

11/21+
TSV05

TSV05 CTRL LT3
CNTSCAO

COPYRIGHT (c) 1982-84
AH-T819A-MC
FICHE 01 OF 02

JUL 1984
digital
Made In USA

Table with multiple columns and rows of data, including headers like 'TSV05', 'CTRL', and 'CNTSCAO'. The content is highly repetitive and difficult to read due to low contrast and small font.

11/21+
TSV05

TSV05 CTRL LT3
CNTSCAO

COPYRIGHT (c) 1982-84
AH-T819A-MC
FICHE 02 OF 02

JUL 1984
digital
Made In USA

The image shows a microfiche card with a grid of 12 columns and 24 rows of tiny, illegible data frames. The frames appear to be small tables or lists of text. A barcode is visible in the top right corner of the grid area.

.REM

IDENTIFICATION

PRODUCT ID: AC T818A MC
PRODUCT TITLE: CNTSCAO TSV05 CTRL L13
DECO/DEPO: 1.0
DEPARTMENT: ISS/DIAGNOSTIC SERVICES
DATE: APRIL 09, 1984

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

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

COPYRIGHT (C) 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

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

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS IS A SBC 11/21+ RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSV05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A SBC 11/21+ SYSTEM (Q BUS). THE PROGRAM PROVIDES ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS THAT AID IN THE REPAIR OF THE DEVICE. THIS DIAGNOSTIC CONSIST OF EIGHT TEST WHICH ARE EXECUTED IN SEQUENCE.

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

1.2 SYSTEM REQUIREMENTS

SBC 11/21+ PROCESSOR AND MEMORY
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE I.E. 4K FOR I/O PAGE)
TSV05 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)
CONSOLE TERMINAL
PDP-11 DIAGNOSTIC SUPERVISOR (MSAAA.SYS VERSION 34 OR LATER)
PDP 11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

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

1.4 DIAGNOSTIC HIERARCY PPEREQUISITES

FUNCTIONAL SBC-11/21+ CENTRAL PROCESSOR AND MEMORY
FUNCTIONAL CONSOLE TERMINAL
FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR
FUNCTIONAL DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.5 ASSUMPTIONS

ALL HARDWARE EXCEPT THE HARDWARE UNDER TEST IS ASSUMED TO WORK

PROPERLY OR FALSE ERRORS CAN BE REPORTED.
 THE TAPE BEING USED ON THE TSV05 TRANSPORT IS A KNOWN GOOD REEL
 OF TAPE.
 CNTSAA AND CNTSBA 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.

2.1 COMMANDS

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

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

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

2.1.1 OPERATOR COMMANDS

THE TSV05 DIAGNOSTIC IS A SBC-11/21+ DIAGNOSTIC SUPERVISOR COMPATIBLE
 PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE
 XXDP+ USERS MANUAL. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC MEDIA

```
.R NTSC??
DIAG. RUN-TIME SERVICES REV D. APR 79
CNTSC-A-0
****TSV05 LOGIC DIAGNOSTIC****
UNIT IS TSV05
>DR
```

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH. IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY '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
MOE	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 = 176000, VECTOR = 224
```


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

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

UNIT 0

DEVICE ADDRESS (O) 176000 ? <ENTER THE ADDRESS OF THE
TSBA/TSDB REGISTER>

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

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE '# UNITS?' QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING AS FOLLOWS:
UP TO 4 TSV05 CONTROLLERS PER 11/21, AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES IN A TABLE THAT DESCRIBES THE DEVICES UNDER TEST. THE SIMPLEST WAY TO BUILD THIS TABLE IS TO ANSWER ALL QUESTIONS FOR EACH UNIT TO BE TESTED. IF YOU HAVE A MULTIPLEXED DEVICE SUCH AS

A MASS STORAGE CONTROLLER WITH SEVERAL DRIVES OR A COMMUNICATION DEVICE WITH SEVERAL LINES, THIS BECOMES TEDIOUS SINCE MOST OF THE ANSWERS ARE REPETITIOUS.

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

UNITS (D) ? 8<CR>

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

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

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

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

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

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

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

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

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

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

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

```
# UNITS (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
 CNTSC HRD ERR 01610 ON UNIT 00 TST 016 SUB 002 PC: 040624
 FIFO STATUS (IN WORD 9) INCORRECT AFTER WRITE FIFO

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

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

MESSAGE BUFFER ADDRESS = 047352

MESSAGE BUFFER CONTENTS:

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

ERROR MESSAGE EXAMPLE 2

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

CNTSC 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.

CNTSC 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 (SBC 11/21*)

DR>STA/FLA:PNT:HCE

UNITS (D) ? 1

UNIT 0

DEVICE ADDRESS (0) 176000 ? <CR>

VECTOR (0) 224 ? <CR>

CHANGE SW (L) ? N<CR>

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

TST: 001 INITIALIZE #4 TEST
TST: 002 OFF-LINE REJECT AND REWIND TEST
TST: 003 BASIC WRITE DATA TEST
TST: 004 BASIC READ DATA TEST
TST: 005 SPACE RECORDS TEST
TST: 006 REREADS TEST
TST: 007 WRITE DATA RETRY TEST
TST: 008 WRITE TAPE MARK TEST

0 ERRORS

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

PROGRAM RUN TIMES

THE AVERAGE RUN TIMES OF THE PROGRAM ARE LISTED BELOW. THESE FIGURES ARE TO BE USED AS A GUIDE. THE TIMING WAS DONE ON A FALCON PROCESSOR.

THE PROGRAM RUNS IN TWO MODES; NO ITERATIONS AND DEFAULT MODE. IN THE NO ITERATIONS MODE, EACH TEST IS RUN ONCE, WITH NO ITERATIONS. IN THE DEFAULT MODE EACH TEST IS REPEATED BY THE NUMBER OF TIMES INDICATED BY

THE ITERATION COUNT. NO ITERATIONS MODE IS SELECTED BY ANSWERING THE INHIBIT ITERATIONS QUESTION WITH A 'Y' (YES).

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

Q.V. 7 MINUTES
DEFAULT 34 MINUTES

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

5.0 DEVICE INFORMATION TABLES

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

CHANGE HW (L) ?

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

UNIT 0

DEVICE ADDRESS (O) 176000 ? <ENTER THE ADDRESS OF THE
TSBA/TSDB REGISTER>

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

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

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

CHANGE SW (L) ?

INHIBIT ITERATIONS (L) N ?

6.0 TEST SUMMARIES

TEST 1: INITIALIZE #4 TEST

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

TEST 2: OFF-LINE AND REJECT REWIND

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

TEST 3: BASIC WRITE DATA

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

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

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

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

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

TEST 5: SPACE RECORDS

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

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

TEST 6: REREADS

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

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

TEST 7: WRITE DATA RETRY

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

TEST 8: WRITE/READ TAPE MARK

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

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

CVTSCAO => CNTSCAO

JAKI BERG

9 APR-1984

CHANGES WERE MADE TO CVTSCAO TO PRODUCE CNTSCAO FOR THE

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

```

2          .TITLE  TSV2  PROGRAM HEADER
3          .SBTTL  PROGRAM HEADER
4
10         .MCALL  SVC
11 000000  SVC          ; INITIALIZE SUPERVISOR MACROS
12         .ENABLE LC
13         .NLIST  BEX,CND
19 000000  .ENABL  ABS,AMA
20         .*2000
21 002000  BGNMOD  TSV2
    002000
22
23         ;**
24         ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
25         ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
26         ;
27
28
29 002000  POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
30 002000  HEADER  CNTSC,A,0,655.,0
    002000  L$NAME:: ;DIAGNOSTIC NAME
    002000     103   .ASCII /C/
    002001     116   .ASCII /N/
    002002     124   .ASCII /T/
    002003     123   .ASCII /S/
    002004     103   .ASCII /C/
    002005     000   .BYTE  0
    002006     000   .BYTE  0
    002007     000   .BYTE  0
    002010  L$REV:: ;REVISION LEVEL
    002010     101   .ASCII /A/
    002011  L$DEPO:: ;0
    002011     060   .ASCII /0/
    002012  L$UNIT:: ;NUMBER OF UNITS
    002012 000000  .WORD  0
    002014  L$TIML:: ;LONGEST TEST TIME
    002014 001217  .WORD  655.
    002016  L$HPCP:: ;PTR. TO H.W. QUES.
    002016 112370  .WORD  L$HARD
    002020  L$SPCP:: ;PTR. TO S.W. QUES.
    002020 112522  .WORD  L$SOFT
    002022  L$HPTP:: ;PTR. TO DEF. H.W. PTABLE
    002022 002146  .WORD  L$HW
    002024  L$SPTP:: ;PTR. TO S.W. PTABLE
    002024 002156  .WORD  L$SW
    002026  L$LADP:: ;DIAG. END ADDRESS
    002026 113004  .WORD  L$LAST
    002030  L$STA:: ;RESERVED FOR APT STATS
    002030 000000  .WORD  0
    002032  L$CO::
    002032 000000  .WORD  0
    002034  L$DTYP:: ;DIAGNOSTIC TYPE
    002034 000000  .WORD  0
    002036  L$APT:: ;APT EXPANSION
    002036 000000  .WORD  0
    002040  L$DTP:: ;PTR. TO DISPATCH TABLE
    002040 002124  .WORD  L$DISPATCH

```

PROGRAM HEADER

002042		L\$PRIO::		;DIAGNOSTIC RUN PRIORITY
002042	000000		.WORD 0	
002044		L\$ENVI::		;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000		.WORD 0	
002046		L\$EXP1::		;EXPANSION WORD
002046	000000		.WORD 0	
002050		L\$MREV::		;SVC REV AND EDIT #
002050	003		.BYTE C\$REVISION	
002051	003		.BYTE C\$EDIT	
002052		L\$EF::		;DIAG. EVENT FLAGS
002052	000000		.WORD 0	
002054	000000		.WORD 0	
002056		L\$SPC::		
002056	000000		.WORD 0	
002060		L\$DEVP::		; POINTER TO DEVICE TYPE LIST
002060	003372		.WORD L\$DVTYP	
002062		L\$REPP::		;PTR. TO REPORT CODE
002062	023002		.WORD L\$RPT	
002064		L\$EXP4::		
002064	000000		.WORD 0	
002066		L\$EXP5::		
002066	000000		.WORD 0	
002070		L\$AUT::		;PTR. TO ADD UNIT CODE
002070	022470		.WORD L\$AU	
002072		L\$DUT::		;PTR. TO DROP UNIT CODE
002072	022566		.WORD L\$DU	
002074		L\$LUN::		;LUN FOR EXERCISERS TO FILL
002074	000000		.WORD 0	
002076		L\$DESP::		;POINTER TO DIAG. DESCRIPTION
002076	003400		.WORD L\$DESC	
002100		L\$LOAD::		;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		EMT E\$LOAD	
002102		L\$ETP::		;POINTER TO ERR TBL
002102	000000		.WORD 0	
002104		L\$ICP::		;PTR. TO INIT CODE
002104	021646		.WORD L\$INIT	
002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
002106	022754		.WORD L\$CLEAN	
002110		L\$ACP::		;PTR. TO AUTO CODE
002110	022674		.WORD L\$AUTO	
002112		L\$PRT::		;PTR. TO PROTECT TABLE
002112	021636		.WORD L\$PROT	
002114		L\$TEST::		;TEST NUMBER
002114	000000		.WORD 0	
002116		L\$DLY::		;DELAY COUNT
002116	000000		.WORD 0	
002120		L\$HIME::		;PTR. TO HIGH MEM
002120	000000		.WORD 0	

DISPATCH TABLE

```

32
33
34
35
36
37
38
39 002122
    002122 000010
    002124
    002124 023564
    002126 024702
    002130 027362
    002132 034352
    002134 046466
    002136 055404
    002140 074726
    002142 104744

```

.SBITL 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

```

40

1-1

DEFAULT HARDWARE P TABLE

```

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

```

SOFTWARE P TABLE

```

56                                     .SBTTL SOFTWARE P TABLE
57
58                                     ;**
59                                     ; THE SOFTWARE P TABLE CONTAINS THE VALUES OF THE PROGRAM
60                                     ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
61                                     ;
62 002154                               BGNSW   SFPTBL
   002154 000004                       .WORD  L10001 L$SW/2
   002156
   002156
63
64 002156 000000                       TRANSTST:: .WORD 0 ; ENABLE TEST OF TRANSPORT(S) IF =1
65 002160 000000                       NOITS:: .WORD 0 ; INHIBIT ITERATION OPTION.
66                                     ; ... 0 = ITERATE.
67                                     ; ...NZ = INHIBIT ITERATE.
68 002162 000017                       LERRMAX:: .WORD 15. ; LOCAL (PER TEST) ERROR LIMIT
69 002164 00C310                       GERRMAX:: .WORD 200. ; GLOBAL (PER UNIT) ERROR LIMIT
70 002166                               ENDSW
   002166                               L10001:
71
72 002166                               ENDMOD

```

SOFTWARE P-TABLE

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

002166
002166

002166

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

000040
000037
000036
000035
000034

```
.TITLE TSV3 - GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

BGNMOD TSV3

TSV3::

.SBTTL GLOBAL EQUATES SECTION

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

```
EQUALS ; GET STANDARD EQUATES.
```

```
; BIT DEFINITIONS
```

```
BIT15== 100000
BIT14== 40000
BIT13== 20000
BIT12== 10000
BIT11== 4000
BIT10== 2000
BIT09== 1000
BIT08== 400
BIT07== 200
BIT06== 100
BIT05== 40
BIT04== 20
BIT03== 10
BIT02== 4
BIT01== 2
BIT00== 1
```

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

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

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


Kc'

GLOBAL EQUATES SECTION

```

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

```

```

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

```

33
34 002166

```

KT11 .. ;DEFINE MEMORY MANAGEMENT REGISTERS
.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

```

```

000250
177572
177574
177576
172516

```

MEMORY MANAGEMENT DEFINITIONS

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

MEMORY MANAGEMENT DEFINITIONS

```
SDPAR3= 172266
SDPAR4= 172270
SDPAR5= 172272
SDPAR6= 172274
SDPAR7= 172276
.ENDC
.ENDC
;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
172300 KIPDR0= 172300
172302 KIPDR1= 172302
172304 KIPDR2= 172304
172306 KIPDR3= 172306
172310 KIPDR4= 172310
172312 KIPDR5= 172312
172314 KIPDR6= 172314
172316 KIPDR7= 172316
.IF NB
;*KERNEL "D" PAGE DESCRIPTOR REGISTERS
KDPDR0= 172320
KDPDR1= 172322
KDPDR2= 172324
KDPDR3= 172326
KDPDR4= 172330
KDPDR5= 172332
KDPDR6= 172334
KDPDR7= 172336
.ENDC
;*KERNEL "I" PAGE ADDRESS REGISTERS
172340 KIPAR0= 172340
172342 KIPAR1= 172342
172344 KIPAR2= 172344
172346 KIPAR3= 172346
172350 KIPAR4= 172350
172352 KIPAR5= 172352
172354 KIPAR6= 172354
172356 KIPAR7= 172356
.IF NB
;*KERNEL "D" PAGE ADDRESS REGISTERS
KDPAR0= 172360
KDPAR1= 172362
KDPAR2= 172364
KDPAR3= 172366
KDPAR4= 172370
KDPAR5= 172372
KDPAR6= 172374
KDPAR7= 172376
.ENDC
```

TSV05 REGISTER AND PACKET DEFINITIONS

```

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

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

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

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

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

```

05

TSV05 REGISTER AND PACKET DEFINITIONS

```

210 ;
211 ;
212 ;
213 000006 XST0== 6 ;EXTENDED STATUS REGISTER 0 (WORD 4)
214 000010 XST1== 8. ;EXTENDED STATUS REGISTER 1 (WORD 5)
215 000012 XST2== 10. ;EXTENDED STATUS REGISTER 2 (WORD 6)
216 000014 XST3== 12. ;EXTENDED STATUS REGISTER 3 (WORD 7)
217 000016 XST4== 14. ;EXTENDED STATUS REGISTER 4 (WORD 8)
218 ;
219 ;*
220 ;
221 ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
222 ;
223 ;
224 ;
225 000002 PKLOW = 2 ;LOW ORDER CHARACTERISTIC DATA POINTER
226 000004 PKHI = 4 ;HIGH ORDER CHARACTERISTIC DATA POINTER
227 000006 PKBCNT = 6 ;NUMBER OF BYTES IN DATA PACKET
228 ;
229 000010 EXBCNT=10 ;NUMBER OF BYTES IN EXTENDED DATA PACKET
230 ;
231 ;*
232 ;DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
233 ;-
234 000000 BSEL0 = 0 ;BYTE 0
235 000001 BSEL1 = 1 ;BYTE 1
236 000002 SEL2 = 2 ;WORD 2
237 000004 SELDATA = 4 ;WORD 3
238 ;
239 ;*
240 ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
241 ;-
242 000000 PW.NOP = 0 ;NO-OP
243 000001 PW.RDRAM = 1 ;READ RAM
244 000002 PW.WTRAM = 2 ;WRITE RAM
245 000003 PW.RFIFO = 3 ;READ FIFO
246 000004 PW.WFIFO = 4 ;WRITE FIFO
247 000005 PW.RDSTAT = 5 ;READ STATUS
248 000006 PW.WCTL = 6 ;WRITE TAPE CONTROL
249 000007 PW.WFMT = 7 ;WRITE TAPE FORMAT
250 000010 PW.WMISC = 10 ;WRITE MISCELLANEOUS
251 000011 PW.WNPR = 11 ;WRITE NPR CONTROL
252 000020 PW.D22 = 20 ;DO MICROTEST 22
253 000021 PW.D11 = 21 ;DO MICROTEST 11
254 000022 PW.D13 = 22 ;DO MICROTEST 13
255 000023 PW.NO1311 = 23 ;DISABLE MICROTEST 11 AND 13
256 000024 PW.RDEXT = 24 ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
257 ;
258 ;*
259 ;BSEL1 CODES FOR WRITE TAPE CONTROL
260 ;-
261 000200 WC.IFAD = BIT7 ;IFAD - FORMATTER ADDRESS
262 000100 WC.IOTAD = BIT6 ;ITADO - TRANSPORT ADDRESS BIT 0
263 000040 WC.I1TAD = BIT5 ;ITAD1 - TRANSPORT ADDRESS BIT 1
264 000020 WC.ISRESV = BIT4 ;IRESVS - RESERVED #5
265 000010 WC.IREW = BIT3 ;IREW - REWIND
266 000004 WC.IRWU = BIT2 ;IRWU - REWIND AND UNLOAD

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

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

```


TSV05 REGISTER AND PACKET DEFINITIONS

```

324      000100      SO.IRDY      = BIT6      ;      IRDY L
325      000040      SO.IONL      = BIT5      ;      IONL L
326      000020      SO.ILDP      = BIT4      ;      ILDP L
327      000010      SO.IDBY      = BIT3      ;      IDBY L
328      000004      SO.IRWD      = BIT2      ;      IRWD L
329      000002      SO.IFBY      = BIT1      ;      IFBY L
330      000001      SO.IFPT      = BIT0      ;      IFPT L
331
332      .SBTTL      SPECIAL MACROS AND OPDEFS.
333
334      ;*
335      ;SAVE GENERAL REGS 1 TO 5
336      ;-
337      .MACRO      SAVREG
338      JSR      R5,REGSAV
339      .ENDM
340
341      ;*
342      ; MACRO TO FORCE AN ERROR
343      ;
344      .MACRO      FORCERROR      TAG,NOTSSR
345      .NLIST
346      .IIF NDF LISTALL, .NLIST
347      .LIST
348      .IF B NOTSSR
349      MOV      TSSR(R5),R1      ;READ TSSR
350      .ENDC
351      MOV      FORCER,FORCER      ;IS FORCER SET? (LEAVE C BIT ALONE)
352      BNE      TAG      ;BR IF YES
353      .NLIST
354      .IIF NDF LISTALL, .LIST
355      .LIST
356      .ENDM
357
358      ;*
359      ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
360      ; WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
361      ; SO TO FORCE ERRORS AND EXIT ON 1 ERRGR SET
362      ; FORCER TO 177777
363      ; TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
364      ;-
365      .MACRO      FORCEEXIT      TAG
366      .NLIST
367      .IIF NDF LISTALL, .NLIST
368      .LIST
369      MOV      FORCER,FORCER      ;IS FORCER NEGATIVE?
370      BMI      TAG      ;BR IF YES
371      .NLIST
372      .IIF NDF LISTALL, .LIST
373      .LIST
374      .ENDM
375      ;*
376      ; MACRO TO INCREMENT ERROR COUNTS
377      ;-
378      .MACRO      NEXT.ERRNO
379      .NLIST
380      ;;;.IIF NDF LISTALL, .NLIST

```

G3

SPECIAL MACROS AND OPDEFS.

```

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

```

GLOBAL DATA SECTION

```

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

```

TSTBLK TEST DATA TABLE

```

495 003036 167777 .WORD ↑CBIT12
496 003040 157777 .WORD ↑CBIT13
497 003042 137777 .WORD ↑CBIT14
498 003044 077777 .WORD ↑CBIT15
499 003046 125252 .WORD 125252 ;ALTERNATING ONES, ZEROS
500 003050 052525 .WORD 052525 ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
501 003052 003052
502
503
504 ;
505 ;STORAGE FOR DEVICE REGISTERS
506 003052 000000 100000 000000 DUMMY: 0,100000,0,0 ;DUMMY DEVICE REGISTERS...
507 003062 000000 000000 000000 0,0,0,0,0,0,0,0,0
508 ;...FOR MULTI-UNIT CHECKOUT.
509
510 003102 000000 DUFLG: .WORD 0 ;"DROPPED UNIT" FLAG.
511 ;INHIBITS CODE IN "CLEAN UP".
512 003104 000000 NCDEV: .WORD 0 ;FLAG TO SAY NO DEVICE.
513
514 003106 000000 TEMP1: .WORD 0 ;SOME TEMP LOCATIONS.
515 003110 000000 TEMP2: .WORD 0
516 003112 000000 XXCOMM: .WORD 0 ;XXDP+ COMM BLOCK POINTER.
517 003114 000000 FREE: .WORD 0 ;1ST FREE MEMORY ADDRESS...
518 003116 000000 FRESIZ: .WORD 0 ;...AND SIZE (IN WORDS).
519 003120 000000 FREEHI: .WORD 0 ;LAST WORD IN FREE SPACE
520 003122 000000 KTFLG: .WORD 0 ;KT11, MEM AVAIL FLAG -
521 ;- .WORD 0 = <24K OR NO KT -
522 ;- NZ = >24K AND KT.
523 003124 000000 KTENABLE: .WORD 0 ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
524 003126 000000 NXMFLG: .WORD 0 ;SET IF WE CAN TEST CLEARED OTHERWISE
525 003130 000000 NXMLO: .WORD 0 ;NXM LO ADDRESS BITS
526 003132 000000 NXMHI: .WORD 0 ;NXM HI ADDRESS BITS FOR DAL'S 16-21
527 003134 000000 T23A: .WORD 0 ;11/23A FLAG
528 003136 000000 T23B: .WORD 0 ;11/23B FLAG
529 003140 000000 T3BFLG: .WORD 0 ;TEST 3B FLAG ↑0
530 003142 002000 PST32W: .WORD 2000 ;32W BLOCK ADDRESS FOR 32K START
531 003144 000000 SIFLAG: .WORD 0
532 003146 000000 BADDAT: .WORD 0 ;ACTUAL DATA
533 003150 000000 GDDAT: .WORD 0 ;EXPECTED DATA
534 003152 000000 LOOPFL: .WORD 0
535 003154 CTAB: ;CONFIGURATION TABLES.
536 003154 000000 CTABM: .WORD 0 ;CONFIG WORK.
537 003156 000000 .WORD 0
538 003160 000000 .WORD 0
539 003162 000000 .WORD 0
540 003164 177777 .WORD -1 ;END OF MEM TABLE.
541 003166
542 CTABE: ;
543 ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
544 ;
545 ; 0 = UNIT NOT TESTED
546 ; 100000 = UNIT ONLINE, NO ERRORS
547 ; 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
548 ; 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
549 ; 160001 = UNIT DROPPED, NOT IDLE AT START
550 ; 14XXXX = UNIT DROPPED, ENCOUNTERED XXXX ERRORS
551 003166
ERTABL: .BLKW 64.

```

J3

GLOBAL ENVIRONMENT STORAGE

552 003366 000000
553
554 003370 000000

ERTABE: .WORD 0

SKIPT: .WORD 0

:1=SKIP SUBTEST 0=NO SKIP OF SUBTEST

GLOBAL TEXT MESSAGES

```

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

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

594
595 ;*
596 ; BIT TO ASCII CONVERSION FOR TSSR REGISTER
597 ;*
598
599 003500 003540 003543 003547 TSSRBIT::          .WORD  1$,2$,3$,4$,5$,6$,7$,8$
600 003520 003601 003605 003611          .WORD  9$,10$,11$,12$,13$,14$,15$,16$
601 003540          123    103    000    1$:          .ASCIZ  'SC'
602 003543          102    111    105    2$:          .ASCIZ  'BIE'
603 003547          123    103    105    3$:          .ASCIZ  'SCE'
604 003553          122    115    122    4$:          .ASCIZ  'RMR'
605 003557          116    130    115    5$:          .ASCIZ  'NXM'
606 003563          116    102    101    6$:          .ASCIZ  'NBA'
607 003567          102    111    124    7$:          .ASCIZ  'BIT9'
608 003574          102    111    124    8$:          .ASCIZ  'BIT8'
609 003601          123    123    122    9$:          .ASCIZ  'SSR'
610 003605          117    106    114   10$:          .ASCIZ  'OFL'
611 003611          102    111    124   11$:          .ASCIZ  'BIT5'
612 003616          102    111    124   12$:          .ASCIZ  'BIT4'
613 003623          102    111    124   13$:          .ASCIZ  'BIT3'
614 003630          102    111    124   14$:          .ASCIZ  'BIT2'
615 003635          102    111    124   15$:          .ASCIZ  'BIT1'
616 003642          102    111    124   16$:          .ASCIZ  'BIT0'
617          .EVEN
618 003650          124    123    123   SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
619 003703          124    123    123   SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
620 003736          040    040    116   NXR:     .ASCIZ  / NON-EXISTANT DEVICE REGISTER/
621 003775          045    101    040   NXR:     .ASCIZ  /#A ADDRESS: #06/
622 004016          045    101    040   TSSX:    .ASCII  /#A TSBA,TSSR EXP'D: #06#A,#06#N/
623 004056          045    101    040   TSSX:    .ASCIZ  /#A TSBA,TSSR REC'D: #06#A,#06/
624 004115          045    116    045   FUSI:    .ASCII  /#N#A/
625 004121          040    040    125   USI:     .ASCIZ  / UNEXPECTED INTERRUPT/
626 004150          040    040    111   NSI:     .ASCIZ  / INTERRUPT EXPECTED, NOT RECEIVED/
627 004213          045    116    045   FN0INTR: .ASCII  /#N#A/

```

GLOBAL TEXT MESSAGES

```

628 004217      040      040      116 NOINTR: .ASCIZ / NO INTERRUPT WAS GENERATED/
629 004254      040      040      111 IFAULT: .ASCIZ / INTERRUPT FAULT/
630 004276      045      101      040 INTX: .ASCIZ /%A CPU PC: %06%A TSBA: %06/
631 004333      040      040      042 NOINIT: .ASCIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
632 004405      040      040      042 NSINIT: .ASCIZ / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
633 004455      040      040      042 BRINIT: .ASCIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
634
635 004525      000
636 004526      045      116      000 NULCR: .ASCIZ /%N/
637 004531      045      101      040 EXPGOT: .ASCIZ /%A EXP'D: %06%A, REC'D: %06/
638 004565      045      116      045 EXPGT2: .ASCIZ /%N%A EXP'D: %06%A, %06%N%A REC'D: %0%A, %06/
639 004641      045      101      040 DUAD12: .ASCIZ /%A REG(W) WRITTEN TO: %06%A REG(R) READ; EXP'D: %06%A, REC'D: %06/
640 004743      122      101      115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
641 005011      040      040      103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
642 005054      127      122      111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
643 005111      124      123      123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
644 005204      124      123      123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
645 005276      106      101      124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
646 005370      105      122      122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
647 005456      045      116      045 NOMEM: .ASCIZ '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****N'
648 005552      045      116      045 M8186: .ASCIZ '%N%A ***** 11/23A SYSTEM *****N'
649 005643      045      116      045 M8189: .ASCIZ '%N%A ***** 11/23B SYSTEM *****N'

```

.EVEN
.SBTTL GLOBAL ERROR REPORT SECTION

```

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

```

```

659 005734      BGNMSG  NXRERR      ;NON-EXISTANT DEVICE REGISTER.
005734
660 005734      PRINTX  %NXRX,NODEV ;NODEV = NEXM ADDRESS.
005734 013746 003104
005740 012746 003775
005744 012746 000002
005750 010600
005752 104415
005754 062706 000006
661 005760 004737 005766
662 005764
005764
005764 104423
663
664
665
666
667
668 005766 005727
669 005770 000000
670 005772 001402
671 005774 004777 177770
672 006000
006000 012746 004526
006004 012746 000001
006010 010600

```

```

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,RO

```

M3

GLOBAL ERROR REPORT SECTION

006012	104415	
006014	062705	000004
673 006020	000207	

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

PRITSSR PRINT TSSR CONTENTS

```

675 .SBITL PRITSSR - PRINT TSSR CONTENTS
676
677 ;*
678 ;
679 ;ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
680 ;THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
681 ;BY A MESSAGE PRINTING ROUTINE
682 ;
683 ;INPUTS:
684 ;
685 ; R1 CONTENTS OF TSSR
686 ;
687 ;SUBORDINATE ROUTINES:
688 ;
689 ; CHKAMB CHECK FOR AMBIGUOUS CONTENTS
690 ;
691 ;-
692
693 PRITSSR:
694 SAVREG ;SAVE GENERAL REGISTERS
695 MOV R1,R4 ;SAVE THE TSSR CONTENTS
696 PRINTB @TSSRFOR,R4 ;PRINT THE CONTENTS OF TSSR
006030 010446 MOV R4,-(SP)
006032 012746 006505 MOV @TSSRFOR,-(SP)
006036 012746 000002 MOV @2,-(SP)
006042 010600 MOV SP,R0
006044 104414 TRAP C$PNTB
006046 062706 000006 ADD @6,SP
697 006052 010400 MOV R4,R0 ;GET TSSR BACK FOR CHKAMB
698 006054 004737 016134 JSR PC,CHKAMB ;ARE CONTENTS AMBIGUOUS ?
699 006060 103410 BCS 5$ ;BRANCH IF NOT
700 006062 PRINTX @AMBTSSR ;SHOW CONTENTS ARE AMBIGUOUS
006062 012746 006725 MOV @AMBTSSR,-(SP)
006066 012746 000001 MOV @1,-(SP)
006072 010600 MOV SP,R0
006074 104415 TRAP C$PNTX
006076 062706 000004 ADD @4,SP
701 006102 010403 5$: MOV R4,R3 ;CONTENTS OF TSSR
702 006104 042703 001476 BIC @HIADDR!FATERR!TERCLS,R3 ;CLEAR ALL MULTIPLE BIT FIELDS
703 006110 001434 BEQ 20$ ;NO BITS ARE SET
704 006112 012702 002622 MOV @TMPBFR,R2 ;TEMPORARY ASCII BUFFER
705 006116 012701 003500 MOV @TSSRBIT,R1 ;ASCII EQUIVALENT OF BITS
706 006122 005703 10$: TST R3 ;REMAINING BITS TO CONVERT
707 006124 001413 BEQ 15$ ;BRANCH WHEN ALL ARE DONE
708 006126 000241 CLC ;CLEAR CARRY FOR SHIFT
709 006130 006103 ROL R3 ;SHIFT NEXT BIT TO CARRY
710 006132 103006 BCC 13$ ;BRANCH IF BIT NOT SET
711 006134 011100 MOV (R1),R0 ;POINTER TO BIT DEFINITION
712 006136 112022 11$: MOVB (R0)+,(R2)+ ;MOVE ASCII TO BUFFER
713 006140 001376 BNE 11$ ;MOVE ALL BITS
714 006142 112762 000054 177777 MOVB @',,-1(R2) ;INSERT A COMMA TO TERMINATE
715 006150 005721 13$: TST (R1)+ ;POINT TO NEXT DESCRIPTION
716 006152 000763 BR 10$ ;GET THE REMAINING BITS
717 006154 105042 15$: CLRB -(R2) ;TERMINATE THE LINE
718 006156 PRINTX @TSSDEF,@TMPBFR ;PRINT THE BIT DEFINITIONS
006156 012746 002622 MOV @TMPBFR,-(SP)
006162 012746 006676 MOV @TSSDEF,-(SP)

```

PRITSSR PRINT TSSR CONTENTS

```

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

```

PRITSSR PRINT TSSR CONTENTS

```

764 007006      116      157      162  1$: .ASCIZ 'Normal Termination'
765 007031      124      145      162  2$: .ASCIZ 'Termination Condition'
766 007057      124      141      160  3$: .ASCIZ 'Tape Status Alert'
767 007101      106      165      156  4$: .ASCIZ 'Function Reject'
768 007121      122      145      143  5$: .ASCIZ 'Recoverable Error - Tape Position One Record Down'
769 007203      122      145      143  6$: .ASCIZ 'Recoverable Error - Tape Was Not Moved'
770 007252      125      156      162  7$: .ASCIZ 'Unrecoverable Error'
771 007276      106      141      164  8$: .ASCIZ 'Fatal Controller Error'
772          .EVEN
773
774 007326 007336 007372 007403 TSFCOD: .WORD 1$,2$,3$,4$
775 007336      111      156      164  1$: .ASCIZ 'Internal Diagnostic Failure'
776 007372      122      145      163  2$: .ASCIZ 'Reserved'
777 007403      102      165      163  3$: .ASCIZ 'Bus Interface or Sanity Check Error'
778 007447      122      145      163  4$: .ASCIZ 'Reserved'
779          .EVEN
780          .SBTTL PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
781
782          ;;
783          ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
784          ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
785          ;
786          ;INPUT:
787          ;
788          ;      R0      NUMBER OF WORDS IN PACKET
789          ;      R3      HIGH ORDER COMMAND PACKET ADDRESS
790          ;      R4      ADDRESS OF COMMAND PACKET
791          ;
792          ;      NOTE:  R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
793          ;
794          ;-
795 007460          PRIPKT::
796 007460          SAVREG          ;SAVE THE REGISTERS
797 007464 010005          MOV      R0,R5          ;SAVE NO. OF WORDS IN PACKET
798 007466 005737 003124  TST      KTENABLE          ;ABOVE 28K UNDER TEST?
799 007472 001001          BNE     10$          ;BR IF YES
800 007474 005003          CLR     R3          ;SET HIGH ORDER ADDRESS TO 0
801 007476 010301          10$:  MOV     R3,R1          ;COPY HIGH ORDER ADDRESS
802 007500 010400          MOV     R4,R0          ;GET LOWER ADDRESS
803 007502 006100          ROL     R0          ;SHIFT BIT 15 INTO C BIT
804 007504 006101          ROL     R1          ;AND INTO HIGH ORDER.
805 007506          PRINTB  #PKTADD,R1,R4 ;PRINT PACKET ADDRESS
806          007506 010446          MOV     R4,(SP)
807          007510 010146          MOV     R1,-(SP)
808          007512 012746 007644  MOV     #PKTADD,(SP)
809          007516 012746 000003  MOV     #3,-(SP)
810          007522 010600          MOV     SP,R0
811          007524 104414          TRAP   C#PNTB
812          007526 062706 000010  ADD     #10,SP
813 007532 010300          15$:  MOV     R3,R0          ;GET HIGH ORDER ADDRESS
814 007534 001404          BEQ    20$          ;BR IF NOT ABOVE 28K.
815 007536 010401          MOV     R4,R1          ;GET LOW ORDER ADDRESS
816 007540 004737 017406  JSR    PC,SETMAP          ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
817 007544 010004          MOV     R0,R4          ;GET RETURNED PAR6 ADDRESS BIAS
818 007546 005001          20$:  CLR     R1          ;SAVE WORD NUMBER
819 007550 012402          25$:  MOV     (R4),R2          ;GET PACKET CONTENTS
820 007552          PRINTB  #PKTFRM,R1,R2 ;PRINT THE DATA

```

PRIPKT PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

```

007552 010246      MOV      R2, (SP)
007554 010146      MOV      R1, (SP)
007556 012746 007606  MOV      @PKTFRM, (SP)
007562 012746 000003  MOV      @3, (SP)
007566 010600      MOV      SP,R0
007570 104414      TRAP     C$PNTB
007572 062706 000010  ADD      @10,SP
814 007576 005201      INC      R1          ;NEXT WORD NUMBER
815 007600 020105      CMP      R1,R5      ;DONE ALL PACKET WORDS?
816 007602 002762      BLT     25$        ;LOOP TILL ALL DONE
817 007604 000207      RTS     PC          ;RETURN
818
819 007606      045      116      045  PKTFRM: .ASCIZ  'N$A Packet Word #D1$A = #06'
820 007644      045      116      045  PKTADD: .ASCIZ  'N$A Packet Address = #01#05'
821
822          .EVEN
823          .SBTTL  PRIBXOR - PRINT EXPD, RECV AND XOR BYTE
824
825          ;*
826          ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
827          ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
828
829          ;INPUTS:
830
831          ;      R1      RECEIVED DATA
832          ;      R2      EXPECTED DATA
833
834          ;OUTPUT:
835
836          ;      R0      XOR OF EXPECTED/RECEIVED DATA
837
838          ;-
839
840 007702      -
841 007702      SAVREG          ;SAVE THE REGISTERS
842 007706 010203      MOV      R2,R3      ;EXPECTED DATA
843 007710      XOR      R1,R3      ;FORM THE EXCLUSIVE OR
844 007720 012700 177400  MOV      @+C<377>,R0 ;BYTE MASK
845 007724 040001      BIC      R0,R1      ;SAVE LOW BYTE RECV
846 007726 040002      BIC      R0,R2      ;SAVE LOW BYTE EXPD
847 007730 040003      BIC      R0,R3      ;SAVE LOW BYTE XOR
848 007732      PRINTB @XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
      007732 010346      MOV      R3,-(SP)
      007734 010146      MOV      R1,-(SP)
      007736 010246      MOV      R2,-(SP)
      007740 012746 007764  MOV      @XORBFOR,-(SP)
      007744 012746 000004  MOV      @4,-(SP)
      007750 010600      MOV      SP,R0
      007752 104414      TRAP     C$PNTB
      007754 062706 000012  ADD      @12,SP
849 007760 010300      MOV      R3,R0      ;R0 HAS XOR ON RETURN
850 007762 000207      RTS     PC          ;RETURN TO CALLER
851
852 007764      045      116      045  XORBFOR: .ASCIZ  'N$A EXPD: #03$A RECV: #03$A XOR: #03'
853          .EVEN
854          .SBTTL  PRIXOR  PRINT EXPD, RECV AND XOR
855

```

PRIXOR PRINT EXPD. RECV AND XOR

```

856 ;*
857 ;
858 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
859 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
860 ;
861 ;INPUTS:
862 ;
863 ; R1 RECEIVED DATA
864 ; R2 EXPECTED DATA
865 ;
866 ;OUTPUT:
867 ;
868 ; R0 XOR OF EXPECTED/RECEIVED DATA
869 ;
870 ;-
871 ;
872 010032 PRIXOR::
873 010032 SAVREG ;SAVE THE REGISTERS
874 010036 010203 MOV R2,R3 ;EXPECTED DATA
875 010040 XOR R1,R3 ;FORM THE EXCLUSIVE OR
876 010050 PRINTB @XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
      010050 010346 MOV R3,-(SP)
      010052 010146 MOV R1,-(SP)
      010054 010246 MOV R2,-(SP)
      010056 012746 010102 MOV @XORFOR,-(SP)
      010062 012746 000C04 MOV @4,-(SP)
      010066 010600 MOV SP,R0
      010070 104414 TRAP C#PNTB
      010072 062706 000012 ADD @12,SP
677 010076 010300 MOV R3,R0 ;R0 HAS XOR ON RETURN
878 010100 000207 RTS PC ;RETURN TO CALLER
879 ;
880 010102 045 116 045 XORFOR: .ASCIZ '#N#A EXPD: #06#A RECV: #06#A XOR: #06#'
881 .EVEN
882 .SBTTL PRIEQU PRINT BIT NUMBEPS AS ASCII EQUIVALENT
883 ;
884 ;*
885 ;
886 ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
887 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
888 ;
889 ;INPUTS:
890 ;
891 ; R0 OCTAL VALUE TO CONVERT
892 ; R1 TABLE OF POINTERS TO ASCII EQUIVALENT
893 ;
894 ;
895 ;
896 010150 PRIEQU:
897 010150 SAVREG ;SAVE THE REGISTERS
898 010154 000207 RTS PC ;RETURN TO CALLER
899 ;
900 .SBTTL PRIRAM - PRINT RAM ADDRESS
901 ;
902 ;
903 ;PRINT CONTROLLER RAM ADDRESS.
904 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.

```

PRIRAM PRINT RAM ADDRESS

```

905
906 ; INPUTS:
907 ;
908 ; R4 RAM ADDRESS
909 ;
910 ;
911 PRIRAM:
912 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
913 PRINTB @RAMFOR,R4 ;PRINT RAM ADDRESS IN ERROR
    010162 010446
    010164 012746 010206
    010170 012746 000002
    010174 010600
    010176 104414
    010200 062706 000006
914 010204 000207
915
916 010206 045 116 045 RAMFOR: .ASCIZ 'MEMA CONTROLLER RAM ADDRESS = #06'
917 .EVEN
918
919 .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
920
921 ;
922 ;PRINT MEMORY ADDRESS
923 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
924 ;
925 ; IMPLICIT INPUTS
926 ;
927 ; ERRHI - HIGH ORDER ADDRESS
928 ; ERRLO - LOW ORDER ADDRESS
929 ;
930 ;-
931 PRIADD:
932 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
933 MOV ERRHI,R0 ;GET HIGH ADDRESS
934 MOV ERRLO,R1 ;GET LOW ADDRESS
935 MOV R1,R2 ;COPY LOW ADDRESS
936 ROL R1 ;SHIFT BIT 15 TO C BIT
937 ROL R0 ;SHIFT INTO HIGH ORDER
938 PRINTB @PRIA0,R0,R2 ;PRINT MEMORY ADDRESS IN ERROR
    010272 010246
    010274 010046
    010276 012746 010320
    010302 012746 000003
    010306 010600
    010310 104414
    010312 062706 000010
939 010316 000207
940
941 010320 045 116 045 PRIA0: .ASCIZ 'MEMA MEMORY ERROR ADDRESS = #01#05'
942 .EVEN
943
944 .SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
945
946 ;
947 ;PRINT MEMORY ADDRESS
948 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.

```

PRITADD PRINT MEMORY TEST ADDRESS

```

949
950      ; IMPLICIT INPUTS
951      ;
952      ; ERRHI - HIGH ORDER ADDRESS
953      ; ERRLO - LOW ORDER ADDRESS
954      ;
955      ;
956 010364 PRITADD.
957 010364 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
958 010370 013702 002226 MOV ERRHI,R2 ;GET HIGH ADDRESS
959 010374 013701 002230 MOV ERRLO,R1 ;GET LOW ADDRESS
960      ;MOV R1,R2 ;COPY LOW ADDRESS
961      ;ROL R1 ;SHIFT BIT 15 TO C BIT
962      ;ROL R0 ;SHIFT INTO HIGH ORDER
963 010400 PRINTB @PRIT0,R1 ;PRINT MEMORY ADDRESS LOW IN ERROR
    010400 010146 MOV R1,-(SP)
    010402 012746 010446 MOV @PRIT0,-(SP)
    010406 012746 000002 MOV @2,-(SP)
    010412 010600 MOV SP,R0
    010414 104414 TRAP C:PNTB
    010416 062706 000006 ADD @6,SP
964 010422 PRINTB @PRIT1,R2 ;PRINT MEMORY ADDRESS HIGH IN ERROR
    010422 010246 MOV R2,-(SP)
    010424 012746 010511 MOV @PRIT1,-(SP)
    010430 012746 000002 MOV @2,-(SP)
    010434 010600 MOV SP,R0
    010436 104414 TRAP C:PNTB
    010440 062706 000006 ADD @6,SP
965 010444 000207 RTS PC ;RETURN
966
967 010446 045 116 045 PRIT0: .ASCIZ 'MEMA MEMORY TEST ADDRESS LOW = #06'
968 010511 045 116 045 PRIT1: .ASCIZ 'MEMA MEMORY TEST ADDRESS HIGH = #06'
969      .EVEN
970      .SBTTL SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND
971
972
973      ;*
974      ;
975      ;ROUTINE TO ISSUE A SPACE RECORDS
976      ;COMMAND (FORWARD OR REVERSE)
977      ;
978      ;INPUT:
979      ;
980      ; R3 NUMBER OF RECORDS TO BE SPACED OVER
981      ; BIT15 CONTROLS DIRECTION
982      ; BIT15 = 0 IS FORWARD
983      ; BIT15 = 1 IS REVERSE
984      ; R5 FIRST DEVICE UNIBUS ADDRESS
985      ;
986      ; REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
987      ;
988      ;OUTPUT:
989      ;
990      ; CARRY SET - SPACE RECORDS COMMAND OK
991      ; CLR - SPACE RECORDS FAILED
992      ;
993      ;
994      ; R0 THE CONTENTS OF R4 IS MOVED TO R0

```

SPACE SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```

994
995
996      ;IMPLICIT OUTPUT:
997      ;
998      ;     TAPE HAS BEEN MOVED
999      ;
1000     ;SIDE EFFECTS:
1001     ;
1002     ;
1003     ;-
1004
1005 010556      SPACE::
1006 010556      SAVREG
1007 010562 012737 000764 010750      MOV     #500.,SDELAY      ;SAVE THE GENERAL REGISTERS
1008 010570 012737 140010 010740      MOV     #140010,80$      ;SET UP DELAY
1009 010576 005703      TST     R3                ;SET UP COMMAND, SPACE FORWARD
1010 010600 10C403      BMI     5$                ;CHECK FOR DIRECTION
1011 010602 010337 010742      MOV     R3,90$           ;BR, IF REVERSE INDICATED
1012 010606 000407      BR      10$              ;LOAD UP NUMBER OF RECORDS TO SPACE
1013 010610 042703 100000      5$:   BIC     #BIT15,R3      ;GO DO COMMAND
1014 010614 010337 010742      MOV     R3,90$           ;CLEAR DIRECTION BIT
1015 010620 052737 000400 010740      BIS     #BIT8,80$        ;LOAD UP NUMBER OF RECORDS TO SPACE
1016 010626 012704 010740      10$:  MOV     #80$,R4        ;SET REVERSE BIT IN COMMAND PACKET
1017 010632 010465 000000      MOV     R4,TSDB(R5)      ;SET UP R4 WITH PACKET ADDRESS
1018 010636 004737 016340      15$:  JSR     PC,WAITF       ;SEND OUT COMMAND
1019 010642 103420      BCS     20$              ;WAIT FOR SSR
1020 010644      DELAY  250              ;BR, IF SSR IS SET AND OK
                                ;DELAY ABOUT .25 SECONDS
                                010644 012727 000250      MOV     #250,(PC)+
                                010650 000000      .WORD  0
                                010652 013727 002116      MOV     L$DLY,(PC)+
                                010656 000000      .WORD  0
                                010660 005367 177772      DEC     -6(PC)
                                010664 001375      BNE     .-4
                                010666 005367 177756      DEC     -22(PC)
                                010672 001367      BNE     .-20
1021 010674 005337 010750      DEC     SDELAY
1022 010700 001356      BNE     15$              ;BUMP DELAY COUNTER DOWN
1023 010702 000411      BR      60$              ;BR, IF MORE DELAY
1024 010704 016501 000002      20$:  MOV     TSSR(R5),R1    ;BR IF TROUBLE CARRY = CLEAR
1025 010710 012702 000200      MOV     #SSR,R2          ;READ TSSR
1026 010714 020201      25$:  CMP     R2,R1           ;SET UP EXPECTED
1027 010716 001401      BEQ     40$              ;ARE THEY OK
1028 010720 000402      BR      60$              ;BR, IF EQUAL = OK
1029 010722 000261      40$:  SEC                ;TROUBLE EXIT
1030 010724 000401      BR      70$              ;SET CARRY NO TROUBLE
1031 010726 000241      60$:  CLC                ;EXIT
1032 010730      70$:
1033 010730 010400      MOV     R4,R0           ;PASS PACKET ADDRESS
1034 010732 000207      RTS     PC              ;RETURN

```


SPACE SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```

1036 ;
1037 ;
1038 ;
1039 ;PACKET FOR SPACE COMMAND
1040 ;
1042 010740 .=<.*10>&177770
1044 ;
1045 ;COMMAND WORD
1046 010740 000000 80$: .WORD
1047 ;NUMBER OF RECORDS TO BE SPACED OVER WORD
1048 010742 000000 90$: .WORD
1049 010744 000000 .WORD
1050 010746 000000 .WORD
1051 010750 000000 SDELAY: .WORD 0 ;DELAY COUNTER
1052 .EVEN
1053 .SBTTL WRTCHR WRITE CHARACTERISTICS COMMAND
1054 ;
1055 ;*
1056 ;
1057 ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1058 ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1059 ;
1060 ;INPUT:
1061 ;
1062 ; R4 ADDRESS OF PACKET FROM TEST
1063 ; R5 FIRST DEVICE UNIBUS ADDRESS
1064 ; REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1065 ;
1066 ;OUTPUT:
1067 ;
1068 ; R0 TSSR CONTENTS
1069 ; CARRY SET - WRITE CHARACTERISTICS COMMAND OK
1070 ; CLR - WRITE CHARACTERISTICS FAILED
1071 ;
1072 ;IMPLICIT OUTPUT:
1073 ;
1074 ; MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
1075 ; SOFTWARE SWITCHES SET AS FOLLOWS:
1076 ; EXTFEA = EXTENDED FEATURES PRESENT
1077 ; BENBSW = BUFFER ENABLE SWITCH ON OR OFF
1078 ;
1079 ;
1080 ;SIDE EFFECTS:
1081 ;
1082 ;
1083 ;-
1084 ;
1085 010752 WRTCHR::
1086 010752 SAVREG
1087 010756 005037 002220 CLR BENBSW ;SAVE THE GENERAL REGISTERS
1088 010762 005037 002216 CLR EXTFEA ;CLEAR BUFFER ENABLE SWITCH
1089 010766 010465 000000 10$: MOV R4,TSD8(R5) ;CLEAR EXTENDED FEATURES SW SWITCH
1090 010772 004737 016426 JSR PC,CHKTSSR ;SEND OUT COMMAND
1091 010776 103401 BCS 20$ ;WAIT FOR SSR
1092 011000 000435 BR 60$ ;BR, IF SSR IS SET AND OK
1093 011002 016501 000002 20$: MOV TSSR(R5),R1 ;BR IF TROUBLE CARRY = CLEAR
1094 011006 012702 000200 MOV #SSR,R2 ;READ TSSR
;SET UP EXPECTED

```

WRTCHR WRITE CHARACTERISTICS COMMAND

```

1095 011012 032701 000100 BIT #OFL,R1 ;WAS OFF LINE SET IN TSSR
1096 011016 001402 BEQ 25$ ;BR, IF NO OFL SET
1097 011020 052702 000100 BIS #OFL,R2 ;MAKE THEM LOOK ALIKE
1098 011024 020201 25$: CMP R2,R1 ;ARE THEY OK
1099 011026 001401 BEQ 40$ ;BR, IF EQUAL = OK
1100 011030 000421 BR 60$ ;TROUBLE EXIT
1101 011032 062704 000010 40$: ADD #8.,R4 ;POINT TO WRT CHARA DATA PACKET
1102 011036 011403 MOV (R4),R3 ;GET ADDRESS OF MESSAGE BUFFER
1103 011040 032763 000200 000012 BIT #X2.EXTF,XST2(R3) ;EXTENDED FEATURES BIT SET?
1104 011046 001402 BEQ 45$ ;BR IF NO
1105 011050 005237 002216 INC EXTFEA ;SET EXTENDED FEATURES SW SWITCH
1106 011054 45$:
1107 011054 032763 000100 000012 BIT #X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
1108 011062 001402 BEQ 50$ ;BR, IF SWITCH NOT SET
1109 011064 005237 002220 INC BENBSW ;SET SOFTWARE SWITCH FOR ENABLED
1110 011070 50$:
1111 011070 00C261 SEC ;SET CARRY NO TROUBLE
1112 011072 000401 BR 70$ ;EXIT
1113 011074 000241 60$: CLC ;CARRY CLEAR = ERROR
1114 011076 016500 000002 70$: MOV TSSR(R5),R0 ;RETURN TSSR CONTENTS
1115 011102 000207 RTS PC ;RETURN
1116 .SBTTL REWIND - POSITION TAPE (REWIND) COMMAND
1117
1118 ;*
1119 ;
1120 ; THIS ROUTINE WILL REWIND THE SELECTED TAPE.
1121 ;
1122 ; CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
1123 ; TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
1124 ; SSR TO SET IN THE TSSR
1125 ;
1126 ;
1127 ; CALLING SEQUENCE:
1128 ;
1129 ; DO A SOFT INIT
1130 ; DO A WRITE CHARACTERISTICS
1131 ; JSR PC,REWIND
1132 ;
1133 ; INPUT:
1134 ;
1135 ; R5 FIRST DEVICE UNIBUS ADDRESS
1136 ;
1137 ;
1138 ; OUTPUT
1139 ;
1140 ; R0 THE CONTENTS OF R4 IS PASSED TO R0
1141 ;
1142 ;
1143 ;-
1144 REWIND::
1145 011104 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1146 011110 012704 011200 MOV #RWPACK,R4 ;GET PACKET ADDRESS
1147 011114 010465 000000 MOV R4,TSDB(R5) ;SEND PACKET ADDRESS TO EXECUTE
1148 011120 012703 000550 MOV #360.,R3 ;ENOUGH TIME FOR 2400' REEL TO REWIND
1149 011124 004737 016340 10$: JSR PC,WAITF ;WAIT FOR SSR TO SET
1150 011130 103417 BCS 20$ ;LEAVE WHEN SSR IS SET
1151 011132 DELAY 250. ;WAIT FOR .25 SECONDS

```

REWIND POSITION TAPE (REWIND) COMMAND

```

011132 012727 000372      MOV      #250.,(PC)+
011136 000000      .WORD   0
011140 013727 002116      MOV      L$DLT,(PC)+
011144 000000      .WORD   0
011146 005367 177772      DEC      -6(PC)
011152 001375      BNE      . 4
011154 005367 177756      DEC      -22(PC)
011160 001367      BNE      . -20
1152 011162 005303      DEC      R3          ;BUMP COUNTER DOWN
1153 011164 001357      BNE      10$        ;KEEP GOING
1154 011166 000241      CLC          ;CLEAR CARRY TO SET ERROR
1155 011170 010400      20$: MOV      R4,R0   ;PASS THE PACKET ADDRESS
1156 011172 000207      RTS         PC      ;RETURN

```

```

1159          011200      RWPACK: .=<..+10>&17777U
1161 011200      .WORD   102010      ;POSTION COMMAND (REWIND)
1162 011200 102010      .WORD   0           ;NOT USED
1163 011202 000000      .SBTTL  CKRAM      - COMPARE RAM TO I/O PACKET

```

```

1166          ;*
1167          ;
1168          ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
1169          ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
1170          ;
1171          ;INPUT:
1172          ;
1173          ;      R4      ADDRESS OF THE COMMAND PACKET
1174          ;      R5      FIRST DEVICE UNIBUS ADDRESS
1175          ;
1176          ;OUTPUT:
1177          ;
1178          ;      CARRY   SET - RAM MATCHES PACKET
1179          ;             CLR - RAM DOES NOT MATCH PACKET
1180          ;
1181          ;IMPLICIT OUTPUT:
1182          ;
1183          ;      THE TABLE RAMDATA IS FILLED WITH THE
1184          ;      DATA HELD IN RAM.
1185          ;      RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
1186          ;
1187          ;SIDE EFFECTS:
1188          ;
1189          ;      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1190          ;
1191          ;-

```

```

1192          CKRAM::
1193 011204      SAVREG          ;SAVE THE GENERAL REGISTERS
1194 011204      MOV      #RAMDATA,R1 ;ADDRESS TO SAVE THE RAM DATA
1195 011210 012701 002232      MOV      #RMPKTBEG,R2 ;BYTE ADDRESS OF FIRST RAM DATA
1196 011214 012702 000201      CLR      R3          ;CLEAR THE ERROR FLAG
1197 011220 005003      JSR      PC,CHKTSSR ;WAIT FOR SSR
1198 011222 004737 016426      MOVB    #0,TSUB(R5) ;SET MAINTENANCE MODE
1199 011226 112765 000000 000000 10$: JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
1200 011234 004737 016426      MOV      R2,TSDB(R5) ;SELECT NEXT RAM ADDRESS
1201 011240 010265 000000      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
1202 011244 004737 016426

```

CKRAM COMPARE RAM TO I/O PACKET

```

1203 011250 116511 000000      MOVB    TSBA(R5),(R1)      ;READ THE RAM DATA
1204 011254 122124      CMPB    (R1)+,(R4)+      ;COMPARE TO EXPECTED
1205 011256 001401      BEQ     20$              ;BRANCH IF OK
1206 011260 005203      INC     R3               ;SET ERROR FLAG
1207 011262 005202      20$:  INC     R2          ;ADDRESS OF NEXT RAM LOCATION
1208 011264 020227 000210      CMP     R2,#RMPKTEND     ;REACHED END YET ?
1209 011270 003761      BLE    10$              ;BRANCH TILL ALL READ
1210 011272 005703      TST    R3               ;WAS AN ERROR FOUND ?
1211 011274 001402      BEQ     30$              ;BRANCH IF NOT
1212 011276 000241      CLC                    ;CLEAR CARRY TO SHOW ERROR
1213 011300 000401      BR     50$              ;AND EXIT
1214 011302 000261      30$:  SEC                    ;SHOW GOOD COMPARE
1215 011304 012737 000010 002272 50$:  MOV     #8.,RAMSIZ       ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
1216 011312 000207      RTS     PC               ;RETURN
1217                                     .SBITL  CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA
1218                                     ;+
1219                                     ;
1220                                     ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
1221                                     ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
1222                                     ;
1223                                     ;INPUT:
1224                                     ;
1225                                     ;       R4      ADDRESS OF THE CHARACTERISTICS DATA
1226                                     ;       R5      FIRST DEVICE UNIBUS ADDRESS
1227                                     ;
1228                                     ;OUTPUT:
1229                                     ;
1230                                     ;       CARRY   SET - RAM MATCHES PACKET
1231                                     ;               CLR - RAM DOES NOT MATCH PACKET
1232                                     ;
1233                                     ;IMPLICIT OUTPUT:
1234                                     ;
1235                                     ;       THE TABLE RAMDATA IS FILLED WITH THE
1236                                     ;       DATA HELD IN RAM.
1237                                     ;       RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
1238                                     ;
1239                                     ;SIDE EFFECTS:
1240                                     ;
1241                                     ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1242                                     ;
1243                                     ;-
1244
1245 011314      CKRAM2::
1246 011314      SAVREG                    ;SAVE THE GENERAL REGISTERS
1247 011320 012701 002232      MOV     #RAMDATA,R1      ;ADDRESS TO SAVE THE RAM DATA
1248 011324 012702 000167      MOV     #RMCHBEG,R2     ;BYTE ADDRESS OF FIRST RAM DATA
1249 011330 005003      CLR     R3               ;CLEAR THE ERROR FLAG
1250 011332 004737 016426      JSR    PC,CHKTSSR       ;WAIT FOR SSR
1251 011336 112765 000000 000000      MOVB   #0,TSDB(R5)      ;SET MAINTENANCE MODE
1252 011344 004737 016426      10$:  JSR    PC,CHKTSSR       ;WAIT FOR SSR TO SET
1253 011350 010265 000000      MOV     R2,TSDB(R5)     ;SELECT NEXT RAM ADDRESS
1254 011354 004737 016426      JSR    PC,CHKTSSR       ;WAIT FOR SSR TO SET
1255 011360 116511 000000      MOVB   TSBA(R5),(R1)    ;READ THE RAM DATA
1256 011364 122124      CMPB   (R1)+,(R4)+     ;COMPARE TO EXPECTED
1257 011366 001401      BEQ    20$              ;BRANCH IF OK
1258 011370 005203      INC    R3               ;SET ERROR FLAG
1259 011372 005202      20$:  INC    R2          ;ADDRESS OF NEXT RAM LOCATION

```


CKMSG COMPARE WRITE CHAR. MESSAGE BUFFERS

```

1317 011530 020427 000014      CMP      R4,#14      ;DONE FIRST 7 WORDS?
1318 011534 003764              BLE      15$        ;BR IF NO
1319 011536 032765 000200 000012  BIT      #X2.EXTF,XST2(R5); IS EXTENDED FEATURES SET IN EXPD?
1320 011544 001403              BEQ      50$        ;BR IF NO
1321 011546 020427 000016      CMP      R4,#16      ;DONE EXTENDED FEATURES WORD?
1322 011552 003755              BLE      15$        ;BR IF NO
1323 011554 005703              50$:    TST      R3      ;ANY ERRORS SEEN?
1324 011556 001402              BEQ      55$        ;BR IF NO
1325 011560 000241              CLC                      ;SET FAILURE
1326 011562 000401              BR       60$        ;
1327 011564 000261              55$:    SEC                      ;SET SUCCESS
1328 011566 000207              60$:    RTS      PC      ;RETURN
1329
1330                          .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
1331
1332                          ;*
1333                          ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
1334                          ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1335                          ;ERROR PRINT ROUTINES.
1336                          ;INPUT:
1337
1338                          ;
1339                          ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1340                          ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1341                          ;      R2      EXPD MESSAGE BUFFER ADDRESS
1342                          ;      R3      NUMBER OF BYTES TO COMPARE
1343                          ;OUTPUT:
1344
1345                          ;      CARRY   SET - MESSAGE BUFFERS MATCH
1346                          ;      CLR     - MESSAGE BUFFERS DON'T MATCH
1347
1348                          ;IMPLICIT OUTPUT:
1349
1350                          ;      EXPMSG   BUFFER IS SET TO EXPD DATA
1351                          ;      RECVMSG  BUFFER IS SET TO RECV DATA
1352                          ;      RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1353                          ;      RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
1354
1355
1356 011570
1357 011570
1358 011574 020327 000144      *KMSG2:: SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
1359 011600 003412              CMP      R3,#RECVMSG-EXPMSG; @@D IS COUNT ABOVE MAX ALLOWED?
1360 011602 012703 000144      BLE      5$        ;@@D BR IF NO
1361 011606              MOV      #RECVMSG-EXPMSG,R3;@@D
1362 011606 012746 011722      PRINTF  #DEBUGMSG   ;@@D
1363 011612 012746 000001      MOV      #DEBUGMSG,-(SP)
1364 011616 010600              MOV      #1,-(SP)
1365 011620 104417              MOV      SP,R0
1366 011622 062706 000004      TRAP    C#PNTF
1367 011626 010037 002274      5$:    ADD      #4,SP
1368 011632 010137 002276      MOV      R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
1369 011636 005737 003124      MOV      R1,RCVLOAD  ;SAVE RECV LOW ADDRESS
1370 011642 001403              TST      KENABLE     ;TESTING ABOVE 28K?
1371 011644 004737 017406      BEQ      10$        ;BR IF NO
1372 011650 010001              JSR      PC,SETMAP   ;RETURN ADDRESS BIASED TO PAR6 IN R0
1373 011652 005004              MOV      R0,R1      ;GET RETURNED ADDRESS BIASED TO PAR6
1374                          10$:    CLR      R4          ;WORD IN BUFFER

