014 001406      BEQ      240$            ;BR, IF EQUAL (OK)
13710 100016 005237 002212      INC      FATFLG          ;BUMP COUNT
13714 100022      ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    724
                                .WORD    T27BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
      100022 104456
      100024 001324
      100026 103741
      100030 015614
13715 100032      240$: CKLOOP          ;LOOP IF SELECTED
      100032 104406
13716 100034 012703 000024      MOV      #20.,R3         ;STARTING RECORD SIZE
13717 100040 013737 003116 103042  MOV      FREE,T27RB      ;STARTING READ BUFFER ADDRESS
13718
13719      ;*****
13720      ;
13721      ;READ DATA,ACK COMMAND
13722      ;
13723      ;*****
13724
13725 100046 012737 100001 103040 265$: MOV      #100001,T27PK3    ;READ DATA,ACK COMMAND
13726 100054 012704 103040      MOV      #T27PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
13727 100060 010337 103046      MOV      R3,T27SZ        ;SET UP RECORD SIZE IN PACKET
13728 100064 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
13729 100070 004737 016370      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
13730 100074 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
13731 100100 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED
13732 100104 020102      CMP      R1,R2           ;ARE THEY EQUAL
13733 100106 001406      BEQ      280$            ;BR, IF OK
13734 100110 005237 002212      INC      FATFLG          ;BUMP COUNT
13738 100114      ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    725
                                .WORD    RDERR
                                .WORD    PKTSSR
      100114 104456
      100116 001325
      100120 005206
      100122 012166
13739 100124      280$: CKLOOP          ;LOOP IF SELECTED
      100124 104406
                                TRAP      C$CLP1
13740 100126 013702 003116      MOV      FREE,R2         ;GET BUFFER ADDRESS
13741 100132 010304      MOV      R3,R4           ;GET RECORD SIZE
13742 100134 162704 000024      SUB      #20.,R4         ;POINT BACK TO 1ST RECORD
13743 100140 060204      285$: ADD      R2,R4      ;POINT TO 1ST LOC IN BUFFER
13744 100142 021403      CMP      (R4),R3        ;DATA WRITTEN = READ
13745 100144 001410      BEQ      290$            ;BR, IF DATA OK (GOOD)
13746 100146 011401      MOV      (R4),R1        ;PICK UP BAD DATA
13747 100150 010302      MOV      R3,R2           ;SET UP EXPECTED
13748 100152 005237 002212      INC      FATFLG          ;BUMP COUNT
13752 100156      ERRHRD  ERRNO,T27DTA,EXPREC ;DATA IN BUFFER NOT CORRECT
                                TRAP      C$ERHRD
                                .WORD    726
                                .WORD    T27DTA
                                .WORD    EXPREC
      100156 104456
      100160 001326
      100162 105726
      100164 015614
13753 100166      290$: CKLOOP          ;LOOP IF SELECTED
      100166 104406
                                TRAP      C$CLP1
13754 100170 005724      TST      (R4),          ;BUMP TO NEXT ADDRESS
13755 100172 160204      SUB      R2,R4          ;BACK TO RECORD SIZE

```


H7

TEST 7: WRITE DATA RETRY

```

13803 100326          ERRDF  ERRNO,SFIERR,SFIMSG  ;FATAL ERROR TSSR WAS NOT OK
      100326 104455          TRAP  C$ERDF
      100330 001327          .WORD 727
      100332 003652          .WORD SFIERR
      100334 012154          .WORD SFIMSG
13804 100336 013737 002172 102730 20$:  MOV  UNITN,T27DSW  ;SET UP UNIT (DRIVE) NUMBER
13805 100344 012704 102710  MOV  #T27PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
13806
13807 ;*****
13808 ;
13809 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
13810 ;
13811 ;*****
13812
13813 100350 004737 010752      JSR  PC,WRTCHR  ;ISSUE WRITE CHARACTERISTICS
13814 100354 103407          BCS  23$      ;BR, IF COMMAND ISSUED OK
13815 100356 005237 002212  INC  FATFLG   ;BUMP COUNT
13819 100362 010001          MOV  R0,R1    ;SAVE CONTENTS OF TSSR
13820 100364          ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      100364 104456          TRAP  C$ERHRD
      100366 001330          .WORD 728
      100370 005056          .WORD WRTMSG
      100372 012154          .WORD SFIMSG
13821 100374          23$:  CKLOOP  ;LOOP IF SELECTED
      100374 104406          TRAP  C$CLP1
13822
13823 ;*****
13824 ;
13825 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
13826 ;
13827 ;*****
13828
13829 100376 004737 011136      JSR  PC,REWIND ;CALL TAPE REWIND COMMAND
13830 100402 103411          BCS  30$      ;BR, IF NO PROBLEM
13831 100404 016501 000002  MOV  TSSR(R5),R1 ;GET TSSR CONTENTS
13832 100410 010004          MOV  R0,R4    ;GET PACKET ADDRESS
13833 100412 005237 002212  INC  FATFLG   ;BUMP COUNT
13837 100416          ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
      100416 104456          TRAP  C$ERHRD
      100420 001331          .WORD 729
      100422 104245          .WORD T27RWN
      100424 012166          .WORD PKTSSR
13838 100426          30$:  CKLOOP  ;LOOP IF SELECTED
      100426 104406          TRAP  C$CLP1
13839
13840 ;*****
13841 ;
13842 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
13843 ;
13844 ;*****
13845
13846 100430 013701 102740      MOV  T27BFR-6,R1 ;PICK UP XSTO
13847 100434 010102          MOV  R1,R2    ;SET UP EXPECTED
13848 100436 052702 000002  BIS  #BIT1,R2 ;SET BOT BIT IN EXPECTED
13849 100442 020102          CMP  R1,R2    ;DOES EXP = REC'D
13850 100444 001406          BEQ  40$      ;BR, IF EQUAL (OK)
13851 100446 005237 002212  INC  FATFLG   ;BUMP COUNT

```

TEST 7: WRITE DATA RETRY

```

13855 100452          ERRHRD  ERRNO,T27BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      100452 104456          TRAP          C4ERHRD
      100454 001332          .WORD          730
      100456 103741          .WORD          T27BOT
      100460 015614          .WORD          EXPREC
13856 100462          40$:  CKLOOP                    ;LOOP IF SELECTED          TRAP          C4CLP1
      100462 104406
13857 100464 012703 000024          MOV          #20.,R3      ;STARTING RECORD SIZE
13858 100470 013737 003116 103042  MOV          FREE,T27WB  ;STARTING WRITE BUFFER ADDRESS
13859
13860          ;*****
13861          ;
13862          ;WRITE DATA,CVC=1,ACK COMMAND
13863          ;
13864          ;*****
13865
13866 100476 012737 140005 103040 65$:  MOV          #140005,T27PK3  ;WRITE DATA,CVC=1,ACK COMMAND
13867 100504 012704 103040          MOV          #T27PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
13868 100510 010300          MOV          R3,R0      ;SET PATTERN IN CORRECT REGISTER
13869 100512 004737 017542          JSR          PC,FILLMEM  ;FILL MEMORY WITH RECORD SIZE
13870 100516 010337 103046          MOV          R3,T27SZ   ;SET UP RECORD SIZE IN PACKET
13871 100522 010465 000000          MOV          R4,TSDB(R5) ;ISSUE COMMAND
13872 100526 004737 016370          JSR          PC,WAITF   ;WAIT FOR SSR TO SET
13873 100532 016501 000002          MOV          TSSR(R5),R1 ;GET TSSR CONTENTS
13874 100536 012702 000200          MOV          #SSR,R2   ;SET UP EXPECTED
13875 100542 020102          CMP          R1,R2     ;ARE THEY EQUAL
13876 100544 001406          BEQ          80$      ;BR, IF OK
13877 100546 005237 002212          INC          FATFLG    ;BUMP COUNT
13881 100552          ERRHRD  ERRNO,WRERR,PKTSSR  ;TSSR INCORRECT AFTER WRITE DATA
      100552 104456          TRAP          C4ERHRD
      100554 001333          .WORD          731
      100556 005113          .WORD          WRERR
      100560 012166          .WORD          PKTSSR
13882 100562          80$:  CKLOOP                    ;LOOP IF SELECTED          TRAP          C4CLP1
      100562 104406
13883
13884          ;*****
13885          ;
13886          ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
13887          ;
13888          ;*****
13889
13890 100564 012737 111005 103040          MOV          #111005,T27PK3  ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
13891 100572 010465 000000          MOV          R4,TSDB(R5)  ;ISSUE COMMAND
13892 100576 004737 016370          JSR          PC,WAITF   ;WAIT FOR SSR TO SET
13893 100602 016501 000002          MOV          TSSR(R5),R1  ;GET TSSR CONTENTS
13894 100606 012702 000200          MOV          #SSR,R2     ;SET UP EXPECTED
13895 100612 020102          CMP          R1,R2     ;ARE THEY EQUAL
13896 100614 001406          BEQ          90$      ;BR, IF OK
13897 100616 005237 002212          INC          FATFLG    ;BUMP COUNT
13901 100622          ERRHRD  ERRNO,T27WRF,EXPREC  ;TSSR INCORRECT AFTER WRITE DATA RETRY
      100622 104456          TRAP          C4ERHRD
      100624 001334          .WORD          732
      100626 105646          .WORD          T27WRF
      100630 015614          .WORD          EXPREC
13902 100632          90$:  CKLOOP                    ;LOOP IF SELECTED          TRAP          C4CLP1
      100632 104406

```

TEST 7: WRITE DATA RETRY

```

13903 100634 005723          TST      (R3).          ;BUMP RECORD SIZE COUNTER
13904 100636 020327 000050  CMP      R3,#40.       ;AT 40 SIZE YET
13905 100642 001315          BNE      65#           ;BR, IF MORE RECORDS TO WRITE
13906
13907          ;*****
13908          ;
13909          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
13910          ;
13911          ;*****
13912
13913 100644 004737 011136     JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
13914 100650 103411          BCS      230#         ;BR, IF NO PROBLEM
13915 100652 016501 000002     MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
13916 100656 010004          MOV      R0,R4        ;GET PACKET ADDRESS
13917 100660 005237 002212     INC      FATFLG       ;BUMP COUNT
13921 100664          ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
          100664 104456          TRAP    C#ERHRD
          100666 001335          .WORD  733
          100670 104245          .WORD  T27RWN
          100672 012166          .WORD  PKTSSR
13922 100674          230# : CKLOOP          ;LOOP IF SELECTED
          100674 104406          TRAP    C#CLP1
13923
13924          ;*****
13925          ;
13926          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
13927          ;
13928          ;*****
13929
13930 100676 013701 102740     MOV      T27BFR+6,R1  ;PICK UP XSTO
13931 100702 010102          MOV      R1,R2        ;SET UP EXPECTED
13932 100704 052702 000002     BIS      #BIT1,R2     ;SET BOT BIT IN EXPECTED
13933 100710 020102          CMP      R1,R2        ;DOES EXP = REC'D
13934 100712 001406          BEQ      240#         ;BR, IF EQUAL (OK)
13935 100714 005237 002212     INC      FATFLG       ;BUMP COUNT
13939 100720          ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          100720 104456          TRAP    C#ERHRD
          100722 001336          .WORD  734
          100724 103741          .WORD  T27BOT
          100726 015614          .WORD  EXPREC
13940 100730          240# : CKLOOP          ;LOOP IF SELECTED
          100730 104406          TRAP    C#CLP1
13941 100732 012703 000024     MOV      #20.,R3      ;STARTING RECORD SIZE
13942 100736 013737 003116 103042  MOV      FREE,T27RB   ;STARTING READ BUFFER ADDRESS
13943
13944          ;*****
13945          ;
13946          ;READ DATA,ACK COMMAND
13947          ;
13948          ;*****
13949
13950 100744 012737 100001 103040 265# : MOV      #100001,T27PK3 ;READ DATA,ACK COMMAND
13951 100752 012704 103040     MOV      #T27PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
13952 100756 010337 103046     MOV      R3,T27SZ     ;SET UP RECORD SIZE IN PACKET
13953 100762 010465 000000     MOV      R4,TSD8(R5)  ;ISSUE COMMAND
13954 100766 004737 016370     JSR      PC,WAITF     ;WAIT FOR SSR TO SET
13955 100772 016501 000002     MOV      TSSR(R5),R1 ;GET TSSR CONTENTS

```


M7

TEST 7: WRITE DATA RETRY

