

11/21+
TSV05

TSV05 CTRL LT4
CNTSDAO

COPYRIGHT (c) 1982-84
AH-T821A-MC
FICHE 01 OF 02

JUL 1984
digital
Made In USA

The main body of the document is a microfiche grid containing approximately 10 columns and 20 rows of data. Each cell in the grid contains a small, high-contrast image of a document page, which is a scan of a microfiche frame. The text within these frames is extremely small and difficult to read, but it appears to be organized into columns and rows, possibly representing a data table or a series of related documents. The overall appearance is that of a standard microfiche sheet used for data storage and retrieval.

11/21+
TSV05

TSV05 CTRL LT4
CNTSDAO

COPYRIGHT (c) 1982-84
RH-T821A-MC
FICHE 02 OF 02

JUL 1984
digital
Made In USA

.REM_

IDENTIFICATION

PRODUCT ID: AC-T820A-MC
PRODUCT TITLE: CNTSDAO TSV05 CTRL LT4
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 AND 4K RESERVED 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 PREREQUISITES

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

1.5 ASSUMPTIONS

ALL HARDWARE EXCEPT THE HARDWARE UNDER TEST IS ASSUMED TO WORK

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

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

COMMAND	EFFECT
-----	-----
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER +C)
PROCEED	CONTINUE FROM AN ERROR F.A.L.T
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 XXDP MEDIA

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

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION.

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

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

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

CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

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

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

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

2.4 HARDWARE QUESTIONS

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

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

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

```
TSBA/TSDB = 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 "IXE" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 SPECIFIC ERROR MESSAGES

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

ERROR MESSAGE EXAMPLE 1

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

TST: 016 FIFO EXERCISER TEST
 CNTSD 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.

CNTSD 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.

CNTSD HRD ERR 00121 ON UNIT 00 TST 001 SUB 002 PC: 023306
MOT BIT (XSTO) 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:HOE

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

0 ERRORS

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

PROGRAM RUN TIMES

THE AVERAGE RUN TIMES OF THE PROGRAM ARE LISTED BELOW. THESE FIGURES ARE TO BE USED AS A GUIDE. THE TIMING WAS DONE ON A 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 TIME REQUIRED TO RUN TESTS 1 THROUGH 9 IN ONE COMMAND IS 3 HOURS.

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: WRITE TAPE MARK RETRY

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

TEST 2: SKIP TAPE MARKS

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

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

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

TEST 4: ERASE AND OPERATION INCOMPLETE

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

TEST 5: DATA PARITY TEST

THIS TEST VERIFIES THAT THE DATA PARITY CIRCUITRY IN BOTH THE CONTROLLER AND THE TRANSPORT IS OPERATING PROPERLY BY FORCING DATA RECORDS WITH WRONG PARITY TO BE WRITTEN ONTO TAPE AND CHECKING THE RESULTS OBTAINED WHEN THE DATA IS READ.

TEST 6: OPERATIONS AT EOT

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

TEST 7: EXTENDED MODE FEATURES

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

REWIND WITH IMMEDIATE INTERRUPT

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

TEST 8: RECORD BUFFERING

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

TEST 9: FUNCTION TIMING

THIS TEST VERIFIES THAT THE TAPE TRANSPORT SEEMS TO BE WRITING RECORDS, GAPS, AND EXTENDED GAPS OF THE PROPER LENGTH. BOTH LOW AND HIGH SPEED MODES ARE TESTED. IT IS ALSO VERIFIED THAT A SKIP TAPE MARKS COMMAND WITH A COUNT OF 6 OR MORE, OPERATE THE TAPE IN HIGH-SPEED MODE. THIS TEST CAN ONLY BE RUN IF A REAL-TIME CLOCK IS AVAILIABLE ON THE SYSTEM. THE TEST OPERATES BY TIMING VARIOUS TAPE-MOTION OPERATIONS, USING A NUMBER OF DIFFERENT TEST RECORD LENGTHS.

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

REVISION B - APRIL 1983

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

CVTSDBO => CNTSDAO

JAKI BERG

9-APR-1984

CHANGES WERE MADE TO CVTSDBO TO PRODUCE CNTSDAO 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
22         002000
23
24         ;**
25         ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
26         ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
27         ;--
28
29         POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
30 002000  HEADER  CNTSD,A,0,655.,0
          L$NAME::          ;DIAGNOSTIC NAME
          .ASCII /C/
          .ASCII /N/
          .ASCII /T/
          .ASCII /S/
          .ASCII /D/
          .BYTE  0
          .BYTE  0
          .BYTE  0
          L$REV::          ;REVISION LEVEL
          .ASCII  /A/
          L$DEPO::          ;0
          .ASCII  /O/
          L$UNIT::          ;NUMBER OF UNITS
          .WORD   0
          L$TIML::          ;LONGEST TEST TIME
          .WORD   655.
          L$HPCP::          ;PTR. TO H.W. QUES.
          .WORD   L$HARD
          L$SPCP::          ;PTR. TO S.W. QUES.
          .WORD   L$SOFT
          L$HPTP::          ;PTR. TO DEF. H.W. PTABLE
          .WORD   L$HW
          L$SPTP::          ;PTR. TO S.W. PTABLE
          .WORD   L$SW
          L$LADP::          ;DIAG. END ADDRESS
          .WORD   L$LAST
          L$STA::          ;RESERVED FOR APT STATS
          .WORD   0
          L$CO::          ;DIAGNOSTIC TYPE
          .WORD   0
          L$DTP::          ;DIAGNOSTIC TYPE
          .WORD   0
          L$APT::          ;APT EXPANSION
          .WORD   0
          L$DTP::          ;PTR. TO DISPATCH TABLE
          .WORD   L$DISPATCH

```

```

2
3
4
10
11 000000
12
13
19 000000
20         002000
21 002000 002000
22
23
24
25
26
27
28
29 002000
30 002000
          002000
          002000 103
          002001 116
          002002 124
          002003 123
          002004 104
          002005 000
          002006 000
          002007 000
          002010
          002010 101
          002011
          002011 060
          002012
          002012 000000
          002014
          002014 001217
          002016
          002016 105576
          002020
          002020 105730
          002022
          002022 002150
          002024
          002024 002160
          002026
          002026 106404
          002030
          002030 000000
          002032
          002032 000000
          002034
          002034 000000
          002036
          002036 000000
          002040
          002040 002124

```

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	003374	.WORD	L\$DVTYP	
002062		L\$REPP::		;PTR. TO REPORT CODE
002062	022772	.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	022460	.WORD	L\$AU	
002072		L\$DUT::		;PTR. TO DROP UNIT CODE
002072	022556	.WORD	L\$DU	
002074		L\$LUN::		;LUN FOR EXERCISERS TO FILL
002074	000000	.WORD	0	
002076		L\$DESP::		;POINTER TO DIAG. DESCRIPTION
002076	003402	.WORD	L\$DESC	
002100		L\$LOAD::		;GENERATE SPECIAL AUTOLOAD EMT
002100	104035	EMT	E\$LOAD	
002102		L\$ETP::		;POINTER TO ERR_TBL
002102	000000	.WORD	0	
002104		L\$ICP::		;PTR. TO INIT CODE
002104	021636	.WORD	L\$INIT	
002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
002106	022744	.WORD	L\$CLEAN	
002110		L\$ACP::		;PTR. TO AUTO CODE
002110	022664	.WORD	L\$AUTO	
002112		L\$PRT::		;PTR. TO PROTECT TABLE
002112	021626	.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	

F2

DISPATCH TABLE

.SBTTL DISPATCH TABLE

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

32
33
34
35
36
37
38
39 002122
002122 000011
002124
002124 023554
002126 032364
002130 041462
002132 047020
002134 053076
002136 056072
002140 063444
002142 073374
002144 101150
40

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

DEFAULT HARDWARE P-TABLE

```

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 002146    BGNHW   DFPTBL      ;DEFAULT HARD-P-TABLE
          002146    000003    .WORD   L10000-L$HW/2
          002150
          002150
50
51 002150    176000    .WORD   176000      ; 1ST (OF 2) REGISTERS.
52 002152    000224    .WORD   224        ; INTERRUPT VECTOR
53 002154    000200    .WORD   PRI04      ; INTERRUPT PRIORITY.
54 002156
          002156    ENDDHW
          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          BGNSW  SFPTBL
           002156 000004 .WORD  L10001-L$SW/2
           002160
           002160
63          L$SW::
           SFPTBL::
64          TRANSTST:: .WORD  0      ; ENABLE TEST OF TRANSPORT(S) IF =1
65          NOITS::   .WORD  0      ; INHIBIT ITERATION OPTION.
66                                     ; ... 0 = ITERATE.
67                                     ; ...NZ = INHIBIT ITERATE.
68          LERRMAX:: .WORD  15.    ; LOCAL (PER TEST) ERROR LIMIT
69          GERRMAX:: .WORD  200.   ; GLOBAL (PER UNIT) ERROR LIMIT
70          ENDSW
           002170
           002170
71          L10001:
72          002170          ENDMOD
    
```

SOFTWARE P-TABLE

7
8
13
19
20 002170
002170
21
22
23
24
25
26
27
28
32 002170

```

.TITLE TSV3 - GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

BGNMOD TSV3
TSV3::

.SBTTL GLOBAL EQUATES SECTION

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

EQUALS          ; GET STANDARD EQUATES.

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

;
001000          BIT9==  BIT09
000400          BIT8==  BIT08
000200          BIT7==  BIT07
000100          BIT6==  BIT06
000040          BIT5==  BIT05
000020          BIT4==  BIT04
000010          BIT3==  BIT03
000004          BIT2==  BIT02
000002          BIT1==  BIT01
000001          BIT0==  BIT00

;
; EVENT FLAG DEFINITIONS
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
;
;
000040          EF.START==      32.          ; BIT POSITION IN SECOND STATUS WORD
000037          EF.RESTART==    31.          ; (100000) START COMMAND WAS ISSUED
000036          EF.CONTINUE==   30.          ; (040000) RESTART COMMAND WAS ISSUED
000035          EF.NEW==        29.          ; (020000) CONTINUE COMMAND WAS ISSUED
000034          EF.PWR==        28.          ; (010000) A NEW PASS HAS BEEN STARTED
;                                     ; (004000) A POWER-FAIL/POWER-UP OCCURRED
;
;

```

GLOBAL EQUATES SECTION

```

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

```

; OPERATOR FLAG BITS
;
000004 EVL== 4
000010 LOT== 10
000020 ADR== 20
000040 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 002170

```

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


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  TTIVEC==     60          ; INTERRUPT VECTOR FOR CONSOLE INPUT
47          177560  TTICSR==    177560       ; BUS ADDRESS OF CONSOLE INPUT
48          177562  TTIBFR==    177562       ; CONSOLE INPUT DATA BUFFER
49          177520  BDVPCR==    177520       ; BDV11 PAGE CONTROL REGISTER
50
51          ;+
52          ;BIT DEFINITIONS FOR TSSR REGISTER
53          ;-
54
55          100000  SC=      BIT15          ;SPECIAL CONDITION
56          040000  BIE=     BIT14          ;BUS INTERFACE ERROR
57          020000  SCE=     BIT13          ;SANITY CHECK ERROR
58          010000  RMR=     BIT12          ;MODIFICATION REFUSED
59          004000  NXM=     BIT11          ;NONEXISTANT MEMORY ERROR
60          002000  NBA=     BIT10          ;NEED BUFFER ADDRESS
61          001400  HIADDR=  BIT9!BIT8      ;EXTENDED ADDRESS BITS
62          000200  SSR=     BIT7           ;SUB SYSTEM READY
63          000100  OFL=     BIT6           ;OFF LINE BIT
64          000060  FATERR=  BIT4!BIT5      ;FATAL TERMINATION ERROR CODES
65          000016  TERCLS=  BIT3!BIT2!BIT1 ;TERMINATION CODES
66
67          ;+
68          ;
69          ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
70          ;(XST0)
71          ;
72          ;-
73
74          100000  XSOTMK=  BIT15          ;TAPE MARK DETECTED
75          040000  XSORLS=  BIT14          ;RECORD LENGTH SHORT
76          020000  XSOLET=  BIT13          ;LOGICAL END OF TAPE
77          010000  XSORLL=  BIT12          ;RECORD LENGTH LONG
78          004000  XSOWLE=  BIT11          ;WRITE LOCK ERROR
79          002000  XSONEF=  BIT10          ;NON EXECUTABLE FUNCTION
80          001000  XSOILC=  BIT9           ;ILLEGAL COMMAND
81          000400  XSOILA=  BIT8           ;ILLEGAL ADDRESS
82          000200  XSOMOT=  BIT7          ;TAPE IN MOTION
83          000100  XSOONL=  BIT6          ;TRANSPORT ON LINE
84          000040  XSOIE=   BIT5          ;INTERRUPT ENABLE
85          000020  XSOVCK=  BIT4          ;VOLUME CHECK BIT
86          000010  XSOPED=  BIT3          ;PHASE ENCODED DRIVE
87          000004  XSOWLK=  BIT2          ;WRITE LOCKED
88          000002  XSOBOT=  BIT1          ;BEGINNING OF TAPE
89          000001  XSOEOT=  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     000077      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 = BIT17: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:BIT5      ;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

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

210      ;
211      ;
212      ;
213      000006      XST0== 6      ;EXTENDED STATUS REGISTER 0 (WORD 4)
214      000010      XST1== 8      ;EXTENDED STATUS REGISTER 1 (WORD 5)
215      000012      XST2== 10     ;EXTENDED STATUS REGISTER 2 (WORD 6)
216      000014      XST3== 12     ;EXTENDED STATUS REGISTER 3 (WORD 7)
217      000016      XST4== 14     ;EXTENDED STATUS REGISTER 4 (WORD 8)
218
219      ;*
220      ;
221      ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
222      ;
223      ;
224
225      000002      PKLOW   = 2      ;LOW ORDER CHARACTERISTIC DATA POINTER
226      000004      PKHI    = 4      ;HIGH ORDER CHARACTERISTIC DATA POINTER
227      000006      PKBCNT  = 6      ;NUMBER OF BYTES IN DATA PACKET
228
229      000010      EXBCNT=10      ;NUMBER OF BYTES IN EXTENDED DATA PACKET
230
231      ;*
232      ;DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
233      ;
234      000000      BSELO   = 0      ;BYTE 0
235      000001      BSEL1   = 1      ;BYTE 1
236      000002      SEL2    = 2      ;WORD 2
237      000004      SELDATA = 4      ;WORD 3
238
239      ;*
240      ;BSELO SELECT CODES FOR WRITE SUBSYSTEM COMMAND
241      ;
242      000000      PW.NOP    = 0      ;NO-OP
243      000001      PW.RDRAM  = 1      ;READ RAM
244      000002      PW.WTRAM  = 2      ;WRITE RAM
245      000003      PW.RFIFO  = 3      ;READ FIFO
246      000004      PW.WFIFO  = 4      ;WRITE FIFO
247      000005      PW.RDSTAT = 5      ;READ STATUS
248      000006      PW.WCTL   = 6      ;WRITE TAPE CONTROL
249      000007      PW.WFMT   = 7      ;WRITE TAPE FORMAT
250      000010      PW.WMISC  = 10     ;WRITE MISCELLANEOUS
251      000011      PW.WNPR   = 11     ;WRITE NPR CONTROL
252      000020      PW.D22    = 20     ;DO MICROTEST 22
253      000021      PW.D11    = 21     ;DO MICROTEST 11
254      000022      PW.D13    = 22     ;DO MICROTEST 13
255      000023      PW.NO1311 = 23     ;DISABLE MICROTEST 11 AND 13
256      000024      PW.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    ;IRESV5 - RESERVED #5
265      000010      WC.IREW    = BIT3    ;IREW - REWIND
266      000004      WC.IRWU    = BIT2    ;IRWU - REWIND AND UNLOAD

```

D3

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 TRANSITION
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

```

E3

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 ERROR SET
362      ; FORCER TO 17777
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