```

CKMSG2 COMPARE EXPD RECV MESSAGE BUFFERS

```

1369 011654 005005          CLR      R5          ;CLEAR ERROR SEEN FLAG
1370 011656 111264 002312 15:  MOVB    (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1371 011652 111164 002456          MOVB    (R1),RCFMSG(R4) ;SAVE RECV FOR ERROR REPORT
1372 011666 122221          CMPB    (R2),.(R1).    ;EXPD EQUAL RECV?
1373 011670 001401          BEQ     25:          ;BR IF YES
1374 011672 005205          INC     R5          ;SET ERROR SEEN FLAG
1375 011674 062704 000001 25:  ADD     @1,R4        ;POINT TO NEXT BYTE
1376 011700 020403          CMP     R4,R3       ;DONE ALL BYTES?
1377 011702 002001          BGE    50:          ;BR IF YES
1378 011704 000764          BR     15:          ;DO NEXT BYTE
1379 011706 005705          50:  TST     R5          ;ANY ERRORS SEEN?
1380 011710 001402          BEQ    55:          ;BR IF NO
1381 011712 000241          CLC                    ;SET FAILURE
1382 011714 000401          BR     60:          ;
1383 011716 000261          55:  SEC                    ;SET SUCCESS
1384 011720 000207          60:  RTS     PC         ;RETURN
1385
1386 011722          120      122      117  DEBUGMSG:      .ASCIZ  'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED ' ;@@D
1387 012012          045      116      045  FERCM:  .ASCII  /WMA ***/
1388 012023          040      040      124  ERCM:   .ASCIZ  / TSSR ERROR CODE REC'D = /
1389 012056          056      056      056  SIMSG: .ASCIZ  /... AFTER DOING SOFT INIT/
1390 012111          124      105      123  TINERR: .ASCIZ  /TEST: .../
1391
1392
1393
1394          ;*
1395          ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
1396
1397          ;INPUT:
1398
1399          ;          R1      CONTENTS OF TSSR AT ERROR
1400
1401          ;SIDE EFFECTS:
1402
1403          ;          EXECUTES DROP UNIT TO CEASE TESTING
1404
1405          ;-
1406
1407 012124          BGNMSG  SFIMSG
1408 012124 004737 006022 SFIMSG: JSR     PC,PRITSSR ;PRINT CONTENTS OF TSSR REGISTER
1409 012130 004737 017272          JSR     PC,CKDROP  ;DROP UNIT, IF ALLOWED
1410 012134
1411 012134
1412 012134 104423          L10003: TRAP    CMSG
1413
1414          ;*
1415          ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1416          ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
1417
1418          ;INPUTS:
1419
1420          ;          R1      TSSR CONTENTS
1421          ;          R4      ADDRESS OF COMMAND PACKET
1422

```

CKMSG2 COMPARE EXPD RECV MESSAGE BUFFERS

```

1423 012136          BGNMSG  PKTSSR
      012136          PRTSSR:
1424 012136 004737 006022      JSR    PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1425 012142 012700 000004      MOV    #4,R0          ;NO. OF WORDS IN PACKET
1426 012146 004737 007460      JSR    PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1427 012152          ENDMSG
      012152          L10004:
      012152 104423      TRAP   C$MSG

1428
1429
1430
1431          ;*
1432          ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1433          ;TSSR AND A GET STATUS COMMAND PACKET.
1434          ;
1435          ;INPUTS:
1436          ;
1437          ;       R1      TSSR CONTENTS
1438          ;       R4      ADDRESS OF COMMAND PACKET
1439          ;
1440 012154          BGNMSG  PKTGETS
      012154          PKTGETS:
1441 012154 004737 006022      JSR    PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1442 012160 012700 000002      MOV    #2,R0          ;NO. OF WORDS IN GET STATUS PACKET
1443 012164 004737 007460      JSR    PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1444 012170          ENDMSG
      012170          L10005:
      012170 104423      TRAP   C$MSG

1445
1446
1447          ;*
1448          ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1449          ;
1450          ;INPUTS:
1451          ;
1452          ;       R1      TSSR CONTENTS
1453          ;       R4      ADDRESS OF COMMAND PACKET
1454          ;
1455 012172          BGNMSG  SFFMSG
      012172          SFFMSG:
1456 012172 004737 006022      JSR    PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1457 012176          ENDMSG
      012176          L10006:
      012176 104423      TRAP   C$MSG

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

```


PKTMES PRINT TSSR AND MESSAGE BUFFER

```

1471
1472 012200
      012200
1473 012200 004737 006022
1474 012204 010200
1475 012206 010301
1476 012210 004737 014332
1477 012214
      012214
      012214 104423
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490 012216
      012216
1491 012216 004737 010364
1492 012222 016501 000002
1493 012226 004737 006022
1494 012232
      012232
      012232 104423
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508 012234
      012234
1509 012234 012700 000007
1510 012240 005737 002216
1511 012244 001402
1512 012246 012700 000010
1513 012252 004737 014642
1514 012256
      012256
      012256 104423
1515
1516
1517
1518

```

```

;
; BGNMSG PKTMES
PKTMES::
JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR
MOV R2,R0 ;LOW ORDER ADDRESS
MOV R3,R1 ;HIGH ORDER ADDRESS
JSR PC,PRMESS ;PRINT THE MESSAGE BUFFER
ENDMSG
L10007:
TRAP C$MSG
.SBTTL ADDSSR - PRINT TEST ADDRESS AND TSSR
;
;PRINT ROUTINE TO PRINT THE CONTENTS OF
;TSSR AND A MEMORY TEST ADDRESS
;
;INPUTS:
;
; R5 FIRST DEVICE UNIBUS ADDRESS
; ERRHI HIGH ORDER MEMORY TEST ADDRESS
; ERRLO LOW ORDER MEMORY TEST ADDRESS
;-
;
; BGNMSG ADDSSR
ADDSSR::
JSR PC,PRITADD ;PRINT MEMORY TEST ADDRESS
MOV TSSR(R5),R1 ;GET CURRENT TSSR
JSR PC,PRITSSR ;PRINT THE CONTENTS OF TSSR REGISTER
ENDMSG
L10010:
TRAP C$MSG
.SBTTL MSGEXP PRINT WRITE CHAR. EXPD-RCV MESSAGE BUFFERS
;
;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
;
;IMPLICIT INPUTS:
;
; EXPMSG - EXPECTED MESSAGE BUFFER
; RECMSG - RECEIVED MESSAGE BUFFER
; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
;-
; BGNMSG MSGEXP
MSGEXP::
MOV #7,R0 ;ASSUME NO EXT FEATURES
TST EXTFEA ;EXT FEATURES SET?
BEQ S$ ;BR IF NO
MOV #8,R0 ;EXT FEATURE BUFFER IS 8 WORDS
S$: JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
ENDMSG
L10011:
TRAP C$MSG
.SBTTL FIFEXP - PRINT FIFO EXP/RCV DATA
;
;PRINT ROUTINE TO PRINT FIFO EXP/RCV DATA

```

F1,

FIFEXP PRINT FIFO EXP/RCV DATA

```

1519
1520
1521
1522
1523
1524
1525
1526
1527 012260
      012260
1528 012260 010146 012332
      012262 012746 012332
      012266 012746 000002
      012272 010600
      012274 104415
      012276 062706 000006
1529 012302
      012302 012746 012401
      012306 012746 000001
      012312 010600
      012314 104415
      012316 062706 000004
1530 012322 010100
1531 012324 004737 015212
1532 012330
      012330
      012330 104423
1533 012332 045 116 045 FIF1MSG:
1534 012401 045 116 045 FIF2MSG:
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549 012440
      012440
1550 012440 012701 012502
1551 012444 012100
1552 012446 001410
1553 012450
      012450 010046
      012452 012746 000001
      012456 010600
      012460 104415
      012462 062706 000004
1554 012466 000766
1555 012470 012700 000012

```

```

:
: R1 BYTE COUNT
:
:IMPLICIT INPUTS:
:
: EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
: RECVMSG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
:
:
: BGNMSG FIFEXP
FIFEXP:
: PRINTX @FIF1MSG,R1 ;PRINT BYTES TRANSFERRED
: MOV R1,-(SP)
: MOV @FIF1MSG,-(SP)
: MOV @2,-(SP)
: MOV SP,R0
: TRAP C$PNTX
: ADD @6,SP
: PRINTX @FIF2MSG ;PRINT HEADER MSG
: MOV @FIF2MSG,-(SP)
: MOV @1,-(SP)
: MOV SP,R0
: TRAP C$PNTX
: ADD @4,SP
: MOV R1,R0 ;GET BYTE COUNT
: JSR PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
: ENDMMSG
L10012:
: TRAP C$MSG
: .ASCIZ '###A NUMBER OF BYTES TRANSFERRED = #D2'
: .ASCIZ '###A FIFO DATA BYTES IN ERROR:'
: .EVEN
: .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
:
:
:PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
:
:IMPLICIT INPUTS:
:
: EXPMSG - EXPECTED MESSAGE BUFFER
: RECVMSG - RECEIVED MESSAGE BUFFER
: RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
: RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
:
:
: BGNMSG MSGSTAT
MSGSTAT:
: MOV @STATCOD,R1 ;ASCII ADDRESS TABLE
10$: : MOV (R1)+,R0 ;DONE ALL MSG LINES?
: BEQ 20$ ;BR IF YES
: PRINTX R0 ;PRINT STATUS BIT NAMES
: MOV R0,(SP)
: MOV @1,-(SP)
: MOV SP,R0
: TRAP C$PNTX
: ADD @4,SP
: BR 10$ ;DO ANOTHER MSG LINE
20$: : MOV @10.,R0 ;NUMBER OF WORDS IN A READ STATUS BUFFER

```

MSGSTAT PRINT STATUS HEADER AND MESSAGE BUFFERS

```

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

```

MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```

1603      ;
1604      ;
1605      ;IMPLICIT INPUTS:
1606      ;
1607      ;     EXPMSG - EXPECTED MESSAGE BUFFER
1608      ;     RECMSG - RECEIVED MESSAGE BUFFER
1609      ;     RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1610      ;     RCVLOADD RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1611      ;
1612      ;     BGNMSG MSGSUB
1613      ;MSGSUB::
1614      ;     MOV     #10,R0           ;SIZE OF WRITE SUBSYSTEM BUFFER
1615      ;     JSR     PC,PRMSGEXP    ;PRINT EXPD/RCV MESSAGE BUFFERS
1616      ;     ENDMSG
1617      ;
1618      ;L10015:
1619      ;     TRAP   C$MSG
1620      ;
1621      ;     .SBTTL MEMADD PRINT MEMORY ADDRESS DATA ERROR
1622      ;
1623      ;*
1624      ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
1625      ;
1626      ;IMPLICIT INPUTS:
1627      ;
1628      ;     ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
1629      ;     ERRLO - MEMORY ERROR LOW ORDER ADDRESS
1630      ;     EXP   - EXPECTED DATA
1631      ;     RECV  - RECEIVED DATA
1632      ;
1633      ;     BGNMSG MEMADD
1634      ;MEMADD::
1635      ;     JSR     PC,PRIADD        ;PRINT MEMORY ADDRESS IN ERROR
1636      ;     MOV     EXPD,R1         ;GET EXPD DATA
1637      ;     MOV     RECV,R2         ;GET RECEIVED DATA
1638      ;     JSR     PC,PRIXOR       ;PRINT EXPD/RCV
1639      ;     ENDMSG
1640      ;L10016:
1641      ;     TRAP   C$MSG
1642      ;     .SBTTL PRAMPKT - PRINT RAM AND PACKET DATA
1643      ;
1644      ;*
1645      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1646      ;WHEN THE RAM DATA DOES NOT MATCH.
1647      ;
1648      ;INPUTS:
1649      ;
1650      ;     R4     POINTER TO COMMAND PACKET
1651      ;
1652      ;IMPLICIT INPUTS:
1653      ;
1654      ;     RAMDATA DATA AS READ FROM THE RAM
1655      ;     RAMSIZ  NUMBER OF BYTES IN PACKET
1656      ;             IF RAMSIZ=0 THEN DEFAULT TO 8.
1657      ;
1658      ;IMPLICIT OUTPUTS:
1659      ;
1660      ;     RAMSIZ SET TO 0

```

H5

PRAMPKT PRINT RAM AND PACKET DATA

```

1654
1655
1656 014066
1657 014066
1658 014072 012701 002232
1659 014076 005002
1660 014100 122124
1661 014102 001005
1662 014104
1663 014114 000436
1664 014116 116105 177777
1665 014122 116403 177777
1666 014126
1667 014136 042703 177400
1668 014142 116137 177777 002224
1669 014150 116437 177777 002222
1670 014156
    014156 010346
    014160 013746 002222
    014164 013746 002224
    014170 010246
    014172 012746 014246
    014176 012746 000005
    014202 010600
    014204 104414
    014206 062706 000014
1671 014212 005202
1672 014214 005737 002272
1673 014220 001404
1674 014222 020237 002272
1675 014226 003724
1676 014230 000403
1677 014232 020227 000010
1678 014236 002720
1679 014240 005037 002272
1680 014244 000207
1681
1682 014246 045 116 045 RAMASC: .ASCIZ 'N#A BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03'
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701 014332

```

```

;
PRAMPKT:
    SAVREG
    MOV    #RAMDATA,R1
    CLR   R2
    5$:   CMPB  (R1),,(R4)
    BNE   7$
    FORCERROR 7$,NOTSSR
    BR    10$
    7$:   MOVB  1(R1),R5
    MOVB  -1(R4),R3
    XOR   R5,R3
    BIC   #177400,R3
    MOVB  -1(R1),RECV
    MOVB  -1(R4),EXPD
    PRINTB #RAMASC,R2,RECV,EXPD,R3
    MOV   R3,-(SP)
    MOV   EXPD,-(SP)
    MOV   RECV,-(SP)
    MOV   R2,(SP)
    MOV   #RAMASC,-(SP)
    MOV   #5,-(SP)
    MOV   SP,R0
    TRAP  C$PNTB
    ADD   #14,SP
    10$:  INC   R2
    TST   RAMSIZ
    BEQ   15$
    CMP   R2,RAMSIZ
    BLE   5$
    BR    25$
    15$:  CMP   R2,#8
    20$:  BLT   5$
    25$:  CLR   RAMSIZ
    RTS   PC
;SAVE R1-R5 UNTIL NEXT RETURN
;DATA FROM THE RAM
;INIT BYTE NUMBER
;COMPARE EXPECTED, RECEIVED
;BR IF NO MATCH
;880
;GET RECV RAM DATA
;GET EXPD PACKET DATA
;XOR EXPD/RECV
;LOW BYTE ONLY
;GET RECEIVED RAM DATA
;GET EXPECTED RAM DATA
;UPDATE BYTE COUNT
;DEFAULT TO 8.?
;BR IF YES
;DONE ALL BYTES?
;BR IF NO
;
;DONE DEFAULT NUMBER OF BYTES?
;BR IF NO
;SET DEFAULT RAMSIZ
;RETURN
;SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
;
; THIS ROUTINE PRINTS THE CONTENTS OF
; THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE
; TSV-05.
;
; INPUT:
;
; R0 LOW ORDER ADDRESS OF MESSAGE BUFFER
; R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER
; NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
;
; THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
;
;
PRMESS:

```

PRMESS PRINT CONTENTS OF MESSAGE BUFFER

```

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

```

PRMSGEXP PRINT EXPD/RCV MESSAGE BUFFERS

```

1740 ; EXPMSG - EXPECTED MESSAGE BUFFER
1741 ; RECMSG RECEIVED MESSAGE BUFFER
1742 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1743 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1744 ;
1745 PRMSGEXP::
1746 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1747 MOV RO,R5 ;SAVE NUMBER OF WORDS
1748 MOV RCVLOADD,RO ;GET RECV LOW ADDRESS
1749 MOV RO,R4 ;COPY LOW ADDRESS
1750 MOV RCVHIADD,R1 ;GET RECV HIGH ADDRESS
1751 ROL RO ;SHIFT BIT15 TO C BIT
1752 ROL R1 ;SHIFT TO HIGH ORDER FOP PRINTOUT
1753 PRINTX @PRMSG0,R1,R4 ;PRINT MESSAGE BUFFER ADDRESS
      MOV R4,-(SP)
      MOV R1,-(SP)
      MOV @PRMSG0,-(SP)
      MOV @3,-(SP)
      MOV SP,RO
      TRAP C:PNTX
      ADD @10,SP
1754 PRINTX @PRMSG1 ;PRINT HEADER FOR CONTENTS
      MOV @PRMSG1,-(SP)
      MOV @1,-(SP)
      MOV SP,RO
      TRAP C:PNTX
      ADD @4,SP
      CLR R4 ;NUMBER OF THE CURRENT WORD
1755 MOV @EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
1756 MOV @RECMSG,R2 ;GET RECV BUFFER ADDRESS
1757 MOV (R1),RO ;GET EXPD
1758 MOV (R2),R3 ;GET RECV
1759 XOR RO,R3 ;XOR EXPD/RCV
1760 PRINTX @PRMSG2,R4,(R1)+,(R2)+,R3
1761 MOV R3,-(SP)
      MOV (R2)+,-(SP)
      MOV (R1)+,-(SP)
      MOV R4,-(SP)
      MOV @PRMSG2,-(SP)
      MOV @5,-(SP)
      MOV SP,RO
      TRAP C:PNTX
      ADD @14,SP
1762 INC R4 ;NUMBER OF THE NEXT
1763 CMP R4,R5 ;DONE ALL YET?
1764 BGE 50$ ;BR IF YES
1765 BR 20$ ;DO ANOTHER
1766 50$: RTS PC ;RETURN
1767
1768 045 116 045 PRMSG0: .ASCIZ 'N#A Message Buffer Address = #01#05'
1769 045 116 045 PRMSG1: .ASCIZ 'N#A Message Buffer Contents:'
1770 045 116 045 PRMSG2: .ASCIZ 'N#A WORD #D2#A EXPD: #06#A RECV: #06#A XOR: #06
1771 .EVEN
1772 .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
1773 ;*
1774 ;
1775 ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS

```

PRBYTEXP PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

1776 ; ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
1777 ;
1778 ; RO - NUMBER OF BYTES IN BUFFER
1779 ;
1780 ;IMPLICIT INPUTS:
1781 ;
1782 ; EXPMSG - EXPECTED MESSAGE BUFFER
1783 ; RECMMSG - RECEIVED MESSAGE BUFFER
1784 ;
1785 015212 PRBYTEXP::
1786 015212 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1787 015216 010005 MOV R0,R5 ;SAVE NUMBER OF BYTES
1788 015220 005037 002310 CLR PRMNO ;INIT ERROR COUNT
1789 015224 005004 CLR R4 ;NUMBER OF THE CURRENT BYTE
1790 015226 012701 002312 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
1791 015232 012702 002456 MOV #RECMMSG,R2 ;GET RECV BUFFER ADDRESS
1792 015236 111100 20$: MOV R0,(R1),R0 ;GET EXPD BYTE
1793 015240 042700 177400 BIC #C<377>,R0 ;CLEAR UPPER BYTE
1794 015244 110037 015560 MOV R0,PRBEXP ;SAVE FOR ERROR REPORT
1795 015250 111203 MOV R2,R3 ;GET RECV BYTE
1796 015252 042703 177400 BIC #C<377>,R3 ;CLEAR UPPER BYTE
1797 015256 110337 015562 MOV R3,PRBREC ;FOR ERROR REPORT
1798 015262 XOR R0,R3 ;XOR EXPD/RECV
1799 015272 122122 CMPB (R1)+,(R2)+ ;EXPD = RECV?
1800 015274 001431 BEQ 30$ ;BR IF YES
1801 015276 005237 002310 INC PRMNO ;UPDATE ERROR COUNT
1802 015302 023727 002310 000010 CMP PRMNO,#8. ;PRINTED 8?
1803 015310 101023 BHI 30$ ;BR IF YES
1804 015312 27$: PRINTX #PRBMSG,R4,PRBEXP,PRBREC,R3
015312 010346 MOV R3,-(SP)
015314 013746 015562 MOV PRBREC,-(SP)
015320 013746 015560 MOV PRBEXP,-(SP)
015324 010446 MOV R4,-(SP)
015326 012746 015426 MOV #PRBMSG,-(SP)
015332 012746 000005 MOV #5,-(SP)
015336 010600 MOV SP,R0
015340 104415 TRAP C$PNTX
015342 062706 000014 ADD #14,SP
1805 015346 FORCEEXIT 50$ ;@@D
1806 015356 000404 BR 35$ ;@@D
1807 015360 30$:
1808 015360 FORCERROR 27$,NOTSSR ;@@D
1809 015370 35$:
1810 015370 005204 INC R4 ;NUMBER OF THE NEXT
1811 015372 020405 CMP R4,R5 ;DONE ALL YET?
1812 015374 002001 BGE 50$ ;BR IF YES
1813 015376 000717 BR 20$ ;DO ANOTHER
1814 015400 50$: PRINTX #PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
015400 013746 002310 MOV PRMNO,-(SP)
015404 012746 015513 MOV #PRBTOT,-(SP)
015410 012746 000002 MOV #2,-(SP)
015414 010600 MOV SP,R0
015416 104415 TRAP C$PNTX
015420 062706 000006 ADD #6,SP
1815 015424 000207 RTS PC ;RETURN
1816
1817 015426 045 116 045 PRBMSG: .ASCIZ 'N#A BYTE #D2#A EXPD: #03#A RECV: #03#A XOR: #03

```


PRBYTEXP PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

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

```

RAMERR PRINT RAM AND PACKET DATA

1869
 1870
 1871
 1872
 1873
 1874
 1875 015600
 015600
 1876 015600 004737 014066
 1877 015604
 015604
 015604 104423

1878
 1879
 1880
 1881
 1882
 1883
 1884
 1885
 1886
 1887
 1888
 1889
 1890
 1891
 1892
 1893
 1894
 1895
 1896
 1897
 1898
 1899
 1900
 1901 015606
 015606
 1902 015606 004737 010364
 1903 015612 004737 014066
 1904 015616
 015616
 015616 104423

1905
 1906
 1907
 1908
 1909
 1910
 1911
 1912
 1913
 1914
 1915
 1916
 1917
 1918 015620
 015620

```

;
;IMPLICIT OUTPUTS:
;
;      RAMSIZ  SET TO 0
;-
;
;      BGNMSG  RAMERR
RAMERR:  JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
          ENDMSG
L10021:  TRAP    C$MSG
;
;      .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
;+
;
;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
;
;INPUTS:
;
;      R4      POINTER TO COMMAND PACKET
;
;IMPLICIT INPUTS:
;
;      RAMDATA  DATA AS READ FROM THE RAM
;      RAMSIZ  NUMBER OF BYTES IN PACKET
;              IF RAMSIZ=0 THEN DEFAULT TO 8.
;      ERRHI   HIGH ORDER TEST ADDRESS
;      ERRLO   LOW ORDER TEST ADDRESS
;
;IMPLICIT OUTPUTS:
;
;      RAMSIZ  SET TO 0
;-
;
;      BGNMSG  RAMTADD
RAMTADD:  JSR      PC,PRITADD      ;PRINT TEST ADDRESS
          JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
          ENDMSG
L10022:  TRAP    C$MSG
;
;      .SBTTL  RAMEXP - PRINT RAM EXPD/RECV DATA
;+
;
;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
;
;INPUTS:
;
;      R1      RECEIVED DATA
;      R2      EXPECTED DATA
;      R4      CONTROLLER RAM ADDRESS
;-
;
;      BGNMSG  RAMEXP
RAMEXP:  
```

RAMEXP PRINT RAM EXPD/RECV DATA

```

1919 015620 042701 177400
1920 015624 042702 177400
1921 015630 004737 010156
1922 015634 004737 010032
1923 015640
      015640
      015640 104423
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937 015642
      015642
1938 015642
      015642 012746 015670
      015646 012746 000001
      015652 010600
      015654 104415
      015656 062706 000004
1939 015662 004737 010032
1940 015666
      015666
      015666 104423
1941
1942 015670      045      116      045
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957 015770
      015770
1958 015770 010246
1959 015772 042702 177400
1960 015776
      015776 010246
      016000 012746 016030
      016004 012746 000002
      016010 010600

```

```

      BIC      #+C<377>,R1      ;SAVE EXPD RAM DATA BYTE
      BIC      #+C<377>,R2      ;SAVE EXPD RAM DATA BYTE
      JSR      PC,PRIRAM        ;PRINT THE RAM ADDRESS
      JSR      PC,PRIXOR        ;PRINT THE DATA
      ENDMSG
L10023:
      TRAP     C$MSG
      .SBTTL   TIMEXP - PRINT TIMER A,B AND EXP/REC
;+
;
;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
;AND TIMER A,B HEADER MESSAGE
;
;INPUTS:
;
;      R1      RECEIVED DATA
;      R2      EXPECTED DATA
;-
      BGNMSG   TIMEXP
TIMEXP::
      PRINTX   #TIMSGO          ;PRINT HEADER
      MOV      #TIMSGO,-(SP)
      MOV      #1,-(SP)
      MOV      SP,R0
      TRAP     C$PNTX
      ADD      #4,SP
      JSR      PC,PRIXOR        ;PRINT THE DATA
      ENDMSG
L10024:
      TRAP     C$MSG
045 TIMSGO: .ASCIZ 'TIMER A STATUS IS IN BIT 3'
      .EVEN   .SBTTL   BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
;+
;
;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
;
;INPUTS:
;
;      R1      CONTENTS OF TSSR
;      R2      DATA WRITTEN (8 BITS)
;
      BGNMSG   BADSSR
BADSSR::
      MOV      R2,-(SP)          ;SAVE DATA TRANSFERRED
      BIC      #177400,R2       ;GET JUST ONE BYTE
      PRINTB   #XFERASC,R2
      MOV      R2,-(SP)
      MOV      #XFERASC,-(SP)
      MOV      #2,-(SP)
      MOV      SP,R0

```

BADSSR - PRINT TSSR ERROR ON DATA TRANSFERS

```

016012 104414          TRAP  C:PNTB
016014 062706 000006  ADD   #6,SP
1961 016020 012602          MOV   (SP),R2          ;RESTORE R2
1962 016022 004737 006022  JSR   PC,PRITSSR      ;DECODE TSSR CONTENTS
1963 016026          ENDMMSG
016026          L10025:
016026 104423          TRAP  C:MSG
016030 045 116 045 XFERASC. .ASCIZ  #NBA Data Transferred = #03
1964          .SBTTL  GLOBAL SUBROUTINES SECTION
1965
1966
1967
1968 ;**
1969 ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
1970 ; THAT ARE USED IN MORE THAN ONE TEST.
1971 ;
1972 ; .SBTTL  SOFINIT  SOFT INITIALIZE OF CONTROLLER
1973
1974 ;*
1975 ;
1976 ; ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
1977 ; BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
1978 ; THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
1979 ; DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
1980 ;
1981 ; INPUTS:
1982 ;
1983 ; R5  ADDRESS OF FIRST REGISTER
1984 ;
1985 ; OUTPUTS:
1986 ;
1987 ; R0  CONTENTS OF TSSR, IF ERROR
1988 ; CARRY SET IF INIT WAS OKAY
1989 ; CLEAR IF FATAL ERROR
1990 ;
1991 ; CALLING SEQUENCE:
1992 ;
1993 ; MOV   #ADDRESS,R5
1994 ; JSR   PC,SOFINIT
1995 ; BCS  CONTINUE
1996 ; ERDF          ;REPORT FATAL ERROR
1997 ;
1998 ;-
1999 016064          SOFINIT::
2000 016064          SAVREG          ; SAVE THE REGISTERS
2001 016070 012765 000000 000000  MOV   #0,TSSR(R5)      ; DO THE INIT.
2002 016076 004737 016340          JSR   PC,WAITF          ; WAIT FOR SSR
2003 016102 016500 000002          MOV   TSSR(R5),R0     ; GET THE TSSR REGISTER
2004 016106 010004          MOV   R0,R4           ; TSSR CONTENTS
2005 016110 042704 176277          BIC   #1<HIADDR!OFL>,R4
2006 016114 052704 002200          BIS   #SSR!NBA,R4    ; R4 HAS EXPECTED CONTENTS
2007 016120 020400          CMP   R4,R0          ; ONLY EXPECTED BITS SET ?
2008 016122 001402          BEQ   S#            ; BRANCH IF OKAY
2009 016124 000241          CLC                   ; CLEAR THE CARRY FOR ERROR
2010 016126 000401          BR   10#           ; GO TO EXIT
2011 016130 000261          SEC                   ; SET THE CARRY BIT
2012 016132 000207          RTS                  ; RETURN TO CALLER
2013          .SBTTL  CHKAMB - CHECK TSSR FOR AMBIGUITY

```

CHKAMB CHECK TSSR FOR AMBIGUITY

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

```

;
;
; THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
; FOR AMBIGUITY
;
; INPUT:
;
;     RO     CONTENTS OF TSSR
;
; OUTPUT:
;
;     RO     CONTENTS OF TSSR
;
;     CARRY  SET   NO AMBIGUITY
;           CLR   AMBIGUOUS CONTENTS
;
;
CHKAMB:
    SAVREG                ;SAVE THE GENERAL REGISTERS
    MOV     RO,R4         ;CONTENTS OF TSSR
    BIT     @SC,RO        ;IS BIT 15 SET ?
    BNE     5$           ;BRANCH IF YES
    BIT     @C<NBA!OFL!SSR!HIADDR>,RO ;ANY OTHER BITS SET ?
    BNE     40$          ;MUST BE AN ERROR
    BR      45$          ;RETURN WITH SUCCESS
    BIT     @SSR,RO       ;IS READY BIT SET ?
    BNE     10$          ;BRANCH IF READY BIT IS SET.
    BIT     @BITS,RO      ;IS FATAL ERROR BIT SET ?
    BEQ     40$          ;ERROR IF NOT
    BIC     @CTERCLS,R4   ;CLEAR ALL BUT TERMINATION CODE
    CMP     R4,#16        ;ALL THREE BITS MUST BE SET
    PNE     40$          ;ERROR IF NOT SET
    BR      45$          ;OK IF ALL ARE SET
    BIT     @BITS,RO      ;IS FATAL ERROR BIT SET ?
    BEQ     45$          ;ERROR IF BIT IS SET WITH SSR
    BIT     @BIT2!BIT1,RO ;IS THIS A FUNCTION REJECT
    BNE     45$          ;BR, IF TSSR IS OK
    CLC     40$          ;AMBIGUOUS CONTENTS
    BR      50$
    SEC     45$          ;SHOW SUCCESS - NO AMBIGUITY
    RTS     50$          ;RETURN TO CALLER
    .SBTTL ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
;
; DEFAULT DISPLAY INTERRUPT HANDLERS.
; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
;
; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
;
;     IOKCKIN=BIT7      ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
;     IOKSTP=BIT0       ; EXPECT "STOP" INTERRUPT.
;
; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
INTMASK: .BYTE 0
    
```

ENAINIT,DSBINT - ENABLE/DISABLE INTERRUPTS

```

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

```

FF,

WAITF WAIT FOR SUBSYSTEM READY

```

2124
2125 016356 100420          BMI      3$          ; EXIT ON STOP FLAG.
2126 016360          DELAY    1          ; WAIT 100 USEC
      016360 012727 000001    MOV     #1,(PC)+
      016364 000000          .WORD   0
      016366 013727 002116    MOV     L$DLY,(PC)+
      016372 000000          .WORD   0
      016374 005367 177772    DEC     -6(PC)
      016400 001375          BNE     --4
      016402 005367 177756    DEC     -22(PC)
      016406 001367          BNE     --20
2127 016410 005316          DEC     (SP)          ;REDUCE DELAY COUNT
2128 016412 001356          BNE     2$          ;RETRY UNTIL TIMER EXPIRES
2129 016414 000241          CLC
2130 016416 000401          BR      4$          ; C = 0, CONTROLLER STILL RUNNING...
2131 016420 000261          3$: SEC          ;...OR HUNG-UP AFTER 300 MSEC.
2132 016422 005326          4$: DEC     (SP)+    ; C = 1, CONTROLLER IS STOPPED.
2133 016424 000207          RTS     PC          ;RESTORE STACK WITHOUT CHANGING CARRY BIT
2134
2135          .SBTTL  CHKTSSR - CHECK TSSR FOR READY
2136
2137          ;*
2138          ; THIS ROUTINE WAITS FOR READY IN THE TSSR
2139          ; AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
2140
2141          ; INPUT:
2142
2143          ;          R5          ADDRESS OF CSR REGISTERS
2144
2145          ; OUTPUT:
2146
2147          ;          R0          CONTENTS OF TSSR
2148          ;          CARRY      SET - OKAY
2149          ;                   CLR - NOT READY AMBIGUOUS, OR SC SET
2150
2151          ;
2152          ;
2153 016426          CHKTSSR:
2154 016426 004737 016340          JSR     PC,WAITF    ;WAIT FOR READY
2155 016432 103014          BCC     20$         ;BRANCH IF TIME OUT
2156 016434 004737 016134          JSR     PC,CHKAMB   ;TSSR AMBIGUOUS?
2157 016440 103006          BCC     10$         ;BR IF YES
2158 016442 032700 100000          BIT     @SC,R0      ;SPECIAL CONDITION SET?
2159 016446 001405          BEQ     15$         ;BR IF NO
2160 016450 032700 074000          BIT     @<SCE!BIE!RMR!NXM>,R0 ;ANY ERROR BITS SET?
2161 016454 001402          BEQ     15$         ;BR IF NO
2162 016456 000241          10$: CLC
2163 016460 000401          BR      20$         ;SET FAILURE
2164 016462 000261          15$: SEC
2165 016464 000207          20$: RTS     PC          ;SET SUCCESS
2166          .SBTTL  XNXM - CHECK FOR NONEXISTENT MEMORY
2167
2168          ;*
2169          ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
2170          ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
2171          ;                   "C" = 0, ALL ADDRESSES OK.
2172          ;
          ;CALL:  MOV ADR1,R1
    
```

XNXM CHECK FOR NONEXISTENT MEMORY

```

2173      ;      MOV ADR2,R2
2174      ;      JSR PC,NXM
2175      ;      RETURN          ;TEST "C" AND PROCEED.
2176      ;
2177 016466 012737 016520 000004 XNXM:  MOV    #2$,#04      ; SET BUSERR VECTOR.
2178 016474 012737 000200 000006      MOV    #PRI04,#06
2179 016502 005003      CLR    R3          ;FLAG.
2180 016504 005711 1$:  TST    (R1)       ;TEST THE ADDRESS(ES).
2181      ;
2182 016506 020102      CMP    R1,R2      ;IF ANY TRAP, CONTINUE AT 2$.
2183 016510 001407      BEQ    3$        ;OTHERWISE, CONTINUE HERE.
2184 016512 062701 000002      ADD    #2,R1     ;BR IF FINISHED (NO NEXM'S).
2185 016516 000772      BR    1$        ;SET NEXT ADDRESS...
2186      ;
2187 016520 005103 2$:  COM    R3          ;GOT ONE, SET FLAG...
2188 016522 012716 016530      MOV    #3$,(SP)
2189 016526 00C002      RTI           ;...AND DISMISS INTERRUPT...
2190 016530 012700 000004 3$:  CLRVEC #4        ;...AND GIVE BACK THE VECTOR.
      016534 104436      MOV    #4,R0
2191 016536 005703      TRAP  C$CVEC
2192 016540 001401      TST    R3        ;DID WE CATCH ONE ??
2193 016542 000261      BEQ    .+4       ;NO, "C" = 0, SKIP NEXT.
2194 016544 000207      SEC           ;YES, "C" = 1, (R1) = NEXM ADDR.
2195      RTS    PC
2196
2197      .SBTTL TSTLOOP CHECK ITERATION COUNT
2198      ;*
2199      ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
2200      ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON ZERO.
2201      ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
2202      ;
2203      ; CALL: LOOPTO ARG
2204      ;
2205 016546 005737 002160 TSTLOOP:;
2206 016546 001006      TST    NOITS     ; ITERATIONS INHIBITED?
2207 016552 005737 002174      BNE    1$        ; YES.
2208 016554 100403      TST    QVP       ; NO.
2209 016560 005337 002206      BMI    1$        ;LOOPS DISALLOWED IN QUICK PASS.
2210 016562 001002      DEC    LOOPCNT  ; BUMP LOOP COUNTER.
2211 016566 000241      BNE    2$
2212 016570 000401 1$:  CLC           ;LOOP DISALLOWED, OR DONE.
2213 016572 000261      BR    3$
2214 016574 000207 2$:  SEC           ;LOOP ENABLED.
2215 016576 000207 3$:  RTS    PC
2216
2217      .SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
2218      ;*
2219      ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
2220      ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
2221      ; IN THE CURRENT RUN SEQUENCE.
2222      ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
2223      ;
2224      ;INPUT:
2225      ;
2226      ;      R0      POINTER TO TEST ID ASCIZ STRING
2227      ;

```


GF

TSTSETUP PRINT TEST NAME AND INIT ERROR COUNTS

```

2228 ;OUTPUT:
2229 ;
2230 ; R5 ADDRESS OF FIRST DEVICE REGISTER
2231 ;
2232 ;IMPLICIT OUTPUTS:
2233 ;
2234 ; TSTCNT UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
2235 ;
2236 ;SIDE EFFECTS:
2237 ;
2238 ; INTERRUPT LEVEL IS RASIED TO LEVEL OF
2239 ; THE DEVICE UNDER TEST
2240 ;
2241 ;-
2242
2243 016600 TSTSETUP::
2244 016600 01C046 MOV R0,-(SP) ;SAVE THE TEST ID MESSAGE
2245 016602 005037 003144 CLR SIFLAG ; CLEAR "SOFT INIT" FLAG
2246 016606 005037 017046 CLR ERRK ; CLEAR LOCAL ERROR COUNTER.
2247 016612 005037 005770 CLR EXTA ; CLEAR ERROR EXTENSION FLAG.
2248 016616 105037 016234 CLRB INTMASK ; CLEAR INTERRUPT MASK (CHECK ERROR)
2249 016622 013700 002172 MOV UNITN,R0 ; GET THE UNIT NUMBER,
2250 016626 006300 ASL R0 ; ... AND MAKE IT A WORD OFFSET.
2251 016630 005737 003104 TST NODEV ; DID STARTUP FIND THE DEVICE?
2252 016634 001430 BEQ 4$ ; BR IF YES
2253 016636 100010 BPL 3$ ; BR IF NOT IDLE
2254 016640 052760 160000 003166 BIS #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
2255 016646 ERRDF 1,NXR,NXRERR ; NO DEVICE HERE -- PRINT IT
    016646 104455 TRAP C$ERRDF
    016650 000001 .WORD 1
    016652 003736 .WORD NXR
    016654 005734 .WORD NXRERR
2256 016656 000407 BR 2$
2257 016660 052760 160001 003166 3$: BIS #160001,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
2258 016666 ERRDF 2,NOINIT ; DEVICE NOT IDLE
    016666 104455 TRAP C$ERRDF
    016670 000002 .WORD 2
    016672 004333 .WORD NOINIT
    016674 000000 .WORD 0
2259 016676 012737 177777 003102 2$: MOV #-1,DUFLG ; DROP THE UNIT
2260 016704 DODU UNITN
    016704 013700 002172 MOV UNITN,R0
    016710 104451 TRAP C$DODU
2261 016712 DOCLN ; ABORT THE PASS
    016712 104444 TRAP C$DCLN
2262 016714 000423 BR 5$
2263
2264 016716 4$: RFLAGS R0 ; GET THE OPERATOR FLAGS.
    016716 104421 TRAP C$RFLA
2265 016720 032700 001000 BIT #PNT,R0 ; PRINT THE TEST NUMBERS?
2266 016724 001412 BEQ 1$ ; BR IF NO
2267 016726 011600 MOV (SP),R0 ;GET THE ID MESSAGE
2268 016730 PRINTF #TNAM,R0 ;DISPLAY THE TEST ID
    016730 010046 MOV R0,-(SP)
    016732 012746 016774 MOV #TNAM,-(SP)
    016736 012746 000002 MOV #2,-(SP)
    016742 010600 MOV SP,R0

```

TSTSETUP PRINT TEST NAME AND INIT ERROR COUNTS

```

016744 104417          TRAP  C$PNTF
016746 062706 000006  ADD   #6,SP
2269 016752 005237 002204 1$:   INC   TSTCNT          ; BUMP TEST COUNTER.
2270 016756          SETPRI IPRI             ; PRIORITY THAT OF DEVICE
016756 013700 002202  MOV   IPRI,RO
016762 104441          TRAP  C$SPRI
2271 016764 005726 5$:   TST   (SP),          ; FIX UP THE STACK
2272 016766 013705 002176  MOV   CSRADDR,R5     ; ADDRESS OF TSV REGISTERS ON UNIBUS
2273 016772 000207  RTS   PC
2274 016774 045      123  045  TNAM:  .ASCIZ  'S#T#A Test'
2275          .EVEN
2276          .SBTTL  TSTEND  PRINT ERRORS RECEIVED
2277          ;
2278          ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2279          ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2280          ;
2281 017010          TSTEND: RFLAGS  RO
017010 104421          TRAP  C$RFLA
2282 017012 030027 020000  BIT   RO,#IER
2283 017016 001412          BEQ   1$             ; BR IF "IER" NOT SET.
2284 017020          PRINTF #ESUM,ERRK          ; PRINT ERROR COUNT.
017020 013746 017046  MOV   ERRK,(SP)
017024 012746 017050  MOV   #ESUM,-(SP)
017030 012746 000002  MOV   #2,-(SP)
017034 010600          MOV   SP,RO
017036 104417          TRAP  C$PNTF
017040 062706 000006  ADD   #6,SP
2285 017044 000207 1$:   RTS   PC
2286
2287 017046 000000          ERRK:  0             ; LOCAL ERROR COUNT.
2288 017050 045      101  040  ESUM:  .ASCIZ  /#A #D#A ERRORS/
2289 017067 105      122  122  EMAXDU: .ASCIZ  /ERROR LIMIT REACHED -- DROPPING UNIT/
2290          .EVEN
2291          ;
2292          .SBTTL  INCERK  - INCREMENT LOCAL ERROR COUNT
2293          ;
2294          ; *
2295          ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
2296          ; -
2296 017134 005237 017046  INCERK: INC  ERRK          ; INCREMENT LOCAL ERROR COUNT
2297 017140 010046          MOV   RO,-(SP)         ; SAVE RO
2298 017142 013700 002172  MUV   UNITN,RO        ; GET UNIT NUMBER,
2299 017146 006300          ASL   RO              ; ... AND MAKE IT A WORD OFFSET.
2300 017150 062700 003166  ADD   #ERTABL,RO      ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
2301 017154 005210          INC   (RO)           ; INCREMENT THE DEVICE ERROR COUNT
2302 017156 032710 007777  BIT   #7777,(RO)     ; DID WE OVERFLOW THE FIELD?
2303 017162 001001          BNE  1$             ; BR IF NO.
2304 017164 005310          DEC   (RO)          ; YES -- BACK IT UP TO 7777.
2305 017166 012600 1$:   MOV   (SP)+,RO        ; RESTORE RO
2306 017170 000207          RTS   PC            ; RETURN TO CALLER.
2307
2308 017172 010046          CKEMAX: MOV  RO,-(SP)     ; SAVE RO
2309 017174 013700 002172  MOV   UNITN,RO        ; GET UNIT NUMBER
2310 017200 006300          ASL   RO              ; ... AND MAKE IT A WORD OFFSET
2311 017202 016000 003166  MOV   ERTABL(RO),RO   ; GET ERROR TABLE ENTRY
2312 017206 042700 170000  BIC   #170000,RO      ; EXTRACT ERROR COUNT FIELD
2313 017212 020037 002164  CMP   RO,GERRMAX     ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
2314 017216 103004          BHS  1$             ; BR IF YES

```

Tr,

INCERK INCREMENT LOCAL ERROR COUNT

```

2315 C17220 023737 017046 002162      CMP      ERRK,LERRMAX      ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2316 017226 103417                    BLO      2$                ; BR IF NO
2317 017230                    1$:    RFLAGS  RO                ; GET OPERATOR FLAGS
      017230 104421                    TRAP    C$RFLA
2318 017232 032700 000040            BIT      @IDU,RO           ; IS DROPPING INHIBITED?
2319 017236 001013                    BNE      2$                ; BR IF YES.
2320 017240 012737 177777 003102      MOV      @-1,DUFLG        ; NO DROP THE UNIT
2321 017246                    ERRDF   4,EMAXDU
      017246 104455                    TRAP    C$ERDF
      017250 000004                    .WORD  4
      017252 017067                    .WORD  EMAXDU
      017254 000000                    .WORD  0
2322 017256                    DODU    UNITN
      017256 013700 002172            MOV      UNITN,RO
      017262 104451                    TRAP    C$DODU
2323 017264                    DOCLN
      017264 104444                    TRAP    C$DCLN
2324 017266 012600                    2$:    MOV      (SP)+,RO      ; RESTORE RO
2325 017270 000207                    RTS      PC                ; RETURN TO CALLER
2326                    .SBTTL  CKDROP - CHECK IF UNIT SHOULD BE DROPPED
2327                    ;
2328                    ; CHECK IF UNIT SHOULD BE DROPPED
2329                    ;
2330 017272 010046                    CKDROP: MOV      RO,-(SP)
2331 017274                    FORCERROR 1$,NOTSSR
2332 017304                    RFLAGS  RO
      017304 104421                    TRAP    C$RFLA
2333 017306 032700 000040            BIT      @IDU,RO
2334 017312 001010                    BNE      1$
2335 017314 011600                    MOV      (SP),RO
2336 017316 012737 177777 003102      MOV      @-1,DUFLG
2337 017324                    DODU    UNITN
      017324 013700 002172            MOV      UNITN,RO
      017330 104451                    TRAP    C$DODU
2338 017332                    DOCLN
      017332 104444                    TRAP    C$DCLN                ;ABORT THE PASS
2339 017334 012600                    1$:    MOV      (SP)+,RO
2340 017336 000207                    RTS      PC
2341
2342
2343                    .SBTTL  CONFIG - DETERMINE CONFIGURATION OF SYSTEM
2344                    ;
2345                    ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
2346                    ;
2347 017340                    CONFIG:
2348 017340 004737 016064                JSR      PC,SOFINIT
2349 017344 000207                    RTS      PC
2350                    .SBTTL  KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
2351                    ;
2352                    ; SUBROUTINE  ENABLE MEM MGT.
2353                    ;
2354 017346 005737 003122                KTON:   TST      @TFLG      ; GOT KT?
2355 017352 001403                    BEQ      1$                ; NO.
2356 017354 012737 000001 177572      MOV      @1,SRO           ; YES. ENABLE KT11.
2357 017362 000207                    1$:    RTS      PC
2358
2359                    ;

```

KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT

```

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

```

FILLMEM FILL MEMORY WITH BACKGROUND PATTERN

```

2417 ; OUTPUTS:
2418 ;
2419 ; NONE
2420 ; -
2421 ;
2422 FILLMEM:
2423 SAVREG ;SAVE R1 R5 UNTIL NEXT RETURN
2424 JSR PC,KTOFF ;DISABLE KT.
2425 MOV R0,R3 ;COPY TEST PATTERN
2426 MOV FREE,R1 ;GET FIRST FREE LOCATION
2427 MOV FRESIZ,R2 ;SIZE OF FREE SPACE BELOW 28K.
2428 10$: MOV R3,(R1)+ ;STORE A BACKGROUND WORD
2429 DEC R2 ;DONE ALL MEMORY IN FREE SPACE?
2430 BGT 10$ ;BR IF NO
2431 TST KIFLG ; GOT KT?
2432 BEQ 55$ ; NO. GET OUT.
2433 JSR PC,KTON ; YES. ENABLE KT.
2434 CLR R0 ;HIGH ORDER ADDRESS START
2435 MOV PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
2436 .REPT 6
2437 CLC ;CLEAR C BIT
2438 ROL R1 ;CONVERT BLOCKS TO WORDS
2439 ROL R0 ;MAKE IT DOUBLE PRECISION
2440 .ENDR
2441 JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
2442 30$: MOV R3,(R0)+ ;STORE TEST PATTERN IN >28K ADDRESS
2443 CMP R0,#160000 ;END OF PAR6 MAPPING AREA?
2444 BLO 30$ ;BR IF NO
2445 SUB #20000,R0 ;BACKUP INTO PAR6 MAPPING BEGIN
2446 ADD #200,#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2447 CMP #KIPAR6,KIFLG ;END OF MEMORY?
2448 BEQ 50$ ;BR IF YES
2449 TST T23A ;11/23A?
2450 BEQ 35$ ;NO KEEP GOING
2451 MOV SRO,R4 ;GET SRO CONTENTS
2452 BIC #17761,R4 ;CLEAR ALL BUT PAGE NUMBER
2453 CMP #16,R4 ;SEE IF PAGE 7
2454 BEQ 50$ ;EXIT IF THERE
2455 35$: TST T23B ;11/23B?
2456 BEQ 45$ ;NO KEEP GOING
2457 CMP #KIPAR6,#7600 ;REACHED 18 BITS?
2458 BHIS 40$ ;YES
2459 BR 45$ ;NO KEEP GOING
2460 40$: MOV #20,SR3 ;SET 22 BIT RELOCATION
2461 45$: JMP 30$ ;KEEP GOING ON ETC.
2462 50$: JSR PC,KTOFF ; DISABLE KT.
2463 55$: RTS PC
2464 .SBTTL CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
2465 ;*
2466 ; COMPARE MEMORY WITH A BACKGROUND PATTERN
2467 ;
2468 ; INPUTS:
2469 ;
2470 ; RO = BACKGROUND PATTERN
2471 ; FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2472 ; KIFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2473 ;

```

CMPMEM COMPARE MEMORY TO BACKGROUND PATTERN

```

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

```

CMPMEM COMPARE MEMORY TO BACKGROUND PATTERN

```

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

```

GETPAT GET 8 BIT PATTERN FROM OPERATOR

```

2588
2589
2590
2591
2592
2593 020264
2594 020264
2595 020270
020270 104443
020272 000406
020274 020320
020276 000022
020300 020322
020302 000377
020304 000000
020306 000377
020310
2596 020310
020310 103367 020320
2597 020312 013700 020320
2598 020316 000207
2599
2600
2601
2602
2603
2604 020320 000000
2605 020322 105 116 124
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624 020346
2625 020346
2626 020352 010002
2627 020354 010203
2628 020356 005713
2629 020360 001412
2630 020362
020362 012346
020364 012746 020532
020370 012746 000002
020374 010600

;
; JSR PC,GETPAT
;
;
GETPAT::
; SAVREG ;SAVE THE GENERAL REGISTERS
1$: GMANID DATASC,PATDAT,0,377,0,377,NO
TRAP C$GMAN
BR 10000$
.WORD PATDAT
.WORD T$CODE
.WORD DATASC
.WORD 377
.WORD T$LLOLIM
.WORD T$HILIM
10000$:
BNCOMPLETE 1$ ;RETRY IF ERROR
BCC 1$
MOV PATDAT,RO ;DATA PATTERN FROM OPERATOR
RTS PC ;RETURN TO CALLER

;+
;LOCAL DATA AREA
;-
PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
DATASC: .ASCIZ 'ENTER DATA PATTERN'
.EVEN
.SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

;+
;ROUTINE TO ISSUE A MENU AND GET
;THE OPERATOR'S RESPONSE.
;
;INPUTS:
;
; RO ADDRESS OF ASCIZ STRING OF MENU
; R1 MAXIMUM ALLOWABLE OPERATOR RESPONSE
;
;OUTPUTS:
;
; RO NUMBER OF THE OPERATOR'S SELECTION
;-

GETSEL::
; SAVREG ;SAVE GENERAL REGISTERS
MOV R0,R2 ;SAVE THE MENU ADDRESS
1$: MOV R2,R3 ;START OF MENU STRING
2$: TST (R3) ;END OF ASCII ?
BEQ 3$ ;BRANCH IF ALL LINES DISPLAYED
PRINTF #SELASC,(R3)+ ;DISPLAY THE MENU
MOV (R3)+,-(SP)
MOV #SELASC,-(SP)
MOV #2,-(SP)
MOV SP,R0

```


GETSEL ISSUE MENU AND GET OPERATOR RESPONSE

```

020376 104417 TRAP C$PNTF
020400 062706 000006 ADD #6,SP
2631 020404 000764 BR 2$
2632 020406 3$: G$MANID MENASC,MENRES,D,1,0,1,NC
020406 104443 TRAP C$G$MAN
020410 000406 BR 10001$
020412 020566 .WORD MENRES
020414 000042 .WORD T$CODE
020416 020537 .WORD MENASC
020420 177777 .WORD -1
020422 000000 .WORD T$LQIM
020424 177777 .WORD T$HILIM
020426 10001$:
2633 020426 B$NCOMPLETE 1$ ;RETRY IF ERROR
020426 103352 BCC 1$
2634 020430 013700 020566 MOV MENRES,R0 ;GET THE OPERATOR'S REPLY
2635 020434 020001 CMP R0,R1 ;COMPARE TO MAXIMUM ALLOWED
2636 020436 101411 BLOS 5$ ;BRANCH IF OK
2637 020440 PRINTF #MENERR ;DISPLAY ERROR MESSAGE
020440 012746 020464 MOV #MENERR,-(SP)
020444 012746 000001 MOV #1,-(SP)
020450 010600 MOV SP,R0
020452 104417 TRAP C$PNTF
020454 062706 000004 ADD #4,SP
2638 020460 000735 BR 1$ ;RETRY
2639 020462 000207 RTS PC ;RETURN TO CALLER
2640 020464 045 116 045 MENERR: .ASCIZ 'M$A *** Menu Selection Too Large ***'
2641 020532 045 116 045 SELASC: .ASCIZ 'M$T'
2642 020537 105 156 164 MENASC: .ASCIZ 'Enter Menu Selection: '
2643 .EVEN
2644 020566 000000 MENRES: .WORD 0
2645 .SBTTL CH$KMAN CHECK MANUAL INTERVENTION LEGALITY
2646 ;
2647 ;
2648 ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2649 ;
2650 ;INPUT:
2651 ;
2652 ; NONE.
2653 ;
2654 ;OUTPUT:
2655 ;
2656 ; CARRY 0 MANUAL INTERVENTION NOT ALLOWED
2657 ; 1 MANUAL INTERVENTION IS OK
2658 ;
2659 ;SIDE EFFECTS:
2660 ;
2661 ; A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2662 ; NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2663 ; ALLOWED.
2664 ;
2665 ;-
2666 ;
2667 020570 CH$KMAN: SAVREG ;SAVE THE REGISTERS
2668 020570 MANUAL ;SEE IF MANUAL INTERVENTION OK
2669 020574 104450 TRAP C$MANI

```

CHKMAN CHECK MANUAL INTERVENTION 'EGALITY

```

2670 020576          RCOMPLETE 1$          ;BRANCH IF ALLOWED
      020576 103411  BCS 1$
2671 020600          PRINTF #NOMAN          ;PRINT THE WARNING MESSAGE
      020600 012746 020624  MOV #NOMAN, -(SP)
      020604 012746 000001  MOV #1, (SP)
      020610 010600  MOV SP,R0
      020612 104417  TRAP C$PNTF
      020614 062706 000C04  ADD #4,SP
2672 020620 000241  CLC          ;CLEAR CARRY FOR ERROR
2673 020622 000207  1$: RTS PC          ;RETURN
2674
2675 020624 045 116 045  NOMAN: .ASCIIZ '#NSA *** Manual Intervention not Allowed Test Aborted ***'
2676 .even
2677 .SBTTL ENVIRN SETUP FREE DIAGNOSTIC SPACE
2678 ;
2679 ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
2680 ;
2681 020720  ENVIRN: MEMORY R0
      020720 104431  TRAP C$MEM
2682 020722 010037 003114  MOV R0 FREE          ; GET 1ST FREE ADDRESS...
2683 020726 062737 000002 003114  ADD #2,FREE
2684 020734 011037 003116  MOV (R0),FRESIZ      ;...AND WORD COUNT.
2685 020740 162737 000004 003116  SUB #4,FRESIZ
2686 020746 013702 002012  MOV L$UNIT,R2        ; GET NUMBER OF UNITS
2687 020752 162737 000007 003116 10$: SUB #7,FRESIZ      ; TAKE AWAY 7 WORDS PER UNIT
2688 020760 005302  DEC R2
2689 020762 001373  BNE 10$
2690 020764 013700 003114  MOV FREE,R0          ;GET FIRST FREE ADDRESS
2691 020770 063700 003116  ADD FRESIZ,R0        ;POINT TO LAST FREE ADDRESS
2692 020774 162700 000002  SUB #2,R0             ;BACKUP 1 WORD
2693 021000 010037 003120  MOV R0,FREEM        ;STORE LAST FREE ADDRESS
2694 021004 000240  NOP
2695 021006 012701 177520  MOV #BDVPCR,R1      ;GET BDV11 PCR ADDRESS
2696 021012 010102  MOV R1,R2            ;COPY TO R2
2697 021014 062702 000002  ADD #2,R2            ;SET THE RANGE
2698 021020 004737 016466  JSR PC,XNXM        ;SEE IF WE HAVE ONE
2699 021024 103001  BCC 15$            ;OK TO SET FLAGS
2700 021026 000445  BR 40$            ;RETURN WITH FLAGS CLEAR
2701 021030 013701 177520 15$: MOV BDVPCR,R1      ;SAVE PCR CONTENTS
2702 021034 062701 000001  ADD #1,R1           ;ADD ONE TO IT
2703 021040 012702 177520  MOV #BDVPCR,R2      ;GET BDV11 PCR ADDRESS
2704 021044 005212  INC (R2)            ;TRY TO WRITE TO IT
2705 021046 013703 177520  MOV BDVPCR,R3      ;GET RESULTS
2706 021052 020103  CMP R1,R3           ;DID IT CHANGE?
2707 021054 001017  BNE 20$            ;NO, MUST BE 11/238
2708 021056 005237 003134  INC T23A           ;SET THE FLAG
2709 021062 042737 170000 002120  BIC #170000,L$HIME ;SUPERVISOR COULD BE WRONG
2710 021070 000240  NOP
2711 021072  PRINTF #M8186          ;TELL THE SYSTEM TYPE
      021072 012746 005552  MOV #M8186, -(SP)
      021076 012746 000001  MOV #1, -(SP)
      021102 010600  MOV SP,R0
      021104 104417  TRAP C$PNTF
      021106 062706 000004  ADD #4,SP
2712 021112 000413  BR 40$            ;RETURN
2713 021114 005237 003136 20$: INC T23B          ;SET THE FLAG
2714 021120 000240  NOP          ;BR 40$ FOR RELEASE

```

D7

CE2 0091

ENVIRN SETUP FREE DIAGNOSTIC SPACE

```

2715 021122          PRINTF  #M8189          ; TELL THE SYSTEM TYPE
      021122 012746 005643  MOV      #M8189, (SP)
      021126 012746 000001  MOV      #1, (SP)
      021132 010600          MOV      SP,RO
      021134 104417          TRAP     C$PNTF
      021136 062706 000004  ADD      #4,SP
2716 021142 000207          40$:  RTS      PC          ; RETURN
2717          .SBTTL  KTINIT  SETUP KT11 MEMORY MANAGEMENT REGISTERS
2718          ;*
2719          ;
2720          ;ROUTINE TO INIT KT 11
2721          ;
2722          ;
2723          ;
2724 021144          KTINIT:
2725 021144 005037 003122  CLR      KTFLG          ; INIT >28K MEMORY FLAG
2726 021150 005037 003124  CLR      KTENABLE       ; INIT TEST >28K FLAG
2727 021154 023727 002120 001577  CMP      L$HIME,#1577   ; GOT ENOUGH MEMORY (>28K)?
2728 021162 101444          BLOS    9$              ; NO.
2729 021164 013700 000004  MOV      @ERRVEC,RO    ; SAVE OLD ERR VEC PTR.
2730 021170 012737 021262 000004  MOV      #2,@ERRVEC   ; SET ERR VEC PTR.
2731 021176 005737 177572  TST     @SRO           ; GOT KT11?
2732 021202 000240          NOP                    ; (TRAP IF NO).
2733 021204 013737 002120 003122  MOV      L$HIME,KTFLG ; YES. SET KT FLAG.
2734 021212 042737 000177 003122  BIC     #177,KTFLG    ;
2735 021220 010037 000004  MOV      RO,@ERRVEC   ; RESTORE OLD ERR VEC PTR.
2736 021224 005000          CLR      RO            ; RO = AR DATA.
2737 021226 012701 172340  MOV      #KIPAR0,R1   ; R1 = KI REGS PTR.
2738 021232 012761 077406 177740 1$:  MOV      #77406,-40(R1) ; SET DESCRIPTOR REG.
2739 021240 010021          MOV      RO,(R1)+     ; SET KIPAR REG.
2740 021242 062700 000200  ADD     #200,RO       ; BUMP AR DATA BY "4K".
2741 021246 020027 002000  CMP     RO,#2000      ; AT "I/O"?
2742 021252 001367          BNE    1$              ; NO.
2743 021254 012741 177600  MOV     #177600,-(R1) ; YES. SET KIPAR7 FOR I/O.
2744 021260 000405          BR     9$              ;
2745          ;
2746 021262 012716 021270 2$:  MOV     #6,(SP)        ; SET UP RETURN
2747 021266 000002          RTI                    ; RTI TO NEXT LOCATION
2748          ;
2749 021270 010037 000004 6$:  MOV     RO,@ERRVEC    ; RESTORE OLD ERR VEC PTR.
2750          ;
2751 021274 000207          9$:  RTS     PC
2752          ;*
2753          ;
2754          ;
2755          ;
2756          ;
2757          ;
2758          ;
2759          ;INPUTS:
2760          ;      R5      CURRENT UNIT NUMBER
2761          ;OUTPUTS:
2762          ;      The Extended Features Switch is set.
2763          ;
2764          ;-
2765 021276          INVERT::
2766          ;

```

KTINIT SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

2767 021276 005737 002216          TST      EXTFEA          ; IS SWITCH SET?
2768 021302 001020          BNE      1$            ; YES,EXIT STAGE RIGHT!(on the next one outa town!)
2769 021304 012737 100206 021350  MOV      @100206,CMDPKT ; WRT SUB SYS MEM CMD
2770 021312 012737 021360 021352  MOV      @WSMBK,CMDPKT+2 ; MSG BUF ADDR
2771 021320 012737 000006 021356  MOV      @6,CMDPKT+6    ; BYTE COUNT
2772 021326 012737 100010 021360  MOV      @100010,WSMBK ; INVERT THE SWITCH
2773 021334 012704 021350          MOV      @CMDPKT,R4    ; SET CMDPKT INTO R4
2774 021340 004737 010752          JSR      PC,WRTCHR    ; DO IT
2775 021344 000207          1$:     RTS          PC ; RETURN
2776
2777          ;          COMMAND PACKET.
2778
2779          021350          .          "          <.,+3>&177774 ;MUST BE ON MOD 4 BOUNDRY.
2780
2781 021350 000000          CMDPKT:: 0          ;1ST WORD IS TS05 COMMAND.
2782 021352 000000          0          ;2ND WORD IS THE BUFFER LOW ADDRESS.
2783 021354 000000          0          ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2784 021356 000000          0          ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2785
2786          ;          WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
2787
2788 021360 000000          WSMBK:: 0          ;1ST WORD:: SEL 0
2789 021362 000000          0          ;2ND WORD:: SEL 2
2790 021364 000000          0          ;3RD WORD:: SEL 4
2791          .EVEN
2792
2793          ;          SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
2794          ;
2795          ;
2796          ;INPUTS:
2797          ;OUTPUTS:
2798          ;          The NXMFLG is set if we can test.
2799          ;          The NXML0 and NXMHI addresses are setup.
2800          ;
2801          ;
2802 021366          MEMCK::
2803
2804 021366          SAVREG          ;SAVE THE REGISTERS
2805 021372 005037 003126          CLR      NXMFLG        ;CLEAR THE FLAG
2806 021376 005037 003130          CLR      NXML0        ;CLEAR THE TEST ADDRESS LO
2807 021402 005037 003132          CLR      NXMHI        ;CLEAR THE TEST ADDRESS HI
2808 021406 005737 003136          TST      T23B         ;IS IT A 11/23B?
2809 021412 001407          BEQ      1$            ;NO
2810 021414 023727 002120 007777  CMP      L$HIME,@7777  ; GREATER THAN 128K
2811 021422 103406          BLO      2$            ; NO
2812 021424 004737 021542          JSR      PC,NXMTST    ;SETUP THE ADDRESS
2813 021430 000427          BR       13$          ;SET THE FLAG AND EXIT
2814 021432 005737 003134          1$:     TST      T23A         ;IS IT A 11/23A?
2815 021436 001413          BEQ      4$            ;NO
2816 021440 023727 002120 005777  2$:     CMP      L$HIME,@5777 ;GREATER THAN 96K
2817 021446 101023          BHI      14$          ;YES,23A/23B WITH 128K MEMORY
2818 021450 023727 002120 003777  CMP      L$HIME,@3777  ;GREATER THAN 64K BUT LESS THAN 92K?
2819 021456 103403          BLO      4$            ;NO, CHECK 24K
2820 021460 004737 021542          JSR      PC,NXMTST    ;SETUP THE ADDRESS
2821 021464 000411          BR       13$          ;SET THE FLAG AND EXIT
2822 021466 023727 002120 001577  4$:     CMP      L$HIME,@1577  ;GREATER THAN 24K BUT LESS THAN 64K?
2823 021474 103410          BLO      14$          ;NO, TELL THEM AND EXIT WITH FLAG CLEAR