```

14056 101244          ERRDF  ERRNO,SFIERR,SFIMSG      ;FATAL ERROR TSSR WAS NOT OK
      101244 104455          TRAP                      C#ERDF
      101246 001341          .WORD                    737
      101250 003652          .WORD                    SFIERR
      101252 012154          .WORD                    SFIMSG
14057 101254 013737 002172 102730 204:  MOV      UNITN,T27DSW      ;SET UP UNIT NUMBER
14058
14059 101262 012704 102710          MOV      @T27PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
14060
14061          ;*****
14062          ;
14063          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
14064          ;
14065          ;*****
14066
14067 101266 004737 010752          JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
14068 101272 103407          BCS      234           ;BR, IF COMMAND ISSUED OK
14069 101274 005237 002212          INC      FATFLG        ;BUMP COUNT
14073 101300 010001          MOV      R0,R1         ;SAVE CONTENTS OF TSSR
14074 101302          ERRHRD  ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTIC FAILED
      101302 104456          TRAP                      C#ERHRD
      101304 001342          .WORD                    738
      101306 005056          .WORD                    WRTMSG
      101310 012154          .WORD                    SFIMSG
14075 101312          234:  CKLOOP          ;LOOP IF SELECTED
      101312 104406          TRAP                      C#CLP1
14076
14077          ;*****
14078          ;
14079          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
14080          ;
14081          ;*****
14082
14083 101314 004737 011136          JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
14084 101320 103411          BCS      304           ;BR, IF NO PROBLEM
14085 101322 016501 000002          MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
14086 101326 010004          MOV      R0,R4         ;GET PACKET ADDRESS
14087 101330 005237 002212          INC      FATFLG        ;BUMP COUNT
14091 101334          ERRHRD  ERRNO,T27RWN,PKTSSR      ;REWIND NOT ACCEPTED
      101334 104456          TRAP                      C#ERHRD
      101336 001343          .WORD                    739
      101340 104245          .WORD                    T27RWN
      101342 012166          .WORD                    PKTSSR
14092 101344          304:  CKLOOP          ;LOOP IF SELECTED
      101344 104406          TRAP                      C#CLP1
14093
14094          ;*****
14095          ;
14096          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
14097          ;
14098          ;*****
14099
14100 101346 013701 102740          MOV      T27BFR+6,R1  ;PICK UP XSTO
14101 101352 010102          MOV      R1,R2         ;SET UP EXPECTED
14102 101354 052702 000002          BIS      #BIT1,R2     ;SET BOT BIT IN EXPECTED
14103 101360 020102          CMP      R1,R2         ;DOES EXP = REC'D
14104 101362 001406          BEQ      404           ;BR, IF EQUAL (OK)

```

N7

TEST 7: WRITE DATA RETRY

```

14105 101364 005237 002212          INC    FATFLG          ;BUMP COUNT
14109 101370          ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      101370 104456          TRAP    C4ERHRD
      101372 001344          .WORD  740
      101374 103741          .WORD  T27BOT
      101376 015614          .WORD  EXPREC
14110 101400          40$:   CKLOOP          ;LOOP IF SELECTED
      101400 104406          TRAP    C4CLP1
14111 101402 012703 000144          MOV    #100,R3          ;NUMBER OF RECORDS TO BE WRITTEN
14112 101406 013737 003116 103042    MOV    FREE,T27WB      ;STARTING WRITE BUFFER ADDRESS
14113
14114          ;*****
14115          ;
14116          ;WRITE DATA,ACK,CVC=1 COMMAND
14117          ;
14118          ;*****
14119
14120 101414 012737 140005 103040    65$:   MOV    #140005,T27PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
14121 101422 012704 103040          MOV    #T27PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
14122 101426 012737 000024 103046    MOV    #20,T27SZ       ;SET UP RECORD SIZE IN PACKET
14123 101434 010465 000000          MOV    R4,T5DB(R5)     ;ISSUE COMMAND
14124 101440 004737 016370          JSR    PC,WAITF        ;WAIT FOR SSR TO SET
14125 101444 016501 000002          MOV    T5SR(R5),R1     ;GET T5SR CONTENTS
14126 101450 012702 000200          MOV    #SSR,R2        ;SET UP EXPECTED
14127 101454 020102          CMP    R1,R2          ;ARE THEY EQUAL
14128 101456 001406          BEQ    70$            ;BR, IF OK
14129 101460 005237 002212          INC    FATFLG          ;BUMP COUNT
14133 101464          ERRHRD  ERRNO,WRERR,PKTSSR ;T5SR INCORRECT AFTER WRITE DATA
      101464 104456          TRAP    C4ERHRD
      101466 001345          .WORD  741
      101470 005113          .WORD  WRERR
      101472 012166          .WORD  PKTSSR
14134 101474          70$:   CKLOOP          ;LOOP IF SELECTED
      101474 104406          TRAP    C4CLP1
14135 101476 005303          DEC    R3              ;DEC RECORD COUNTER
14136 101500 001345          BNE    65$            ;BR, IF MORE RECORDS TO WRITE
14137
14138          ;*****
14139          ;
14140          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
14141          ;
14142          ;*****
14143
14144 101502 004737 011136          JSR    PC,REWIND       ;CALL TAPE REWIND COMMAND
14145 101506 103411          BCS    130$           ;BR, IF NO PROBLEM
14146 101510 016501 000002          MOV    T5SR(R5),R1     ;GET T5SR CONTENTS
14147 101514 010004          MOV    R0,R4          ;GET PACKET ADDRESS
14148 101516 005237 002212          INC    FATFLG          ;BUMP COUNT
14152 101522          ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
      101522 104456          TRAP    C4ERHRD
      101524 001346          .WORD  742
      101526 104245          .WORD  T27RWN
      101530 012166          .WORD  PKTSSR
14153 101532          130$: CKLOOP          ;LOOP IF SELECTED
      101532 104406          TRAP    C4CLP1
14154
14155          ;*****

```

TEST 7: WRITE DATA RETRY

```

14156
14157 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
14158 ;
14159 ;*****
14160
14161 101534 013701 102740          MOV      T27BFR+6,R1          ;PICK UP XSTO
14162 101540 010102          MOV      R1,R2              ;SET UP EXPECTED
14163 101542 052702 000002        BIS      #BIT1,R2           ;SET BOT BIT IN EXPECTED
14164 101546 020102          CMP      R1,R2              ;DOES EXP = REC'D
14165 101550 001406          BEQ     140$                ;BR, IF EQUAL (OK)
14166 101552 005237 002212        INC      FATFLG             ;BUMP COUNT
14170 101556          ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP     C$ERHRD
                                .WORD    743
                                .WORD    T27BOT
                                .WORD    EXPREC
14171 101566          140$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP1
14172 101570 012704 103040          MOV      #T27PK3,R4         ;SET UP PACKET ADDRESS
14173 101574 012737 000010 103042  MOV      #10,T27RB         ;SET UP RECORDS TO SPACE OVER
14174
14175 ;*****
14176 ;
14177 ;ACK,CVC=1,SPACE FORWARD COMMAND
14178 ;
14179 ;*****
14180
14181 101602 012737 140010 103040          MOV      #140010,T27PK3    ;ACK,CVC=1,SPACE FORWARD COMMAND
14182 101610 010465 000000          150$:  MOV      R4,TSDB(R5) ;ISSUE COMMAND
14183 101614 005237 103066          152$:  INC      T27CNT      ;BUMP TIMER
14184 101620          DELAY  1                ;DELAY ABOUT 100US
                                MOV      #1,(PC)+
                                .WORD    0
                                MOV      L$DLY,(PC)+
                                .WORD    0
                                DEC      -6(PC)
                                BNE     .-4
                                DEC      -22(PC)
                                BNE     .-20
14185 101650 016501 000002          MOV      TSSR(R5),R1       ;GET TSSR
14186 101654 032701 000200          BIT      #BIT7,R1         ;CHECK FOR TSSR'S SSR SET
14187 101660 001755          BEQ     152$                ;KEEP COUNTING UNTIL SET
14188 101662 016501 000002          MOV      TSSR(R5),R1       ;GET STATUS FROM TSSR
14189 101666 012702 000200          MOV      #SSR,R2          ;SET UP EXPECTED
14190 101672 020201          CMP      R2,R1             ;WAS EVERYTHING OK
14191 101674 001406          BEQ     160$                ;BR, IF ALL IS WELL
14192 101676 005237 002212        INC      FATFLG             ;BUMP COUNT
14196 101702          ERRHRD  ERRNO,T27SCF,PKTSSR ;SPACE FORWARD DIDN'T WORK OUT
                                TRAP     C$ERHRD
                                .WORD    744
                                .WORD    T27SCF
                                .WORD    PKTSSR
14197 101712          160$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP1
14198
14199 ;*****
14200 ;

```

TEST 7: WRITE DATA RETRY

```

14201 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
14202 ;
14203 ;*****
14204
14205 101714 004737 011136 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
14206 101720 004737 016456 JSR PC,CHKTSSR ;SEE HOW TSSR IS
14207 101724 103407 BCS 170$ ;BR, IF NO PROBLEM
14208 101726 010001 MOV R0,R1 ;SAVE TSSR
14209 101730 005237 002212 INC FATFLG ;BUMP COUNT
14213 101734 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERHRD
; .WORD 745
; .WORD T27RWN
; .WORD PKTSSR
14214 101744 170$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
14215
14216 ;*****
14217 ;
14218 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
14219 ;
14220 ;*****
14221
14222 101746 013701 102740 MOV T27BFR-6,R1 ;PICK UP XSTO
14223 101752 010102 MOV R1,R2 ;SET UP EXPECTED
14224 101754 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
14225 101760 020102 CMP R1,R2 ;DOES EXP = REC'D
14226 101762 001406 BEQ 175$ ;BR, IF EQUAL (OK)
14227 101764 005237 002212 INC FATFLG ;BUMP COUNT
14231 101770 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
; TRAP C$ERHRD
; .WORD 746
; .WORD T27BOT
; .WORD EXPREC
14232 102000 175$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
14233 102002 012703 000144 MOV #100,R3 ;STARTING RECORD SIZE
14234 102006 013737 003116 103042 177$: MOV FREE,T27WB ;STARTING WRITE BUFFER ADDRESS
14235
14236 ;*****
14237 ;
14238 ;WRITE DATA,CVC-1,ACK COMMAND
14239 ;
14240 ;*****
14241
14242 102014 012737 140005 103040 MOV #140005,T27PK3 ;WRITE DATA,CVC-1,ACK COMMAND
14243 102022 012704 103040 MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
14244 102026 012737 000024 103046 MOV #20,T27SZ ;SET UP RECORD SIZE IN PACKET
14245 102034 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
14246 102040 004737 016370 JSR PC,WAITF ;WAIT FOR SSR TO SET
14247 102044 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
14248 102050 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
14249 102054 020102 CMP R1,R2 ;ARE THEY EQUAL
14250 102056 001406 BEQ 180$ ;BR, IF OK
14251 102060 005237 002212 INC FATFLG ;BUMP COUNT
14255 102064 ERRHRD ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
; TRAP C$ERHRD
102064 104456

```

TEST 7: WRITE DATA RETRY