```


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 000000 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 002170 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 002172 000000 EPRTSW:: .WORD 0 ;PRINT SWITCH
420 002174 000000 UNITN:: .WORD 0 ;UNIT # UNDER TEST.
421 002176 000000 QVP:: .WORD 0 ;QUICK VERIFY FLAG.
422 002200 000000 CSRADDR:: .WORD 0 ;ADDRESS OF CSR FOR CURRENT DEVICE
423 002202 000224 IVEC:: .WORD 224 ;INTERRUPT VECTOR
424 002204 000200 IPRI:: .WORD PRI04 ;INTERRUPT PRIORITY.
425 002206 000000 TSTCNT:: .WORD 0 ;NUMBER OF TESTS RUN IN THIS PASS
426 002210 000000 LOOPCNT:: .WORD 0 ;REMAINING ITERATION COUNT FOR TEST
427 002212 000000 DEVCNT:: .WORD 0 ;NUMBER OF DEVICE UNDER TEST
428 002214 000000 FATFLG:: .WORD 0 ;SET IF FATAL ERROR IS DETECTED IN TEST
429 002216 000000 INTRECV:: .WORD 0 ;SET IF TAPE INTERRUPT WAS RECEIVED
430 002220 000000 EXTFEA:: .WORD 0 ;EXTENDED FEATURES SOFTWARE SW 0=OFF;1=ON
431 002222 000000 BENBSW:: .WORD 0 ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
432 002224 000000 EXPD:: .WORD 0 ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
433 002226 000000 RECV:: .WORD 0 ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
434 002230 000000 ERRHI:: .WORD 0 ;HIGH ADDRESS MEMORY ERROR
435 002232 000000 ERRLO:: .WORD 0 ;LOW ADDRESS MEMORY ERROR
436 002234 RAMDATA:: .BLKW 16. ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
437 002274 000000 RAMSIZ:: .WORD 0 ;RAM DATA SIZE FOR PRAMPKT ROUTINE

```

GLOBAL DATA SECTION

```

438 002276 000000 RCVHIADD:: .WORD 0 ;RECEIVED BUFFER HIGH ADDRESS
439 002300 000000 RCVLOADD:: .WORD 0 ;RECEIVED BUFFER LOW ADDRESS
440 002302 000000 COUNT:: .WORD 0 ;TEST COUNT PATTERN
441 002304 000000 DATA:: .WORD 0 ;TEST DATA
442 002306 000000 TSTFLAG:: .WORD 0 ;TEST FLAG WORD
443 002310 000000 TSTPTR:: .WORD 0 ;TSTBLK POINTER
444 002312 000000 PRMNO:: .WORD 0 ;PRINT ROUTINE TEMP
445 002314 EXPMSG:: .BLKB 100. ;EXPECTED MESSAGE BUFFER DATA
446 002460 RECMSG:: .BLKB 100. ;RECEIVED MESSAGE BUFFER DATA
447 002624 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 002744 TSTBLK::
465 002744 000000 .WORD 0 ;ALL ZEROS
466 002746 177777 .WORD 177777 ;ALL ONES
467 002750 000001 .WORD BIT0 ;DATA FOR WALKING ONES
468 002752 000002 .WORD BIT1
469 002754 000004 .WORD BIT2
470 002756 000010 .WORD BIT3
471 002760 000020 .WORD BIT4
472 002762 000040 .WORD BIT5
473 002764 000100 .WORD BIT6
474 002766 000200 .WORD BIT7
475 002770 000400 .WORD BIT8
476 002772 001000 .WORD BIT9
477 002774 002000 .WORD BIT10
478 002776 004000 .WORD BIT11
479 003000 010000 .WORD BIT12
480 003002 020000 .WORD BIT13
481 003004 040000 .WORD BIT14
482 003006 100000 .WORD BIT15
483 003010 177776 .WORD †CBIT0 ;DATA FOR WALKING ZEROS
484 003012 177775 .WORD †CBIT1
485 003014 177773 .WORD †CBIT2
486 003016 177767 .WORD †CBIT3
487 003020 177757 .WORD †CBIT4
488 003022 177737 .WORD †CBIT5
489 003024 177677 .WORD †CBIT6
490 003026 177577 .WORD †CBIT7
491 003030 177377 .WORD †CBIT8
492 003032 176777 .WORD †CBIT9
493 003034 175777 .WORD †CBIT10
494 003036 173777 .WORD †CBIT11

```

TSTBLK - TEST DATA TABLE

```

495 003040 167777 .WORD †CBIT12
496 003042 157777 .WORD †CBIT13
497 003044 137777 .WORD †CBIT14
498 003046 077777 .WORD †CBIT15
499 003050 125252 .WORD 125252 ;ALTERNATING ONES, ZEROS
500 003052 052525 .WORD 052525 ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
501          003054
502          TBLEND==.
503          .SBTTL GLOBAL ENVIRONMENT STORAGE
504          ;
505          ;STORAGE FOR DEVICE REGISTERS
506 003054 000000 100000 000000 DUMMY: 0,100000,0,0 ;DUMMY DEVICE REGISTERS...
507 003064 000000 000000 000000 0,0,0,0,0,0,0,0,0
508          ;...FOR MULTI-UNIT CHECKOUT.
509
510 003104 000000 DUFLG:: .WORD 0 ;"DROPPED UNIT" FLAG.
511          ;INHIBITS CODE IN "CLEAN-UP".
512 003106 000000 NCDEV:: .WORD 0 ;FLAG TO SAY NO DEVICE.
513
514 003110 000000 TEMP1:: .WORD 0 ;SOME TEMP LOCATIONS.
515 003112 000000 TEMP2:: .WORD 0
516 003114 000000 XXCOMM:: .WORD 0 ;XXDP* COMM BLOCK POINTER.
517 003116 000000 FREE:: .WORD 0 ;1ST FREE MEMORY ADDRESS...
518 003120 000000 FRESIZ:: .WORD 0 ;...AND SIZE (IN WORDS).
519 003122 000000 FREEHI: .WORD 0 ;LAST WORD IN FREE SPACE
520 003124 000000 KTFLG:: .WORD 0 ;KT11, MEM AVAIL FLAG -
521          ;- .WORD 0 = <24K OR NO KT -
522          ;- NZ = >24K AND KT.
523 003126 000000 KTENABLE:: .WORD 0 ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
524 003130 000000 NXMFLG:: .WORD 0 ;SET IF WE CAN TEST CLEARED OTHERWISE
525 003132 000000 NXMLO:: .WORD 0 ;NXM LO ADDRESS BITS
526 003134 000000 NXMHI:: .WORD 0 ;NXM HI ADDRESS BITS FOR DAL'S 16-21
527 003136 000000 T23A:: .WORD 0 ;11/23A FLAG
528 003140 000000 T23B:: .WORD 0 ;11/23B FLAG
529 003142 000000 T3BFLG:: .WORD 0 ;TEST 3B FLAG †0
530 003144 002000 PST32W:: .WORD 2000 ;32W BLOCK ADDRESS FOR 32K START
531 003146 000000 SIFLAG:: .WORD 0
532 003150 000000 BADDAT:: .WORD 0 ;ACTUAL DATA
533 003152 000000 GDDAT:: .WORD 0 ;EXPECTED DATA
534 003154 000000 LOOPFL:: .WORD 0
535 003156
536 003156 000000 CTAB:: .WORD 0 ;CONFIGURATION TABLES.
537 003160 000000 CTABM:: .WORD 0 ;CONFIG WORK.
538 003162 000000 .WORD 0
539 003164 000000 .WORD 0
540 003166 177777 .WORD 0
541 003170 .WORD -1 ;END OF MEM TABLE.
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 003170          ;
          ;ERTABL: .BLKW 64.

```

GLOBAL ENVIRONMENT STORAGE

552 003370 000000
553
554 003372 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                                     ;*
565                                     ;NAMES OF DEVICES SUPPORTED
566                                     ;-
567 003374                                DEVTYP <TSV05>
    003374                                L$DVTYP::
    003374      124      123      126      .ASCIZ  #TSV05#
                                           .EVEN
568
569                                     ;*
570                                     ;TEST DESCRIPTION
571                                     ;-
572 003402                                DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - CHECK TRANSPORT IF ERROR ****>
    003402                                L$DESC::
    003402      052      052      052      .ASCIZ  /**** TSV05 LOGIC DIAGNOSTIC - CHECK TRANSPORT IF ERROR ****/
                                           .EVEN
573
574                                     ;*
575                                     ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
576                                     ;-
577
578
579 003476      003536      003541      003545      TSSRBIT::      .WORD  1$,2$,3$,4$,5$,6$,7$,8$
580 003516      003577      003603      003607      .WORD  9$,10$,11$,12$,13$,14$,15$,16$
581 003536      123      103      000      1$:      .ASCIZ  'SC'
582 003541      102      111      105      2$:      .ASCIZ  'BIE'
583 003545      123      103      105      3$:      .ASCIZ  'SCE'
584 003551      122      115      122      4$:      .ASCIZ  'RMR'
585 003555      116      130      115      5$:      .ASCIZ  'NXM'
586 003561      116      102      101      6$:      .ASCIZ  'NBA'
587 003565      102      111      124      7$:      .ASCIZ  'BIT9'
588 003572      102      111      124      8$:      .ASCIZ  'BIT8'
589 003577      123      123      122      9$:      .ASCIZ  'SSR'
590 003603      117      106      114      10$:     .ASCIZ  'OFL'
591 003607      102      111      124      11$:     .ASCIZ  'BIT5'
592 003614      102      111      124      12$:     .ASCIZ  'BIT4'
593 003621      102      111      124      13$:     .ASCIZ  'BIT3'
594 003626      102      111      124      14$:     .ASCIZ  'BIT2'
595 003633      102      111      124      15$:     .ASCIZ  'BIT1'
596 003640      102      111      124      16$:     .ASCIZ  'BIT0'
597
598                                     .EVEN
599 003646      124      123      123      SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
600 003701      124      123      123      SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
601 003734      040      040      116      NXR:      .ASCIZ  / NON-EXISTANT DEVICE REGISTER/
602 003773      045      101      040      NXR:      .ASCIZ  /#A ADDRESS: #06/
603 004014      045      101      040      TSSX:      .ASCII  /#A TSBA,TSSR EXP'D: #06#A,#06#N/
604 004054      045      101      040      TSSX:      .ASCIZ  /#A TSBA,TSSR REC'D: #06#A,#06#N/
605 004113      045      116      045      FUSI:      .ASCII  /#N#A/
606 004117      040      040      125      USI:      .ASCIZ  / UNEXPECTED INTERRUPT/
607 004146      040      040      111      NSI:      .ASCIZ  / INTERRUPT EXPECTED, NOT RECEIVED/
608 004211      045      116      045      FNOINTR:  .ASCII  /#N#A/

```

GLOBAL TEXT MESSAGES

```

628 004215    040    040    116 NOINTR: .ASCIZ / NO INTERRUPT WAS GENERATED/
629 004252    040    040    111 IFAULT: .ASCIZ / INTERRUPT FAULT/
630 004274    045    101    040 INTX: .ASCIZ /%A CPU PC: %06%A TSBA: %06/
631 004331    040    040    042 NOINIT: .ASCIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
632 004403    040    040    042 NSINIT: .ASCIZ / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
633 004453    040    040    042 BRINIT: .ASCIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
634
635 004523    000          NUL: .ASCIZ //
636 004524    045    116    000 NULCR: .ASCIZ /%N/
637 004527    045    101    040 EXPGOT: .ASCIZ /%A EXP'D: %06%A, REC'D: %06/
638 004563    045    116    045 EXPGT2: .ASCIZ /%N%A EXP'D: %06%A, %06%N%A REC'D: %0%A, %06/
639 004637    045    101    040 DUAD12: .ASCIZ /%A REG(W) WRITTEN TO: %06%A REG(R) READ; EXP'D: %06%A, REC'D: %06/
640 004741    122    101    115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
641 005007    040    040    103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
642 005052    127    122    111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
643 005107    124    123    123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
644 005202    124    123    123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
645 005274    106    101    124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
646 005366    105    122    122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
647 005454    045    116    045 NOMEM: .ASCIZ '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****N'
648 005550    045    116    045 M8186: .ASCIZ '%N%A ***** 11/23A SYSTEM *****N'
649 005641    045    116    045 M8189: .ASCIZ '%N%A ***** 11/23B SYSTEM *****N'
650
651          .EVEN
652          .SBTTL GLOBAL ERROR REPORT SECTION
653
654          ;**
655          ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
656          ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
657          ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
658          ;--
659 005732    BGNMSG NXRERR          ;NON-EXISTANT DEVICE REGISTER.
660 005732    NXRERR: PRINTX #NXRX,NODEV ;NODEV = NEXM ADDRESS.
        005732    013746 003106    MOV NODEV,-(SP)
        005736    012746 003773    MOV #NXRX,-(SP)
        005742    012746 000002    MOV #2,-(SP)
        005746    010600    MOV SP,R0
        005750    104415    TRAP C$PNTX
        005752    062706 000006    ADD #6,SP
661 005756    004737 005764    JSR PC,EXTEND ; PRINT EXTENSION IF REQUIRED.
662 005762    ENDMSG
        005762
        005762    104423    L10002: TRAP C$MSG
663
664          ;
665          ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
666          ; TO ANY OF THE ABOVE ERROR SIGNATURES.
667          ;
668 005764    005727    EXTEND: TST (PC)+
669 005766    000000    EXTA: 0 ; 0 = NO EXTENSION.
670 005770    001402    BEQ 1$
671 005772    004777 177770    JSR PC,@EXTA ; APPEND EXTENSION TEXT.
672 005776    PRINTX #NULCR ; PRINT A BLANK LINE
        005776    012746 004524    MOV #NULCR,-(SP)
        006002    012746 000001    MOV #1,-(SP)
        006006    010600    MOV SP,R0

```

L3

GLOBAL ERROR REPORT SECTION

006010 104415
006012 062706 000004
673 006016 000207

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

PRITSSR - PRINT TSSR CONTENTS

```

675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693 006020
694 006020
695 006024 010104
696 006026
    006026 010446
    006030 012746 006473
    006034 012746 000002
    006040 010600
    006042 104414
    006044 062706 000006
697 006050 010400
698 006052 004737 016124
699 006056 103410
700 006060
    006060 012746 006713
    006064 012746 000001
    006070 010600
    006072 104415
    006074 062706 000004
701 006100 010403
702 006102 042703 001476
703 006106 001434
704 006110 012702 002624
705 006114 012701 003476
706 006120 005703
707 006122 001413
708 006124 000241
709 006126 006103
710 006130 103006
711 006132 011100
712 006134 112022
713 006136 001376
714 006140 112762 000054 177777
715 006146 005721
716 006150 000763
717 006152 105042
718 006154
    006154 012746 002624
    006160 012746 006664

```

```

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

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