```

KTINIT SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

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

```

67

KTINIT SETUP KT11 MEMORY MANAGEMENT REGISTERS

7
8
9 021636
021636
10
16

.TITLE TSV4 - MISCELLANEOUS SECTIONS

BGNMOD TSV4

TSV4::

H/

SEQ 0095

PROTECTION TABLE

18						.SBTTL PROTECTION TABLE
19	021636					BGNPROT
	021636					L\$PROT::
20	021636	177777	177777	177777		.WORD 1. 1. 1. 1
21	021646					ENDPROT
22						

;NO DEVICE PROTECTION REQUIRED.

INITIALIZE SECTION

```

24                                     .SBTTL INITIALIZE SECTION
25
26                                     ;**
27                                     ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
28                                     ;AT THE BEGINNING OF EACH PASS.
29
30                                     ;
31                                     ;IF "START" OR "RESTART", SET QUICK PASS FLAG AND BUS-INIT.
32                                     ;IF "CONTINUE", NOTHING IS REQUIRED.
33                                     ;
34                                     ;--
35                                     ;*
36                                     ;INSERT TEMPORARY JUMP TO ODT
37 021646                               ;-
38                                     BGNINIT
39 021646                               L$INIT::
40
41 021646                               SETVEC  #140,#170000,#340          ;ODT ROM ADDRESS
42 021646 012746 000340                 MOV    #340,(SP)
43 021652 012746 170000                 MOV    #170000,(SP)
44 021656 012746 000140                 MOV    #140,(SP)
45 021662 012746 000003                 MOV    #3,(SP)
46 021666 104437                        TRAP   C$SVEC
47 021670 062706 000010                 ADD    #10,SP
48
49 021674 005037 002216                 40$:  CLR   EXTFEA
50 021700 005037 003126                 CLR   NXMFLG
51 021704 012737 006356 002170         MOV   #EPRT1,EPRTSW          ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
52 021712 005037 003144                 CLR   SIFLAG                ;CLEAR SOFT INIT' FLAG
53 021716 005037 003124                 CLR   KTENABLE              ;CLEAR TEST ABOVE 28K FLAG
54 021722 005037 002272                 CLR   RAMSIZ                ;CLEAR RAM SIZE FOR RAMERR ROUTINE
55 021726
56 021726 012700 000036                 READEF #EF.CONTINUE
57 021732 104447                        MOV   #EF.CONTINUE,R0
58 021734
59 021734                               TRAP  C$REFG
60 021734                               BNCOMPLETE 1$
61 021736 103023                        BCC   1$
62 021736 023737 002172 002012         CMP   UNITN,L$UNIT          ;UNIT IN RANGE?
63 021744 103070                        BHIS  4$                    ;BR IF NO.
64 021746 005737 003102                 TST  DUFLG                  ;DROPPED UNIT?
65 021752 100472                        BMI  NXTU                   ;BR IF YES
66 021754 013701 002172                 MOV   UNITN,R1
67 021760 006301                        ASL   R1
68 021762 005761 003166                 TST  ERTABL(R1)
69 021766 001516                        BEQ   SETU
70 021770 032761 040000 003166         BIT   #BIT14,ERTABL(R1)    ;DROPPED?
71 021776 001060                        BNE  NXTU
72 022000
73 022000                               EXIT  INIT                  ;DO NOTHING IF "CONTINUE".
74 022000 104432                        TRAP  C$EXIT
75 022002 000416                        .WORD L10030-.
76 022004
77 022004                               1$:  READEF #EF.NEW
78 022004 012700 000035                 MOV   #EF.NEW,R0
79 022010 104447                        TRAP  C$REFG
80 022012
81 022012                               BNCOMPLETE NXTU            ;TAKE NEXT UNIT IF NOT NEW PASS.
82 022012 103052                        BCC  NXTU
83 022014
84 022014                               READEF #EF.START
85 022014 012700 000040                 MOV   #EF.START,R0
86 022020 104447                        TRAP  C$REFG
87 022022                               BCOMPLETE 2$

```


INITIALIZE SECTION

```

022022 103404      BCS      2$
64 022024      REAFDEF  #EF.RESTART
022024 012700 000037  MOV      #EF.RESTART,RO
022030 104447      TRAP     C$REFG
65 022032      BNCOMPLETE 31$
022032 103031      BCC      31$
66 022034      2$:
67 022034      BRESET
022034 104433      TRAP     C$RESET
68 022036 005037 002204  CLR      TSTCNT      ;NUMBER OF TESTS RUN IN PASS
69 022042 005037 002212  CLR      FATFLG     ;CLEAR FATAL ERROR COUNT
70 022046 005037 003134  C_R     T23A        ;CLEAR 11/23A FLAG
71 022052 005037 003136  CLR      T23B        ;CLEAR 11/23B FLAG
72      ;
73      ;
74      ;
75 022056 005037 003370  CLR      SKIPT      ;;@ENTER THE DEBUGGER
76 022062      20$:
77 022062 012737 177777 002174  MOV      # 1,QVP     ;...QUICK VERIFY...
78 022070 004737 020720      JSR      PC,ENVIRN   ;SET ENVIRONMENT.
79 022074 004737 021144      JSR      PC,KTINIT   ;INITIALIZE KT MEMORY MANAGEMENT
80 022100 012700 003166      MOV      #ERTABL,RO
81 022104 005020      CLR      (RO)+      ;CLEAR THE ERROR TABLE
82 022106 020027 003366  CMP      RO,#ERTABE
83 022112 103774      BLO     30$
84 022114 000404      BR      4$
85 022116 005037 002174  CLR      QVP
86 022122 000137 022172  JMP      PASRPT     ;GO REPORT THE STATUS
87
88 022126      4$:
89 022126 012737 177777 002172  NEWPAS: MOV      #-1,UNITN   ;INIT UNIT NUMBER...
90 022134 005037 002210      CLR      DEVCNT     ;CLEAR COUNT OF DEVICES RUNNING
91 022140      NXTU:
022140 104422      BREAK
92 022142 005237 002172      TRAP     C$BRK
93 022146 023737 002172 002012  INC      UNITN      ;...AND SET NEXT UNIT NUMBER.
94 022154 103423      CMP      UNITN,L$UNIT
95 022156 012737 177777 003102  BLO     SETU
96 022164 000401      MOV      #-1,DUFLG
97 022166      BR      11$
022166 104444      DOCLN
98 022170 000240      TRAP     C$DCLN
99 022172      11$:
100 022172 023727 002012 000001  PASRPT: CMP      L$UNIT,#1   ;HOW MANY UNITS SELECTED?
101 022200 101752      BLOS     NEWPAS     ;BR IF ONLY 1
102 022202 005737 002210      TST      DEVCNT     ;ARE ANY STILL RUNNING?
103 022206 001747      BEQ     NEWPAS     ;BR IF NO
104 022210      RFLAGS  RO
022210 104421      TRAP     C$RFLA
105 022212 032700 000100  BIT      #ISR,RO    ;SHOULD WE PRINT STATISTICS
106 022216 001343      BNE     NEWPAS     ;BR IF NO
107
108 022220      DORPT
022220 104424      TRAP     C$DRPT
109 022222 000741      BR      NEWPAS
110 022224      10$:
111

```

INITIALIZE SECTION

```

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

```

INITIALIZE SECTION

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

ADD AND DROP UNITS SECTION

.SBTTL ADD AND DROP UNITS SECTIONS

```

163
164
165
166
167
168
169
170 022470
    022470
171 022470 010001
172 022472 006301
173 022474 052761 100000 003166
174 022502 042761 040000 003166
175 022510
    022510 010046
    022512 012746 022536
    022516 012746 000002
    022522 010600
    022524 104417
    022526 062706 000006
176 022532
    022532 000167
    022534 000026
177 022536 045 116 045 1$:
178
179
180 022564
    022564
    022564 104452
181
182
183
184
185
186
187
188
189
190
191
192 022566
    022566
193 022566 012737 177777 003102
194 022574 010001
195 022576 006301
196 022600 052761 140000 003166
197 022606 000240 000240 000240
198 022614
    022614 010046
    022616 012746 022642
    022622 012746 000002
    022626 010600
    022630 104417
    022632 062706 000006
199 022636
    022636 000167
    022640 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-.

```

ADD AND DROP UNITS SECTIONS

```

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

```

CLEAN-UP AND REPORT CODING SECTIONS

.SBITI CLEAN-UP AND REPORT CODING SECTIONS

```

218
219
220
221
222
223
224
225 022754
    022754
226 022754 013705 002176
227 022760 005737 003102
228 022764 100405
229
230
231 022766 012765 000000 000002
232 022774 004737 016340
233 023000
234 023000
    023000
    023000 104412
235
236
237
238
239 023002
    023002
240 023002
    023002 012746 023244
    023006 012746 000001
    023012 010600
    023014 104416
    023016 062706 000004
241 023022 010246
242 023024 010346
243 023026 010446
244 023030 012704 003166
245 023034 005003
246 023036 011402
247 023040 001467
248 023042 100066
249 023044 032702 040000
250 023050 001015
251 023052 042702 170000
252 023056
    023056 010246
    023060 010346
    023062 012746 023301
    023066 012746 000003
    023072 010600
    023074 104416
    023076 062706 000010
253 023102 000446
254 023104 020227 160000
255 023110 001012
256 023112
    023112 010346
    023114 012746 023351

    ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
    ; EXECUTED AT THE END OF EACH PASS (OR SUB PASS).
    ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
    ;--
    BGNCLN
L$CLEAN::
    MOV    CSRADDR,R5          ;POINT TO DEVICE REGISTER
    TST    DUF1G              ;"DROPPED" FLAG IS SET ON...
    BMI    1$                ;...AND GROSS CONTROLLER FAULT...
                                ;...DON'T TRY TO XCT CLEANUP CODE.
    MOV    #0,TSSR(R5)        ;DO SOFT INIT
    JSR    PC,WAITF
1$:
2$:
L10034:
    TRAP   C$CLEAN

    ; THE REPORT CODING SECTION CONTAINS THE
    ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
    ;--
    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

```

023120 012746 000002      MOV      #2,(SP)
023124 010600      MOV      SP,R0
023126 104416      TRAP     C:PNTS
023130 062706 000006      ADD      #6,SP
257 023134 000431      BR       4$
258 023136 020227 160001      3$:     CMP      R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
259 023142 001012      BNE      30$              ; BR IF NO.
260 023144      PRINTS  #DEVNRD,R3
023144 010346      MOV      R3,-(SP)
023146 012746 023433      MOV      #DEVNRD,-(SP)
023152 012746 000002      MOV      #2,(SP)
023156 010600      MOV      SP,R0
023160 104416      TRAP     C:PNTS
023162 062706 000006      ADD      #6,SP
261 023166 000414      BR       4$
262 023170 042702 170000      30$:    BIC      #C7777,R2
263 023174      PRINTS  #DEVDR0,R3,R2
023174 010246      MOV      R2,-(SP)
023176 010346      MOV      R3,-(SP)
023200 012746 023514      MOV      #DEVDR0,-(SP)
023204 012746 000003      MOV      #3,-(SP)
023210 010600      MOV      SP,R0
023212 104416      TRAP     C:PNTS
023214 062706 000010      ADD      #10,SP
264 023220 062704 000002      4$:     ADD      #2,R4
265 023224 005203      INC      R3
266 023226 020427 003366      CMP      R4,#ERTABE
267 023232 103701      BLO     1$
268 023234 012604      MOV      (SP),R4
269 023236 012603      MOV      (SP),R3
270 023240 012602      MOV      (SP),R2
271 023242      ENDRPT              ; UNUSED.
023242      L10035:
023242 104425      TRAP     C:RPT
272
273
274 023244      045      116      045  DEVSUM: .ASCIZ  /#N#ADEVICE STATUS SUMMARY:#N/
275 023301      045      101      040  DEVONL: .ASCIZ  /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
276 023351      045      101      040  DEVNXR: .ASCIZ  /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
277 023433      045      101      040  DEVNRD: .ASCIZ  /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
278 023514      045      101      040  DEVDR0: .ASCIZ  /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
279
280
281 023564      ENDMOD
282
283

```

CLEAN UP AND REPORT CODING SECTIONS

1
2
9
10
16
24

.TITLE TSV7 - HARDWARE TESTS 1-8

TSV7:: BGNMOD TSV7

023564
023564



TEST 1: INITIALIZE #4 TEST

```

26          .SBTTL TEST 1: INITIALIZE #4 TEST
27          ;*
28          ;
29          ; THIS TEST VERIFIES THAT WRITING INTO THE TSSR RETURNS THE
30          ; CONTROLLER TO ITS INITIALIZED STATE FROM VARIOUS CONDITIONS
31          ; (I.E. LOOPBACK ENABLED, FORCING WRONG PARITY, INVERTING SENSE OF
32          ; EXTENDED FEATURES SWITCH, ETC.)
33          ;
34          ;-
35 023564          BGNTST
36 023564 012737 006356 002170          MOV      #EPRT1,EPRTSW          ;SET UP PRIMARY ERROR MESSAGE
37
38          ;
39          ; TEST 1
40          ;
41          ;
42          ;
43          ;-
44
49 023572 004737 016274          JSR      PC,DSBINT          ;DISABLE INTERRUPTS
50 023576 012700 024524          MOV      #TST21ID,R0          ;ASCII MESSAGE TO IDENTIFY TEST
51 023602 004737 016600          JSR      PC,TSTSETUP          ;DO INITIAL TEST SETUP
52 023606 012737 000005 002206          MOV      #5,LOOPCNT          ;PERFORM 5 ITERATIONS
53 023614          T21LOOP:
54 023614 004737 024546          JSR      PC,T21REST          ;SET COMMAND PACKET
55 023620 004737 024636          JSR      PC,T21RT2          ;SET UP OTHER COMMAND PACKET
56
57          ;*****
58          ;
59          ; ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
60          ;
61          ;*****
62
63 023624 012737 176750 024202          MOV      #65000.,T21DLY          ;SET DELAY ROUTINE
64 023632 004737 016064          JSR      PC,SOFINIT          ;DO INITIALIZE ON CONTROLLER
65 023636 103426          BCS     20$          ;BR IF INIT WAS OK
66 023640          DELAY     250          ;DELAY FOR A REWIND TO FINISH
67          MOV      #250,(PC)+          ;
68          .WORD     0          ;
69          MOV      L#DLY,(PC)+          ;
70          .WORD     0          ;
71          DEC     6(PC)          ;
72          BNE     . 4          ;
73          DEC     -22(PC)          ;
74          BNE     . 20          ;
75 023670          DEC     T21DLY          ;BUMP COUNTER DOWN
76 023674          BNE     11$          ;BR, IF MORE TIME TO GO
77 023676          INC     FATFLG          ;BUMP COUNT
78 023702          MOV     R0,R1          ;CONTENTS OF TSSR REGISTER
79 023704          ERRDF  ERRNO,SFIERR,SFIMSG          ;FATAL ERROR TSSR WAS NOT OK
80          TRAP     C#ERDF          ;
81          .WORD     101          ;
82          .WORD     SFIERR          ;
83          .WORD     SFIMSG          ;
84
85 023714          20$:
86 023714 012704 024160          MOV      #T21PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS

```


TEST 1: INITIALIZE #4 TEST

```

140
141
142
144      024160
146 024160 100004
147 024160 024170
148 024162 024170
149 024164 000000
150 024166 000012
151 024170
152 024170 024204
153 024172 000000
154 024174 000024
155 024176 000000
156 024200 000000
157 024202 000000
158 024204
159
160
161
163      024270
165 024270 100206
166 024270 024300
167 024272 024300
168 024274 000000
169 024276 000006
170
171
172 024300
173 024300 000
174 024301 000
175 024302 000000
176 024304 000000
177
178
179

```

```

;+
;LOCAL STORAGE FOR THIS TEST
;-
      .=<.10>E177770
T21PACKET:
      .WORD 100004
      .WORD T21DATA
      .WORD 0
      .WORD 10.
T21DATA:
      .WORD T21BFR
      .WORD 0
      .WORD 20.
      .WORD 0
T21DSW: .WORD 0
T21DLY: .WORD 0
T21BFR: .BLKW 25.
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
      .=<.10>E177770
T21PK2:
      .WORD 100206
      .WORD T21BF2
      .WORD 0
      .WORD 6.
      .EVEN
T21BF2:
T21BS0: .BYTE 0
T21BS1: .BYTE 0
T21S2: .WORD 0
T21S3: .WORD 0

```

```

;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH, ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
;DRIVE SELECT WORD
;DELAY COUNTER
;MESSAGE BUFFER
;WRITE SUB SYS MEM COMMAND, IE AND ACK
;ADDRESS OF SELECT BLOCK DATA
;SIZE OF DATA PACKET
;BSEL0 AREA --- COMMAND" BYTE
;BSEL1 AREA
;SEL 2 AREA
;DATA AREA

```

TEST 1: INITIALIZE #4 TEST

```

181
182
183 ;*
184 ;LOCAL TEXT MESSAGES FOR TEST
185 ;-
186 024306 127 122 111 T21SSR: .ASCIZ 'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted
187 024403 124 123 123 T21AM3: .ASCIZ 'TSSR Init. Failed After WRITE MISCELLANEOUS CONTROL/READ STATUS
188 024503 104 162 151 T21OFL: .ASCIZ 'Drive is OFFLINE
189 024524 111 156 151 T21ID: .ASCIZ 'Initialization #4'
190 .EVEN
191 ;*
192 ;
193 ;ROUTINE TO RESTORE COMMAND PACKET TO START UP (DEFAULT) VALUES
194 ;WRITE SUBSYSTEM MEMORY COMMAND
195 ;
196 ;
197 024546 T21REST:
198 024546 SAVREG ;SAVE THE REGISTERS
199 024552 012701 024160 MOV #T21PACKET,R1 ;START OF THE PACKET
200 024556 012721 100004 MOV #100004,(R1); ;WRITE SUBSYSTEM MEM. WITH ACK.
201 024562 012721 024170 MOV #T21DATA,(R1); ;ADDRESS OF CHARAISTICS DATA BLOCK
202 024566 005021 CLR (R1); ;EXTENDED ADDRESS
203 024570 012721 000010 MOV #8,(R1); ;SIZE OF DATA BLOCK IN BYTES
204 024574 012721 024204 MOV #T21BFR,(R1); ;ADDRESS OF MESSAGE BUFFER
205 024600 005021 CLR (R1);
206 024602 012721 000024 MOV #20,(R1); ;LENGTH OF MESSAGE BUFFER
207 024606 005021 CLP (R1);
208 024610 005011 CLP (R1);
209 024612 012702 000020 MOV #20,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
210 024616 012762 177777 024204 64$: MOV #177777,T21BFR(R2) ;ALL ONES TO MESSAGE BUFFER
211 024624 005742 TST -(R2) ;NEXT LOCATION
212 024626 020227 000000 CMP R2,#0 ;CHECK R2 FOR ZERO
213 024632 001371 BNE 64$ ;BR, IF NOT AT ZERO YET
214 024634 000207 RTS PC ;RETURN
215
216 024636 T21RT2:
217 024636 SAVREG ;SAVE THE REGISTERS
218 024642 012701 024270 MOV #T21PK2,R1 ;START OF THE PACKET
219 024646 012721 100206 MOV #100206,(R1); ;WRITE SUBSYSTEM MEM. WITH ACK. IE
220 024652 012721 024300 MOV #T21BF2,(R1); ;ADDRESS OF DATA BLOCK
221 024656 005021 CLR (R1); ;EXTENDED ADDRESS
222 024660 012721 000006 MOV #6,(R1); ;SIZE OF DATA BLOCK IN BYTES
223 024664 005021 CLR (R1);
224 024666 012701 024300 MOV #T21BF2,R1 ;ADDRESS OF DATA FOR WRT SUB SYS MEM
225 024672 005021 CLR (R1);
226 024674 005011 CLR (R1);
227 024676 000207 RTS PC ;RETURN
228 024700 ENDTST
024700
024700 104401 L10036: TRAP C$ETST

```


TEST 2: OFF LINE AND REJECT REWIND

```

288
289
290
291
292
293
294
295 024774 004737 010752          JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
296 025000 103407                  BCS      23$                ;BR, IF COMMAND ISSUED ON
297 025002 005237 002212          INC      FATFLG             ;BUMP COUNT
301 025006 010001                  MOV      R0,R1              ;SAVE CONTENTS OF TSSR
302 025010                  ERRHRD  ERRNO,WRTMSG,SFMSG  ;WRITE CHARACTERISTICS FAILED
      025010 104456                  TRAP     C$ERHRD
      025012 000312                  .WORD   202
      025014 005054                  .WORD   WRTMSG
      025016 012124                  .WORD   SFMSG
303 025020          23$:  CKLOOP                  TRAP     C$CLP1
      025020 104406
304 025022 013701 026300          MOV      T22BFR+6,R1       ;PICK UP XT50
305 025026 032701 000004          BIT      #4,R1             ;IS UNIT WRITE LOCKED?
306 025032 001407                  BEQ      24$                ;NO,PROCEED WITH TESTING
307 025034 005237 002212          INC      FATFLG             ;BUMP COUNT
311 025040                  ERRDF  ERRNO,T22WLK,SFMSG  ;TAPE IS WRITE LOCKED
      025040 104455                  TRAP     C$ERDF
      025042 000313                  .WORD   203
      025044 027102                  .WORD   T22WLK
      025046 012124                  .WORD   SFMSG
312 025050                  DOCLN                      TRAP     C$DCLN
      025050 104444
313 025052          24$:  CKLOOP                  TRAP     C$CLP1
      025052 104406
314 025054 005737 002216          TST      EXTFEA            ;CHECK FOR EXTENDED FEATURES SW SWITCH
315 025060 001041                  BNE      50$                ;BR IF SWITCH IS ON
316 025062 112737 000200 026371  MOVB     #200,T22BS1       ;WRITE MISCELLANEOUS CONT/READ STATUS
317 025070 112737 000010 026370  MOVB     #10,T22BS0       ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
318 025076 012704 026360          MOV      #T22PK2,R4        ;WRITE SUBSYS MEM PACKET
319 025102 010465 000000          MOV      R4,TSDB(R5)       ;ISSUE COMMAND
320 025106 004737 016426          JSR      PC,CHKTSSR        ;WAIT FOR SSR
321 025112 103407                  BCS      30$                ;BR, IF NO ERROR
322 025114 010001                  MOV      R0,R1              ;ERROR, SAVE TSSR
323 025116 005237 002212          INC      FATFLG             ;BUMP COUNT
327 025122                  ERRHRD  ERRNO,T22SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      025122 104456                  TRAP     C$ERHRD
      025124 000314                  .WORD   204
      025126 026410                  .WORD   T22SSR
      025130 012136                  .WORD   PKTSSR
328 025132          30$:  CKLOOP                  ;LOOP IF SELECTED
      025132 104406                  TRAP     C$CLP1
329 025134 012704 026250          MOV      #T22PACKET,R4     ;SUBROUTINE NEEDS PACKET ADDRESS
330
331
332
333
334
335
336
337 025140 004737 010752          JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS

```

TEST 2: OFF LINE AND REJECT REWIND

338	025144	103407		BCS	50\$;BR, IF COMMAND ISSUED OK		
339	025146	005237	002212	INC	FATFLG			;BUMP COUNT		
343	025152	010001		MOV	R0,R1			;SAVE CONTENTS OF TSSR		
344	025154			ERRHRD	ERRNO,WRTMSG,SFIMSG			;WRITE CHARACTERISTIC FAILED		
	025154	104456						TRAP	C\$ERHRD	
	025156	000315						.WORD	205	
	025160	005054						.WORD	WRTMSG	
	025162	012124						.WORD	SFIMSG	
345	025164			50\$:	CKLOOP			;SCOPE LOOP		
	025164	104406						TRAP	C\$CLP1	
346	025166	016501	000002	MOV	TSSR(R5),R1			;GET TSSR CONTENTS		
347	025172	032701	000100	BIT	#OFL,R1			;CHECK FOR THE OFFLINE BIT SET		
348	025176	001006		BNE	60\$;BR, IF OFFLINE (GOOD)		
349	025200	005237	002212	INC	FATFLG			;BUMP COUNT		
353	025204			ERRDF	ERRNO,T22OFL,SFIMSG			;OFF LINE SHOULD HAVE BEEN SET (BAD)		
	025204	104455						TRAP	C\$ERDF	
	025206	000316						.WORD	206	
	025210	026605						.WORD	T22OFL	
	025212	012124						.WORD	SFIMSG	
354	025214			60\$:	CKLOOP			;LOOP IF SELECTED		
	025214	104406						TRAP	C\$CLP1	
355	025216	012703	026376	MOV	#T22RD,R3			; POINTER FOR COMMANDS		
356	025222	011337	026360	MOV	(R3),T22PK2			;TAPE MOTION COMMAND IN PLACE		
357	025226	012704	026360	MOV	#T22PK2,R4			;R4 = POINTER TO PACKET		
358	025232	010465	000000	MOV	R4,TSDB(R5)			;ISSUE COMMAND		
359	025236	004737	016340	JSR	PC,WAITF			;WAIT FOR SSR TO SET		
360	025242	016501	000002	MOV	TSSR(R5),R1			;GET TSSR CONTENTS		
361	025246	012702	100306	MOV	#SSR!SC!OFL!BIT1!BIT2,R2			;SET UP EXPECTED		
362	025252	020102		CMP	R1,R2			;ARE THEY EQUAL		
363	025254	001406		BEQ	80\$;BR, IF OK ESP. FUNCTION REJECT		
364	025256	005237	002212	INC	FATFLG			;BUMP COUNT		
368	025262			ERRHRD	ERRNO,T22TM,EXPREC			;TSSR INCORRECT AFTER TAPE MOTION CMD		
	025262	104456						TRAP	C\$ERHRD	
	025264	000317						.WORD	207	
	025266	026660						.WORD	T22TM	
	025270	015564						.WORD	EXPREC	
369	025272			80\$:	CKLOOP			;LOOP IF SELECTED		
	025272	104406						TRAP	C\$CLP1	
370	025274	005723		TST	(R3),			;POINT TO NEXT COMMAND		
371	025276	022713	177777	CMP	#177777,(R3)			;END OF THE COMMANDS YET		
372	025302	001401		BEQ	90\$;BR, IF DONE		
373	025304	000746		BR	65\$;MORE COMMAND(S) TO GO		
374	025306			90\$:						
375	025306			ENDSUB				; >>>>>>>>>> END SUBTEST >>>>>>>>>>>>>>>>		
	025306	104403						L10040:		
	025310	023727	002212 000017	CMP	FATFLG,#15.			TRAP	C\$ESUB	
376	025310	023727	002212 000017	CMP	FATFLG,#15.			; IS ERROR COUNT AT 25		
377	025316	103402		BLO	999\$;BR, IF LESS THAN 25		
378	025320	004737	017272	JSR	PC,CKDROP			;TRY TO DROP THE UNIT		
379	025324			999\$:						

174

TEST 2: OFF LINE AND REJECT REWIND

	025630				
	025630	104403			
483	025632	023727	002212	000017	
484	025640	103402			
485	025642	004737	017272		
486	025646				

9998:

CMP	FATFLG.#15.
BLO	9998
JSR	PC.CKDROP

L10041:	TRAP	C18ESUB
	!IS ERROR COUNT AT 25	
	!BR, IF LESS THAN 25	
	!TRY TO DROP THE UNIT	

TEST 2: OFF LINE AND REJECT REWIND

```

488 ;
489 ;
490 ; TEST 2, SUBTEST 3
491 ;
492 ; VERIFIES THAT A REWIND COMMAND WITH CVC-1 CLEARS VCK
493 ; AND RETURNS PROPER STATUS IN THE MESSAGE BUFFER.
494 ;
495 ;
496 ;
497 025646           BGNSUB                 ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
       025646           T2.3:
       025646 104402           TRAP      C#BSUB
498 025650 004737 027222       JSR      PC,T22REST           ;SET COMMAND PACKET
499 025654 004737 027314       JSR      PC,T22RT2          ;SET UP OTHER COMMAND PACKET
500 ;
501 ; *****
502 ;
503 ; ISSUE CONTROLLER "SOFT" INITIALIZE  CARRY BIT CLEAR IF ERROR
504 ;
505 ; *****
506 ;
507 025660 004737 016064       JSR      PC,SOFINIT          ;DO INITIALIZE ON CONTROLLER
508 025664 103407           BCS     20$               ;BR IF INIT WAS OK
509 025666 005237 002212       INC     FATFLG            ;BUMP COUNT
513 025672 010001           MOV     R0,R1             ;CONTENTS OF TSSP REGISTER
514 025674           ERRDF   ERRNO,SFIERR,SFIMSG           ;FATAL ERROR TSSR WAS NOT OK
       025674 104455           TRAP      C#ERDF
       025676 000326           .WORD   214
       025700 003650           .WORD   SFIERR
       025702 012124           .WORD   SFIMSG
515 025704           20$:
516 025704 012704 026250       MOV     #T22PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
517 ;
518 ; *****
519 ;
520 ; WRITE CHARACTERISTICS COMMAND (CALL TO WRTOCHR)
521 ;
522 ; *****
523 ;
524 025710 004737 010752       JSR      PC,WRTOCHR          ;ISSUE WRITE CHARACTERISTICS
525 025714 103407           BCS     23$               ;BR. IF COMMAND ISSUED OK
526 025716 005237 002212       INC     FATFLG            ;BUMP COUNT
530 025722 010001           MOV     R0,R1             ;SAVE CONTENTS OF TSSR
531 025724           ERRMRD  ERRNO,WRMSG,SFIMSG           ;WRITE CHARACTERISTIC FAILED
       025724 104456           TRAP      C#ERMRD
       025726 000327           .WORD   215
       025730 005054           .WORD   WRMSG
       025732 012124           .WORD   SFIMSG
532 025734 005737 002216       23$:  TST     EXTFEA           ;CHECK FOR EXTENDED FEATURES SW SWITCH
533 025740 001041           BNE     50$               ;BR IF SWITCH IS ON
534 ;
535 025742 112737 000200 026371  MOVB   #200,T22BS1        ;WRITE MISCELLANEOUS CONT/READ STATUS
536 025750 112737 000010 026370  MOVB   #10,T22BS0        ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
537 025756 012704 026360           MOV     #T22PK2,R4        ;WRITE SUBSYS MEM PACKET
538 025762 010465 000000           MOV     R4,TSDB(R5)       ;ISSUE COMMAND
539 025766 004737 016426       JSR      PC,CHKTSSR        ;WAIT FOR SSR
540 025772 103407           BCS     30$               ;BR. IF NO ERROR

```

TEST 2: OFF LINE AND REJECT REWIND

```

541 025774 010001          MOV      R0,R1          ;ERROR, SAVE TSSR
542 025776 005237 002212  INC      FATFLG        ;BUMP COUNT
546 026002          ERRHRD  ERRNO,T22SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
                                TRAP      C$ERHRD
                                .WORD    216
                                .WORD    T22SSR
                                .WORD    PKTSSR
-47 026012          30$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
548 026014 012704 026250  MOV      @T22PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
549
550 ;*****
551 ;
552 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
553 ;
554 ;*****
555
556 026020 004737 010752          JSR      PC,WRTPHR      ;ISSUE WRITE CHARACTERISTICS
557 026024 103407          BCS     50$            ;BR, IF COMMAND ISSUED OK
558 026026 005237 002212  INC      FATFLG        ;BUMP COUNT
562 026032 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
563 026034          ERRHRD  ERRNO,WRTPMSG,SFMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP      C$ERHRD
                                .WORD    217
                                .WORD    WRTPMSG
                                .WORD    SFMSG
564 026044          50$:   CKLOOP          ;SCOPE LOOP
                                TRAP      C$CLP1
565 026046 016501 000002  MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
566 026052 032701 000100  BIT      @OFL,R1        ;CHECK FOR THE OFFLINE BIT SET
567 026056 001006          BNE     60$            ;BR, IF OFFLINE (GOOD)
568 026060 005237 002212  INC      FATFLG        ;BUMP COUNT
572 026064          ERRDF  ERRNO,T22OFL,SFMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
                                TRAP      C$ERDF
                                .WORD    218
                                .WORD    T22OFL
                                .WORD    SFMSG
573 026074          60$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
574 026076 012737 142010 026360 65$:   MOV      #142010,T22PK2 ;POSITION COMMAND (REWIND MODE) CVC=1
575 026104 012704 026360  MOV      @T22PK2,R4     ;R4 = POINTER TO PACKET
576 026110 010465 000000  MOV      R4,TSDB(R5)   ;ISSUE COMMAND
577 026114 004737 016340  JSR      PC,WAITF      ;WAIT FOR SSR TO SET
578 026120 016501 000002  MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
579 026124 012702 100306  MOV      @SSR!OFL!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
580 026130 020102          CMP     R1,R2          ;ARE THEY EQUAL
581 026132 001406          BEQ     80$            ;BR, IF OK ESP. FUNCTION REJECT
582 026134 005237 002212  INC      FATFLG        ;BUMP COUNT
586 026140          ERRHRD  ERRNO,T22RWJ,EXPREC ;TSSR INCORRECT AFTER TAPE MOTION CMD
                                TRAP      C$ERHRD
                                .WORD    219
                                .WORD    T22RWJ
                                .WORD    EXPREC
587 026150          80$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
588 026152 012703 026272  MOV      @T22BFR,R3     ;POINTER TO MESSAGE BUFFER
589 026156 016301 000006  MOV      XST0(R3),R1    ;PICK UP XST0 FROM MESSAGE BUFFER

```


TEST 2: OFF LINE AND REJECT REWIND

610			;		
611			;	LOCAL STORAGE FOR THIS TEST	
612			;		
614		026250	;	.=<.*10>E177770	
616	026250		T22PACKET:		;COMMAND PACKET FOR TEST
617	026250	100204	.WORD	100204	;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
618	026252	026260	.WORD	T22DATA	;ADDRESS OF CHARACTERISTICS BLOCK
619	026254	000000	.WORD	0	
620	026256	000012	.WORD	10.	;STARTING VALUE OF BLOCK SIZE
621	026260		T22DATA:		;CHARACTERISTIC DATA BLOCK
622	026260	026272	.WORD	T22BFR	;ADDRESS OF MESSAGE BUFFER
623	026262	000000	.WORD	0	
624	026264	000024	.WORD	20.	;LENGTH OF MESSAGE BUFFER
625	026266	000000	.WORD	0	
626	026270	000007	.WORD	7	;SELECT DRIVE 7
627	026272		T22BFR: .BLKW	25.	;MESSAGE BUFFER
628			;		
629			;	WRITE SUBSYSTEM MEMORY COMMAND PACKET	
630			;		
632		026360	;	.=<.*10>E177770	
634	026360		T22PK2:		
635	026360	100206	.WORD	100206	;WRITE SUB SYS MEM COMMAND, IE AND ACK
636	026362	026370	.WORD	T22BF2	;ADDRESS OF SELECT BLOCK DATA
637	026364	000000	.WORD	0	
638	026366	000006	.WORD	6.	;SIZE OF DATA PACKET
639					
640			.EVEN		
641	026370		T22BF2:		
642	026370	000	T22BS0: .BYTE	0	;BSELO AREA
643	026371	000	T22BS1: .BYTE	0	;BSEL1 AREA
644	026372	000000	T22S2: .WORD	0	;SEL 2 AREA
645	026374	000000	T22S3: .WORD	0	;DATA AREA
646			;		
647			;		
648			.EVEN		
649			;	TAPE MOTION PACKET COMMAND VALUES	
650	026376	100201	T22RD: .WORD	100201	;READ TAPE FORWARD
651	026400	100205	T22WRT: .WORD	100205	;WRITE TAPE FORWARD
652	026402	100210	T22POS: .WORD	100210	;POSITION TAPE
653	026404	100211	T22FOR: .WORD	100211	;FORMAT TAPE
654	026406	177777	.WORD	177777	;END OF DATA
655					
656					

TEST 2: OFF LINE AND REJECT REWIND

```

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

```


TEST 3: BASIC WRITE DATA

```

770 027464
771 027464 012737 000007 032510 20$:  MOV    #7,T23DSW      ;SET DRIVE NUMBER IN PACKET
772 027472 012704 032470      MOV    #T23PACKET,R4  ;SUBROUTINE NEEDS PACKET ADDRESS
773
774 ;*****
775 ;
776 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
777 ;
778 ;*****
779
780 027476 004737 010752      JSR    PC,WRTPHR      ;ISSUE WRITE CHARACTERISTICS
781 027502 103407      BCS    23$           ;BR, IF COMMAND ISSUED OK
782 027504 005237 002212      INC    FATFLG        ;BUMP COUNT
786 027510 010001      MOV    R0,R1         ;SAVE CONTENTS OF TSSR
787 027512      ERRHRD  ERRNO,WRTPHR,SFIMSG ;WRITE CHARACTERISTICS FAILED
      TRAP    C$ERRHRD
      .WORD   302
      .WORD   WRTPHR
      .WORD   SFIMSG
788 027522 005737 002216 23$:  TST    EXTFEA        ;CHECK FOR EXTENDED FEATURES SW SWITCH
789 027526 001044      BNE    50$           ;BR IF SWITCH IS ON
790
791 027530 112737 000200 032623  MOVB   #200,T23BS1    ;WRITE MISCELLANEOUS CONT/READ STATUS
792 027536 112737 000010 032622  MOVB   #10,T23BS0     ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
793 027544 012704 032600      MOV    #T23PK2,R4    ;WRITE SUBSYS MEM PACKET
794 027550 010465 000000      MOV    R4,TSDB(R5)   ;ISSUE COMMAND
795 027554 004737 016426      JSR    PC,CHKTSSR    ;WAIT FOR SSR
796 027560 103407      BCS    30$           ;BR, IF NO ERROR
797 027562 010001      MOV    R0,R1         ;ERROR, SAVE TSSR
798 027564 005237 002212      INC    FATFLG        ;BUMP COUNT
802 027570      ERRHRD  ERRNO,T23SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      TRAP    C$ERRHRD
      .WORD   303
      .WORD   T23SSR
      .WORD   PKTSSR
803 027600 104406 30$:  CKLOOP      ;LOOP IF SELECTED
      TRAP    C$CLP1
804 027602 012737 000007 032510  MOV    #7,T23DSW      ;SET DRIVE NUMBER IN PACKET
805 027610 012704 032470      MOV    #T23PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
806
807 ;*****
808 ;
809 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
810 ;
811 ;*****
812
813 027614 004737 010752      JSR    PC,WRTPHR      ;ISSUE WRITE CHARACTERISTICS
814 027620 103407      BCS    50$           ;BR, IF COMMAND ISSUED OK
815 027622 005237 002212      INC    FATFLG        ;BUMP COUNT
819 027626 010001      MOV    R0,R1         ;SAVE CONTENTS OF TSSR
820 027630      ERRHRD  ERRNO,WRTPHR,SFIMSG ;WRITE CHARACTERISTICS FAILED
      TRAP    C$ERRHRD
      .WORD   304
      .WORD   WRTPHR
      .WORD   SFIMSG
821 027640 104406 50$:  CKLOOP      ;SCOPE LOOP
      TRAP    C$CLP1
    
```

TEST 3: BASIC WRITE DATA

```

822 027642 016501 000002      MOV     TSSR(R5),R1      ;GET TSSR CONTENTS
823 027646 032701 000100      BIT     @OFL,R1         ;CHECK FOR THE OFFLINE BIT SET
824 027652 001006                BNE     60$             ;BR, IF OFFLINE (GOOD)
825 027654 005237 002212      INC     FATFLG          ;BUMP COUNT
829 027660                ERRDF   ERRNO,T23OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
           027660 104455                          TRAP   C$ERDF
           027662 000461                          .WORD 305
           027664 033306                          .WORD T23OFL
           027666 012124                          .WORD SFIMSG
830 027670                60$:   CKLOOP          ;LOOP IF SELECTED
           027670 104406                          TRAP   C$CLP1
831 027672 012703 032634      MOV     @T23WD,R3       ;POINTER FOR COMMANDS
832 027676 011337 032600      65$:   MOV     (R3),T23PK2 ;TAPE MOTION COMMAND IN PLACE
833 027702 012704 032600      MOV     @T23PK2,R4      ;R4 = POINTER TO PACKET
834 027706 010465 000000      MOV     R4,TSDB(R5)     ;ISSUE COMMAND
835 027712 004737 016340      JSR    PC,WAIF         ;WAIT FOR SSR TO SET
836 027716 016501 000002      MOV     TSSR(R5),R1     ;GET TSSR CONTENTS
837 027722 012702 100306      MOV     @SSR!SC!OFL!BIT1!BIT2,R2 ;SET UP EXPECTED
838 027726 020102                CMP     R1,R2           ;ARE THEY EQUAL
839 027730 001406                BEQ     80$             ;BR, IF OK ESP. FUNCTION REJECT
840 027732 005237 002212      INC     FATFLG          ;BUMP COUNT
844 027736                ERRHRD  ERRNO,T23TM,EXPREC ;TSSR INCORRECT AFTER TAPE MOTION CMD
           027736 104456                          TRAP   C$ERHRD
           027740 000462                          .WORD 306
           027742 033042                          .WORD T23TM
           027744 015564                          .WORD EXPREC
845 027746                80$:   CKLOOP          ;LOOP IF SELECTED
           027746 104406                          TRAP   C$CLP1
846 027750 005723                TST     (R3)+           ;POINT TO NEXT COMMAND
847 027752 022713 177777      CMP     @177777,(R3)    ;END OF THE COMMANDS YET
848 027756 001401                BEQ     90$             ;BR, IF DONE
849 027760 000746                BR      65$             ;MORE COMMAND(S) TO GO
850 027762                90$:                ENDSUB             ;>>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>>>>>>>
851 027762                L10044:
           027762 104403                          TRAP   C$ESUB
852 027764 023727 002212 000017  CMP     FATFLG,@15.     ;IS ERROR COUNT AT 25
853 027772 103402                BLO     999$            ;BR, IF LESS THAN 25
854 027774 004737 017272      JSR    PC,CKDROP       ;TRY TO DROP THE UNIT
855 030000                999$:

```


TEST 3: BASIC WRITE DATA

```

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

```

TEST 3: BASIC WRITE DATA

```

964 030274          90$:  CKLOOP          ;LOOP IF SELECTED
      030274 104406          TRAP      C$CLP1
965 030276 020327 007376      CMP      R3,#7376      ;ONLY CHECK RAM UNTIL ITS FULL
966 030302 002114          BGE      115$          ;IT WRAPS AROUND ETC.
967 030304 004737 034074      JSR      PC,T23RT2     ;MAKE SURE PACKET AND DATA ARE CLEAN
968 030310 012737 000400 032624  MOV      #256.,T23S2  ;STARTING RAM ADDRESS
969 030316 112737 000000 032622  MOVVB   #0,T23BS0     ;STOP INTERNAL TSV05 DIAGNOSTICS
970 030324 112737 000000 032623  MOVVB   #0,T23BS1     ;SIZE OF RAM READ
971 030332 012704 032600      MOV      #T23PK2,R4   ;SET R4 WITH PACKET ADDRESS
972 030336 010465 000000      MOV      R4,TSDB(R5)  ;ISSUE WRITE SUB SYS MEM COMMAND
973 030342 004737 016426      JSR      PC,CHKTSSR   ;CHECK TSSR AND WAIT FOR SSR TO SET
974 030346 103407          BCS      92$          ;BR, IF NO ERRORS IN TSSR
975 030350 010001          MOV      R0,R1        ;SAVE TSSR
976 030352 005237 002212      INC      FATFLG       ;BUMP COUNT
980 030356          ERRHRD   ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
      030356 104456          TRAP      C$ERHRD
      030360 000466          .WORD    310
      030362 033677          .WORD    T23WSS
      030364 012136          .WORD    PKTSSR
981 030366          92$:  CKLOOP          ;LOOP IF SELECTED
      030366 104406          TRAP      C$CLP1
982 030370 004737 034074      JSR      PC,T23RT2     ;MAKE SURE PACKET AND DATA ARE CLEAN
983 030374 012737 000400 032624  MOV      #256.,T23S2  ;STARTING RAM ADDRESS
984 030402 112737 000001 032622  MOVVB   #1,T23BS0     ;READ RAM COMMAND FOR WRITE SUB SYS M.
985 030410 112737 000002 032623  MOVVB   #2,T23BS1     ;SIZE OF RAM READ
986 030416 012704 032600      MOV      #T23PK2,R4   ;SET R4 WITH PACKET ADDRESS
987 030422 010465 000000 95$:  MOV      R4,TSDB(R5)  ;ISSUE WRITE SUB SYS MEM COMMAND
988 030426 004737 016426      JSR      PC,CHKTSSR   ;CHECK TSSR AND WAIT FOR SSR TO SET
989 030432 103407          BCS      100$         ;BR, IF NO ERRORS IN TSSR
990 030434 010001          MOV      R0,R1        ;SAVE TSSR
991 030436 005237 002212      INC      FATFLG       ;BUMP COUNT
995 030442          ERRHRD   ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
      030442 104456          TRAP      C$ERHRD
      030444 000467          .WORD    311
      030446 033677          .WORD    T23WSS
      030450 012136          .WORD    PKTSSR
996 030452          100$: CKLOOP          ;LOOP IF SELECTED
      030452 104406          TRAP      C$CLP1
997 030454 005001          CLR      R1            ;CLEAR REGISTER
998 030456 005002          CLR      R2            ;CLEAR REGISTER
999 030460 013701 032532      MOV      T23BFR+20,R1 ;PICK UP BYTE READ FROM RAM
1000 030464 010302          MOV      R3,R2         ;SET UP EXPECTED
1001 030466 020102          CMP      R1,R2         ;IS RAM DATA CORRECT
1002 030470 001406          BEQ      110$         ;BR, IF OK (EQUAL)
1003 030472 005237 002212      INC      FATFLG       ;BUMP COUNT
1007 030476          ERRHRD   ERRNO,T23RNC,EXPREC ;RNC=RAM NOT CORRECT
      030476 104456          TRAP      C$ERHRD
      030500 000470          .WORD    312
      030502 033165          .WORD    T23RNC
      030504 015564          .WORD    EXPREC
1008 030506          110$: CKLOOP          ;LOOP IF SELECTED
      030506 104406          TRAP      C$CLP1
1009 030510 005237 032624      INC      T23S2        ;BUMP RAM ADDRESS TO BE CHECKED
1010 030514 005237 032624      INC      T23S2        ;BUMP RAM ADDRESS TO BE CHECKED
1011 030520 010301          MOV      R3,R1         ;GET SIZE OF RECORD
1012 030522 062701 000400      ADD      #256.,R1     ;FIGURE OUT END RECORD ADDRESS
1013 030526 023701 032624      CMP      T23S2,R1     ;AT END OF RAM CHECK YET

```


TEST 3: BASIC WRITE DATA

```

1095
1096
1097
1098
1099
1100 030756 012737 150005 032610      MOV      #150005,T23PK3      ;WRITE DATA,CVC=1,ACK,SWB COMMAND
1101 030764 012737 150005 032632      MOV      #150005,T23WRT    ;SETUP FOR RETRY COMMAND
1102 030772 052737 004000 032632      BIS      #4000,T23WRT     ;MAKE IT A RETRY
1103 031000 012704 032610      MOV      #T23PK3,R4       ;SET UP R4 WITH PACKET ADDRESS
1104 031004 010300      MOV      R3,R0            ;SET PATTERN IN CORRECT REGISTER
1105 031006 004737 017512      JSR      PC,FILLMEM       ;FILL MEMORY WITH RECORD SIZE
1106 031012 010337 032616      MOV      R3,T23SZ        ;SET UP RECORD SIZE IN PACKET
1107 031016 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
1108 031022 004737 016340      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
1109 031026 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
1110 031032 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
1111 031036 020102      CMP      R1,R2           ;ARE THEY EQUAL
1112 031040 001402      BEQ      80$            ;BR, IF OK
1113 031042 004737 034162      JSR      PC,T23CHK       ;CHECK SPECIAL CONDITION
1114 031046      80$:      CKLOOP                ;LOOP IF SELECTED
1115 031050 016501 000000      MOV      TSBA(R5),R1     ;GET TSBA CONTENTS          TRAP      C$CLP1
1116 031054 012702 032512      MOV      #T23BFR,R2     ;SET UP EXPECTED
1117 031060 062702 000016      ADD      #16,R2         ;SET TO END OF MESSAGE BUFFER
1118 031064 005737 002216      TST      EXTFEA         ;CHECK FOR EXTENDED FEATURES SW SET
1119 031070 001402      BEQ      85$            ;BR, IF IT NOT SET
1120 031072 062702 000002      ADD      #2,R2          ;BUMP R2 FOR EXTRA DATA
1121 031076 020102      85$:      CMP      R1,R2         ;ARE THEY EQUAL
1122 031100 001406      BEQ      90$            ;BR, IF TSBA IS CORRECT
1123 031102 005237 002212      INC      FATFLG         ;BUMP COUNT
1127 031106      ERRHRD  ERRNO,T23BA,EXPREC ;TSBA WAS NOT CORRECT AFTER WRITE DATA
1128 031106 104456      TRAP      C$ERHRD
1128 031110 000474      .WORD    316
1128 031112 033625      .WORD    T23BA
1128 031114 015564      .WORD    EXPREC
1128 031116      90$:      CKLOOP                ;LOOP IF SELECTED
1129 031120 020327 007376      CMP      R3,#7376       ;ONLY CHECK RAM UNTIL ITS FULL
1130 031124 002115      BGE      115$          ;IT WRAPS AROUND ETC.
1131 031126 004737 034074      JSR      PC,T23RT2      ;MAKE SURE PACKET AND DATA ARE CLEAN
1132 031132 012737 000400 032624      MOV      #256.,T23S2    ;STARTING RAM ADDRESS
1133 031140 112737 000000 032622      MOV      #0,T23B50     ;STOP INTERNAL TSV05 DIAGNOSTICS
1134 031146 112737 000000 032623      MOV      #0,T23B51     ;SIZE OF RAM READ
1135 031154 012704 032600      MOV      #T23PK2,R4     ;SET R4 WITH PACKET ADDRESS
1136 031160 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE WRITE SUB SYS MEM COMMAND
1137 031164 004737 016426      JSR      PC,CHKTSSR     ;CHECK TSSR AND WAIT FOR SSR TO SET
1138 031170 103407      BCS      92$            ;BR, IF NO ERRORS IN TSSR
1139 031172 010001      MOV      R0,R1          ;SAVE TSSR
1140 031174 005237 002212      INC      FATFLG         ;BUMP COUNT
1144 031200      ERRHRD  ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
1144 031200 104456      TRAP      C$ERHRD
1144 031202 000475      .WORD    317
1144 031204 033677      .WORD    T23WSS
1144 031206 012136      .WORD    PKTSSR
1145 031210      92$:      CKLOOP                ;LOOP IF SELECTED
1146 031212 004737 034074      JSR      PC,T23RT2      ;MAKE SURE PACKET AND DATA ARE CLEAN

```

TEST 3: BASIC WRITE DATA

```

1147 031216 012737 000400 032624      MOV      #256.,T23S2      ;STARTING RAM ADDRESS
1148 031224 112737 000001 032622      MOVB     #1,T23BS0      ;READ RAM COMMAND FOR WRITE SUB SYS M.
1149 031232 112737 000002 032623      MOVB     #2,T23BS1      ;SIZE OF RAM READ
1150 031240 012704 032600      MOV      #T23PK2,R4     ;SET R4 WITH PACKET ADDRESS
1151 031244 010465 000000      95$:    MOV      R4,TSDB(R5)   ;ISSUE WRITE SUB SYS MEM CMD (READ RAM)
1152 031250 004737 016426      JSR      PC,CHKTSSR     ;CHECK TSSR AND WAIT FOR SSR TO SET
1153 031254 103407      BCS      100$          ;BR, IF NO ERRORS IN TSSR
1154 031256 010001      MOV      R0,R1         ;SAVE TSSR
1155 031260 005237 002212      INC      FATFLG        ;BUMP COUNT
1159 031264      ERRHRD   ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
                                TRAP      C$ERHRD
                                .WORD     318
                                .WORD     T23WSS
                                .WORD     PKTSSR
                                TRAP      C$CLP1
                                .WORD     318
                                .WORD     T23RNC
                                .WORD     EXPREC
1160 031274      100$:    CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD     318
                                .WORD     T23RNC
                                .WORD     EXPREC
1161 031276 005001      CLR      R1            ;CLEAR REGISTERS
1162 031300 005002      CLR      R2            ;CLEAR REGISTERS
1163 031302 013701 032532      MOV      T23BFR+20,R1  ;PICK UP BYTE READ FROM RAM
1164 031306 010302      MOV      R3,R2        ;SET UP EXPECTED
1165 031310 000302      SWAB     R2           ;SWAP BYTES
1166 031312 020102      CMP      R1,R2        ;IS RAM DATA CORRECT
1167 031314 001406      BEQ      110$         ;BR, IF OK (EQUAL)
1168 031316 005237 002212      INC      FATFLG        ;BUMP COUNT
1172 031322      ERRHRD   ERRNO,T23RNC,EXPREC ;RNC=RAM NOT CORRECT
                                TRAP      C$ERHRD
                                .WORD     319
                                .WORD     T23RNC
                                .WORD     EXPREC
                                TRAP      C$CLP1
                                .WORD     319
                                .WORD     T23RNC
                                .WORD     EXPREC
1173 031332      110$:    CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD     319
                                .WORD     T23RNC
                                .WORD     EXPREC
1174 031334 005237 032624      INC      T23S2        ;BUMP RAM ADDRESS TO BE CHECKED
1175 031340 005237 032624      INC      T23S2        ;BUMP RAM ADDRESS TO BE CHECKED
1176 031344 010301      MOV      R3,R1        ;GET SIZE OF RECORD
1177 031346 062701 000400      ADD      #256.,R1     ;FIGURE OUT END RECORD ADDRESS
1178 031352 023701 032624      CMP      T23S2,R1     ;AT END OF RAM CHECK YET
1179 031356 001332      BNE      95$         ;BR, IF MORE TO CHECK
1180 031360 062703 001750      115$:    ADD      #1000.,R3    ;NEXT RECORD SIZE/DATA PATTERN
1181 031364 020337 032620      CMP      R3,T23RSZ   ;IS R3 OVER MAX RECORD SIZE
1182 031370 002005      BGE      120$        ;IF RECORD SIZE IS TOO BIG QUIT
1183 031372 020327 177776      CMP      R3,#65534.  ;END OF SUBTEST MAX RECORD SIZE
1184 031376 001402      BEQ      120$        ;BR, IF COMPLETED
1185 031400 000137 030750      JMP      65$         ;DO MORE RECORDS
1186 031404      120$:    JSR      PC,T23RT2     ;CLEAN UP PACKET
1187 031404 004737 034074      MOV      #102010,T23PK2 ;REWIND (POSITION) COMMAND
1188 031410 012737 102010 032600      MOV      #T23PK2,R4   ;LOAD R4 WITH PACKET ADDRESS
1189 031416 012704 032600      MOV      R4,TSDB(R5)  ;ISSUE REWIND COMMAND
1190 031422 010465 000000      JSR      PC,CHKTSSR   ;WAIT FOR SSR TO SET
1191 031426 004737 016426      BCS      130$        ;BR, IF TSSR IS OK (GOOD)
1192 031432 103407      MOV      R0,R1        ;SAVE TSSR CONTENTS
1193 031434 010001      INC      FATFLG        ;BUMP COUNT
1194 031436 005237 002212      ERRHRD   ERRNO,T23RWN,PKTSSR ;TSSR IS INCORRECT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD     320
                                .WORD     T23RWN
                                .WORD     PKTSSR
1198 031442      104456
                                000500
                                033116
                                012136

```

FLO

TEST 3: BASIC WRITE DATA

```
1199 031452
1200 031452
    031452
    031452 104403
1201 031454 023727 002212 000017
1202 031462 103402
1203 031464 004737 017272
1204 031470
    999$:
```

```
130$:
ENDSUB
;..... END SUBTEST .....
L10046:
        TRAP      CSFSJB
        ;IS ERROR COUNT AT 25
        ;BR, IF LESS THAN 25
        ;TRY TO DROP THE UNIT
```


TEST 3: BASIC WRITE DATA

```

1341 032006 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      032006 104456 TRAP C$ERHRD
      032010 000505 .WORD 325
      032012 005054 .WORD WRTMSG
      032014 012124 .WORD SFIMSG
1342
1343 ;*****
1344 ;
1345 ;WRITE DATA, ACK, CVC=1
1346 ;
1347 ;*****
1348
1349 032016 23$:
1350 032016 004737 021276 JSR PC,INVERT ;INVERT THE EXTENDED FEATURES SWITCH
1351 032022 012737 140005 032610 MOV #140005,T23PK3 ;WRITE DATA, ACK, CVC=1
1352 032030 013737 003130 032612 MOV NXML0,T23WB ;SET UP WRITE BUFFER ADDRESS
1353 032036 013737 003132 032614 MOV NXMHI,T23WB+2 ;HIGH ORDER ADDRESS BITS
1354 032044 012737 000100 032616 MOV #64.,T23SZ ;SET UP BUFFER SIZE
1355 032052 012704 032610 MOV #T23PK3,R4 ;R4 = POINTER TO PACKET
1356 032056 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
1357 032062 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
1358 032066 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
1359 032072 012702 104210 MOV #SC!NXM!SSR!BIT3,R2 ;SET UP EXPECTED
1360 032076 020102 CMP R1,R2 ;ARE THEY EQUAL
1361 032100 001406 BEQ 80$ ;BR, IF OK ESP. FUNCTION REJECT
1362 032102 005237 002212 INC FATFLG ;BUMP COUNT
1366 032106 ERRHRD ERRNO,T23TM,PKTSSR ;TSSR INCORRECT AFTER WRITE COMMAND
      032106 104456 TRAP C$ERHRD
      032110 000506 .WORD 326
      032112 033042 .WORD T23TM
      032114 012136 .WORD PKTSSR
1367 032116 80$: CKLOOP ;LOOP IF SELECTED
      032116 104406 TRAP C$CLP1
1368 032120 90$:
1369 032120 ENDSUB ;>>>>>>>>>> END SUBTEST >>>>>>>>>
      032120 L10050:
      032120 104403 TRAP C$ESUB
1370 032122 023727 002212 000017 CMP FATFLG,#15. ;IS ERROR COUNT AT 25
1371 032130 103402 BLO 999$ ;BR, IF LESS THAN 25
1372 032132 004737 017272 JSR PC,CKDROP ;TRY TO DROP THE UNIT
1373 032136 999$:

```


TEST 3: BASIC WRITE DATA

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

163\$:

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

```

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

```

;ALL DONE THIS TEST

TRAP	C\$EXIT
.WORD	L10043 .