```

102066 001353 .WORD 747
102070 005113 .WORD WRTERR
102072 012166 .WORD PKTSSR
14256 102074 104406 180$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
14257 102076 005303 DEC R3 ;COUNT NUMBER OF RECORDS
14258 102100 001342 BNE 177$ ;BR, IF MORE RECORDS TO WRITE
14259
14260 ;*****
14261 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
14262 ;
14263 ;*****
14264
14265
14266 102102 004737 011136 JSR PC,REWIND ;ISSUE REWIND
14267 102106 103411 BCS 182$ ;BR, IF ALL IS WELL
14268 102110 010004 MOV R0,R4 ;GET PACKET ADDRESS
14269 102112 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
14270 102116 005237 002212 INC FATFLG ;BUMP COUNT
14274 102122 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND FAILED
102122 104456 TRAP C$ERHRD
102124 001354 .WORD 748
102126 104245 .WORD T27RWN
102130 012166 .WORD PKTSSR
14275 102132 182$: CKLOOP ;SELECT LOOP MAYBE TRAP C$CLP1
102132 104406
14276
14277 ;*****
14278 ;
14279 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
14280 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
14281 ;
14282 ;*****
14283
14284 102134 012703 000001 MOV #1.,R3 ;SPACE 1 RECORD FORWARD
14285 102140 004737 010560 JSR PC,SPACE ;ISSUE SPACE COMMAND
14286 102144 103411 BCS 185$ ;BR, IF COMMAND OK
14287 102146 010004 MOV R0,R4 ;GET PACKET ADDRESS
14288 102150 016501 000002 MOV TSSR(R5),R1 ;GET TSSR STATUS
14289 102154 005237 002212 INC FATFLG ;BUMP COUNT
14293 102160 ERRHRD ERRNO,T27SCF,PKTSSR ;SPACE FORWARD COMMAND FAILED
102160 104456 TRAP C$ERHRD
102162 001355 .WORD 749
102164 105507 .WORD T27SCF
102166 012166 .WORD PKTSSR
14294 102170 185$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
102170 104406
14295 102172 012703 000144 MOV #100.,R3 ;NUMBER OF RECORDS TO BE WRITTEN
14296 102176 013737 003116 103042 MOV FREE,T27WB ;STARTING WRITE BUFFER ADDRESS
14297
14298 ;*****
14299 ;
14300 ;WRITE DATA RETRY,ACK COMMAND
14301 ;
14302 ;*****
14303
14304 102204 012737 101005 103040 190$: MOV #101005,T27PK3 ;WRITE DATA RETRY,ACK COMMAND

```

TEST 7: WRITE DATA RETRY

```

14305 102212 012704 103040          MOV      #T27PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
14306 102216 012737 000024 103046  MOV      #20.,T27SZ         ;SET UP RECORD SIZE IN PACKET
14307 102224 010465 000000          MOV      R4,TSD8(R5)        ;ISSUE COMMAND
14308 102230 004737 016370          JSR      PC,WAITF           ;WAIT FOR SSR TO SET
14309 102234 016501 000002          MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
14310 102240 012702 000200          MOV      #SSR,R2          ;SET UP EXPECTED
14311 102244 020102                   CMP      R1,R2             ;ARE THEY EQUAL
14312 102246 001406                   BEQ      200$              ;BR, IF OK
14313 102250 005237 002212          INC      FATFLG            ;BUMP COUNT
14317 102254                   ERRHRD  ERRNO,T27WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C4ERHRD
                                .WORD    750
                                .WORD    T27WDC
                                .WORD    PKTSSR
                                TRAP      C4CLP1
14318 102264                   200$:  CKLOOP              ;LOOP IF SELECTED
14319 102266 013737 003116 103042  MOV      FREE,T27WB        ;STARTING WRITE BUFFER ADDRESS
14320
14321 ;*****
14322 ;WRITE DATA,CVC=1,ACK COMMAND
14323 ;
14324 ;*****
14325
14326
14327 102274 012737 140005 103040  MOV      #140005,T27PK3    ;WRITE DATA,CVC=1,ACK COMMAND
14328 102302 012704 103040          MOV      #T27PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
14329 102306 012737 000024 103046  MOV      #20.,T27SZ         ;SET UP RECORD SIZE IN PACKET
14330 102314 010465 000000          MOV      R4,TSD8(R5)        ;ISSUE COMMAND
14331 102320 004737 016370          JSR      PC,WAITF           ;WAIT FOR SSR TO SET
14332 102324 016501 000002          MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
14333 102330 012702 000200          MOV      #SSR,R2          ;SET UP EXPECTED
14334 102334 020102                   CMP      R1,R2             ;ARE THEY EQUAL
14335 102336 001406                   BEQ      210$              ;BR, IF OK
14336 102340 005237 002212          INC      FATFLG            ;BUMP COUNT
14340 102344                   ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C4ERHRD
                                .WORD    751
                                .WORD    WRERR
                                .WORD    PKTSSR
                                TRAP      C4CLP1
14341 102354                   210$:  CKLOOP              ;LOOP IF SELECTED
14342 102356 005303                   DEC      R3                ;BUMP DOWN RECORD COUNTER
14343 102360 001311                   BNE     190$              ;BR, IF MORE RECORDS TO WRITE RETRY
14344
14345 ;*****
14346 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
14347 ;
14348 ;*****
14349
14350
14351 102362 004737 011136          JSR      PC,REWIND         ;CALL TAPE REWIND COMMAND
14352 102366 103411                   BCS     230$              ;BR, IF NO PROBLEM
14353 102370 016501 000002          MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
14354 102374 010004                   MOV      R0,R4            ;GET PACKET ADDRESS
14355 102376 005237 002212          INC      FATFLG            ;BUMP COUNT
14359 102402                   ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C4ERHRD
                                .WORD    751
                                .WORD    WRERR
                                .WORD    PKTSSR

```


F8

TEST 7: WRITE DATA RETRY

```

102404 001360 .WORD 752
102406 104245 .WORD T27RWN
102410 012166 .WORD PKTSSR
14360 102412 2304: CKLOOP ;LOOP IF SELECTED TRAP C4CLP1
102412 104406
14361
14362 ;*****
14363 ;
14364 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
14365 ;
14366 ;*****
14367
14368 102414 013701 102740 MOV T27BFR+6,R1 ;PICK UP XSTO
14369 102420 010102 MOV R1,R2 ;SET UP EXPECTED
14370 102422 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
14371 102426 020102 CMP R1,R2 ;DOES EXP = REC'D
14372 102430 001406 BEQ 2404 ;BR, IF EQUAL (OK)
14373 102432 005237 002212 INC FATFLG ;BUMP COUNT
14377 102436 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
102436 104456 TRAP C4ERHRD
102440 001361 .WORD 753
102442 103741 .WORD T27BOT
102444 015614 .WORD EXPREC
14378 102446 2404: CKLOOP ;LOOP IF SELECTED TRAP C4CLP1
102446 104406
14379 102450 012704 103040 MOV #T27PK3,R4 ;SET UP PACKET ADDRESS
14380 102454 012737 000010 103042 MOV #10,T27RB ;SET UP RECORDS TO SPACE OVER
14381
14382 ;*****
14383 ;
14384 ;ACK,CVC=1,SPACE FORWARD COMMAND
14385 ;
14386 ;*****
14387
14388 102462 012737 140010 103040 MOV #140010,T27PK3 ;ACK,CVC=1,SPACE FORWARD COMMAND
14389 102470 010465 000000 2504: MOV R4,TSDB(R5) ;ISSUE COMMAND
14390 102474 005237 103070 2524: INC T27CNU ;BUMP TIMER
14391 102500 DELAY 1 ;DELAY ABOUT 100US
102500 012727 000001 MOV #1,(PC)-
102504 000000 .WORD 0
102506 013727 002116 MOV L4DLY,(PC)-
102512 000000 .WORD 0
102514 005367 177772 DEC -6(PC)
102520 001375 BNE -4
102522 005367 177756 DEC -22(PC)
102526 001367 BNE -20
14392 102530 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
14393 102534 032701 000200 BIT #BIT7,R1 ;CHECK FOR TSSR'S SSR SET
14394 102540 001755 BEQ 2524 ;KEEP COUNTING UNTIL SET
14395 102542 016501 000002 MOV TSSR(R5),R1 ;GET STATUS FROM TSSR
14396 102546 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
14397 102552 020201 CMP R2,R1 ;WAS EVERYTHING OK
14398 102554 001406 BEQ 2604 ;BR, IF ALL IS WELL
14399 102556 005237 002212 INC FATFLG ;BUMP COUNT
14403 102562 ERRHRD ERRNO,T27SCF,PKTSSR ;SPACE FORWARD DIDN'T WORK OUT
102562 104456 TRAP C4ERHRD
102564 001362 .WORD 754

```


TEST 7: WRITE DATA RETRY

```

14454
14456 103014
14458 103020
14459 103020 100006
14460 103022 103050
14461 103024 000000
14462 103026 000006
14463
14465 103030
14467 103040
14468 103040 100005
14469 103042
14470 103042 003116
14471 103044 000000
14472 103046 000000
14473
14474
14475
14476
14477 103050
14478 103050 010
14479 103051 200
14480 103052 000000
14481 103054 000000
14482
14483
14484
14485
14486
14487 103056 100205
14488 103060 100605
14489 103062 102205
14490 103064 177777
14491
14492
14493 103066 000000
14494 103070 000000
14495 103072 000000
14496
14497
14498
14499
14500
14501
14502
14503 103074 124 141 160
14504 103162 124 123 123
14505 103231 122 105 122
14506 103326 120 117 123
14507 103410 122 111 102
14508 103460 124 123 123
14509 103535 111 154 154
14510 103616 122 105 122
14511 103652 124 123 123
14512 103741 124 141 160
14513 104034 127 122 111
14514 104111 122 105 122

```