```


PRITSSR - PRINT TSSR CONTENTS

```

006164 012746 000002      MOV    #2,-(SP)
006170 010600      MOV    SP,R0
006172 104415      TRAP  C$PNTX
006174 062706 000006      ADD    #6,SP
719
720 006200 010403      20$:  MOV    R4,R3          ;GET THE TSSR CONTENTS
721 006202 042703 177761      BIC    #+CTERCLS,R3   ;CLEAR ALL BUT TERMINATION
722 006206 016303 006754      MOV    TCOCOD(R3),R3  ;GET THE TERMINATION CODE MEANING
723 006212      PRINTX #TCOASC,R3     ;PRINT THE TERMINATION CODE
      006212 010346      MOV    R3,-(SP)
      006214 012746 006554      MOV    #TCOASC,-(SP)
      006220 012746 000002      MOV    #2,-(SP)
      006224 010600      MOV    SP,R0
      006226 104415      TRAP  C$PNTX
      006230 062706 000006      ADD    #6,SP
724 006234 010403      MOV    R4,R3          ;TSSR CONTENTS AGAIN
725 006236 042703 177717      BIC    #+CFATERR,R3  ;CLEAR ALL BUT FATAL TERMINATION
726 006242 001416      BEQ    25$           ;DON'T PRINT IF ZERO
727 006244 006203      ASR    R3
728 006246 006203      ASR    R3
729 006250 006203      ASR    R3
730 006252 016303 007314      MOV    TSFCOD(R3),R3  ;ALINE TERMINATION CODE FOR INDEX
731 006256      PRINTX #TFCASC,R3     ;GET THE FATAL TERMINATION CODE
      006256 010346      MOV    R3,-(SP)       ;PRINT THE FATAL TERMINATION CODE
      006260 012746 006615      MOV    #TFCASC,-(SP)
      006264 012746 000002      MOV    #2,-(SP)
      006270 010600      MOV    SP,R0
      006272 104415      TRAP  C$PNTX
      006274 062706 000006      ADD    #6,SP
732 006300 042704 176377      25$:  BIC    #+CHIADDR,R4   ;CLEAR ALL BUT EXTENDED ADDRESS
733 006304 001411      BEQ    30$           ;DON'T PRINT IF ZERO
734 006306      PRINTX #TEXASC,R4     ;PRINT THE EXTENDED ADDRESS BITS
      006306 010446      MOV    R4,-(SP)
      006310 012746 006513      MOV    #TEXASC,-(SP)
      006314 012746 000002      MOV    #2,-(SP)
      006320 010600      MOV    SP,R0
      006322 104415      TRAP  C$PNTX
      006324 062706 000006      ADD    #6,SP
735 006330 013703 002172      30$:  MOV    EPRTSW,R3      ;PRINT MEASGE BUFFER ADDRESS
736 006334      PRINTX R3            ;PRINT PROPER MESSAGE
      006334 010346      MOV    R3,-(SP)
      006336 012746 000001      MOV    #1,-(SP)
      006342 010600      MOV    SP,R0
      006344 104415      TRAP  C$PNTX
      006346 062706 000004      ADD    #4,SP
737 006352 000207      RTS    PC             ;RETURN TO CALLER
738
753 006354      045      116      045  EPRT1:  .ASCIZ  'N$A *****CHECK TRANSPORT*****'
754 006413      045      116      045  EPRT2:  .ASCIZ  'N$A *****CHECK PARITY SWITCH IN TRANSPORT*****'
756 006473      045      116      045  TSSRFOR: .ASCIZ  'N$A TSSR = #06'
757 006513      045      116      045  TEXASC:  .ASCIZ  'N$A Extended Address Bits = #06'
758 006554      045      116      045  TCOASC:  .ASCIZ  'N$A Termination Class Code = #T'
759 006615      045      116      045  TFCASC:  .ASCIZ  'N$A Fatal Termination Class Code = #T'
760 006664      045      116      045  TSSDEF:  .ASCIZ  'N$A TSSR Bits Set: #T'
761 006713      045      116      045  AMBTSSR: .ASCIZ  'N$A TSSR Contents Are Ambiguous'
762
763 006754 006774 007017 007045 TCOCOD: .EVEN
      .WORD  1$,2$,3$,4$,5$,6$,7$,8$

```

PRITSSR - PRINT TSSR CONTENTS

764	006774	116	157	162	1\$:	.ASCII7	'Normal Termination'
765	007017	124	145	162	2\$:	.ASCII7	'Termination Condition'
766	007045	124	141	160	3\$:	.ASCII7	'Tape Status Alert'
767	007067	106	165	156	4\$:	.ASCII7	'Function Reject'
768	007107	122	145	143	5\$:	.ASCII7	'Recoverable Error - Tape Position One Record Down'
769	007171	122	145	143	6\$:	.ASCII7	'Recoverable Error - Tape Was Not Moved'
770	007240	125	156	162	7\$:	.ASCII7	'Unrecoverable Error'
771	007264	106	141	164	8\$:	.ASCII7	'Fatal Controller Error'
772						.EVEN	
773							
774	007314	007324	007360	007371	TSFCOD:	.WORD	1\$,2\$,3\$,4\$
775	007324	111	156	164	1\$:	.ASCII7	'Internal Diagnostic Failure'
776	007360	122	145	163	2\$:	.ASCII7	'Reserved'
777	007371	102	165	163	3\$:	.ASCII7	'Bus Interface or Sanity Check Error'
778	007435	122	145	163	4\$:	.ASCII7	'Reserved'
779						.EVEN	
780						.SBTTL	PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
781							
782							
783							;;
784							; THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
785							; THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
786							;
787							; INPUT:
788							;
789						R0	NUMBER OF WORDS IN PACKET
790						R3	HIGH ORDER COMMAND PACKET ADDRESS
791						R4	ADDRESS OF COMMAND PACKET
792							;
793							NOTE: R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
794							;-
795	007446					PRIPKT::	
796	007446					SAVREG	;SAVE THE REGISTERS
797	007452	010005				MOV R0,R5	;SAVE NO. OF WORDS IN PACKET
798	007454	005737	003126			TST KTENABLE	;ABOVE 28K UNDER TEST?
799	007460	001001				BNE 10\$;BR IF YES
800	007462	005003				CLR R3	;SET HIGH ORDER ADDRESS TO 0
801	007464	010301		10\$:		MOV R3,R1	;COPY HIGH ORDER ADDRESS
802	007466	010400				MOV R4,R0	;GET LOWER ADDRESS
803	007470	006100				ROL R0	;SHIFT BIT 15 INTO C BIT
804	007472	006101				ROL R1	;AND INTO HIGH ORDER.
805	007474					PRINTB @PKTADD,R1,R4	;PRINT PACKET ADDRESS
	007474	010446				MOV R4,-(SP)	
	007476	010146				MOV R1,-(SP)	
	007500	012746	007632			MOV @PKTADD,-(SP)	
	007504	012746	000003			MOV @3,-(SP)	
	007510	010600				MOV SP,R0	
	007512	104414				TRAP C:PNTB	
	007514	062706	000010			ADD @10,SP	
806	007520	010300		15\$:		MOV R3,R0	;GET HIGH ORDER ADDRESS
807	007522	001404				BEQ 20\$;BR IF NOT ABOVE 28K.
808	007524	010401				MOV R4,R1	;GET LOW ORDER ADDRESS
809	007526	004737	017376			JSR PC,SETMAP	;SETUP PAR6 MAPPING FOR 1@ BIT ADDRESS
810	007532	010004				MOV R0,R4	;GET RETURNED PAR6 ADDRESS BIAS
811	007534	005001		20\$:		CLR R1	;SAVE WORD NUMBER
812	007536	012402		25\$:		MOV (R4),R2	;GET PACKET CONTENTS
813	007540					PRINTB @PKTFRM,R1,R2	;PRINT THE DATA

C4

PRIPKT PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

```

007540 010246      MOV      R2,-(SP)
007542 010146      MOV      R1,-(SP)
007544 012746 007574  MOV      @PKTFRM,-(SP)
007550 012746 000003  MOV      @3,-(SP)
007554 010600      MOV      SP,R0
007556 104414      TRAP     C$PNTB
007560 062706 000010  ADD      @10,SP
814 007564 005201      INC      R1                ;NEXT WORD NUMBER
815 007566 020105      CMP      R1,R5            ;DONE ALL PACKET WORDS?
816 007570 002762      BLT     25$              ;LOOP TILL ALL DONE
817 007572 000207      RTS     PC                ;RETURN
818
819 007574      045      116      045  PKTFRM: .ASCIZ  '##N##A Packet Word ##D1##A = ##06'
820 007632      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 007670      PRIBXOR::
841 007670      SAVREG                ;SAVE THE REGISTERS
842 007674 010203  MOV      R2,R3                ;EXPECTED DATA
843 007676      XOR      R1,R3                ;FORM THE EXCLUSIVE OR
844 007706 012700 177400  MOV      @+C<377>,R0          ;BYTE MASK
845 007712 040001  BIC      R0,R1                ;SAVE LOW BYTE RECV
846 007714 040002  BIC      R0,R2                ;SAVE LOW BYTE EXPD
847 007716 040003  BIC      R0,R3                ;SAVE LOW BYTE XOR
848 007720      PRINTB @XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
      MOV      R3,-(SP)
      MOV      R1,-(SP)
      MOV      R2,-(SP)
      MOV      @XORBFOR,-(SP)
      MOV      @4,-(SP)
      MOV      SP,R0
      TRAP     C$PNTB
      ADD      @12,SP
849 007746 010300  MOV      R3,R0                ;R0 HAS XOR ON RETURN
850 007750 000207  RTS     PC                ;RETURN TO CALLER
851
852 007752      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 010020      PRIXOR::
873 010020      SAVREG          ;SAVE THE REGISTERS
874 010024      MOV      R2,R3      ;EXPECTED DATA
875 010026      XOR      R1,R3      ;FORM THE EXCLUSIVE OR
876 010036      PRINTB   @XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
877      010036      MOV      R3,-(SP)
878      010040      MOV      R1,-(SP)
879      010042      MOV      R2,-(SP)
880      010044      MOV      @XORFOR,-(SP)
881      010050      MOV      @4,-(SP)
882      010054      MOV      SP,R0
883      010056      TRAP   C$PNTB
884      010060      ADD      @12,SP
885      010064      MOV      R3,R0      ;R0 HAS XOR ON RETURN
886      010066      RTS      PC        ;RETURN TO CALLER
887
888 010070      045      116      045  XORFOR: .ASCIZ  'N#A EXPD: #06#A RECV: #06#A XOR: #06'
889      .EVEN
890      .SBTTL  PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
891
892      ;*
893      ;
894      ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
895      ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
896      ;
897      ;INPUTS:
898      ;
899      ;      R0      OCTAL VALUE TO CONVERT
900      ;      R1      TABLE OF POINTERS TO ASCII EQUIVALENT
901      ;
902      ;-
903
904 010136      PRIEQU:
905 010136      SAVREG          ;SAVE THE REGISTERS
906 010142      RTS      PC        ;RETURN TO CALLER
907
908      .SBTTL  PRIRAM - PRINT RAM ADDRESS
909
910      ;*
911      ;
912      ;PRINT CONTROLLER RAM ADDRESS.
913      ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
914

```

E4

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
          ; MOV      R4,-(SP)
          ; MOV     #RAMFOR,-(SP)
          ; MOV     #2,-(SP)
          ; MOV     SP,R0
          ; TRAP    C#PNTB
          ; ADD     #6,SP
914      ; RTS      PC ;RETURN
915
916      010174      045      116      045 RAMFOR: .ASCIZ 'N#A CONTROLLER RAM ADDRESS = #06'
917      ; .EVEN
918
919      ; .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
920      ;*
921      ;
922      ; PRINT MEMORY ADDRESS
923      ; THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
924      ;
925      ; IMPLICIT INPUTS
926      ;
927      ; ERRHI - HIGH ORDER ADDRESS
928      ; ERRLO - LOW ORDER ADDRESS
929      ;
930      ;
931      ; 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
          ; MOV     R2,-(SP)
          ; MOV     R0,-(SP)
          ; MOV     #PRIA0,-(SP)
          ; MOV     #3,-(SP)
          ; MOV     SP,R0
          ; TRAP    C#PNTB
          ; ADD     #10,SP
939      ; RTS      PC ;RETURN
940
941      010306      045      116      045 PRIA0: .ASCIZ 'N#A MEMORY ERROR ADDRESS = #01#05'
942      ; .EVEN
943
944      ; .SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
945      ;*
946      ;
947      ; PRINT MEMORY ADDRESS
948      ; THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.

```

PRITADD - PRINT MEMORY TEST ADDRESS

```

949
950
951
952
953
954
955
956 010352
957 010352
958 010356 013702 002230
959 010362 013701 002232
960
961
962
963 010366
    010366 010146
    010370 012746 010434
    010374 012746 000002
    010400 010600
    010402 104414
    010404 062706 000006
964 010410
    010410 010246
    010412 012746 010477
    010416 012746 000002
    010422 010600
    010424 104414
    010426 062706 000006
965 010432 000207
966
967 010434 045 116 045 PRIT0: .ASCIZ 'N/A MEMORY TEST ADDRESS LOW = #06'
968 010477 045 116 045 PRIT1: .ASCIZ 'N/A MEMORY TEST ADDRESS HIGH = #06'
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993

```

```

;
; IMPLICIT INPUTS
;
; ERRHI - HIGH ORDER ADDRESS
; ERRLO - LOW ORDER ADDRESS
;
; -
; PRITADD.
; SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
; MOV ERRHI,R2 ;GET HIGH ADDRESS
; MOV ERRLO,R1 ;GET LOW ADDRESS
; MOV R1,R2 ;COPY LOW ADDRESS
; ROL R1 ;SHIFT BIT 15 TO C BIT
; ROL R0 ;SHIFT INTO HIGH ORDER
; PRINTB @PRIT0,R1 ;PRINT MEMORY ADDRESS LOW IN ERROR
; MOV R1,-(SP)
; MOV @PRIT0,-(SP)
; MOV @2,-(SP)
; MOV SP,R0
; TRAP C:PNTB
; ADD @6,SP
; PRINTB @PRIT1,R2 ;PRINT MEMORY ADDRESS HIGH IN ERROR
; MOV R2,-(SP)
; MOV @PRIT1,-(SP)
; MOV @2,-(SP)
; MOV SP,R0
; TRAP C:PNTB
; ADD @6,SP
; RTS PC ;RETURN
;
; PRIT0: .ASCIZ 'N/A MEMORY TEST ADDRESS LOW = #06'
; PRIT1: .ASCIZ 'N/A MEMORY TEST ADDRESS HIGH = #06'
; .EVEN
; .SBTTL SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND
;
; *
;
; ROUTINE TO ISSUE A SPACE RECORDS
; COMMAND (FORWARD OR REVERSE)
;
; INPUT:
;
; R3 NUMBER OF RECORDS TO BE SPACED OVER
; BIT15 CONTROLS DIRECTION
; BIT15 = 0 IS FORWARD
; BIT15 = 1 IS REVERSE
;
; R5 FIRST DEVICE UNIBUS ADDRESS
;
; REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
;
; OUTPUT:
;
; CARRY SET - SPACE RECORDS COMMAND OK
; CLR - SPACE RECORDS FAILED
;
;
; R0 THE CONTENTS OF R4 IS MOVED TO R0

```

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     010544      SPACE::
1006     010544      SAVREG
1007     J10550      012737 000764 010740      MOV      #500.,SDELAY      ;SAVE THE GENERAL REGISTERS
1008     010556      012737 140010 010730      MOV      #140010,80$      ;SET UP DELAY
1009     010564      005703                      TST      R3                ;SET UP COMMAND, SPACE FORWARD
1010     010566      100403                      BMI      5$                ;CHECK FOR DIRECTION
1011     010570      010337 010732              MOV      R3,90$           ;BR, IF REVERSE INDICATED
1012     010574      000407                      BR       10$              ;LOAD UP NUMBER OF RECORDS TO SPACE
1013     010576      042703 100000              5$:     BIC      #BIT15,R3    ;GO DO COMMAND
1014     010602      010337 010732              MOV      R3,90$           ;CLEAR DIRECTION BIT
1015     010606      052737 000400 010730      10$:    BIS      #BIT8,80$   ;LOAD UP NUMBER OF RECORDS TO SPACE
1016     010614      012704 010730              MOV      #80$,R4          ;SET REVERSE BIT IN COMMAND PACKET
1017     010620      010465 000000              15$:    MOV      R4,TSDB(R5)   ;SET UP R4 WITH PACKET ADDRESS
1018     010624      004737 016330              JSR      PC,WAITF         ;SEND OUT COMMAND
1019     010630      103420                      BCS     20$               ;WAIT FOR SSR
1020     010632      012727 000250              DELAY   250              ;BR, IF SSR IS SET AND OK
1021     010636      000000                      MOV      #250,(PC)+      ;DELAY ABOUT .25 SECONDS
1022     010640      013727 002116              .WORD   0
1023     010644      000000                      MOV      L#DLY,(PC)+
1024     010646      005367 177772              .WORD   0
1025     010652      001375                      DEC     -6(PC)
1026     010654      005367 177756              BNE     .-4
1027     010660      001367                      DEC     -22(PC)
1028     010662      005337 010740              BNE     .-20
1029     010666      001356                      DEC     SDELAY
1030     010670      000411                      BNE     15$
1031     010672      016501 000002              BR      60$
1032     010676      012702 000200              20$:    MOV      TSSR(R5),R1  ;BUMP DELAY COUNTER DOWN
1033     010702      020201              MOV      #SSR,R2         ;BR, IF MORE DELAY
1034     010704      001401              25$:    CMP      R2,R1        ;BR IF TROUBLE CARRY = CLEAR
1035     010706      000402              BEQ     40$              ;READ TSSR
1036     010710      000261              BR      60$              ;SET UP EXPECTED
1037     010712      000401              40$:    SEC                ;ARE THEY OK
1038     010714      000241              BR      70$              ;BR, IF EQUAL = OK
1039     010716      010400              60$:    CLC                ;TROUBLE EXIT
1040     010720      000207              RTS     PC                ;SET CARRY NO TROUBLE
1041                                     ;EXIT
1042                                     ;CARRY CLEAR = ERROR
1043                                     ;PASS PACKET ADDRESS
1044                                     ;RETURN