TEST 3: BASIC WRITE DATA

```

1486
1487
1488
1490      032470
1492 032470 100004
1493 032470 032500
1494 032472 000000
1495 032474 000010
1496 032476 032512
1497 032500 000000
1498 032502 000012
1499 032504 000000
1500 032506 000000
1501 032510 000000
1502 032512
1503
1504
1505
1506
1508      032600
1510 032600 100006
1511 032602 032622
1512 032604 000000
1513 032606 000006
1514
1515
1519 032610 100005
1520 032612 000000
1521 032614 000000
1522 032616 000000
1523
1524
1525
1526 032620 000000
1527
1528
1529 032622      010
1530 032622      200
1531 032624 000000
1532 032626 000000
1533
1534
1535
1536 032630 000000
1537 032632 000000
1538
1539
1540
1541
1542 032634 100005
1543 032636 100405
1544 032640 102005
1545 032642 177777
1546
1547

```

```

;+
;LOCAL STORAGE FOR THIS TEST
;-
      .=<.10>&177770
T23PACKET:
      .WORD 100004
      .WORD T23DATA
      .WORD 0
      .WORD 8.
T23DATA:
      .WORD T23BFR
      .WORD 0
      .WORD 10.
      .WORD 0
T23DSW: .WORD 0
T23BFR: .BLKW 25.
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
      .=<.10>&177770
T23PK2:
      .WORD 100006
      .WORD T23BF2
      .WORD 0
      .WORD 6.
T23PK3:
      .WORD 100005
T23WB: .WORD 0
      .WORD 0
T23SZ: .WORD 0
      .EVEN
;
T23RSZ: .WORD 0
;
T23BF2:
T23BS0: .BYTE 10
T23BS1: .BYTE 200
T23S2: .WORD 0
T23S3: .WORD 0
;
;
T23TMP: .WORD 0
T23WRT: .WORD 0
;
      .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T23WD: .WORD 100005
T23WDR: .WORD 100405
T23CON: .WORD 102005
      .WORD 177777

```

```

;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
;SELECT DRIVE 0
;MESSAGE BUFFER
;WRITE SUB SYS MEM COMMAND, AND ACK
;ADDRESS OF SELECT BLOCK DATA
;SIZE OF DATA PACKET
;WRITE COMMAND, AND ACK
;ADDRESS OF WRITE BUFFER
;SIZE OF BUFFER (EXTENT)
;LARGEST TAPE RECORD IN BYTES
;BSELO AREA
;BSEL1 AREA
;SEL 2 AREA
;DATA AREA
;TEMPORARY REGISTER
;RETRY COMMAND
;WRITE DATA (NEXT)
;WRITE DATA RETRY
;WRITE CONTINUOUS
;END OF DATA

```

TEST 3: BASIC WRITE DATA

```

1549
1550
1551          ;*
1552          ;LOCAL TEXT MESSAGES FOR TEST
1553          ;-
1554 032644      127      122      111  T23SSR: .ASCIZ  'WRITE Command Not Accepted'
1555 032677      105      117      124  T23ET:  .ASCIZ  'EOT Not Found In 12000 4k Writes, (Use Shorter Tape)'
1556 032764      127      122      111  T23EOT: .ASCIZ  'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
1557 033042      124      123      123  T23TM:  .ASCIZ  'TSSR Not Correct After WRITE Command Reject'
1558 033116      122      145      167  T23RWN: .ASCIZ  'Rewind (POSITION) Command Not Accepted'
1559 033165      122      101      115  T23RNC: .ASCIZ  'RAM Error, Correct Data Pattern Not In Ram'
1560 033240      124      123      123  T23AM3: .ASCIZ  'TSSR Init. Failed After WRITE Command'
1561 033306      104      162      151  T23OFL: .ASCIZ  'Drive 7 Select Failed To Set "OFL" In TSSR'
1562 033361      124      123      123  T23WDD: .ASCIZ  'TSSR Not Correct After WRITE DATA Command, SWB Bit Set'
1563 033450      124      123      123  T23WDC: .ASCIZ  'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
1564 033552      103      126      103  T23VCK: .ASCIZ  'CVC Set, Didn't Reset VCK In Message Buffer'
1565 033625      124      123      102  T23BA:  .ASCIZ  'TSBA Not Correct After WRITE DATA Command'
1566 033677      127      122      111  T23WSS: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
1567 033766      102      141      163  T23ID:  .ASCIZ  'Basic Write'
1568
1569          .EVEN
1570
1571          ;*
1572          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1573          ;WRITE SUBSYSTEM MEMORY COMMAND
1574          ;
1575          ;-
1576 034002
1577 034002
1578 034006      012701    032470
1579 034012      012721    100004
1580 034016      012721    032500
1581 034022      005021
1582 034024      012721    000012
1583 034030      012721    032512
1584 034034      005021
1585 034036      012721    000024
1586 034042      005021
1587 034044      012711    000000
1588 034050      012702    000030
1589 034054      012762    177777    032512    64$:
1590 034062      005742
1591 034064      020227    000000
1592 034070      001371
1593 034072      000207
1594
1595
1596 034074
1597 034074
1598 034100      012701    032600
1599 034104      012721    100006
1600 034110      012721    032622
1601 034114      005021
1602 034116      012721    000006
1603 034122      012701    032622
1604 034126      005021
1605 034130      005021

          T23REST:
          SAVREG
          MOV      #T23PACKET,R1          ;SAVE THE REGISTERS
          MOV      #100004,(R1)+        ;START OF THE PACKET
          MOV      #T23DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK
          CLR      (R1)+                ;ADDRESS OF CHARAISTICS DATA BLOCK
          MOV      #10.,(R1)+          ;EXTENDED ADDRESS
          MOV      #T23BFR,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
          CLR      (R1)+                ;ADDRESS OF MESSAGE BUFFER
          MOV      #20.,(R1)+          ;LENGTH OF MESSAGE BUFFER
          CLR      (R1)+
          MOV      #0,(R1)              ;SELECT DRIVE ZERO
          MOV      #24.,R2              ;NUMBER OF LOCATIONS TO BE CLEARED
          MOV      #177777,T23BFR(R2)  ;ALL ONES TO MESSAGE BUFFER
          TST      -(R2)                 ;BUMP DOWN TO NEXT LOCATION
          CMP      R2,#0                 ;R2 AT ZERO YET
          BNE     64$                   ;KEEP GOING UNTIL DONE
          RTS      PC                    ;RETURN

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

```


TEST 3: BASIC WRITE DATA

SEQ 0132

1662 034344 004737 017272
1663 034350
034350
034350 104401

JSR PC.CKDROP
ENDTST

;DROP THE UNIT

L10043: TRAP CSEFST

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

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

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

1770 034632 012136          30$:  CKLOOP          ;LOOP IF SELECTED          .WORD  PKTSSR
1770 034634          TRAP  C$CLP1
1771 034634 104406          MOV      #7,T24DSW          ;SET DRIVE NUMBER IN PACKET
1772 034636 012737 000007 043640  MOV      #T24PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
1772 034644 012704 043620
1773
1774          ;*****
1775          ;
1776          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
1777          ;
1778          ;*****
1779
1780 034650 004737 010752          JSR      PC,WRTPHR          ;ISSUE WRITE CHARACTERISTICS
1781 034654 103407          BCS     50$                ;BR, IF COMMAND ISSUED OK
1782 034656 005237 002212          INC     FATFLG             ;BUMP COUNT
1786 034662 010001          MOV     R0,R1              ;SAVE CONTENTS OF TSSR
1787 034664          ERRHRD  ERRNO,WRTPHR,SFIMSG ;WRITE CHARACTERISTICS FAILED
1787 034664 104456          TRAP   C$ERRHRD           ;
1787 034666 000624          .WORD  404                ;
1787 034670 005054          .WORD  WRTPHR             ;
1787 034672 012124          .WORD  SFIMSG             ;
1788 034674          50$:  CKLOOP          ;SCOPE LOOP          TRAP  C$CLP1
1788 034674 104406
1789 034676 016501 000002          MOV     TSSR(R5),R1        ;GET TSSR CONTENTS
1790 034702 032701 000100          BIT    #OFL,R1            ;CHECK FOR THE OFFLINE BIT SET
1791 034706 001006          BNE    60$                ;BR, IF OFFLINE (GOOD)
1792 034710 005237 002212          INC     FATFLG             ;BUMP COUNT
1796 034714          ERRDF  ERRNO,T24OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
1796 034714 104455          TRAP   C$ERRDF           ;
1796 034716 000625          .WORD  405                ;
1796 034720 045265          .WORD  T24OFL             ;
1796 034722 012124          .WORD  SFIMSG             ;
1797 034724          60$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
1798 034724 104406          MOV     #T24RN,R3          ;POINTER FOR COMMANDS
1798 034726 012703 043756
1799
1800          ;*****
1801          ;
1802          ;TAPE READ COMMAND IN PLACE
1803          ;
1804          ;*****
1805
1806 034732 011337 043740          65$:  MOV     (R3),T24PK3    ;TAPE READ COMMAND IN PLACE
1807 034736 012704 043740          MOV     #T24PK3,R4        ;R4 = POINTER TO PACKET
1808 034742 010465 000000          MOV     R4,TSD8(R5)       ;ISSUE COMMAND
1809 034746 004737 016340          JSR     PC,WAITF           ;WAIT FOR SSR TO SET
1810 034752 016501 000002          MOV     TSSR(R5),R1       ;GET TSSR CONTENTS
1811 034756 012702 100306          MOV     #SSR!SC!OFL!BIT1!BIT2,R2 ;SET UP EXPECTED
1812 034762 020102          CMP     R1,R2             ;ARE THEY EQUAL
1813 034764 001406          BEQ    80$                ;BR, IF OK ESP. FUNCTION REJECT
1814 034766 005237 002212          INC     FATFLG             ;BUMP COUNT
1818 034772          ERRHRD  ERRNO,T24TM,PKTSSR ;TSSR INCORRECT AFTER TAPE MOTION CMD
1818 034772 104456          TRAP   C$ERRHRD           ;
1818 034774 000626          .WORD  406                ;
1818 034776 045023          .WORD  T24TM              ;
1818 035000 012136          .WORD  PKTSSR             ;
1819 035002          80$:  CKLOOP          ;LOOP IF SELECTED

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

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

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

1885 035132 012124
035134 104406
1886
1887
1888
1889
1890
1891
1892
1893 035136 004737 011104
1894 035142 103407
1895 035144 010001
1896 035146 005237 002212
1900 035152
035152 104456
035154 000631
035156 045076
035160 012136
1901 035162
035162 104406
1902
1903
1904
1905
1906
1907
1908
1909 035164 013701 043650
1910 035170 010102
1911 035172 052702 000002
1912 035176 020102
1913 035200 001406
1914 035202 005237 002212
1918 035206
035206 104456
035210 000632
035212 044613
035214 015564
1919 035216
035216 104406
1920 035220 012703 000400
1921 035224 013737 003114 043742
1922
1923
1924
1925
1926
1927
1928
1929 035232 012737 140005 043740
1930 035240 012704 043740
1931 035244
1932 035244 010300
1933 035246 004737 017512
1934 035252 010337 043746
1935 035256 010465 000000

24$: CKLOOP ;LOOP IF SELECTED .WORD SFMSG
TRAP C$CLP1
;*****
;
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
JSR PC,REWIND ;CALL TAPE REWIND COMMAND
BCS 30$ ;BR, IF NO PROBLEM
MOV R0,R1 ;SAVE TSSR
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
TRAP C$ERHRD
.WORD 409
.WORD T24RWN
.WORD PKTSSR

30$: CKLOOP ;LOOP IF SELECTED
TRAP C$CLP1
;*****
;
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;
;*****
MOV T24BFR+6,R1 ;PICK UP XSTO
MOV R1,R2 ;SET UP EXPECTED
BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
CMP R1,R2 ;DOES EXP = REC'D
BEQ 40$ ;BR, IF EQUAL (OK)
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
TRAP C$ERHRD
.WORD 410
.WORD T24BOT
.WORD EXPREC

40$: CKLOOP ;LOOP IF SELECTED
TRAP C$CLP1
MOV #256.,R3 ;RECORD SIZE
MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
;*****
;
;WRITE DATA,CVC=1,ACK COMMAND
;
;*****
MOV #140005,T24PK3 ;WRITE DATA,CVC=1,ACK COMMAND
MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS

65$:
MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
MOV R4,TSDB(R5) ;ISSUE COMMAND

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

1936 035262 004737 016340      JSR    PC,WAITF      ;WAIT FOR SSR TO SET
1937 035266 016501 000002      MOV    TSSR(R5),R1  ;GET TSSR CONTENTS
1938 035272 012702 000200      MOV    #SSR,R2      ;SET UP EXPECTED
1939 035276 020102              CMP    R1,R2        ;ARE THEY EQUAL
1940 035300 001406              BEQ    75$          ;BR, IF OK
1941 035302 005237 002212      INC    FATFLG       ;BUMP COUNT
1945 035306              ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERHRD
                                .WORD  411
                                .WORD  WRTErr
                                .WORD  PKTSSR
                                75$:  CKLOOP      ;LOOP IF SELECTED
                                TRAP    C$CLP1
1946 035316 104456              TST    (R3)+        ;BUMP RECORD SIZE
                                CMP    #268.,R3      ;END OF RECORD YET
1947 035320 005723              BNE    65$          ;BR, IF MORE RECORDS TO WRITE
1948 035322 022703 000414      80$:  CKLOOP      ;LOOP IF SELECTED
1949 035326 001346              TRAP    C$CLP1
1950 035330 104406              120$:
1951 035332              ;*****
1952              ;
1953              ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
1954              ;
1955              ;*****
1956              ;
1957              ;
1958              ;
1959 035332 004737 011104      JSR    PC,REWIND    ;CALL TAPE REWIND COMMAND
1960 035336 004737 016426      JSR    PC,CHKTSSR  ;SEE HOW TSSR IS
1961 035342 103407              BCS    130$        ;BR, IF NO PROBLEM
1962 035344 010001              MOV    R0,R1        ;SAVE TSSR
1963 035346 005237 002212      INC    FATFLG       ;BUMP COUNT
1967 035352              ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD  412
                                .WORD  T24RWN
                                .WORD  PKTSSR
1968 035362 104406              130$:  CKLOOP      ;LOOP IF SELECTED
                                TRAP    C$CLP1
1969              ;*****
1970              ;
1971              ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
1972              ;
1973              ;*****
1974              ;
1975              ;
1976 035364 013701 043650      MOV    T248FR+6,R1 ;PICK UP XSTO
1977 035370 010102              MOV    R1,R2        ;SET UP EXPECTED
1978 035372 052702 000002      BIS    #BIT1,R2     ;SET BOT BIT IN EXPECTED
1979 035376 020102              CMP    R1,R2        ;DOES EXP = REC'D
1980 035400 001406              BEQ    140$        ;BR, IF EQUAL (OK)
1981 035402 005237 002212      INC    FATFLG       ;BUMP COUNT
1985 035406              ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD  413
                                .WORD  T24BOT
                                .WORD  EXPREC
1986 035416              140$:  CKLOOP      ;LOOP IF SELECTED

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

1987 035416 104406                    TRAP C$CLP1
1988 035420 012703 000400      MOV #256.,R3      ;RECORD SIZE
1988 035424 013737 003114 043742  MOV FREE,T24RB  ;STARTING READ BUFFER ADDRESS
1989
1990 ;*****
1991 ;
1992 ;READ DATA,CVC=1,ACK COMMAND
1993 ;
1994 ;*****
1995
1996 035432 012737 140001 043740      MOV #140001,T24PK3  ;READ DATA,CVC=1,ACK COMMAND
1997 035440 012704 043740 165$:   MOV #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
1998 035444 010337 043746          MOV R3,T24SZ       ;SET UP RECORD SIZE IN PACKET
1999 035450 010465 000000          MOV R4,TSDB(R5)    ;ISSUE COMMAND
2000 035454 004737 016340          JSR PC,WAITF       ;WAIT FOR SSR TO SET
2001 035460 016501 000002          MOV TSSR(R5),R1   ;GET TSSR CONTENTS
2002 035464 012702 000200          MOV #SSR,R2       ;SET UP EXPECTED
2003 035470 020102                    CMP R1,R2         ;ARE THEY EQUAL
2004 035472 001406                    BEQ 170$          ;BR, IF OK
2005 035474 005237 002212          INC FATFLG        ;BUMP COUNT
2009 035500          ERRHRD ERRNO,RDERR,PKTSSR  ;TSSR INCORRECT AFTER READ DATA
104456                    TRAP C$ERHRD
035502 000636                    .WORD 414
035504 005204                    .WORD RDERR
035506 012136                    .WORD PKTSSR
2010 035510          170$: CKLOOP              ;LOOP IF SELECTED
104406                    TRAP C$CLP1
035512 013702 003114          MOV FREE,R2       ;GET BUFFER ADDRESS
2012 035516 010304          MOV R3,R4         ;CURRENT RECORD SIZE
2013 035520 162704 000400          SUB #256.,R4     ;FIRST LOCATION IN BUFFER
2014 035524 060204          173$: ADD R2,R4        ;GET LOCATION IN BUFFER (ADDRESS)
2015 035526 021403          CMP (R4),R3      ;CHECK DATA READ (R3=DATA ALSO)
2016 035530 001410          BEQ 180$        ;BR, IF ALL IS WELL
2017 035532 011401          MOV (R4),R1     ;RECD DATA
2018 035534 010302          MOV R3,R2       ;EXPECTED DATA
2019 035536 005237 002212          INC FATFLG      ;BUMP COUNT
2023 035542          ERRHRD ERRNO,T24DTA,EXPREC  ;DATA READ NOT = WRITTEN
104456                    TRAP C$ERHRD
035544 000637                    .WORD 415
035546 044660                    .WORD T24DTA
035550 015564                    .WORD EXPREC
2024 035552          180$: CKLOOP              ;LOOP IF SELECTED
104406                    TRAP C$CLP1
035554 005724          TST (R4)+        ;BUMP TO NEXT LOCATION
2026 035556 160204          SUB R2,R4        ;GET BACK TO CORRECT SIZE
2027 035560 020403          CMP R4,R3       ;END OF RECORD YET
2028 035562 001360          BNE 173$        ;BR, IF NOT AT END OF RECORD
2029 035564 005723          TST (R3)+       ;BUMP RECORD SIZE
2030 035566 022703 000412          CMP #266.,R3    ;END OF RECORD YET
2031 035572 001322          BNE 165$        ;BR, IF MORE RECORDS TO WRITE
2032 035574          190$: CKLOOP              ;LOOP IF SELECTED
104406                    TRAP C$CLP1
035576          ENDSUB              ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
L10054:
035576 104403                    TRAP C$ESUB
2034 035600 023727 002212 000017   CMP FATFLG,#15.  ;IS ERROR COUNT AT 25
2035 035606 103402          BLO 999$        ;BR, IF LESS THAN 25

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

2036 035610 004737 017272
2037 035614

999%: JSR PC,CKDROP

;TRY TO DROP THE UNIT

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2039          ;*
2040          ;
2041          ;TEST 4, SUBTEST 3
2042          ;
2043          ;VERIFIES THAT READ DATA COMMANDS WITH CVC=1 AND THE
2044          ;SWAP BYTES (SWB) BIT SET OPERATES PROPERLY. THE TEST
2045          ;SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2.
2046          ;THE RESULTS, EXCEPT FOR RAM CONTENTS, SHOULD BE THE SAME.
2047          ;
2048          ;
2049          ;
2050          ;-
2051 035614      BGNSUB                      ;>>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
          035614                               T4.3:
          035614 104402                       TRAP          C$BSUB
2052 035616 004737 046434                    JSR      PC,T24RT3        ;SET UP OTHER COMMAND PACKET
2053 035622 004737 046300                    JSR      PC,T24REST      ;SET COMMAND PACKET
2054 035626 004737 046372                    JSR      PC,T24RT2      ;SET UP OTHER COMMAND PACKET
2055          ;
2056          ;*****
2057          ;
2058          ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
2059          ;
2060          ;*****
2061          ;
2062 035632 004737 016064                      JSR      PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
2063 035636 103407                                BCS      20$             ;BR IF INIT WAS OK
2064 035640 005237 002212                      INC      FATFLG         ;BUMP COUNT
2068 035644 010001                                MOV      R0,R1          ;CONTENTS OF TSSP REGISTER
2069 035646                                ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
          035646 104455                               TRAP          C$ERDF
          035650 000640                               .WORD        416
          035652 003650                               .WOR         SFIERR
          035654 012124                               .WORD        SFIMSG
2070 035656                                20$:
2071 035656 013737 002172 043640              MOV      UNITN,T24DSW    ;SET DRIVE NUMBER IN PACKET
2072 035664 012704 043620                      MOV      #T24PACKET,R4  ;SUBROUTINE NEEDS PACKET ADDRESS
2073          ;
2074          ;*****
2075          ;
2076          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
2077          ;
2078          ;*****
2079          ;
2080 035670 004737 010752                      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
2081 035674 103407                                BCS      24$             ;BR, IF COMMAND ISSUED OK
2082 035676 005237 002212                      INC      FATFLG         ;BUMP COUNT
2086 035702 010001                                MOV      R0,R1          ;SAVE CONTENTS OF TSSR
2087 035704                                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
          035704 104456                               TRAP          C$ERHRD
          035706 000641                               .WORD        417
          035710 005054                               .WORD        WRTMSG
          035712 012124                               .WORD        SFIMSG
2088 035714                                24$:  CKLOOP                    ;LOOP IF SELECTED
          035714 104406                               TRAP          C$CLP1
2089          ;
2090          ;*****

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

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

```

File

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2144 036062 005237 002212          INC      FATFLG          ;BUMP COUNT
2148 036066          ERPHRD  ERRNO,WTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      036066 104456          TRAP      C$ERRRD
      036070 000644          .WORD    420
      036072 005111          .WORD    WTERR
      036074 012136          .WORD    PKTSSR
2149 036076          75%:   CKLOOP          ;LOOP IF SELECTED          TRAP      C$CLP1
      036076 104406          ;BUMP RECORD SIZE
2150 036100 005723          TST      (R3),          ;END OF RECORD YET
2151 036102 022703 000414          CMP      #268.,R3      ;BR, IF MORE RECORDS TO WRITE
2152 036106 001346          BNE      65%           ;LOOP IF SELECTED          TRAP      C$CLP1
2153 036110          80%:   CKLOOP
      036110 104406          120%:
2154 036112          ;*****
2155          ;
2156          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2157          ;
2158          ;*****
2159          ;
2160          ;
2161          ;
2162 036112 004737 011104          JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
2163 036116 103407          BCS      130%          ;BR, IF NO PROBLEM
2164 036120 010001          MOV      R0,R1         ;SAVE TSSR
2165 036122 005237 002212          INC      FATFLG          ;BUMP COUNT
2169 036126          ERRHRD  ERRNO,T24RWN,EXPREC ;REWIND NOT ACCEPTED          TRAP      C$ERRRD
      036126 104456          .WORD    421
      036130 000645          .WORD    T24RWN
      036132 045076          .WORD    EXPREC
2170 036136          130%:  CKLOOP          ;LOOP IF SELECTED          TRAP      C$CLP1
      036136 104406          ;*****
2171          ;
2172          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
2173          ;
2174          ;*****
2175          ;
2176          ;
2177          ;
2178 036140 013701 043650          MOV      T24BFR+6,R1   ;PICK UP XSTO
2179 036144 010102          MOV      R1,R2         ;SET UP EXPECTED
2180 036146 052702 000002          BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
2181 036152 020102          CMP      R1,R2         ;DOES EXP = REC'D
2182 036154 001406          BEQ      140%          ;BR, IF EQUAL (OK)
2183 036156 005237 002212          INC      FATFLG          ;BUMP COUNT
2187 036162          ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND          TRAP      C$ERRRD
      036162 104456          .WORD    422
      036164 000646          .WORD    T24BOT
      036166 044613          .WORD    EXPREC
2188 036172          140%:  CKLOOP          ;LOOP IF SELECTED          TRAP      C$CLP1
      036172 104406          ;RECORD SIZE
2189 036174 012703 000400          MOV      #256.,R3      ;STARTING READ BUFFER ADDRESS
2190 036200 013737 003114 043742          MOV      FREE,T24RB
2191          ;*****
2192          ;
2193          ;

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2241 ;
2242 ;
2243 ;TEST 4, SUBTEST 4
2244 ;
2245 ;VERIFIES THAT A READ FORWARD COMMAND READING A RECORD
2246 ;LONGER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE
2247 ;STATUS ALERT TERMINATION WITH THE RECORD LENGTH LONG
2248 ;(RLL) BIT SET.
2249 ;
2250 ;
2251 ;
2252 ;
2253 036370          BGNSUB          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
        036370          T4.4:
        036370 104402          TRAP      C1BSUB
2254 036372 004737 046434      JSR      PC,T24RT3      ;SET UP OTHER COMMAND PACKET
2255 036376 004737 046300      JSR      PC,T24REST  ;SET COMMAND PACKET
2256 036402 004737 046372      JSR      PC,T24RT2  ;SET UP OTHER COMMAND PACKET
2257 ;
2258 ;*****
2259 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
2260 ;
2261 ;
2262 ;*****
2263 ;
2264 036406 004737 016064      JSR      PC,SOFINIT  ;DO INITIALIZE ON CONTROLLER
2265 036412 103407          BCS      20$        ;BR IF INIT WAS OK
2266 036414 005237 002212      INC      FATFLG     ;BUMP COUNT
2270 036420 010001          MOV      R0,R1      ;CONTENTS OF TSSR REGISTER
2271 036422          ER,DF  ERRNO,SFIERR,SFIMSG  ;FATAL ERROR TSSR WAS NOT OK
        036422 104455          TRAP      C$ERDF
        036424 000651          .WORD   425
        036426 003650          .WORD   SFIERR
        036430 012124          .WORD   SFIMSG
2272 036432          20$:
2273 036432 013737 002172 043640 MOV      UNITN,T24DSW  ;SET DRIVE NUMBER IN PACKET
2274 036440 012704 043620      MOV      @T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
2275 ;
2276 ;*****
2277 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRCHR)
2278 ;
2279 ;
2280 ;*****
2281 ;
2282 036444 004737 010752      JSR      PC,WRCHR   ;ISSUE WRITE CHARACTERISTICS
2283 036450 103407          BCS      24$        ;BR, IF COMMAND ISSUED OK
2284 036452 005237 002212      INC      FATFLG     ;BUMP COUNT
2288 036456 010001          MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2289 036460          ERHRD  ERRNO,WRTMSG,SFIMSG  ;WRITE CHARACTERISTIC FAILED
        036460 104456          TRAP      C$ERHRD
        036462 000652          .WORD   426
        036464 005054          .WORD   WRTMSG
        036466 012124          .WORD   SFIMSG
2290 036470          24$: CKLOOP          ;LOOP IF SELECTED
        036470 104406          TRAP      C$CLP1
2291 ;
2292 ;*****

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

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

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

036634 104456 TRAP C$ERHRD
036636 000655 .WORD 429
036640 005111 .WORD WRTERR
036642 012136 .WORD PKTSSR
2349 036644 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
036644 104406
2350 036646 120$:
2351
2352 ;*****
2353 ;
2354 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2355 ;
2356 ;*****
2357
2358 036646 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2359 036652 103407 BCS 130$ ;BR, IF NO PROBLEM
2360 036654 010001 MOV R0,R1 ;SAVE TSSR
2361 036656 005237 002212 INC FATFLG ;BUMP COUNT
2365 036662 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
036662 104456 TRAP C$ERHRD
036664 000656 .WORD 430
036666 045076 .WORD T24RWN
036670 012136 .WORD PKTSSR
2366 036672 130$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
036672 104406
2367
2368 ;*****
2369 ;
2370 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
2371 ;
2372 ;*****
2373
2374 036674 013701 043650 MOV T24BFR+6,R1 ;PICK UP XSTO
2375 036700 010102 MOV R1,R2 ;SET UP EXPECTED
2376 036702 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
2377 036706 020102 CMP R1,R2 ;DOES EXP = REC'D
2378 036710 001406 BEQ 140$ ;BR, IF EQUAL (OK)
2379 036712 005237 002212 INC FATFLG ;BUMP COUNT
2383 036716 ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
036716 104456 TRAP C$ERHRD
036720 000657 .WORD 431
036722 044613 .WORD T24BOT
036724 015564 .WORD EXPREC
2384 036726 140$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
036726 104406
2385 036730 012703 000400 MOV #256.,R3 ;RECORD SIZE
2386 036734 013737 003114 043742 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
2387
2388 ;*****
2389 ;
2390 ;READ DATA,ACK,CVC=1 COMMAND
2391 ;
2392 ;*****
2393
2394 036742 012737 140001 043740 MOV #140001,T24PK3 ;READ DATA,ACK,CVC=1 COMMAND
2395 036750 012704 043740 165$: MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2396 036754 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET

```

GLC

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2397 036760 010465 000000      MOV     R4,TSD8(R5)      ;ISSUE COMMAND
2398 036764 004737 016340      JSR     PC,WAITF        ;WAIT FOR SSR TO SET
2399 036770 016501 000002      MOV     TSSR(R5),R1     ;GET TSSR CONTENTS
2400 036774 012702 100204      MOV     #SSR!SC!BIT2,R2 ;SET UP EXPECTED
2401 037000 020102                CMP     R1,R2           ;ARE THEY EQUAL
2402 037002 001406                BFD    170$             ;BR. IF OK
2403 037004 005237 002212      INC     FATFLG          ;BUMP COUNT
2407 037010                ERPHRD  ERRNO,T24TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP     C$ERHRD
                                .WORD   432
                                .WORD   T24TRL
                                .WORD   PKTSSR
                                TRAP     C$CLP;
2408 037020                170$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP;
2409                                ;*****
2410                                ;
2411                                ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
2412                                ;
2413                                ;*****
2414                                ;
2415                                ;
2416 037022 013701 043650      MOV     T24BFR+6,R1     ;GET MESSAGE BUFFER
2417 037026 010102                MOV     R1,R2           ;SET UP EXPECTED
2418 037030 052702 010000      BIS     #BIT12,R2       ;SET THE RLL BIT IN EXPECTED
2419 037034 020102                CMP     R1,R2           ;ARE THEY EQUAL
2420 037036 001406                BEQ    180$             ;BR. IF EQUAL (ALL IS WELL)
2421 037040 005237 002212      INC     FATFLG          ;BUMP COUNT
2425 037044                ERPHRD  ERRNO,T24LON,EXPREC ;THE RLL BIT WAS NOT SET IN XST0
                                TRAP     C$ERHRD
                                .WORD   433
                                .WORD   T24LON
                                .WORD   EXPREC
2426 037054                180$:  ENDSUR          ;>>>>>>>>> END SUBTEST >>>>>>>>>
2427 037054                L10056:
                                TRAP     C$ESUB
2428 037056 023727 002212 000017  CMP     FATFLG,#15.     ;IS ERROR COUNT AT 25
2429 037064 103402                BLO    999$             ;BR. IF LESS THAN 25
2430 037066 004737 017272      JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
2431 037072                999$:

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2433    ;*
2434    ;
2435    ;TEST 4, SUBTEST 5
2436    ;
2437    ;VERIFIES THAT A READ FORWARD COMMAND READING A RECORD
2438    ;SHORTER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE
2439    ;STATUS ALERT TERMINATION WITH THE RECORD LENGTH SHORT
2440    ;(RLS) BITS SET. IT IS VERIFIED THAT THE RESIDUAL BYTE
2441    ;COUNT (RBPCR) IN THE MESSAGE BUFFER CONTAINS THE
2442    ;PROPER NONZERO VALUE (E.G. THE DIFFERENCE BETWEEN
2443    ;THE ORIGINAL BYTE COUNT AND THE ACTUAL RECORD
2444    ;LENGTH).
2445    ;
2446    ;
2447    ;
2448    ;
2449    ;
2450    037072      BGNSUB                      ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
         037072          T4.5:
         037072 104402          TRAP         C$BSUB
2451    037074 004737 046434      JSR         PC,T24RT3      ;SET UP OTHER COMMAND PACKET
2452    037100 004737 046300      JSR         PC,T24REST    ;SET COMMAND PACKET
2453    037104 004737 046372      JSR         PC,T24RT2    ;SET UP OTHER COMMAND PACKET
2454
2455    ;*****
2456    ;
2457    ;ISSUE CONTROLLER "SOFT" INITIALIZE CARRY BIT CLEAR IF ERROR
2458    ;
2459    ;*****
2460
2461    037110 004737 016064      JSR         PC,SOFINIT   ;DO INITIALIZE ON CONTROLLER
2462    037114 103407             BCS        24$          ;BR IF INIT WAS OK
2463    037116 005237 002212      INC         FATFLG       ;BUMP COUNT
2464    037122 010001             MOV         R0,R1        ;CONTENTS OF TSSR REGISTER
2465    037124             ERRDF    ERRNO,SFIERR,SFIMSG  ;FATAL ERROR TSSR WAS NOT OK
         037124 104455          TRAP         C$ERDF
         037126 000662          .WORD      434
         037130 003650          .WORD      SFIERR
         037132 012124          .WORD      SFIMSG
2469    037134 013737 002172 043640 20$:      MOV         UNITN,T24DSW ;SET DRIVE NUMBER IN PACKET
2470    037142 012704 043620      MOV         @T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
2471
2472    ;*****
2473    ;
2474    ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
2475    ;
2476    ;*****
2477
2478
2479    037146 004737 010752      JSR         PC,WRTCHR    ;ISSUE WRITE CHARACTERISTICS
2480    037152 103407             BCS        24$          ;BR, IF COMMAND ISSUED OK
2481    037154 005237 002212      INC         FATFLG       ;BUMP COUNT
2485    037160 010001             MOV         R0,R1        ;SAVE CONTENTS OF TSSR
2486    037162             ERRHRD    ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
         037162 104456          TRAP         C$ERHRD
         037164 000663          .WORD      435
         037166 005054          .WORD      WRTMSG

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2487 037170 012124
037172 104406
2488
2489
2490
2491
2492
2493
2494
2495 037174 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2496 037200 103407 BCS 30$ ;BR, IF NO PROBLEM
2497 037202 010001 MOV R0,R1 ;SAVE TSSR
2498 037204 005237 002212 INC FATFLG ;BUMP COUNT
2502 037210 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
037210 104456 TRAP C$ERHRD
037212 000664 .WORD 436
037214 045076 .WORD T24RWN
037216 012136 .WORD PKTSSR
2503 037220 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
037220 104406
2504 037222 012703 000400 MOV #256.,R3 ;RECORD SIZE
2505 037226 013737 003114 043742 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
2506
2507
2508
2509
2510
2511
2512
2513 037234 012737 140005 043740 MOV #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
2514 037242 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2515 037246 65$:
2516 037246 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
2517 037252 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
2518 037256 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
2519 037262 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
2520 037266 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
2521 037272 020102 CMP R1,R2 ;ARE THEY EQUAL
2522 037274 001406 BEQ 75$ ;BR, IF OK
2523 037276 005237 002212 INC FATFLG ;BUMP COUNT
2527 037302 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
037302 104456 TRAP C$ERHRD
037304 000665 .WORD 437
037306 005111 .WORD WRERR
037310 012136 .WORD PKTSSR
2528 037312 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
037312 104406
2529 037314 120$:
2530
2531
2532
2533
2534
2535
2536
2537 037314 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

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

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2590 037476 020102          CMP      R1,R2           ;IS THE DIFFERENCE CORRECT
2591 037500 001406          BEQ      190$           ;BR, IF CORRECT
2592 037502 005237 002212  INC      FATFLG        ;BUMP COUNT
2596 037506          ERRHRD  ERRNO,T24PBP,EXPRES  ;RBPCR NOT CORRECT
      037506 104456          TRAP     C$ERHRD
      037510 000671          .WORD   441
      037512 046056          .WORD   T24PBP
      037514 015564          .WORD   EXPRES
2597 037516          190$:  CKLOOP          ;LOOP IF SELECTED
      037516 104406          TRAP     C$CLP1
2598 037520          ENDSUB          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
      037520          L10057:
      037520 104403          TRAP     C$ESUB
2599 037522 023727 002212 000017  CMP      FATFLG,#15.   ;IS ERROR COUNT AT 25
2600 037530 103402          BLO     999$           ;BR, IF LESS THAN 25
2601 037532 004737 017272          JSR     PC,CKDROP     ;TRY TO DROP THE UNIT
2602 037536          999$:

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2659 037634 012124          24$:  CKLOOP          ;LOOP IF SELECTED          .WORD  SFIMSG
      037636 104406          ;*****
2660          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2661          ;*****
2662          ;
2663          ;
2664          ;
2665          ;*****
2666          ;
2667 037640 004737 011104      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
2668 037644 103407          BCS      30$              ;BR, IF NO PROBLEM
2669 037646 010001          MOV      R0,R1            ;SAVE TSSR
2670 037650 005237 002217      INC      FATFLG          ;BUMP COUNT
2674 037654          ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      037654 104456          TRAP     C$ERHRD
      037656 000674          .WORD   444
      037660 045076          .WORD   T24RWN
      037662 012136          .WORD   PKTSSR
2675 037664          30$:  CKLOOP          ;LOOP IF SELECTED          TRAP   C$CLP1
      037664 104406          ;
2676 037666 012703 000400      MOV      #256.,R3        ;RECORD SIZE
2677 037672 013737 003114 043742  MOV      FREE,T24RB      ;STARTING WRITE BUFFER ADDRESS
2678          ;*****
2679          ;
2680          ;WRITE DATA,ACK,CVC=1 COMMAND
2681          ;
2682          ;
2683          ;*****
2684          ;
2685 037700 012737 140005 043740  MOV      #140005,T24PK3  ;WRITE DATA,ACK,CVC=1 COMMAND
2686 037706 012704 043740      MOV      #T24PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
2687 037712          65$:  MOV      R3,R0          ;SET PATTERN IN CORRECT REGISTER
2688 037712 010300          JSR      PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
2689 037714 004737 017512      MOV      R3,T24SZ       ;SET UP RECORD SIZE IN PACKET
2690 037720 010337 043746      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
2691 037724 010465 000000      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
2692 037730 004737 016340      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
2693 037734 016501 000002      MOV      #SSR,R2       ;SET UP EXPECTED
2694 037740 012702 000200      CMP      R1,R2         ;ARE THEY EQUAL
2695 037744 020102          BEQ      75$           ;BR, IF OK
2696 037746 001406          INC      FATFLG        ;BUMP COUNT
2697 037750 005237 002212      ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
2701 037754          TRAP     C$ERHRD
      037754 104456          .WORD   445
      037756 000675          .WORD   WRERR
      037760 005111          .WORD   PKTSSR
      037762 012136          ;
2702 037764          75$:  CKLOOP          ;LOOP IF SELECTED          TRAP   C$CLP1
      037764 104406          ;
2703 037766 005723          TST      (R3)+          ;BUMP RECORD SIZE
2704 037770 022703 000414      CMP      #268.,R3      ;END OF RECORD YET
2705 037774 001346          BNE      65$           ;BR, IF MORE RECORDS TO WRITE
2706 037776          80$:  CKLOOP          ;LOOP IF SELECTED
      037776 104406          TRAP     C$CLP1
2707 040000 005743          TST      -(R3)         ;SET BACK TO 512.
2708 040002 013737 003114 043742  MOV      FREE,T24RB      ;STARTING READ BUFFER ADDRESS

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2709
2710
2711 ;*****
2712 ;READ REVERSE DATA,ACK COMMAND
2713 ;
2714 ;*****
2715
2716 040010 012737 100401 043740      MOV      #100401,T24PK3      ;READ REVERSE DATA,ACK COMMAND
2717 040016 012704 043740      165$:  MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
2718 040022 010337 043746      MOV      R3,T24SZ      ;SET UP RECORD SIZE IN PACKET
2719 040026 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
2720 040032 004737 016340      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
2721 040036 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
2722 040042 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
2723 040046 020102      CMP      R1,R2      ;ARE THEY EQUAL
2724 040050 001406      BEQ      170$      ;BR, IF OK
2725 040052 005237 002212      INC      FATFLG      ;BUMP COUNT
2729 040056      ERRHRD  ERRNO,T24WDC,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      TRAP      C$ERRHRD
      .WORD      446
      .WORD      T24WDC
      .WORD      PKTSSR
2730 040066      170$:  CKLOOP      ;LOOP IF SELECTED
      TRAP      C$CLP1
      .WORD      104406
2731 040070 013702 003114      MOV      FREE,R2      ;GET BUFFER ADDRESS
2732 040074 010304      MOV      R3,R4      ;CURRENT RECORD SIZE
2733 040076 162704 000400      SUB      #256.,R4      ;FIRST LOCATION IN BUFFER
2734 040102 060204      173$:  ADD      R2,R4      ;SET POINTER TO FRAME (WORD)
2735 040104 021403      CMP      (R4),R3      ;CHECK DATA READ (R3=DATA ALSO)
2736 040106 001410      BEQ      180$      ;BR, IF ALL IS WELL
2737 040110 011401      MOV      (R4),R1      ;RECD DATA
2738 040112 010302      MOV      R3,R2      ;EXPECTED DATA
2739 040114 005237 002212      INC      FATFLG      ;BUMP COUNT
2743 040120      ERRHRD  ERRNO,T24DTA,EXPREC ;DATA READ NOT = WRITTEN
      TRAP      C$ERRHRD
      .WORD      447
      .WORD      T24DTA
      .WORD      EXPREC
2744 040130      180$:  CKLOOP      ;LOOP IF SELECTED
      TRAP      C$CLP1
      .WORD      104406
2745 040132 005724      TST      (R4)+      ;BUMP TO NEXT LOCATION
2746 040134 160204      SUB      R2,R4      ;GET RID OF BASE ADDRESS
2747 040136 020403      CMP      R4,R3      ;END OF RECORD YET
2748 040140 001360      BNE      173$      ;BR, IF NOT AT END OF RECORD
2749 040142 005743      TST      -(R3)      ;BUMP RECORD SIZE
2750 040144 022703 000400      CMP      #256.,R3      ;END OF RECORD YET
2751 040150 001322      BNE      165$      ;BR, IF MORE RECORDS TO WRITE
2752 040152      190$:  CKLOOP      ;LOOP IF SELECTED
      TRAP      C$CLP1
      .WORD      104406
2753 040154      ENDSUB      ;>>>>>>>>>> END SUBTEST >>>>>>>>>>
      .WORD      104403
      L10060:
2754 040156 023727 002212 000017      CMP      FATFLG,#15.      ;IS ERROR COUNT AT 25
2755 040164 103402      BLO      999$      ;BR, IF LESS THAN 25
2756 040166 004737 017272      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
2757 040172      999$:

```

TEST 4. BASIC READ DATA (FORWARD AND REVERSE)

```

2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2770
2771
2772 040172
040172
040172 104402
2773 040174 004737 046434
2774 040200 004737 046300
2775 040204 004737 046372
2776
2777
2778
2779
2780
2781
2782
2783 040210 004737 016064
2784 040214 103407
2785 040216 005237 002212
2789 040222 010001
2790 040224
040224 104455
040226 000700
040230 003650
040232 012124
2791 040234
2792 040234 013737 002172 043640
2793 040242 012704 043620
2794
2795
2796
2797
2798
2799
2800
2801 040246 004737 010752
2802 040252 103407
2803 040254 005237 002212
2807 040260 010001
2808 040262
040262 104456
040264 000701
040266 005054
040270 012124
2809 040272
040272 104406
2810
2811

```

;*
;TEST 4, SUBTEST 7
;VERIFIES THAT READ DATA COMMANDS WITH CVC-1 AND THE
;SWAP BYTES (SMB) BIT SET OPERATES PROPERLY. THE TEST
;SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2.
;THE RESULTS, EXCEPT FOR RAM CONTENTS, SHOULD BE THE SAME.
;
;
;
;
;
;
;
;
;
;*****
;ISSUE CONTROLLER "SOFT" INITIALIZE CARRY BIT CLEAR IF ERROR
;*****
;DO INITIALIZE ON CONTROLLER
;BR IF INIT WAS OK
;BUMP COUNT
;CONTENTS OF TSSR REGISTER
;FATAL ERROR TSSR WAS NOT OK
;
;*****
;WRITE CHARACTERISTICS COMMAND (CALL TO WRNCHR)
;*****
;ISSUE WRITE CHARACTERISTICS
;BR, IF COMMAND ISSUED OK
;BUMP COUNT
;SAVE CONTENTS OF TSSR
;WRITE CHARACTERISTICSC FAILED
;
;*****
;LOOP IF SELECTED
;*****

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2812
2813 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2814 ;
2815 ;*****
2816
2817 040274 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2818 040300 103407 BCS 30$ ;BR, IF NO PROBLEM
2819 040302 010001 MOV R0,R1 ;SAVE TSSR
2820 040304 005237 002212 INC FATFLG ;BUMP COUNT
2824 040310 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      040310 104456
      040312 000702 TRAP C$ERHRD
      040314 045076 .WORD 450
      040316 012136 .WORD T24RWN
      .WORD PKTSSR
2825 040320 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      040320 104406
2826 040322 012703 000400 OV #256.,R3 ;RECORD SIZE
2827 040326 013737 003114 043742 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
2828
2829 ;*****
2830 ;
2831 ;WRITE DATA,ACK,CVC=1,SWB COMMAND
2832 ;
2833 ;*****
2834
2835 040334 012737 150005 043740 MOV #150005,T24PK3 ;WRITE DATA,ACK,CVC=1,SWB COMMAND
2836 040342 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2837 040346 65$:
2838 040346 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
2839 040350 004737 017512 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
2840 040354 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
2841 040360 010465 000000 MOV R4,TSD8(R5) ;ISSUE COMMAND
2842 040364 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
2843 040370 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
2844 040374 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
2845 040400 020102 CMP R1,R2 ;ARE THEY EQUAL
2846 040402 001406 BEQ 75$ ;BR, IF OK
2847 040404 005237 002212 INC FATFLG ;BUMP COUNT
2851 040410 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      040410 104456 TRAP C$ERHRD
      040412 000703 .WORD 451
      040414 005111 .WORD WRERR
      040416 012136 .WORD PKTSSR
2852 040420 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      040420 104406
2853 040422 005723 TST (R3). ;BUMP RECORD SIZE
2854 040424 022703 000414 CMP #268.,R3 ;END OF RECORD YET
2855 040430 001346 BNE 65$ ;BR, IF MORE RECORDS TO WRITE
2856 040432 80$: CKLOOP ;LOOP IF SELECTED
      040432 104406 TRAP C$CLP1
2857 040434 005743 TST -(R3) ;SET RECORD SIZE BACK TO 512.
2858 040436 013737 003114 043742 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
2859
2860 ;*****
2861 ;
2862 ;READ REVERSE DATA,ACK,SWB COMMAND
2863 ;

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

2961
2962 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2963 ;
2964 ;*****
2965
2966 040730 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2967 040734 004737 016426 JSR PC,CHKTSSR ;SEE HOW TSSR IS
2968 040740 103407 BCS 30$ ;BR, IF NO PROBLEM
2969 040742 010001 MOV R0,R1 ;SAVE TSSR
2970 040744 005237 002212 INC FATFLG ;BUMP COUNT
2974 040750 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      040750 104456 TRAP C$ERRRD
      040752 000710 .WORD 456
      040754 045076 .WORD T24RWN
      040756 012136 .WORD PKTSSR
2975 040760 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      040760 104406
2976 040762 012703 001000 MOV #512.,R3 ;RECORD SIZE
2977 040766 013737 003114 043742 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
2978
2979 ;*****
2980 ;
2981 ;WRITE DATA,ACK,CVC=1 COMMAND
2982 ;
2983 ;*****
2984
2985 040774 012737 140005 043740 MOV #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
2986 041002 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2987 041006 65$:
2988 041006 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
2989 041012 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
2990 041016 004737 016340 ISR PC,WAITF ;WAIT FOR SSR TO SET
2991 041022 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
2992 041026 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
2993 041032 020102 CMP R1,R2 ;ARE THEY EQUAL
2994 041034 001406 BEQ 75$ ;BR, IF OK
2995 041036 005237 002212 INC FATFLG ;BUMP COUNT
2999 041042 ERRHRD ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      041042 104456 TRAP C$ERRRD
      041044 000711 .WORD 457
      041046 005111 .WORD WRTErr
      041050 012136 .WORD PKTSSR
3000 041052 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      041052 104406
3001 041054 012703 000400 MOV #256.,R3 ;SIZE OF RECORD
3002 041060 013737 003114 043742 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
3003
3004 ;*****
3005 ;
3006 ;READ DATA,ACK COMMAND
3007 ;
3008 ;*****
3009
3010 041066 012737 100401 043740 MOV #100401,T24PK3 ;READ DATA,ACK COMMAND
3011 041074 012704 043740 165$: MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
3012 041100 010337 043746 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
3013 041104 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3014 041110 004737 016340      JSR      PC,WAITF           ;WAIT FOR SSR TO SET
3015 041114 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
3016 041120 012702 100204      MOV      #SSR!SC!BIT2,R2  ;SET UP EXPECTED
3017 041124 020102              CMP      R1,R2             ;ARE THEY EQUAL
3018 041126 001406              BEQ      170$              ;BR, IF OK
3019 041130 005237 002212      INC      FATFLG            ;BUMP COUNT
3023 041134              ERRHRD  ERRNO,T24TRL,EXPREC ;TSSR INCORRECT AFTER READ DATA
      041134 104456              TRAP    C$ERHRD
      041136 000712              .WORD  458
      041140 046144              .WORD  T24TRL
      041142 015564              .WORD  EXPREC
3024 041144              170$: CKLOOP              ;LOOP IF SELECTED
      041144 104406              TRAP    C$CLP1
3025
3026
3027
3028
3029
3030
3031
      :*****
      :
      :READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
      :
      :*****
3032 041146 013701 043650      MOV      T24BFR+6,R1       ;GET MESSAGE BUFFER (XSTO)
3033 041152 010102              MOV      R1,R2             ;SET UP EXPECTED
3034 041154 052702 010000      BIS      #BIT12,R2         ;SET THE RLL BIT IN EXPECTED
3035 041160 020102              CMP      R1,R2             ;ARE THEY EQUAL
3036 041162 001406              BEQ      180$              ;BR, IF EQUAL (ALL IS WELL)
3037 041164 005237 002212      INC      FATFLG            ;BUMP COUNT
3041 041170              ERRHRD  ERRNO,T24LON,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
      041170 104456              TRAP    C$ERHRD
      041172 000713              .WORD  459
      041174 045712              .WORD  T24LON
      041176 015564              .WORD  EXPREC
3042 041200              180$: CKLOOP
      041200 104406              TRAP    C$CLP1
3043 041202              ENDSUB                      ;>>>>>>>>> END SUBTEST >>>>>>>>>
      041202 104403              L10062:
      041202 023727 002212 000017      CMP      FATFLG,#15        ;IS ERROR COUNT AT 25
3044 041204 023727 002212 000017      BLO     999$               ;BR, IF LESS THAN 25
3045 041212 103402              JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
3046 041214 004737 017272      999$:
3047 041220

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3049          ;*
3050          ;
3051          ;TEST 4, SUBTEST 9
3052          ;
3053          ;VERIFIES THAT A READ REVERSE COMMAND SPECIFYING A DATA
3054          ;BUFFER STARTING IN NONEXISTANT MEMORY TERMINATES WITH
3055          ;THE PROPER ERROR STATUS WITHOUT MOVING TAPE
3056          ;
3057          ;   *****
3058          ;                   CAUTION
3059          ;
3060          ;               The LSI BUS drivers for all available address lines(16 21)
3061          ;               are only checked when running on a 11/238 system with more than
3062          ;               128K words of memory!
3063          ;   *****
3064          ;
3065          041220          BGNSUB                                ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
3066          041220          104402                                T4.9:
3067          041220          005737          003126          TST          NXMFLG          ;DO WE HAVE IT?
3068          041226          001002          BNE          10$          ;BR, IF ENOUGH
3069          041230          000137          041616          JMP          180$          ;SKIP THIS TEST IF NOT
3070          041234          004737          046434          JSR          PC,T24RT3          ;SET UP OTHER COMMAND PACKET
3071          041240          004737          046300          JSR          PC,T24REST          ;SET COMMAND PACKET
3072          041244          004737          046372          JSR          PC,T24RT2          ;SET UP OTHER COMMAND PACKET
3073          ;
3074          ; *****
3075          ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
3076          ;
3077          ; *****
3078          ;
3079          041250          004737          016064          JSR          PC,SOFINIT          ;DO INITIALIZE ON CONTROLLER
3080          041254          103407          BCS          20$          ;BR IF INIT WAS OK
3081          041256          005237          002212          INC          FATFLG          ;BUMP COUNT
3082          041262          010001          MOV          R0,R1          ;CONTENTS OF TSSR REGISTER
3083          041264          104455          ERRDF          ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
3084          041266          000714          TRAP          C$ERDF          ;
3085          041270          003650          .WORD          460
3086          041272          012124          .WORD          SFIERR
3087          041274          012124          .WORD          SFIMSG
3088          041274          013737          002172          043640          20$: MOV          UNITN,T24DSW          ;SET DRIVE NUMBER IN PACKET
3089          041302          012704          043620          MOV          @T24PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
3090          ;
3091          ; *****
3092          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTHCR)
3093          ;
3094          ; *****
3095          ;
3096          ;
3097          041306          004737          010752          JSR          PC,WRTHCR          ;ISSUE WRITE CHARACTERISTICS
3098          041312          103407          BCS          24$          ;BR, IF COMMAND ISSUED OK
3099          041314          005237          002212          INC          FATFLG          ;BUMP COUNT
3100          041320          010001          MOV          R0,R1          ;SAVE CONTENTS OF TSSR
3101          041322          104456          ERRHRD          ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
3102          041322          TRAP          C$ERHRD

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

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

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3295      ;*
3296      ;
3297      ;TEST 4. SUBTEST 11
3298      ;
3299      ;VERIFIES THAT ILLEGAL BUFFER ADDRESSES CAUSE A
3300      ;FUNCTION REJECT TERMINATION WITH ILLEGAL ADDRESS
3301      ;(ILA) ERROR BIT SET.
3302      ;
3303      ;
3304      ;
3305      ;-
3306
3307      BGNSUB                                ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
                                          T4.11:
3308      042102      104402                      JSR      PC,T24RT3              ;SET COMMAND PACKET UP CLEAR  TRAP  C$BSUB
3309      042104      004737      046434          JSR      PC,T24REST           ;SET COMMAND PACKET
3310      042114      004737      046372          JSR      PC,T24RT2           ;SET UP OTHER COMMAND PACKET
3311
3312      ;*****
3313      ;
3314      ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
3315      ;
3316      ;*****
3317
3318      042120      004737      016064          JSR      PC,SOFINIT          ;DO INITIALIZE ON CONTROLLER
3319      042124      103407                      BCS      20$                 ;BR IF INIT WAS OK
3320      042126      005237      002212          INC      FATFLG              ;BUMP COUNT
3324      042132      010001                      MOV      R0,R1               ;CONTENTS OF TSSR REGISTER
3325      042134                      ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                          TRAP  C$ERDF
                                          .WORD 470
                                          .WORD SFIERR
                                          .WORD SFIMSG
3326      042144
3327      042144      013737      002172      043640  20$:  MOV      UNITN,T24DSW          ;SET UP DRIVE NUMBER
3328      042152      012704      043620          MOV      @T24PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
3329
3330      ;*****
3331      ;
3332      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
3333      ;
3334      ;*****
3335
3336      042156      004737      010752          JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
3337      042162      103407                      BCS      24$                 ;BR, IF COMMAND ISSUED OK
3338      042164      005237      002212          INC      FATFLG              ;BUMP COUNT
3342      042170      010001                      MOV      R0,R1               ;SAVE CONTENTS OF TSSR
3343      042172                      ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
                                          TRAP  C$ERHRD
                                          .WORD 471
                                          .WORD WRTMSG
                                          .WORD SFIMSG
3344      042202
3345      042202      104406                      24$:  CKLOOP                 ;LOOP IF SELECTED
                                          TRAP  C$CLP1
3346      042204      013737      003114      043742  MOV      FREE,T24RB          ;ILLEGAL STARTING READ BUFFER ADDRESS
3346      042212      012737      177700      043744  MOV      @177700,T24RB+2   ;CREATE ILLEGAL ADDRESS

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```
3347
3348 ;*****
3349 ;
3350 ;LEGAL MODE,ACK,CVC=1,READ COMMAND
3351 ;
3352 ;*****
3353
3354 042220 012737 140001 043740          MOV     #140001,T24PK3           ;LEGAL MODE,ACK,CVC=1,READ COMMAND
3355 042226 012704 043740                  MOV     #T24PK3,R4             ;SET UP R4 WITH PACKET ADDRESS
3356 042232 012737 000400 043746          MOV     #256.,T24SZ           ;SET UP RECORD SIZE IN PACKET
3357 042240 010465 000000                  MOV     R4,T24SZ              ;ISSUE COMMAND
3358 042244 004737 016340                  JSR     PC,WAITF              ;WAIT FOR SSR!BIT1!BIT2 TO SET
3359 042250 016501 000002                  MOV     TSSR(R5),R1           ;GET TSSR CONTENTS
3360 042254 012702 100206                  MOVB   #SSR!SC!BIT1!BIT2,R2  ;SET UP EXPECTED
3361 042260 020102                          CMP     R1,R2                 ;ARE THEY EQUAL
3362 042262 001406                          BEQ     75$                   ;BR, IF OK
3363 042264 005237 002212                  INC     FATFLG                ;BUMP COUNT
3367 042270 ERRHRD  ERRNO,T24WDG,PKTSSR          ;TSSR INCORRECT AFTER READ DATA
     042270 104456
     042272 000730                          TRAP   C$ERHRD
     042274 044112                          .WORD  472
     042276 012136                          .WORD  T24WDG
3368 042300 75$:   CKLOOP                    ;LOOP IF SELECTED              .WORD  PKTSSR
     042300 104406                          TRAP   C$CLP1
3369
3370 ;*****
3371 ;
3372 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
3373 ;
3374 ;*****
3375
3376 042302 013701 043650          MOV     T24BFR+6,R1           ;GET MESSAGE BUFFER
3377 042306 010102                  MOV     R1,R2                 ;SET UP EXPECTED
3378 042310 052702 000400          BIS     #BIT8,R2              ;SET THE ILA BIT IN EXPECTED
3379 042314 020102                  CMP     R1,R2                 ;ARE THEY EQUAL
3380 042316 001406                          BEQ     180$                   ;BR, IF EQUAL (ALL IS WELL)
3381 042320 005237 002212                  INC     FATFLG                ;BUMP COUNT
3385 042324 ERRHRD  ERRNO,T24ILA,EXPREC          ;THE ILA BIT WAS NOT SET IN XSTO
     042324 104456                          TRAP   C$ERHRD
     042326 000731                          .WORD  473
     042330 044342                          .WORD  T24ILA
     042332 015564                          .WORD  EXPREC
3386 042334 180$:  CKLOOP                    ;LOOP IF SELECTED              TRAP   C$CLP1
     042334 104406                          TRAP   C$CLP1
3387 042336 ENDSUB                    ;>>>>>>>>> END SUBTEST >>>>>>>>>
     042336 104403                          L10065:
     042336 023727 002212 000017          CMP     FATFLG,#15.           ;IS ERROR COUNT AT 25
3388 042340 103402                  BLO    999$                   ;BR, IF LESS THAN 25
3389 042346 004737 017272          JSR     PC,CKDROP             ;TRY TO DROP THE UNIT
3390 042350
3391 042354 999$:
```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410 042354          BGNSUB                    ;>>>>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>>>>
      042354          104402                    T4.12: TRAP C185 E
      042354 005737 003126                    TST     NXMFLG          ;DO WE HAVE IT?
3411 042356 005737 003126                    BNE     10$           ;BR, IF ENOUGH
3412 042362 001002                    JMP     80$           ;SKIP THIS TEST IF NOT
3413 042364 000137 042572                    10$: JSR     PC,T24RT3  ;SET COMMAND PACKET UP CLEAR
3414 042370 004737 046434                    JSR     PC,T24REST   ;SET COMMAND PACKET
3415 042374 004737 046300                    JSR     PC,T24RT2   ;SET UP OTHER COMMAND PACKET
3416 042400 004737 046372
3417
3418 ;.....
3419 ;
3420 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
3421 ;
3422 ;.....
3423
3424 042404 004737 016064                    JSR     PC,SOFINIT   ;DO INITIALIZE ON CONTROLLER
3425 042410 103407                    BCS     20$           ;BR IF INIT WAS OK
3426 042412 005237 002212                    INC     FATFLG       ;BUMP COUNT
3430 042416 010001                    MOV     R0,R1        ;CONTENTS OF TSSR REGISTER
3431 042420                    ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      042420 104455                    TRAP   C$ERRDF
      042422 000732                    .WORD  474
      042424 003650                    .WORD  SFIERR
      042426 012124                    .WORD  SFIMSG
3432 042430
3433 042430 013737 002172 043640 20$: MOV     UNITN,T24DSW  ;SET UP DRIVE NUMBER
3434 042436 012704 043620                    MOV     @T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
3435
3436 ;.....
3437 ;
3438 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
3439 ;
3440 ;.....
3441
3442 042442 004737 010752                    JSR     PC,WRTPHR    ;ISSUE WRITE CHARACTERISTICS
3443 042446 103407                    BCS     24$           ;BR, IF COMMAND ISSUED OK
3444 042450 005237 002212                    INC     FATFLG       ;BUMP COUNT
3448 042454 010001                    MOV     R0,R1        ;SAVE CONTENTS OF TSSR
3449 042456                    ERRHRD ERRNO,WRTPMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3535
3536 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
3537 ;
3538 ;*****
3539
3540 042712 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
3541 042716 004737 016426 JSR PC,CHKTSSR ;SEE HOW TSSR IS
3542 042722 103407 BCS 30$ ;BR, IF NO PROBLEM
3543 042724 010001 MOV R0,R1 ;SAVE TSSR
3544 042726 005237 002212 INC FATFLG ;BUMP COUNT
3548 042732 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      042732 104456
      042734 000737 TRAP C$ERRHRD
      042736 045076 .WORD 479
      042740 012136 .WORD T24RWN
3549 042742 30$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
      042742 104406 TRAP C$CLP1
3550 042744 012703 000400 MOV #256.,R3 ;RECORD SIZE
3551 042750 013737 003114 043742 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
3552
3553 ;*****
3554 ;
3555 ;READ REVERSE DATA,ACK COMMAND
3556 ;
3557 ;*****
3558
3559 042756 012737 100401 043740 MOV #100401,T24PK3 ;READ REVERSE DATA,ACK COMMAND
3560 042764 012704 043740 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
3561 042770
3562 042770 010337 043746 65$: MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
3563 042774 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
3564 043000 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
3565 043004 016501 006002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
3566 043010 012702 100206 MOV #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
3567 043014 020102 CMP R1,R2 ;ARE THEY EQUAL
3568 043016 001406 BEQ 75$ ;BR, IF OK
3569 043020 005237 002212 INC FATFLG ;BUMP COUNT
3573 043024 ERRHRD ERRNO,T24WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      043024 104456
      043026 000740 TRAP C$ERRHRD
      043030 044541 .WORD 480
      043032 012136 .WORD T24WDE
3574 043034 75$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
      043034 104406 TRAP C$CLP1
3575
3576 ;*****
3577 ;
3578 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
3579 ;
3580 ;*****
3581
3582 043036 013701 043650 MOV T24BFR+6,R1 ;GET MESSAGE BUFFER
3583 043042 010102 MOV R1,R2 ;SET UP EXPECTED
3584 043044 052702 002000 BIS #BIT10,R2 ;SET THE NEF BIT IN EXPECTED
3585 043050 020102 CMP R1,R2 ;ARE THEY EQUAL
3586 043052 001406 BEQ 180$ ;BR, IF EQUAL (ALL IS WELL)
3587 043054 005237 002212 INC FATFLG ;BUMP COUNT

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

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

```


TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

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

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3764
3765
3766
3768 043620
3770 043620 100204
3771 043620 043630
3772 043622 000000
3773 043624 000012
3774 043626 043642
3775 043630 000000
3776 043632 000024
3777 043634 000000
3778 043636 000000
3779 043640 000000
3780 043642
3781
3782
3783
3784
3786 043730
3788 043730 100206
3789 043732 043750
3790 043734 000000
3791 043736 000006
3792
3793
3797 043740
3798 043740 100205
3799 043742
3800 043742 003114
3801 043744 000000
3802 043746 000000
3803
3804
3805
3806
3807 043750
3808 043750 010
3809 043751 200
3810 043752 000000
3811 043754 000000
3812
3813
3814
3815
3816
3817 043756 100005
3818 043760 100405
3819 043762 102005
3820 043764 177777
3821 043766 000000
3822
3823

```

```

;
;LOCAL STORAGE FOR THIS TEST
;
;=<.10>&177770
T24PACKET: ;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
;DRIVE SELECTION BITS 2-0
;MESSAGE BUFFER
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
;=<.10>&177770
T24PK2: ;WRITE SUB SYS MEM COMMAND, IE AND ACK
;ADDRESS OF SELECT BLOCK DATA
;SIZE OF DATA PACKET
;READ COMMAND, IE AND ACK
;ADDRESS OF WRITE BUFFER
;SIZE OF BUFFER (EXTENT,
;
;
;
T24BF2:
T24BS0: .BYTE 10 ;BSELO AREA
T24BS1: .BYTE 200 ;BSEL1 AREA
T24S2: .WORD 0 ;SEL 2 AREA
T24S3: .WORD 0 ;DATA AREA
;
;
;EVEN
;TAPE MOTION PACKET COMMAND VALUES
T24RN: .WORD 100005 ;READ DATA (NEXT)
T24WDR: .WORD 100405 ;READ DATA RETRY
T24CON: .WORD 102005 ;WRITE CONTINOUS
;END OF DATA
T24DLY: .WORD 0 ;DELAY STORAGE AREA

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

3825
3826
3827 ;*
3828 ;LOCAL TEXT MESSAGES FOR TEST
3829 ;-
3830 043770 116 105 106 T24NEF: .ASCIZ 'NEF Not Set After NON-EXECUTABLE FUNCTION'
3831 044042 122 111 102 T24LOR: .ASCIZ 'RIB Not Set After READ REVERSE Into BOT'
3832 044112 124 123 123 T24WDG: .ASCIZ 'TSSR Not Correct After Illegal Buffer Address Bits Set'
3833 044201 124 123 123 T24NXM: .ASCIZ 'TSSR Not Correct After NXM Memory Address In Packet'
3834 044265 124 123 123 T24WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
3835 044342 111 154 154 T24ILA: .ASCIZ 'Illegal Address Bits, Failed To Set ILA Bit In XST0'
3836 044426 111 154 154 T24LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
3837 044507 122 105 101 T24SSR: .ASCIZ 'READ COMMAND Not Accepted'
3838 044541 124 123 123 T24WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
3839 044613 124 141 160 T24BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
3840 044660 104 141 164 T24DTA: .ASCIZ 'Data Written To Tape Not Equal To Data Read From Tape'
3841 044746 122 105 101 T24EOT: .ASCIZ 'READ DATA OVER EOT GAVE NO TAPE STATUS ALERT'
3842 045023 124 123 123 T24TM: .ASCIZ 'TSSR Not Correct After READ COMMAND Reject'
3843 045076 122 145 167 T24RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
3844 045145 122 101 115 T24RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
3845 045220 124 123 123 T24AM3: .ASCIZ 'TSSR Init. Failed After READ COMMAND'
3846 045265 104 162 151 T24OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
3847 045340 124 123 123 T24WDD: .ASCIZ 'TSSR Not Correct After READ DATA Command, SWB Bit Set'
3848 045426 124 123 123 T24WDC: .ASCIZ 'TSSR Not Correct After READ DATA Command'
3849 045477 103 126 103 T24VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
3850 045552 124 123 102 T24BA: .ASCIZ 'TSBA Not Correct After READ DATA Command'
3851 045623 127 122 111 T24WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
3852 045712 122 145 141 T24LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
3853 045774 122 145 141 T24LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
3854 046056 122 145 163 T24PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
3855 046144 122 145 141 T24TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
3856 046232 102 141 163 TST24ID: .ASCIZ 'Basic Read Data (Forward and Reverse)'
3857 .EVEN
3858 ;*
3859 ;
3860 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3861 ;WRITE SUBSYSTEM MEMORY COMMAND
3862 ;
3863 ;-
3864
3865 046300 T24REST:
3866 046300 SAVREG ;SAVE THE REGISTERS
3867 046304 012701 043620 MOV #T24PACKET,R1 ;START OF THE PACKET
3868 046310 012721 100004 MOV #100004,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK.
3869 046314 012721 043630 MOV #T24DATA,(R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
3870 046320 005021 CLR (R1)+ ;EXTENDED ADDRESS
3871 046322 012721 000012 MOV #10,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
3872 046326 012721 043642 MOV #T24BFR,(R1)+ ;ADDRESS OF MESSAGE BUFFER
3873 046332 005021 CLR (R1)+
3874 046334 012721 000024 MOV #20,(R1)+ ;LENGTH OF MESSAGE BUFFER
3875 046340 005021 CLR (R1)+
3876 046342 012711 000000 MOV #0,(R1) ;SELECT DRIVE ZERO
3877 046346 012702 000030 MOV #24,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
3878 046352 012762 177777 043642 64$: MOV #177777,T24BFR(R2) ;ALL ONES TO MESSAGE BUFFER
3879 046360 005742 TST -(R2) ;NEXT LOCATION
3880 046362 022702 000000 CMP #0,R2 ;CHECK FOR END OF LOOP
3881 046366 001371 BNE 64$ ;KEEP GOING UNTIL DONE

```

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

```

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

```


TEST 5: SPACE RECORDS

```

3965 046546 004737 016064      58:      JSR      PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
3966 046552 103427              BCS      108              ;BR IF INIT WAS OK
3967 046554              DELAY    250              ;DELAY IF REQUIRED
      046554 012727 000250              MOV      @250,(PC),.
      046560 000000              .WORD   0
      046562 013727 002116              MOV      @SDLY,(PC),.
      046566 000000              .WORD   0
      046570 005367 177772              DEC      6(PC)
      046574 001375              BNE     -.4
      046576 005367 177756              DEC     -22(PC)
      046602 001367              BNE     .20
3968 046604 005337 054032              DEC      T25DL1          ;DEC DELAY COUNTER
3969 046610 001356              BNE     58              ;BR, IF LOOP IS REQUIRED
3970 046612 005237 002212              INC      FATFLG          ;BUMP COUNT
3974 046616 016501 000002              MOV      TSSR(R5),R1     ;CONTENTS OF TSSR REGISTER
3975 046622              ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      046622 104455              TRAP    C$ERDF
      046624 000765              .WORD   501
      046626 003650              .WORD   SFIERR
      046630 012124              .WORD   SFIMSG
3976 046632              108:
3977 046632 013737 002172 053700      MOV      UNITN,T25DSW     ;SET UP DRIVE NUMBER
3978 046640 012704 053660      MOV      @T25PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
3979
3980 ;*****
3981 ;
3982 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
3983 ;
3984 ;*****
3985
3986 046644 004737 010752              JSR      PC,WRTPHR        ;ISSUE WRITE CHARACTERISTICS
3987 046650 103407              BCS      158              ;BR, IF COMMAND ISSUED OK
3988 046652 005237 002212              INC      FATFLG          ;BUMP COUNT
3992 046656 010001              MOV      R0,R1           ;SAVE CONTENTS OF TSSR
3993 046660              ERRMRD  ERRNO,WRTPMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      046660 104456              TRAP    C$ERRMRD
      046662 000766              .WORD   502
      046664 005054              .WORD   WRTPMSG
      046666 012124              .WORD   SFIMSG
3994
3995 ;*****
3996 ;
3997 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
3998 ;
3999 ;*****
4000
4001 046670              158:  CKLOOP
      046670 104406              TRAP    C$CLP1
4002 046672 004737 011104              JSR      PC,REWIND        ;CALL TAPE REWIND COMMAND
4003 046676 103407              BCS      308              ;BR, IF NO PROBLEM
4004 046700 010001              MOV      R0,R1           ;SAVE TSSR
4005 046702 005237 002212              INC      FATFLG          ;BUMP COUNT
4009 046706              ERRMRD  ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      046706 104456              TRAP    C$ERRMRD
      046710 000767              .WORD   503
      046712 055005              .WORD   T25RWN
      046714 012136              .WORD   PKTSSR