```

;
T27PK2: .BLKB 10-<.-TSV2&7>
        .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: .BLKB 10-<.-TSV2&7>
        .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
;
;*
;LOCAL TEXT MESSAGES FOR TEST
;'-
T27WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
T27RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
T27RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
T27SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
T27LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
T27WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
T27LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
T27SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
T27WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command,At BOT'
T27BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
T27TIM: .ASCIZ 'WRITE DATA RETRY'S Erase Tape Not Long Enough'
T27EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'

```

TEST 7: WRITE DATA RETRY

14515	104170	124	123	123	T27TM:	.ASCIZ	'TSSR Not Correct After REREAD COMMAND Reject'
14516	104245	122	145	167	T27RW:	.ASCIZ	'Rewind (POSITION) Command Not Accepted'
14517	104314	122	101	115	T27RNC:	.ASCIZ	'RAM Error, Correct Data Pattern Not In Ram'
14518	104367	124	123	123	T27AM3:	.ASCIZ	'TSSR Init. Failed After REREAD COMMAND'
14519	104436	104	162	151	T27OFL:	.ASCIZ	'Drive 7 Select Failed To Set "OFL" In TSSR'
14520	104511	124	123	123	T27WDD:	.ASCIZ	'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
14521	104601	124	123	123	T27WDC:	.ASCIZ	'TSSR Not Correct After REREAD DATA Command'
14522	104654	103	126	103	T27VCK:	.ASCIZ	'CVC Set, Didn't Reset VCK In Message Buffer'
14523	104727	124	123	102	T27BA:	.ASCIZ	'TSBA Not Correct After REREAD DATA Command'
14524	105002	127	122	111	T27WSS:	.ASCIZ	'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
14525	105071	122	145	141	T27LON:	.ASCIZ	'Reading Long Record Failed To Set RLL Bit In XST0'
14526	105153	122	145	141	T27LOP:	.ASCIZ	'Reading Long Record Failed To Set RLS Bit In XST0'
14527	105235	122	145	163	T27PBP:	.ASCIZ	'Residual Byte Count Incorrect After Short Record Read'
14528	105323	122	145	141	T27TRL:	.ASCIZ	'Reading Long Record Failed To Give Tape Status Alert'
14529	105411	127	122	111	T27NEF:	.ASCIZ	'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
14530	105507	124	123	123	T27SCF:	.ASCIZ	'TSSR Not Correct After SPACE RECORDS Command'
14531	105564	124	123	123	T27TSA:	.ASCIZ	'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
14532	105646	124	123	123	T27WRF:	.ASCIZ	'TSSR Not Correct After WRITE DATA RETRY Command'
14533	105726	104	141	164	T27DTA:	.ASCIZ	'Data Compare Error, Data Read From Tape Not Equal To Written'
14534	106023	127	162	151	TST27ID:	.ASCIZ	'Write Data Retry'
14535						.EVEN	
14536							
14537							
14538							
14539							
14540							
14541							
14542							
14543	106044				T27REST:		
14544	106044				SAVREG		
14545	106050	012701	102710		MOV	#T27PACKET,R1	;SAVE THE REGISTERS
14546	106054	012721	100004		MOV	#100004,(R1)+	;START OF THE PACKET
14547	106060	012721	102720		MOV	#T27DATA,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK,
14548	106064	005021			CLR	(R1)+	;ADDRESS OF CHARAISTICS DATA BLOCK
14549	106066	012721	000012		MOV	#10,(R1)+	;EXTENDED ADDRESS
14550	106072	012721	102732		MOV	#T27BFR,(R1)+	;SIZE OF DATA BLOCK IN BYTES
14551	106076	005021			CLR	(R1)+	;ADDRESS OF MESSAGE BUFFER
14552	106100	012721	000024		MOV	#20,(R1)+	;LENGTH OF MESSAGE BUFFER
14553	106104	005021			CLR	(R1)+	
14554	106106	012711	000000		MOV	#0,(R1)	;SELECT DRIVE ZERO
14555	106112	012702	000030		MOV	#24,R2	;NUMBER OF LOCATIONS TO BE CLEARED
14556	106116	012762	177777	102732 64#:	MOV	#177777,T27BFR(R2)	;ALL ONES TO MESSAGE BUFFER
14557	106124	005742			TST	-(R2)	;NEXT LOCATION
14558	106126	022702	000000		CMP	#0,R2	;AT END OF LOOP YET
14559	106132	001371			BNE	64#	;KEEP GOING UNTIL DONE
14560	106134	000207			RTS	PC	;RETURN
14561							
14562							
14563	106136				T27RT2:		
14564	106136				SAVREG		
14565	106142	012701	103020		MOV	#T27PK2,R1	;SAVE THE REGISTERS
14566	106146	012721	100006		MOV	#100006,(R1)+	;START OF THE PACKET
14567	106152	012721	103050		MOV	#T27BF2,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK,
14568	106156	005021			CLR	(R1)+	;ADDRESS OF DATA BLOCK
14569	106160	012721	000006		MOV	#6,(R1)+	;EXTENDED ADDRESS
14570	106164	005021			CLR	(R1)+	;SIZE OF DATA BLOCK IN BYTES
14571	106166	012701	103050		MOV	#T27BF2,R1	;POINT TO DATA SEL AREA

TEST 8: WRITE/READ TAPE MARK

14628	106262	004737	113506			JSR	PC,T28REST		;SET COMMAND PACKET
14629	106266	004737	113600			JSR	PC,T28RT2		;SET UP OTHER COMMAND PACKET
14630	106272	004737	113642			JSR	PC,T28RT3		;SET UP OTHER COMMAND PACKET
14631	106276	004737	016114			JSR	PC,SOFINIT		;DO INITIALIZE ON CONTROLLER
14632	106302	103407				BCS	20		;BR IF INIT WAS OK
14633	106304	005237	002212			INC	FATFLG		;BUMP COUNT
14637	106310	010001				MOV	RO,R1		;CONTENTS OF TSSR REGISTER
14638	106312					ERRDF	ERRNO,SFIERR,SFIMSG		;FATAL ERROR TSSR WAS NOT OK
	106312	104455							TRAP C#ERDF
	106314	001441							.WORD 801
	106316	003652							.WORD SFIERR
	106320	012154							.WORD SFIMSG
14639	106322	012737	000007	111440	20	MOV	#7,T28DSW		;SET UP DRIVE NUMBER
14640	106330	012704	111420			MOV	#T28PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS
14641	106334	004737	010752			JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS
14642	106340	103407				BCS	24		;BR, IF COMMAND ISSUED OK
14643	106342	005237	002212			INC	FATFLG		;BUMP COUNT
14647	106346	010001				MOV	RO,R1		;SAVE CONTENTS OF TSSR
14648	106350					ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTIC FAILED
	106350	104456							TRAP C#ERHRD
	106352	001442							.WORD 802
	106354	005056							.WORD WRTMSG
	106356	012154							.WORD SFIMSG
14649	106360				24	CKLOOP			TRAP C#CLP1
14650	106360	104406							
14650	106362	005737	002216			TST	EXTFEA		;CHECK FOR EXTENDED FEATURES SW SWITCH
14651	106366	001044				BNE	50		;BR IF SWITCH IS ON
14652									
14653	106370	112737	000200	111561		MOVB	#200,T28BS1		;WRITE MISCELLANEOUS CONT/READ STATUS
14654	106376	112737	000010	111560		MOVB	#10,T28BS0		;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
14655	106404	012704	111530			MOV	#T28PK2,R4		;WRITE SUBSYS MEM PACKET
14656	106410	010465	000000			MOV	R4,TSD8(R5)		;ISSUE COMMAND
14657	106414	004737	016456			JSR	PC,CHKTSSR		;WAIT FOR SSR
14658	106420	103407				BCS	30		;BR, IF NO ERROR
14659	106422	010001				MOV	RO,R1		;ERROR, SAVE TSSR
14660	106424	005237	002212			INC	FATFLG		;BUMP COUNT
14664	106430					ERRHRD	ERRNO,T28SSR,PKTSSR		;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
	106430	104456							TRAP C#ERHRD
	106432	001443							.WORD 803
	106434	112255							.WORD T28SSR
	106436	012166							.WORD PKTSSR
14665	106440				30	CKLOOP			;LOOP IF SELECTED
	106440	104406							TRAP C#CLP1
14666	106442	012704	111420			MOV	#T28PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS
14667	106446	012737	000007	111440		MOV	#7,T28DSW		;SELECT DRIVE 7
14668	106454	004737	010752			JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS
14669	106460	103407				BCS	50		;BR, IF COMMAND ISSUED OK
14670	106462	005237	002212			INC	FATFLG		;BUMP COUNT
14674	106466	010001				MOV	RO,R1		;SAVE CONTENTS OF TSSR
14675	106470					ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTIC FAILED
	106470	104456							TRAP C#ERHRD
	106472	001444							.WORD 804
	106474	005056							.WORD WRTMSG
	106476	012154							.WORD SFIMSG
14676	106500				50	CKLOOP			;SCOPE LOOP
	106500	104406							TRAP C#CLP1
14677	106502	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS

M8

TEST 8: WRITE/READ TAPE MARK

```

14727 106652 004737 113642      JSR      PC,T28RT3      ;SET UP OTHER COMMAND PACKET
14728 106656 004737 016114      JSR      PC,SOFINIT    ;DO INITIALIZE ON CONTROLLER
14729 106662 103407              BCS      20$           ;BR IF INIT WAS OK
14730 106664 005237 002212      INC      FATFLG        ;BUMP COUNT
14734 106670 010001              MOV      R0,R1         ;CONTENTS OF TSSR REGISTER
14735 106672      ERRDF      ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      106672 104455              TRAP     C$ERDF
      106674 001447              .WORD   807
      106676 003652              .WORD   SFIERR
      106700 012154              .WORD   SFIMSG
14736 106702 013737 002172 111440 20$:  MOV      UNITN,T28DSW  ;SET UP DRIVE NUMBER
14737 106710 012704 111420      MOV      @T28PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
14738 106714 004737 010752      JSR      PC,WRTCHR     ;ISSUE WRITE CHARACTERISTICS
14739 106720 103407              BCS      24$           ;BR, IF COMMAND ISSUED OK
14740 106722 005237 002212      INC      FATFLG        ;BUMP COUNT
14744 106726 010001              MOV      R0,R1         ;SAVE CONTENTS OF TSSR
14745 106730      ERRHRD      ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      106730 104456              TRAP     C$ERHRD
      106732 001450              .WORD   808
      106734 005056              .WORD   WRTMSG
      106736 012154              .WORD   SFIMSG
14746 106740      24$:  CKLOOP      ;LOOP IF SELECTED
      106740 104406              TRAP     C$CLP1
14747 106742 012703 111566      MOV      @T28IMV,R3    ;SET POINTER FOR COMMANDS
14748 106746 011337 111550      MOV      (R3),T28PK3   ;SET UP NEXT COMMAND
14749 106752 013737 003116 111552 30$:  MOV      FREE,T28RB    ;STARTING WRITE BUFFER ADDRESS
14750 106760 012704 111550      MOV      @T28PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
14751 106764 012737 000400 111556      MOV      @256,T28SZ    ;SET UP RECORD SIZE IN PACKET
14752 106772 010465 000000      MOV      R4,T28B(R5)   ;ISSUE COMMAND
14753 106776 004737 016370      JSR      PC,WAITF      ;WAIT FOR SSR!BIT1!BIT2 TO SET
14754 107002 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
14755 107006 012702 100206      MOV      @SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
14756 107012 020102              CMP      R1,R2         ;ARE THEY EQUAL
14757 107014 001406              BEQ     75$           ;BR, IF OK
14758 107016 005237 002212      INC      FATFLG        ;BUMP COUNT
14762 107022      ERRHRD      ERRNO,T28WDF,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      107022 104456              TRAP     C$ERHRD
      107024 001451              .WORD   809
      107026 112117              .WORD   T28WDF
      107030 012166              .WORD   PKTSSR
14763 107032      75$:  CKLOOP      ;LOOP IF SELECTED
      107032 104406              TRAP     C$CLP1
14764 107034 013701 111450      MOV      T28BFR+6,R1   ;GET MESSAGE BUFFER
14765 107040 010102              MOV      R1,R2         ;SET UP EXPECTED
14766 107042 052702 001000      BIS      @BIT9,R2      ;SET THE ILC BIT IN EXPECTED
14767 107046 020102              CMP      R1,R2         ;ARE THEY EQUAL
14768 107050 001406              BEQ     180$          ;BR, IF EQUAL (ALL IS WELL)
14769 107052 005237 002212      INC      FATFLG        ;BUMP COUNT
14773 107056      ERRHRD      ERRNO,T28LOG,EXPREC ;THE ILC BIT WAS NOT SET IN XSTO
      107056 104456              TRAP     C$ERHRD
      107060 001452              .WORD   810
      107062 112174              .WORD   T28LOG
      107064 015614              .WORD   EXPREC
14774 107066      180$: CKLOOP      ;LOOP IF SELECTED
      107066 104406              TRAP     C$CLP1
14775 107070 005723              TST     (R3),          ;BUMP TO NEXT ADDRESS (COMMAND)
14776 107072 021327 177777      CMP     (R3),@177777  ;CHECK FOR END OF COMMAND BUFFER

```


TEST 8: WRITE/READ TAPE MARK

15071	110340	012704	111550	MOV	#T28PK3,R4	;SET UP R4 WITH PACKET ADDRESS			
15072	110344	010465	000000	MOV	R4,TSD8(R5)	;ISSUE COMMAND			
15073	110350	004737	016370	JSR	PC,WAITF	;WAIT FOR SSR TO SET			
15074	110354	016501	000002	MOV	TSSR(R5),R1	;GET TSSR CONTENTS			
15075	110360	012702	100204	MOV	#SSR!SC!BIT2,R2	;SET UP EXPECTED			
15076	110364	020102		CMP	R1,R2	;ARE THEY EQUAL			
15077	110366	001406		BEQ	222+	;BR, IF OK			
15078	110370	005237	002212	INC	FATFLG	;BUMP COUNT			
15082	110374			ERRHRD	ERRNO,T28RDG,PKTSSR	;TSSR INCORRECT AFTER SPACE CMD.			
	110374	104456				TRAP	C#ERHRD		
	110376	001475				.WORD	829		
	110400	112035				.WORD	T28RDG		
	110402	012166				.WORD	PKTSSR		
15083	110404			222+:	CKLOOP	;LOOP IF SELECTED			
	110404	104406				TRAP	C#CLP1		
15084	110406	013701	111450	MOV	T28BFR+6,R1	;PICK UP XST0			
15085	110412	010102		MOV	R1,R2	;SET UP EXPECTED			
15086	110414	052702	100000	BIS	#BIT15,R2	;TMK SHOULD BE SET			
15087	110420	020102		CMP	R1,R2	;IS TMK SET			
15088	110422	001406		BEQ	226+	;BR, IF TMK WAS SET (GOOD)			
15089	110424	005237	002212	INC	FATFLG	;BUMP COUNT			
15093	110430			ERRHRD	ERRNO,T28RRN,EXPREC	;TMK NOT SET AFTER SPACE REV			
	110430	104456				TRAP	C#ERHRD		
	110432	001476				.WORD	830		
	110434	113145				.WORD	T28RRN		
	110436	015614				.WORD	EXPREC		
15094	110440			226+:	CKLOOP	;LOOP IF SELECTED			
	110440	104406				TRAP	C#CLP1		
15095	110442	004737	011136	JSR	PC,REWIND	;CALL TAPE REWIND COMMAND			
15096	110446	103411		BCS	230+	;BR, IF NO PROBLEM			
15097	110450	010004		MOV	R0,R4	;SAVE PACKET ADDRESS			
15098	110452	016501	000002	MOV	TSSR(R5),R1	;GET TSSR			
15099	110456	005237	002212	INC	FATFLG	;BUMP COUNT			
15103	110462			ERRHRD	ERRNO,T28RWN,PKTSSR	;REWIND NOT ACCEPTED			
	110462	104456				TRAP	C#ERHRD		
	110464	001477				.WORD	831		
	110466	112541				.WORD	T28RWN		
	110470	012166				.WORD	PKTSSR		
15104	110472			230+:	CKLOOP	;LOOP IF SELECTED			
	110472	104406				TRAP	C#CLP1		
15105	110474	013701	111450	MOV	T28BFR+6,R1	;PICK UP XST0			
15106	110500	010102		MOV	R1,R2	;SET UP EXPECTED			
15107	110502	052702	000002	BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED			
15108	110506	020102		CMP	R1,R2	;DOES EXP = REC'D			
15109	110510	001406		BEQ	240+	;BR, IF EQUAL (OK)			
15110	110512	005237	002212	INC	FATFLG	;BUMP COUNT			
15114	110516			ERRHRD	ERRNO,T28BOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND			
	110516	104456				TRAP	C#ERHRD		
	110520	001500				.WORD	832		
	110522	112417				.WORD	T28BOT		
	110524	015614				.WORD	EXPREC		
15115	110526			240+:	CKLOOP	;LOOP IF SELECTED			
	110526	104406				TRAP	C#CLP1		
15116	110530	012700	177777	MOV	#177777,R0	;VALUE TO WRITTEN TO MEMORY			
15117	110534	004737	017542	JSR	PC,FILLMEM	;FILL MEM WITH ALL ONES			
15118	110540	013737	003116	MOV	FREE,T28RB	;STARTING READ BUFFER ADDRESS			111552
15119	110546	012737	100001	MOV	#100001,T28PK3	;READ FORWARD,ACK, COMMAND			111550

H9

TEST 8: WRITE/READ TAPE MARK