```

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```

1036 ;
1037 ;
1038 ;
1039 ;PACKET FOR SPACE COMMAND
1040 ;
1042      010730      .=<.*10>&177770
1044 ;
1045 ;COMMAND WORD
1046 010730 000000 80$: .WORD
1047 ;NUMBER OF RECORDS TO BE SPACED OVER WORD
1048 010732 000000 90$: .WORD
1049 010734 000000 .WORD
1050 010736 000000 .WORD
1051 010740 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 010742 WRTCHR::
1086 010742 SAVREG
1087 010746 005037 002222 CLR BENBSW ;SAVE THE GENERAL REGISTERS
1088 010752 005037 002220 CLR EXTFEA ;CLEAR BUFFER ENABLE SWITCH
1089 010756 010465 000000 10$: MOV R4,TSDB(R5) ;CLEAR EXTENDED FEATURES SW SWITCH
1090 010762 004737 016416 JSR PC,CHKTSSR ;SEND OUT COMMAND
1091 010766 103401 BCS 20$ ;WAIT FOR SSR
1092 010770 000435 BR 60$ ;BR, IF SSR IS SET AND OK
1093 010772 016501 000002 20$: MOV TSSR(R5),R1 ;B? IF TROUBLE CARRY = CLEAR
1094 010776 012702 000200 MOV @SSR,R2 ;READ TSSR
;SET UP EXPECTED

```


WRTCHR - WRITE CHARACTERISTICS COMMAND

```

1095 011002 032701 000100 BIT #OFL,R1 ;WAS OFF LINE SET IN TSSR
1096 011006 001402 BEQ 25$ ;BR, IF NO OFL SET
1097 011010 052702 000100 BIS #OFL,R2 ;MAKE THEM LOOK ALIKE
1098 011014 020201 25$: CMP R2,R1 ;ARE THEY OK
1099 011016 001401 BEQ 40$ ;BR, IF EQUAL = OK
1100 011020 000421 BR 60$ ;TROUBLE EXIT
1101 011022 062704 000010 40$: ADD #8.,R4 ;POINT TO WRT CHARA DATA PACKET
1102 011026 011403 MOV (R4),R3 ;GET ADDRESS OF MESSAGE BUFFER
1103 011030 032763 000200 000012 BIT #X2.EXTF,XST2(R3) ;EXTENDED FEATURES BIT SET?
1104 011036 001402 BEQ 45$ ;BR IF NO
1105 011040 005237 002220 INC EXTFEA ;SET EXTENDED FEATURES SW SWITCH
1106 011044 45$:
1107 011044 032763 000100 000012 BIT #X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
1108 011052 001402 BEQ 50$ ;BR, IF SWITCH NOT SET
1109 011054 005237 002222 INC BENBSW ;SET SOFTWARE SWITCH FOR ENABLED
1110 011060 50$:
1111 011060 000261 SEC ;SET CARRY NO TROUBLE
1112 011062 000401 BR 70$ ;EXIT
1113 011064 000241 60$: CLC ;CARRY CLEAR = ERROR
1114 011066 016500 000002 70$: MOV TSSR(R5),R0 ;RETURN TSSR CONTENTS
1115 011072 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 011074 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1146 011074 MOV #RWPACK,R4 ;GET PACKET ADDRESS
1147 011100 012704 011170 MOV R4,TSDB(R5) ;SEND PACKET ADDRESS TO EXECUTE
1148 011110 012703 000550 MOV #360.,R3 ;ENOUGH TIME FOR 2400' REEL TO REWIND
1149 011114 004737 016330 10$: JSR PC,WAITF ;WAIT FOR SSR TO SET
1150 011120 103417 BCS 20$ ;LEAVE WHEN SSR IS SET
1151 011122 DELAY 250. ;WAIT FOR .25 SECONDS

```

REWIND - POSITION TAPE (REWIND) COMMAND

```

011122 012727 000372      MOV      #250.,(PC)+
011126 000000      .WORD   0
011130 013727 002116      MOV      L$DLY,(PC)+
011134 000000      .WORD   0
011136 005367 177772      DEC      -6(PC)
011142 001375      BNE      .-4
011144 005367 177756      DEC      -22(PC)
011150 001367      BNE      .-20
1152 011152 005303      DEC      R3          ;BUMP COUNTER DOWN
1153 011154 001357      BNE      10$        ;KEEP GOING
1154 011156 000241      CLC          ;CLEAR CARRY TO SET ERROR
1155 011160 010400      20$: MOV      R4,R0   ;PASS THE PACKET ADDRESS
1156 011162 000207      RTS         PC      ;RETURN
1157
1159          011170
1161 011170      RWPACK: .=<.+10>&177770
1162 011170 102010      .WORD   102010      ;POSTION COMMAND (REWIND)
1163 011172 000000      .WORD   0           ;NOT USED
1164          .SBTTL  CKRAM - COMPARE RAM TO I/O PACKET
1165
1166          ;*
1167          ;
1168          ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
1169          ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
1170          ;
1171          ;INPUT:
1172          ;
1173          ;      R4      ADDRESS OF THE COMMAND PACKET
1174          ;      R5      FIRST DEVICE UNIBUS ADDRESS
1175          ;
1176          ;OUTPUT:
1177          ;
1178          ;      CARRY   SET - RAM MATCHES PACKET
1179          ;             CLR - RAM DOES NOT MATCH PACKET
1180          ;
1181          ;IMPLICIT OUTPUT:
1182          ;
1183          ;      THE TABLE RAMDATA IS FILLED WITH THE
1184          ;      DATA HELD IN RAM.
1185          ;      RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
1186          ;
1187          ;SIDE EFFECTS:
1188          ;
1189          ;      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1190          ;
1191          ;-
1192
1193 011174      CKRAM:: SAVREG
1194 011174      MOV      #RAMDATA,R1      ;SAVE THE GENERAL REGISTERS
1195 011200 012701 002234      MOV      #RMPKTBEG,R2     ;ADDRESS TO SAVE THE RAM DATA
1196 011204 012702 000201      CLR      R3              ;BYTE ADDRESS OF FIRST RAM DATA
1197 011210 005003      JSR      PC,CHKTSSR      ;CLEAR THE ERROR FLAG
1198 011212 004737 016416      JSR      #0,TSDB(R5)     ;WAIT FOR SSR
1199 011216 112765 000000 000000      MOV      R2,TSDB(R5)     ;SET MAINTENANCE MODE
1200 011224 004737 016416      10$: JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
1201 011230 010265 000000      MOV      R2,TSDB(R5)     ;SELECT NEXT RAM ADDRESS
1202 011234 004737 016416      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET

```


CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

```

1260 011364 012737 000010 002274      MOV      #8.,RAMSIZ      ;ASSUME EXTFEA NOT SET
1261 011372 005737 002220      TST      EXTFEA        ;IS THE SOFTWARE EXTENDED FEATURES SET
1262 011376 001407      BEQ      25$           ;BR, IF NOT SET
1263 011400 012737 000012 002274      MOV      #10.,RAMSIZ   ;SET RAMSIZ FOR EXTEND FEATURES
1264 011406 020227 000200      CMP      R2,#RMCHEND   ;AT END OF EXTENDED BUFFER
1265 011412 003750      BLE     10$           ;BR, IF NOT AT END YET
1266 011414 000403      BR      27$           ;AT END BRANCH
1267 011416 020227 000176 25$:      CMP      R2,#RMCHEND-2 ;REACHED END YET ?
1268 011422 003744      BLE     10$           ;BRANCH TILL ALL READ
1269 011424 005703 27$:      TST      R3            ;WAS AN ERROR FOUND ?
1270 011426 001402      BEQ      30$           ;BRANCH IF NOT
1271 011430 000241      CLC      ;CLEAR CARRY TO SHOW ERROR
1272 011432 000401      BR      50$           ;AND EXIT
1273 011434 000261 30$:      SEC      ;SHOW GOOD COMPARE
1274 011436 000207 50$:      RTS      PC           ;RETURN
1275      .SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS
1276      ;+
1277      ;
1278      ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
1279      ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1280      ;ERROR PRINT ROUTINES.
1281      ;
1282      ;INPUT:
1283      ;
1284      ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1285      ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1286      ;      R2      EXPD MESSAGE BUFFER ADDRESS
1287      ;OUTPUT:
1288      ;
1289      ;      CARRY   SET - MESSAGE BUFFERS MATCH
1290      ;            CLR -MESSAGE BUFFERS DON'T MATCH
1291      ;
1292      ;IMPLICIT OUTPUT:
1293      ;
1294      ;      EXPMSG   BUFFER IS SET TO EXPD DATA
1295      ;      RECMMSG  BUFFER IS SET TO RECV DATA
1296      ;      RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1297      ;      RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
1298      ;
1299      ;-
1300 011440      CKMSG::
1301 011440      SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
1302 011444 010037 002276      MOV      R0,RCVHIADD  ;SAVE RECV HIGH ADDRESS
1303 011450 010137 002300      MOV      R1,RCVLOAD  ;SAVE RECV LOW ADDRESS
1304 011454 005737 003126      TST      KTENABLE    ;TESTING ABOVE 28K?
1305 011460 001403      BEQ      10$         ;BR IF NO
1306 011462 004737 017376      JSR      PC,SETMAP   ;RETURN ADDRESS BIASED TO PAR6 IN R0
1307 011466 010001      MOV      R0,R1       ;GET RETURNED ADDRESS BIASED TO PAR6
1308 011470 005004 10$:      CLR      R4          ;WORD IN BUFFER
1309 011472 005003      CLR      R3          ;CLEAR ERROR SEEN FLAG
1310 011474 010205      MOV      R2,R5       ;GET EXPD BUFFER ADDRESS
1311 011476 011264 002314 15$:      MOV      (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1312 011502 011164 002460      MOV      (R1),RECMMSG(R4) ;SAVE RECV FOR ERROR REPORT
1313 011506 022221      CMP      (R2)+,(R1)+ ;EXPD EQUAL RECV?
1314 011510 001401      BEQ      25$         ;BR IF YES
1315 011512 005203      INC      R3          ;SET ERROR SEEN FLAG
1316 011514 062704 000002 25$:      ADD      #2,R4       ;POINT TO NEXT WORD ADDRESS

```

CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

```

1317 011520 020427 000014          CMP      R4,#14          ;DONE FIRST 7 WORDS?
1318 011524 003764                BLE      15$            ;BR IF NO
1319 011526 032765 000200 000012  BIT      #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
1320 011534 001403                BEQ      50$            ;BR IF NO
1321 011536 020427 000016          CMP      R4,#16          ;DONE EXTENDED FEATURES WORD?
1322 011542 003755                BLE      15$            ;BR IF NO
1323 011544 005703                50$:    TST      R3            ;ANY ERRORS SEEN?
1324 011546 001402                BEQ      55$            ;BR IF NO
1325 011550 000241                CLC                    ;SET FAILURE
1326 011552 000401                BR       60$            ;
1327 011554 000261                55$:    SEC                    ;SET SUCCESS
1328 011556 000207                60$:    RTS      PC          ;RETURN
1329                                .SBTTL  CKMSG2  - COMPARE EXPD RECV MESSAGE BUFFERS
1330                                ;+
1331                                ;
1332                                ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
1333                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1334                                ;ERROR PRINT ROUTINES.
1335                                ;
1336                                ;INPUT:
1337                                ;
1338                                ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1339                                ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1340                                ;      R2      EXPD MESSAGE BUFFER ADDRESS
1341                                ;      R3      NUMBER OF BYTES TO COMPARE
1342                                ;
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 011560                                CKMSG2::
1357 011560                                SAVREG                    ;SAVE R1-R5 UNTIL NEXT RETURN
1358 011564 020327 000144          CMP      R3,#RECVMSG-EXPMSG;@@D IS COUNT ABOVE MAX ALLOWED?
1359 011570 003412                BLE      5$            ;@@D BR IF NO
1360 011572 012703 000144          MOV      #RECVMSG-EXPMSG,R3;@@D
1361 011576                                PRINTF  #DEBUGMSG        ;@@D
1362                                MOV      #DEBUGMSG,-(SP)
1363                                MOV      #1,-(SP)
1364                                MOV      SP,R0
1365                                TRAP     C$PNTF
1366                                ADD      #4,SP
1367 011616 010037 002276          5$:    MOV      R0,RCVHIADD   ;SAVE RECV HIGH ADDRESS
1368 011622 010137 002300          MOV      R1,RCVLOAD     ;SAVE RECV LOW ADDRESS
1369 011626 005737 003126          TST      KTENABLE      ;TESTING ABOVE 28K?
1370 011632 001403                BEQ      10$           ;BR IF NO
1371 011634 004737 017376          JSR      PC,SETMAP     ;RETURN ADDRESS BIASED TO PAR6 IN R0
1372 011640 010001                MOV      R0,R1         ;GET RETURNED ADDRESS BIASED TO PAR6
1373 011642 005004                10$:    CLR      R4          ;WORD IN BUFFER

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

1369 011644 005035          CLR      R5          ;CLEAR ERROR SEEN FLAG
1370 011646 111264 002314 15$:  MOVB    (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1371 011652 111164 002460          MOVB    (R1),RECMSG(R4) ;SAVE RECV FOR ERROR REPORT
1372 011656 122221          CMPB    (R2)*,(R1)*    ;EXPD EQUAL RECV?
1373 011660 001401          BEQ     25$          ;BR IF YES
1374 011662 005205          INC     R5          ;SET ERROR SEEN FLAG
1375 011664 062704 000001 25$:  ADD     #1,R4        ;POINT TO NEXT BYTE
1376 011670 020403          CMP     R4,R3        ;DONE ALL BYTES?
1377 011672 002001          BGE    50$          ;BR IF YES
1378 011674 000764          BR     15$          ;DO NEXT BYTE
1379 011676 005705          50$:  TST     R5          ;ANY ERRORS SEEN?
1380 011700 001402          BEQ    55$          ;BR IF NO
1381 011702 000241          CLC                    ;SET FAILURE
1382 011704 000401          BR     60$          ;
1383 011706 000261          55$:  SEC                    ;SET SUCCESS
1384 011710 000207          60$:  RTS     PC          ;RETURN
1385
1386 011712          120      122      117  DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-' ;@@D
1387 012002          045      116      045  FERCM: .ASCII /NWA ***/
1388 012013          040      040      124  ERCM: .ASCIZ / TSSR ERROR CODE REC'D = /
1389 012046          056      056      056  SIMSG: .ASCIZ /.... AFTER DOING SOFT INIT/
1390 012101          124      105      123  TINERR: .ASCIZ /TEST: .../
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407 012114          BGNMSG SFMSG
      012114          SFMSG:: JSR     PC,PRITSSR    ;PRINT CONTENTS OF TSSR REGISTER
1408 012114 004737 006020          JSR     PC,CKDROP    ;DROP UNIT, IF ALLOWED
1409 012120 004737 017262          ENDMSG
1410 012124
      012124          L10003: TRAP    C$MSG
      012124 104423
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