```

015

TEST 5: SPACE RECORDS

```

4010 046716 104406 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      046716
4011
4012 ;*****
4013 ;
4014 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
4015 ;
4016 ;*****
4017
4018 046720 013701 053710 MOV T25BFR-6,R1 ;PICK UP XSTO
4019 046724 010102 MOV R1,R2 ;SET UP EXPECTED
4020 046726 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
4021 046 32 020102 CMP R1,R2 ;DOES EXP = REC'D
4022 046734 001406 BEQ 40$ ;BR, IF EQUAL (OK)
4023 046736 005237 002212 INC FATFLG ;BUMP COUNT
4027 046742 ERRHRD ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      046742 104456 TRAP C$ERHRD
      046744 000770 .WORD 504
      046746 054175 .WORD T25BOT
      046750 015564 .WORD EXPREC
4028 046752 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      046752 104406
4029 046754 012703 000400 MOV #256.,R3 ;RECORD SIZE
4030 046760 013737 003114 054002 MOV FREE,T25RB ;STARTING WRITE BUFFER ADDRESS
4031
4032 ;*****
4033 ;
4034 ;WRITE DATA,ACK,CVC=1 COMMAND
4035 ;
4036 ;*****
4037
4038 046766 012737 140005 054000 MOV #140005,T25PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
4039 046774 012704 054000 MOV #T25PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
4040 047000
4041 047000 010337 054006 65$: MOV R3,T25SZ ;SET UP RECORD SIZE IN PACKET
4042 047004 013777 054030 134102 MOV T25CNT,#FREE ;LOAD UP RECORD COUNTER IN WRT BUFFER
4043 047012 062737 000001 054030 ADD #1,T25CNT ;GET READY FOR NEXT RECORD
4044 047020 010465 000000 MOV R4,T25DB(R5) ;ISSUE COMMAND
4045 047024 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
4046 047030 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4047 047034 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
4048 047040 020102 CMP R1,R2 ;ARE THEY EQUAL
4049 047042 001411 BEQ 75$ ;BR, IF OK
4050 047044 032701 000004 BIT #BIT2,R1 ;CHECK FOR TAPE STATUS ALERT
4051 047050 001014 BNE 120$ ;BR, IF TSA IS SET (SUSPECT IS EOT)
4052 047052 005237 002212 TNC FATFLG ;BUMP COUNT
4056 047056 ERRHRD ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      047056 104456 TRAP C$ERHRD
      047060 000771 .WORD 505
      047062 005111 .WORD WRTERR
      047064 012136 .WORD PKTSSR
4057 047066 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      047066 104406
4058 047070 005203 INC R3 ;BUMP RECORD SIZE
4059 047072 022703 001000 CMP #512.,R3 ;END OF RECORD YET
4060 047076 001340 BNE 65$ ;BR, IF MORE RECORDS TO WRITE
4061 047100 000415 BR 125$ ;ENOUGH RECORDS

```

TEST 5: SPACE RECORDS

015

```

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

```

TEST 5: SPACE RECORDS

```

047220 104406
4115 047222 005737 002216      TST      EXTFEA      ;CHECK FOR EXTENDED FEATURES
4116 047226 001044      BNE      160$      ;BR IF SWITCH IS ON
4117
4118 047230 112737 000200 054011      MOV      #200,T25BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
4119 047236 112737 000010 054010      MOV      #10,T25BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4120 047244 012704 053770      MOV      #T25PK2,R4 ;WRITE SUBSYS MEM PACKET
4121 047250 010465 000000      MOV      R4,T5D8(R5) ;ISSUE COMMAND
4122 047254 004737 016426      JSR      PC,CHKTSSR ;WAIT FOR SSR
4123 047260 103407      BCS      150$      ;BR, IF NO ERROR
4124 047262 010001      MOV      R0,R1      ;ERROR, SAVE TSSR
4125 047264 005237 002212      INC      FATFLG     ;BUMP COUNT
4129 047270      ERRHRD  ERRNO,T25SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      047270 104456      TRAP    C$ERHRD
      047272 000775      .WORD  509
      047274 054034      .WORD  T25SSR
      047276 012136      .WORD  PKTSSR
4130 047300      150$:  CKLOOP      ;LOOP IF SELECTED      TRAP    C$CLP1
      047300 104406
4131 047302 012737 000007 053700      MOV      #7,T25DSW ;SET UP DRIVE NUMBER
4132 047310 012704 053660      MOV      #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4133
4134      ;*****
4135      ;
4136      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
4137      ;
4138      ;*****
4139
4140 047314 004737 010752      JSR      PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
4141 047320 103407      BCS      160$      ;BR, IF COMMAND ISSUED OK
4142 047322 005237 002212      INC      FATFLG     ;BUMP COUNT
4146 047326 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4147 047330      ERRHRD  ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICS FAILED
      047330 104456      TRAP    C$ERHRD
      047332 000776      .WORD  510
      047334 005054      .WORD  WRTMSG
      047336 012124      .WORD  SFMSG
4148 047340      160$:  CKLOOP      ;SCOPE LOOP      TRAP    C$CLP1
      047340 104406
4149 047342 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
4150 047346 032701 000100      BIT      #0FL,R1    ;CHECK FOR THE OFFLINE BIT SET
4151 047352 001006      BNE      170$      ;BR, IF OFFLINE (GOOD)
4152 047354 005237 002212      INC      FATFLG     ;BUMP COUNT
4156 047360      ERRDF  ERRNO,T25OFL,SFMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
      047360 104455      TRAP    C$ERDF
      047362 000777      .WORD  511
      047364 055054      .WORD  T25OFL
      047366 012124      .WORD  SFMSG
4157 047370      170$:  CKLOOP      ;LOOP IF SELECTED      TRAP    C$CLP1
      047370 104406
4158
4159      ;*****
4160      ;
4161      ;SPACE FORWARD COMMAND IN PLACE
4162      ;
4163      ;*****
4164

```


TEST 5: SPACE RECORDS

```

4239
4240
4241 047572      ;*****
      047572 104406      15$:  CKLOOP
4242 047574 004737 011104      JSR  PC,REWIND      ;CALL TAPE REWIND COMMAND      TRAP  C$CLP1
4243 047600 103407      BCS  30$            ;BR, IF NO PROBLEM
4244 047602 010001      MOV  R0,R1          ;SAVE TSSR
4245 047604 005237 002212      INC  FATFLG        ;BUMP COUNT
4249 047610      ERRHRD  ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      047610 104456      ;
      047612 001003      ;
      047614 055005      ;
      047616 012136      ;
4250 047620      30$:  CKLOOP      ;LOOP IF SELECTED      TRAP  C$CLP1
      047620 104406      ;
4251 047622 005737 002216      140$: TST  EXTFEA    ;CHECK FOR EXTENDED FEATURES SW SWITCH
4252 047626 001044      BNE  160$          ;BR IF SWITCH IS ON
4253
4254 047630 112737 000200 054011      MOVB  #200,T25BS1  ;WRITE MISCELLANEOUS CONT/READ STATUS
4255 047636 112737 000010 054010      MOVB  #10,T25BS0  ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4256 047644 012704 053770      MOV  #T25PK2,R4   ;WRITE SUBSYS MEM PACKET
4257 047650 010465 000000      MOV  R4,TSDB(R5)  ;ISSUE COMMAND
4258 047654 004737 016426      JSR  PC,CHKTSSR   ;WAIT FOR SSR
4259 047660 103407      BCS  150$          ;BR, IF NO ERROR
4260 047662 010001      MOV  R0,R1        ;ERROR, SAVE TSSR
4261 047664 005237 002212      INC  FATFLG        ;BUMP COUNT
4265 047670      ERRHRD  ERRNO,T25SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      047670 104456      ;
      047672 001004      ;
      047674 054034      ;
      047676 012136      ;
4266 047700      150$: CKLOOP      ;LOOP IF SELECTED      TRAP  C$CLP1
      047700 104406      ;
4267 047702 012737 000007 053700      MOV  #7,T25DSW    ;SET UP DRIVE NUMBER
4268 047710 012704 053660      MOV  #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4269
4270
4271      ;*****
4272      ;
4273      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
4274      ;
4275      ;*****
4276 047714 004737 010752      JSR  PC,WRTPHR    ;ISSUE WRITE CHARACTERISTICS
4277 047720 103407      BCS  160$          ;BR, IF COMMAND ISSUED OK
4278 047722 005237 002212      INC  FATFLG        ;BUMP COUNT
4282 047726 010001      MOV  R0,R1        ;SAVE CONTENTS OF TSSR
4283 047730      ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
      047730 104456      ;
      047732 001005      ;
      047734 005054      ;
      047736 012124      ;
4284 047740      160$: CKLOOP      ;SCOPE LOOP      TRAP  C$CLP1
      047740 104406      ;
4285 047742 016501 000002      MOV  TSSR(R5),R1  ;GET TSSR CONTENTS
4286 047746 032701 000100      BIT  #0FL,R1      ;CHECK FOR THE OFFLINE BIT SET
4287 047752 001006      BNE  170$          ;BR, IF OFFLINE (GOOD)
4288 047754 005237 002212      INC  FATFLG        ;BUMP COUNT

```


TEST 5: SPACE RECORDS

```

4375
4376 050172          15$:  CKLOOP                                TRAP      C$CLP1
      050172 104406
4377 050174 004737 011104      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
4378 050200 103407          BCS      30$           ;BR, IF NO PROBLEM
4379 050202 010001          MOV      R0,R1         ;SAVE TSSR
4380 050204 005237 002212      INC      ATFLG        ;BUMP COUNT
4384 050210          ERRHRD  ERRNO,T25RWN,PKTSSR      ;REWIND NOT ACCEPTED
      050210 104456                                TRAP      C$ERHRD
      050212 001012                                .WORD    522
      050214 055005                                .WORD    T25RWN
      050216 012136                                .WORD    PKTSSR
4385 050220          30$:  CKLOOP                                ;LOOP IF SELECTED      TRAP      C$CLP1
      050220 104406
4386
4387      ;*****
4388      ;
4389      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
4390      ;
4391      ;*****
4392
4393 050222 013701 053710      MOV      T25BFR+6,R1  ;PICK UP XSTO
4394 050226 010102          MOV      R1,R2         ;SET UP EXPECTED
4395 050230 052702 000002      BIS      #BIT1,R2     ;SET BOT BIT IN EXPECTED
4396 050234 020102          CMP      R1,R2        ;DOES EXP = REC'D
4397 050236 001406          BEQ      40$           ;BR, IF EQUAL (OK)
4398 050240 005237 002212      INC      FATFLG        ;BUMP COUNT
4402 050244          ERRHRD  ERRNO,T25BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      050244 104456                                TRAP      C$ERHRD
      050246 001013                                .WORD    523
      050250 054175                                .WORD    T25BOT
      050252 015564                                .WORD    EXPREC
4403 050254          40$:  CKLOOP                                ;LOOP IF SELECTED      TRAP      C$CLP1
      050254 104406
4404 050256 012737 000001 054002 .  MOV      #000001,T25R8  ;NUMBER OF RECORDS TO SPACE OVER
4405
4406      ;*****
4407      ;
4408      ;SPACE FORWARD,ACK,CVC=1 COMMAND
4409      ;
4410      ;*****
4411
4412 050264 012737 140010 054000      MOV      #140010,T25PK3 ;SPACE FORWARD,ACK,CVC=1 COMMAND
4413 050272 012704 054000      MOV      #T25PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
4414 050276          65$:
4415 050276 010465 000000      MOV      R4,TSDB(R5)  ;ISSUE COMMAND
4416 050302 004737 016340      JSR      PC,WAITF     ;WAIT FOR SSR TO SET
4417 050306 016501 000002      MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
4418 050312 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
4419 050316 020102          CMP      R1,R2        ;ARE THEY EQUAL
4420 050320 001411          BEQ      75$           ;BR, IF OK
4421 050322 032701 000004      BIT      #BIT2,R1     ;CHECK FOR TAPE STATUS ALERT
4422 050326 001006          BNE      75$           ;BR, IF TSA IS SET (SUSPECT IS EOT)
4423 050330 005237 002212      INC      FATFLG        ;BUMP COUNT
4427 050334          ERRHRD  ERRNO,T25WDE,EXPREC      ;TSSR INCORRECT AFTER READ DATA
      050334 104456                                TRAP      C$ERHRD
      050336 001014                                .WORD    524

```

TEST 5: SPACE RECORDS

```

050340 054115 .WORD T25WDE
050342 015564 .WORD EXPREC
4428 050344 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
050344 104406
4429 050346 120$:
4430
4431 ;*****
4432 ;
4433 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
4434 ;
4435 ;*****
4436
4437 050346 013701 053710 MOV T25BFR+6,R1 ;QUICK CHECK FOR BOT SET
4438 050352 010102 MOV R1,R2 ;SET UP EXPECTED
4439 050354 042702 000002 BIC #BIT1,R2 ;CLEAR THE BOT BIT (XSTO)
4440 050360 020102 CMP R1,R2 ;IS THE EOT BIT SET IN XSTO
4441 050362 001406 BEQ 125$ ;BR, IF SET (GOOD)
4442 050364 005237 002212 INC FATFLG ;BUMP COUNT
4446 050370 ERRHRD ERRNO,T25BNC,EXPREC ;BOT NOT CLEARED AFTER SPACE FROM BOT
050370 104456 TRAP C$ERHRD
050372 001015 .WORD 525
050374 054470 .WORD T25BNC
050376 015564 .WORD EXPREC
4447 050400 125$: CKLOOP TRAP C$CLP1
050400 104406
4448 050402 004737 055352 JSR PC,T25RT3 ;CLEAN UP PACKET
4449 050406 012737 000401 054006 MOV #257.,T25SZ ;SET THE CORRECT SIZE UP
4450
4451 ;*****
4452 ;
4453 ;READ DATA COMMAND IN PLACE
4454 ;
4455 ;*****
4456
4457 050414 012737 140001 054000 MOV #140001,T25PK3 ;READ DATA COMMAND IN PLACE
4458 050422 013737 003114 054002 MOV FREE,T25RB ;READ BUFFER ADDRESS TO PACKET
4459 050430 012704 054000 MOV #T25PK3,R4 ;R4 = POINTER TO PACKET
4460 050434 010465 000000 MOV R4,T5DB(R5) ;ISSUE COMMAND
4461 050440 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
4462 050444 016501 000002 MOV T5SR(R5),R1 ;GET T5SR CONTENTS
4463 050450 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
4464 050454 020102 CMP R1,R2 ;ARE THEY EQUAL
4465 050456 001406 BEQ 190$ ;BR, IF OK ESP. FUNCTION REJECT
4466 050460 005237 002212 INC FATFLG ;BUMP COUNT
4470 050464 ERRHRD ERRNO,RDERR,PKTSSR ;T5SR INCORRECT AFTER READ DATA CMD
050464 104456 TRAP C$ERHRD
050466 001016 .WORD 526
050470 005204 .WORD RDERR
050472 012136 .WORD PKTSSR
4471 050474 190$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
050474 104406
4472 050476 017701 132412 MOV @FREE,R1 ;GET FIRST WORD FROM BUFFER
4473 050502 012702 000001 MOV #1,R2 ;SET UP EXPECTED
4474 050506 020102 CMP R1,R2 ;WAS RECORD NUMBERED 1
4475 050510 001406 BEQ 200$ ;BR, IF CORRECT RECORD
4476 050512 005237 002212 INC FATFLG ;BUMP COUNT
4480 050516 ERRHRD ERRNO,T25WNG,EXPREC ;SHOULD HAVE BEEN RECORD NUMBER 1

```


TEST 5: SPACE RECORDS

```

4540
4541
4542
4543 050650 004737 011104      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
4544 050654 103407              BCS      50$               ;BR, IF NO PROBLEM
4545 050656 010001              MOV      R0,R1             ;SAVE TSSR
4546 050660 005237 002212      INC      FATFLG            ;BUMP COUNT
4-50 050664              ERRHRD  ERRNO,T25RWN,PKT5,R ;REWIND NOT ACCEPTED
      050664 104456
      050666 001022              TRAP    C$ERRRD           ;
      050670 055005              .WORD  530                ;
      050672 012136              .WORD  T25RWN             ;
4551 050674              .WORD  PKT5,R             ;
      050674 104406      30$:   CKLOOP              ;LOOP IF SELECTED
                                      TRAP    C$CLP1           ;
4552
4553
4554
4555
4556
4557
4558
4559 050676 013701 053710      MOV      T25BFR+6,R1      ;PICK UP XSTO
4560 050702 010102              MOV      R1,R2            ;SET UP EXPECTED
4561 050704 052702 000002      BIS      @BIT1,R2         ;SET BOT BIT IN EXPECTED
4562 050710 020102              CMP      R1,R2            ;DOES EXP = REC'D
4563 050712 001406              BEQ      40$              ;BR, IF EQUAL (OK)
4564 050714 005237 002212      INC      FATFLG            ;BUMP COUNT
4568 050720              ERRHRD  ERRNO,T25BOT,E$PREC ;TAPE NOT AT BOT AFTER REWIND
      050720 104456              TRAP    C$ERRRD           ;
      050722 001023              .WORD  531                ;
      050724 054175              .WORD  T25BOT             ;
      050726 015564              .WORD  EXPREC            ;
4569 050730              40$:   CKLOOP              ;LOOP IF SELECTED
      050730 104406              TRAP    C$CLP1           ;
4570
4571
4572
4573
4574
4575
4576
4577
4578 050732 012703 000001      MOV      @000001,R3       ;NUMBER OF RECORDS TO SPACE FORWARD
4579 050736 004737 010556      JSR      PC,SPACE         ;CALL SPACE COMMAND
4580 050742 103410              BCS      50$               ;CHECK FOR ERROR
4581 050744 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
4582 050750 005237 002212      INC      FATFLG            ;BUMP COUNT
4586 050754              ERRHRD  ERRNO,T25WDE,SFFMSG ;SPACE FORWARD FAILED
      050754 104456              TRAP    C$ERRRD           ;
      050756 001024              .WORD  532                ;
      050760 054115              .WORD  T25WDE             ;
      050762 012172              .WORD  SFFMSG            ;
4587 050764              50$:   CKLOOP              ;LOOP IF SELECTED
      050764 104406              TRAP    C$CLP1           ;
4588 050766 012737 000001 054002  MOV      @1,T25RB         ;NUMBER OF RECORDS TO SPACE OVER
4589
4590

```

TEST 5: SPACE RECORDS

```

4591
4592 ;SPACE REVERSE,ACK,CVC-1 COMMAND
4593 ;
4594 ;*****
4595
4596 050774 012757 140410 054000      MOV    #140410,T25PK3      ;SPACE REVERSE,ACK,CVC-1 COMMAND
4597 051002 012704 054000      MOV    #T25PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4598 051006
4599 051006 010465 000000      65$:  MOV    R4,TSDB(R5)        ;ISSUE COMMAND
4600 051012 004737 016340      JSR    PC,WAITF          ;WAIT FOR SSR TO SET
4601 051016 016501 000002      MOV    TSSR(R5),R1      ;GET TSSR CONTENTS
4602 051022 012702 000200      MOV    #SSR,R2          ;SET UP EXPECTED
4603 051026 020102      CMP    R1,R2            ;ARE THEY EQUAL
4604 051030 001406      BEQ    75$              ;BR, IF OK
4605 051032 005237 002212      INC    FATFLG           ;BUMP COUNT
4609 051036      ERRHRD  ERRNO,T25WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP    C$ERHRD
                                .WORD  533
                                .WORD  T25WDE
                                .WORD  PKTSSR
                                TRAP    C$CLP1
4610 051046      75$:  CKLOOP                ;LOOP IF SELECTED
                                .WORD  533
                                .WORD  T25WDE
                                .WORD  PKTSSR
                                TRAP    C$CLP1
4611 051050
4612 051050 012703 000400      120$: MOV    #256,R3          ;RECORD SIZE
4613 051054 013737 003114 054002      MOV    FREE,T25RB       ;STARTING READ BUFFER ADDRESS
4614
4615 ;*****
4616 ;
4617 ;READ DATA,ACK,CVC-1 COMMAND
4618 ;
4619 ;*****
4620
4621 051062 012737 140001 054000      165$: MOV    #140001,T25PK3    ;READ DATA,ACK,CVC-1 COMMAND
4622 051070 012704 054000      MOV    #T25PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4623 051074 010337 054006      MOV    R3,T25SZ        ;SET UP RECORD SIZE IN PACKET
4624 051100 010465 000000      MOV    R4,TSDB(R5)    ;ISSUE COMMAND
4625 051104 004737 016340      JSR    PC,WAITF        ;WAIT FOR SSR TO SET
4626 051110 016501 000002      MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
4627 051114 012702 000200      MOV    #SSR,R2        ;SET UP EXPECTED
4628 051120 020102      CMP    R1,R2          ;ARE THEY EQUAL
4629 051122 001406      BEQ    170$           ;BR, IF OK
4630 051124 005237 002212      INC    FATFLG         ;BUMP COUNT
4634 051130      ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP    C$ERHRD
                                .WORD  534
                                .WORD  RDERR
                                .WORD  PKTSSR
                                TRAP    C$CLP1
4635 051140      170$: CKLOOP                ;LOOP IF SELECTED
                                .WORD  534
                                .WORD  RDERR
                                .WORD  PKTSSR
                                TRAP    C$CLP1
4636 051142 017701 131746      MOV    #FREE,R1        ;GET FIRST WORD FROM BUFFER
4637 051146 012702 000000      MOV    #0,R2           ;SET UP EXPECTED
4638 051152 020102      CMP    R1,R2          ;WAS RECORD NUMBERED 1
4639 051154 001406      BEQ    200$           ;BR, IF CORRECT RECORD
4640 051156 005237 002212      INC    FATFLG         ;BUMP COUNT
4644 051162      ERRHRD  ERRNO,T25WNG,EXPREC ;SHOULD HAVE BEEN RECORD NUMBER 1
                                TRAP    C$ERHRD
                                .WORD  535
                                .WORD  535

```


TEST 5: SPACE RECORDS

4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664

;-
;
;TEST 5, SUBTEST 5
;
;VERIFIES THAT SPACE RECORDS FORWARD CAN SPACE A
;MULTIPLE NUMBER OF RECORDS (2 THROUGH 64K OR THE
;MAXIMUM NUMBER OF RECORDS WRITTEN ON THE TAPE,
;WHICH EVER IS LESS.
;
;
;-

4665 051212

BGN SUB

; >>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
T5.5:

051212

051212

104402

TRAP C:BSJB

4666 051214 004737 055216
4667 051220 004737 055310
4668 051224 004737 055352
4669 051230 013737 054030 054026
4670 051236 162737 000002 054026
4671 051244 013737 054030 054032

JSR PC,T25REST
JSR PC,T25RT2
JSR PC,T25RT3
MOV T25CNT,T25CN2
SUB #2,T25CN2
MOV T25CNT,T25DLY

;SET COMMAND PACKET
;SET UP OTHER COMMAND PACKET
;SET UP OTHER COMMAND PACKET
;HOLD NUMBER OF RECORDS
;ACTUAL NUMBER OF RECORDS ON TAPE
;SET UP REWIND DELAY COUNTER

4672
4673
4674
4675
4676
4677
4678

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

4679 051252 004737 016064
4680 051256 103427
4681 051260

10\$: JSR PC,SOFINIT
BCS 20\$
DELAY 250

;DO INITIALIZE ON CONTROLLER
;BR IF INIT WAS OK
;WAIT ABOUT .25 SECONDS

051260 012727 000250
051264 000000
051266 013727 002116
051272 000000
051274 005367 177772
051300 001375
051302 005367 177756
051306 001367

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

4682 051310 005337 054032
4683 051314 001356
4684 051316 016501 000002
4685 051322 005237 002212
4689 051326

DEC T25DLY
BNE 10\$
MOV TSSR(R5),R1
INC FATFLG
ERRDF ERRNO,SFIERR,SFIMSG

;DEC COUNTER
;BR, IF MORE LOOPS REQUIRED
;CONTENTS OF TSSR REGISTER
;BUMP COUNT
;FATAL ERROR TSSR WAS NOT OK

051326 104455
051330 001030
051332 003650
051334 012124

TRAP C:ERDF
.WORD 536
.WORD SFIERR
.WORD SFIMSG

4690 051336
4691 051336 013737 002172 053700
4692 051344 012704 053660

20\$: MOV UNITN,T25DSW
MOV #T25PACKET,R4

;SET UP UNIT NUMBER
;SUBROUTINE NEEDS PACKET ADDRESS

4693
4694
4695
4696
4697

;*****
;
;WRITE CHARACTERISTICS COMMAND (CALL TO WRITCHR)
;

TEST 5: SPACE RECORDS

```

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

```

TEST 5: SPACE RECORDS

```

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

```


TEST 5: SPACE RECORDS

```

4804 ;*
4805 ;
4806 ;TEST 5, SUBTEST 6
4807 ;
4808 ;VERIFIES THAT SPACE RECORDS REVERSE CAN SPACE A
4809 ;MULTIPLE NUMBER OF RECORDS (2 THROUGH 64K OR THE
4810 ;MAXIMUM NUMBER OF RECORDS WRITTEN ON THE TAPE
4811 ;WHICH EVER IS LESS.
4812 ;
4813 ;
4814 ;
4815 ;-
4816 051752          BGNSUB                  ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
      051752                                T5.6:
      051752 104402                                TRAP          C$B5J8
4817 051754 004737 055216                      JSR          PC,T25REST        ;SET COMMAND PACKET
4818 051760 013737 054030 054026                MOV          T25CNT,T25CN2     ;SET UP RECORD COUNTER
4819 051766 004737 055310                      JSR          PC,T25RT2        ;SET UP OTHER COMMAND PACKET
4820 051772 004737 055352                      JSR          PC,T25RT3        ;SET UP OTHER COMMAND PACKET
4821 051776 013737 054030 054032                MOV          T25CNT,T25DLY     ;SET UP REWIND DELAY COUNTER
4822
4823 ;*****
4824 ;
4825 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
4826 ;
4827 ;*****
4828
4829 052004 004737 016064          10$: JSR          PC,SOFINIT        ;DO INITIALIZE ON CONTROLLER
4830 052010 103427                  BCS         20$               ;BR IF INIT WAS OK
4831 052012                      DELAY        250              ;WAIT ABOUT .25 SECONDS
      052012 012727 000250                                MOV          #250,(PC)*
      052016 000000                                .WORD       0
      052020 013727 002116                                MOV          L$DLY,(PC)*
      052024 000000                                .WORD       0
      052026 005367 177772                                DEC          -6(PC)
      052032 001375                                BNE         -.4
      052034 005367 177756                                DEC          -22(PC)
      052040 001367                                BNE         . 20
4832 052042 005337 054032          DEC          T25DLY             ;DEC COUNTER
4833 052046 001356          BNE         10$             ;BR, IF MORE LOOPS REQUIRED
4834 052050 016501 000002          MOV          TSSR(R5),R1        ;CONTENTS OF TSSR REGISTER
4835 052054 005237 002212          INC          FATFLG            ;BUMP COUNT
4839 052060          ERRDF      ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      052060 104455                                TRAP          C$ERDF
      052062 001036                                .WORD       542
      052064 003650                                .WORD       SFIERR
      052066 012124                                .WORD       SFIMSG
4840 052070 013737 002172 053700 20$: MOV          UNITN,T25DSW        ;SET UP UNIT NUMBER
4841
4842 052076 012704 053660          MOV          #T25PACKET,R4     ;SUBROUTINE NEEDS PACKET ADDRESS
4843
4844 ;*****
4845 ;
4846 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRCHR)
4847 ;
4848 ;*****
4849 ;

```

TEST 5: SPACE RECORDS

```

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

```

115

TEST 5: SPACE RECORDS

```

4901 052234 010337 054002          MOV      R3,T25RB          ;NUMBER OF RECORDS TO SPACE OVER
4902
4903          ;*****
4904          ;
4905          ;SPACE FORWARD,ACK,CVC=1 COMMAND
4906          ;
4907          ;*****
4908
4909 052240 012737 140010 054000    MOV      #140010,T25PK3    ;SPACE FORWARD,ACK,CVC=1 COMMAND
4910 052246 012704 054000          MOV      #T25PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4911 052252 010465 000000          MOV      R4,TSDB(R5)     ;ISSUE COMMAND
4912 052256 013737 054026 054032    MOV      T25CN2,T25DLY   ;SET UP DELAY COUNTER
4913 052264 004737 016340 48$:   JSR      PC,WAITF        ;WAIT FOR SSR TO SET
4914 052270 016501 000002          MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
4915 052274 012702 000200          MOV      #SSR,R2       ;SET UP EXPECTED
4916 052300 020102          CMP      R1,R2         ;ARE THEY EQUAL
4917 052302 001425          BEQ      50$          ;BR, IF OK
4918 052304          DELAY   250          ;WAIT .25 SECONDS
          MOV      #250,(PC)
          .WORD   0
          MOV      L$DLY,(PC)
          .WORD   0
          DEC      6(PC)
          BNE      .4
          DEC      22(PC)
          BNE      .20
4919 052334 005337 054032          DEC      T25DLY        ;DEC THE DELAY COUNTER
4920 052340 001351          BNE      48$          ;BR, IF COUNTER HASN'T EXPIRED
4921 052342 005237 002212          INC      FATFLG        ;BUMP COUNT
4925 052346          ERRHRD  ERRNO,T25WDE,EXPREC ;TSSR INCORRECT AFTER READ DATA
          TRAP   C$ERHRD
          .WORD  546
          .WORD  T25WDE
          .WORD  EXPREC
4926 052356 104406 50$:   CKLOOP          TRAP   C$CLP1
4927 052360 013701 054026          MOV      T25CN2,R1     ;NUMBER OF RECORDS ON TAPE
4928 052364 012702 177776          MOV      #65534.,R2   ;MAX IT CAN SPACE OVER
4929 052370 020201          CMP      R2,R1        ;WHICH VALUE CAN WE USE
4930 052372 003002          BGT      55$          ;BR, IF # WRITTEN > 64K
4931 052374 010103          MOV      R1,R3        ;# WRITTEN CAN BE USED
4932 052376 000401          BR      60$          ;MOVE ON
4933 052400 010203 55$:   MOV      R2,R3        ;USE MAX NUMBER
4934 052402 162703 000001 60$:   SUB      #1,R3        ;DON'T GO ALL THE WAY YET
4935 052406 010337 054002          MOV      R3,T25RB     ;NUMBER OF RECORDS TO SPACE OVER
4936
4937          ;*****
4938          ;
4939          ;SPACE REVERSE,ACK,CVC=1 COMMAND
4940          ;
4941          ;*****
4942
4943 052412 012737 140410 054000    MOV      #140410,T25PK3  ;SPACE REVERSE,ACK,CVC=1 COMMAND
4944 052420 012704 054000          MOV      #T25PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
4945 052424 010465 000000          MOV      R4,TSDB(R5)   ;ISSUE COMMAND
4946 052430 013737 054026 054032    MOV      T25CN2,T25DLY  ;SET UP COUNTER
4947 052436 004737 016340 70$:   JSR      PC,WAITF        ;WAIT FOR SSR TO SET

```

TEST 5: SPACE RECORDS

```

4948 052442 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
4949 052446 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
4950 052452 020102              CMP      R1,R2          ;ARE THEY EQUAL
4951 052454 001425              BEQ      75$            ;BR, IF OK
4952 052456              DELAY    250            ;WAIT ABOUT .25 SECONDS
                                MOV      #250,(PC)+
                                .WORD    0
                                MOV      L$DLY,(PC)+
                                .WORD    0
                                DEC      5(PC)
                                BNE      . 4
                                DEC      22(PC)
                                BNE      . 20
                                TRAP     C$ERHRD
                                .WORD    547
                                .WORD    T25WDE
                                .WORD    EXPREC
052456 012727 000250
052462 000000
052464 013727 002116
052470 000000
052472 005367 177772
052476 001375
052500 005367 177756
052504 001367
4953 052506 005337 054032      DEC      T25DLY          ;BUMP COUNTER DOWN
4954 052512 001351              BNE      70$            ;BR, IF COUNTER HASN T EXPIRED
4955 052514 005237 002212      INC      FATFLG          ;BUMP COUNT
4959 052520              ERRHRD   ERRNO,T25WDE,EXPREC ;TSSR INCORRECT AFTER READ DATA
                                TRAP     C$ERHRD
                                .WORD    547
                                .WORD    T25WDE
                                .WORD    EXPREC
                                TRAP     C$CLP1
052520 104456
052522 001043
052524 054115
052526 015564
4960 052530              75$:   CKLOOP          ;LOOP IF SELECTED
052530 104406
4961 052532 012703 010000      MOV      #4096.,R3      ;RECORD SIZE
4962 052536 013737 003114 054002      MOV      FREE,T25RB     ;STARTING READ BUFFER ADDRESS
4963
4964 ;*****
4965 ;
4966 ;READ DATA,ACK COMMAND
4967 ;
4968 ;*****
4969
4970 052544 012737 100001 054000      MOV      #100001,T25PK3 ;READ DATA,ACK COMMAND
4971 052552 012704 054000      165$:   MOV      #T25PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
4972 052556 012700 177777      MOV      #177777,R0     ;SET ALL ONES INTO CORRECT REGISTER
4973 052562 004737 017512      JSR      PC,FILLMEM     ;FILL MEMORY WITH RECORD SIZE
4974 052566 010337 054006      MOV      R3,T25SZ       ;SET UP RECORD SIZE IN PACKET
4975 052572 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
4976 052576 004737 016340      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
4977 052602 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
4978 052606 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
4979 052612 020102              CMP      R1,R2          ;ARE THEY EQUAL
4980 052614 001411              BEQ      170$           ;BR, IF OK
4981 052616 032701 000004      BIT      #BIT2,R1       ;CHECK FOR TAPE STATUS ALERT
4982 052622 001006              BNE      170$           ;BR, IF BIT SET
4983 052624 005237 002212      INC      FATFLG          ;BUMP COUNT
4987 052630              ERRHRD   ERRNO,RDERR,EXPREC ;TSSR INCORRECT AFTER READ DATA
                                TRAP     C$ERHRD
                                .WORD    548
                                .WORD    RDERR
                                .WORD    EXPREC
                                TRAP     C$CLP1
052630 104456
052632 001044
052634 005204
052636 015564
4988 052640              170$:   CKLOOP          ;LOOP IF SELECTED
052640 104406
4989 052642 017701 130246      MOV      #FREE,R1       ;GET FIRST WORD FROM BUFFER
4990 052646 012702 000001      MOV      #1,R2          ;SET UP EXPECTED
4991 052652 020102              CMP      R1,R2          ;WAS RECORD NUMBERED R3
4992 052654 001406              BEQ      200$           ;BR, IF CORRECT RECORD

```


TEST 5: SPACE RECORDS

```

5057 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5058 ;
5059 ;*****
5060
5061 053014 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
5062 053020 103407 BCS 30$ ;BR, IF NO PROBLEM
5063 053022 010001 MOV R0,R1 ;SAVE TSSR
5064 053024 005237 002212 INC FATFLG ;BUMP COUNT
5068 053030 ERRHRD ERRNO,T25RWN,PKTS,R ;REWIND NOT ACCEPTED
; TRAP C$ERHRD
; .WORD 552
; .WORD T25RWN
; .WORD PKTSSR
5069 053040 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
5070 ;
5071 ;*****
5072 ;
5073 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
5074 ;
5075 ;*****
5076
5077 053042 013701 053710 MOV T25BFR+6,R1 ;PICK UP YST0
5078 053046 010102 MOV R1,R2 ;SET UP EXPECTED
5079 053050 052702 000002 BIS @BIT1,R2 ;SET BOT BIT IN EXPECTED
5080 053054 020102 CMP R1,R2 ;DOES EXP = REC'D
5081 053056 001406 BEQ 40$ ;BR, IF EQUAL (OK)
5082 053060 005237 002212 INC FATFLG ;BUMP COUNT
5086 053064 ERRHRD ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
; TRAP C$ERHRD
; .WORD 553
; .WORD T25BOT
; .WORD EXPREC
5087 053074 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
5088 053076 012737 000001 054002 MOV @1,T25RB ;NUMBER OF RECORDS TO SPACE OVER
5089 ;
5090 ;*****
5091 ;
5092 ;SPACE REVERSE,ACK COMMAND
5093 ;
5094 ;*****
5095
5096 053104 012737 100410 054000 MOV @100410,T25PK3 ;SPACE REVERSE,ACK COMMAND
5097 053112 012704 054000 MOV @T25PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
5098 053116
5099 053116 010465 000000 65$: MOV R4,T5DB(R5) ;ISSUE COMMAND
5100 053122 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
5101 053126 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
5102 053132 012702 100206 MOV @SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
5103 053136 020102 CMP R1,R2 ;ARE THEY EQUAL
5104 053140 001406 BEQ 75$ ;BR, IF OK
5105 053142 005237 002212 INC FATFLG ;BUMP COUNT
5109 053146 ERRHRD ERRNO,T25WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
; TRAP C$ERHRD
; .WORD 554
; .WORD T25WDE

```

TEST 5: SPACE RECORDS

```

5110 053154 012136
5110 053156 104406
5111
5112
5113
5114
5115
5116
5117
5118 053160 013701 053710
5119 053164 010102
5120 053166 052702 002000
5121 053172 020102
5122 053174 001406
5123 053176 005237 002212
5127 053202
5127 053202 104456
5127 053204 001053
5127 053206 054643
5127 053210 015564
5128 053212
5128 053212 104406
5129 053214
5129 053214
5129 053214 104403
5130 053216 023727 002212 000017
5131 053224 103402
5132 053226 004737 017272
5133 053232

```

```

758: CKLOOP ;LOOP IF SELECTED
;*****
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;*****
MOV T25BFR+6,R1 ;GET XSTO STATUS WORD
MOV R1,R2 ;SET UP EXPECTED
BIS #BIT10,R2 ;SET THE NEF BIT
CMP R1,R2 ;ARE THEY EQUAL
BEQ 170$ ;BR. IF EQUAL (GOOD)
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T25NEF,EXPREC ;NEF SHOULD BE SET
TRAP C$ERHRD
;WORD 555
;WORD T25NEF
;WORD EXPREC
170$: CKLOOP TRAP C$CLP1
ENDSUB
L10100: TRAP C$ESUB
CMP FATFLG,#15. ;IS ERROR COUNT AT 25
BLO 999$ ;BR. IF LESS THAN 25
JSR PC,CKDROP ;TRY TO DROP THE UNIT
999$:

```


TEST 5: SPACE RECORDS

```

5187 ;*****
5188 ;
5189 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5190 ;
5191 ;*****
5192 ;
5193 053334 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
5194 053340 103407 BCS 30$ ;BR, IF NO PROBLEM
5195 053342 010001 MOV R0,R1 ;SAVE TSSR
5196 053344 005237 002212 INC FATFLG ;BUMP COUNT
5200 053350 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      053350 104456 TRAP C$ERHRD
      053352 001056 .WORD 558
      053354 055005 .WORD T25RWN
      053356 012136 .WORD PKTSSR
5201 053360 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      053360 104406
5202 ;*****
5203 ;
5204 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5205 ;
5206 ;*****
5207 ;
5208 ;
5209 053362 013701 053710 MOV T25BFR+6,R1 ;PICK UP XSTO
5210 053366 010102 MOV R1,R2 ;SET UP EXPECTED
5211 053370 052702 000002 BIS @BIT1,R2 ;SET BOT BIT IN EXPECTED
5212 053374 020102 CMP R1,R2 ;DOES EXP = REC'D
5213 053376 001406 BEQ 40$ ;BR, IF EQUAL (OK)
5214 053400 005237 002212 INC FATFLG ;BUMP COUNT
5218 053404 ERRHRD ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      053404 104456 TRAP C$ERHRD
      053406 001057 .WORD 559
      053410 054175 .WORD T25BOT
      053412 015564 .WORD EXPREC
5219 053414 40$: CKLOOP TRAP C$CLP1
      053414 104406
5220 053416 012737 000001 054002 MOV @1,T25RB ;NUMBER OF RECORDS TO SPACE OVER
5221 ;*****
5222 ;
5223 ;SPACE FORWARD,IE,ACK,CVC=1 COMMAND
5224 ;
5225 ;*****
5226 ;
5227 ;
5228 053424 012737 140210 054000 MOV @140210,T25PK3 ;SPACE FORWARD,IE,ACK,CVC=1 COMMAND
5229 053432 012704 054000 MOV @T25PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
5230 053436 010465 000000 MOV R4,T5DB(R5) ;ISSUE COMMAND
5231 053442 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
5232 053446 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
5233 053452 012702 000200 MOV @SSR,R2 ;SET UP EXPECTED
5234 053456 020102 CMP R1,R2 ;ARE THEY EQUAL
5235 053460 001406 BEQ 75$ ;BR, IF OK
5236 053462 005237 002212 INC FATFLG ;BUMP COUNT
5240 053466 ERRHRD ERRNO,T25WDE,EXPREC ;TSSR INCORRECT AFTER READ DATA
      053466 104456 TRAP C$ERHRD
      053470 001060 .WORD 560

```

TEST 5: SPACE RECORDS

053472	054115						.WORD	T25WDE
053474	015564						.WORD	EXPREC
5241	053476			75\$:	CKLOOP			; LOOP IF SELECTED
053476	104406							TRAP C\$CLP1
5242	053500	012737	000020	054002	MOV	#20,T25RB		; NUMBER OF RECORDS TO SPACE OVER
5243								
5244					:*****			
5245					:SPACE REVERSE,IE,ACK COMMAND			
5246								
5247					:*****			
5248								
5249								
5250	053506	012737	100610	054000	MOV	#100610,T25PK3		; SPACE REVERSE,IE,ACK COMMAND
5251	053514	012704	054000		MOV	#T25PK3,R4		; SET UP R4 WITH PACKET ADDRESS
5252	053520	010465	000000		MOV	R4,T25DB(R5)		; ISSUE COMMAND
5253	053524	004737	016340		JSR	PC,WAITF		; WAIT FOR SSR TO SET
5254	053530	016501	000002		MOV	T25SR(R5),R1		; GET T25SR CONTENTS
5255	053534	012702	100204		MOV	#SSR!BIT2!SC,R2		; SET UP EXPECTED
5256	053540	020102			CMP	R1,R2		; ARE THEY EQUAL
5257	053542	001406			BEQ	175\$; BR, IF OK
5258	053544	005237	002212		INC	FATFLG		; BUMP COUNT
5262	053550				ERRHRD	ERRNO,T25WDE,EXPREC		; T25SR INCORRECT AFTER READ DATA
053550	104456						TRAP	C\$ERHRD
053552	001061						.WORD	561
053554	054115						.WORD	T25WDE
053556	015564						.WORD	EXPREC
5263	053560			175\$:	CKLOOP			; LOOP IF SELECTED
053560	104406							TRAP C\$CLP1
5264	053562	013701	053716		MOV	T25BFR+14,R1		; GET XST3 STATUS WORD
5265	053566	010102			MOV	R1,R2		; SET UP EXPECTED
5266	053570	052702	000001		BIS	#BIT0,R2		; SET THE RIB BIT
5267	053574	020102			CMP	R1,R2		; ARE THEY EQUAL
5268	053576	001406			BEQ	180\$; BR, IF EQUAL (GOOD)
5269	053600	005237	002212		INC	FATFLG		; BUMP COUNT
5273	053604				ERRHRD	ERRNO,T25NEF,EXPREC		; NEF SHOULD BE SET
053604	104456						TRAP	C\$ERHRD
053606	001062						.WORD	562
053610	054643						.WORD	T25NEF
053612	015564						.WORD	EXPREC
5274	053614			180\$:	CKLOOP			
053614	104406							TRAP C\$CLP1
5275	053616				ENDSUB			; >>>>>>>>> END SUBTEST >>>>>>>>>
053616								TRAP C\$ESUB
053616	104403							L10101:
5276	053620	023727	002212	000017	CMP	FATFLG,#15.		; IS ERROR COUNT AT 25
5277	053626	103402			BLO	999\$; BR, IF LESS THAN 25
5278	053630	004737	017272		JSR	PC,CKDROP		; TRY TO DROP THE UNIT
5279	053634			999\$:				
5280				:				
5281				:				
5282				:				
5283	053634	004737	016546		JSR	PC,TSTLOOP		; DO WE NEED TO ITERATE TEST
5284	053640	103002			BCC	193\$; BR, IF NO LOOP REQUIRED
5285	053642	000137	046516		JMP	T25LOOP		; EXECUTE AGAIN
5286	053646			193\$:				
5287	053646				EXIT	TST		; ALL DONE THIS TEST
053646	104432						TRAP	C\$EXIT

H1

TSV7 HARDWARE TESTS 1 8

MACRO M1200 23 MAR 84 09:44 PAGE 69 3

SEQ 0214

TEST 5: SPACE RECORDS

053650 001532

.WORD L10071 .

TEST 5: SPACE RECORDS

```

5289
5290
5291
5293          053660
5295 053660  100004
5296 053660  053670
5297 053662  053670
5298 053664  000000
5299 053666  000010
5300 053670
5301 053670  053702
5302 053672  000000
5303 053674  000012
5304 053676  000000
5305 053700  000000
5306 053702
5307
5308
5309
5311          053770
5313 053770
5314 053770  100006
5315 053772  054010
5316 053774  000000
5317 053776  000006
5318
5322 054000
5323 054000  100005
5324 054002
5325 054002  003114
5326 054004  000000
5327 054006  000000
5328
5329
5330
5331
5332 054010
5333 054010    010
5334 054011    200
5335 054012  000000
5336 054014  000000
5337
5338
5339
5340
5341
5342 054016  100005
5343 054020  100405
5344 054022  102005
5345 054024  177777
5346
5347

```

```

;
;LOCAL STORAGE FOR THIS TEST
;
;
;=<.+10>E177770
T25PACKET:
;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;WORD 100004
;WORD T25DATA
;WORD 0
;WORD 8.
;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
T25DATA:
;WORD T25BFR
;WORD 0
;WORD 10.
;WORD 0
;LENGTH OF MESSAGE BUFFER
T25DSW: .WORD 0
;SELECT DRIVE ZERO
T25BFR: .BLKW 25.
;MESSAGE BUFFER
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
;=<.+10>E177770
T25PK2:
;WRITE SUB SYS MEM COMMAND, AND ACK
;ADDRESS OF SELECT BLOCK DATA
;WORD 100006
;WORD T25BFR2
;WORD 0
;WORD 6.
;SIZE OF DATA PACKET
T25PK3:
;WORD 100005
;READ COMMAND, AND ACK
T25RB:
;ADDRESS OF WRITE BUFFER
T25WB: .WORD FREE
;WORD 0
;WORD 0
;SIZE OF BUFFER (EXTENT)
;WORD 0
;EVEN
;
;
T25BF2:
;BSELO AREA
T25BS0: .BYTE 10
;BSEL1 AREA
T25BS1: .BYTE 200
;SEL 2 AREA
T25S2: .WORD 0
;DATA AREA
T25S3: .WORD 0
;
;
;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

```

TEST 5: SPACE RECORDS

```

5349 054026 000000      T25CN2: .WORD 0          ;COUNTER FOR RECORDS
5350 054030 000000      T25CNT: .WORD 0          ;COUNTER FOR RECORDS
5351 054032 000000      T25DLY: .WORD 0          ;COUNTER FOR RECORDS
5352
5353
5354                    ;*
5355                    ;LOCAL TEXT MESSAGES FOR TEST
5356                    ;-
5357 054034      127      122      111  T25SSR: .ASCIZ 'WRITE SUBSYSTEM Miscellaneous Read Status Failed'
5358 054115      124      123      123  T25WDE: .ASCIZ 'TSSR Not Correct After POSITION (SPACE) Command'
5359 054175      124      141      160  T25BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
5360 054242      124      123      123  T25TM: .ASCIZ 'TSSR Not Correct After POSITION (Space Command) Reject
5361 054331      127      162      151  T25NET: .ASCIZ 'Write Tape, Status Alert, But No EOT Sensed'
5362 054405      123      160      141  T25WNG: .ASCIZ 'Space Forward Failed To Position On Correct Record'
5363 054470      123      160      141  T25BNC: .ASCIZ 'Space Forward, From BOT, Failed To Clear BOT Indication
5364 054560      123      160      141  T25WNH: .ASCIZ 'Space Reverse Failed To Position On Correct Record'
5365 054643      123      160      141  T25NEF: .ASCIZ 'Space Reverse, At BOT, Failed To Set NEF (XST0)'
5366 054723      123      160      141  T25RIB: .ASCIZ 'Space Reverse, Into BOT, Failed To Set RIB (XST3)'
5367 055005      122      145      167  T25RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
5368 055054      104      162      151  T25OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
5369 055127      124      123      123  T25WDC: .ASCIZ 'TSSR Not Correct After READ DATA Command'
5370 055200      123      160      141  T25ID: .ASCIZ 'Space Records'
5371                    .EVEN
5372                    ;*
5373                    ;
5374                    ;ROUTINE TO RESTORE CMMAND PACKET TO START UP (DEFAULT) VALUES
5375                    ;WRITE SUBSYSTEM MEMORY COMMAND
5376                    ;
5377                    ;-
5378
5379 055216      T25REST:
5380 055216      SAVREG
5381 055222      012701  053660      MOV      #T25PACKET,R1      ;SAVE THE REGISTERS
5382 055226      012721  100004      MOV      #100004,(R1)+      ;START OF THE PACKET
5383 055232      012721  053670      MOV      #T25DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK
5384 055236      005021      CLR      (R1)+              ;ADDRESS OF CHARAISTICS DATA BLOCK
5385 055240      012721  000012      MOV      #10.,(R1)+         ;EXTENDED ADDRESS
5386 055244      012721  053702      MOV      #T25BFR,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
5387 055250      005021      CLR      (R1)+              ;ADDRESS OF MESSAGE BUFFER
5388 055252      012721  000024      MOV      #20.,(R1)+         ;LENGTH OF MESSAGE BUFFER
5389 055256      005021      CLR      (R1)+
5390 055260      012711  000000      MOV      #0,(R1)           ;SELECT DRIVE ZERO
5391 055264      012702  000030      MOV      #24.,R2           ;NUMBER OF LOCATIONS TO BE CLEARED
5392 055270      012762  177777  053702  64$:  MOV      #177777,T25BFR(R2) ;ALL ONES TO MESSAGE BUFFER
5393 055276      005742      TST      -(R2)              ;NEXT LOCATION
5394 055300      022702  000000      CMP      #0,R2             ;IS R2 AT ZERO YET
5395 055304      001371      BNE      64$                ;KEEP GOING UNTIL DONE
5396 055306      000207      RTS      PC                 ;RETURN
5397
5398 055310      T25RT2:
5399 055310      SAVREG
5400 055314      012701  053770      MOV      #T25PK2,R1        ;SAVE THE REGISTERS
5401 055320      012721  100006      MOV      #100006,(R1)+      ;START OF THE PACKET
5402 055324      012721  054010      MOV      #T25BFR2,(R1)+     ;WRITE SUBSYSTEM MEM. WITH ACK.
5403 055330      005021      CLR      (R1)+              ;ADDRESS OF DATA BLOCK
5404 055332      012721  000006      MOV      #6.,(R1)+         ;EXTENDED ADDRESS
5405 055336      005021      CLR      (R1)+              ;SIZE OF DATA BLOCK IN BYTES

```

KL

TEST 5: SPACE RECORDS

5406	055340	012701	054010
5407	055344	005021	
5408	055346	005011	
5409	055350	000207	
5410	055352		
5411	055352		
5412	055356	012701	054000
5413	055362	012721	000000
5414	055366	012721	000000
5415	055372	005021	
5416	055374	012721	000000
5417	055400	000207	
5418	055402		
	055402		
	055402	104401	

```

MOV    #T25BF2,R1      ;POINT TO DATA SEL AREA
CLR    (R1)+
CLR    (R1)
RTS    PC              ;RETURN

T25RT3: SAVREG         ;SAVE THE REGISTERS
MOV    #T25PK3,R1     ;START OF THE PACKET
MOV    #0,(R1)+       ;WRITE SUBSYSTEM MEM. WITH ACK,
MOV    #0,(R1)+       ;ADDRESS OF DATA BLOCK
CLR    (R1)+          ;EXTENDED ADDRESS
MOV    #0,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
RTS    PC              ;RETURN
ENDTST

```

```

L10071: TRAP    C$ETST

```


TEST 6: REREADS

```

5478
5479 ;ISSUE CONTROLLER "SOFT" INITIALIZE CARRY BIT CLEAR IF ERROR
5480 ;
5481 ;*****
5482
5483 055470 004737 016064 10$: JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
5484 055474 103426 BCS 20$ ;BR IF INIT WAS OK
5485 055476 DELAY 250 ;DELAY FOR A REWIND TO FINISH
055476 012727 000250 MOV #250,(PC)+
055502 000000 .WORD 0
055504 013727 002116 MOV L$DLY,(PC)+
055510 000000 .WORD 0
055512 005367 177772 DEC -6(PC)
055516 001375 BNE -4
055520 005367 177756 DEC -22(PC)
055524 001367 BNE . 20
5486 055526 005337 072004 DEC T26DLY ;DEC COUNTER
5487 055532 001356 BNE 10$ ;BR, IF DELAY NOT READY
5488 055534 005237 002212 INC FATFLG ;BUMP COUNT
5492 055540 010001 MOV RO,R1 ;CONTENTS OF TSSR REGISTER
5493 055542 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
055542 104455 TRAP C$ERDF
055544 001131 .WORD 601
055546 003650 .WORD SFIERR
055550 012124 .WORD SFIMSG
5494 055552
5495 055552 013737 002172 071650 20$: MOV UNITN,T26DSW ;SET UP UNIT NUMBER
5496 055560 012704 071630 MOV #T26PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
5497
5498 ;*****
5499 ;
5500 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
5501 ;
5502 ;*****
5503
5504 055564 004737 010752 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
5505 055570 103407 BCS 26$ ;BR, IF COMMAND ISSUED OK
5506 055572 005237 002212 INC FATFLG ;BUMP COUNT
5510 055576 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
5511 055600 ERPHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
055600 104456 TRAP C$ERHRD
055602 001132 .WORD 602
055604 005054 .WORD WRTMSG
055606 012124 .WORD SFIMSG
5512 055610 26$: CKLOOP ;LOOP IF SELECTED
055610 104406 TRAP C$CLP1
5513
5514 ;*****
5515 ;
5516 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5517 ;
5518 ;*****
5519
5520 055612 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
5521 055616 103413 BCS 30$ ;BR, IF NO PROBLEM
5522 055620 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
5523 055624 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR

```

TEST 6: REREADS

```

5524 055630 010004          MOV    R0,R4          ;PACKET ADDRESS SET UP
5525 055632 005237 002212  INC    FATFLG        ;BUMP COUNT
5529 055636          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      055636 104456          TRAP   C$ERHRD
      055640 001133          .WORD 603
      055642 073304          .WORD T26RWN
      055644 012136          .WORD PKTSSR
5530 055646          30$:  CKLOOP          ;LOOP IF SELECTED
      055646 104406          TRAP   C$CLP1
5531
5532          ;*****
5533          ;
5534          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5535          ;
5536          ;*****
5537
5538 055650 013701 071660          MOV    T26BFR+6,R1    ;PICK UP XSTO
5539 055654 010102          MOV    R1,R2          ;SET UP EXPECTED
5540 055656 052702 000002          BIS    #BIT1,R2      ;SET BOT BIT IN EXPECTED
5541 055662 020102          CMP    R1,R2          ;DOES EXP = REC'D
5542 055664 001406          BEQ    40$           ;BR, IF EQUAL (OK)
5543 055666 005237 002212  INC    FATFLG        ;BUMP COUNT
5547 055672          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      055672 104456          TRAP   C$ERHRD
      055674 001134          .WORD 604
      055676 073015          .WORD T26BOT
      055700 015564          .WORD EXPREC
5548 055702          40$:  CKLOOP          ;LOOP IF SELECTED
      055702 104406          TRAP   C$CLP1
5549 055704 012703 000400          MOV    #256.,R3      ;RECORD SIZE
5550 055710 013737 003114 071752  MOV    FREE,T26RB    ;STARTING WRITE BUFFER ADDRESS
5551
5552          ;*****
5553          ;
5554          ;WRITE DATA,ACK,CVC=1 COMMAND
5555          ;
5556          ;*****
5557
5558 055716 012737 140005 071750          MOV    #140005,T26PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
5559 055724 012704 071750          MOV    #T26PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
5560 055730          65$:
5561 055730 010300          MOV    R3,R0          ;SET PATTERN IN CORRECT REGISTER
5562 055732 004737 017512          JSR    PC,FILLMEM    ;FILL MEMORY WITH RECORD SIZE
5563 055736 010337 071756          MOV    R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
5564 055742 010465 000000          MOV    R4,TSDB(R5)   ;ISSUE COMMAND
5565 055746 004737 016340          JSR    PC,WAITF      ;WAIT FOR SSR TO SET
5566 055752 016501 000002          MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
5567 055756 012702 000200          MOV    #SSR,R2       ;SET UP EXPECTED
5568 055762 020102          CMP    R1,R2          ;ARE THEY EQUAL
5569 055764 001406          BEQ    75$           ;BR, IF OK
5570 055766 005237 002212  INC    FATFLG        ;BUMP COUNT
5574 055772          ERRHRD  ERRNO,WRTErr,EXPREC ;TSSR INCORRECT AFTER WRITE DATA
      055772 104456          TRAP   C$ERHRD
      055774 001135          .WORD 605
      055776 005111          .WORD WRTErr
      056000 015564          .WORD EXPREC
5575 056002          75$:  CKLOOP          ;LOOP IF SELECTED

```

TEST 6: REREADS

```

056002 104406
5576 056004 005723
5577 056006 022703 000414
5578 056012 001346
5579 056014 104406
80$: CKLOOP
5580 056016 120$:
5581
5582
5583
5584
5585
5586
5587
5588 056016 004737 011104
5589 056022 103413
5590 056024 016501 000002
5591 056030 012702 000200
5592 056034 010004
5593 056036 005237 002212
5597 056042
056042 104456
056044 001136
056046 073304
056050 012136
5598 056052 130$: CKLOOP
056052 104406
5599
5600
5601
5602
5603
5604
5605
5606 056054 013701 071660
5607 056060 010102
5608 056062 052702 000002
5609 056066 020102
5610 056070 001406
5611 056072 005237 002212
5615 056076
056076 104456
056100 001137
056102 073015
056104 012136
5616 056106 140$: CKLOOP
056106 104406
5617 056110 012737 000400 072002
5618
5619
5620
5621
5622
5623
5624
5625
5626 056116 012703 000001 145$: MOV #1,R3

```

```

;BUMP RECORD SIZE
;END OF RECORD YET
;BR, IF MORE RECORDS TO WRITE
;LOOP IF SELECTED
TRAP C$CLP1

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

;CALL TAPE REWIND COMMAND
;BR, IF NO PROBLEM
;GET TSSR
;SET UP EXPECTED TSSR
;PACKET ADDRESS SET UP
;BUMP COUNT
;REWIND NOT ACCEPTED
TRAP C$ERRRD
.WORD 606
.WORD T26RWN
.WORD PKTSSR

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

;PICK UP XSTO
;SET UP EXPECTED
;SET BOT BIT IN EXPECTED
;DOES EXP = REC'D
;BR, IF EQUAL (OK)
;BUMP COUNT
;TAPE NOT AT BOT AFTER REWIND
TRAP C$ERRRD
.WORD 607
.WORD T26BOT
.WORD PKTSSR

;SET RECORD SIZE
TRAP C$CLP1

;*****
;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
;*****

;SPACE ONE RECORD PARAMETER

```

TEST 6: REREADS

```

5627 056122 004737 010556      JSR    PC,SPACE      ;CALL SPACE ROUTINE
5628 056126 103412      BCS    150$          ;BR, IF NO PROBLEM WITH SPACE COMMAND
5629 056130 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR
5630 056134 012702 000200      MOV    #SSR,R2       ;SET UP EXPECTED TSSR
5631 056140 005237 002212      INC    FATFLG        ;BUMP COUNT
5635 056144      ERRHRD  ERRNO,T26SC,FYPREC ;POSITION (SPACE RECORDS) FAILED
      056144 104456      TRAP   C$ERRHRD
      056146 001140      .WORD  608
      056150 07241$      .WORD  T26SC
      056152 015564      .WORD  EYPREC
5636 056154      150$:  CKLOOP      TRAP   C$CLP1
      056154 104406
5637 056156 013703 072002      MOV    T26RSZ,R3     ;RECORD SIZE
5638 056162 013737 003114 071752      MOV    FREE,T26RB    ;STARTING READ BUFFER ADDRESS
5639
5640      ;*****
5641      ;
5642      ;REREREAD DATA,CVC=1,ACK COMMAND
5643      ;
5644      ;*****
5645
5646 056170 012737 141001 071750      MOV    #141001,T26PK3 ;REREREAD DATA,CVC=1,ACK COMMAND
5647 056176 012704 071750      165$:  MOV    #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
5648 056202 010337 071756      MOV    R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
5649 056206 010465 000000      MOV    R4,T26SDB(R5) ;ISSUE COMMAND
5650 056212 004737 016340      JSR    PC,WAITF      ;WAIT FOR SSR TO SET
5651 056216 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
5652 056222 012702 000200      MOV    #SSR,R2       ;SET UP EXPECTED
5653 056226 020102      CMP    R1,R2         ;ARE THEY EQUAL
5654 056230 001406      BEQ    170$          ;BR, IF OK
5655 056232 005237 002212      INC    FATFLG        ;BUMP COUNT
5659 056236      ERRHRD  ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      056236 104456      TRAP   C$ERRHRD
      056240 001141      .WORD  609
      056242 073640      .WORD  T26WDC
      056244 012136      .WORD  PKTSSR
5660 056246      170$:  CKLOOP      ;LOOP IF SELECTED
      056246 104406      TRAP   C$CLP1
5661 056250 013702 003114      MOV    FREE,R2       ;CURRENT BUFFER ADDRESS TO R2
5662 056254 010304      MOV    R3,R4         ;CURRENT RECORD SIZE
5663 056256 162704 000400      SUB    #256.,R4      ;FIRST LOCATION IN BUFFER
5664 056262 060204      173$:  ADD    R2,R4         ;SET UP POINTER
5665 056264 021403      CMP    (R4),R3       ;CHECK DATA READ (R3=DATA ALSO)
5666 056266 001410      BEQ    180$          ;BR, IF ALL IS WELL
5667 056270 011401      MOV    (R4),R1       ;RECD DATA
5668 056272 010302      MOV    R3,R2         ;EXPECTED DATA
5669 056274 005237 002212      INC    FATFLG        ;BUMP COUNT
5673 056300      ERRHRD  ERRNO,T26DTA,EXPREC ;DATA READ NOT = WRITTEN
      056300 104456      TRAP   C$ERRHRD
      056302 001142      .WORD  610
      056304 073062      .WORD  T26DTA
      056306 015564      .WORD  EXPREC
5674 056310      180$:  CKLOOP      ;LOOP IF SELECTED
      056310 104406      TRAP   C$CLP1
5675 056312 005724      TST    (R4),         ;BUMP TO NEXT LOCATION
5676 056314 160204      SUB    R2,R4         ;CORRECT RECORDS SIZE VALUE
5677 056316 020403      CMP    R4,R3        ;END OF RECORD YET

```


(D.)