```

15120 110554 012704 111550          MOV      #T28PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
15121 110560 013737 000024 111556  MOV      20,T28SZ           ;SET UP RECORD SIZE IN PACKET
15122 110566 010465 000000          MOV      R4,TSDB(R5)       ;ISSUE COMMAND
15123 110572 004737 016370          JSR      PC,WAITF          ;WAIT FOR SSR TO SET
15124 110576 016501 000002          MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
15125 110602 012702 100204          MOV      #SSR!SC!BIT2,R2  ;SET UP EXPECTED
15126 110606 020102          CMP      R1,R2             ;ARE THEY EQUAL
15127 110610 001406          BEQ      245$              ;BR, IF OK
15128 110612 005237 002212          INC      FATFLG            ;BUMP COUNT
15132 110616          ERRHRD  ERRNO,T28WDE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD      833
                                .WORD      T28WDE
                                .WORD      PKTSSR
                                TRAP      C$CLP1
15133 110626          245$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
15134 110630 104406          MOV      T28BFR+6,R1       ;PICK UP XSTO
15135 110634 013701 111450          MOV      R1,R2             ;SET UP EXPECTED
15136 110636 052702 100000          BIS      #BIT15,R2         ;TMK SHOULD BE SET
15137 110642 020102          CMP      R1,R2             ;IS TMK SET
15138 110644 001406          BEQ      247$              ;BR, IF TMK WAS SET (GOOD)
15139 110646 005237 002212          INC      FATFLG            ;BUMP COUNT
15143 110652          ERRHRD  ERRNO,T28RRP,EXPREC ;TMK NOT SET AFTER READ REV
                                TRAP      C$ERHRD
                                .WORD      834
                                .WORD      T28RRP
                                .WORD      EXPREC
                                TRAP      C$CLP1
15144 110662          247$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
15145 110664 017701 072226          MOV      @FREE,R1          ;FIRST LOC IN READ BUFFER
15146 110670 012702 177777          MOV      #177777,R2       ;EXPECTED IF NO DATA TRANS.
15147 110674 020102          CMP      R1,R2             ;DID ANY DATA GET TRANSFERRED
15148 110676 001406          BEQ      250$              ;BR, IF NO DATA TRANS (GOOD)
15149 110700 005237 002212          INC      FATFLG            ;BUMP COUNT
15153 110704          ERRHRD  ERRNO,T28DTR,EXPREC ;DATA TRANSFERRED ON READ TAPE MARK
                                TRAP      C$ERHRD
                                .WORD      835
                                .WORD      T28DTR
                                .WORD      EXPREC
                                TRAP      C$CLP1
15154 110714          250$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
15155 110716 012737 100410 111550  MOV      #100410,T28PK3    ;SPACE REVERSE,ACK, COMMAND
15156 110724 012737 000005 111552  MOV      #5,T28RB          ;NUMBER OF RECORDS TO SPACE BACK
15157 110732 012704 111550          MOV      #T28PK3,R4       ;SET UP R4 WITH PACKET ADDRESS
15158 110736 010465 000000          MOV      R4,TSDB(R5)       ;ISSUE COMMAND
15159 110742 004737 016370          JSR      PC,WAITF          ;WAIT FOR SSR TO SET
15160 110746 016501 000002          MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
15161 110752 012702 100204          MOV      #SSR!SC!BIT2,R2  ;SET UP EXPECTED
15162 110756 020102          CMP      R1,R2             ;ARE THEY EQUAL
15163 110760 001406          BEQ      260$              ;BR, IF OK
15164 110762 005237 002212          INC      FATFLG            ;BUMP COUNT
15168 110766          ERRHRD  ERRNO,T28RDG,PKTSSR ;TSSR INCORRECT AFTER SPACE REV CMD.
                                TRAP      C$ERHRD
                                .WORD      836
                                .WORD      T28RDG
                                .WORD      PKTSSR
                                TRAP      C$CLP1
15169 110776          260$:  CKLOOP              ;LOOP IF SELECTED

```


TEST 8: WRITE/READ TAPE MARK

```

111406 002256 .WORD L10130-.
15266
15267 ;*
15268 ;LOCAL STORAGE FOR THIS TEST
15269 ;-
15271 111410 .BLKB 10-<.-TSV2&7>
15273 111420 T28PACKET: ;COMMAND PACKET FOR TEST
15274 111420 100004 .WORD 100004 ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
15275 111422 111430 .WORD T28DATA ;ADDRESS OF CHARACTERISTICS BLOCK
15276 111424 000000 .WORD 0
15277 111426 000012 .WORD 10. ;STARTING VALUE OF BLOCK SIZE
15278 111430 T28DATA: ;CHARACTERISTICS DATA BLOCK
15279 111430 111442 .WORD T28BFR ;ADDRESS OF MESSAGE BUFFER
15280 111432 000000 .WORD 0
15281 111434 000024 .WORD 20. ;LENGTH OF MESSAGE BUFFER
15282 111436 000000 .WORD 0
15283 111440 000000 T28DSW: .WORD 0 ;SELECT DRIVE 0
15284 111442 T28BFR: .BLKW 25. ;MESSAGE BUFFER
15285
15286 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
15287 ;
15289 111524 .BLKB 10-<.-TSV2&7>
15291 111530 T28PK2: ;WRITE SUB SYS MEM COMMAND, IE AND ACK
15292 111530 100006 .WORD 100006 ;ADDRESS OF SELECT BLOCK DATA
15293 111532 111560 .WORD T28BF2
15294 111534 000000 .WORD 0 ;SIZE OF DATA PACKET
15295 111536 000006 .WORD 6.
15296
15298 111540 .BLKB 10-<.-TSV2&7>
15300 111550 T28PK3: ;REREAD COMMAND, AND ACK
15301 111550 100005 .WORD 100005 ;ADDRESS OF WRITE BUFFER
15302 111552 T28RB: ;SIZE OF BUFFER (EXTENT)
15303 111552 003116 T28WB: .WORD FREE
15304 111554 000000 .WORD 0
15305 111556 000000 T28SZ: .WORD 0
15306 .EVEN
15307 ;
15308 ;
15309 ;
15310 111560 T28BF2: ;BSELO AREA
15311 111560 010 T28BS0: .BYTE 10 ;BSEL1 AREA
15312 111561 200 T28BS1: .BYTE 200 ;SEL 2 AREA
15313 111562 000000 T28S2: .WORD 0 ;DATA AREA
15314 111564 000000 T28S3: .WORD 0
15315 ;
15316 ;
15317 .EVEN
15318 ;TAPE MOTION PACKET COMMAND VALUES
15319 ;
15320 111566 T28IMV: ;ILLEGAL MODE BITS TEST DATA
15321 111566 101411 .WORD 101411
15322 111570 102011 .WORD 102011
15323 111572 103411 .WORD 103411
15324 111574 177777 .WORD 177777
15325 111576 100011 T28RN: .WORD 100011 ;WRITE TAPE MARK COMMAND
15326 111600 100411 T28WDR: .WORD 100411 ;ERASE COMMAND
15327 111602 101011 T28CON: .WORD 101011 ;WRITE TAPE MARK RETRY

```

TEST 8: WRITE/READ TAPE MARK