1423 012126          BGNMSG  PKTSSR
      012126          PKTSSR::
1424 012126 004737 006020      JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1425 012132 012700 000004      MOV      #4,R0          ;NO. OF WORDS IN PACKET
1426 012136 004737 007446      JSR      PC,PRIPKT     ;PRINT THE CONTENTS OF COMMAND PACKET
1427 012142          ENDMSG
      012142          L10004:
      012142 104423      TRAP      C#MSG

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

1440 012144          BGNMSG  PKTGETS
      012144          PKTGETS::
1441 012144 004737 006020      JSR      PC,PRITSSR     ;PRINT THE CONTENTS OF TSSR REGISTER
1442 012150 012700 000002      MOV      #2,R0          ;NO. OF WORDS IN GET STATUS PACKET
1443 012154 004737 007446      JSR      PC,PRIPKT     ;PRINT THE CONTENTS OF COMMAND PACKET
1444 012160          ENDMSG
      012160          L10005:
      012160 104423      TRAP      C#MSG

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

1455 012162          BGNMSG  SFFMSG
      012162          SFFMSG::
1456 012162 004737 006020      JSR      PC,PRITSSR     ;PRINT CONTENTS OF TSSR REGISTER
1457 012166          ENDMSG
      012166          L10006:
      012166 104423      TRAP      C#MSG

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

```

05

PKTMES - PRINT TSSR AND MESSAGE BUFFER

```

1471
1472 012170
1473 012170 004737 006020
1474 012174 010200
1475 012176 010301
1476 012200 004737 014322
1477 012204
    012204
    012204 104423
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490 012206
    012206
1491 012206 004737 010352
1492 012212 016501 000002
1493 012216 004737 006020
1494 012222
    012222
    012222 104423
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508 012224
    012224
1509 012224 012700 000007
1510 012230 005737 002220
1511 012234 001402
1512 012236 012700 000010
1513 012242 004737 014632
1514 012246
    012246
    012246 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
; ENDMMSG
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
; ERR HIGH ORDER MEMORY TEST ADDRESS
; ERR 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
; ENDMMSG
L10010:
; TRAP C#MSG
; .SBTTL MSGEXP - PRINT WRITE CHAR. EXPD-RCV MESSAGE BUFFERS
;
; *
; PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
;
; IMPLICIT INPUTS:
;
; EXPMSG - EXPECTED MESSAGE BUFFER
; RECMG - RECEIVED MESSAGE BUFFER
; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
;
;
; BGNMSG MSGEXP
MSGEXP::
; MOV #7,R0 ;ASSUME NO EXT FEATURES
; TST EXTFEA ;EXT FEATURES SET?
; BEQ 5$ ;BR IF NO
; MOV #8.,R0 ;EXT FEATURE BUFFER IS 8 WORDS
5$:
; JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
; ENDMMSG
L10011:
; TRAP C#MSG
; .SBTTL FIFEXP - PRINT FIFO EXP/RCV DATA
;
; *
; PRINT ROUTINE TO PRINT FIFO EXP/RCV DATA

```


FIFEXP PRINT FIFO EXP/RCV DATA

```

1519
1520
1521
1522
1523
1524
1525
1526
1527 012250
      012250
1528 012250 010146 012322
      012252 012746 012322
      012256 012746 000002
      012262 010600
      012264 104415
      012266 062706 000006
1529 012272
      012272 012746 012371
      012276 012746 000001
      012302 010600
      012304 104415
      012306 062706 000004
1530 012312 010100
1531 012314 004737 015202
1532 012320
      012320
      012320 104423
1533 012322 045 116
1534 012371 045 116
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549 012430
      012430
1550 012430 012701 012472
1551 012434 012100
1552 012436 001410
1553 012440
      012440 010046
      012442 012746 000001
      012446 010600
      012450 104415
      012452 062706 000004
1554 012456 000766
1555 012460 012700 000012
    
```

```

;
; R1 - BYTE COUNT
;
; IMPLICIT INPUTS:
;
; EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
; RECMG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
;
; -
; BGNMSG FIFEXP
FIFEXP::
PRINTX #FIF1MSG,R1 ;PRINT BYTES TRANSFERRED
MOV R1,-(SP)
MOV #FIF1MSG,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C:PNTX
ADD #6,SP
PRINTX #FIF2MSG ;PRINT HEADER MSG
MOV #FIF2MSG,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C:PNTX
ADD #4,SP
MOV R1,R0 ;GET BYTE COUNT
JSR PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
ENDMSG
L10012:
TRAP C:MSG
045 FIF1MSG: .ASCIZ '#N#A NUMBER OF BYTES TRANSFERRED = #D2'
045 FIF2MSG: .ASCIZ '#N#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
; RECMG - 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
    
```

E5

MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS

```

1556 012464 004737 014632          JSR    PC,PRMSGEXP      ;PRINT EXPD/RCV MESSAGE BUFFERS
1557 012470          ENDMMSG
      012470          L10013:
      012470 104423          TRAP   C$MSG
1558
1559 012472 012510 012552 012643 STATCOD:      .WORD  1$,2$,3$,4$,5$,6$,0
1560 012510          045    116    045  1$: .ASCIZ  'N$A Tape Bus Signals in Word #8:'
1561 012552          045    116    045  2$: .ASCIZ  'N$A PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
1562 012643          045    116    045  3$: .ASCIZ  'N$A IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
1563 012734          045    116    045  4$: .ASCIZ  'N$A IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
1564 013025          045    116    045  5$: .ASCIZ  'N$A Tape Bus Signals in Word #9:'
1565 013067          045    116    045  6$: .ASCIZ  'N$A DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
1566          .EVEN
1567
1568          .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
1569          ;*
1570          ;
1571          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1572          ;
1573          ;IMPLICIT INPUTS:
1574          ;
1575          ; EXPMSG - EXPECTED MESSAGE BUFFER
1576          ; RECMMSG - RECEIVED MESSAGE BUFFER
1577          ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1578          ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1579          ;-
1580 013144          BGNMSG MSGLOOP
      013144          MSGLOOP:
1581 013144 012701 013206          MOV    @LOOPCOD,R1      ;ASCII ADDRESS TABLE
1582 013150 012100          10$: MOV    (R1)+,R0      ;DONE ALL MSG LINES?
1583 013152 001410          BEQ   20$           ;BR IF YES
1584 013154          PRINTX RO      ;PRINT STATUS BIT NAMES
      013154 010046          MOV    RO,-(SP)
      013156 012746 000001          MOV    #1,-(SP)
      013162 010600          MOV    SP,R0
      013164 104415          TRAP  C$PNTX
      013166 062706 000004          ADD    #4,SP
1585 013172 000766          BR    10$           ;DO ANOTHER MSG LINE
1586 013174 012700 000012          20$: MOV    #10,R0      ;NUMBER OF WORDS IN A READ STATUS BUFFER
1587 013200 004737 014632          JSR   PC,PRMSGEXP      ;PRINT EXPD/RCV MESSAGE BUFFERS
1588 013204          ENDMMSG
      013204          L10014:
      013204 104423          TRAP  C$MSG
1589
1590 013206 013226 013301 013400 LOOPCOD:      .WORD  1$,2$,3$,4$,5$,6$,7$,0
1591 013226          045    116    045  1$: .ASCIZ  'N$A Tape Bus Loopback Signals in Word #8:'
1592 013301          045    116    045  2$: .ASCIZ  'N$A PARERR<15> IRESV2<14> IRESV1<13>'
1593 013400          045    116    045  3$: .ASCIZ  'N$A IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
1594 013477          045    116    045  4$: .ASCIZ  'N$A IWFM =>IFMK<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
1595 013576          045    116    045  5$: .ASCIZ  'N$A ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDP <04>'
1596 013675          045    116    045  6$: .ASCIZ  'N$A IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
1597 013774          045    116    045  7$: .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 014022      BGNMSG  MSGSUB
1613 014022      MSGSUB::
1614 014022 012700 000012      MOV     #10.,R0      ;SIZE OF WRITE SUBSYSTEM BUFFER
1615 014026 004737 014632      JSR     PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
1616 014032      ENDMSG
1617 014032      L10015:
1618 014032 104423      TRAP    C$MSG
1619      ;
1620      ;     .SBTTL MEMADD - PRINT MEMORY ADDRESS DATA ERROR
1621      ;+
1622      ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
1623      ;IMPLICIT INPUTS:
1624      ;
1625      ;     ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
1626      ;     ERRLO - MEMORY ERROR LOW ORDER ADDRESS
1627      ;     EXP - EXPECTED DATA
1628      ;     RECV - RECEIVED DATA
1629      ;-
1630 014034      BGNMSG  MEMADD
1631 014034      MEMADD::
1632 014034 004737 010236      JSR     PC,PRIADD   ;PRINT MEMORY ADDRESS IN ERROR
1633 014040 013701 002224      MOV     EXPD,R1    ;GET EXPD DATA
1634 014044 013702 002226      MOV     RECV,R2   ;GET RECEIVED DATA
1635 014050 004737 010020      JSR     PC,PRIXOR  ;PRINT EXPD/RCV
1636 014054      ENDMSG
1637 014054      L10016:
1638 014054 104423      TRAP    C$MSG
1639      ;
1640      ;     .SBTTL PRAMPKT - PRINT RAM AND PACKET DATA
1641      ;+
1642      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1643      ;WHEN THE RAM DATA DOES NOT MATCH.
1644      ;INPUTS:
1645      ;
1646      ;     R4      POINTER TO COMMAND PACKET
1647      ;IMPLICIT INPUTS:
1648      ;
1649      ;     RAMDATA  DATA AS READ FROM THE RAM
1650      ;     RAMSIZ  NUMBER OF BYTES IN PACKET
1651      ;              IF RAMSIZ=0 THEN DEFAULT TO 8.
1652      ;IMPLICIT OUTPUTS:
1653      ;
1654      ;     RAMSIZ  SET TO 0

```

PRAMPKT - PRINT RAM AND PACKET DATA

```

1654
1655
1656 014056
1657 014056
1658 014062 012701 002234
1659 014066 005002
1660 014070 122124
1661 014072 001005
1662 014074
1663 014104 000436
1664 014106 116105 177777
1665 014112 116403 177777
1666 014116
1667 014126 042703 177400
1668 014132 116137 177777 002226
1669 014140 116437 177777 002224
1670 014146
    014146 010346
    014150 013746 002224
    014154 013746 002226
    014160 010246
    014162 012746 014236
    014166 012746 000005
    014172 010600
    014174 104414
    014176 062706 000014
1671 014202 005202
1672 014204 005737 002274
1673 014210 001404
1674 014212 020237 002274
1675 014216 003724
1676 014220 000403
1677 014222 020227 000010
1678 014226 002720
1679 014230 005037 002274
1680 014234 000207
1681
1682 014236 045 116 045 RAMASC: .ASCIZ 'N#A BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03'
1683 .EVEN
1684 .SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
1685
1686
1687 ;*
1688 ;
1689 ;THIS ROUTINE PRINTS THE CONTENTS OF
1690 ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE
1691 ;TSV-05.
1692 ;
1693 ;INPUT:
1694 ;
1695 ; R0 LOW ORDER ADDRESS OF MESSAGE BUFFER
1696 ; R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER
1697 ; NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
1698 ;
1699 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
1700 ;
1701 PRMESS:

```

```

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

```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

1702 014322 SAVREG ;SAVE THE REGISTERS
1703 014326 010005 MOV R0,R5 ;SAVE LOW ORDER ADDRESS
1704 014330 005737 003126 TST KTENABLE ;ADDRESS ABOVE 28K?
1705 014334 001001 BNE 10$ ;BR IF YES
1706 014336 005001 CLR R1 ;SET HIGH ORDER ADDRESS TO 0
1707 014340 010103 10$: MOV R1,R3 ;SAVE HIGH ORDER ADDRESS
1708 014342 006100 ROL R0 ;SHIFT BIT15 TO C BIT
1709 014344 006101 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
1710 014346 PRINTX @PROASC,R1,R5 ;PRINT MESSAGE BUFFER ADDRESS
      014346 010546 MOV R5,-(SP)
      014350 010146 MOV R1,-(SP)
      014352 012746 014500 MOV @PROASC,-(SP)
      014356 012746 000003 MOV @3,-(SP)
      014362 010600 MOV SP,R0
      014364 104415 TRAP C$PNTX
      014366 062706 000010 ADD #10,SP
1711 014372 PRINTX @PR1ASC ;PRINT HEADER FOR CONTENTS
      014372 012746 014545 MOV @PR1ASC,-(SP)
      014376 012746 000001 MOV #1,-(SP)
      014402 010600 MOV SP,R0
      014404 104415 TRAP C$PNTX
      014406 062706 000004 ADD #4,SP
1712 014412 005004 CLR R4 ;NUMBER OF THE NEXT WORD
1713 014414 010501 MOV R5,R1 ;COPY LOW ORDER ADDRESS
1714 014416 010300 MOV R3,R0 ;COPY HIGH ORDER ADDRESS
1715 014420 001403 BEQ 20$ ;BR IF NOT ABOVE 28K
1716 014422 004737 017376 JSR PC,SETMAP ;SETUP PAR ADDRESS IN R0
1717 014426 010005 MOV R0,R5 ;GET PAR FORMAT ADDRESS ABOVE 28K
1718 014430 20$: PRINTX @PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
      014430 012546 MOV (R5)+,-(SP)
      014432 010446 MOV R4,-(SP)
      014434 012746 014603 MOV @PRASC,-(SP)
      014440 012746 000003 MOV @3,-(SP)
      014444 010600 MOV SP,R0
      014446 104415 TRAP C$PNTX
      014450 062706 000010 ADD #10,SP
1719 014454 005204 INC R4 ;NUMBER OF THE NEXT
1720 014456 020427 000007 CMP R4,#7 ;DONE ALL YET ?
1721 014462 003005 BGT 50$ ;BRANCH IF ALL DONE
1722 014464 002761 BLT 20$ ;PRINT FIRST 7 WORDS
1723 014466 032763 000200 000012 BIT @X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
1724 014474 001355 BNE 20$ ;PRINT EXTENDED STATUS WORD
1725 014476 000207 50$: RTS PC ;RETURN
1726
1727 014500 045 116 045 PROASC: .ASCIZ 'N$A Message Buffer Address = #01#05'
1728 014545 045 116 045 PR1ASC: .ASCIZ 'N$A Message Buffer Contents:'
1729 014603 045 116 045 PRASC: .ASCIZ 'N$A Word#D1$A: #0'
1730 .EVEN
1731 .SBTTL PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
1732 ;*
1733 ;
1734 ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
1735 ;
1736 ; R0 - NUMBER OF WORDS IN BUFFER
1737 ;
1738 ;IMPLICIT INPUTS:
1739 ;