SEQ 0223

TEST 6: REREADS

```

5678 056320 001360          BNE     173$           ;BR, IF NOT AT END OF RECORD
5679 056322 005723          TST     (R3).         ;BUMP RECORD SIZE
5680 056324 010337 072002    MOV     R3,T26RSZ     ;RESET RECORD SIZE
5681 056330 022703 000412    CMP     #266.,R3     ;END OF RECORD YET
5682 056334 001270          BNE     145$           ;BR, IF MORE RECORDS TO READ
5683 056336          190$:    CKLOOP   ;LOOP IF SELECTED
    056336 104406
5684 056340          ENDSUB  ;TRAP C#CLP1
    056340
    056340 104403
5685 056342 023727 002212 000017  CMP     FATFLG,#15.   ;IS ERROR COUNT AT 25
5686 056350 103402          BLO     999$         ;BR, IF LESS THAN 25
5687 056352 004737 017272          JSR     PC,CKDROP    ;TRY TO DROP THE UNIT
5688 056356          999$:
  
```


TEST 6: REREADS

```

5742 ;*****
5743 ;
5744 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5745 ;
5746 ;*****
5747 ;
5748 056460 004737 011104      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
5749 056464 103413              BCS      30$            ;BR, IF NO PROBLEM
5750 056466 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR
5751 056472 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED TSSR
5752 056476 010004              MOV      R0,R4         ;PACKET ADDRESS SET UP
5753 056500 005237 002212      INC      FATFLG        ;BUMP COUNT
5757 056504              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                    056504 104456              TRAP    C$ERHRD
                    056506 001145              .WORD  613
                    056510 073304              .WORD  T26RWN
                    056512 012136              .WORD  PKTSSR
5758 056514              30$:   CKLOOP                ;LOOP IF SELECTED
                    056514 104406              TRAP    C$CLP1
5759 ;*****
5760 ;
5761 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5762 ;
5763 ;*****
5764 ;
5765 ;
5766 056516 013701 071660      MOV      T26BFR+6,R1   ;PICK UP XSTO
5767 056522 010102              MOV      R1,R2         ;SET UP EXPECTED
5768 056524 052702 000002      BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
5769 056530 020102              CMP      R1,R2         ;DOES EXP = REC'D
5770 056532 001406              BEQ      40$           ;BR, IF EQUAL (OK)
5771 056534 005237 002212      INC      FATFLG        ;BUMP COUNT
5775 056540              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                    056540 104456              TRAP    C$ERHRD
                    056542 001146              .WORD  614
                    056544 073015              .WORD  T26BOT
                    056546 015564              .WORD  EXPREC
5776 056550              40$:   CKLOOP                ;LOOP IF SELECTED
                    056550 104406              TRAP    C$CLP1
5777 056552 012703 000400      MOV      #256.,R3      ;RECORD SIZE
5778 056556 013737 003114 071752  MOV      FREE,T26RB     ;STARTING WRITE BUFFER ADDRESS
5779 ;*****
5780 ;
5781 ;WRITE DATA,ACK,SWB COMMAND
5782 ;
5783 ;*****
5784 ;
5785 ;
5786 056564 012737 110005 071750      MOV      #110005,T26PK3 ;WRITE DATA,ACK,SWB COMMAND
5787 056572 012704 071750              MOV      #T26PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
5788 056576              65$:
5789 056576 010300              MOV      R3,R0         ;SET PATTERN IN CORRECT REGISTER
5790 056600 004737 017512      JSR      PC,FILLMEM    ;FILL MEMORY WITH RECORD SIZE
5791 056604 010337 071756      MOV      R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
5792 056610 010465 000000              MOV      R4,TSDB(R5)   ;ISSUE COMMAND
5793 056614 004737 016340      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
5794 056620 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS

```

TEST 6: REREADS

```

5795 056624 012702 000200      MOV     #SSR,R2      ;SET UP EXPECTED
5796 056630 020102      CMP     R1,R2       ;ARE THEY EQUAL
5797 056632 001406      BEQ     75$         ;BR, IF OK
5798 056634 005237 002212      INC     FATFLG      ;BUMP COUNT
5802 056640      ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      056640 104456      TRAP   C$ERRHRD
      056642 001147      .WORD  615
      056644 005111      .WORD  WRTERR
      056646 012136      .WORD  PKTSSR
5803 056650      75$:   CKLOOP      ;LOOP IF SELECTED
      056650 104406      TRAP   C$CLP1
5804 056652 005723      TST     (R3)+       ;BUMP RECORD SIZE
5805 056654 022703 000414      CMP     #268.,R3   ;END OF RECORD YET
5806 056660 001346      BNE     65$         ;BR, IF MORE RECORDS TO WRITE
5807 056662      80$:   CKLOOP      ;LOOP IF SELECTED
      056662 104406      TRAP   C$CLP1
5808 056664      120$:
5809
5810      ;*****
5811      ;
5812      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5813      ;
5814      ;*****
5815
5816 056664 004737 011104      JSR     PC,REWIND   ;CALL TAPE REWIND COMMAND
5817 056670 103413      BCS     130$       ;BR, IF NO PROBLEM
5818 056672 016501 000002      MOV     TSSR(R5),R1 ;GET TSSR
5819 056676 012702 000200      MOV     #SSR,R2    ;SET UP EXPECTED TSSR
5820 056702 010004      MOV     R0,R4      ;PACKET ADDRESS SET UP
5821 056704 005237 002212      INC     FATFLG      ;BUMP COUNT
5825 056710      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      056710 104456      TRAP   C$ERRHRD
      056712 001150      .WORD  616
      056714 073304      .WORD  T26RWN
      056716 012136      .WORD  PKTSSR
5826 056720      130$:  CKLOOP      ;LOOP IF SELECTED
      056720 104406      TRAP   C$CLP1
5827
5828      ;*****
5829      ;
5830      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5831      ;
5832      ;*****
5833
5834 056722 013701 071660      MOV     T26BFR+6,R1 ;PICK UP XSTO
5835 056726 010102      MOV     R1,R2      ;SET UP EXPECTED
5836 056730 052702 000002      BIS     #BIT1,R2   ;SET BOT BIT IN EXPECTED
5837 056734 020102      CMP     R1,R2      ;DOES EXP = REC'D
5838 056736 001406      BEQ     140$       ;BR, IF EQUAL (OK)
5839 056740 005237 002212      INC     FATFLG      ;BUMP COUNT
5843 056744      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      056744 104456      TRAP   C$ERRHRD
      056746 001151      .WORD  617
      056750 073015      .WORD  T26BOT
      056752 015564      .WORD  EXPREC
5844 056754      140$:  CKLOOP      ;LOOP IF SELECTED
      056754 104406      TRAP   C$CLP1

```

TEST 6: REREADS

```

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

```


TEST 6: REREADS

```

5972
5973 057300 012704 071630          MOV      #T26PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
5974
5975          ;*****
5976          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
5977          ;
5978          ;*****
5979
5980
5981 057304 004737 010752          JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
5982 057310 103407          BCS      26$              ;BR, IF COMMAND ISSUED OK
5983 057312 005237 002212          INC      FATFLG          ;BUMP COUNT
5987 057316 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
5988 057320          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
          057320 104456          TRAP      C$ERHRD
          057322 001156          .WORD    622
          057324 005054          .WORD    WRTMSG
          057326 012124          .WORD    SFIMSG
5989 057330          26$:   CKLOOP          ;LOOP IF SELECTED
          057330 104406          TRAP      C$CLP1
5990
5991          ;*****
5992          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5993          ;
5994          ;*****
5995
5996
5997 057332 004737 011104          JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
5998 057336 103413          BCS      30$              ;BR, IF NO PROBLEM
5999 057340 016501 000002          MOV      TSSR(R5),R1      ;GET TSSR
6000 057344 012702 000200          MOV      #SSR,R2          ;SET UP EXPECTED TSSR
6001 057350 010004          MOV      R0,R4          ;PACKET ADDRESS SET UP
6002 057352 005237 002212          INC      FATFLG          ;BUMP COUNT
6006 057356          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
          057356 104456          TRAP      C$ERHRD
          057360 001157          .WORD    623
          057362 073304          .WORD    T26RWN
          057364 012136          .WORD    PKTSSR
6007 057366          30$:   CKLOOP          ;LOOP IF SELECTED
          057366 104406          TRAP      C$CLP1
6008
6009          ;*****
6010          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
6011          ;
6012          ;*****
6013
6014
6015 057370 013701 071660          MOV      T26BFR+6,R1      ;PICK UP XST0
6016 057374 010102          MOV      R1,R2          ;SET UP EXPECTED
6017 057376 052702 000002          BIS      #BIT1,R2          ;SET BOT BIT IN EXPECTED
6018 057402 020102          CMP      R1,R2          ;DOES EXP = REC'D
6019 057404 001406          BEQ      40$              ;BR, IF EQUAL (OK)
6020 057406 005237 002212          INC      FATFLG          ;BUMP COUNT
6024 057412          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          057412 104456          TRAP      C$ERHRD
          057414 001160          .WORD    624
          057416 073015          .WORD    T26BOT

```


TEST 6: REREADS

```

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

```

M2

TEST 6: REREADS

```

6078 057610 012136 130$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
      057612 104406 TRAP C$CLP1
      057612 104406
6079
6080 ;*****
6081 ;
6082 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6083 ;
6084 ;*****
6085
6086 057614 013701 071660 MOV T26BFR+6,R1 ;PICK UP XSTO
6087 057620 010102 MOV R1,R2 ;SET UP EXPECTED
6088 057622 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6089 057626 020102 CMP R1,R2 ;DOES EXP = REC'D
6090 057630 001406 BEQ 140$ ;BR, IF EQUAL (OK)
6091 057632 005237 002212 INC FATFLG ;BUMP COUNT
6095 057636 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      057636 104456 TRAP C$ERHRD
      057640 001163 .WORD 627
      057642 073015 .WORD T26BOT
      057644 015564 .WORD EXPREC
6096 057646 140$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      057646 104406
6097
6098 ;*****
6099 ;
6100 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
6101 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
6102 ;
6103 ;*****
6104
6105 057650 012703 000001 MOV #1,R3 ;SPACE 1 RECORD FORWARD
6106 057654 004737 010556 JSR PC,SPACE ;SPACE CALL
6107 057660 012703 000400 MOV #256.,R3 ;RECORD SIZE
6108 057664 013737 003114 071752 150$: MOV FREE,T26RB ;STARTING READ BUFFER ADDRESS
6109
6110 ;*****
6111 ;
6112 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6113 ;
6114 ;*****
6115
6116 057672 012737 161001 071750 165$: MOV #161001,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6117 057700 012704 071750 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6118 057704 010337 071756 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6119 057710 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6120 057714 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
6121 057720 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6122 057724 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6123 057730 020102 CMP R1,R2 ;ARE THEY EQUAL
6124 057732 001406 BEQ 170$ ;BR, IF OK
6125 057734 005237 002212 INC FATFLG ;BUMP COUNT
6129 057740 ERRHRD ERRNO,T26RRG,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      057740 104456 TRAP C$ERHRD
      057742 001164 .WORD 628
      057744 072322 .WORD T26RRG
      057746 012136 .WORD PKTSSR

```

N2

TEST 6: REREADS

```

6130 057750      170$:  CKLOOP                       ;LOOP IF SELECTED
      057750      104406
6131 057752      005723
      057752      005723
6132 057754      062737 000001 071776          TST      (R3),
      057754      062737          ADD      #1,T26CNT          ;BUMP RECORD SIZE
      057754      062737          ;BUMP TAPE RECORD COUNTER
6133
6134
6135          ;*****
6136          ;READ DATA, CVC=1, ACK COMMAND
6137          ;
6138          ;*****
6139
6140 057762      012737 140001 071750          MOV      #140001,T26PK3      ;READ DATA, CVC=1, ACK COMMAND
6141 057770      010337 071756          MOV      R3,T26SZ          ;SET SIZE INTO PACKET
6142 057774      010465 000000          MOV      R4,TSDB(R5)       ;ISSUE READ DATA COMMAND
6143 060000      004737 016340          JSR      PC,WAITF          ;WAIT FOR SSR
6144 060004      016501 000002          MOV      TSSR(R5),R1       ;PICK UP THE TSSR
6145 060010      012702 000200          MOV      #SSR,R2          ;SET UP EXPECTED
6146 060014      020102          CMP      R1,R2             ;IS THE TSSR OK
6147 060016      001406          BEQ      195$              ;BR, IF TSSR OK (GOOD)
6148 060020      005237 002212          INC      FATFLG            ;BUMP COUNT
6152 060024      060024      104456          ERRHRD  ERRNO,RDERR,FKTSSR ;READ DATA COMMAND FAILED
      060024      001165
      060030      005204
      060032      012136
      060034      104406
6153 060034      195$:  CKLOOP                       ;LOOP IF SELECTED
      060034      104406
6154 060036      017701 123052          MOV      @FREE,R1          ;FIRST WORD FROM READ BUFFER
6155 060042      013702 071776          MOV      T26CNT,R2        ;SET UP EXPECTED
6156 060046      020102          CMP      R1,R2             ;IS TAPE POSITION CORRECT
6157 060050      001406          BEQ      197$              ;KEEP GOING POSITION OK
6158 060052      005237 002212          INC      FATFLG            ;BUMP COUNT
6162 060056      060056      104456          ERRHRD  ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
      060056      001166
      060060      072006
      060062      015564
      060064      015564
      060066      104406
6163 060066      197$:  CKLOOP                       ;LOOP IF SELECTED
      060066      104406
6164 060070      022703 000412          CMP      #266.,R3         ;AT MAX SIZE YET
6165 060074      001401          BEQ      200$              ;BR, IF AT END OF THE SUBTEST
6166 060076      000672          BR       150$              ;KEEP GOING MORE RECORDS
6167 060100
6168 060100      200$:  ENDSUB                         ;>>>>>>>>> END SUBTEST >>>>>>>>>
      060100      104403
      060100      023727 002212 000017          CMP      FATFLG,#15.       ;IS ERROR COUNT AT 25
6169 060102      023727          BLO      999$              ;BR, IF LESS THAN 25
6170 060110      103402          JSR      PC,CKDROP         ;TRY TO DROP THE UNIT
6171 060112      004737 017272
6172 060116      999$:

```


TEST 6: REREADS

```

6227 060222 104406 26$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
060222
6228 ;*****
6229 ;
6230 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6231 ;
6232 ;*****
6233
6234
6235 060224 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6236 060230 103413 BCS 50$ ;BR, IF NO PROBLEM
6237 060232 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
6238 060236 012702 000200 MOV @SSR,R2 ;SET UP EXPECTED TSSR
6239 060242 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
6240 060244 005237 002212 INC FATFLG ;BUMP COUNT
6244 060250 ERRHRD ERRNO,T26RWN,PKTSSW ;REWIND NOT ACCEPTED
060250 104456 TRAP C$FRHRD
060252 001171 .WORD 633
060254 073304 .WORD T26RWN
060256 012136 .WORD PKTSSR
6245 060260 104406 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
060260
6246 ;*****
6247 ;
6248 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6249 ;
6250 ;*****
6251
6252
6253 060262 013701 071660 MOV T26BFR+6,R1 ;PICK UP XSTO
6254 060266 010102 MOV R1,R2 ;SET UP EXPECTED
6255 060270 052702 000002 BIS @BIT1,R2 ;SET BOT BIT IN EXPECTED
6256 060274 020102 CMP R1,R2 ;DOES EXP = REC'D
6257 060276 001406 BEQ 40$ ;BR, IF EQUAL (OK)
6258 060300 005237 002212 INC FATFLG ;BUMP COUNT
6262 060304 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
060304 104456 TRAP C$ERHRD
060306 001172 .WORD 634
060310 073015 .WORD T26BOT
060312 015564 .WORD EXPREC
6263 060314 104406 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
060314
6264 060316 012703 000400 MOV @256.,R3 ;RECORD SIZE
6265 060322 013737 003114 071752 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
6266 ;*****
6267 ;
6268 ;WRITE DATA,CVC=1,ACK COMMAND
6269 ;
6270 ;*****
6271
6272
6273 060330 012737 140005 071750 MOV @140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
6274 060336 012704 071750 MOV @T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6275 060342
6276 060342 010300 65$: MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
6277 060344 004737 017512 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
6278 060350 010337 071756 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET

```

TEST 6: REREADS

```

6279 060354 013777 071776 122532      MOV      T26CNT,0FREE      ;MOVE TAPE RECORD NUMBER TO BUFFER
6280 060362 062737 000001 071776      ADD      #1,T26CNT        ;NUMBER READY FOR NEXT RECORD
6281 060370 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
6282 060374 004737 016340      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
6283 060400 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
6284 060404 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
6285 060410 020102      CMP      R1,R2           ;ARE THEY EQUAL
6286 060412 001406      BEQ      75$            ;BR, IF OK
6287 060414 005237 002212      INC      FATFLG         ;BUMP COUNT
6291 060420      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
        060420 104456
        060422 001173          TRAP     C$ERHRD
        060424 005111          .WORD   635
        060426 012136          .WORD   WRTErr
6292 060430      75$:   CKLOOP          ;LOOP IF SELECTED          .WORD   PKTSSR
        060430 104406          TRAP     C$CLP1
6293 060432 005723      TST      (R3).          ;BUMP THE RECORD SIZE
6294 060434 022703 000412      CMP      #266.,R3      ;MAXIMUM SIZE YET
6295 060440 001401      BEQ      120$          ;BR, IF AT END OF WRITE SEQUENCE
6296 060442 000737      BR       65$           ;WRITE MORE RECORDS
6297 060444
6298 060444 005037 071776      120$:   CLR      T26CNT      ;SET RECORD COUNTER BACK TO ZERO
6299
6300      ;*****
6301      ;
6302      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6303      ;
6304      ;*****
6305
6306 060450 004737 011104      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
6307 060454 103413      BCS      130$          ;BR, IF NO PROBLEM
6308 060456 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR
6309 060462 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED TSSR
6310 060466 010004      MOV      R0,R4         ;PACKET ADDRESS SET UP
6311 060470 005237 002212      INC      FATFLG         ;BUMP COUNT
6315 060474      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
        060474 104456          TRAP     C$ERHRD
        060476 001174          .WORD   636
        060500 073304          .WORD   T26RWN
        060502 012136          .WORD   PKTSSR
6316 060504      130$:  CKLOOP          ;LOOP IF SELECTED          TRAP     C$CLP1
        060504 104406          TRAP     C$CLP1
6317
6318      ;*****
6319      ;
6320      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6321      ;
6322      ;*****
6323
6324 060506 013701 071660      MOV      T26BFR+6,R1   ;PICK UP XSTO
6325 060512 010102      MOV      R1,R2         ;SET UP EXPECTED
6326 060514 052702 000002      BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
6327 060520 020102      CMP      R1,R2         ;DOES EXP = REC'D
6328 060522 001406      BEQ      140$          ;BR, IF EQUAL (OK)
6329 060524 005237 002212      INC      FATFLG         ;BUMP COUNT
6333 060530      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
        060530 104456          TRAP     C$ERHRD
    
```

TEST 6: REREADS

```

060532 001175
060534 073015
060536 015564
6334 060540 104406 140$: CKLOOP ;LOOP IF SELECTED .WORD 637
060540 104406 TRAP C$CLP1 .WORD T26BOT
6335 ;***** .WORD EXPREC
6336 ;
6337 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
6338 ;BIT 15 SETS DIRECTION 0=FORWARD 1=REVERSE
6339 ;
6340 ;*****
6341 ;
6342 ;
6343 060542 012703 000001 MOV #1,R3 ;SET UP SPACE FORWARD 1
6344 060546 004737 010556 JSR PC,SPACE ;ISSUE SPACE COMMAND
6345 060552 012703 000400 MOV #256,R3 ;RECORD SIZE
6346 060556 013737 003114 071752 150$: MOV FREE,T26RB ;STARTING READ BUFFER ADDRESS
6347 ;
6348 ;*****
6349 ;
6350 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6351 ;
6352 ;*****
6353 ;
6354 060564 012737 171001 071750 MOV #171001,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6355 060572 012704 071750 165$: MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6356 060576 010337 071756 MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6357 060602 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6358 060606 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
6359 060612 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6360 060616 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6361 060622 020102 CMP R1,R2 ;ARE THEY EQUAL
6362 060624 001406 BEQ 170$ ;BR, IF OK
6363 060626 005237 002212 INC FATFLG ;BUMP COUNT
6367 060632 ERRHRD ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
060632 104456 TRAP C$ERHRD
060634 001176 .WORD 638
060636 072225 .WORD T26RRF
060640 012136 .WORD PKTSSR
6368 060642 104406 170$: CKLOOP ;LOOP IF SELECTED .WORD
060642 104406 TRAP C$CLP1
6369 060644 017701 122244 MOV #FREE,R1 ;FIRST WORD FROM READ BUFFER
6370 060650 013702 071776 MOV T26CNT,R2 ;SET UP EXPECTED
6371 060654 000302 SWAB R2 ;SWAP BYTES IN EXPECTED
6372 060656 020102 CMP R1,R2 ;IS TAPE POSITION CORRECT
6373 060660 001406 BEQ 190$ ;KEEP GOING POSITION OK
6374 060662 005237 002212 INC FATFLG ;BUMP COUNT
6378 060666 ERRHRD ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
060666 104456 TRAP C$ERHRD
060670 001177 .WORD 639
060672 072006 .WORD T26WNG
060674 015564 .WORD EXPREC
6379 060676 104406 190$: CKLOOP ;LOOP IF SELECTED .WORD
060676 104406 TRAP C$CLP1
6380 060700 005723 TST (R3), ;NEXT RECORD SIZE
6381 060702 062737 000001 071776 ADD #1,T26CNT ;BUMP TAPE RECORD COUNTER
6382

```


TEST 6: REREADS

```

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

```

T3

TEST 6: REREADS

```

6528 061304 012702 000200      MOV     #SSR,R2      ;SET UP EXPECTED
6529 061310 020102      CMP     R1,R2       ;ARE THEY EQUAL
6530 061312 001406      BEQ     75$         ;BR, IF OK
6531 061314 005237 002212      INC     FATFLG      ;BUMP COUNT
6535 061320      ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
        061320 104456      TRAP   C$ERHRD
        061322 001206      .WORD 646
        061324 005111      .WORD WRTERR
        061326 012136      .WORD PKTSSR
6536 061330      75$:  CKLOOP      ;LOOP IF SELECTED
        061330 104406      TRAP   C$CLP1
6537 061332 005303      DEC     R3          ;SET RECORD SIZE TO 511.
6538 061334 013737 003114 071752      MOV     FREE,T26RB ;STARTING READ BUFFER ADDRESS
6539
6540      ;*****
6541      ;
6542      ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
6543      ;
6544      ;*****
6545
6546 061342 012737 161001 071750      MOV     #161001,T26PK3 ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
6547 061350 012704 071750      165$: MOV     #T26PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
6548 061354 010337 071756      MOV     R3,T26SZ    ;SET UP RECORD SIZE IN PACKET
6549 061360 010465 000000      MOV     R4,TSDB(R5) ;ISSUE COMMAND
6550 061364 004737 016340      JSR     PC,WAITF    ;WAIT FOR SSR TO SET
6551 061370 016501 000002      MOV     TSSR(R5),R1 ;GET TSSR CONTENTS
6552 061374 012702 100204      MOV     #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6553 061400 020102      CMP     R1,R2       ;ARE THEY EQUAL
6554 061402 001406      BEQ     170$        ;BR, IF OK
6555 061404 005237 002212      INC     FATFLG      ;BUMP COUNT
6559 061410      ERRHRD  ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
        061410 104456      TRAP   C$ERHRD
        061412 001207      .WORD 647
        061414 074362      .WORD T26TRL
        061416 012136      .WORD PKTSSR
6560 061420      170$: CKLOOP      ;LOOP IF SELECTED
        061420 104406      TRAP   C$CLP1
6561
6562      ;*****
6563      ;
6564      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6565      ;
6566      ;*****
6567
6568 061422 013701 071660      MOV     T26BFR+6,R1 ;GET MESSAGE BUFFER
6569 061426 010102      MOV     R1,R2       ;SET UP EXPECTED
6570 061430 052702 010000      BIS     #BIT12,R2   ;SET THE RLL BIT IN EXPECTED
6571 061434 020102      CMP     R1,R2       ;ARE THEY EQUAL
6572 061436 001406      BEQ     180$        ;BR, IF EQUAL (ALL IS WELL)
6573 061440 005237 002212      INC     FATFLG      ;BUMP COUNT
6577 061444      ERRHRD  ERRNO,T26LON,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
        061444 104456      TRAP   C$ERHRD
        061446 001210      .WORD 648
        061450 074130      .WORD T26LON
        061452 015564      .WORD EXPREC
6578 061454      180$: CKLOOP      ;LOOP IF SELECTED
        061454 104406      TRAP   C$CLP1
    
```


TEST 6: REREADS

```

6681 061720 012124
061722 26$: CKLOOP ;LOOP IF SELECTED .WORD SFIMSG
061722 104406 TRAP C$CLP1
6682
6683 ;*****
6684 ;
6685 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6686 ;
6687 ;*****
6688
6689 061724 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6690 061730 103413 BCS 30$ ;BR, IF NO PROBLEM
6691 061732 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
6692 061736 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
6693 061742 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
6694 061744 005237 002212 INC FATFLG ;BUMP COUNT
6698 061750 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
061750 104456 TRAP C$ERHRD
061752 001215 .WORD 653
061754 073304 .WORD T26RWN
061756 012136 .WORD PKTSSR
6699 061760 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
061760 104406 TRAP C$CLP1
6700
6701 ;*****
6702 ;
6703 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
6704 ;
6705 ;*****
6706
6707 061762 013701 071660 MOV T26FR+6,R1 ;PICK UP XST0
6708 061766 010102 MOV R1,R2 ;SET UP EXPECTED
6709 061770 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
6710 061774 020102 CMP R1,R2 ;DOES EXP = REC'D
6711 061776 001406 BEQ 40$ ;BR, IF EQUAL (OK)
6712 062000 005237 002212 INC FATFLG ;BUMP COUNT
6716 062004 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
062004 104456 TRAP C$ERHRD
062006 001216 .WORD 654
062010 073015 .WORD T26BOT
062012 015564 .WORD EXPREC
6717 062014 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
062014 104406 TRAP C$CLP1
6718 062016 012703 000400 MOV #256.,R3 ;RECORD SIZE
6719 062022 013737 003114 071752 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
6720
6721 ;*****
6722 ;
6723 ;WRITE DATA,CVC=1,ACK COMMAND
6724 ;
6725 ;*****
6726
6727 062030 012737 140005 071750 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
6728 062036 012704 071750 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6729 062042
6730 062042 010337 071756 65$: MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6731 062046 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND

```

TEST 6: REREADS

```

6732 062052 004737 016340          JSR      PC,WAITF          ;WAIT FOR SSR TO SET
6733 062056 016501 000002          MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
6734 062062 012702 000200          MOV      #SSR,R2         ;SET UP EXPECTED
6735 062066 020102                   CMP      R1,R2           ;ARE THEY EQUAL
6736 062070 001406          BEQ      75$             ;BR, IF OK
6737 062072 005237 002212          INC      FATFLG          ;BUMP COUNT
6741 062076                   ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    655
                                .WORD    WRERR
                                .WORD    PKTSSR
                                TRAP      C$CLP1
6742 062106                   75$:   CKLOOP            ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104456
6743 062110 012703 001000          MOV      #512.,R3        ;RECORD SIZE
6744 062114 013737 003114 071752    MOV      FREE,T26RB      ;STARTING READ BUFFER ADDRESS
6745
6746          ;*****
6747          ;
6748          ;REREAD PREVIOUS,ACK,CVC=1,OPP=1
6749          ;
6750          ;*****
6751
6752 062122 012737 161001 071750    MOV      #161001,T26PK3  ;REREAD PREVIOUS,ACK,CVC=1,OPP=1
6753 062130 012704 071750    165$:   MOV      #T26PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
6754 062134 010337 071756    MOV      R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
6755 062140 010465 000000    MOV      R4,TSD8(R5)     ;ISSUE COMMAND
6756 062144 004737 016340          JSR      PC,WAITF          ;WAIT FOR SSR TO SET
6757 062150 016501 000002          MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
6758 062154 012702 100204          MOV      #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6759 062160 020102                   CMP      R1,R2           ;ARE THEY EQUAL
6760 062162 001406          BEQ      170$           ;BR, IF OK
6761 062164 005237 002212          INC      FATFLG          ;BUMP COUNT
6765 062170                   ERRHRD  ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    656
                                .WORD    T26TRL
                                .WORD    PKTSSR
6766 062200                   170$:  CKLOOP            ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406
6767
6768          ;*****
6769          ;
6770          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6771          ;
6772          ;*****
6773
6774 062202 013701 071660          MOV      T26BFR+6,R1     ;GET MESSAGE BUFFER
6775 062206 010102          MOV      R1,R2           ;SET UP EXPECTED
6776 062210 052702 040000          BIS      #BIT14,R2       ;SET THE RLS BIT IN EXPECTED
6777 062214 020102                   CMP      R1,R2           ;ARE THEY EQUAL
6778 062216 001406          BEQ      180$           ;BR, IF EQUAL (ALL IS WELL)
6779 062220 005237 002212          INC      FATFLG          ;BUMP COUNT
6783 062224                   ERRHRD  ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
                                TRAP      C$ERHRD
                                .WORD    657
                                .WORD    T26LOP
                                .WORD    EXPREC
6784 062224 104456
6785 062226 001221
6786 062230 074212
6787 062232 015564

```

TEST 6: REREADS

```

6784 062234      180$:  CKLOOP
      062234 104406
6785 062236 013701 071656      MOV    T26BFR+4,R1      ;PICK UP RESIDUAL BYTE COUNTER
6786 062242 012702 000400      MOV    #256.,R2        ;THIS SHOULD BE THE DIFFEREN
6787 062246 020102      CMP    R1,R2           ;IS THE DIFFERENCE CORRECT
6788 062250 001406      BEQ    190$            ;BR, IF CORRECT
6789 062252 005237 002212      INC    FATFLG          ;BUMP COUNT
6793 062256      ERRHRD ERRNO,T26PBP,EXPREC ;RBPCR NOT CORRECT
      062256 104456
      062260 001222      TRAP   C$ERHRD
      062262 074274      .WORD 658
      062264 015564      .WORD T26PBP
6794 062266      .WORD  EXPREC
      062266 104406      190$:  CKLOOP          ;LOOP IF SELECTED
6795 062270 012703 001000      MOV    #512.,R3        ;RECORD SIZE      TRAP   C$CLP1
6796 062274 013737 003114 071752      MOV    FREE,T26RB      ;STARTING READ BUFFER ADDRESS
6797
6798
6799
6800 ;*****
6801 ;REREAD PREVIOUS,ACK,CVC=1,C^P=0
6802 ;
6803 ;*****
6804 062302 012737 141001 071750      MOV    #141001,T26PK3  ;REREAD PREVIOUS,ACK,CVC=1,OPP=0
6805 062310 012704 071750      MOV    #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
6806 062314 010337 071756      MOV    R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
6807 062320 010465 000000      MOV    R4,T5DB(R5)     ;ISSUE COMMAND
6808 062324 004737 016340      JSR    PC,WAITF        ;WAIT FOR SSR TO SET
6809 062330 016501 000002      MOV    T5SR(R5),R1     ;GET T5SR CONTENTS
6810 062334 012702 100204      MOV    #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6811 062340 020102      CMP    R1,R2           ;ARE THEY EQUAL
6812 062342 001406      BEQ    270$            ;BR, IF OK
6813 062344 005237 002212      INC    FATFLG          ;BUMP COUNT
6817 062350      ERRHRD ERRNO,T26TRL,PKT5SR ;T5SR INCORRECT AFTER READ DATA
      062350 104456      TRAP   C$ERHRD
      062352 001223      .WORD 659
      062354 074362      .WORD T26TRL
      062356 012136      .WORD  PKT5SR
6818 062360      270$:  CKLOOP          ;LOOP IF SELECTED
      062360 104406      TRAP   C$CLP1
6819
6820 ;*****
6821 ;
6822 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6823 ;
6824 ;
6825 ;*****
6826 062362 013701 071660      MOV    T26BFR+6,R1     ;GET MESSAGE BUFFER
6827 062366 010102      MOV    R1,R2           ;SET UP EXPECTED
6828 062370 052702 040000      BIS    #8BIT14,R2      ;SET THE RLS BIT IN EXPECTED
6829 062374 020102      CMP    R1,R2           ;ARE THEY EQUAL
6830 062376 001406      BEQ    280$            ;BR, IF EQUAL (ALL IS WELL)
6831 062400 005237 002212      INC    FATFLG          ;BUMP COUNT
6835 062404      ERRHRD ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
      062404 104456      TRAP   C$ERHRD
      062406 001224      .WORD 660
      062410 074212      .WORD T26LOP

```


TEST 6: REREADS

6836	062412	015564								
	062414		280:	CKLOOP				.WORD	EXPREC	
	062414	104406						TRAP	C%CP1	
6837	062416	013701	071656	MOV	T26BFR,4,R1					
6838	062422	012702	000400	MOV	#256.,R2					
6839	062426	020102		CMP	R1,R2					
6840	062430	001405		BEQ	290:					
6844	062434			ERRHRD	ERRNO,T26PBP,EXPREC					
	062434	104456								
	062436	001224						TRAP	C\$ERHPO	
	062440	074274						.WORD	650	
	062442	015564						.WORD	T26PBP	
6845	062444		290:	CKLOOP				.WORD	EXPREC	
	062444	104406								
6846	062446			ENDSUB						
	062446									
	062446	104403								
6847	062450	023727	002212	000017	CMP	FATFLG,#15.				
6848	062456	103402			BLO	999:				
6849	062460	004737	017272		JSR	PC,CKDROP				
6850	062464		999:							

;PICK UP RESIDUAL BYTE COUNTER
 ;THIS SHOULD BE THE DIFFERENCE
 ;IS THE DIFFERENCE CORRECT
 ;BR, IF CORRECT
 ;RBPOR NOT CORRECT

;LOOP IF SELECTED

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

TEST 6: REREADS

```

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

```

TRAP C\$ERHRD
.WORD 662
.WORD WRTMSG
.WORD SFIMSG

TRAP C\$CLP1

TRAP C\$ERHRD
.WORD 663
.WORD T26RWN
.WORD PKTSSR

TRAP C\$CLP1

TRAP C\$ERHRD
.WORD 664
.WORD T26BOT
.WORD EXPREC

TRAP C\$CLP1

TEST 6: REREADS

```

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

```

```

65$:
MOV      R3,R0           ;SET PATTERN IN CORRECT REGISTER
JSR      PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
MOV      R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
MOV      R4,TSD8(R5)     ;ISSUE COMMAND
JSR      PC,WAITF        ;WAIT FOR SSR TO SET
MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
MOV      @SSR,R2         ;SET UP EXPECTED
CMP      R1,R2           ;ARE THEY EQUAL
BEQ      75$             ;BR, IF OK
INC      FATFLG          ;BUMP COUNT
ERRHRD   ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP   C$ERHRD
                                .WORD  665
                                .WORD  WRERR
                                .WORD  PKTSSR
75$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP   C$CLP1
TST      (R3)+           ;BUMP RECORD SIZE
CMP      @268.,R3        ;END OF RECORD YET
BNE      65$             ;BR, IF MORE RECORDS TO WRITE
80$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP   C$CLP1
120$:
;*****
;
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
JSR      PC,REWIND       ;CALL TAPE REWIND COMMAND
BCS     130$            ;BR, IF NO PROBLEM
MOV      TSSR(R5),R1     ;GET TSSR
MOV      @SSR,R2         ;SET UP EXPECTED TSSR
MOV      R0,R4           ;PACKET ADDRESS SET UP
INC      FATFLG          ;BUMP COUNT
ERRHRD   ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP   C$ERHRD
                                .WORD  666
                                .WORD  T26RWN
                                .WORD  PKTSSR
130$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP   C$CLP1
;*****
;
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;
;*****
MOV      T268FR+6,R1     ;PICK UP XSTO
MOV      R1,R2           ;SET UP EXPECTED
BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
CMP      R1,R2           ;DOES EXP = REC'D
BEQ      140$           ;BR, IF EQUAL (OK)
INC      FATFLG          ;BUMP COUNT

```

TEST 6: REREADS

```

7012 063052          ERRHRD  ERRNO,T26BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      063052      104456          TRAP          C$ERHRD
      063054      001233          .WORD        667
      063056      073015          .WORD        T26BOT
      063060      015564          .WORD        EXPREC
7013 063062          140$:  CKLOOP                      ;LOOP IF SELECTED
      063062      104406          TRAP          C$CLP1
7014 063064      012737  000400  072002      MOV      #256.,T26RSZ      ;STORE START RECORD SIZE
7015 063072      000420          BR        150$           ;SKIP THE SPACE THIS TIME
7016
7017
7018
7019
7020
7021
7022
7023
      ;*****
      ;
      ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
      ;BIT 15 SETS DIRECTION - 0=FORWARD  1=REVERSE
      ;
      ;*****
7024 063074      012703  000001  145$:  MOV      #1,R3          ;SPACE ONE RECORD PARAMETER
7025 063100      004737  010556          JSR      PC,SPACE        ;CALL SPACE ROUTINE
7026 063104      103413          BCS     150$           ;BR, IF NO PROBLEM WITH SPACE COMMAND
7027 063106      016501  000002          MOV      TSSR(R5),R1     ;GET TSSR
7028 063112      012702  000200          MOV      #SSR,R2        ;SET UP EXPECTED TSSR
7029 063116      010004          MOV      R0,R4          ;PACKET ADDRESS SET UP
7030 063120      005237  002212          INC     FATFLG          ;BUMP COUNT
7034 063124          ERRHRD  ERRNO,T26SC,EXPREC      ;POSITION (SPACE RECORDS) FAILED
      063124      104456          TRAP          C$ERHRD
      063126      001234          .WORD        668
      063130      072417          .WORD        T26SC
      063132      015564          .WORD        EXPREC
7035 063134          150$:  CKLOOP                      ;LOOP IF SELECTED
      063134      104406          TRAP          C$CLP1
7036 063136      013703  072002      MOV      T26RSZ,R3      ;RECORD SIZE
7037 063142      013737  003114  071752      MOV      FREE,T26RB     ;STARTING READ BUFFER ADDRESS
7038
7039
7040
7041
7042
7043
7044
      ;*****
      ;
      ;REREREAD DATA,CVC=1,ACK COMMAND
      ;
      ;*****
7045 063150      012737  141401  071750      MOV      #141401,T26PK3 ;REREREAD DATA,CVC=1,ACK COMMAND
7046 063156      012704  071750  165$:  MOV      #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
7047 063162      010337  071756          MOV      R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
7048 063166      010465  000000          MOV      R4,TSD8(R5)    ;ISSUE COMMAND
7049 063172      004737  016340          JSR      PC,WAITF       ;WAIT FOR SSR TO SET
7050 063176      016501  000002          MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
7051 063202      012702  000200          MOV      #SSR,R2        ;SET UP EXPECTED
7052 063206      020102          CMP     R1,R2          ;ARE THEY EQUAL
7053 063210      001406          BEQ    170$           ;BR, IF OK
7054 063212      005237  002212          INC     FATFLG          ;BUMP COUNT
7058 063216          ERRHRD  ERRNO,T26WDC,PKTSSR      ;TSSR INCORRECT AFTER REREAD DATA
      063216      104456          TRAP          C$ERHRD
      063220      001235          .WORD        669
      063222      073640          .WORD        T26WDC
      063224      012136          .WORD        PKTSSR
7059 063226          170$:  CKLOOP                      ;LOOP IF SELECTED
      063226      104406          TRAP          C$CLP1

```

TEST 6: REREADS

```

7060 063230 013702 003114      MOV     FREE,R2          ;CURRENT BUFFER ADDRESS TO R2
7061 063234 010304              MOV     R3,R4          ;CURRENT RECORD SIZE
7062 063236 162704 000400      SUB     #256.,R4       ;FIRST LOCATION IN BUFFER
7063 063242 060204              173$: ADD     R2,R4         ;SET UP POINTER
7064 063244 021403              CMP     (R4),R3        ;CHECK DATA READ (R3=DATA ALSO)
7065 063246 001410              BEQ     180$          ;BR, IF ALL IS WELL
7066 063250 011401              MOV     (R4),R1       ;RECD DATA
7067 063252 010302              MOV     R3,R2         ;EXPECTED DATA
7068 063254 005237 002212      INC     FATFLG         ;BUMP COUNT
7072 063260              ERRHRD ERRNO,T26DTA,EXPREC ;DATA READ NOT - WRITTEN
                                TRAP     C$ERRHRD
                                .WORD    670
                                .WORD    T26DTA
                                .WORD    EXPREC
7073 063270              180$: CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP1
7074 063272 104406              TST     (R4)+         ;BUMP TO NEXT LOCATION
7075 063274 160204              SUB     R2,R4         ;CORRECT RECORDS SIZE VALUE
7076 063276 020403              CMP     R4,R3         ;END OF RECORD YET
7077 063300 001360              BNE     173$          ;BR, IF NOT AT END OF RECORD
7078 063302 005723              TST     (R3)+         ;BUMP RECORD SIZE
7079 063304 010337 072002      MOV     R3,T26RSZ     ;STORE PRESENT RECORD SIZE
7080 063310 022703 000410      CMP     #264.,R3     ;END OF RECORD YET
7081 063314 001267              BNE     145$          ;BR, IF MORE RECORDS TO READ
7082 063316              190$: CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP1
7083 063320              ENDSUB               ;>>>>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>>>
                                L10111:
                                TRAP     C$ESUB
7084 063322 104403              000017      CMP     FATFLG,#15.   ;IS ERROR COUNT AT 25
7085 063330 023727 002212 000017      BLO     999$         ;BR, IF LESS THAN 25
7086 063332 103402              017272      JSR     PC,CKDROP    ;TRY TO DROP THE UNIT
7087 063336              999$:

```


TEST 6: REREADS

```

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

```


TEST 6: REREADS

```

7194 063604 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
7195 063610 020102      CMP      R1,R2      ;ARE THEY EQUAL
7196 063612 001406      BEQ      75$      ;BR, IF OK
7197 063614 005237 002212      INC      FATFLG      ;BUMP COUNT
7201 063620      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      063620 104456
      063622 001243      TRAP    C$ERHRD
      063624 005111      .WORD  675
      063626 012136      .WORD  WRTErr
7202 063630      75$:   CKLOOP      ;LOOP IF SELECTED      .WORD  PKTSSR
      063630 104406
7203 063632 005723      TST      (R3)+      ;BUMP RECORD SIZE      TRAP    C$CLP1
7204 063634 022703 000414      CMP      #268.,R3   ;END OF RECORD YET
7205 063640 001346      BNE      65$      ;BR, IF MORE RECORDS TO WRITE
7206 063642      80$:   CKLOOP      ;LOOP IF SELECTED
      063642 104406
7207 063644      120$:
7208
7209
7210
7211
7212
7213
7214
7215 063644 004737 011104      JSR      PC,REWIND   ;CALL TAPE REWIND COMMAND
7216 063650 103413      BCS      130$      ;BR, IF NO PROBLEM
7217 063652 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR
7218 063656 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED TSSR
7219 063662 010004      MOV      R0,R4      ;PACKET ADDRESS SET UP
7220 063664 005237 002212      INC      FATFLG      ;BUMP COUNT
7224 063670      FRRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      063670 104456      TRAP    C$ERHRD
      063672 001244      .WORD  676
      063674 073304      .WORD  T26RWN
7225 063700      130$:  CKLOOP      ;LOOP IF SELECTED      .WORD  PKTSSR
      063700 104406      TRAP    C$CLP1
7226
7227
7228
7229
7230
7231
7232
7233 063702 013701 071660      MOV      T26BFR+6,R1 ;PICK UP XSTO
7234 063706 010102      MOV      R1,R2      ;SET UP EXPECTED
7235 063710 052702 000002      BIS      #BIT1,R2   ;SET BOT BIT IN EXPECTED
7236 063714 020102      CMP      R1,R2      ;DOES EXP = REC'D
7237 063716 001406      BEQ      140$      ;BR, IF EQUAL (OK)
7238 063720 005237 002212      INC      FATFLG      ;BUMP COUNT
7242 063724      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      063724 104456      TRAP    C$ERHRD
      063726 001245      .WORD  677
      063730 073015      .WORD  T26BOT
      063732 015564      .WORD  EXPREC
7243 063734      140$:  CKLOOP      ;LOOP IF SELECTED      TRAP    C$CLP1
      063734 104406

```

164

TEST 6: REREADS

```

7244 063736 012737 000400 072002      MOV    #256.,T26RSZ      ;START RECORD SIZE
7245 063744 000420                    BR     150$             ;SKIP SPACE THIS TIME
7246
7247 ;*****
7248 ;
7249 ;ISSUE SPACE RECORDS COMMAND  VALUE IN R3 SETS NUMBER OF RECORDS
7250 ;BIT 15 SETS DIRECTION - 0=FORWARD  1=REVERSE
7251 ;
7252 ;*****
7253
7254 063746 012703 000001 145$:  MOV    #1,R3          ;SPACE ONE RECORD PARAMETER
7255 063752 004737 010556      JSR    PC,SPACE        ;CALL SPACE ROUTINE
7256 063756 103413                    BCS    150$           ;BR, IF NO PROBLEM WITH SPACE COMMAND
7257 063760 016501 000002      MOV    TSSR(R5),R1     ;GET TSSR
7258 063764 012702 000200      MOV    #SSR,R2        ;SET UP EXPECTED TSSR
7259 063770 010004                    MOV    R0,R4          ;PACKET ADDRESS SET UP
7260 063772 005237 002212      INC    FATFLG         ;BUMP COUNT
7264 063776                    ERRHRD  ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
                                TRAP    C$ERHRD
                                .WORD   678
                                .WORD   T26SC
                                .WORD   EXPREC
7265 064006 104456 064000 150$:  CKLOOP                                TRAP    C$CLP1
                                .WORD   678
                                .WORD   T26SC
                                .WORD   EXPREC
                                TRAP    C$CLP1
7266 064010 013703 072002      MOV    T26RSZ,R3      ;RECORD SIZE
7267 064014 013737 003114 071752  MOV    FREE,T26RB     ;STARTING READ BUFFER ADDRESS
7268
7269 ;*****
7270 ;
7271 ;REREAD DATA,ACK,CVC=1,SWB COMMAND
7272 ;
7273 ;*****
7274
7275 064022 012737 151401 071750 165$:  MOV    #151401,T26PK3  ;REREAD DATA,ACK,CVC=1,SWB COMMAND
7276 064030 012704 071750      MOV    #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
7277 064034 010337 071756      MOV    R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
7278 064040 010465 000000      MOV    R4,TSDB(R5)    ;ISSUE COMMAND
7279 064044 004737 016340      JSR    PC,WAITF       ;WAIT FOR SSR TO SET
7280 064050 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
7281 064054 012702 000200      MOV    #SSR,R2       ;SET UP EXPECTED
7282 064060 020102                    CMP    R1,R2          ;ARE THEY EQUAL
7283 064062 001406                    BEQ    170$           ;BR, IF OK
7284 064064 005237 002212      INC    FATFLG         ;BUMP COUNT
7288 064070                    ERRHRD  ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                                TRAP    C$ERHRD
                                .WORD   679
                                .WORD   T26WDC
                                .WORD   PKTSSR
7289 064100 104456 064100 170$:  CKLOOP                                ;LOOP IF SELECTED
                                TRAP    C$CLP1
7290 064102 013702 003114      MOV    FREE,R2        ;CURRENT BUFFER ADDRESS TO R2
7291 064106 010304                    MOV    R3,R4          ;CURRENT RECORD SIZE
7292 064110 162704 000400      SUB    #256.,R4       ;FIRST LOCATION IN BUFFER
7293 064114 060204 173$:  ADD    R2,R4          ;SET UP POINTER
7294 064116 021403                    CMP    (R4),R3        ;CHECK DATA READ (R3=DATA ALSO)
7295 064120 001410                    BEQ    180$           ;BR, IF ALL IS WELL
7296 064122 011401                    MOV    (R4),R1        ;RECD DATA

```


TEST 6: REREADS

```

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

```

TEST 6: REREADS

```

064404 015564
7426 064406          40$:  CKLOOP                ;LOOP IF SELECTED                .WORD  E$PRFC
      064406 104406
7427 064410 012703 000400          MOV    #256.,R3                ;RECORD SIZE
7428 064414 013737 003114 071752    MOV    FREE,T26RB             ;STARTING WRITE BUFFER ADDRESS
7429
7430          ;*****
7431          ;
7432          ;WRITE DATA,CVC-1,ACK COMMAND
7433          ;
7434          ;*****
7435
7436 064422 012737 140005 071750    MOV    #140005,T26PK3        ;WRITE DATA,CVC-1,ACK COMMAND
7437 064430 012704 071750          MOV    #T26PK3,R4           ;SET UP R4 WITH PACKET ADDRESS
7438 064434
7439 064434 010337 071756          65$:  MOV    R3,T26SZ            ;SET UP RECORD SIZE IN PACKET
7440 064440 013777 071776 116446    MOV    T26CNT,#FREE         ;MOVE TAPE RECORD NUMBER TO BUFFER
7441 064446 062737 000001 071776    ADD    #1,T26CNT           ;NUMBER READY FOR NEXT RECORD
7442 064454 010465 000000          MOV    R4,TSD8(R5)         ;ISSUE COMMAND
7443 064460 004737 016340          JSR    PC,WAITF            ;WAIT FOR SSR TO SET
7444 064464 016501 000002          MOV    TSSR(R5),R1         ;GET TSSR CONTENTS
7445 064470 012702 000200          MOV    #SSR,R2            ;SET UP EXPECTED
7446 064474 020102          CMP    R1,R2              ;ARE THEY EQUAL
7447 064476 001406          BEQ    75$                ;BR, IF OK
7448 064500 005237 002212          INC    FATFLG             ;BUMP COUNT
7452 064504          ERRHRD  ERRNO,WRERR,PKTSSR  ;TSSR INCORRECT AFTER REREAD DATA
      064504 104456          TRAP   C$ERRRD
      064506 001255          .WORD  685
      064510 005111          .WORD  WRERR
      064512 012136          .WORD  PKTSSR
7453 064514          75$:  CKLOOP                ;LOOP IF SELECTED                .WORD  E$PRFC
      064514 104406          TRAP   C$CLP1
7454 064516 005723          TST    (R3).              ;BUMP THE RECORD SIZE
7455 064520 022703 000414          CMP    #268.,R3           ;MAXIMUM SIZE YET
7456 064524 001401          BEQ    120$               ;BR, IF AT END OF WRITE SEQUENCE
7457 064526 000742          BR     65$                ;WRITE MORE RECORDS
7458 064530
7459 064530 005037 071776          120$: CLR    T26CNT           ;SET RECORD COUNTER BACK TO ZERO
7460
7461          ;*****
7462          ;
7463          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7464          ;
7465          ;*****
7466
7467 064534 004737 011104          JSR    PC,REWIND          ;CALL TAPE REWIND COMMAND
7468 064540 103411          BCS    130$              ;BR, IF NO PROBLEM
7469 064542 016501 000002          MOV    TSSR(R5),R1         ;GET TSSR
7470 064546 010004          MOV    R0,R4              ;PACKET ADDRESS SET UP
7471 064550 005237 002212          INC    FATFLG             ;BUMP COUNT
7475 064554          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      064554 104456          TRAP   C$ERRRD
      064556 001256          .WORD  686
      064560 073304          .WORD  T26RWN
      064562 012136          .WORD  PKTSSR
7476 064564          130$: CKLOOP                ;LOOP IF SELECTED                .WORD  E$PRFC
      064564 104406          TRAP   C$CLP1

```

TEST 6: REREADS

```

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

```

```

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

```


TEST 6: REREADS

```

7617 065170          26$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      065170 104406
7618
7619 ;*****
7620 ;
7621 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7622 ;
7623 ;*****
7624
7625 065172 004737 011104      JSR    PC,REWIND          ;CALL TAPE REWIND COMMAND
7626 065176 016501 000002      MOV    TSSR(R5),R1      ;GET TSSR
7627 065202 012702 000200      MOV    #SSR,R2         ;SET UP EXPECTED TSSR
7628 065206 103407          BCS    30$             ;BR, IF NO PROBLEM
7629 065210 010004          MOV    R0,R4           ;PACKET ADDRESS SET UP
7630 065212 005237 002212      INC    FATFLG          ;BUMP COUNT
7634 065216          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      065216 104456          TRAP  C$ERHRD
      065220 001265          .WORD 693
      065222 073304          .WORD T26RWN
      065224 012136          .WORD PKTSSR
7635 065226          30$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      065226 104406
7636
7637 ;*****
7638 ;
7639 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7640 ;
7641 ;*****
7642
7643 065230 013701 071660      MOV    T26BFR+6,R1     ;PICK UP XSTO
7644 065234 010102          MOV    R1,R2           ;SET UP EXPECTED
7645 065236 052702 000002      BIS    #BIT1,R2        ;SET BOT BIT IN EXPECTED
7646 065242 020102          CMP    R1,R2           ;DOES EXP = REC'D
7647 065244 001406          BEQ    40$             ;BR, IF EQUAL (OK)
7648 065246 005237 002212      INC    FATFLG          ;BUMP COUNT
7652 065252          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      065252 104456          TRAP  C$ERHRD
      065254 001266          .WORD 694
      065256 073015          .WORD T26BOT
      065260 015564          .WORD EXPREC
7653 065262          40$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      065262 104406
7654 065264 012703 000400      MOV    #256.,R3        ;RECORD SIZE
7655 065270 013737 003114 071752  MOV    FREE,T26RB      ;STARTING WRITE BUFFER ADDRESS
7656
7657 ;*****
7658 ;
7659 ;WRITE DATA,CVC=1,ACK COMMAND
7660 ;
7661 ;*****
7662
7663 065276 012737 140005 071750      MOV    #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
7664 065304 012704 071750      MOV    #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
7665 065310
7666 065310 010337 071756          65$:  MOV    R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
7667 065314 013777 071776 115572  MOV    T26CNT,#FREE    ;MOVE TAPE RECORD NUMBER TO BUFFER
7668 065322 062737 000001 071776  ADD    #1,T26CNT        ;NUMBER READY FOR NEXT RECORD
    
```

TEST 6: REREADS

```

7669 065330 010465 000000      MOV      R4,TSDB(R5)          ;ISSUE COMMAND
7670 065334 004737 016340      JSR      PC,WAITF           ;WAIT FOR SSR TO SET
7671 065340 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
7672 065344 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
7673 065350 020102              CMP      R1,R2             ;ARE THEY EQUAL
7674 065352 001406              BFQ      75$               ;BR, IF OK
7675 065354 005237 002212      INC      FATFLG            ;BUMP COUNT
7679 065360              ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    695
                                .WORD    WRERR
                                .WORD    PKTSSR
                                TRAP      C$CLP1
                                .WORD    696
                                .WORD    T26RWN
                                .WORD    PKTSSR
7680 065370              75$:   CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    696
                                .WORD    T26RWN
                                .WORD    PKTSSR
7681 065372 005723              TST      (R3)+             ;BUMP THE RECORD SIZE
7682 065374 022703 000414      CMP      #268.,R3         ;MAXIMUM SIZE YET
7683 065400 001401              BEQ      120$              ;BR, IF AT END OF WRITE SEQUENCE
7684 065402 000742              BR       65$               ;WRITE MORE RECORDS
7685 065404              120$:  CLR      T26CNT         ;SET RECORD COUNTER BACK TO ZERO
7686 065404 005037 071776
7687
7688 ;*****
7689 ;
7690 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7691 ;
7692 ;*****
7693
7694 065410 004737 011104      JSR      PC,REWIND         ;CALL TAPE REWIND COMMAND
7695 065414 103411              BCS      130$              ;BR, IF NO PROBLEM
7696 065416 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR
7697 065422 010004              MOV      R0,R4            ;PACKET ADDRESS SET UP
7698 065424 005237 002212      INC      FATFLG            ;BUMP COUNT
7702 065430              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    696
                                .WORD    T26RWN
                                .WORD    PKTSSR
7703 065440              130$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    696
                                .WORD    T26RWN
                                .WORD    PKTSSR
7704 065440 104406
7705 ;*****
7706 ;
7707 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7708 ;
7709 ;*****
7710
7711 065442 013701 071660      MOV      T26BFR+6,R1       ;PICK UP XSTO
7712 065446 010102              MOV      R1,R2            ;SET UP EXPECTED
7713 065450 052702 000002      BIS      #BIT1,R2         ;SET BOT BIT IN EXPECTED
7714 065454 020102              CMP      R1,R2            ;DOES EXP = REC'D
7715 065456 001406              BEQ      135$              ;BR, IF EQUAL (OK)
7716 065460 005237 002212      INC      FATFLG            ;BUMP COUNT
7720 065464              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    697
                                .WORD    T26BOT
                                .WORD    EXPREC
065464 104456
065466 001271
065470 073015
065472 015564

```

TEST 6: REREADS

```

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


TEST 6: REREADS

```

066040 104406                                TRAP    C$CLP1
7844
7845      ;*****
7846      ;
7847      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7848      ;
7849      ;*****
7850
7851 066042 004737 011104                    JSR     PC,REWIND          ;CALL TAPE REWIND COMMAND
7852 066046 016501 000002                    MOV     TSSR(R5),R1      ;GET TSSR
7853 066052 012702 000200                    MOV     #SSR,R2         ;SET UP EXPECTED TSSR
7854 066056 103407                                BCS     30$             ;BR, IF NO PROBLEM
7855 066060 010004                                MOV     R0,R4           ;PACKET ADDRESS SET UP
7856 066062 005237 002212                    INC     FATFLG          ;BUMP COUNT
7860 066066                                ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD   703
                                .WORD   T26RWN
                                .WORD   PKTSSR
066066 104456
066070 001277
066072 073304
066074 012136
7861 066076                                30$:   CKLOOP           ;LOOP IF SELECTED
066076 104406                                TRAP    C$CLP1
7862
7863      ;*****
7864      ;
7865      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7866      ;
7867      ;*****
7868
7869 066100 013701 071660                    MOV     T26BFR+6,R1     ;PICK UP XSTO
7870 066104 010102                                MOV     R1,R2           ;SET UP EXPECTED
7871 066106 052702 000002                    BIS     #BIT1,R2        ;SET BOT BIT IN EXPECTED
7872 066112 020102                                CMP     R1,R2           ;DOES EXP = REC'D
7873 066114 001406                                BEQ     40$             ;BR, IF EQUAL (OK)
7874 066116 005237 002212                    INC     FATFLG          ;BUMP COUNT
7878 066122                                ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD   704
                                .WORD   T26BOT
                                .WORD   EXPREC
066122 104456
066124 001300
066126 073015
066130 015564
7879 066132                                40$:   CKLOOP           ;LOOP IF SELECTED
066132 104406                                TRAP    C$CLP1
7880 066134 012703 001000                    MOV     #512.,R3        ;RECORD SIZE
7881 066140 013737 003114 071752            MOV     FREE,T26RB      ;STARTING WRITE BUFFER ADDRESS
7882
7883      ;*****
7884      ;
7885      ;WRITE DATA,CVC=1,ACK COMMAND
7886      ;
7887      ;*****
7888
7889 066146 012737 140005 071750            MOV     #140005,T26PK3  ;WRITE DATA,CVC=1,ACK COMMAND
7890 066154 012704 071750                    MOV     #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
7891 066160                                65$:   MOV     R3,T26SZ   ;SET UP RECORD SIZE IN PACKET
7892 066160 010337 071756                    MOV     R4,TSDB(R5)    ;ISSUE COMMAND
7893 066164 010465 000000                    JSR     PC,WAITF        ;WAIT FOR SSR TO SET
7894 066170 004737 016340                    MOV     TSSR(R5),R1    ;GET TSSR CONTENTS
7895 066174 016501 000002

```

TEST 6: REREADS

```

7896 066200 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
7897 066204 020102      CMP      R1,R2      ;ARE THEY EQUAL
7898 066206 001406      BEQ      75$        ;BR, IF OK
7899 066210 005237 002212      INC      FATFLG     ;BUMP COUNT
7903 066214      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      066214 104456      TRAP      C$ERHRD
      066216 001301      .WORD    705
      066220 005111      .WORD    WRTErr
      066222 012136      .WORD    PKTSSR
7904 066224      75$:   CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
      066224 104406
7905
7906      ;*****
7907      ;
7908      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7909      ;
7910      ;*****
7911
7912 066226 004737 011104      JSR      PC,REWIND  ;CALL TAPE REWIND COMMAND
7913 066232 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR
7914 066236 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED TSSR
7915 066242 103407      BCS     130$        ;BR, IF NO PROBLEM
7916 066244 010004      MOV      R0,R4      ;PACKET ADDRESS SET UP
7917 066246 005237 002212      INC      FATFLG     ;BUMP COUNT
7921 066252      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      066252 104456      TRAP      C$ERHRD
      066254 001302      .WORD    706
      066256 073304      .WORD    T26RWN
      066260 012136      .WORD    PKTSSR
7922 066262      130$:  CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
      066262 104406
7923
7924      ;*****
7925      ;
7926      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7927      ;
7928      ;*****
7929
7930 066264 013701 071660      MOV      T26BFR+6,R1 ;PICK UP XSTO
7931 066270 010102      MOV      R1,R2      ;SET UP EXPECTED
7932 066272 052702 000002      BIS     #BIT1,R2     ;SET BOT BIT IN EXPECTED
7933 066276 020102      CMP      R1,R2      ;DOES EXP = REC'D
7934 066300 001406      BEQ     140$        ;BR, IF EQUAL (OK)
7935 066302 005237 002212      INC      FATFLG     ;BUMP COUNT
7939 066306      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      066306 104456      TRAP      C$ERHRD
      066310 001303      .WORD    707
      066312 073015      .WORD    T26BOT
      066314 015564      .WORD    EXPREC
7940 066316      140$:  CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
      066316 104406
7941 066320 005303      DEC      R3          ;SET RECORD SIZE TO 511.
7942 066322 013737 003114 071752      MOV      FREE,T26RB ;STARTING READ BUFFER ADDRESS
7943
7944      ;*****
7945      ;
7946      ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND

```


TEST 6: REREADS

```

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

```


TEST 6: REREADS

```

8031 ;*
8032 |
8033 ;TEST 6, SUBTEST 12
8034 |
8035 ;VERIFIES THAT A REREAD NEXT COMMAND READING A
8036 ;RECORD SHORTER THAN THE SPECIFIED BYTE COUNT CAUSES
8037 ;TAPE STATUS ALERT TERMINATION WITH THE RECORD LENGTH
8038 ;SHORT (RLS) BIT SET. IT IS VERIFIED THAT THE
8039 ;RESIDUAL BYTE COUNTER (RBPCR) IN THE MESSAGE BUFFER
8040 ;CONTAINS THE PROPER NONZERO VALUE (E.G., THE
8041 ;DIFFERENCE BETWEEN THE ORIGINAL BYTE COUNT AND THE
8042 ;ACTUAL RECORD LENGTH). RESULTS ARE VERIFIED FOR BOTH
8043 ;STATES OF OPP (0 AND 1).
8044 |
8045 |
8046 |
8047 |
8048 :-
066610          BGNSUB                                ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
066610          T6.12:
066610 104402          TRAP    C1BSUB
8049 066612 004737 074540  JSR    PC,T26REST        ;SET COMMAND PACKET
8050 066616 004737 074632  JSR    PC,T26RT2       ;SET UP OTHER COMMAND PACKET
8051 066622 004737 074674  JSR    PC,T26RT3       ;SET UP OTHER COMMAND PACKET
8052
8053 ;*****
8054 |
8055 ;ISSUE CONTROLLER "SOFT" INITIALIZE CARRY BIT CLEAR IF ERROR
8056 |
8057 ;*****
8058
8059 066626 004737 016064  JSR    PC,SOFINIT     ;DO INITIALIZE ON CONTROLLER
8060 066632 103407          BCS    20$            ;BR IF INIT WAS OK
8061 066634 005237 002212  INC    FATFLG         ;BUMP COUNT
8065 066640 010001          MOV    R0,R1          ;CONTENTS OF TSSR REGISTER
8066 066642          ERRDF  ERRNO,SFIERR,SFIMSG      ;FATAL ERROR TSSR WAS NOT OK
066642 104455          TRAP    C1ERRDF
066644 001310          .WORD  712
066646 003650          .WORD  SFIERR
066650 012124          .WORD  SFIMSG
8067 066652 013737 002172 071650 20$:  MOV    UNITN,T26DSW    ;SET UP UNIT NUMBER
8068
8069 066660 012704 071630  MOV    @T26PACKET,R4  ;SUBROUTINE NEEDS PACKET ADDRESS
8070
8071 ;*****
8072 |
8073 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
8074 |
8075 ;*****
8076
8077 066664 004737 010752  JSR    PC,WRTPHR     ;ISSUE WRITE CHARACTERISTICS
8078 066670 103407          BCS    26$            ;BR, IF COMMAND ISSUED OK
8079 066672 005237 002212  INC    FATFLG         ;BUMP COUNT
8083 066676 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
8084 066700          ERRHRD  ERRNO,WRTPHR,SFIMSG      ;WRITE CHARACTERISTICS FAILED
066700 104456          TRAP    C1ERRHRD
066702 001311          .WORD  713
066704 005054          .WORD  WRTPHR

```

(t)

TEST 6: REREADS