```

15328 111604 177777          .WORD 177777          ;END OF DATA
15329
15330
15331 111606 000000          T28CNT: .WORD 0          ;TAPE TIMER COUNTER STORAGE AREA
15332 111610 000000          T28CNU: .WORD 0          ;TAPE TIMER COUNTER STORAGE AREA
15333 111612 000000          T28DLY: .WORD 0          ;DELAY COUNTER
15334
15335
15336
15337          ;*
15338          ;LOCAL TEXT MESSAGES FOR TEST
15339          ;-
15340
15341 111614      124      141      160 T28RIB: .ASCIZ 'Tape Position Not Correct, RIB Should Be Set'
15342 111671      122      145      163 T28PBP: .ASCIZ 'Residual Byte Counter Register (RBPCR) Not Correct'
15343 111754      124      123      123 T28RDF: .ASCIZ 'TSSR Incorrect After READ REVERSE Into TAPE MARK'
15344 112035      124      123      123 T28RDG: .ASCIZ 'TSSR Incorrect After SPACE Command Into TAPE MARK'
15345 112117      124      123      123 T28WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
15346 112174      111      154      154 T28LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
15347 112255      127      122      111 T28SSR: .ASCIZ 'WRITE MISCELLANEOUS Command Not Accepted'
15348 112326      124      123      123 T28WDE: .ASCIZ 'TSSR Not Correct After READ DATA Command, Into TAPE MARK'
15349 112417      124      141      160 T28BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
15350 112464      124      123      123 T28TM: .ASCIZ 'TSSR Not Correct After FORMAT Command Reject'
15351 112541      122      145      167 T28RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
15352 112610      104      162      151 T28OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
15353 112663      124      123      123 T28WDC: .ASCIZ 'TSSR Not Correct After WRITE TAPE MARK Command'
15354 112742      103      126      103 T28VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
15355 113015      124      115      113 T28TMK: .ASCIZ 'TMK Not Set After WRITE TAPE MARK Command'
15356 113067      124      115      113 T28RRM: .ASCIZ 'TMK Not Set After READ REVERSE Into TAPE MARK'
15357 113145      124      115      113 T28RRN: .ASCIZ 'TMK Not Set After SPACE REVERSE Into TAPE MARK'
15358 113224      124      115      113 T28RRP: .ASCIZ 'TMK Not Set After READ FORWARD Into TAPE MARK'
15359 113302      104      141      164 T28DTR: .ASCIZ 'Data Transferred On READ REVERSE Into A TAPE MARK'
15360 113364      104      141      164 T28DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
15361 113461      127      162      151 TST28ID: .ASCIZ 'Write/Read Tape Mark'
15362
15363          .EVEN
15364
15365          ;*
15366          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
15367          ;WRITE SUBSYSTEM MEMORY COMMAND
15368          ;-
15369
15370 113506          T28REST:
15371 113506          SAVREG
15372 113512 012701 111420          MOV #T28PACKET,R1          ;SAVE THE REGISTERS
15373 113516 012721 100004          MOV #100004,(R1)          ;START OF THE PACKET
15374 113522 012721 111430          MOV #T28DATA,(R1)          ;WRITE SUBSYSTEM MEM. WITH ACK,
15375 113526 005021          CLR (R1)          ;ADDRESS OF CHARAISTICS DATA BLOCK
15376 113530 012721 000012          MOV #10,(R1)          ;EXTENDED ADDRESS
15377 113534 012721 111442          MOV #T28BFR,(R1)          ;SIZE OF DATA BLOCK IN BYTES
15378 113540 005021          CLR (R1)          ;ADDRESS OF MESSAGE BUFFER
15379 113542 012721 000024          MOV #20,(R1)          ;LENGTH OF MESSAGE BUFFER
15380 113546 005021          CLR (R1)
15381 113550 012711 000000          MOV #0,(R1)          ;SELECT DRIVE ZERO
15382 113554 012702 000030          MOV #24,R2          ;NUMBER OF LOCATIONS TO BE CLEARED
15383 113560 012762 177777 111442 64: MOV #177777,T28BFR(R2)          ;ALL ONES TO MESSAGE BUFFER
15384 113566 005742          TST -(R2)          ;NEXT LOCATION

```

TEST 8: WRITE/READ TAPE MARK

15385 113570 020227 000000
 15386 113574 001371
 15387 113576 000207
 15388
 15389
 15390 113600
 15391 113600
 15392 113604 012701 111530
 15393 113610 012721 100006
 15394 113614 012721 111560
 15395 113620 005021
 15396 113622 012721 000006
 15397 113626 005021
 15398 113630 012701 111560
 15399 113634 005021
 15400 113636 005011
 15401 113640 000207
 15402 113642
 15403 113642
 15404 113646 012701 111550
 15405 113652 005021
 15406 113654 005021
 15407 113656 005021
 15408 113660 005011
 15409 113662 000207
 15410 113664
 113664
 113664 104401
 15411 113666
 15412
 15418
 15423
 15429
 15430 113666
 113666
 15431
 15432
 15433
 15434
 15435
 15436
 15437
 15438
 15439
 15440
 15441
 15442 113666
 113666 000010
 113670
 15443
 15444 113670
 113670 000031
 113672 113710
 113674 160010
 113676 177776
 15445 113700
 113700 001031

```

CMP      R2,#0          ;CHECK FOR END
BNE      64$           ;KEEP GOING UNTIL DONE
RTS      PC            ;RETURN

T28RT2:
SAVREG
MOV      @T28PK2,R1    ;SAVE THE REGISTERS
MOV      @100006,(R1)+ ;START OF THE PACKET
MOV      @T28BF2,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
CLR      (R1)+         ;ADDRESS OF DATA BLOCK
MOV      @6,(R1)+     ;EXTENDED ADDRESS
CLR      (R1)+         ;SIZE OF DATA BLOCK IN BYTES
MOV      @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

L10130: TRAP C#ETST

ENDMOD
.TITLE  TSV6 - PARAMETER CODING

BGNMOD  TSV6

TSV6::
.SBTTL  HARDWARE PARAMETER CODING SECTION

; **
; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
; WITH THE OPERATOR.
; --
BGNHRD
.WORD L10134-L#HARD/2
L#HARD::

GPRMA   HPM1,0,0,160010,177776,YES ;GET TSBA/TSDB REGISTER ADDRESS.
.WORD   T#CODE
.WORD   HPM1
.WORD   T#LOLIM
.WORD   T#HILIM
GPRMA   HPM2,2,0,0,776,YES          ;GET VECTOR ADDRESS.
.WORD   T#CODE

```


SOFTWARE PARAMETER CODING SECTION

.SBTTL SOFTWARE PARAMETER CODING SECTION

```

15453
15454
15455
15456
15457
15458
15459
15460
15461
15462
15463 114020
      114020 000003
      114022
15464
15465 114022
      114022 001130
      114024 114060
      114026 177777
15466
15467
15468 114030
      114030
15469
15470 114030 105 116 101
15471 114060 111 116 110
15472 114110 120 105 122
15473 114140 120 105 122
15474
15475
15476
15477
15478
15479
15480
15481
15482
15483 114170
15484
15485 114170
15486
15488 114400 114400
15490 114400
      114400 000000
      114402 000000
      114404
15491 114404
15492 000001

```