```

PRMSGEXP - PRINT EXPD/RCV MESSAGE BUFFERS

```

1740 ; EXPMSG - EXPECTED MESSAGE BUFFER
1741 ; RECMMSG - RECEIVED MESSAGE BUFFER
1742 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1743 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1744 ;
1745 014632 PRMSGEXP::
1746 014632 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1747 014636 010005 MOV RO,R5 ;SAVE NUMBER OF WORDS
1748 014640 013700 002300 MOV RCVLOADD,RO ;GET RECV LOW ADDRESS
1749 014644 010004 MOV RO,R4 ;COPY LOW ADDRESS
1750 014646 013701 002276 MOV RCVHIADD,R1 ;GET RECV HIGH ADDRESS
1751 014652 006100 ROL RO ;SHIFT BIT15 TO C BIT
1752 014654 006101 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
1753 014656 PRINTX @PRMSG0,R1,R4 ;PRINT MESSAGE BUFFER ADDRESS
014656 010446 MOV R4,-(SP)
014660 010146 MOV R1,-(SP)
014662 012746 015012 MOV @PRMSG0,-(SP)
014666 012746 000003 MOV @3,-(SP)
014672 010600 MOV SP,RO
014674 104415 TRAP C#PNTX
014676 062706 000010 ADD @10,SP
1754 014702 PRINTX @PRMSG1 ;PRINT HEADER FOR CONTENTS
014702 012746 015057 MOV @PRMSG1,-(SP)
014706 012746 000001 MOV @1,-(SP)
014712 010600 MOV SP,RO
014714 104415 TRAP C#PNTX
014716 062706 000004 ADD @4,SP
1755 014722 005004 CLR R4 ;NUMBER OF THE CURRENT WORD
1756 014724 012701 002314 MOV @EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
1757 014730 012702 002460 MOV @RECMMSG,R2 ;GET RECV BUFFER ADDRESS
1758 014734 011100 20$: MOV (R1),RO ;GET EXPD
1759 014736 011203 MOV (R2),R3 ;GET RECV
1760 014740 XOR RO,R3 ;XOR EXPD/RCV
1761 014750 PRINTX @PRMSG2,R4,(R1)+,(R2)+,R3
014750 010346 MOV R3,-(SP)
014752 012246 MOV (R2)+,-(SP)
014754 012146 MOV (R1)+,-(SP)
014756 010446 MOV R4,-(SP)
014760 012746 015115 MOV @PRMSG2,-(SP)
014764 012746 000005 MOV @5,-(SP)
014770 010600 MOV SP,RO
014772 104415 TRAP C#PNTX
014774 062706 000014 ADD @14,SP
1762 015000 005204 INC R4 ;NUMBER OF THE NEXT
1763 015002 020405 CMP R4,R5 ;DONE ALL YET?
1764 015004 002001 BGE 50$ ;BR IF YES
1765 015006 000752 BR 20$ ;DO ANOTHER
1766 015010 000207 50$: RTS PC ;RETURN
1767
1768 015012 045 116 045 PRMSG0: .ASCIZ '#N#A Message Buffer Address = #01#05'
1769 015057 045 116 045 PRMSG1: .ASCIZ '#N#A Message Buffer Contents:'
1770 015115 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      ; RECMG  - RECEIVED MESSAGE BUFFER
1784      ;
1785 015202 PRBYTEXP::
1786 015202 SAVREG
1787 015206 010005 MOV R0,R5 ;SAVE R1-R5 UNTIL NEXT RETURN
1788 015210 005037 002312 CLR PRMNO ;SAVE NUMBER OF BYTES
1789 015214 005004 CLR R4 ;INIT ERROR COUNT
1790 015216 012701 002314 MOV #EXPMSG,R1 ;NUMBER OF THE CURRENT BYTE
1791 015222 012702 002460 MOV #RECMG,R2 ;GET EXPD BUFFER ADDRESS
1792 015226 111100 20$: MOVB (R1),R0 ;GET RECV BUFFER ADDRESS
1793 015230 042700 177400 BIC #C<377>,R0 ;GET EXPD BYTE
1794 015234 110037 015550 MOVB R0,PRBEXP ;CLEAR UPPER BYTE
1795 015240 111203 MOVB (R2),R3 ;SAVE FOR ERROR REPORT
1796 015242 042703 177400 BIC #C<377>,R3 ;GET RECV BYTE
1797 015246 110337 015552 MOVB R3,PRBREC ;CLEAR UPPER BYTE
1798 015252 XOR R0,R3 ;FOR ERROR REPORT
1799 015262 122122 CMPB (R1)+,(R2)+ ;XOR EXPD/RECV
1800 015264 001431 BEQ 30$ ;EXPD = RECV?
1801 015266 005237 002312 INC PRMNO ;BR IF YES
1802 015272 023727 002312 000010 CMP PRMNO,#8. ;UPDATE ERROR COUNT
1803 015300 101023 BHI 30$ ;PRINTED 8?
1804 015302 27$: PRINTX #PRBMSG,R4,PRBEXP,PRBREC,R3 ;BR IF YES
1805 015302 010346 MOV R3,-(SP)
1806 015304 013746 015552 MOV PRBREC,-(SP)
1807 015310 013746 015550 MOV PRBEXP,-(SP)
1808 015314 010446 MOV R4,-(SP)
1809 015316 012746 015416 MOV #PRBMSG,-(SP)
1810 015322 012746 000005 MOV #5,-(SP)
1811 015326 010600 MOV SP,R0
1812 015330 104415 TRAP C$PNTX
1813 015332 062706 000014 ADD #14,SP
1814 015336 FORCEXIT 50$ ;@@D
1815 015346 000404 BR 35$ ;@D
1816 30$: FORCERROR 27$,NOTSSR ;@D
1817 35$: ;@D
1818 INC R4 ;NUMBER OF THE NEXT
1819 CMP R4,R5 ;DONE ALL YET?
1820 BGE 50$ ;BR IF YES
1821 BR 20$ ;DO ANOTHER
1822 50$: PRINTX #PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
1823 MOV PRMNO,-(SP)
1824 MOV #PRBTOT,-(SP)
1825 MOV #2,-(SP)
1826 MOV SP,R0
1827 TRAP C$PNTX
1828 ADD #6,SP
1829 RTS PC ;RETURN
1815 015414 000207
1816
1817 015416 045 116 045 PRBMSG: .ASCIZ 'N%A BYTE #D2%A EXPD: #03%A RECV: #03%A XOR: #03'

```

K5

PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

1818 015503      045      116      045 PRBTOT: .ASCIZ 'N#A NUMBER OF BYTES IN ERROR = #D2'
1819                                     .EVEN
1820 015550 000000 PRBEXP: .WORD 0 ;EXPD
1821 015552 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 015554      BGNMSG EXPREC
1835 015554 004737 010020 EXPREC:: JSR PC,PRIXOR ;PRINT THE DATA
1836 015560      ENDMSG
1837 015560 104423 L10017: TRAP C$MSG
1838                                     .SBTTL EXPBREC - PRINT EXPD/RECV BYTE DATA
1839                                     ;*
1840                                     ;
1841                                     ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
1842                                     ;
1843                                     ;INPUTS:
1844                                     ;
1845                                     ;      R1      RECEIVED DATA BYTE
1846                                     ;      R2      EXPECTED DATA BYTE
1847                                     ;
1848                                     ;-
1849
1850 015562      BGNMSG EXPBREC
1851 015562 004737 007670 EXPBREC:: JSR PC,PRIBXOR ;PRINT THE DATA
1852 015566      ENDMSG
1853 015566 104423 L10020: TRAP C$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 015570
      015570
1876 015570 004737 014056
1877 015574
      015574
      015574 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 015576
      015576
1902 015576 004737 010352
1903 015602 004737 014056
1904 015606
      015606
      015606 104423
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918 015610
      015610
    
```

```

;
;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/RCV DATA

```

1919 015610 042701 177400      BIC    #1C<377>,R1      ;SAVE EXPD RAM DATA BYTE
1920 015614 042702 177400      BIC    #1C<377>,R2      ;SAVE EXPD RAM DATA BYTE
1921 015620 004737 010144      JSR    PC,PRIRAM        ;PRINT THE RAM ADDRESS
1922 015624 004737 010020      JSR    PC,PRIXOR        ;PRINT THE DATA
1923 015630      ENDMSG
      015630
      015630 104423
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937 015632      BGNMSG  TIMEXP
      015632      TIMEXP::
1938 015632      PRINTX  #TIMSGO      ;PRINT HEADER
      015632 012746 015660      MOV    #TIMSGO,-(SP)
      015636 012746 000001      MOV    #1,-(SP)
      015642 010600      MOV    SP,R0
      015644 104415      TRAP  C$PNTX
      015646 062706 000004      ADD    #4,SP
1939 015652 004737 010020      JSR    PC,PRIXOR        ;PRINT THE DATA
1940 015656      ENDMSG
      015656
      015656 104423
1941
1942 015660 045 116 045 TIMSGO: .ASCIZ 'N#A TIMER A STATUS IS IN BIT 3#N#A TIMER B STATUS IS IN BIT 2'
1943 .EVEN
1944 .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957 015760      BGNMSG  BADSSR
      015760      BADSSR::
1958 015760 010246      MOV    R2,-(SP)      ;SAVE DATA TRANSFERRED
1959 015762 042702 177400      BIC    #177400,R2      ;GET JUST ONE BYTE
1960 015766      PRINTB  #XFERASC,R2
      015766 010246      MOV    R2,-(SP)
      015770 012746 016020      MOV    #XFERASC,-(SP)
      015774 012746 000002      MOV    #2,-(SP)
      016000 010600      MOV    SP,R0

```

BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

```

016002 104414 TRAP C$PNTB
016004 062706 000006 ADD #6,SP
1961 016010 012602 MOV (SP),R2 ;RESTORE R2
1962 016012 004737 006020 JSR PC,PRITSSR ;DECODE TSSR CONTENTS
1963 016016 ENDMSG
016016 L10025:
016016 104423 TRAP C$MSG
1964 016020 045 116 045 XFERASC: .ASCIZ 'N#A Data Transferred = #03'
1965 .SBTTL GLOBAL SUBROUTINES SECTION
1966
1967 ;**
1968 ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
1969 ; THAT ARE USED IN MORE THAN ONE TEST.
1970 ;--
1971 .SBTTL SOFINIT - SOFT INITIALIZE OF CONTROLLER
1972
1973 ;*
1974 ;
1975 ;ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
1976 ;BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
1977 ;THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
1978 ;DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
1979 ;
1980 ;INPUTS:
1981 ;
1982 ; R5 ADDRESS OF FIRST REGISTER
1983 ;
1984 ;OUTPUTS:
1985 ;
1986 ; R0 CONTENTS OF TSSR, IF ERROR
1987 ; CARRY SET IF INIT WAS OKAY
1988 ; CLEAR IF FATAL ERROR
1989 ;
1990 ;CALLING SEQUENCE:
1991 ;
1992 ; MOV #ADDRESS,R5
1993 ; JSR PC,SOFINIT
1994 ; BCS CONTINUE
1995 ; ERRDF ;REPORT FATAL ERROR
1996 ;
1997 ;-
1998
1999 016054 SOFINIT::
2000 016054 SAVREG ; SAVE THE REGISTERS
2001 016060 012765 000000 000002 MOV #0,TSSR(R5) ; DO THE INIT.
2002 016066 004737 016330 JSR PC,WAITF ; WAIT FOR SSR
2003 016072 016500 000002 MOV TSSR(R5),R0 ;GET THE TSSR REGISTER
2004 016076 010004 MOV R0,R4 ;TSSR CONTENTS
2005 016100 042704 176277 BIC #C<HIADDR!OFL>,R4
2006 016104 052704 002200 BIS #SSR!NBA,R4 ;R4 HAS EXPECTED CONTENTS
2007 016110 020400 CMP R4,R0 ;ONLY EXPECTED BITS SET ?
2008 016112 001402 BEQ 5$ ;BRANCH IF OKAY
2009 016114 000241 CLC ;CLEAR THE CARRY FOR ERROR
2010 016116 000401 BR 10$ ;GO TO EXIT
2011 016120 000261 5$: SEC ;SET THE CARRY BIT
2012 016122 000207 10$: RTS PC ;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 016124
 2034 016124
 2035 016130 010004
 2036 016132 032700 100000
 2037 016136 001004
 2038 016140 032700 174077
 2039 016144 001023
 2040 016146 000424
 2041 016150 032700 000200
 2042 016154 001011
 2043 016156 032700 000040
 2044 016162 001414
 2045 016164 042704 177761
 2046 016170 020427 000016
 2047 016174 001007
 2048 016176 000410
 2049 016200 032700 000040
 2050 016204 001405
 2051 016206 032700 000006
 2052 016212 001002
 2053 016214 000241
 2054 016216 000401
 2055 016220 000261
 2056 016222 000207
 2057
 2058
 2059
 2060
 2061
 2062
 2063
 2064
 2065
 2066 000200
 2067 000001
 2068
 2069
 2070 016224 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
5%:    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
    BNE     40%         ;ERROR IF NOT SET
    BR      45%         ;OK IF ALL ARE SET
10%:   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
40%:   CLC                ;AMBIGUOUS CONTENTS
    BR      50%
45%:   SEC                ;SHOW SUCCESS - NO AMBIGUITY
50%:   RTS     PC        ;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
    
```

06

ENAIN,DSBINT - ENABLE/DISABLE INTERRUPTS

```

2071 ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2072 016225 000 INTFLAG: .BYTE 0
2073
2074 ;SAVED INTERRUPT VECTOR:
2075 016226 000000 INTVEC: .WORD 0
2076 ;SAVE CPU PC
2077 016230 000000 INTCPC: .WORD 0
2078
2079 ;SUBROUTINE TO ENABLE INTERRUPTS:
2080 016232 010046 ENAIN: MOV RO,-(SP) ;SAVE RO
2081 016234 013700 002202 MOV IVEC,RO ;GET POINTER TO VECTORS
2082 016240 012720 016276 MOV @INTR,(RO) ;SET UP INTERRUPT VECTOR
2083 016244 012720 000300 MOV @PRI06,(RO)
2084 016250 012600 MOV (SP)+,RO ;RESTORE RO
2085 016252 011646 MOV (SP),-(SP)
2086 016254 012766 000000 000002 MOV @0,2(SP) ;SET CPU TO LEVEL 0
2087 016262 00C002 RTI
2088
2089 ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 6)
2090 016264 011646 DSBINT: MOV (SP),-(SP)
2091 016266 012766 000300 000002 MOV @PRI06,2(SP)
2092 016274 000002 RTI
2093 .SBTTL INTR - INTERRUPT HANDLERS
2094
2095 016276 BGNSRV INTR ;DEFINE INTERRUPT ENTRY
016276
2096 016276 012737 000001 002216 INTR:: MOV @1,INTRECV ;SET FLAG TO SHOW INTERRUPT RECEIVED
2097 016304 105037 016225 CLR B INTFLAG ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2098 016310 132737 000001 016224 BIT B @I0KSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2099 016316 001003 BNE 1$ ;BR IF YES
2100 016320 152737 000001 016225 BIS B @I0KSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2101
2102 ;SAVE REGISTERS, MSG BUFFER, ETC.
2103 016326 1$:
2104 016326 [NDSRV
016326
L10026: RTI
016326 000002 .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
2105
2106 ;
2107 ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2108 ;
2109 ;INPUTS:
2110 ;
2111 ; R5 ADDRESS OF FIRST DEVICE REGISTER
2112 ;
2113 ;OUTPUTS:
2114 ;
2115 ; RO CONTENTS OF LAST TSSR READ
2116 ; CARRY SET - READY BIT SET
2117 ; CLR - TIMEOUT WAITING FOR READY
2118 ;
2119 016330 000401 WAITF:: BR 1$ ;NOP WHEN SUPER FIXED
2120 016332 BREAK ; DO A SUPVSR BREAK FIRST.
016332 104422 TRAP C#BRK
2121 016334 012746 011000 1$: MOV @11000,-(SP) ;25-APRIL-83 REV B - 1100 MSEC TIMER
2122 016340 016500 000002 2$: MOV TSSR(R5),RO ;READ THE TSSR REGISTER
2123 016344 105700 TST B RO ;TEST FOR READY BIT SET