```

8085 066706 012124
      066710 26$: CKLOOP ;LOOP IF SELECTED .WORD SFMSG
      066710 104406 ;TRAP C$CLP1
8086
8087 ;*****
8088 ;
8089 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8090 ;
8091 ;*****
8092
8093 066712 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
8094 066716 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
8095 066722 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
8096 066726 103407 BCS 30$ ;BR, IF NO PROBLEM
8097 066730 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
8098 066732 005237 002212 INC FATFLG ;BUMP COUNT
8102 066736 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      066736 104456 ;TRAP C$ERHRD
      066740 001312 ;.WORD 714
      066742 073304 ;.WORD T26RWN
      066744 012136 ;.WORD PKTSSR
8103 066746 30$: CKLOOP ;LOOP IF SELECTED .WORD C$CLP1
      066746 104406 ;TRAP
8104
8105 ;*****
8106 ;
8107 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8108 ;
8109 ;*****
8110
8111 066750 013701 071660 MOV T26BFR-6,R1 ;PICK UP XSTO
8112 066754 010102 MOV R1,R2 ;SET UP EXPECTED
8113 066756 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
8114 066762 020102 CMP R1,R2 ;DOES EXP = REC'D
8115 066764 001406 BEQ 40$ ;BR, IF EQUAL (OK)
8116 066766 005237 002212 INC FATFLG ;BUMP COUNT
8120 066772 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      066772 104456 ;TRAP C$ERHRD
      066774 001313 ;.WORD 715
      066776 073015 ;.WORD T26BOT
      067000 015564 ;.WORD EXPREC
8121 067002 40$: CKLOOP ;LOOP IF SELECTED .WORD C$CLP1
      067002 104406 ;TRAP
8122 067004 012703 000400 MOV #256.,R3 ;RECORD SIZE
8123 067010 013737 003114 071752 MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
8124
8125 ;*****
8126 ;
8127 ;WRITE DATA,CVC=1,ACK COMMAND
8128 ;
8129 ;*****
8130
8131 067016 012737 140005 071750 MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
8132 067024 012704 071750 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8133 067030
8134 067030 010337 071756 65$: MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
8135 067034 010465 000000 MOV R4,T5DB(R5) ;ISSUE COMMAND

```

De

TEST 6: REREADS

```

8136 067040 004737 016340      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
8137 067044 016501 000002      MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
8138 067050 012702 000200      MOV      #SSR,R2     ;SET UP EXPECTED
8139 067054 020102              CMP      R1,R2       ;ARE THEY EQUAL
8140 067056 001406              BEQ      75$         ;BR, IF OK
8141 067060 005237 002212      INC      FATFLG      ;BUMP COUNT
8145 067064              ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      067064 104456              TRAP    C$ERRRD
      067066 001314              .WORD  716
      067070 005111              .WORD  WRTERR
      067072 012136              .WORD  PKTSSR
8146 067074              75$:   CKLOOP          ;LOOP IF SELECTED
      067074 104406              TRAP    C$CLP1
8147 067076              120$:
8148
8149      ;*****
8150      ;
8151      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8152      ;
8153      ;*****
8154
8155 067076 004737 011104      JSR      PC,REWIND   ;CALL TAPE REWIND COMMAND
8156 067102 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR
8157 067106 012702 000200      MOV      #SSR,R2     ;SET UP EXPECTED TSSR
8158 067112 103407              BCS      130$        ;BR, IF NO PROBLEM
8159 067114 010004              MOV      RO,R4       ;PACKET ADDRESS SET UP
8160 067116 005237 002212      INC      FATFLG      ;BUMP COUNT
8164 067122              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      067122 104456              TRAP    C$ERRRD
      067124 001315              .WORD  717
      067126 073304              .WORD  T26RWN
      067130 012136              .WORD  PKTSSR
8165 067132              130$:  CKLOOP          ;LOOP IF SELECTED
      067132 104406              TRAP    C$CLP1
8166
8167      ;*****
8168      ;
8169      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8170      ;
8171      ;*****
8172
8173 067134 013701 071660      MOV      T26BFR+6,R1 ;PICK UP XSTO
8174 067140 010102              MOV      R1,R2       ;SET UP EXPECTED
8175 067142 052702 000002      BIS      #BIT1,R2    ;SET BOT BIT IN EXPECTED
8176 067146 020102              CMP      R1,R2       ;DOES EXP = REC'D
8177 067150 001406              BEQ      135$        ;BR, IF EQUAL (OK)
8178 067152 005237 002212      INC      FATFLG      ;BUMP COUNT
8182 067156              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      067156 104456              TRAP    C$ERRRD
      067160 001316              .WORD  718
      067162 073015              .WORD  T26BOT
      067164 015564              .WORD  EXPREC
8183 067166              135$:  CKLOOP          ;LOOP IF SELECTED
      067166 104406              TRAP    C$CLP1
8184 067170 012703 001000      MOV      #512.,R3    ;RECORD SIZE
8185 067174 013737 003114 071752  MOV      FREE,T26RB  ;STARTING READ BUFFER ADDRESS
8186

```

TEST 6: REREADS

```

8187
8188
8189
8190
8191
8192
8193 067202 012737 161401 071750
8194 067210 012704 071750
8195 067214 010337 071756
8196 067220 010465 000000
8197 067224 004737 016340
8198 067230 016501 000002
8199 067234 012702 100204
8200 067240 020102
8201 067242 001406
8202 067244 005237 002212
8206 067250
      067250 104456
      067252 001317
      067254 074362
      067256 012136
8207 067260
      067260 104406
8208
8209
8210
8211
8212
8213
8214
8215 067262 013701 071660
8216 067266 010102
8217 067270 052702 040000
8218 067274 020102
8219 067276 001406
8220 067300 005237 002212
8224 067304
      067304 104456
      067306 001320
      067310 074212
      067312 015564
8225 067314
      067314 104406
8226 067316 013701 071656
8227 067322 012702 000400
8228 067326 020102
8229 067330 001405
8233 067334
      067334 104456
      067336 001320
      067340 074274
      067342 015564
8234 067344
      067344 104406
8235 067346 012703 001000
8236 067352 013737 003114 071752
8237

```

```

;*****
;
;REREAD NEXT,ACK,CVC=1,OPP=1
;
;*****
      MOV      #161401,T26PK3      ;REREAD NEXT,ACK,CVC=1,OPP=1
165$:  MOV      #T26PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
      MOV      R3,T26SZ           ;SET UP RECORD SIZE IN PACKET
      MOV      R4,T26SDB(R5)      ;ISSUE COMMAND
      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
      MOV      T26SSR(R5),R1      ;GET T26SSR CONTENTS
      MOV      #T26SSR!SC!BIT2,R2 ;SET UP EXPECTED
      CMP      R1,R2             ;ARE THEY EQUAL
      BEQ      170$              ;BR, IF OK
      INC      FATFLG            ;BUMP COUNT
      ERRHRD  ERRNO,T26TRL,PKTSSR ;T26SSR INCORRECT AFTER READ DATA
                                TRAP  C$ERHRD
                                .WORD 719
                                .WORD T26TRL
                                .WORD PKTSSR
170$:  CKLOOP                      ;LOOP IF SELECTED
                                TRAP  C$CLP1
;*****
;
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;
;*****
      MOV      T26BFR+6,R1        ;GET MESSAGE BUFFER
      MOV      R1,R2             ;SET UP EXPECTED
      BIS      #BIT14,R2         ;SET THE RLS BIT IN EXPECTED
      CMP      R1,R2             ;ARE THEY EQUAL
      BEQ      180$              ;BR, IF EQUAL (ALL IS WELL)
      INC      FATFLG            ;BUMP COUNT
      ERRHRD  ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
                                TRAP  C$ERHRD
                                .WORD 720
                                .WORD T26LOP
                                .WORD EXPREC
180$:  CKLOOP                      ;LOOP IF SELECTED
                                TRAP  C$CLP1
      MOV      T26BFR+4,R1        ;PICK UP RESIDUAL BYTE COUNTER
      MOV      #256.,R2          ;THIS SHOULD BE THE DIFFERENCE
      CMP      R1,R2             ;IS THE DIFFERENCE CORRECT
      BEQ      190$              ;BR, IF CORRECT
      ERRHRD  ERRNO,T26PBP,EXPREC ;RBPCT NOT CORRECT
                                TRAP  C$ERHRD
                                .WORD 720
                                .WORD T26PBP
                                .WORD EXPREC
190$:  CKLOOP                      ;LOOP IF SELECTED
                                TRAP  C$CLP1
      MOV      #512.,R3          ;RECORD SIZE
      MOV      FREE,T26RB        ;STARTING READ BUFFER ADDRESS

```


(1)

SEQ 0278

TEST 6: REREADS

8287	067526	023727	002212	000017	
8288	067534	103402			
8289	067536	004737	017272		
8290	067542				999\$:

CMP	FATFLG,015.
BLO	999\$
JSR	PC.CKDROP

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

TEST 6: REREADS

```

8349 067646 010001          MOV    R0,R1          ;SAVE CONTENTS OF T5,R
8350 067650          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      067650 104456          TRAP    C$ERHRD
      067652 001324          .WORD  724
      067654 005054          .WORD  WRTMSG
      067656 012124          .WORD  SFIMSG
8351 067660          26$:   CKLOOP          ;LOOP IF SELECTED
      067660 104406          TRAP    C$CLP1
8352
8353          ;*****
8354          ;
8355          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8356          ;
8357          ;*****
8358
8359 067662 004737 021276    JSR    PC,INVERT      ;INVERT THE EXTENDED FEATURES SWITCH
8360 067666 004737 011104    JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
8361 067672 103411          BCS    30$           ;BR, IF NO PROBLEM
8362 067674 016501 000002    MOV    TSSR(R5),R1   ;GET TSSR
8363 067700 010004          MOV    R0,R4         ;PACKET ADDRESS SET UP
8364 067702 005237 002212    INC    FATFLG        ;BUMP COUNT
8368 067706          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      067706 104456          TRAP    C$ERHRD
      067710 001325          .WORD  725
      067712 073304          .WORD  T26RWN
      067714 012136          .WORD  PKTSSR
8369 067716          30$:   CKLOOP          ;LOOP IF SELECTED
      067716 104406          TRAP    C$CLP1
8370
8371          ;*****
8372          ;
8373          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8374          ;
8375          ;*****
8376
8377 067720 013701 071660    MOV    T26BFR+6,R1   ;PICK UP XSTO
8378 067724 010102          MOV    R1,R2         ;SET UP EXPECTED
8379 067726 052702 000002    BIS    #BIT1,R2      ;SET BOT BIT IN EXPECTED
8380 067732 020102          CMP    R1,R2         ;DOES EXP = REC'D
8381 067734 001406          BEQ    40$           ;BR, IF EQUAL (OK)
8382 067736 005237 002212    INC    FATFLG        ;BUMP COUNT
8386 067742          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      067742 104456          TRAP    C$ERHRD
      067744 001326          .WORD  726
      067746 073015          .WORD  T26BOT
      067750 015564          .WORD  EXPREC
8387 067752          40$:   CKLOOP          ;LOOP IF SELECTED
      067752 104406          TRAP    C$CLP1
8388 067754 013737 003114 071752  MOV    FREE,T26RB    ;STARTING WRITE BUFFER ADDRESS
8389
8390          ;*****
8391          ;
8392          ;WRITE DATA,CVC=1,ACK COMMAND
8393          ;
8394          ;*****
8395
8396 067762 012737 140005 071750  MOV    #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND

```

TEST 6: REREADS

```

8397 067770 012704 071750      MOV      #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
8398 067774 012737 000400 071756 65$:  MOV      #256.,T26SZ    ;SET UP RECORD SIZE IN PACKET
8399 070002 013777 071776 113104  MOV      T26CNT,8FREE   ;MOVE TAPE RECORD NUMBER TO BUFFER
8400 070010 062737 000001 071776  ADD      #1,T26CNT      ;NUMBER READY FOR NEXT RECORD
8401 070016 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
8402 070022 004737 016340      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
8403 070026 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
8404 070032 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED
8405 070036 020102      CMP      R1,R2         ;ARE THEY EQUAL
8406 070040 001406      BEQ      75$           ;BR, IF OK
8407 070042 005237 002212      INC      FATFLG        ;BUMP COUNT
8411 070046      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      070046 104456      TRAP    C$ERRHRD
      070050 001327      .WORD  727
      070052 005111      .WORD  WRTErr
      070054 012136      .WORD  PKTSSR
8412 070056      75$:  CKLOOP          ;LOOP IF SELECTED
      070056 104406      TRAP    C$CLP1
8413 070060 022737 000013 071776  CMP      #11.,T26CNT   ;CHECK NUMBER OF RECORDS WRITTEN
8414 070066 001401      BEQ      120$         ;BR, IF AT END OF WRITE SEQUENCE
8415 070070 000741      BR       65$          ;WRITE MORE RECORDS
8416 070072      120$:  CLR      T26CNT      ;SET RECORD COUNTER BACK TO ZERO
8417 070072 005037 071776
8418
8419 ;*****
8420 ;
8421 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8422 ;
8423 ;*****
8424
8425 070076 004737 011104      JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
8426 070102 103411      BCS     130$          ;BR, IF NO PROBLEM
8427 070104 016501 000002      MOV      TSSR(R5),R1  ;GET TSSR
8428 070110 010004      MOV      R0,R4        ;PACKET ADDRESS SET UP
8429 070112 005237 002212      INC      FATFLG       ;BUMP COUNT
8433 070116      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      070116 104456      TRAP    C$ERRHRD
      070120 001330      .WORD  728
      070122 073304      .WORD  T26RWN
      070124 012136      .WORD  PKTSSR
8434 070126      130$:  CKLOOP          ;LOOP IF SELECTED
      070126 104406      TRAP    C$CLP1
8435
8436 ;*****
8437 ;
8438 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8439 ;
8440 ;*****
8441
8442 070130 013701 071660      MOV      T268FR+6,R1  ;PICK UP XSTO
8443 070134 010102      MOV      R1,R2        ;SET UP EXPECTED
8444 070136 052702 000002      BIS     #BIT1,R2      ;SET BOT BIT IN EXPECTED
8445 070142 020102      CMP     R1,R2         ;DOES EXP = REC'D
8446 070144 001406      BEQ     140$         ;BR, IF EQUAL (OK)
8447 070146 005237 002212      INC     FATFLG       ;BUMP COUNT
8451 070152      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      070152 104456      TRAP    C$ERRHRD

```

TEST 6: REREADS

```

070154 001331
070156 073015
070160 015564
8452 070162 140$: CKLOOP ;LOOP IF SELECTED
070162 104406 ;TRAP C$CLP1
8453 070164 012703 071766 MOV #T26RN,R3 ;COMMAND BUFFER ADDRESS
8454 070170 013737 003130 071752 150$: MOV NXML0,T26RB ;STARTING READ BUFFER ADDRESS
8455 070176 013737 003132 071754 MOV NXMMI,T26RB+2 ;SET UP HIGH ORDER ADDRESS BITS
8456
8457 ;*****
8458 ;
8459 ;REREAD DATA,IE,ACK, OPP COMMAND
8460 ;
8461 ;*****
8462
8463 070204 011337 071750 MOV (R3),T26PK3 ;REREAD DATA,IE,ACK, OPP COMMAND
8464 070210 012704 071750 165$: MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8465 070214 012737 000400 071756 MOV #256.,T26SZ ;SET UP RECORD SIZE IN PACKET
8466 070222 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
8467 070226 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
8468 070232 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8469 070236 012702 104210 MOV #SSR!NXM!SC!BIT3,R2 ;SET UP EXPECTED
8470 070242 020102 CMP R1,R2 ;ARE THEY EQUAL
8471 070244 001414 BEQ 170$ ;BR, IF OK
8472 070246 031327 001000 BIT (R3),#BIT9 ;CHECK FOR A READ COMMAND
8473 070252 001403 BEQ 168$ ;BR, IF IT WAS A READ COMMAND
8474 070254 030127 000002 BIT R1,#BIT1 ;WAS BIT1 SET
8475 070260 001006 BNE 170$ ;BR, IF REREAD AND BIT1 SET
8476 070262
8477 070262 005237 002212 168$: INC FATFLG ;BUMP COUNT
8481 070266 ERRHRD ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
070266 104456 ;TRAP C$ERHRD
070270 001332 ;WORD 730
070272 072225 ;WORD T26RRF
070274 012136 ;WORD PKTSSR
8482 070276 170$: CKLOOP ;LOOP IF SELECTED
070276 104406 ;TRAP C$CLP1
8483
8484 ;*****
8485 ;
8486 ;READ DATA, ACK,CVC=1 COMMAND
8487 ;
8488 ;*****
8489
8490 070300 012737 140001 071750 MOV #140001,T26PK3 ;READ DATA, ACK,CVC=1 COMMAND
8491 070306 012737 000400 071756 MOV #256.,T26SZ ;SET SIZE INTO PACKET
8492 070314 005037 071754 CLR T26RB+2 ;CLEAR OUT HIGH ADDRESS BITS
8493 070320 013737 003114 071752 MOV FREE,T26RB ;GIVE READ A GOOD BUFFER
8494 070326 010465 000000 MOV R4,TSDB(R5) ;ISSUE READ DATA COMMAND
8495 070332 004737 016340 JSR PC,WAITF ;WAIT FOR SSR
8496 070336 016501 000002 MOV TSSR(R5),R1 ;PICK UP THE TSSR
8497 070342 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
8498 070346 020102 CMP R1,R2 ;IS THE TSSR OK
8499 070350 001406 BEQ 180$ ;BR, IF TSSR OK (GOOD)
8500 070352 005237 002212 INC FATFLG ;BUMP COUNT
8504 070356 ERRHRD ERRNO,RDERR,PKTSSR ;READ DATA COMMAND FAILED
070356 104456 ;TRAP C$ERHRD

```

TEST 6: REREADS

```

070360 001333 .WORD 731
070362 005204 .WORD RDERR
070364 012136 .WORD PKTSSR
8505 070366 180$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
070366 104406 ;FIRST WORD FROM READ BUFFER
8506 070370 017701 112520 MOV $FREE,R1 ;SET UP EXPECTED
8507 070374 012702 000001 MOV #1,R2 ;IS TAPE POSITION CORRECT
8508 070400 020102 CMP R1,R2 ;KEEP GOING POSITION OK
8509 070402 001406 BEQ 190$ ;BUMP COUNT
9510 070404 005237 002212 INC FATFLG ;TAPE POSITION INCORRECT
8514 070410 ERRHRD ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
070410 104456 TRAP C$ERHRD
070412 001334 .WORD 732
070414 072006 .WORD T26WNG
070416 015564 .WORD EXPREC
8515 070420 190$: CKLOOP TRAP C$CLP1
070420 104406
8516
8517 ;*****
8518 ;
8519 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8520 ;
8521 ;*****
8522
8523 070422 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
8524 070426 103411 BCS 194$ ;BR, IF NO PROBLEM
8525 070430 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
8526 070434 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
8527 070436 005237 002212 INC FATFLG ;BUMP COUNT
8531 070442 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
070442 104456 TRAP C$ERHRD
070444 001335 .WORD 733
070446 073304 .WORD T26RWN
070450 012136 .WORD PKTSSR
8532 070452 194$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
070452 104406
8533
8534 ;*****
8535 ;
8536 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8537 ;
8538 ;*****
8539
8540 070454 013701 071660 MOV T26BFR+6,R1 ;PICK UP XSTO
8541 070460 010102 MOV R1,R2 ;SET UP EXPECTED
8542 070462 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
8543 070466 020102 CMP R1,R2 ;DOES EXP = REC'D
8544 070470 001406 BEQ 196$ ;BR, IF EQUAL (OK)
8545 070472 005237 002212 INC FATFLG ;BUMP COUNT
8549 070476 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
070476 104456 TRAP C$ERHRD
070500 001336 .WORD 734
070502 073015 .WORD T26BOT
070504 015564 .WORD EXPREC
8550 070506 196$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
070506 104406
8551 070510 010302 MOV R3,R2 ;SAVE R3 FOR A MOMENT

```


TEST 6: REREADS

```

8625
8626
8627
8628
8629
8630
8631
8632 070656 004737 011104
8633 070662 016501 000002
8634 070666 012702 000200
8635 070672 103407
8636 070674 010004
8637 070676 005237 002212
8641 070702
      070702 104456
      070704 001341
      070706 073304
      070710 012136
8642 070712
      070712 104406
8643
8644
8645
8646
8647
8648
8649
8650 070714 013701 071660
8651 070720 010102
8652 070722 052702 000002
8653 070726 020102
8654 070730 001406
8655 070732 005237 002212
8659 070736
      070736 104456
      070740 001342
      070742 073015
      070744 015564
8660 070746
      070746 104406
8661 070750 012737 000400 071756
8662 070756 013737 003114 071752
8663 070764 005703
8664 070766 001404
8665
8666
8667
8668
8669
8670
8671
8672 070770 012737 161001 071750
8673 070776 000403
8674
8675
8676
8677

```

```

;*****
;
;ISSUE REWIND COMMAND TO SELECTED TAPF DRIVE
;
;*****
26$: JSR PC,REWIND ;CALL TAPE REWIND COMMAND
      MOV TSSR(R5),R1 ;GET TSSR
      MOV #SSR,R2 ;SET UP EXPECTED TSSR
      BCS 30$ ;BR, IF NO PROBLEM
      MOV R0,R4 ;PACKET ADDRESS SET UP
      INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                     TRAP C$ERHRD
                                     .WORD 737
30$: CKLOOP ;LOOP IF SELECTED
                                     .WORD T26RWN
                                     .WORD PKTSSR
                                     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$ERHRD
                                     .WORD 738
40$: CKLOOP ;LOOP IF SELECTED
                                     .WORD T26BOT
                                     .WORD EXPREC
                                     TRAP C$CLP1
      MOV #256.,T26SZ ;SET UP RECORD SIZE IN PACKET
      MOV FREE,T26RB ;ADDRESS OF READ BUFFER
      TST R3 ;CHECK NUMBER OF TIMES THROUGH HERE
      BEQ 50$ ;BR, IF FIRST TIME THROUGH HERE
;*****
;
;REREAD,CVC=1,ACK COMMAND
;
;*****
      MOV #161001,T26PK3 ;REREAD,CVC=1,ACK COMMAND
      BR 55$ ;SKIP NEXT COMMAND
;*****
;
;REREAD,ACK COMMAND

```


TEST 6: REREADS

```

8678
8679
8680
8681 071000 012737 141001 071750 50$: MOV #141001,T26PK3 ;REREAD,ACK COMMAND
8682 071006 55$:
8683 071006 012704 071750 65$: MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8684 071012
8685 071012 010465 000000 MOV R4,TSD8(R5) ;ISSUE COMMAND
8686 071016 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
8687 071022 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8688 071026 012702 100206 MOV #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
8689 071032 020102 CMP R1,R2 ;ARE THEY EQUAL
8690 071034 001406 BEQ 75$ ;BR, IF OK
8691 071036 005237 002212 INC FATFLG ;BUMP COUNT
8695 071042 ERRHRD ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      071042 104456 TRAP C$ERHRD
      071044 001343 .WORD 739
      071046 072743 .WORD T26WDE
      071050 012136 .WORD PKTSSR
8696 071052 75$: CKLOOP ;LOOP IF SELECTED
      071052 104406 TRAP C$CLP1
8697
8698
8699
8700 ;*****
8701 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8702 ;
8703 ;*****
8704 071054 C13701 071660 MOV T26BFR+6,R1 ;GET XSTO STATUS WORD
8705 071060 010102 MOV R1,R2 ;SET UP EXPECTED
8706 071062 052702 002000 BIS #BIT10,R2 ;SET THE NEF BIT
8707 071066 020102 CMP R1,R2 ;ARE THEY EQUAL
8708 071070 001406 BEQ 170$ ;BR, IF EQUAL (GOOD)
8709 071072 005237 002212 INC FATFLG ;BUMP COUNT
8713 071076 ERRHRD ERRNO,T26NEF,EXPREC ;NEF SHOULD BE SET
      071076 104456 TRAP C$ERHRD
      071100 001344 .WORD 740
      071102 072074 .WORD T26NEF
      071104 015564 .WORD EXPREC
8714 071106 170$: CKLOOP TRAP C$CLP1
      071106 104406
8715 071110 005103 COM R3 ;RESET THE SWITCH
8716 071112 001261 BNE 26$ ;BR, IF FIRST TIME THROUGH HERE
8717 071114 ENDSUB
      071114
      071114 104403 L10120: TRAP C$ESUB
8718 071116 023727 002212 000017 CMP FATFLG,#15. ;IS ERROR COUNT AT 25
8719 071124 103402 BLO 999$ ;BR, IF LESS THAN 25
8720 071126 004737 017272 JSR PC,CKDROP ;TRY TO DROP THE UNIT
8721 071132 999$:

```


TEST 6: REREADS

```

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

```

TEST 6: REREADS

```

071360 001351 .WORD 745
071362 072743 .WORD T26WDE
071364 012136 .WORD PKTSSR
8830 071366 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
071366 104406

8831
8832 ;*****
8833 ;
8834 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
8835 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
8836 ;
8837 ;*****
8838
8839 071370 012703 100001 MOV #10000,R3 ;SET SPACE REVERSE 1 RECORD
8840 071374 004737 010556 JSR PC,SPACE ;ISSUE COMMAND
8841 071400 103411 BCS 175$ ;GO ON IF ALL IS WELL
8842 071402 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8843 071406 010004 MOV R0,R4 ;SET UP EXPECTED (PACKET CONTENTS)
8844 071410 005237 002212 INC FATFLG ;BUMP COUNT
8848 071414 ERRHRD ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
071414 104456 TRAP C$ERHRD
071416 001352 .WORD 746
071420 072743 .WORD T26WDE
071422 012136 .WORD PKTSSR
8849 071424 175$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
071424 104406
8850 071426 013737 003114 071752 MOV FREE,T26RB ;ADDRESS OF BUFFER
8851 071434 005737 072000 TST T26CNU ;CHECK FOR TIMES THROUGH HERE
8852 071440 001404 BEQ 176$ ;BR, IF FIRST TIME THROUGH
8853
8854 ;*****
8855 ;
8856 ;REREAD (PREVIOUS),IE,ACK,OPP=1 CMD.
8857 ;
8858 ;*****
8859
8860 071442 012737 161001 071750 MOV #161001,T26PK3 ;REREAD (PREVIOUS),IE,ACK,OPP=1 CMD.
8861 071450 000403 BR 178$ ;SKIP NEXT COMMAND
8862
8863 ;*****
8864 ;
8865 ;REREAD ,ACK,OPP=1 COMMAND
8866 ;
8867 ;*****
8868
8869 071452 012737 141001 071750 176$: MOV #141001,T26PK3 ;REREAD ,ACK,OPP=1 COMMAND
8870 071460 178$:
8871 071460 012704 071750 MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8872 071464 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
8873 071470 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
8874 071474 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8875 071500 012702 100204 MOV #SSR!SC!BIT2,R2 ;SET UP EXPECTED
8876 071504 020102 CMP R1,R2 ;ARE THEY EQUAL
8877 071506 001406 BEQ 180$ ;BR, IF OK
8878 071510 005237 002212 INC FATFLG ;BUMP COUNT
8882 071514 ERRHRD ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
071514 104456 TRAP C$ERHRD

```


TEST 6: REREADS

```

8911
8912
8913
8915      071630
8917 071630
8913 071630 014004
8919 071632 071640
8920 071634 000000
8921 071636 000012
8922 071640
8923 071640 071652
8924 071642 000000
8925 071644 000024
8926 071646 000000
8927 071650 000000
8928 071652
8929
8930
8931
8933      071740
8935 071740
8936 071740 100006
8937 071742 071760
8938 071744 000000
8939 071746 000006
8940
8944 071750
8945 071750 140005
8946 071752
8947 071752 003114
8948 071754 000000
8949 071756 000000
8950
8951
8952
8953
8954 071760
8955 071760      010
8956 071761      200
8957 071762 000000
8958 071764 000000
8959
8960
8961
8962
8963
8964 071766 140001
8965 071770 141401
8966 071772 161401
8967 071774 177777
8968
8969
8970 071776 000000
8971 072000 000000
8972
8973 072002 000000
8974
;
; LOCAL STORAGE FOR THIS TEST
;
;
; 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
; T26DATA: ; CHARACTERISTICS DATA BLOCK
; .WORD T26BFR ; ADDRESS OF MESSAGE BUFFER
; .WORD 0
; .WORD 20. ; LENGTH OF MESSAGE BUFFER
; .WORD 0
; T26DSW: .WORD 0 ; SELECT DRIVE 0
; T26BFR: .BLKW 25. ; MESSAGE BUFFER
;
; WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
; T26PK2:
; .WORD 100006 ; WRITE SUB SYS MEM COMMAND, AND ACK
; .WORD T26BF2 ; ADDRESS OF SELECT BLOCK DATA
; .WORD 0
; .WORD 6. ; SIZE OF DATA PACKET
;
; T26PK3:
; .WORD 140005 ; REREAD COMMAND, CVC=1 AND ACK
; T26RB:
; T26WB: .WORD FREE ; ADDRESS OF WRITE BUFFER
; .WORD 0
; T26SZ: .WORD 0 ; SIZE OF BUFFER (EXTENT)
; .EVEN
;
;
; T26BF2:
; T26BS0: .BYTE 10 ; BSELO AREA
; T26BS1: .BYTE 200 ; BSEL1 AREA
; T26S2: .WORD 0 ; SEL 2 AREA
; T26S3: .WORD 0 ; DATA AREA
;
;
; .EVEN
; TAPE 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

```

TSV* HARDWARE TESTS 1 8

MACRO M1200 23 MAR 84 09:44 PAGE 87 1

17

TEST 6: REREADS

SEG 0293

8975 072004 000000
8976

T?6DLY: .WORD 0

;DELAY COUNTER AREA

TEST 6: REREADS

```

8978
8979
8980
8981
8982
8983
8984 072006      124      141      160 T26WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
8985 072074      122      105      122 T26NEF: .ASCIZ 'REREAD PREVIOUS, At BOT, Failed To Set NEF (XST0)'
8986 072156      124      123      123 T26RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
8987 072225      122      105      122 T26RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
8988 072322      122      105      122 T26RRG: .ASCIZ 'REREAD Previous (Read Reverse, Space Forward) Command Failed'
8989 072417      120      117      123 T26SC:  .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
8990 072501      122      111      102 T26LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
8991 072551      124      123      123 T26WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
8992 072626      111      154      154 T26LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
8993 072707      122      105      122 T26SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
8994 072743      124      123      123 T26WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
8995 073015      124      141      160 T26BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
8996 073062      104      141      164 T26DTA: .ASCIZ 'Data Written To Tape Not Equal To Data Read From Tape'
8997 073150      122      105      122 T26EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
8998 073227      124      123      123 T26TM:  .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
8999 073304      122      145      167 T26RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
9000 073353      122      101      115 T26RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
9001 073426      124      123      123 T26AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
9002 073475      104      162      151 T26OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
9003 073550      124      123      123 T26WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
9004 073640      124      123      123 T26WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
9005 073713      103      126      103 T26VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
9006 073766      124      123      102 T26BA:  .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
9007 074041      127      122      111 T26WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
9008 074130      122      145      141 T26LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
9009 074212      122      145      141 T26LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
9010 074274      122      145      163 T26PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
9011 074362      122      145      141 T26TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
9012 074450      104      141      164 T26NEQ: .ASCIZ 'Data REREAD From Tape Not Correct, After SWB=1'
9013 074527      122      145      162 T26ID:  .ASCIZ 'Rereads'
9014
9015
9016
9017
9018
9019
9020
9021
9022 074540
9023 074540
9024 074544      012701  071630
9025 074550      012721  140004
9026 074554      012721  071640
9027 074560      005021
9028 074562      012721  000012
9029 074566      012721  071652
9030 074572      005021
9031 074574      012721  000024
9032 074600      005021
9033 074602      012711  000000
9034 074606      012702  000030

```

```

;*
;LOCAL TEXT MESSAGES FOR TEST
;

```

```

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

```

```

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

```


TEST 6: REREADS

```

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

```


TEST 7: WRITE DATA RETRI

```

075040 005367 177772          DEC      -6(PC)
075044 001375                BNE      . 4
075046 005367 177756          DEC     -22(PC)
075052 001367                BNF      . 20
9118 075054 005337 101612          DEC      T27DLY      ;BUMP COUNTER
9119 075060 001356          BNE      10$         ;BR, IF COUNTER NOT DONE
9120 075062 005237 002212          INC      FATFLG     ;BUMP COUNT
9124 075066 010001          MOV      R0,R1      ;CONTENTS OF TSSR REGISTER
9125 075070          ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
075070 104455                TRAP     C$ERDF
075072 001275                .WORD   701
075074 003650                .WORD   SFIERR
075076 012124                .WORD   SFIMSG
9126 075100 013737 002172 101460 20$:  MOV      UNITN,T27DSW      ;SET UP DRIVE NUMBER
9127 075106 012704 101440          MOV      @T27PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
9128
9129          ;*****
9130          ;
9131          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
9132          ;
9133          ;*****
9134
9135 075112 004737 010752          JSR      PC,WRTPHR   ;ISSUE WRITE CHARACTERISTICS
9136 075116 103407          BCS      25$         ;BR, IF COMMAND ISSUED OK
9137 075120 005237 002212          INC      FATFLG     ;BUMP COUNT
9141 075124 010001          MOV      R0,R1      ;SAVE CONTENTS OF TSSR
9142 075126          ERRHRD ERRNO,WRTPHR,SFIMSG ;WRITE CHARACTERISTIC FAILED
075126 104456                TRAP     C$ERRHRD
075130 001276                .WORD   702
075132 005054                .WORD   WRTPHR
075134 012124                .WORD   SFIMSG
9143 075136          25$:  CKLOOP          ;LOOP IF SELECTED
075136 104406                TRAP     C$CLP1
9144
9145          ;*****
9146          ;
9147          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9148          ;
9149          ;*****
9150
9151 075140 004737 011104          JSR      PC,REWIND   ;CALL TAPE REWIND COMMAND
9152 075144 103407          BCS      30$         ;BR, IF NO PROBLEM
9153 075146 010004          MOV      R0,R4      ;SET UP REWIND PACKET ADDRESS
9154 075150 005237 002212          INC      FATFLG     ;BUMP COUNT
9158 075154          ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
075154 104456                TRAP     C$ERRHRD
075156 001277                .WORD   703
075160 102765                .WORD   T27RWN
075162 012136                .WORD   PKTSSR
9159 075164          30$:  CKLOOP          ;LOOP IF SELECTED
075164 104406                TRAP     C$CLP1
9160
9161          ;*****
9162          ;
9163          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
9164          ;
9165          ;*****

```

TEST 7: WRITE DATA RETRY

```

9166
9167 075166 013701 101470      MOV      T27BFR+6,R1      ;PICK UP XSTO
9168 075172 010102      MOV      R1,R2           ;SET UP EXPECTED
9169 075174 052702 000002      BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
9170 075200 020102      CMP      R1,R2           ;DOES EXP = REC'D
9171 075202 001406      BEQ      40$             ;BR, IF EQUAL (OK)
9172 075204 005237 002212      INC      FATFLG          ;BUMP COUNT
9176 075210      ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    704
                                .WORD    T27BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
9177 075220      40$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
9178 075222 012737 000400 10.566      MOV      #256.,T27SZ     ;SET UP RECORD SIZE
9179 075230 013737 003114 101562      MOV      FREE,T27WB      ;ADDRESS OF WRITE BUFFER
9180
9181      ;*****
9182      ;
9183      ;WRITE DATA RETRY,ACK,CVC=1 COMMAND
9184      ;
9185      ;*****
9186
9187 075236 012737 141005 101560      MOV      #141005,T27PK3  ;WRITE DATA RETRY,ACK,CVC=1 COMMAND
9188 075244 012704 101560      MOV      #T27PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
9189 075250 010465 000000      MOV      R4,TSD8(R5)     ;ISSUE COMMAND
9190 075254 00 737 016340      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
9191 075260 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
9192 075264 012702 100206      MOV      #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
9193 075270 020102      CMP      R1,R2           ;ARE THEY EQUAL
9194 075272 001406      BEQ      75$             ;BR, IF OK
9195 075274 005237 002212      INC      FATFLG          ;BUMP COUNT
9199 075300      ERRHRD  ERRNO,T27WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    705
                                .WORD    T27WDE
                                .WORD    PKTSSR
9200 075310      75$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
9201
9202      ;*****
9203      ;
9204      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9205      ;
9206      ;*****
9207
9208 075312 013701 101470      MOV      T27BFR+6,R1     ;GET XSTO STATUS WORD
9209 075316 010102      MOV      R1,R2           ;SET UP EXPECTED
9210 075320 052702 002000      BIS      #BIT10,R2       ;SET THE NEF BIT
9211 075324 020102      CMP      R1,R2           ;ARE THEY EQUAL
9212 075326 001406      BEQ      170$            ;BR, IF EQUAL (GOOD)
9213 075330 005237 002212      INC      FATFLG          ;BUMP COUNT
9217 075334      ERRHRD  ERRNO,T27NEF,EXPREC ;NEF SHOULD BE SET
                                TRAP      C$ERHRD
                                .WORD    706
                                .WORD    T27NEF
                                .WORD    EXPREC
9218 075334 104456
9219 075336 001302
9220 075340 104131
9221 075342 015564

```

TEST 7: WRITE DATA RETRY

```

9218 075344          1'08:  CMLDOP
      075344 104406
9219 075346          ENDSUB
      075346
      075346 104403
9220 075350 023727 002212 00001'  CMP  FATFLG.#15.
9221 075356 103402          HLO  9998
9222 075360 004737 017272          JSR  PC,CKDROP
9223 075364          9998:

```

L10123: TRAP C0ESUB
;IS ERROR COUNT AT 25
;BR. IF LESS THAN 25
;TRY TO DROP THE UNIT

TEST 7: WRITE DATA RETRY

```

9225 ;
9226 ;
9227 ;TEST 7, SUBTEST 2
9228 ;
9229 ;VERIFIES THAT WRITE DATA RETRY COMMAND ISSUED WHILE
9230 ;THE TAPE IS POSITIONED BEFORE THE FIRST RECORD ON
9231 ;TAPE (BUT NOT AT BOT) RESULTS IN TAPE STATUS ALERT
9232 ;TERMINATION, WITH THE REVERSE INTO BOT (RIB) STATUS
9233 ;ERROR BIT SET.
9234 ;
9235 ;
9236 ;
9237 ;
9238 075364                BGNSUB                               ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>
        075364                T7.2:                               TRAP      C%BSUB
        075364    104402                JSR      PC,T27REST          ;SET COMMAND PACKET
9239 075366    004737    104564                JSR      PC,T27RT2          ;SET UP OTHER COMMAND PACKET
9240 075372    004737    104656                JSR      PC,T27RT3          ;SET UP OTHER COMMAND PACKET
9241 075376    004737    104720
9242 ;
9243 ;*****
9244 ;
9245 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
9246 ;
9247 ;*****
9248 ;
9249 075402    004737    016064                JSR      PC,SOFINIT        ;DO INITIALIZE ON CONTROLLER
9250 075406    103407                BCS     20%                ;BR IF INIT WAS OK
9251 075410    005237    002212                INC     FATFLG            ;BUMP COUNT
9255 075414    010001                MOV     R0,R1             ;CONTENTS OF TSSR REGISTER
9256 075416    104455    ERRDF    ERRNO,SFIERR,SFIMSG    ;FATAL ERROR TSSR WAS NOT OK
        075416    104455                TRAP      C%ERDF
        075420    001303                .WORD   707
        075422    003650                .WORD   SFIERR
        075424    012124                .WORD   SFIMSG
9257 075426    013737    002172    101460    20%:  MOV     UNITN,T27DSW          ;SET UP DRIVE NUMBER
9258 075434    012704    101440                MOV     @T27PACKET,R4     ;SUBROUTINE NEEDS PACKET ADDRESS
9259 ;
9260 ;*****
9261 ;
9262 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
9263 ;
9264 ;*****
9265 ;
9266 075440    004737    010752                JSR      PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
9267 075444    103407                BCS     25%                ;BR, IF COMMAND ISSUED OK
9268 075446    005237    002212                INC     FATFLG            ;BUMP COUNT
9272 075452    010001                MOV     R0,R1             ;SAVE CONTENTS OF TSSR
9273 075454    104456    ERRHRD    ERRNO,WRTMSG,SFIMSG    ;WRITE CHARACTERISTICSC FAILED
        075454    104456                TRAP      C%ERHRD
        075456    001304                .WORD   708
        075460    005054                .WORD   WRTMSG
        075462    012124                .WORD   SFIMSG
9274 075464    104406    25%:  CKLOOP                      ;LOOP IF SELECTED
        075464    104406                TRAP      C%CLP1
9275 ;
9276 ;*****

```

TEST 7: WRITE DATA REWI

```

9277
9278
9279
9280
9281
9282 075466 004737 011104
9283 075472 103411
9284 075474 010004
9285 075476 016501 000002
9286 075502 005237 002212
9290 075506
    075506 104456
    075510 001305
    075512 102765
    075514 012136
9291 075516
    075516 104406
9292 075520 012703 000400
9293 075524 013737 003114 101562
9294
9295
9296
9297
9298
9299
9300
9301 075532 012737 140005 101560
9302 075540 012704 101560
9303 075544 010337 101566
9304 075550 010465 000000
9305 075554 004737 016340
9306 075560 016501 000002
9307 075564 012702 000200
9308 075570 020102
9309 075572 001406
9310 075574 005237 002212
9314 075600
    075600 104456
    075602 001306
    075604 005111
    075606 012136
9315 075610
    075610 104406
9316
9317
9318
9319
9320
9321
9322
9323 075612 004737 011104
9324 075616 103411
9325 075620 016501 000002
9326 075624 010004
9327 075626 005237 002212
9331 075632
    075632 104456

```

```

;
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
;CALL TAPE REWIND COMMAND
;BR, IF NO PROBLEM
;SET UP REWIND PACKET ADDRESS
;GET TSSR CONTENTS
;BUMP COUNT
;REWIND NOT ACCEPTED
;*****
TRAP C$ERRRD
.WORD 709
.WORD T27RWN
.WORD PKTSSR
26$: 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,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
;*****
TRAP C$ERRRD
.WORD 710
.WORD WRTErr
.WORD PKTSSR
28$: CKLOOP ;LOOP IF SELECTED
TRAP C$CLP1
;*****
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;*****
;CALL TAPE REWIND COMMAND
;BR, IF NO PROBLEM
;GET TSSR CONTENTS
;SET UP REWIND PACKET ADDRESS
;BUMP COUNT
;REWIND NOT ACCEPTED
;*****
TRAP C$ERRRD

```

TEST 7: WRITE DATA RETRY

```

075634 001307 .WORD 711
075636 102765 .WORD T27RWN
075640 012136 .WORD PKTSSR
9332 075642 104406 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
075642 104406
9333
9334 ;*****
9335 ;
9336 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST%)
9337 ;
9338 ;*****
9339
9340 075644 013701 101470 MOV T27BFR+6,R1 ;PICK UP XSTO
9341 075650 010102 MOV R1,R2 ;SET UP EXPECTED
9342 075652 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
9343 075656 020102 CMP R1,R2 ;DOES EXP = REC'D
9344 075660 001406 BEQ 40$ ;BR, IF EQUAL (OK)
9345 075662 005237 002212 INC FATFLG ;BUMP COUNT
9349 075666 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
075666 104456 TRAP C$ERHRD
075670 001310 .WORD 712
075672 102461 .WORD T27BOT
075674 015564 .WORD EXPREC
9350 075676 104406 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
075676 104406
9351
9352 ;*****
9353 ;
9354 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
9355 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
9356 ;
9357 ;*****
9358
9359 075700 012703 000001 MOV #1,R3 ;PARAMETER SPACE FORWARD 1 RECORD
9360 075704 004737 010556 JSR PC,SPACE ;CALL SPACE RECORDS ROUTINE
9361 075710 103413 BCS 50$ ;BR, IF NO ERRORS
9362 075712 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9363 075716 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9364 075722 010004 MOV R0,R4 ;SET UP REWIND PACKET ADDRESS
9365 075724 005237 002212 INC FATFLG ;BUMP COUNT
9369 075730 ERRHRD ERRNO,T27SCF,PKTSSR ;SPACE RECORDS COMMAND FAILED
075730 104456 TRAP C$ERHRD
075732 001311 .WORD 713
075734 104227 .WORD T27SCF
075736 012136 .WORD PKTSSR
9370 075740 104406 50$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
075740 104406
9371
9372 ;*****
9373 ;
9374 ;ISSUE SPACE RECORDS COMMAND VALUE IN R3 SETS NUMBER OF RECORDS
9375 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
9376 ;
9377 ;*****
9378
9379 075742 012703 100001 MOV #100001,R3 ;PARAMETER SPACE REVERSE 1 RECORD
9380 075746 004737 010556 JSR PC,SPACE ;CALL SPACE RECORDS ROUTINE

```


TEST 7: WRITE DATA RETRI

```

9483 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9484 ;
9485 ;*****
9486 ;
9487 076250 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
9488 076254 103407 BCS 30$ ;BR, IF NO PROBLEM
9489 076256 010004 MOV R0,R4 ;SET UP REWIND PACKET ADDRESS
9490 076260 005237 002212 INC FATFLG ;BUMP COUNT
9494 076264 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERHRD
; .WORD 719
; .WORD T27RWN
; .WORD PKTSSR
9495 076274 30$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
; .WORD
9496 ;*****
9497 ;
9498 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
9499 ;
9500 ;*****
9501 ;
9502 ;
9503 076276 013701 101470 MOV T27BFR+6,R1 ;PICK UP YST0
9504 076302 010102 MOV R1,R2 ;SET UP EXPECTED
9505 076304 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
9506 076310 020102 CMP R1,R2 ;DOES EXP = REC'D
9507 076312 001406 BEQ 40$ ;BR, IF EQUAL (OK)
9508 076314 005237 002212 INC FATFLG ;BUMP COUNT
9512 076320 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
; TRAP C$ERHRD
; .WORD 720
; .WORD T27BOT
; .WORD EXPREC
9513 076330 40$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
; .WORD
9514 076332 012703 000024 MOV #20.,R3 ;STARTING RECORD SIZE
9515 076336 013737 003114 101562 MOV FREE,T27WB ;STARTING WRITE BUFFER ADDRESS
9516 ;*****
9517 ;
9518 ;WRITE DATA,CVC=1,ACK COMMAND
9519 ;
9520 ;*****
9521 ;
9522 ;
9523 076344 012737 140005 101560 65$: MOV #140005,T27PK3 ;WRITE DATA,CVC=1,ACK COMMAND
9524 076352 012704 101560 MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
9525 076356 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
9526 076360 004737 017512 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
9527 076364 010337 101566 MOV R3,T27SZ ;SET UP RECORD SIZE IN PACKET
9528 076370 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
9529 076374 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
9530 076400 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9531 076404 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9532 076410 020102 CMP R1,R2 ;ARE THEY EQUAL
9533 076412 001406 BEQ 80$ ;BR, IF OK
9534 076414 005237 002212 INC FATFLG ;BUMP COUNT
9538 076420 ERRHRD ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA

```

TEST 7: WRITE DATA RETRY

```

076420 104456                                TRAP  C$ERRRD
076422 001321                                .WORD 721
076424 005111                                .WORD WRTERR
076426 012136                                .WORD PKTSSR
9539 076430 80$: CKLOOP                       ;LOOP IF SELECTED
076430 104406                                TRAP  C$CLP1
9540
9541 ;*****
9542 ;
9543 ;WRITE DATA RETRY,CVC=1,ACK COMMAND
9544 ;
9545 ;*****
9546
9547 076432 012737 141005 101560             MOV    #141005,T27PK3           ;WRITE DATA RETRY,CVC=1,ACK COMMAND
9548 076440 010465 000000                    MOV    R4,TSDB(R5)             ;ISSUE COMMAND
9549 076444 004737 016340                    JSR    PC,WAITF                ;WAIT FOR SSR TO SET
9550 076450 016501 000002                    MOV    TSSR(R5),R1            ;GET TSSR CONTENTS
9551 076454 012702 000200                    MOV    #SSR,R2                ;SET UP EXPECTED
9552 076460 020102                            CMP    R1,R2                  ;ARE THEY EQUAL
9553 076462 001406                            BEQ    90$                    ;BR, IF OK
9554 076464 005237 002212                    INC    FATFLG                 ;BUMP COUNT
9555 076470 ERRHRD ERRNO,T27WRF,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA RETRY
076470 104456                                TRAP  C$ERRRD
076472 001322                                .WORD 722
076474 104366                                .WORD T27WRF
076476 012136                                .WORD PKTSSR
9559 076500 90$: CKLOOP                       ;LOOP IF SELECTED
076500 104406                                TRAP  C$CLP1
9560 076502 005723                            TST   (R3)+                   ;BUMP RECORD SIZE COUNTER
9561 076504 020327 000050                    CMP    R3,#40.               ;AT 40 SIZE YET
9562 076510 001315                            BNE   65$                    ;BR, IF MORE RECORDS TO WRITE
9563
9564 ;*****
9565 ;
9566 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9567 ;
9568 ;*****
9569
9570 076512 004737 011104                    JSR    PC,REWIND              ;CALL TAPE REWIND COMMAND
9571 076516 103407                            BCS   230$                   ;BR, IF NO PROBLEM
9572 076520 010001                            MOV    R0,R1                  ;SAVE TSSR
9573 076522 005237 002212                    INC    FATFLG                 ;BUMP COUNT
9577 076526 ERRHRD ERRNO,T27RWN,EXPREC      ;REWIND NOT ACCEPTED
076526 104456                                TRAP  C$ERRRD
076530 001323                                .WORD 723
076532 102765                                .WORD T27RWN
076534 015564                                .WORD EXPREC
9578 076536 230$: CKLOOP                     ;LOOP IF SELECTED
076536 104406                                TRAP  C$CLP1
9579
9580 ;*****
9581 ;
9582 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9583 ;
9584 ;*****
9585
9586 076540 013701 101470                    MOV    T27BFR+6,R1           ;PICK UP XSTO

```

TEST 7: WRITE DATA RETRI

```

9587 076544 010102          MOV      R1,R2          ;SET UP EXPECTED
9588 076546 052702 000002  BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
9589 076552 020102          CMP      R1,R2          ;DOES EXP - REC'D
9590 076554 001406          BEQ     240$           ;BR, IF EQUAL (OK)
9591 076556 005237 002212  INC     FATFLG         ;BUMP COUNT
9595 076562          ERRHRD  ERRNO,T27BOT,FXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    724
                                .WORD    T27BOT
                                .WORD    EXPREC
9596 076572          240$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
9597 076574 012703 000024  MOV     #20.,R3        ;STARTING RECORD SIZE
9598 076600 013737 003114 101562  MOV     FREE,T27RB     ;STARTING READ BUFFER ADDRESS
9599
9600          ;*****
9601          ;
9602          ;READ DATA,ACK COMMAND
9603          ;
9604          ;*****
9605
9606 076606 012737 100001 101560 265$:  MOV     #100001,T27PK3 ;READ DATA,ACK COMMAND
9607 076614 012704 101560          MOV     #T27PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
9608 076620 010337 101566          MOV     R3,T27SZ      ;SET UP RECORD SIZE IN PACKET
9609 076624 010465 000000          MOV     R4,TSDB(R5)   ;ISSUE COMMAND
9610 076630 004737 016340          JSR    PC,WAITF       ;WAIT FOR SSR TO SET
9611 076634 016501 000002          MOV     TSSR(R5),R1   ;GET TSSR CONTENTS
9612 076640 012702 000200          MOV     #SSR,R2       ;SET UP EXPECTED
9613 076644 020102          CMP     R1,R2         ;ARE THEY EQUAL
9614 076646 001406          BEQ     280$           ;BR, IF OK
9615 076650 005237 002212  INC     FATFLG         ;BUMP COUNT
9619 076654          ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    725
                                .WORD    RDERR
                                .WORD    PKTSSR
9620 076664          280$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
9621 076666 013702 003114          MOV     FREE,R2       ;GET BUFFER ADDRESS
9622 076672 010304          MOV     R3,R4         ;GET RECORD SIZE
9623 076674 162704 000024          SUB     #20.,R4       ;POINT BACK TO 1ST RECORD
9624 076700 060204          285$:  ADD     R2,R4       ;POINT TO 1ST LOC IN BUFFER
9625 076702 021403          CMP     (R4),R3       ;DATA WRITTEN = READ
9626 076704 001410          BEQ     290$           ;BR, IF DATA OK (GOOD)
9627 076706 011401          MOV     (R4),R1       ;PICK UP BAD DATA
9628 076710 010302          MOV     R3,R2         ;SET UP EXPECTED
9629 076712 005237 002212  INC     FATFLG         ;BUMP COUNT
9633 076716          ERRHRD  ERRNO,T27DTA,EXPREC ;DATA IN BUFFER NOT CORRECT
                                TRAP      C$ERHRD
                                .WORD    726
                                .WORD    T27DTA
                                .WORD    EXPREC
9634 076726          290$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
9635 076730          TST     (R4),         ;BUMP TO NEXT ADDRESS
9636 076732 160204          SUB     R2,R4         ;BACK TO RECORD SIZE
9637 076734 020403          CMP     R4,R3         ;AT END OF RECORD YET

```


TEST 7: WRITE DATA RETRY

```

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

```


TEST 7: WRITE DATA RETRY

```

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

```

TEST 7: WRITE DATA RETRY

```

077424 104456
077426 001335
077430 102765
077432 012136
9803 077434 230$: CKLOOP ;LOOP IF SELECTED
077434 104406 TRAP C$ERHRD
9804 TRAP C$CLP1
9805 ;*****
9806 ;
9807 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9808 ;
9809 ;*****
9810
9811 077436 013701 101470 MOV T27BFR+6,R1 ;PICK UP XSTO
9812 077442 010102 MOV R1,R2 ;SET UP EXPECTED
9813 077444 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
9814 077450 020102 CMP R1,R2 ;DOES EXP = REC 0
9815 077452 001406 BEQ 240$ ;BR, IF EQUAL (OK)
9816 077454 005237 002212 INC FATFLG ;BUMP COUNT
9820 077460 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
077460 104456 TRAP C$ERHRD
077462 001336 .WORD 734
077464 102461 .WORD T27BOT
077466 015564 .WORD EXPREC
9821 077470 240$: CKLOOP ;LOOP IF SELECTED
077470 104406 TRAP C$CLP1
9822 077472 012703 000024 MOV #20.,R3 ;STARTING RECORD SIZE
9823 077476 013737 003114 101562 MOV FREE,T27RB ;STARTING READ BUFFER ADDRESS
9824
9825 ;*****
9826 ;
9827 ;READ DATA,ACK COMMAND
9828 ;
9829 ;*****
9830
9831 077504 012737 100001 101560 265$: MOV #100001,T27PK3 ;READ DATA,ACK COMMAND
9832 077512 012704 101560 MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
9833 077516 010337 101566 MOV R3,T27SZ ;SET UP RECORD SIZE IN PACKET
9834 077522 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
9835 077526 004737 016340 JSR PC,WAITF ;WAIT FOR SSR TO SET
9836 077532 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9837 077536 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9838 077542 020102 CMP R1,R2 ;ARE THEY EQUAL
9839 077544 001406 BEQ 280$ ;BR, IF OK
9840 077546 005237 002212 INC FATFLG ;BUMP COUNT
9844 077552 ERRHRD ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
077552 104456 TRAP C$ERHRD
077554 001337 .WORD 735
077556 005204 .WORD RDERR
077560 012136 .WORD PKTSSR
9845 077562 280$: CKLOOP ;LOOP IF SELECTED
077562 104406 TRAP C$CLP1
9846 077564 013702 003114 MOV FREE,R2 ;GET BUFFER ADDRESS
9847 077570 010304 MOV R3,R4 ;GET RECORD SIZE
9848 077572 162704 000024 SUB #20.,R4 ;POINT BACK TO 1ST RECORD
9849 077576 060204 285$: ADD R2,R4 ;POINT TO 1ST LOC IN BUFFER
9850 077600 000303 SWAB R3 ;SWAP BYTES SWB=1 ETC.

```


TEST 7: WRITE DATA RETRY

```

077744 000000                                .WORD 0
077746 013727 002116                        MOV     L$DLY,'PC)'
077752 000000                                .WORD 0
077754 005367 177772                        DEC     -6(PC)
077760 001375                                BNE     . 4
077762 005367 177756                        DEC     22(PC)
077766 001367                                BNE     . 20
9930 077770 005337 101612                    DEC     T27DLY ;BUMP COUNTER
9931 077774 001356                            BNE     10$ ;BR, IF COUNTER NOT DONE
9932 077776 005237 002212                    INC     FATFLG ;BUMP COUNT
9936 100002 010001                            MOV     R0,R1 ;CONTENTS OF TSSR REGISTER
9937 100004                                ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
100004 104455                                TRAP   C$ERDF
100006 001341                                .WORD 737
100010 003650                                .WORD SFIERR
100012 012124                                .WORD SFIMSG
9938 100014 013737 002172 101460 20$:      MOV     UNITN,T27DSW ;SET UP UNIT NUMBER
9939
9940 100022 012704 101440                    MOV     @T27PACKET,04 ;SUBROUTINE NEEDS PACKET ADDRESS
9941
9942 ;*****
9943 ;
9944 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
9945 ;
9946 ;*****
9947
9948 100026 004737 010752                    JSR     PC,WRTPHR ;ISSUE WRITE CHARACTERISTICS
9949 100032 103407                            BCS     23$ ;BR, IF COMMAND ISSUED OK
9950 100034 005237 002212                    INC     FATFLG ;BUMP COUNT
9954 100040 010001                            MOV     R0,R1 ;SAVE CONTENTS OF TSSR
9955 100042                                ERRHRD  ERRNO,WRTPHR,SFIMSG ;WRITE CHARACTERISTICS FAILED
100042 104456                                TRAP   C$ERRRD
100044 001342                                .WORD 738
100046 005054                                .WORD WRTPHR
100050 012124                                .WORD SFIMSG
9956 100052                                23$:   CKLOOP ;LOOP IF SELECTED
100052 104406                                TRAP   C$CLP1
9957
9958 ;*****
9959 ;
9960 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9961 ;
9962 ;*****
9963
9964 100054 004737 011104                    JSR     PC,REWIND ;CALL TAPE REWIND COMMAND
9965 100060 103411                            BCS     30$ ;BR, IF NO PROBLEM
9966 100062 016501 000002                    MOV     TSSR(R5),R1 ;GET TSSR CONTENTS
9967 100066 010004                            MOV     R0,R4 ;GET PACKET ADDRESS
9968 100070 005237 002212                    INC     FATFLG ;BUMP COUNT
9972 100074                                ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
100074 104456                                TRAP   C$ERRRD
100076 001343                                .WORD 739
100100 102765                                .WORD T27RWN
100102 012136                                .WORD PKTSSR
9973 100104                                30$:   CKLOOP ;LOOP IF SELECTED
100104 104406                                TRAP   C$CLP1
9974

```

TEST 7: WRITE DATA RETRI

```

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

```

TEST 7: WRITE DATA RETRY

```

10028 100254 010004          MOV      R0,R4          ;GET PACKET ADDRESS
10029 100256 005237 002212  INC      FATFLG        ;BUMP COUNT
10033 100262          ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
          100262 104456          TRAP      C$ERHRD
          100264 001346          .WORD    742
          100266 102765          .WORD    T27RWN
          100270 012136          .WORD    PKTSSR
10034 100272          130$:  CKLOOP          ;LOOP IF SELECTED
          100272 104406          TRAP      C$CLP1
10035
10036          ;*****
10037          ;
10038          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
10039          ;
10040          ;*****
10041
10042 100274 013701 101470          MOV      T27BFR+6,R1    ;PICK UP XSTO
10043 100300 010102          MOV      R1,R2          ;SET UP EXPECTED
10044 100302 052702 000002          BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
10045 100306 020102          CMP      R1,R2          ;DOES EXP = REC'D
10046 100310 001406          BEQ      140$          ;BR, IF EQUAL (OK)
10047 100312 005237 002212  INC      FATFLG        ;BUMP COUNT
10051 100316          ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          100316 104456          TRAP      C$ERHRD
          100320 001347          .WORD    743
          100322 102461          .WORD    T27BOT
          100324 015564          .WORD    EXPREC
10052 100326          140$:  CKLOOP          ;LOOP IF SELECTED
          100326 104406          TRAP      C$CLP1
10053 100330 012704 101560          MOV      #T27PK3,R4    ;SET UP PACKET ADDRESS
10054 100334 012737 000010 101562  MOV      #10,T27RB     ;SET UP RECORDS TO SPACE OVER
10055
10056          ;*****
10057          ;
10058          ;ACK,CVC=1,SPACE FORWARD COMMAND
10059          ;
10060          ;*****
10061
10062 100342 012737 140010 101560          MOV      #140010,T27PK3 ;ACK,CVC=1,SPACE FORWARD COMMAND
10063 100350 010465 000000          150$:  MOV      R4,TSDB(R5) ;ISSUE COMMAND
10064 100354 005237 101606          152$:  INC      T27CNT    ;BUMP TIMER
10065 100360          DELAY    1            ;DELAY ABOUT 100US
          100360 012727 000001          MOV      #1,(PC)+
          100364 000000          .WORD    0
          100366 013727 002116          MOV      L$DLY,(PC)+
          100372 000000          .WORD    0
          100374 005367 177772          DEC      6(PC)
          100400 001375          BNE      . 4
          100402 005367 177756          DEC      -22(PC)
          100406 001367          BNE      . 20
10066 100410 016501 000002          MOV      TSSR(R5),R1   ;GET TSSR
10067 100414 032701 000200          BIT      #BIT7,R1     ;CHECK FOR TSSR'S SSR SET
10068 100420 001755          BEQ      152$          ;KEEP COUNTING UNTIL SET
10069 100422 016501 000002          MOV      TSSR(R5),R1   ;GET STATUS FROM TSSR
10070 100426 012702 000200          MOV      #SSR,R2      ;SET UP EXPECTED
10071 100432 020201          CMP      R2,R1         ;WAS EVERYTHING OK
10072 100434 001406          BEQ      160$          ;BR, IF ALL IS WELL

```

TEST 7: WRITE DATA RETRI

```

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

```


TEST 7: WRITE DATA RETRY

```

10124 100562 012704 101560      MOV      #T27PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
10125 100566 012737 000024 101566  MOV      #20.,T27SZ      ;SET UP RECORD SIZE IN PACKET
10126 100574 010465 000000      MOV      R4,TSD8(R5)    ;ISSUE COMMAND
10127 100600 004737 016340      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
10128 100604 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
10129 100610 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED
10130 100614 020102      CMP      R1,R2         ;ARE THEY EQUAL
10131 100616 001406      BEQ      180$          ;BR, IF OK
10132 100620 005237 002212      INC      FATFLG        ;BUMP COUNT
10136 100624      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      100624 104456      TRAP    C$ERHRD
      100626 001353      .WORD  747
      100630 005111      .WORD  WRTErr
      100632 012136      .WORD  PKTSSR
10137 100634      180$:  CKLOOP          ;LOOP IF SELECTED
      100634 104406      TRAP    C$CLP1
10138 100636 005303      DEC      R3            ;COUNT NUMBER OF RECORDS
10139 100640 001342      BNE     177$          ;BR, IF MORE RECORDS TO WRITE
10140
10141      ;*****
10142      ;
10143      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10144      ;
10145      ;*****
10146
10147 100642 004737 011104      JSR      PC,REWIND      ;ISSUE REWIND
10148 100646 103411      BCS     182$          ;BR, IF ALL IS WELL
10149 100650 010004      MOV     R0,R4         ;GET PACKET ADDRESS
10150 100652 016501 000002      MOV     TSSR(R5),R1    ;GET TSSR CONTENTS
10151 100656 005237 002212      INC     FATFLG        ;BUMP COUNT
10155 100662      ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND FAILED
      100662 104456      TRAP    C$ERHRD
      100664 001354      .WORD  748
      100666 102765      .WORD  T27RWN
      100670 012136      .WORD  PKTSSR
10156 100672      182$:  CKLOOP          ;SELECT LOOP MAYBE
      100672 104406      TRAP    C$CLP1
10157
10158      ;*****
10159      ;
10160      ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
10161      ;BIT 15 SETS DIRECTION  0=FORWARD  1=REVERSE
10162      ;
10163      ;*****
10164
10165 100674 012703 000001      MOV     #1.,R3        ;SPACE 1 RECORD FORWARD
10166 100700 004737 010556      JSR     PC,SPACE      ;ISSUE SPACE COMMAND
10167 100704 103411      BCS     185$          ;BR, IF COMMAND OK
10168 100706 010004      MOV     R0,R4         ;GET PACKET ADDRESS
10169 100710 016501 000002      MOV     TSSR(R5),R1    ;GET TSSR STATUS
10170 100714 005237 002212      INC     FATFLG        ;BUMP COUNT
10174 100720      ERRHRD  ERRNO,T27SCF,PKTSSR ;SPACE FORWARD COMMAND FAILED
      100720 104456      TRAP    C$ERHRD
      100722 001355      .WORD  749
      100724 104227      .WORD  T27SCF
      100726 012136      .WORD  PKTSSR
10175 100730      185$:  CKLOOP          ;LOOP IF SELECTED

```

TEST 7: WRITE DATA RETRY