```

; **
; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
; WITH THE OPERATOR.
; --
      BGNSFT
      .WORD L10135-L$SOFT/2
L$SOFT::
;      GPRML SPM1,0,-1,YES ; GET TRANSPORT TEST FLAG.
;      GPRML SPM4,2,-1,YES ; GET ITERATION CONTROL.
      .WORD T$CODE
      .WORD SPM4
      .WORD -1
;      GPRMD SPM6,4,D,7777,0,7777,YES ; GET LOCAL ERROR LIMIT
;      GPRMD SPM7,6,D,7777,0,7777,YES ; GET GLOBAL ERROR LIMIT
      ENDSFT
      .EVEN
L10135:
SPM1: .ASCIZ 'ENABLE TRANSPORT TESTS '
SPM4: .ASCIZ 'INHIBIT ITERATIONS '
SPM6: .ASCIZ 'PER TEST ERROR LIMIT '
SPM7: .ASCIZ 'PER UNIT ERROR LIMIT '
      .SBTTL PATCH AREA
;
; FINALLY A GENEROUS PATCH AREA.
;
; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
;
PATCH::
      .BLKW 32.
      .=.!377*1
      LASTAD ;SET LAST USED ADDRESS.
      .EVEN
      .WORD 0
      .WORD 0
L$LAST::
      ENDMOD
      .END

```

Symbol table

ADDSSR	012246	G	C#AU	=	000052	DEVDR0	024336	FREE	003116	G	INCERK	017164					
ADR	=	000020	G	C#AUTO	=	000061	DEVNRD	024255	FREEHI	003122	INTCPC	016270					
AMBTSS	006727		C#BRK	=	000022	DEVNXR	024173	FRESIZ	003120	G	INTFLA	016265					
ASSEMB	=	000010	C#BSEG	=	000004	DEVONL	024123	FUSI	004117	INTMAS	016264						
A1716	=	000003	C#BSUB	=	000002	DEVSUM	024066	F#AU	=	000015	INTR	016336	G				
BADDAT	003150	G	C#CEFG	=	000045	DFPTBL	002146	G	F#AUTO	=	000020	INTREC	002214	G			
BADSSR	016020	G	C#CLCK	=	000062	DIAGMC	=	000000	F#BGN	=	000040	INTVEC	016266				
BDVPCR	=	177520	G	C#CLEA	=	000012	DICEC	=	000001	F#CLEA	=	000007	INTX	004300			
BENBSW	002222	G	C#CLOS	=	000035	DSBINT	016324	F#DU	=	000016	INVERT	021302	G				
BIE	=	040000	C#CLP1	=	000006	DUAD12	004643	F#END	=	000041	IOKCKI	=	000200				
BIT0	=	000001	G	C#CVEC	=	000036	DUFLG	003104	G	F#HARD	=	000004	IOKSTP	=	000001		
BIT00	=	000001	G	C#DCLN	=	000044	DUMMY	003054	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	I#AU	=	000041	
BIT05	=	000040	G	C#ERDF	=	000055	EF.STA	=	000040	G	F#PROT	=	000021	I#AUTO	=	000041	
BIT06	=	000100	G	C#ERHR	=	000056	EMAXDU	017117	F#PWR	=	000017	I#CLN	=	000041			
BIT07	=	000200	G	C#ERRO	=	000060	EN	=	000000	F#RPT	=	000012	I#DU	=	000041		
BIT08	=	000400	G	C#ERSF	=	000054	ENAIN	016272	F#SEG	=	000003	I#HRD	=	000041			
BIT09	=	001000	G	C#ERSO	=	000057	ENVIRN	020750	F#SOFT	=	000005	I#INIT	=	000041			
BIT1	=	000002	G	C#ESCA	=	000010	EPRTSW	002170	G	F#SRV	=	000010	I#MOD	=	000041		
BIT10	=	002000	G	C#ESEG	=	000005	EPRT1	006360	F#SUB	=	000002	I#MSG	=	000041			
BIT11	=	004000	G	C#ESUB	=	000003	EPRT2	006450	F#SW	=	000014	I#PROT	=	000040			
BIT12	=	010000	G	C#ETST	=	000001	ERCM	012053	F#TEST	=	000001	I#PTAB	=	000041			
BIT13	=	020000	G	C#EXIT	=	000032	ERRHI	002230	G	GDDAT	003152	I#PWR	=	000041			
BIT14	=	040000	G	C#GETB	=	000026	ERRK	017076	GERRMA	002164	G	I#RPT	=	000041			
BIT15	=	100000	G	C#GETW	=	000027	ERRLO	002232	G	GETPAT	020314	G	I#SEG	=	000041		
BIT2	=	000004	G	C#GMAN	=	000043	ERRNO	=	001513	GETSEL	020376	G	I#SETU	=	000041		
BIT3	=	000010	G	C#GPHR	=	000042	ERRVEC	=	000004	G	G#CNT0	=	000200	I#SFT	=	000041	
BIT4	=	000020	G	C#GPLO	=	000030	ERTABE	003370	G#DELM	=	000372	I#SRV	=	000041			
BIT5	=	000040	G	C#GPRI	=	000040	ERTABL	003170	G#DISP	=	000003	I#SUB	=	000041			
BIT6	=	000100	G	C#INIT	=	000011	ESUM	017100	G#EXCP	=	000400	I#TST	=	000041			
BIT7	=	000200	G	C#INLP	=	000020	EVL	=	000004	G	G#HILI	=	000002	J#JMP	=	000167	
BIT8	=	000400	G	C#MANI	=	000050	EXBCNT	=	000010	G#LOLI	=	000001	KIPAR0	=	172340		
BIT9	=	001000	G	C#MEM	=	000031	EXIT	035264	G#NO	=	000000	KIPAR1	=	172342			
BOE	=	000400	G	C#MSG	=	000023	EXPBRE	015622	G	G#OFFS	=	000400	KIPAR2	=	172344		
BRINIT	004457		C#OPEN	=	000034	EXPD	002224	G	G#OFSI	=	000376	KIPAR3	=	172346			
BSELO	=	000000	C#PNTB	=	000014	EXPGOT	004533	G	G#PRMA	=	000001	KIPAR4	=	172350			
BSEL1	=	000001	C#PNTF	=	000017	EXPGT2	004567	G	G#PRMD	=	000002	KIPAR5	=	172352			
CHKAMB	016164		C#PNTS	=	000016	EXPMMSG	002314	G	G#PRML	=	000000	KIPAR6	=	172354			
CHKMAN	020620	G	C#PNTX	=	000015	EXPREC	015614	G	G#RADA	=	000140	KIPAR7	=	172356			
CHKTSS	016456		C#QIO	=	000377	EXTA	005772	G#RADB	=	000000	KIPDR0	=	172300				
CKDROP	017322		C#RDBU	=	000007	EXTEND	005770	G#RADD	=	000040	KIPDR1	=	172302				
CKEMAX	017222		C#REFG	=	000047	EXTFEA	002216	G	G#RADL	=	000120	KIPDR2	=	172304			
CKMSG	011500	G	C#RESE	=	000033	E#END	=	002100	G#RADO	=	000020	KIPDR3	=	172306			
CKMSG2	011620	G	C#REVI	=	000003	E#LOAD	=	000035	G#XFER	=	000004	KIPDR4	=	172310			
CKRAM	011234	G	C#RFLA	=	000021	FATAL	035364	G#YES	=	000010	KIPDR5	=	172312				
CKRAM2	011344	G	C#RPT	=	000025	FATERR	=	000060	HIADDR	=	001400	KIPDR6	=	172314			
CHDPKT	022220	G	C#SEFG	=	000046	FATFLG	002212	G	HOE	=	100000	G	KIPDR7	=	172316		
CHPMEM	020000		C#SPRI	=	000041	FERCM	012042	HPM1	113710			G	KTENAB	003126	G		
CONFIG	017370		C#SVEC	=	000037	FIFEXP	012310	G	HPM2	113744			G	KTFLG	003124	G	
COUNT	002302	G	C#TPRI	=	000013	FIF1MS	012362	HPM3	113770					KTINIT	021130		
CSRADD	002176	G	DATA	002304	G	FIF2MS	012431	IBE	=	010000	G			KTOFF	017414		
CTAB	003156	G	DATASC	020352		FILLME	017542	IDU	=	000040	G			KTON	017376		
CTABE	003170	G	DEBUG	011752		FNOINT	004215	IER	=	020000	G			LERRMA	002162	G	
CTABM	003156	G	DEVCNT	002210	G	FORCER	002166	G	IFALT	004256				LISTAL	000001		

Symbol table

LOE = 040000 G	L#UNIT 002012 G	L10071 056602	M8189 005645	PRBEXP 015610
LOOPCN 002206 G	L10000 002154	L10072 050644	NBA = 002000	PRBMSG 015456
LOOPCO 013246	L10001 002166	L10073 051244	NEWPAS 022750	PRBREC 015612
LOOPFL 003154 G	L10002 005766	L10074 051720	NODEV 003106 G	PRBTOT 015543
LOT = 000010 G	L10003 012164	L10075 052364	NOINIT 004335	PRBYTE 015242 G
L#ACP 002110 G	L10004 012202	L10076 053124	NOINTR 004221	PRI = 002000 G
L#APT 002036 G	L10005 012220	L10077 054064	NOITS 002160 G	PRIADD 010252
L#AU 023312 G	L10006 012226	L10100 054404	NOMAN 020654	PRIAO 010322
L#AUT 002070 G	L10007 012244	L10101 055006	NOMEM 005460	PRI BXO 007704 G
L#AUTO 023516 G	L10010 012262	L10102 076164	NP.IR = 000200	PRIEQU 010152
L#CCP 002106 G	L10011 012306	L10103 057544	NP.LOO= 000040	PRIPKT 007462 G
L#CLEA 023576 G	L10012 012360	L10104 060412	NP.OUT= 000100	PRIRAM 010160
L#CO 002032 G	L10013 012530	L10105 061304	NP.WRP= 000020	PRITAD 010366
L#DEPO 002011 G	L10014 013244	L10106 062232	NSI 004152	PRITSS 006024
L#DESC 003402 G	L10015 014072	L10107 063010	NSINIT 004407	PRITO 010450
L#DESC 002076 G	L10016 014114	L10110 063652	NUL 004527	PRIT1 010513
L#DEVP 002060 G	L10017 015620	L10111 064524	NULCR 004530	PRI XOR 010034 G
L#DISP 002124 G	L10020 015626	L10112 065376	NXM = 004000	PRI00 = 000000 G
L#DLY 002116 G	L10021 015634	L10113 066252	NXMFLG 003130 G	PRI01 = 000040 G
L#DTP 002040 G	L10022 015646	L10114 067126	NXMH1 003134 G	PRI02 = 000100 G
L#DTYP 002034 G	L10023 015670	L10115 067776	NXMLO 003132 G	PRI03 = 000140 G
L#DU 023410 G	L10024 015716	L10116 070730	NXMTST 022412	PRI04 = 000200 G
L#DUT 002072 G	L10025 016056	L10117 071760	NXR 003740	PRI05 = 000240 G
L#DVTY 003374 G	L10026 016366	L10120 072340	NXRERR 005736 G	PRI06 = 000300 G
L#EF 002052 G	L10030 023242	L10121 073014	NXRX 003777	PRI07 = 000340 G
L#ENVI 002044 G	L10031 023406	L10122 106222	NXTU 022762	PRMESS 014362
L#ETP 002102 G	L10032 023514	L10123 076606	OFL = 000100	PRMNO 002312 G
L#EXP1 002046 G	L10033 023574	L10124 077370	ONEFIL = 000000	PRMSGE 014672 G
L#EXP4 002064 G	L10034 023622	L10125 100212	O#APTS= 000000	PRMSG0 015052
L#EXP5 002066 G	L10035 024064	L10126 101114	O#AU = 000001	PRMSG1 015117
L#HARD 113670 G	L10036 025520	L10127 102644	O#BGNR= 000001	PRMSG2 015155
L#HIME 002120 G	L10037 030200	L10130 113664	O#BGNS= 000001	PROASC 014540
L#HPCP 002016 G	L10040 026126	L10131 106622	O#DU = 000001	PRIASC 014605
L#HPTP 002022 G	L10041 026450	L10132 107102	O#ERRT= 000000	PST32W 003144 G
L#HW 002146 G	L10042 027030	L10133 111354	O#GNSW= 000001	PUNIT 023244
L#ICP 002104 G	L10043 035410	L10134 113710	O#POIN= 000001	PW.D11= 000021
L#INIT 022516 G	L10044 030606	L10135 114030	O#SETU= 000000	PW.D13= 000022
L#LADP 002026 G	L10045 031472	MEMADD 014074 G	PASRPT 023014	PW.D22= 000020
L#LAST 114404 G	L10046 032326	MEMCK 022236 G	PATCH 114170 G	PW.NOP= 000000
L#LOAD 002100 G	L10047 032542	MENASC 020567	PATDAT 020350	PW.NO1= 000023
L#LUN 002074 G	L10050 033110	MENERR 020514	PC.ERA= 002400	PW.RDE= 000024
L#PREV 002050 G	L10051 033454	MENRES 020616	PC.IER= 002000	PW.RDR= 000001
L#NAME 002000 G	L10052 047654	MVEEC = 000250	PC.NO0= 001000	PW.RDS= 000005
L#PRIO 002042 G	L10053 036062	MSA.FR= 000006	PC.REL= 000000	PW.RFI= 000003
L#PROT 022506 G	L10054 036642	MSA.NO= 000000	PC.REW= 000400	PW.WCT= 000006
L#PRT 002112 G	L10055 037416	MSA.NR= 000004	PKBCNT= 000006	PW.WFI= 000004
L#REPP 002062 G	L10056 040120	MSA.VO= 000002	PKHI = 000004	PW.WFM= 000007
L#REV 002010 G	L10057 040564	MSGEXP 012264 G	PKLOW = 000002	PW.WMI= 000010
L#RPT 023624 G	L10060 041220	MSGLOO 013204 G	PKTADD 007646	PW.WNP= 000011
L#SOFT 114022 G	L10061 041654	MSGSTA 012470 G	PKTFRM 007610	PW.WTR= 000002
L#SPC 002056 G	L10062 042246	MSGSUB 014062 G	PKTGET 012204 G	P.ACK = 100000
L#SPCP 002020 G	L10063 042750	MS.ATT= 000006	PKTMES 012230 G	P.CMD = 000037
L#SPTP 002024 G	L10064 043214	MS.EXT= 000200	PKTRAM 004745 G	P.CONT= 000012
L#STA 002030 G	L10065 043466	MS.RSD= 000001	PKTSSR 012166 G	P.CVC = 040000
L#SW 002156 G	L10066 043752	MS.RSF= 000020	PNT = 001000 G	P.FMT = 000140
L#TEST 002114 G	L10067 044252	MS.RST= 000010	PRAMPK 014116	P.FORM= 000011
L#TIML 002014 G	L10070 044736	M8186 005554	PRASC 014643	P.GETS= 000017

Symbol table

P.IE = 000200	SPM4 = 114060	TSFCOD = 007330	T##AUT = 010033	T22WLK = 027722
P.INIT = 000013	SPM6 = 114110	TSREJ = 000006	T##CLE = 010034	T22WRT = 027220
P.MODE = 007400	SPM7 = 114140	TSSDEF = 006700	T##DU = 010032	T23A = 003136 G
P.OPP = 020000	SRO = 177572	TSSR = 000002 G	T##HAR = 010134	T23AM3 = 034300
P.POSI = 000010	SR1 = 177574	TSSRBI = 003502 G	T##HW = 010000	T23B = 003140 G