```

D6

WAITF - WAIT FOR SUBSYSTEM READY

```

2124
2125 016346 100420          BMI      3$          ; EXIT ON STOP FLAG.
2126 016350          DELAY    1          ; WAIT 100 USEC
      016350 012727 000001    MOV     #1,(PC)+
      016354 000000          .WORD   0
      016356 013727 002116    MOV     L$DLY,(PC)+
      016362 000000          .WORD   0
      016364 005367 177772    DEC     -6(PC)
      016370 001375          BNE     .-4
      016372 005367 177756    DEC     -22(PC)
      016376 001367          BNE     .-20
2127 016400 005316          DEC     (SP)          ;REDUCE DELAY COUNT
2128 016402 001356          BNE     2$          ;RETRY UNTIL TIMER EXPIRES
2129 016404 000241          CLC
2130 016406 000401          BR      4$          ; C = 0, CONTROLLER STILL RUNNING...
2131 016410 000261          3$: SEC          ;...OR HUNG-UP AFTER 300 MSEC.
2132 016412 005326          4$: DEC     (SP)+    ; C = 1, CONTROLLER IS STOPPED.
2133 016414 000207          RTS     PC          ;RESTORE STACK WITHOUT CHANGING CARRY BIT
2134          .SBTTL   CHKTISSR - CHECK TSSR FOR READY
2135
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          CHKTISSR:
2154 016416 004737 016330      JSR     PC,WAITF    ;WAIT FOR READY
2155 016422 103014          BCC     20$          ;BRANCH IF TIME OUT
2156 016424 004737 016124      JSR     PC,CHKAMB   ;TSSR AMBIGUOUS?
2157 016430 103006          BCC     10$          ;BR IF YES
2158 016432 032700 100000      BIT     #SC,R0      ;SPECIAL CONDITION SET?
2159 016436 001405          BEQ     15$          ;BR IF NO
2160 016440 032700 074000      BIT     #<SCE!BIE!RMR!NXM>,R0 ;ANY ERROR BITS SET?
2161 016444 001402          BEQ     15$          ;BR IF NO
2162 016446 000241          10$: CLC          ;SET FAILURE
2163 016450 000401          BR      20$
2164 016452 000261          15$: SEC          ;SET SUCCESS
2165 016454 000207          20$: RTS     PC          ;RETURN TO CALLER
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 016456 012737 016510 000004 XNXM:  MOV   #2$,@#4      ; SET BUSERR VECTOR.
2178 016464 012737 000200 000006      MOV   #PRI04,@#6
2179 016472 005003      CLR   R3          ;FLAG.
2180 016474 005711      1$:   TST   (R1)      ;TEST THE ADDRESS(ES).
2181      :                      ;IF ANY TRAP, CONTINUE AT 2$.
2182 016476 020102      :                      ;OTHERWISE, CONTINUE HERE.
2183 016500 001407      :                      ;BR IF FINISHED (NO NEXM'S).
2184 016502 062701 000002      ADD   #2,R1      ;SET NEXT ADDRESS...
2185 016506 000772      BR    1$        ;...AND CONTINUE.
2186      :
2187 016510 005103      2$:   COM   R3          ;GOT ONE, SET FLAG...
2188 016512 012716 016520      MOV   #3$,(SP)
2189 016516 000002      RTI          ;...AND DISMISS INTERRUPT...
2190 016520      3$:   CLRVEC #4      ;...AND GIVE BACK THE VECTOR.
      016520 012700 000004      MOV   #4,R0
      016524 104436      TRAP  C#CVEC
2191 016526 005703      TST   R3          ;DID WE CATCH ONE ??
2192 016530 001401      BEQ   .+4        ;NO, "C" = 0, SKIP NEXT.
2193 016532 000261      SEC          ;YES, "C" = 1, (R1) = NEXM ADDR.
2194 016534 000207      RTS   PC
2195
2196
2197
2198      .SBTTL TSTLOOP - CHECK ITERATION COUNT
2199      ;*
2200      ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
2201      ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
2202      ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
2203      ;
2204      ; CALL: LOOPTO ARG
2205      ;
2206 016536      TSTLOOP:;
2207 016536 005737 002162      TST   NOITS      ; ITERATIONS INHIBITED?
2208 016542 001006      BNE   1$        ; YES.
2209 016544 005737 002176      TST   QVP        ; NO.
2210 016550 100403      BMI   1$        ;LOOPS DISALLOWED IN QUICK PASS.
2211 016552 005337 002210      DEC   LOOPCNT   ; BUMP LOOP COUNTER.
2212 016556 001002      BNE   2$        ;
2213 016560 000241      1$:   CLC          ;LOOP DISALLOWED, OR DONE.
2214 016562 000401      BR    3$
2215 016564 000261      2$:   SEC          ;LOOP ENABLED.
2216 016566 000207      3$:   RTS   PC

```

```

2195
2196
2197
2198      .SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
2199      ;*
2200      ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
2201      ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
2202      ; IN THE CURRENT RUN SEQUENCE.
2203      ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
2204      ;
2205      ; INPUT:
2206      ;
2207      ; R0      POINTER TO TEST ID ASCIZ STRING
2208      ;
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227

```

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      TSTSETUP::
2244      016570 016046      MOV      RO,-(SP)      ;SAVE THE TEST ID MESSAGE
2245      016572 005037 003146  CLR      SIFLAG      ; CLEAR "SOFT INIT" FLAG
2246      016576 005037 017036  CLR      ERRK        ; CLEAR LOCAL ERROR COUNTER.
2247      016602 005037 005766  CLR      EXTA        ; CLEAR ERROR EXTENSION FLAG.
2248      016606 105037 016224  CLRB    INTMASK     ; CLEAR INTERRUPT MASK (CHECK ERROR)
2249      016612 013700 002174  MOV      UNITN,RO    ; GET THE UNIT NUMBER,
2250      016616 006300      ASL      RO          ; ... AND MAKE IT A WORD OFFSET.
2251      016620 005737 003106  TST     NODEV       ; DID STARTUP FIND THE DEVICE?
2252      016624 001430      BEQ     4$          ; BR IF YES
2253      016626 100010      BPL     3$          ; BR IF NOT IDLE
2254      016630 052760 160000 003170  BIS     #160000,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
2255      016636      ERRDF  1,NXR,NXRERR ; NO DEVICE HERE -- PRINT IT
2256      016636 104455      TRAP   C$ERDF
2257      016640 000001      .WORD  1
2258      016642 003734      .WORD  NXR
2259      016644 005732      .WORD  NXRERR
2260      016646 000407      BR     2$
2261      016650 052760 160001 003170 3$:  BIS     #160001,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
2262      016656      ERRDF  2,NOINIT ; DEVICE NOT IDLE
2263      016656 104455      TRAP   C$ERDF
2264      016660 000002      .WORD  2
2265      016662 004331      .WORD  NOINIT
2266      016664 000000      .WORD  0
2267      016666 012737 177777 003104 2$:  MOV     #-1,DUFLG    ; DROP THE UNIT
2268      016674      DODU   UNITN
2269      016674 013700 002174  MOV     UNITN,RO
2270      016700 104451      TRAP   C$DODU
2271      016702      DOCLN      ; ABORT THE PASS
2272      016702 104444      TRAP   C$DCLN
2273      016704 000423      BR     5$
2274
2275      4$:  RFLAGS  RO          ; GET THE OPERATOR FLAGS.
2276      016706 104421      TRAP   C$RFLA
2277      016710 032700 001000  BIT     #PNT,RO      ; PRINT THE TEST NUMBERS?
2278      016714 001412      BEQ     1$          ; BR IF NO
2279      016716 011600      MOV     (SP),RO     ;GET THE ID MESSAGE
2280      016720      PRINTF #TNAM,RO ;DISPLAY THE TEST ID
2281      016720 010046      MOV     RO,-(SP)
2282      016722 012746 016764  MOV     #TNAM,-(SP)
2283      016726 012746 000002  MOV     #2,-(SP)
2284      016732 010600      MOV     SP,RO

```


TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```

016734 104417 TRAP C$PNTF
016736 062706 000006 ADD #6,SP
2269 016742 005237 002206 1$: INC TSTCNT ; BUMP TEST COUNTER.
2270 016746 SETPRI IPRI ; PRIORITY THAT OF DEVICE
016746 013700 002204 MOV IPRI,RO
016752 104441 TRAP C$SPRI
2271 016754 005726 5$: TST (SP)+ ; FIX UP THE STACK
2272 016756 013705 002200 MOV CSRADDR,R5 ; ADDRESS OF TSV REGISTERS ON UNIBUS
2273 016762 000207 RTS PC
2274 016764 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 017000 TSTEND: RFLAGS RO
017000 104421 TRAP C$RFLA
2282 017002 030027 020000 BIT RO,#IER
2283 017006 001412 BEQ 1$ ; BR IF "IER" NOT SET.
2284 017010 PRINTF #ESUM,ERRK ; PRINT ERROR COUNT.
017010 013746 017036 MOV ERRK,-(SP)
017014 012746 017040 MOV #ESUM,-(SP)
017020 012746 000002 MOV #2,-(SP)
017024 010600 MOV SP,RO
017026 104417 TRAP C$PNTF
017030 062706 000006 ADD #6,SP
2285 017034 000207 1$: RTS PC
2286 ;
2287 017036 000000 ERRK: 0 ; LOCAL ERROR COUNT.
2288 017040 045 101 040 ESUM: .ASCIZ /#A #D#A ERRORS/
2289 017057 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 017124 005237 017036 INCERK: INC ERRK ; INCREMENT LOCAL ERROR COUNT
2297 017130 010046 MOV RO,-(SP) ; SAVE RO
2298 017132 013700 002174 MOV UNITN,RO ; GET UNIT NUMBER
2299 017136 006300 ASL RO ; ... AND MAKE IT A WORD OFFSET.
2300 017140 062700 003170 ADD #ERTABL,RO ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
2301 017144 005210 INC (RO) ; INCREMENT THE DEVICE ERROR COUNT
2302 017146 032710 007777 BIT #7777,(RO) ; DID WE OVERFLOW THE FIELD?
2303 017152 001001 BNE 1$ ; BR IF NO.
2304 017154 005310 DEC (RO) ; YES -- BACK IT UP TO 7777.
2305 017156 012600 1$: MOV (SP)+,RO ; RESTORE RO
2306 017160 000207 RTS PC ; RETURN TO CALLER.
2307 ;
2308 017162 010046 CKEMAX: MOV RO,-(SP) ; SAVE RO
2309 017164 013700 002174 MOV UNITN,RO ; GET UNIT NUMBER
2310 017170 006300 ASL RO ; ... AND MAKE IT A WORD OFFSET
2311 017172 016000 003170 MOV ERTABL(RO),RO ; GET ERROR TABLE ENTRY
2312 017176 042700 170000 BIC #170000,RO ; EXTRACT ERROR COUNT FIELD
2313 017202 020037 002166 CMP RO,GERRMAX ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
2314 017206 103004 BHIS 1$ ; BR IF YES

```

INCERK - INCREMENT LOCAL ERROR COUNT

```

2315 017210 023737 017036 002164      CMP      ERRK,LERRMAX      ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2316 017216 103417                      BLO      2$                ; BR IF NO
2317 017220                      1$:    RFLAGS   RO                ; GET OPERATOR FLAGS
      017220 104421                      TRAP    C$RFLA
2318 017222 032700 000040              BIT      @IDU,RO          ; IS DROPPING INHIBITED?
2319 017226 001013                      BNE     2$                ; BR IF YES.
2320 017230 012737 177777 003104      MOV     #-1,DUFLG        ; NO -- DROP THE UNIT
2321 017236                      ERRDF   4,EMAXDU
      017236 104455                      TRAP    C$ERDF
      017240 000004                      .WORD   4
      017242 017057                      .WORD   EMAXDU
      017244 000000                      .WORD   0
2322 017246                      DODU    UNITN
      017246 013700 002174              MOV     UNITN,RO
      017252 104451                      TRAP    C$DODU
2323 017254                      DOCLN
      017254 104444                      TRAP    C$DCLN
2324 017256 012600                      2$:    MOV     (SP)+,RO      ; RESTORE RO
2325 017260 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 017262 010046                      CKDROP: MOV    RO,-(SP)
2331 017264                      FORCERROR 1$,NOTSSR
2332 017274                      RFLAGS   RO
      017274 104421                      TRAP    C$RFLA
2333 017276 032700 000040              BIT      @IDU,RO
2334 017302 001010                      BNE     1$
2335 017304 011600                      MOV     (SP),RO
2336 017306 012737 177777 003104      MOV     #-1,DUFLG
2337 017314                      DODU    UNITN
      017314 013700 002174              MOV     UNITN,RO
      017320 104451                      TRAP    C$DODU
2338 017322                      DOCLN                                ;ABORT THE PASS
      017322 104444                      TRAP    C$DCLN
2339 017324 012600                      1$:    MOV     (SP)+,RO
2340 017326 000207                      RTS     PC
2341
2342
2343                      .SBTTL  CONFIG - DETERMINE CONFIGURATION OF SYSTEM
2344                      ;
2345                      ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
2346                      ;
2347 017330                      CONFIG:
2348 017330 004737 016054              JSR     PC,SOFINIT
2349 017334 000207                      RTS     PC
2350                      .SBTTL  KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
2351                      ;
2352                      ; SUBROUTINE - ENABLE MEM MGT.
2353                      ;
2354 017336 005737 003124              KTON:   TST    KTFLG      ; GOT KT?
2355 017342 001403                      BEQ     1$                ; NO.
2356 017344 012737 000001 177572      MOV     @1,SRO          ; YES. ENABLE KT11.
2357 017352 000207                      1$:    RTS     PC
2358
2359                      ;