```

10176 100730 104406                                TRAP C$CLP1
10176 100732 012703 000144                        MOV #100.,R3 ;NUMBER OF RECORDS TO BE WRITTEN
10177 100736 013737 003114 101562                MOV FREE,T27WB ;STARTING WRITE BUFFER ADDRESS
10178
10179 ;*****
10180 ;
10181 ;WRITE DATA RETRY,ACK COMMAND
10182 ;
10183 ;*****
10184
10185 100744 012737 101005 101560 190$: MOV #101005,T27PK3 ;WRITE DATA RETRY,ACK COMMAND
10186 100752 012704 101560                    MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
10187 100756 012737 000024 101566                MOV #20.,T27SZ ;SET UP RECORD SIZE IN PACKET
10188 100764 010465 000000                    MOV R4,TSDB(R5) ;ISSUE COMMAND
10189 100770 004737 016340                    JSR PC,WAITF ;WAIT FOR SSR TO SET
10190 100774 016501 000002                    MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10191 101000 012702 000200                    MOV #SSR,R2 ;SET UP EXPECTED
10192 101004 020102                            CMP R1,R2 ;ARE THEY EQUAL
10193 101006 001406                            BEQ 200$ ;BR, IF OK
10194 101010 005237 002212                    INC FATFLG ;BUMP COUNT
10198 101014                                ERRHRD ERRNO,T27WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
10198 101014 104456                                TRAP C$ERHRD
10198 101016 001356                                .WORD 750
10198 101020 103321                                .WORD T27WDC
10198 101022 012136                                .WORD PKTSSR
10199 101024 200$: CKLOOP ;LOOP IF SELECTED
10199 101024 104406                                TRAP C$CLP1
10200 101026 013737 003114 101562                MOV FREE,T27WB ;STARTING WRITE BUFFER ADDRESS
10201
10202 ;*****
10203 ;
10204 ;WRITE DATA,CVC=1,ACK COMMAND
10205 ;
10206 ;*****
10207
10208 101034 012737 140005 101560                    MOV #140005,T27PK3 ;WRITE DATA,CVC=1,ACK COMMAND
10209 101042 012704 101560                    MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
10210 101046 012737 000024 101566                MOV #20.,T27SZ ;SET UP RECORD SIZE IN PACKET
10211 101054 010465 000000                    MOV R4,TSDB(R5) ;ISSUE COMMAND
10212 101060 004737 016340                    JSR PC,WAITF ;WAIT FOR SSR TO SET
10213 101064 016501 000002                    MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10214 101070 012702 000200                    MOV #SSR,R2 ;SET UP EXPECTED
10215 101074 020102                            CMP R1,R2 ;ARE THEY EQUAL
10216 101076 001406                            BEQ 210$ ;BR, IF OK
10217 101100 005237 002212                    INC FATFLG ;BUMP COUNT
10221 101104                                ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
10221 101104 104456                                TRAP C$ERHRD
10221 101106 001357                                .WORD 751
10221 101110 005111                                .WORD WRERR
10221 101112 012136                                .WORD PKTSSR
10222 101114 210$: CKLOOP ;LOOP IF SELECTED
10222 101114 104406                                TRAP C$CLP1
10223 101116 005303                            DEC R3 ;BUMP DOWN RECORD COUNTER
10224 101120 001311                            BNE 190$ ;BR, IF MORE RECORDS TO WRITE RETRY
10225
10226 ;*****
10227 ;

```

TEST 7: WRITE DATA RETRI

```

10228 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10229 ;
10230 ;*****
10231
10232 101122 004737 011104 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
10233 101126 103411 BCS 230$ ;BR, IF NO PROBLEM
10234 101130 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10235 101134 010004 MOV R0,R4 ;GET PACKET ADDRESS
10236 101136 005237 002212 INC FATFLG ;BUMP COUNT
10240 101142 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
101142 104456 TRAP C$ERHRD
101144 001360 .WORD 752
101146 102765 .WORD T27RWN
101150 012136 .WORD PKTSSR
10241 101152 230$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
101152 104406
10242
10243 ;*****
10244 ;
10245 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
10246 ;
10247 ;*****
10248
10249 101154 013701 101470 MOV T27BFR+6,R1 ;PICK UP XST0
10250 101160 010102 MOV R1,R2 ;SET UP EXPECTED
10251 101162 052702 000002 BIS #8BIT1,R2 ;SET BOT BIT IN EXPECTED
10252 101166 020102 CMP R1,R2 ;DOES EXP = REC'D
10253 101170 001406 BEQ 240$ ;BR, IF EQUAL (OK)
10254 101172 005237 002212 INC FATFLG ;BUMP COUNT
10258 101176 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
101176 104456 TRAP C$ERHRD
101200 001361 .WORD 753
101202 102461 .WORD T27BOT
101204 015564 .WORD EXPREC
10259 101206 240$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
101206 104406
10260 101210 012704 101560 MOV #T27PK3,R4 ;SET UP PACKET ADDRESS
10261 101214 012737 000010 101562 MOV #10,T27R8 ;SET UP RECORDS TO SPACE OVER
10262
10263 ;*****
10264 ;
10265 ;ACK,CVC=1,SPACE FORWARD COMMAND
10266 ;
10267 ;*****
10268
10269 101222 012737 140010 101560 MOV #140010,T27PK3 ;ACK,CVC=1,SPACE FORWARD COMMAND
10270 101230 010465 000000 250$: MOV R4,TSD8(R5) ;ISSUE COMMAND
10271 101234 005237 101610 252$: INC T27CNU ;BUMP TIMER
10272 101240 DELAY 1 ;DELAY ABOUT 100US
101240 012727 000001 MOV #1,(PC)+
101244 000000 .WORD 0
101246 013727 002116 MOV L$DLY,(PC)+
101252 000000 .WORD 0
101254 005367 177772 DEC -6(PC)
101260 001375 BNE -4
101262 005367 177756 DEC -22(PC)
101266 001367 BNE -20

```


TEST 7: WRITE DATA RETRY

```

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

;+
;LOCAL STORAGE FOR THIS TEST
;-
T27PACKET:
        .WORD 100004
        .WORD T27DATA
        .WORD 0
        .WORD 10.
T27DATA:
        .WORD T27BFR
        .WORD 0
        .WORD 20.
        .WORD 0
T27DSW: .WORD 0
T27BFR: .BLKW 25.
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
;=<.>10>&177770
T27PK2:
        .WORD 100006
        .WORD T27BF2
        .WORD 0
        .WORD 6.
T27PK3:
        .WORD 100005
T27RB:
T27WB: .WORD FREE
        .WORD 0
T27SZ: .WORD 0
        .EVEN
;
;
;
T27BF2:
T27BS0: .BYTE 10
T27BS1: .BYTE 200
T27S2: .WORD 0
T27S3: .WORD 0
;
;
        .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T27RN: .WORD 100205
T27WDR: .WORD 100605
T27CON: .WORD 102205
        .WORD 177777
;TAPETIMER COUNTER STORAGE AREA
;TAPETIMER COUNTER STORAGE AREA
;DELAY COUNTER
T27CNT: .WORD 0
T27CNU: .WORD 0
T27DLY: .WORD 0
;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH . ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
;SELECT DRIVE 0
;MESSAGE BUFFER
;WRITE SUB SYS MEM COMMAND, AND ACK
;ADDRESS OF SELECT BLOCK DATA
;SIZE OF DATA PACKET
;REREAD COMMAND, AND ACK
;ADDRESS OF WRITE BUFFER
;SIZE OF BUFFER (EXTENT)
;BSEL0 AREA
;BSEL1 AREA
;SEL 2 AREA
;DATA AREA
;REREAD DATA (NEXT)
;REREAD DATA RETRY
;WRITE CONTINOUS
;END OF DATA
;TAPE TIMER COUNTER STORAGE AREA
;TAPE TIMER COUNTER STORAGE AREA
;DELAY COUNTER

```

TEST 7: WRITE DATA RETRY

```

10379
10380
10381          ;*
10382          ;LOCAL TEXT MESSAGES FOR TEST
10383          ;-
10384
10385 101614      124      141      160  T27WNG: .ASCIZ  'Tape Position Incorrect After REREAD Previous (OPP=1)'
10386 101702      124      123      123  T27RDF: .ASCIZ  'TSSR Incorrect After READ DATA Command'
10387 101751      122      105      122  T27RRF: .ASCIZ  'REREAD Previous (Space Reverse, Read Forward) Command Failed'
10388 102046      120      117      123  T27SC:  .ASCIZ  'POSITION (Space Command) Failed, TSSR Not Correct'
10389 102130      122      111      102  T27LOR: .ASCIZ  'RIB NOT SET AFTER READ REVERSE INTO BOT'
10390 102200      124      123      123  T27WDF: .ASCIZ  'TSSR Not Correct After Illegal Mode Bits Set'
10391 102255      111      154      154  T27LOQ: .ASCIZ  'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
10392 102336      122      105      122  T27SSR: .ASCIZ  'REREAD COMMAND Not Accepted'
10393 102372      124      123      123  T27WDE: .ASCIZ  'TSSR Not Correct After WRITE DATA RETRY Command, At BOT'
10394 102461      124      141      160  T27BOT: .ASCIZ  'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
10395 102554      127      122      111  T27TIM: .ASCIZ  'WRITE DATA RETRY'S Erase Tape Not Long Enough'
10396 102631      122      105      122  T27EOT: .ASCIZ  'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
10397 102710      124      123      123  T27TM:  .ASCIZ  'TSSR Not Correct After REREAD COMMAND Reject'
10398 102765      122      145      167  T27RWN: .ASCIZ  'Rewind (POSITION) Command Not Accepted'
10399 103034      122      101      115  T27RNC: .ASCIZ  'RAM Error, Correct Data Pattern Not In Ram'
10400 103107      124      123      123  T27AM3: .ASCIZ  'TSSR Init. Failed After REREAD COMMAND'
10401 103156      104      162      151  T27OFL: .ASCIZ  'Drive 7 Select Failed To Set "OFL" In TSSR'
10402 103231      124      123      123  T27WDD: .ASCIZ  'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
10403 103321      124      123      123  T27WDC: .ASCIZ  'TSSR Not Correct After REREAD DATA Command'
10404 103374      103      126      103  T27VCK: .ASCIZ  'CVC Set, Didn't Reset VCK In Message Buffer'
10405 103447      124      123      102  T27BA:  .ASCIZ  'TSBA Not Correct After REREAD DATA Command'
10406 103522      127      122      111  T27WSS: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
10407 103611      122      145      141  T27LON: .ASCIZ  'Reading Long Record Failed To Set RLL Bit In XST0'
10408 103673      122      145      141  T27LOP: .ASCIZ  'Reading Long Record Failed To Set RLS Bit In XST0'
10409 103755      122      145      163  T27PBP: .ASCIZ  'Residual Byte Count Incorrect After Short Record Read'
10410 104043      122      145      141  T27TRL: .ASCIZ  'Reading Long Record Failed To Give Tape Status Alert'
10411 104131      127      122      111  T27NEF: .ASCIZ  'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
10412 104227      124      123      123  T27SCF: .ASCIZ  'TSSR Not Correct After SPACE RECORDS Command'
10413 104304      124      123      123  T27TSA: .ASCIZ  'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
10414 104366      124      123      123  T27WRF: .ASCIZ  'TSSR Not Correct After WRITE DATA RETRY Command'
10415 104446      104      141      164  T27DTA: .ASCIZ  'Data Compare Error, Data Read From Tape Not Equal To Written'
10416 104543      127      162      151  TST27ID: .ASCIZ  'Write Data Retry'
10417
10418          .EVEN
10419
10420          ;*
10421          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
10422          ;WRITE SUBSYSTEM MEMORY COMMAND
10423          ;
10424          ;-
10425
10425 104564
10426 104564
10427 104570      012701  101440      SAVREG
10428 104574      012721  100004      MOV      #T27PACKET,R1      ;SAVE THE REGISTERS
10429 104600      012721  101450      MOV      #100004,(R1)+     ;START OF THE PACKET
10430 104604      005021                      MOV      #T27DATA,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK,
10431 104606      012721  000012      CLR      (R1)+             ;ADDRESS OF CHARAISTICS DATA BLOCK
10432 104612      012721  101462      MOV      #10.,(R1)+       ;EXTENDED ADDRESS
10433 104616      005021                      MOV      #T27BFR,(R1)+   ;SIZE OF DATA BLOCK IN BYTES
10434 104620      012721  000024      CLR      (R1)+             ;ADDRESS OF MESSAGE BUFFER
10435 104624      005021                      MOV      #20.,(R1)+     ;LENGTH OF MESSAGE BUFFER
10435 104624      005021                      CLR      (R1)+

```

TEST 7: WRITE DATA RETRY

```

10436 104626 012711 000000      MOV    #0,(R1)                ;SELECT DRIVE ZERO
10437 104632 012702 000030      MOV    #24.,R2                ;NUMBER OF LOCATIONS TO BE CLEARED
10438 104636 012762 177777 101462 64$: MOV    #177777,T27BFR(R2)    ;ALL ONES TO MESSAGE BUFFER
10439 104644 005742                TST    (R2)                    ;NEXT LOCATION
10440 104646 022702 000000      CMP    #0,R2                  ;AT END OF LOOP YET
10441 104652 001371                BNE    64$                     ;KEEP GOING UNTIL DONE
10442 104654 000207                RTS    PC                       ;RETURN
10443
10444
10445 104656                T27RT2:
10446 104656                SAVREG
10447 104662 012701 101550      MOV    #T27PK2,R1             ;SAVE THE REGISTERS
10448 104666 012721 100006      MOV    #100006,(R1).          ;START OF THE PACKET
10449 104672 012721 101570      MOV    #T27BF2,(R1).         ;WRITE SUBSYSTEM MEM. WITH ACK.
10450 104676 005021                CLR    (R1).                  ;ADDRESS OF DATA BLOCK
10451 104700 012721 000006      MOV    #6.,(R1).             ;EXTENDED ADDRESS
10452 104704 005021                CLR    (R1).                  ;SIZE OF DATA BLOCK IN BYTES
10453 104706 012701 101570      MOV    #T27BF2,R1             ;POINT TO DATA SEL AREA
10454 104712 005021                CLR    (R1).
10455 104714 005011                CLR    (R1)
10456 104716 000207                RTS    PC                       ;RETURN
10457 104720                T27RT3:
10458 104720                SAVREG
10459 104724 012701 101560      MOV    #T27PK3,R1             ;SAVE REGISTERS
10460 104730 005021                CLR    (R1).                  ;SET UP POINTER ADDRESS
10461 104732 005021                CLR    (R1).                  ;COMMAND SPACE
10462 104734 005021                CLR    (R1).                  ;ADDRESS OF DATA BLOCK
10463 104736 005011                CLR    (R1).                  ;EXTENDED ADDRESS
10464 104740 000207                RTS    PC                       ;SIZE OF DATA TRANSFER BLOCK
10465 104742                ENDTST                          ;RETURN
104742 104401                L10122: TRAP    C$ETST

```


TEST 8: WRITE/READ TAPE MARK

10524	105060	103407			BCS	24:			;BR, IF COMMAND ISSUED OK
10525	105062	005237	002212		INC		FATFLG		;BUMP COUNT
10529	105066	010001			MOV		R0,R1		;SAVE CONTENTS OF TSSR
10530	105070				ERRHRD		ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTIC FAILED
	105070	104456							TRAP C\$ERHRD
	105072	001442							.WORD 802
	105074	005054							.WORD WRTMSG
	105076	012124							.WORD SFIMSG
10531	105100				24:	CKLOOP			
	105100	104406							TRAP C\$CLP1
10532	105102	005737	002216		TST		EXTFEA		;CHECK FOR EXTENDED FEATURES SW SWITCH
10533	105106	001044			BNE		50:		;BR IF SWITCH IS ON
10534									
10535	105110	112737	000200	110261	MOVB		#200,T28BS1		;WRITE MISCELLANEOUS CONT/READ STATUS
10536	105116	112737	000010	110260	MOVB		#10,T28BS0		;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
10537	105124	012704	110240		MOV		#T28PK2,R4		;WRITE SUBSYS MEM PACKET
10538	105130	01C465	000000		MOV		R4,TSDB(R5)		;ISSUE COMMAND
10539	105134	004737	016426		JSR		PC,CHKTSSR		;WAIT FOR SSR
10540	105140	103407			BCS		30:		;BR, IF NO ERROR
10541	105142	010001			MOV		R0,R1		;ERROR, SAVE TSSR
10542	105144	005237	002212		INC		FATFLG		;BUMP COUNT
10546	105150				ERRHRD		ERRNO,T28SSR,PKTSSR		;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
	105150	104456							TRAP C\$ERHRD
	105152	001443							.WORD 803
	105154	110755							.WORD T28SSR
	105156	012136							.WORD PKTSSR
10547	105160				30:	CKLOOP			;LOOP IF SELECTED
	105160	104406							TRAP C\$CLP1
10548	105162	012704	110130		MOV		#T28PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS
10549	105166	012737	000007	110150	MOV		#7,T28DSW		;SELECT DRIVE 7
10550	105174	004737	010752		JSR		PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS
10551	105200	103407			BCS		50:		;BR, IF COMMAND ISSUED OK
10552	105202	005237	002212		INC		FATFLG		;BUMP COUNT
10556	105206	010001			MOV		R0,R1		;SAVE CONTENTS OF TSSR
10557	105210				ERRHRD		ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTIC FAILED
	105210	104456							TRAP C\$ERHRD
	105212	001444							.WORD 804
	105214	005054							.WORD WRTMSG
	105216	012124							.WORD SFIMSG
10558	105220				50:	CKLOOP			;SCOPE LOOP
	105220	104406							TRAP C\$CLP1
10559	105222	016501	000002		MOV		TSSR(R5),R1		;GET TSSR CONTENTS
10560	105226	032701	000100		BIT		#0FL,R1		;CHECK FOR THE OFFLINE BIT SET
10561	105232	001006			BNE		60:		;BR, IF OFFLINE (GOOD)
10562	105234	005237	002212		INC		FATFLG		;BUMP COUNT
10566	105240				ERRDF		ERRNO,T28OFL,SFIMSG		;OFF LINE SHOULD HAVE BEEN SET (BAD)
	105240	104455							TRAP C\$ERDF
	105242	001445							.WORD 805
	105244	111310							.WORD T28OFL
	105246	012124							.WORD SFIMSG
10567	105250				60:	CKLOOP			;LOOP IF SELECTED
	105250	104406							TRAP C\$CLP1
10568	105252	012703	110276		MOV		#T28RN,R3		;POINTER FOR COMMANDS
10569	105256	011337	110250		MOV		(R3),T28PK3		;TAPE READ COMMAND IN PLACE
10570	105262	012704	110250		MOV		#T28PK3,R4		;R4 = POINTER TO PACKET
10571	105266	010465	000000		MOV		R4,TSDB(R5)		;ISSUE COMMAND
10572	105272	004737	016340		JSR		PC,WAITF		;WAIT FOR SSR TO SET

TEST 8: WRITE/READ TAPE MARK

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


TEST 8: WRITE/READ TAPE MARK

10913	106662	004737	017512		JSR	PC,FILLMEM	;FILL MEM WITH ALL ONES		
10914	106666	013737	003114	110252	MOV	FREE,T28WB	;STARTING READ BUFFER ADDRESS		
10915	106674	012737	140401	110250	MOV	#140401,T28PK3	;READ REVERSE,ACK, COMMAND		
10916	106702	012704	110250		MOV	#T28PK3,R4	;SET UP R4 WITH PACKET ADDRESS		
10917	106706	013737	000024	110256	MOV	20.,T28SZ	;SET UP RECORD SIZE IN PACKET		
10918	106714	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND		
10919	106720	004737	016340		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
10920	106724	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
10921	106730	012702	100204		MOV	#SSR!SC!BIT2,R2	;SET UP EXPECTED		
10922	106734	020102			CMP	R1,R2	;ARE THEY EQUAL		
10923	106736	001406			BEQ	200\$;BR, IF OK		
10924	106740	005237	002212		INC	FATFLG	;BUMP COUNT		
10928	106744				ERRHRD	ERRNO,T28RDF,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA		
	106744	104456					TRAP	C\$ERHRD	
	106746	001472					.WORD	826	
	106750	110454					.WORD	T28RDF	
	106752	012136					.WORD	PKTSSR	
10929	106754			200\$:	CKLOOP		;LOOP IF SELECTED		
	106754	104406					TRAP	C\$CLP1	
10930	106756	013701	110160		MOV	T28FR+6,R1	;PICK UP XSTO		
10931	106762	010102			MOV	R1,R2	;SET UP EXPECTED		
10932	106764	052702	100000		BIS	#BIT15,R2	;TMK SHOULD BE SET		
10933	106770	020102			CMP	R1,R2	;IS TMK SET		
10934	106772	001406			BEQ	210\$;BR, IF TMK WAS SET (GOOD)		
10935	106774	005237	002212		INC	FATFLG	;BUMP COUNT		
10939	107000				ERRHRD	ERRNO,T28RRM,EXPREC	;TMK NOT SET AFTER READ REV		
	107000	104456					TRAP	C\$ERHRD	
	107002	001473					.WORD	827	
	107004	111567					.WORD	T28RRM	
	107006	015564					.WORD	EXPREC	
10940	107010			210\$:	CKLOOP		;LOOP IF SELECTED		
	107010	104406					TRAP	C\$CLP1	
10941	107012	017701	074076		MOV	#FREE,R1	;FIRST LOC IN READ BUFFER		
10942	107016	012702	177777		MOV	#177777,R2	;EXPECTED IF NO DATA TRANS.		
10943	107022	020102			CMP	R1,R2	;DID ANY DATA GET TRANSFERRED		
10944	107024	001406			BEQ	220\$;BR, IF NO DATA TRANS (GOOD)		
10945	107026	005237	002212		INC	FATFLG	;BUMP COUNT		
10949	107032				ERRHRD	ERRNO,T28DTR,EXPREC	;DATA TRANSFERRED ON READ TAPE MARK		
	107032	104456					TRAP	C\$ERHRD	
	107034	001474					.WORD	828	
	107036	112002					.WORD	T28DTR	
	107040	015564					.WORD	EXPREC	
10950	107042			220\$:	CKLOOP		;LOOP IF SELECTED		
	107042	104406					TRAP	C\$CLP1	
10951	107044	012737	100410	110250	MOV	#100410,T28PK3	;SPACE REVERSE,ACK, COMMAND		
10952	107052	012737	000001	110252	MOV	#1,T28RB	;NUMBER OF RECORDS TO SPACE BACK		
10953	107060	012704	110250		MOV	#T28PK3,R4	;SET UP R4 WITH PACKET ADDRESS		
10954	107064	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND		
10955	107070	004737	016340		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
10956	107074	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
10957	107100	012702	100204		MOV	#SSR!SC!BIT2,R2	;SET UP EXPECTED		
10958	107104	020102			CMP	R1,R2	;ARE THEY EQUAL		
10959	107106	001406			BEQ	222\$;BR, IF OK		
10960	107110	005237	002212		INC	FATFLG	;BUMP COUNT		
10964	107114				ERRHRD	ERRNO,T28RDG,PKTSSR	;TSSR INCORRECT AFTER SPACE CMD.		
	107114	104456					TRAP	C\$ERHRD	
	107116	001475					.WORD	829	

TEST 8: WRITE/READ TAPE MARK

	107120	110535											
	107122	012136								.WORD	T28RDG		
10965	107124			222\$:	CKLOOP					.WORD	PKTSSR		
	107124	104406											
10966	107126	013701	110160		MOV	T28BFR+6,R1				TRAP	C\$CLP1		
10967	107132	010102			MOV	R1,R2							
10968	107134	052702	100000		BIS	#BIT15,R2							
10969	107140	020102			CMP	R1,R2							
10970	107142	001406			BEQ	226\$							
10971	107144	005237	002212		INC	FATFLG							
10975	107150				ERRHRD	ERRNO,T28RRN,EXPREC							
	107150	104456											
	107152	001476								TRAP	C\$ERHRD		
	107154	111645								.WORD	830		
	107156	015564								.WORD	T28RRN		
10976	107160			226\$:	CKLOOP					.WORD	EXPREC		
	107160	104406											
10977	107162	004737	011104		JSR	PC,REWIND				TRAP	C\$CLP1		
10978	107166	103411			BCS	230\$							
10979	107170	010004			MOV	R0,R4							
10980	107172	016501	000002		MOV	TSSR(R5),R1							
10981	107176	005237	002212		INC	FATFLG							
10985	107202				ERRHRD	ERRNO,T28RWN,PKTSSR							
	107202	104456								TRAP	C\$ERHRD		
	107204	001477								.WORD	831		
	107206	111241								.WORD	T28RWN		
	107210	012136								.WORD	PKTSSR		
10986	107212			230\$:	CKLOOP								
	107212	104406											
10987	107214	013701	110160		MOV	T28BFR+6,R1				TRAP	C\$CLP1		
10988	107220	010102			MOV	R1,R2							
10989	107222	052702	000002		BIS	#BIT1,R2							
10990	107226	020102			CMP	R1,R2							
10991	107230	001406			BEQ	240\$							
10992	107232	005237	002212		INC	FATFLG							
10996	107236				ERRHRD	ERRNO,T28BOT,EXPREC							
	107236	104456								TRAP	C\$ERHRD		
	107240	001500								.WORD	832		
	107242	111117								.WORD	T28BOT		
	107244	015564								.WORD	EXPREC		
10997	107246			240\$:	CKLOOP								
	107246	104406											
10998	107250	012700	177777		MOV	#177777,R0				TRAP	C\$CLP1		
10999	107254	004737	017512		JSR	PC,FILLMEM							
11000	107260	013737	003114	110252	MOV	FREE,T28RB							
11001	107266	012737	100001	110250	MOV	#100001,T28PK3							
11002	107274	012704	110250		MOV	#T28PK3,R4							
11003	107300	013737	000024	110256	MOV	20.,T28SZ							
11004	107306	010465	000000		MOV	R4,T28DB(R5)							
11005	107312	004737	016340		JSR	PC,WAITF							
11006	107316	016501	000002		MOV	TSSR(R5),R1							
11007	107322	012702	100204		MOV	#SSR!SC!BIT2,R2							
11008	107326	020102			CMP	R1,R2							
11009	107330	001406			BEQ	245\$							
11010	107332	005237	002212		INC	FATFLG							
11014	107336				ERRHRD	ERRNO,T28WDE,PKTSSR							
	107336	104456								TRAP	C\$ERHRD		

TEST 8: WRITE/READ TAPE MARK

```

11149
11150
11151
11155 110130
11156 110130 100004
11157 110132 110140
11158 110134 000000
11159 110136 000012
11160 110140
11161 110140 110152
11162 110142 000000
11163 110144 000024
11164 110146 000000
11165 110150 000000
11166 110152
11167
11168
11169
11171 110240
11173 110240
11174 110240 100006
11175 110242 110260
11176 110244 000000
11177 110246 000006
11178
11182 110250
11183 110250 100005
11184 110252
11185 110252 003114
11186 110254 000000
11187 110256 000000
11188
11189
11190
11191
11192 110260
11193 110260 010
11194 110261 200
11195 110262 000000
11196 110264 000000
11197
11198
11199
11200
11201
11202 110266
11203 110266 101411
11204 110270 102011
11205 110272 103411
11206 110274 177777
11207 110276 100011
11208 110300 100411
11209 110302 101011
11210 110304 177777
11211
11212
11213 110306 000000

```

```

;
;LOCAL STORAGE FOR THIS TEST
;
T28PACKET:
;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
;SELECT DRIVE 0
;MESSAGE BUFFER
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
;.<.>10>E177770
T28PK2:
;WRITE SUB SYS MEM COMMAND, IE AND ACK
;ADDRESS OF SELECT BLOCK DATA
;SIZE OF DATA PACKET
;
T28PK3:
;REREAD COMMAND, AND ACK
;ADDRESS OF WRITE BUFFER
;SIZE OF BUFFER (EXTENT)
;
;
;
T28BF2:
T28BS0: .BYTE 10 ;BSELO AREA
T28BS1: .BYTE 200 ;BSEL1 AREA
T28S2: .WORD 0 ;SEL 2 AREA
T28S3: .WORD 0 ;DATA AREA
;
;
;EVEN
;TAPE MOTION PACKET COMMAND VALUES
;
T28IMV:
;ILLEGAL MODE BITS TEST DATA
;
;WRITE TAPE MARK COMMAND
;ERASE COMMAND
;WRITE TAPE MARK RETRY
;END OF DATA
;
;
T28CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA

```

TEST 8: WRITE/READ TAPE MARK

11214 110310 000000
11215 110312 000000
11216
11217

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

;TAPE TIMER COUNTER STORAGE AREA
;DELAY COUNTER

TEST 8: WRITE/READ TAPE MARK

```

11219
11220
11221          ;*
11222          ;LOCAL TEXT MESSAGES FOR TEST
11223          ;-
11224
11225 110314      124      141      160  T28RIB: .ASCIZ   'Tape Position Not Correct, RIB Should Be Set'
11226 110371      122      145      163  T28PBP: .ASCIZ   'Residual Byte Counter Register (RBPCR) Not Correct'
11227 110454      124      123      123  T28RDF: .ASCIZ   'TSSR Incorrect After READ REVERSE Into TAPE MARK'
11228 110535      124      123      123  T28RDG: .ASCIZ   'TSSR Incorrect After SPACE Command Into TAPE MARK'
11229 110617      124      123      123  T28WDF: .ASCIZ   'TSSR Not Correct After Illegal Mode Bits Set'
11230 110674      111      154      154  T28LQG: .ASCIZ   'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
11231 110755      127      122      111  T28SSR: .ASCIZ   'WRITE MISCELLANEOUS Command Not Accepted'
11232 111026      124      123      123  T28WDE: .ASCIZ   'TSSR Not Correct After READ DATA Command, Into TAPE MARK'
11233 111117      124      141      160  T28BOT: .ASCIZ   'Tape Not At BOT After REWIND Command'
11234 111164      124      123      123  T28TM:  .ASCIZ   'TSSR Not Correct After FORMAT Command Reject'
11235 111241      122      145      167  T28RWN: .ASCIZ   'Rewind (POSITION) Command Not Accepted'
11236 111310      104      162      151  T28OFL: .ASCIZ   'Drive 7 Select Failed To Set "OFL" In TSSR'
11237 111363      124      123      123  T28WDC: .ASCIZ   'TSSR Not Correct After WRITE TAPE MARK Command'
11238 111442      103      126      103  T28VCK: .ASCIZ   'CVC Set, Didn't Reset VCK In Message Buffer'
11239 111515      124      115      113  T28TMK: .ASCIZ   'TMK Not Set After WRITE TAPE MARK Command'
11240 111567      124      115      113  T28RRM: .ASCIZ   'TMK Not Set After READ REVERSE Into TAPE MARK'
11241 111645      124      115      113  T28RRN: .ASCIZ   'TMK Not Set After SPACE REVERSE Into TAPE MARK'
11242 111724      124      115      113  T28RRP: .ASCIZ   'TMK Not Set After READ FORWARD Into TAPE MARK'
11243 112002      104      141      164  T28DTR: .ASCIZ   'Data Transferred On READ REVERSE Into A TAPE MARK'
11244 112064      104      141      164  T28DTA: .ASCIZ   'Data Compare Error, Data Read From Tape Not Equal To Written'
11245 112161      127      162      151  TST28ID: .ASCIZ  'Write/Read Tape Mark'
11246
11247          ;*
11248          ;
11249          ;ROUTINE TO RESTORE COMMAND PACKET TO START UP (DEFAULT) VALUES
11250          ;WRITE SUBSYSTEM MEMORY COMMAND
11251          ;
11252          ;-
11253
11254 112206
11255 112206
11256 112212      012701  110130
11257 112216      012721  100004
11258 112222      012721  110140
11259 112226      005021
11260 112230      012721  000012
11261 112234      012721  110152
11262 112240      005021
11263 112242      012721  000024
11264 112246      005021
11265 112250      012711  000000
11266 112254      012702  000030
11267 112260      012762  177777  110152  64$:
11268 112266      005742
11269 112270      020227  000000
11270 112274      001371
11271 112276      000207
11272
11273
11274 112300
11275 112300

          T28REST:
          SAVREG
          MOV     #T28PACKET,R1          ;SAVE THE REGISTERS
          MOV     #100004,(R1)+         ;START OF THE PACKET
          MOV     #T28DATA,(R1)+       ;WRITE SUBSYSTEM MEM. WITH ACK.
          CLR     (R1)+                 ;ADDRESS OF CHARAISTICS DATA BLOCK
          MOV     #10.,(R1)+            ;EXTENDED ADDRESS
          MOV     #T28BFR,(R1)+        ;SIZE OF DATA BLOCK IN BYTES
          CLR     (R1)+                 ;ADDRESS OF MESSAGE BUFFER
          MOV     #20.,(R1)+            ;LENGTH OF MESSAGE BUFFER
          CLR     (R1)+
          MOV     #0,(R1)               ;SELECT DRIVE ZERO
          MOV     #24.,R2               ;NUMBER OF LOCATIONS TO BE CLEARED
          MOV     #177777,T28BFR(R2)   ;ALL ONES TO MESSAGE BUFFER
          TST     -(R2)                 ;NEXT LOCATION
          CMP     R2,#0                 ;CHECK FOR END
          BNE     64$                   ;KEEP GOING UNTIL DONE
          RTS     PC                     ;RETURN

          T28RT2:
          SAVREG
          ;SAVE THE REGISTERS

```

TEST 8: WRITE/READ TAPE MARK

```

11276 112304 012701 110240
11277 112310 012721 100006
11278 112314 012721 110260
11279 112320 005021
11280 112322 012721 000006
11281 112326 005021
11282 112330 012701 110260
11283 112334 005021
11284 112336 005011
11285 112340 000207
11286 112342
11287 112342
11288 112346 012701 110250
11289 112352 005021
11290 112354 005021
11291 112356 005021
11292 112360 005011
11293 112362 000207
11294 112364
      112364
      112364 104401
11295 112366
    
```

T28RT3:

```

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

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

ENDTST
ENDMOD
    
```

L10130: TRAP C\$ETS*

TEST 8: WRITE/READ TAPE MARK

```

1          .TITLE  TSV6 - PARAMETER CODING
.
12
18
19 112366          BGNMOD  TSV6
   112366          TSV6::
20
21          .SBTTL  HARDWARE PARAMETER CODING SECTION
22
23          ;**
24          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
25          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
26          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
27          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
28          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
29          ; WITH THE OPERATOR.
30          ;--
31 112366          BGNHRD
   112366          .WORD  L10134 L$HARD/2
   112370          L$HARD::
32
33 112370          GPRMA  HPM1,0,0,160010,177776,YES          ;GET TSBA/TSDB REGISTER ADDRESS.
   112370          .WORD  T$CODE
   112372          .WORD  HPM1
   112374          .WORD  T$LLOLIM
   112376          .WORD  T$HILIM
34 112400          GPRMA  HPM2,2,0,0,776,YES                ;GET VECTOR ADDRESS.
   112400          .WORD  T$CODE
   112402          .WORD  HPM2
   112404          .WORD  T$LLOLIM
   112406          .WORD  T$HILIM
35          ;GPRMD  HPM3,4,0,340,0,7,YES                ;GET INTERRUPT PRIORITY.
36 112410          ENDHRD
   112410          .EVEN
   112410          L10134:
37 112410          104    105    126  HPM1:  .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
38 112444          111    116    124  HPM2:  .ASCIZ  'INTERRUPT VECTOR '
39 112470          111    116    124  HPM3:  .ASCIZ  'INTERRUPT PRIORITY '
40          .EVEN

```


K I L

SYMBOL TABLE

ADDSR	012216	G	C\$AU	=	000052	DEVDR0	023514	FREE	003114	G	INCERK	017134					
ADR	=	000020	G	C\$AUTO	=	000061	DEVNRD	023433	FREEHI	003120	INTCPC	016240					
AMBTSS	006725	C\$BRK	=	000022	DEVNXR	023351	FRESIZ	003116	G	INTFLA	016235						
ASSEMB	=	000010	C\$BSEG	=	000004	DEVONL	023301	FUSI	004115	INTMAS	016234						
A1716	=	000003	C\$BSUB	=	000002	DEVSUM	023244	F\$AU	=	000015	INTR	016306	G				
BADDAT	003146	G	C\$CEFG	=	000045	DFPTBL	002146	G	F\$AUTO	=	000020	INTREC	002214	G			
BADSSR	015770	G	C\$CLCK	=	000062	DIAGMC	=	000000	F\$BGN	=	000040	INTVEC	016236				
BDVPCR	=	177520	G	C\$CLEA	=	000012	DICEC	=	000001	F\$CLEA	=	000007	INTX	004276			
BENBSW	002220	G	C\$CLOS	=	000035	DSBINT	016274	F\$DU	=	000016	INVERT	021276	G				
BIE	=	040000	C\$CLP1	=	000006	DUAD12	004641	F\$END	=	000041	IOKCKI	=	000200				
BIT0	=	000001	G	C\$CVEC	=	000036	DUFLG	003102	G	F\$HARD	=	000004	IOKSTP	=	000001		
BIT00	=	000001	G	C\$DCLN	=	000044	DUMMY	003052	F\$HW	=	000013	IPRI	002202	G			
BIT01	=	000002	G	C\$DODU	=	000051	EF.CON	=	000036	G	F\$INIT	=	000006	ISR	=	000100	G
BIT02	=	000004	G	C\$DRPT	=	000024	EF.NEW	=	000035	G	F\$JMP	=	000050	IVC	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	017067	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	016242	F\$SEG	=	000003	I\$HRD	=	000041			
BIT09	=	001000	G	C\$ERSO	=	000057	ENVIRN	020720	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	006356	F\$SUB	=	000002	I\$MSG	=	000041			
BIT11	=	004000	G	C\$ESUB	=	000003	EPRT2	006446	F\$SW	=	000014	I\$PROT	=	000040			
BIT12	=	010000	G	C\$ETST	=	000001	ERRM	012023	F\$TEST	=	000001	I\$PTAB	=	000041			
BIT13	=	020000	G	C\$EXIT	=	000032	ERRHI	002226	G	GDDAT	003150	G	I\$PWR	=	000041		
BIT14	=	040000	G	C\$GETB	=	000026	ERRK	017046	GERRMA	002164	G	I\$RPT	=	000041			
BIT15	=	100000	G	C\$GETW	=	000027	ERRLO	002230	G	GETPAT	020264	G	I\$SEG	=	000041		
BIT2	=	000004	G	C\$GMAN	=	000043	ERRNO	=	001513	GETSEL	020346	G	I\$SETU	=	000041		
BIT3	=	000010	G	C\$GPHR	=	000042	ERRVEC	=	000004	G	G\$CNTD	=	000200	I\$SFT	=	000041	
BIT4	=	000020	G	C\$GPLO	=	000030	ERTABE	003366	G\$DELM	=	000372	I\$SRV	=	000041			
BIT5	=	000040	G	C\$GPRI	=	000040	ERTABL	003166	G\$DISP	=	000003	I\$SUB	=	000041			
BIT6	=	000100	G	C\$INIT	=	000011	ESUM	017050	G\$EXCP	=	000400	I\$TST	=	000041			
BIT7	=	000200	G	C\$INLP	=	000020	EVL	=	000004	G	J\$JMP	=	000167				
BIT8	=	000400	G	C\$MANI	=	000050	EXBCNT	=	000010	G\$LOLI	=	000001	KIPAR0	=	172340		
BIT9	=	001000	G	C\$MEM	=	000031	EXIT	034224	G\$NO	=	000000	KIPAR1	=	172342			
BOE	=	000400	G	C\$MSG	=	000023	EXPBRE	015572	G	G\$OFFS	=	000400	KIPAR2	=	172344		
BRINIT	004455	C\$OPEN	=	000034	EXPD	002222	G	G\$OSI	=	000376	KIPAR3	=	172346				
BSELO	=	000000	C\$PNTB	=	000014	EXPGET	004531	G\$PRMA	=	000001	KIPAR4	=	172350				
BSEL1	=	000001	C\$PNTF	=	000017	EXPGET2	004565	G\$PRMD	=	000002	KIPAR5	=	172352				
CHKAMB	016134	C\$PNTS	=	000016	EXPMSG	002312	G	G\$PRML	=	000000	KIPAR6	=	172354				
CHKMAN	020570	G	C\$PNTX	=	000015	EXPREC	015564	G	G\$RADA	=	000140	KIPAR7	=	172356			
CHKTSS	016426	C\$QIO	=	000377	EXTA	005770	G\$RADB	=	000000	KIPDR0	=	172300					
CKDROP	017272	C\$RDBU	=	000007	EXTEND	005766	G\$RADD	=	000040	KIPDR1	=	172302					
CKEMAX	017172	C\$REFG	=	000047	EXTFEA	002216	G	G\$RADL	=	000120	KIPDR2	=	172304				
CKMSG	011450	G	C\$RESE	=	000033	E\$END	=	002100	G\$RADO	=	000020	KIPDR3	=	172306			
CKMSG2	011570	G	C\$REVI	=	000003	E\$LOAD	=	000035	G\$XFER	=	000004	KIPDR4	=	172310			
CKRAM	011204	G	C\$RFLA	=	000021	FATAL	034324	G\$YES	=	000010	KIPDR5	=	172312				
CKRAM2	011314	G	C\$RPT	=	000025	FATERR	=	000060	HIADDR	=	001400	KIPDR6	=	172314			
CMDPKT	021350	G	C\$SEFG	=	000046	FATFLG	002212	G	HOE	=	100000	G	KIPDR7	=	172316		
CMPMEM	017750	C\$SPRI	=	000041	FERCM	012012	HPM1	112410	HPM1	112410	KTENAB	003124	G				
CONFIG	017340	C\$SVEC	=	000037	FIFEXP	012260	G	HPM2	112444	KTFLG	003122	G					
COUNT	002300	G	C\$TPRI	=	000013	FIF1MS	012332	HPM3	112470	KTINIT	021144						
CSRADD	002176	G	DATA	002302	G	FIF2MS	012401	IBE	=	010000	G	KTOFF	017364				
CTAB	003154	G	DATASC	020322	DEBGM	011722	IDU	=	000040	G	KTON	017346					
CT4BE	003166	G	DEBUGM	011722	DEVDR0	002210	G	IER	=	020000	G	LERRMA	002162	G			
CTABM	003154	G	DEVCNT	002210	G	FORCER	002166	G	IF AULT	004254	LISTAL	=	000001				

SYMBOL TABLE

LOE = 040000 G	L\$UNIT 002012 G	L10071 055402	M8189 005643	PRBEXP 015560
LOOPCN 002206 G	L10000 002154	L10072 047454	NBA = 002000	PRBMSG 015426
LOOPCO 013216	L10001 002166	L10073 050054	NEWPAS 022126	PRBREC 015562
LOOPFL 003152 G	L10002 005764	L10074 050530	NODEV 003104 G	PRBTOT 015513
LOT = 000010 G	L10003 012134	L10075 051174	NOINIT 004333	PRBYTE 015212 G
L\$ACP 002110 G	L10004 012152	L10076 051734	NOINTR 004217	PRI = 002000 G
L\$APT 002036 G	L10005 012170	L10077 052674	NOITS 002160 G	PRIADD 010250
L\$AU 022470 G	L10006 012176	L10100 053214	NOMAN 020624	PRIAO 010320
L\$AUT 002070 G	L10007 012214	L10101 053616	NOMEM 005456	PRI BXO 007702 G
L\$AUTO 022674 G	L10010 012232	L10102 074724	NP.IR = 000200	PRIEQU 010150
L\$CCP 002106 G	L10011 012256	L10103 056340	NP.LOO = 000040	PRIPKT 007460 G
L\$CLEA 022754 G	L10012 012330	L10104 057206	NP.OUT = 000100	PRIRAM 010156
L\$CO 002032 G	L10013 012500	L10105 060100	NP.WRP = 000020	PRITAD 010364
L\$DEPO 002011 G	L10014 013214	L10106 061026	NSI 004150	PRITSS 006022
L\$DESC 003400 G	L10015 014042	L10107 061604	NSINIT 004405	PRITO 010446
L\$DESP 002076 G	L10016 014064	L10110 062446	NUL 004525	PRIT1 010511
L\$DEVP 002060 G	L10017 015570	L10111 063320	NULCR 004526	PRIXOR 010032 G
L\$DISP 002124 G	L10020 015576	L10112 064172	NXM = 004000	PRI00 = 000000 G
L\$DLY 002116 G	L10021 015604	L10113 065046	NXMFLG 003126 G	PRI01 = 000040 G
L\$DTP 002040 G	L10022 015616	L10114 065722	NXMHI 003132 G	PRI02 = 000100 G
L\$DTYP 002034 G	L10023 015640	L10115 066572	NXML0 003130 G	PRI03 = 000140 G
L\$DU 022566 G	L10024 015666	L10116 067524	NXMTST 021542	PRI04 = 000200 G
L\$DUT 002072 G	L10025 016026	L10117 070534	NXR 003736	PRI05 = 000240 G
L\$DVTY 003372 G	L10026 016336	L10120 071114	NXRERR 005734 G	PRI06 = 000300 G
L\$EF 002052 G	L10030 022420	L10121 071570	NXRX 003775	PRI07 = 000340 G
L\$ENVI 002044 G	L10031 022564	L10122 104742	NXTU 022140	PRMESS 014332
L\$ETP 002102 G	L10032 022672	L10123 075346	OFL = 000100	PRMNO 002310 G
L\$EXP1 002046 G	L10033 022752	L10124 076130	ONEFIL = 000000	PRMSGE 014642 G
L\$EXP4 002064 G	L10034 023000	L10125 076752	O\$APTS = 000000	PRMSG0 015022
L\$EXP5 002066 G	L10035 023242	L10126 077654	O\$AU = 000001	PRMSG1 015067
L\$HARD 112370 G	L10036 024700	L10127 101404	O\$BGNR = 000001	PRMSG2 015125
L\$HIME 002120 G	L10037 027360	L10130 112364	O\$BGNS = 000001	PROASC 014510
L\$HPCP 002016 G	L10040 025306	L10131 105342	O\$DU = 000001	PRIASC 014555
L\$HPTP 002022 G	L10041 025630	L10132 105622	O\$ERRT = 000000	PST32W 003142 G
L\$HW 002146 G	L10042 026210	L10133 110074	O\$GNSW = 000001	PUNIT 022422
L\$ICP 002104 G	L10043 034350	L10134 112410	O\$POIN = 000001	PW.D11 = 000021
L\$INIT 021646 G	L10044 027762	L10135 112530	O\$SETU = 000000	PW.D13 = 000022
L\$LADP 002026 G	L10045 030632	MEMADD 014044 G	PASRPT 022172	PW.D22 = 000020
L\$LAST 113004 G	L10046 031452	MEMCK 021366 G	PATCH 112670 G	PW.NOP = 000000
L\$LOAD 002100 G	L10047 031666	MENASC 020537	PC.DAT 020320	PW.NO1 = 000023
L\$LUN 002074 G	L10050 032120	MENERR 020464	PC.ERA = 002400	PW.RDE = 000024
L\$MREV 002050 G	L10051 032432	MENRES 020565	PC.IER = 002000	PW.RDR = 000001
L\$NAME 002000 G	L10052 046464	MMVEC = 000250	PC.NOO = 001000	PW.RDS = 000005
L\$PRIO 002042 G	L10053 035016	MSA.FR = 000006	PC.REL = 000000	PW.RFI = 000003
L\$PROT 021636 G	L10054 035576	MSA.NO = 000000	PC.REW = 000400	PW.WCT = 000006
L\$PRT 002112 G	L10055 036352	MSA.NR = 000004	PKBCNT = 000006	PW.WFI = 000004
L\$REPP 002062 G	L10056 037054	MSA.VO = 000002	PKHI = 000004	PW.WFM = 000007
L\$REV 002010 G	L10057 037520	MSGEXP 012234 G	PKLOW = 000002	PW.WMI = 000010
L\$RPT 023002 G	L10060 040154	MSGLOO 013154 G	PKTADD 007644	PW.WNP = 000011
L\$SOFT 112522 G	L10061 040610	MSGSTA 012440 G	PKTFRM 007606	PW.WTR = 000002
L\$SPC 002056 G	L10062 041202	MSGSUB 014032 G	PKTGET 012154 G	P.ACK = 100000
L\$SPCP 002020 G	L10063 041620	MS.ATT = 000006	PKTMES 012200 G	P.CMD = 000037
L\$SPTP 002024 G	L10064 042064	MS.EXT = 000200	PKTRAM 004743 G	P.CONT = 000012
L\$STA 002030 G	L10065 042336	MS.RSD = 000001	PKTSSR 012136 G	P.CVC = 040000
L\$SW 002156 G	L10066 042572	MS.RSF = 000020	PNT = 001000 G	P.FMT = 000140
L\$TEST 002114 G	L10067 043072	MS.RST = 000010	PRAMPK 014066	P.FORM = 000011
L\$TIML 002014 G	L10070 043556	M8186 005552	PRASC 014613	P.GETS = 000017

SYMBOL TABLE

P.IE = 000200	SPM6 112610	TSREJ = 000006	T\$\$CLE= 010034	T22WRT 026400
P.INIT= 000013	SPM7 112640	TSSDEF 006676	T\$\$DU = 010032	T23A 003134 G
P.MODE= 007400	SR0 = 177572	TSSR = 000002 G	T\$\$HAR= 010134	T23AM3 033240
P.OPP = 020000	SR1 = 177574	TSSRBI 003500 G	T\$\$HW = 010000	T23B 003136 G
P.POSI= 000010	SR2 = 177576	TSSRFO 006505	T\$\$INI= 010030	T23BA 033625
P.READ= 000001	SR3 = 172516	TSSR+ = 000003 G	T\$\$MSG= 010025	T23BFR 032512
P.SWB = 010000	SSR = 000200	TSSX 004016	T\$\$PRO= 010027	T23BF2 032622
P.WRIT= 000005	STATCO 012502	TSTBLK 002742 G	T\$\$RPT= 010035	T23BS0 032622
P.WRTC= 000004	SVCGBL= 000000	TSTCNT 002204 G	T\$\$SOF= 010135	T23BS1 032623
P.WRTS= 000006	SVCINS= 000000	TSTEND 017010	T\$\$SRV= 010026	T23CHK 034162
QVP 002174 G	SVCSUB= 000001	TSTFLA 002304 G	T\$\$SUB= 010133	T23CON 032640
RAMASC 014246	SVCTAG= 000000	TSTL00 016546 G	T\$\$SW = 010001	T23DAT 032500
RAMDAT 002232 G	SVCTST= 000001	TSTPTR 002306 G	T\$\$TES= 010130	T23DSW 032510
RAMERR 015600 G	S\$LSYM= 010000	TSTSET 016600 G	T1 023564 G	T23EOT 032764
RAMEXP 015620 G	SO.IDB= 000010	TST21I 024524	T2 024702 G	T23ET 032677
RAMFOR 010206	SO.IFB= 000002	TST22I 027167	T2.1 024732	T23L00 027426
RAMSIZ 002272 G	SO.IFP= 000001	TST23I 033766	T2.2 025324	T23OFL 033306
RAMTAD 015606 G	SO.ILD= 000020	TST24I 046232	T2.3 025646	T23PAC 032470
RCVHIA 002274 G	SO.ION= 000040	TST25I 055200	T21AM3 024403	T23PK2 032600
RCVLOA 002276 G	SO.IRD= 000100	TST26I 074527	T21BFR 024204	T23PK3 032610
RDERR 005204	SO.IRW= 000004	TST27I 104543	T21BF2 024300	T23RES 034002
RECMG 002456 G	SO.ISP= 000200	TST28I 112161	T21BS0 024300	T23RNC 033165
RECV 002224 G	S1.ICE= 002000	TSV2 002000 G	T21BS1 024301	T23RSZ 032620
REGSAV 020230	S1.IEO= 010000	TSV3 002166 G	T21DAT 024170	T23RT2 034074
RETErr 005370	S1.IFM= 001000	TSV4 021636 G	T21DLY 024202	T23RT3 034136
RETRY 034226	S1.IHE= 000400	TSV6 112366 G	T21DSW 024200	T23RWN 033116
REWIND 011104 G	S1.IID= 004000	TSV7 023564 G	T21L00 023614	T23SSR 032644
RMCHBE= 000167	S1.IIR= 020000	TTIBFR= 177562 G	T21OFL 024503	T23SZ 032616
RMCHEN= 000200	S1.I2R= 040000	TTICSR= 177560 G	T21PAC 024160	T23S2 032624
RMMSGB= 000215	S1.PAR= 100000	TTIVEC= 000060 G	T21PK2 024270	T23S3 032626
RMMSGE= 000234	S2.ATI= 000010	T\$ARGC= 000003	T21RES 024546	T23TM 033042
RMPKTB= 000201	S2.BTI= 000004	T\$CODE= 001130	T21RT2 024636	T23TMP 032630
RMPKTE= 000210	S2.DIM= 000200	T\$ERRN= 001513	T21SSR 024306	T23VCK 033552
RMR = 010000	S2.ILW= 000100	T\$EXCP= 000000	T21S2 024302	T23WB 032612
RWPACK 011200	S2.INR= 000020	T\$FLAG= 000040	T21S3 024304	T23WD 032634
SC = 100000	S2.OUT= 000040	T\$GMAN= 000000	T22AM3 026505	T23WDC 033450
SCE = 020000	S2.UND= 000003	T\$HILI= 000776	T22BFR 026272	T23WDD 033361
SCHERR 005276	TBLEND= 003052 G	T\$LAST= 000001	T22BF2 026370	T23WDR 032636
SCME 005011	TCOASC 006566	T\$LOLI= 000000	T22BS0 026370	T23WRT 032632
SDELAY 010750	TCOCOD 006766	T\$LSYM= 010000	T22BS1 026371	T23WSS 033677
SELASC 020532	TEMP1 003106 G	T\$LTNO= 000010	T22DAT 026260	T24AM3 045220
SELDAT= 000004	TEMP2 003110 G	T\$NEST= 177777	T22FOR 026404	T24BA 045552
SEL2 = 000002	TERCLS= 000016	T\$NS0 = 000000	T22L00 024732	T24BFR 043642
SETMAP 017406	TESTNO= 000010	T\$NS1 = 000005	T22OFL 026605	T24BF2 043750
SETU 022224	TEXASC 006525	T\$NS2 = 000002	T22PAC 026250	T24BOT 044613
SFFMSG 012172 G	TFCASC 006627	T\$PTNU= 000000	T22PK2 026360	T24BS0 043750
SFHERR 003703	TIMEXP 015642 G	T\$SAVL= 177777	T22POS 026402	T24BS1 043751
SFIERR 003650	TIMSGO 015670	T\$SEGL= 177777	T22RD 026376	T24CON 043762
SFIMSG 012124 G	TINERR 012111	T\$SUBN= 000003	T22RES 027222	T24DAT 043630
SFPTBL 002156 G	TMPBFR 002622 G	T\$TAGL= 177777	T22RT2 027314	T24DLY 043766
SIFLAG 003144 G	TNAM 016774	T\$TAGN= 010136	T22RWJ 026754	T24DSW 043640
SIMSG 012056	TRANST 002156 G	T\$TEMP= 000000	T22SSR 026410	T24DTA 044660
SKIPT 003370	TSBA = 000000 G	T\$TEST= 000010	T22S2 026372	T24EOT 044746
SOFINI 016064 G	TSBAH = 000001 G	T\$TSTM= 177777	T22S3 026374	T24ILA 044342
SPACE 010556 G	TSDB = 000000 G	T\$TSTS= 000001	T22TM 026660	T24LON 045712
SPM1 112530	TSDBH = 000001 G	T\$\$AU = 010031	T22VCK 027027	T24L00 034412
SPM4 112560	TSFCOD 007326	T\$\$AUT= 010033	T22WLK 027102	T24LOP 045774

SYMBOL TABLE

T24LOQ	044426	T25SSR	054034	T26WDC	073640	T27WDR	101600	T4	034352	G
T24LOR	044042	T25SZ	054006	T26WDD	073550	T27WNG	101614	T4.1	034412	
T24NEF	043770	T25S2	054012	T26WDE	072743	T27WRF	104366	T4.10	041636	
T24NXM	044201	T25S3	054014	T26WDF	072551	T27WSS	103522	T4.11	042102	
T24OFL	045265	T25TM	054242	T26WNG	072006	T28BFR	110152	T4.12	042354	
T24PAC	043620	T25WB	054002	T26WSS	074041	T28BF2	110260	T4.13	042610	
T24PBP	046056	T25WDC	055127	T27AM3	103107	T28BOT	111117	T4.14	043110	
T24PK2	043730	T25WDE	054115	T27BA	103447	T28BS0	110260	T4.2	035034	
T24PK3	043740	T25WDR	054020	T27BFR	101462	T28BS1	110261	T4.3	035614	
T24RB	043742	T25WNG	054405	T27BF2	101570	T28CNT	110306	T4.4	036370	
T24RES	046300	T25WNH	054560	T27BOT	102461	T28CMU	110310	T4.5	037072	
T24RN	043756	T26AM3	073426	T27BS0	101570	T28CON	110302	T4.6	037536	
T24RNC	045145	T26BA	073766	T27BS1	101571	T28DAT	110140	T4.7	040172	
T24RT2	046372	T26BFR	071652	T27CNT	101606	T28DLY	110312	T4.8	040626	
T24RT3	046434	T26BF2	071760	T27CMU	101610	T28DSW	110150	T4.9	041220	
T24RWN	045076	T26BOT	073015	T27CON	101602	T28DTA	112064	T5	046466	G
T24SSR	044507	T26BS0	071760	T27DAT	101450	T28DTR	112002	T5.1	046516	
T24SZ	043746	T26BS1	071761	T27DLY	101612	T28IMV	110266	T5.2	047472	
T24S2	043752	T26CNT	071776	T27DSW	101460	T28LOO	105000	T5.3	050072	
T24S3	043754	T26CMU	072000	T27DTA	104446	T28LOQ	110674	T5.4	050546	
T24TM	045023	T26DAT	071640	T27EOT	102631	T28OFL	111310	T5.5	051212	
T24TRL	046144	T26DLY	072004	T27LON	103611	T28PAC	110130	T5.6	051752	
T24VCK	045477	T26DSW	071650	T27LOO	074766	T28PBP	110371	T5.7	052712	
T24WB	043742	T26DTA	073062	T27LOP	103673	T28PK2	110240	T5.8	053232	
T24WDC	045426	T26EOT	073150	T27LOQ	102255	T28PK3	110250	T6	055404	G
T24WDD	045340	T26LON	074130	T27LOR	102130	T28RB	110252	T6.1	055444	
T24WDE	044541	T26LOO	055444	T27NEF	104131	T28RDF	110454	T6.10	065064	
T24WDF	044265	T26LOP	074212	T27OFL	103156	T28RDG	110535	T6.11	065740	
T24WDG	044112	T26LOQ	072626	T27PAC	101440	T28RES	112206	T6.12	066610	
T24WDR	043760	T26LOR	072501	T27PBP	103755	T28RIB	110314	T6.13	067542	
T24WSS	045623	T26NEF	072074	T27PK2	101550	T28RN	110276	T6.14	070552	
T25BFR	053702	T26NEQ	074450	T27PK3	101560	T28RRM	111567	T6.15	071132	
T25BF2	054010	T26OFL	073475	T27RB	101562	T28RRN	111645	T6.2	056356	
T25BNC	054470	T26PAC	071630	T27RDF	101702	T28RRP	111724	T6.3	057224	
T25BOT	054175	T26PBP	074274	T27RES	104564	T28RT2	112300	T6.4	060116	
T25BS0	054010	T26PK2	071740	T27RN	101576	T28RT3	112342	T6.5	061044	
T25BS1	054011	T26PK3	071750	T27RNC	103034	T28RWN	111241	T6.6	061622	
T25CNT	054030	T26RB	071752	T27RRF	101751	T28SSR	110755	T6.7	062464	
T25CN2	054026	T26RDF	072156	T27RT2	104656	T28SZ	110256	T6.8	063336	
T25CON	054022	T26RES	074540	T27RT3	104720	T28S2	110262	T6.9	064210	
T25DAT	053670	T26RN	071766	T27RWN	102765	T28S3	110264	T7	074726	G
T25DLY	054032	T26RNC	073353	T27SC	102046	T28TM	111164	T7.1	074766	
T25DSW	053700	T26RRF	072225	T27SCF	104227	T28TMK	111515	T7.2	075364	
T25LOO	046516	T26RRG	072322	T27SSR	102336	T28VCK	111442	T7.3	076146	
T25NEF	054643	T26RSZ	072002	T27SZ	101566	T28WB	110252	T7.4	076770	
T25NET	054331	T26RT2	074632	T27S2	101572	T28WDC	111363	T7.5	077672	
T25OFL	055054	T26RT3	074674	T27S3	101574	T28WDE	111026	T8	104744	G
T25PAC	053660	T26RWN	073304	T27TIM	102554	T28WDF	110617	T8.1	105000	
T25PK2	053770	T26SC	072417	T27TM	102710	T28WDR	110300	T8.2	105360	
T25PK3	054000	T26SSR	072707	T27TRL	104043	T3	027362	T8.3	105640	
T25RB	054002	T26SZ	071756	T27TSA	104304	T3BFLG	003140	UAM	= 000200	G
T25RES	055216	T26S2	071762	T27VCK	103374	T3.1	027426	UNITN	= 002172	G
T25RIB	054723	T26S3	071764	T27WB	101562	T3.2	030000	UNREC	= 000006	
T25RN	054016	T26TM	073227	T27WDC	103321	T3.3	030650	USI	= 004121	
T25RT2	055310	T26TRL	074362	T27WDD	103231	T3.4	031470	WAITF	= 016340	G
T25RT3	055352	T26VCK	073713	T27WDE	102372	T3.5	031704	WC. IFA	= 000200	
T25RWN	055005	T26WB	071752	T27WDF	102200	T3.6	032136	WC. IFE	= 000002	

SYMBOL TABLE

WC.IGO= 000001	WRTERR 005111	XSOILA= 000400	X#FALS= 000040	X2.UNI= 000007
WC.IRE= 000010	WRTMSG 005054	XSOILC= 001000	X#OFFS= 000400	X2.WCF= 002000
WC.IRW= 000004	WSMBK 021360 G	XSOLET= 020000	X#TRUE= 000020	X3.DCK= 000010
WC.IOT= 000100	XFERAS 016030	XSMOT= 000200	X1.COR= 020000	X3.MBZ= 000006
WC.IIT= 000040	XNXM 016466	XSONEF= 002000	X1.DLT= 100000	X3.MDE= 177400
WC.ISR= 000020	XORBFO 007764	XSOONL= 000100	X1.MBZ= 017375	X3.OPI= 000100
WF.IED= 000010	XORFOR 010102	XSOPED= 000010	X1.RBP= 000400	X3.REV= 000040
WF.IER= 000004	XST0 = 000006 G	XSORLL 010000	X1.SPA= 040000	X3.RIB= 000001
WF.IHI= 000200	XST1 = 000010 G	XSORLS= 040000	X1.UNC= 000002	X3.SPA= 000200
WF.IRE= 000040	XST2 = 000012 G	XSOTMK= 100000	X2.BUF= 000100	X3.TRF= 000020
WF.IWF= 000020	XST3 = 000014 G	XSOVCK= 000020	X2.EXT= 000200	X4.HSP= 100000
WF.IWR= 000100	XST4 = 000016 G	XSOWLF= 004000	X2.OPM= 100000	X4.MBZ= 017400
WF.I3R= 000002	XSOBOT= 000002	XSOWLK= 000004	X2.RCE= 040000	X4.RCE= 040000
WF.I4R= 000001	XSOEOT= 000001	XXCOMM 003112 G	X2.REV= 000077	X4.TSM= 020000
WRTCHR 010752 G	XSOIE = 000040	X#ALWA= 000000	X2.SPA= 035400	X4.WPC= 000377

. ABS. 113004 000
 000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 28880 WORDS (113 PAGES)

DYNAMIC MEMORY: 20060 WORDS (77 PAGES)

ELAPSED TIME: 00:51:20

CNTSCAO.BIC,CNTSCAO.SEQ/ SP=SVC34/ML,TSV1C,TSV22C,TSV3B,TSV4,TSV7A,TSV6