P.READ = 000001	SR2 = 177576	TSSRFO = 006507	T##INI = 010030	T23BA = 034665
P.SMB = 010000	SR3 = 172516	TSSRH = 000003 G	T##MSG = 010025	T23BFR = 033542
P.WRIT = 000005	SSR = 000200	TSSX = 004020	T##PRO = 010027	T23BF2 = 033662
P.WRTC = 000004	STATCO = 012532	TSTBLK = 002744 G	T##RPT = 010035	T23BS0 = 033662
P.WRTS = 000006	SVCGBL = 000000	TSTCNT = 002204 G	T##SOF = 010135	T23BS1 = 033663
QVP = 002174 G	SVCINS = 000000	TSTEND = 017040	T##SRV = 010026	T23CHK = 035222
RAMASC = 014276	SVCSUB = 000001	TSTFLA = 002306 G	T##SUB = 010133	T23CON = 033700
RANDAT = 002234 G	SVCTAG = 000000	TSTL00 = 016576 G	T##SW = 010001	T23DAT = 033530
RAMERR = 015630 G	SVCTST = 000001	TSTPTR = 002310 G	T##TES = 010130	T23DSW = 033540
RAMEXP = 015650 G	S#LSYM = 010000	TSTSET = 016630 G	T1 = 024406 G	T23EOT = 034024
RAMFOR = 010210	SO.IDB = 000010	TST21I = 025344	T2 = 025522 G	T23ET = 033737
RAMSIZ = 002274 G	SO.IFB = 000002	TST22I = 030007	T2.1 = 025552	T23L00 = 030252
RANTAD = 015636 G	SO.IFP = 000001	TST23I = 035026	T2.2 = 026144	T23OFL = 034346
RCVHIA = 002276 G	SO.ILD = 000020	TST24I = 047422	T2.3 = 026466	T23PAC = 033520
RCVLOA = 002300 G	SO.ION = 000040	TST25I = 056400	T21AM3 = 025223	T23PK2 = 033630
RDERR = 005206	SO.IRD = 000100	TST26I = 075767	T21BFR = 025024	T23PK3 = 033650
RECMMSG = 002460 G	SO.IRW = 000004	TST27I = 106023	T21BF2 = 025120	T23RES = 035042
RECV = 002226 G	SO.ISP = 000200	TST28I = 113461	T21BS0 = 025120	T23RNC = 034225
REGSAV = 020260	S1.ICE = 002000	TSV2 = 002000 G	T21BS1 = 025121	T23RSZ = 033660
RETERR = 005372	S1.IEO = 010000	TSV3 = 002166 G	T21DAT = 025010	T23RT2 = 035134
RETRY = 035266	S1.IFM = 001000	TSV4 = 022506 G	T21DLY = 025022	T23RT3 = 035176
REV = 002220 G	S1.IHE = 000400	TSV6 = 113666 G	T21DSW = 025020	T23RWN = 034156
REWIND = 011136 G	S1.IID = 004000	TSV7 = 024406 G	T21L00 = 024436	T23SSR = 033704
RMCHBE = 000167	S1.IIR = 020000	TTIBFR = 177562 G	T21OFL = 025323	T23SZ = 033656
RMCHEN = 000200	S1.I2R = 040000	TTICSR = 177560 G	T21PAC = 025000	T23S2 = 033664
RMMSGB = 000215	S1.PAR = 100000	TTIVEC = 000060 G	T21PK2 = 025110	T23S3 = 033666
RMMSGE = 000234	S2.ATI = 000010	T#ARGC = 000003	T21RES = 025366	T23TM = 034102
RMPKTB = 000201	S2.BTI = 000004	T#CODE = 001130	T21RT2 = 025456	T23TMP = 033670
RMPKTE = 000210	S2.DIM = 000200	T#ERRN = 001513	T21SSR = 025126	T23VCK = 034612
RMR = 010000	S2.ILW = 000100	T#EXCP = 000000	T21S2 = 025122	T23WB = 033652
RMPACK = 011230	S2.INR = 000020	T#FLAG = 000040	T21S3 = 025124	T23WD = 033674
SC = 100000	S2.OUT = 000040	T#GMAN = 000000	T22AM3 = 027325	T23WDC = 034510
SCE = 020000	S2.UND = 000003	T#HILI = 000776	T22BFR = 027112	T23WDD = 034421
SCHERR = 005300	TBLEND = 003054 G	T#LAST = 000001	T22BF2 = 027210	T23WDR = 033676
SCME = 005013	TCOASC = 006570	T#LOLI = 000000	T22BS0 = 027210	T23WRT = 033672
SDELAY = 010750	TCOCOD = 006770	T#LSYM = 010000	T22BS1 = 027211	T23WSS = 034737
SELASC = 020562	TEMP1 = 003110 G	T#LTND = 000010	T22DAT = 027100	T24AM3 = 046410
SELDAT = 000004	TEMP2 = 003112 G	T#NEST = 177777	T22FOR = 027224	T24BA = 046742
SEL2 = 000002	TERCLS = 000016	T#NSO = 000000	T22L00 = 025552	T24BFR = 045022
SETHAP = 017436	TESTND = 000010	T#NS1 = 000005	T22OFL = 027425	T24BF2 = 045140
SETU = 023046	TEXASC = 006527	T#NS2 = 000002	T22PAC = 027070	T24BOT = 046003
SFFMSG = 012222 G	TFCASC = 006631	T#PTNU = 000000	T22PK2 = 027200	T24BS0 = 045140
SFHERR = 003705	TIMEXP = 015672 G	T#SAVL = 177777	T22POS = 027222	T24BS1 = 045141
SFIERR = 003652	TIMSGO = 015720	T#SEGL = 177777	T22RD = 027216	T24CON = 045152
SFIMSG = 012154 G	TINERR = 012141	T#SUBN = 000003	T22RES = 030042	T24DAT = 045010
SFPTBL = 002156 G	TMPBFR = 002624 G	T#TAGL = 177777	T22RT2 = 030134	T24DLY = 045156
SIFLAG = 003146 G	TNAM = 017024	T#TAGN = 010136	T22RWJ = 027574	T24DSW = 045020
SIMSG = 012106	TRANST = 002156 G	T#TEMP = 000000	T22SSR = 027230	T24DTA = 046050
SKIPT = 003372	TSBA = 000000 G	T#TEST = 000010	T22S2 = 027212	T24EOT = 046136
SOFINI = 016114 G	TSBAH = 000001 G	T#TSTM = 177777	T22S3 = 027214	T24ILA = 045532
SPACE = 010560 G	TSDB = 000000 G	T#TSTS = 000001	T22TM = 027500	T24LON = 047102
SPM1 = 114030	TSDBH = 000001 G	T##AUT = 010031	T22VCK = 027647	T24L00 = 035456

Symbol table

T24LOP	047164	T25RWN	056205	T26WB	073212	T27WDF	103460	T3.6	033126
T24LOQ	045616	T25SSR	055234	T26WDC	075100	T27WDR	103060	T398FR	021466
T24LOR	045232	T25SZ	055206	T26WDD	075010	T27WNG	103074	T398S0	021460
T24NEF	045160	T25S2	055212	T26WDE	074203	T27WRF	105646	T398S1	021461
T24NKM	045371	T25S3	055214	T26WDF	074011	T27WSS	105002	T398S2	021462
T24OFL	046455	T25TM	055442	T26WNG	073246	T28BFR	111442	T39DLY	021444
T24PAC	045000	T25WB	055202	T26WSS	075301	T28BF2	111560	T39DSW	022170
T24PBP	047246	T25WDC	056327	T27AM3	104367	T28B0T	112417	T39DTA	022160
T24PK2	045110	T25WDE	055315	T27BA	104727	T28BS0	111560	T39EAI	022166
T24PK3	045130	T25WDR	055220	T27BFR	102732	T28BS1	111561	T39PAC	021450
T24RB	045132	T25WNG	055605	T27BF2	103050	T28CNT	111606	T39PK2	022150
T24RES	047470	T25WNH	055760	T27BOT	103741	T28CNU	111610	T39PK3	022200
T24RN	045146	T26AM3	074666	T27BS0	103050	T28CON	111602	T39PK4	022210
T24RNC	046335	T26BA	075226	T27BS1	103051	T28DAT	111430	T39SIZ	022216
T24RT2	047562	T26BFR	073102	T27CNT	103066	T28DLY	111612	T39TAD	021460
T24RT3	047624	T26BF2	073220	T27CNU	103070	T28DSW	111440	T39WR	022212
T24RWN	046266	T26B0T	074255	T27CON	103062	T28DTA	113364	T4	035412 G
T24SSR	045677	T26BS0	073220	T27DAT	102720	T28DTR	113302	T4.1	035456
T24SZ	045136	T26BS1	073221	T27DLY	103072	T28IMV	111566	T4.10	042766
T24S2	045142	T26CNT	073236	T27DSW	102730	T28L00	106260	T4.11	043232
T24S3	045144	T26CNU	073240	T27DTA	105726	T28L0Q	112174	T4.12	043504
T24TM	046213	T26DAT	073070	T27EOT	104111	T28OFL	112610	T4.13	043770
T24TRL	047334	T26DLY	073244	T27LON	105071	T28PAC	111420	T4.14	044270
T24VCK	046667	T26DSW	073100	T27L00	076226	T28PBP	111671	T4.2	036100
T24WB	045132	T26DTA	074322	T27L0P	105153	T28PK2	111530	T4.3	036660
T24WDC	046616	T26EOT	074410	T27L0Q	103535	T28PK3	111550	T4.4	037434
T24WDD	046530	T2FLON	075370	T27LOR	103410	T28RB	111552	T4.5	040136
T24WDE	045731	T26L00	056650	T27NEF	105411	T28RDF	111754	T4.6	040602
T24WDF	045455	T26L0P	075452	T27OFL	104436	T28RDG	112035	T4.7	041236
T24WDG	045302	T26L0Q	074066	T27PAC	102710	T28RES	113506	T4.8	041672
T24WDR	045150	T26LOR	073741	T27PBP	105235	T28RIB	111614	T4.9	042264
T24WSS	047013	T26NEF	073334	T27PK2	103020	T28RN	111576	T5	047656 G
T25BFR	055072	T26NEQ	075710	T27PK3	103040	T28RRM	113067	T5.1	047706
T25BF2	055210	T26OFL	074735	T27RB	103042	T28RRN	113145	T5.2	050662
T25BNC	055670	T26PAC	073060	T27RDF	103162	T28RRP	113224	T5.3	051262
T25B0T	055375	T26PBP	075534	T27RES	106044	T28RT2	113600	T5.4	051736
T25BS0	055210	T26PK2	073170	T27RN	103056	T28RT3	113642	T5.5	052402
T25BS1	055211	T26PK3	073210	T27RNC	104314	T28RWN	112541	T5.6	053142
T25CNT	055230	T26RB	073212	T27RRF	103231	T28SSR	112255	T5.7	054102
T25CN2	055226	T26RDF	073416	T27RT2	106136	T28SZ	111556	T5.8	054422
T25CON	055222	T26RES	076000	T27RT3	106200	T28S2	111562	T6	056604 G
T25DAT	055060	T26RN	073226	T27RWN	104245	T28S3	111564	T6.1	056650
T25DLY	055232	T26RNC	074613	T27SC	103326	T28TM	112464	T6.10	066270
T25DSW	055070	T26RRF	073465	T27SCF	105507	T28THK	113015	T6.11	067144
T25L00	047706	T26RRG	073562	T27SSR	103616	T28VCK	112742	T6.12	070014
T25NEF	056043	T26RSZ	073242	T27SZ	103046	T28WB	111552	T6.13	070746
T25NET	055531	T26RT2	076072	T27S2	103052	T28WDC	112663	T6.14	071776
T25OFL	056254	T26RT3	076134	T27S3	103054	T28WDE	112326	T6.15	072356
T25PAC	055050	T26RWN	074544	T27TIM	104034	T28WDF	112117	T6.2	057562
T25PK2	055160	T26SC	073657	T27TM	104170	T28WDR	111600	T6.3	060430
T25PK3	055200	T26SSR	074147	T27TRL	105323	T3	030202 G	T6.4	061322
T25RB	055202	T26SZ	073216	T27TSA	105564	T3BFLG	003142 G	T6.5	062250
T25RES	056416	T26S2	073222	T27VCK	104654	T3.1	030252	T6.6	063026
T25RIB	056123	T26S3	073224	T27WB	103042	T3.2	030624	T6.7	063670
T25RN	055216	T26TM	074467	T27WDC	104601	T3.3	031510	T6.8	064542
T25RT2	056510	T26TRL	075622	T27WDD	104511	T3.4	032344	T6.9	065414
T25RT3	056552	T26VCK	075153	T27WDE	103652	T3.5	032560	T7	076166 G

G10

Symbol table

T7.1	076226	WC.IOT=	000100	XORFOR	010104	XSOTMK=	100000	X2.RCE=	040000
T7.2	076624	WC.IIT=	000040	XST0	= 000006	XSOVCK=	000020	X2.REV=	000077
T7.3	077406	WC.ISR=	000020	XST1	= 000010	XSOWLE=	004000	X2.SPA=	035400
T7.4	100230	WF.IED=	000010	XST2	= 000012	XSOWLK=	000004	X2.UNI=	000007
T7.5	101132	WF.IER=	000004	XST3	= 000014	XXCOMM	003114	X2.WCF=	002000
T8	106224	WF.IHI=	000200	XST4	= 000016	X\$ALWA=	000000	X3.DCK=	000010
T8.1	106260	WF.IRE=	000040	XSOBOT=	000002	X\$FALS=	000040	X3.MBZ=	000006
T8.2	106640	WF.IWF=	000020	XSOEOT=	000001	X\$OFFS=	000400	X3.MDE=	177400
T8.3	107120	WF.IWR=	000100	XSOIE	= 000040	X\$TRUE=	000020	X3.OPI=	000100
UAM	= 000200	WF.I3R=	000002	XSOILA=	000400	X1.COR=	020000	X3.REV=	000040
UNITN	002172	WF.I4R=	000001	XSOILC=	001000	X1.DLT=	100000	X3.RIB=	000001
UNREC	= 000006	WRTCHR	010752	XSOLET=	020000	X1.MBZ=	017375	X3.SPA=	000200
USI	004123	WRTERR	005113	XSOMOT=	000200	X1.RBP=	000400	X3.TRF=	000020
WAITF	016370	WRTMSG	005056	XSONEF=	002000	X1.SPA=	040000	X4.HSP=	100000
WC.IFA=	000200	WSMBK	022230	XSOONL=	000100	X1.UNC=	000002	X4.MBZ=	017400
WC.IFE=	000002	XFERAS	016060	XSOPED=	000010	X2.BUF=	000100	X4.RCE=	040000
WC.IGO=	000001	XNXM	016516	XSORLL=	010000	X2.EXT=	000200	X4.TSM=	020000
WC.IRE=	000010	XORBFO	007766	XSORLS=	040000	X2.OPM=	100000	X4.WRC=	000377
WC.IRW=	000004								

. ABS.	114404	000	(RW,I,GBL,ABS,OVR)
	000000	001	(RW,I,LCL,REL,CON)
ABS	000000	002	(RW,I,LCL,REL,CON)

Errors detected: 0

*** Assembler statistics

Work file reads: 306
Work file writes: 295
Size of work file: 28912 Words (113 Pages)
Size of core pool: 19684 Words (75 Pages)
Operating system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:10:06.95
CVTSCD,CVTSCD/-SP=SVC/ML,CVTSCD