```

KTON,KT0FF - ENABLE/DISABLE MEMORY MANAGEMENT

```

2360 ; SUBROUTINE - DISABLE MEM MGT.
2361 ;
2362 017354 005737 003124 KT0FF: TST KIFLG ; GOT KT11?
2363 017360 001405 BEQ 1$ ; NO.
2364 017362 000240 NOP
2365 017364 000240 NOP
2366 017366 012737 000000 177572 MOV #0,SRO ; DISABLE KT.
2367 017374 000207 1$: RTS PC
2368 ; SBTTL SETMAP - SETUP PAR6 MAPPING
2369 ;
2370 ;*
2371 ;
2372 ; THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
2373 ; AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
2374 ; IS RETURNED BIASED TO PAR6.
2375 ;
2376 ; INPUTS:
2377 ;
2378 ; R0 HIGH ORDER ADDRESS BITS
2379 ; R1 LOW ORDER ADDRESS BITS
2380 ;
2381 ; OUTPUTS:
2382 ;
2383 ; R0 OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
2384 ; CARRY SET IF SUCCESS
2385 ; CLR IF ERROR
2386 ;
2387 ; -
2387 017376 SETMAP:
2388 017376 SAVREG ; SAVE R1-R4 UNTIL NEXT RETURN
2389 017402 005737 003124 TST KIFLG ; SYSTEM HAVE ABOVE 28K?
2390 017406 001433 BEQ 10$ ; BR IF NO
2391 017410 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 017442 042701 000177 BIC #177,R1 ; ALINE FOR LOWER 4K BOUNDARY
2397 017446 020137 003124 CMP R1,KIFLG ; HIGHER THAN EXISTING MEMORY?
2398 017452 103011 BHIS 10$ ; BR IF YES
2399 017454 010137 172354 MOV R1,#KIPAR6 ; SETUP MAPPING REGISTER PAR6
2400 017460 042702 160000 BIC #160000,R2 ; SETUP DISPLACEMENT IN PAGE
2401 017464 062702 140000 ADD #140000,R2 ; ADD IN PAR6 BIAS
2402 017470 010200 MOV R2,R0 ; RETURN IN R0
2403 017472 000261 SEC ; SET SUCCESS
2404 017474 000401 BR 15$ ;
2405 017476 000241 10$: CLC ; SET FAILURE
2406 017500 000207 15$: 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 017502 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2424 017506 JSR PC,KTOFF ;DISABLE KT.
2425 017512 004737 017354 MOV R0,R3 ;COPY TEST PATTERN
2426 017514 013701 003116 MOV FREE,R1 ;GET FIRST FREE LOCATION
2427 017520 013702 003120 MOV FRESIZ,R2 ;SIZE OF FREE SPACE BELOW 28K.
2428 017524 010321 10$: MOV R3,(R1)+ ;STORE A BACKGROUND WORD
2429 017526 005302 DEC R2 ;DONE ALL MEMORY IN FREE SPACE?
2430 017530 003375 BGT 10$ ;BR IF NO
2431 017532 005737 003124 TST KTFLG ; GOT KT?
2432 017536 001477 BEQ 55$ ; NO. GET OUT.
2433 017540 004737 017336 JSR PC,KTON ; YES. ENABLE KT.
2434 017544 005000 CLR R0 ;HIGH ORDER ADDRESS START
2435 017546 013701 003144 MOV PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
2436 000006 .REPT 6
2437 CLC ;CLEAR C BIT
2438 ROL R1 ;CONVERT BLOCKS TO WORDS
2439 ROL R0 ;MAKE IT DOUBLE PRECISION
2440 .ENDR
2441 017616 004737 017376 JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
2442 017622 010320 30$: MOV R3,(R0)+ ;STORE TEST PATTERN IN >28K ADDRESS
2443 017624 020027 160000 CMP R0,#160000 ;END OF PAR6 MAPPING AREA?
2444 017630 103774 BLO 30$ ;BR IF NO
2445 017632 162700 020000 SUB #20000,R0 ;BACKUP INTO PAR6 MAPPING BEGIN
2446 017636 062737 000200 172354 ADD #200,#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2447 017644 023737 172354 003124 CMP #KIPAR6,KTFLG ;END OF MEMORY?
2448 017652 001427 BEQ 50$ ;BR IF YES
2449 017654 005737 003136 TST T23A ;11/23A?
2450 017660 001407 BEQ 35$ ;NO KEEP GOING
2451 017662 013704 177572 MOV SRO,R4 ;GET SRO CONTENTS
2452 017666 042704 177761 BIC #177761,R4 ;CLEAR ALL BUT PAGE NUMBER
2453 017672 022704 000016 CMP #16,R4 ;SEE IF PAGE 7
2454 017676 001415 BEQ 50$ ;EXIT IF THERE
2455 017700 005737 003140 35$: TST T23B ;11/23B?
2456 017704 001410 BEQ 45$ ;NO KEEP GOING
2457 017706 023727 172354 007600 CMP #KIPAR6,#7600 ;REACHED 18 BITS?
2458 017714 103001 BHIS 40$ ;YES
2459 017716 000403 BR 45$ ;NO KEEP GOING
2460 017720 012737 000020 172516 40$: MOV #20,SR3 ;SET 22 BIT RELOCATION
2461 017726 000137 017622 45$: JMP 30$ ;KEEP GOING ON ETC.
2462 017732 004737 017354 50$: JSR PC,KTOFF ;DISABLE KT.
2463 017736 000207 55$: 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 ; KTFLG = 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 017740 CMPMEM:
2487 017740 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2488 017744 010003 MOV R0,R3 ;COPY TEST PATTERN
2489 017746 004737 017354 JSR PC,KTOFF ;DISABLE KT.
2490 017752 013701 003116 MOV FREE,R1 ;GET FIRST FREE LOCATION
2491 017756 013702 003120 MOV FRESIZ,R2 ;SIZE OF FREE SPACE BELOW 28K.
2492 017762 020311 10$: CMP R3,(R1) ;FREE SPACE LOCATION EQUAL TO EXPD?
2493 017764 001411 BEQ 15$ ;BR IF YES
2494 017766 010137 002232 MOV R1,ERRLO ;SAVE ADDRESS IN ERROR
2495 017772 005037 002230 CLR ERRHI ;NO HIGH ADDRESS
2496 017776 010337 002224 MOV R3,EXPD ;SAVE EXPD FOR ERROR REPORT
2497 020002 011137 002226 MOV (R1),RECV ;SAVE RECV FOR ERROR REPORT
2498 020006 000474 BR 50$ ;
2499 020010 005721 15$: TST (R1)+ ;POINT TO NEXT ADDRESS
2500 020012 005302 DEC R2 ;DONE ALL MEMORY IN FREE SPACE?
2501 020014 003362 BGT 10$ ;BR IF NO
2502 020016 005737 003124 TST KTF LG ; GOT KT?
2503 020022 001472 BEQ 55$ ; NO. GET OUT.
2504 020024 004737 017336 JSR PC,KTON ; YES. ENABLE KT.
2505 020030 005000 CLR R0 ;HIGH ORDER ADDRESS START
2506 020032 013701 003144 MOV PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
2507 .REPT 6
2508 ROL R1 ;CONVERT BLOCKS TO WORDS
2509 ROL R0 ;MAKE IT DOUBLE PRECISION
2510 .ENDR
2511 020066 042701 000177 BIC #177,R1 ;ALINE 4K BOUNDARY
2512 020072 010046 MOV R0,-(SP) ;SAVE HIGH ORDER
2513 020074 010146 MOV R1,-(SP) ;SAVE LOW ORDER
2514 020076 004737 017376 JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
2515 020102 010004 MOV R0,R4 ;COPY ADDRESS BIASED TO PAR6
2516 020104 012601 MOV (SP)+,R1 ;RESTORE LOW ORDER IN NON PAR6 FORMAT
2517 020106 012600 MOV (SP)+,R0 ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
2518 020110 020314 30$: CMP R3,(R4) ;ABOVE 28K LOCATION EQUAL EXPD?
2519 020112 001411 BEQ 32$ ;BR IF YES
2520 020114 010037 002230 MOV R0,ERRHI ;SAVE HIGH ORDER IN ERROR
2521 020120 010137 002232 MOV R1,ERRLO ;SAVE LOW ORDER IN ERROR
2522 020124 010337 002224 MOV R3,EXPD ;SAVE EXPD FOR ERROR REPORT
2523 020130 011437 002226 MOV (R4),RECV ;SAVE RECV FOR ERROR REPORT
2524 020134 000421 BR 50$ ;
2525 020136 062701 000002 32$: ADD #2,R1 ;UPDATE NON PAR6 ADDRESS
2526 020142 005500 ADC R0 ;MAKE IT DOUBLE PRECISION ADD
2527 020144 062704 000002 ADD #2,R4 ;UPDATE PAR FORMAT ADDRESS
2528 020150 020427 160000 CMP R4,#160000 ;END OF PAR6 MAPPING AREA?
2529 020154 103755 BLO 30$ ;BR IF NO
2530 020156 162704 020000 SUB #20000,R4 ;BACKUP INTO PAR6 MAPPING BEGIN

```

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```

2531 020162 062737 000200 172354      ADD    #200,@#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2532 020170 023737 172354 003124      CMP    @#KIPAR6,KTFLG ;END OF MEMORY?
2533 020176 101744                      BLOS  30$             ;BR IF NO
2534 020200 004737 017354      50$:  JSR    PC,KTOFF   ;TURN OFF MEMORY MAPPING
2535 020204 000241                      CLC                      ;SET FAILURE
2536 020206 000403                      BR     60$             ;
2537 020210 004737 017354      55$:  JSR    PC,KTOFF   ;TURN OFF MEMORY MAPPING
2538 020214 000261                      SEC                      ;SET SUCCESS
2539 020216 000207      60$:  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 020220                      REGSAV:
2561 020220 010446                      MOV    R4,-(SP)
2562 020222 010346                      MOV    R3,-(SP)
2563 020224 010246                      MOV    R2,-(SP)
2564 020226 010146                      MOV    R1,-(SP)
2565 020230 010546                      MOV    R5,-(SP)
2566 020232 016605 000012          MOV    10.(SP),R5
2567 020236 004736                      JSR    PC,@(SP)+
2568 020240 012601                      MOV    (SP)+,R1
2569 020242 012602                      MOV    (SP)+,R2
2570 020244 012603                      MOV    (SP)+,R3
2571 020246 012604                      MOV    (SP)+,R4
2572 020250 012605                      MOV    (SP)+,R5
2573 020252 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 ; JSR PC,GETPAT
2590 ;
2591 ; -
2592 ;
2593 020254 GETPAT::
2594 020254 SAVREG ;SAVE THE GENERAL REGISTERS
2595 020260 1$: GMANID DATASC,PATDAT,0,377,0,377,NO
          020260 104443 TRAP C$GMAN
          020262 000406 BR 10000$
          020264 020310 .WORD PATDAT
          020266 000022 .WORD T$CODE
          020270 020312 .WORD DATASC
          020272 000377 .WORD 377
          020274 000000 .WORD T$LOLIM
          020276 000377 .WORD T$HILIM
          020300 10000$:
2596 020300 BNCOMPLETE 1$ ;RETRY IF ERROR
          020300 103367 BCC 1$
2597 020302 013700 020310 MOV PATDAT,R0 ;DATA PATTERN FROM OPERATOR
2598 020306 000207 RTS PC ;RETURN TO CALLER
2599 ;
2600 ;+
2601 ;LOCAL DATA AREA
2602 ; -
2603 ;
2604 020310 000000 PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
2605 020312 105 116 124 DATASC: .ASCIZ 'ENTER DATA PATTERN'
2606 .EVEN
2607 .SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
2608 ;+
2609 ;ROUTINE TO ISSUE A MENU AND GET
2610 ;THE OPERATOR'S RESPONSE.
2611 ;
2612 ;INPUTS:
2613 ;
2614 ; R0 ADDRESS OF ASCIZ STRING OF MENU
2615 ; R1 MAXIMUM ALLOWABLE OPERATOR RESPONSE
2616 ;
2617 ;OUTPUTS:
2618 ;
2619 ; R0 NUMBER OF THE OPERATOR'S SELECTION
2620 ;
2621 ; -
2622 ;
2623 ;
2624 020336 GETSEL::
2625 020336 SAVREG ;SAVE GENERAL REGISTERS
2626 020342 010002 MOV R0,R2 ;SAVE THE MENU ADDRESS
2627 020344 010203 MOV R2,R3 ;START OF MENU STRING
2628 020346 005713 1$: TST (R3) ;END OF ASCII ?
2629 020350 001412 2$: BEQ 3$ ;BRANCH IF ALL LINES DISPLAYED
2630 020352 PRINTF #SELASC,(R3)+ ;DISPLAY THE MENU
          020352 012346 MOV (R3)+,-(SP)
          020354 012746 020522 MOV #SELASC,-(SP)
          020360 012746 000002 MOV #2,-(SP)
          020364 010600 MOV SP,R0

```

GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

```

020366 104417 TRAP C$PNTF
020370 062706 000006 ADD #6,SP
2631 020374 000764 BR 2$
2632 020376 3$: GMANID MENASC,MENRES,D,-1,0,-1,NO
020376 104443 TRAP C$GMAN
020400 000406 BR 10001$
020402 020556 .WORD MENRES
020404 000042 .WORD T$CODE
020406 020527 .WORD MENASC
020410 177777 .WORD -1
020412 000000 .WORD T$LOLIM
020414 177777 .WORD T$HILIM
020416 10001$:
2633 020416 BNCOMPLETE 1$ ;RETRY IF ERROR
020416 103352 BCC 1$
2634 020420 013700 020556 MOV MENRES,R0 ;GET THE OPERATOR'S REPLY
2635 020424 020001 CMP R0,R1 ;COMPARE TO MAXIMUM ALLOWED
2636 020426 101411 BLOS 5$ ;BRANCH IF OK
2637 020430 PRINTF #MENERR ;DISPLAY ERROR MESSAGE
020430 012746 020454 MOV #MENERR,-(SP)
020434 012746 000001 MOV #1,-(SP)
020440 010600 MOV SP,R0
020442 104417 TRAP C$PNTF
020444 062706 000004 ADD #4,SP
2638 020450 000735 BR 1$ ;RETRY
2639 020452 000207 RTS PC ;RETURN TO CALLER
2640 020454 045 116 045 MENERR: .ASCIZ '#N%#A *** Menu Selection Too Large ***'
2641 020522 045 116 045 SELASC: .ASCIZ '#N%#T'
2642 020527 105 156 164 MENASC: .ASCIZ 'Enter Menu Selection: '
2643 .EVEN
2644 020556 000000 MENRES: .WORD 0
2645 .SBTTL CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
2646 ;*
2647 ;
2648 ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2649 ;
2650 ;INPUT:
2651 ;
2652 ; NONE.
2653 ;
2654 ;OUTPUT:
2655 ;
2656 ; CARRY 0 MANUAL INTERVENTION NOT ALLOWED
2657 ; 1 MANUAL INTERVENTION IS OK
2658 ;
2659 ;SIDE EFFECTS:
2660 ;
2661 ; A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2662 ; NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2663 ; ALLOWED.
2664 ;
2665 ;-
2666 ;
2667 020560 CHKMAN:: SAVREG ;SAVE THE REGISTERS
2668 020560 MANUAL ;SEE IF MANUAL INTERVENTION OK
2669 020564 TRAP C$MANI
020564 104450

```


CHKMAN - CHECK MANUAL INTERVENTION LEGALITY

```

2670 020566          BCOMPLETE 1$          ;BRANCH IF ALLOWED
      020566 103411  BCS 1$
2671 020570          PRINTF @NOMAN          ;PRINT THE WARNING MESSAGE
      020570 012746 020614  MOV @NOMAN,-(SP)
      020574 012746 000001  MOV @1,-(SP)
      020600 010600  MOV SP,R0
      020602 104417  TRAP C$PNTF
      020604 062706 000004  ADD @4,SP
2672 020610 000241  CLC          ;CLEAR CARRY FOR ERROR
2673 020612 000207  1$: RTS PC          ;RETURN
2674
2675 020614 045 116 045 NOMAN: .ASCII 'NWA *** 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 020710 ENVIRN: MEMORY R0
      020710 104431  TRAP C$MEM
2682 020712 010037 003116  MOV R0,FREE          ; GET 1ST FREE ADDRESS...
2683 020716 062737 000002 003116  ADD @2,FREE
2684 020724 011037 003120  MOV (R0),FRESIZ      ;...AND WORD COUNT.
2685 020730 162737 000004 003120  SUB @4,FRESIZ
2686 020736 013702 002012  MOV L$UNIT,R2          ; GET NUMBER OF UNITS
2687 020742 162737 000007 003120 10$: SUB @7,FRESIZ      ; TAKE AWAY 7 WORDS PER UNIT
2688 020750 005302  DEC R2
2689 020752 001373  BNE 10$
2690 020754 013700 003116  MOV FREE,R0          ;GET FIRST FREE ADDRESS
2691 020760 063700 003120  ADD FRESIZ,R0        ;POINT TO LAST FREE ADDRESS
2692 020764 162700 000002  SUB @2,R0            ;BACKUP 1 WORD
2693 020770 010037 003122  MOV R0,FREEHI        ;STORE LAST FREE ADDRESS
2694 020774 000240  NOP
2695 020776 012701 177520  MOV @BDVPCR,R1        ;GET BDV11 PCR ADDRESS
2696 021002 010102  MOV R1,R2            ;COPY TO R2
2697 021004 062702 000002  ADD @2,R2            ;SET THE RANGE
2698 021010 004737 016456  JSR PC,XNXM          ;SEE IF WE HAVE ONE
2699 021014 103001  BCC 15$            ;OK TO SET FLAGS
2700 021016 000445  BR 40$             ;RETURN WITH FLAGS CLEAR
2701 021020 013701 177520  15$: MOV BDVPCR,R1        ;SAVE PCR CONTENTS
2702 021024 062701 000001  ADD @1,R1            ;ADD ONE TO IT
2703 021030 012702 177520  MOV @BDVPCR,R2        ;GET BDV11 PCR ADDRESS
2704 021034 005212  INC (R2)            ;TRY TO WRITE TO IT
2705 021036 013703 177520  MOV BDVPCR,R3        ;GET RESULTS
2706 021042 020103  CMP R1,R3            ;DID IT CHANGE?
2707 021044 001017  BNE 20$            ;NO, MUST BE 11/238
2708 021046 005237 003136  INC T23A            ;SET THE FLAG
2709 021052 042737 170000 002120  BIC @170000,L$HIME  ;SUPERVISOR COULD BE WRONG
2710 021060 000240  NOP          ;BR 40$ FOR RELEASE
2711 021062          PRINTF @M8186          ;TELL THE SYSTEM TYPE
      021062 012746 005550  MOV @M8186,-(SP)
      021066 012746 000001  MOV @1,-(SP)
      021072 010600  MOV SP,R0
      021074 104417  TRAP C$PNTF
      021076 062706 000004  ADD @4,SP
2712 021102 000413  BR 40$             ;RETURN
2713 021104 005237 003140  20$: INC T23B            ;SET THE FLAG
2714 021110 000240  NOP          ;BR 40$ FOR RELEASE

```

07

ENVIRN - SETUP FREE DIAGNOSTIC SPACE

```

2715 021112          PRINTF  @M8189          ; TELL THE SYSTEM TYPE
      021112 012746 005641      MOV      @M8189, -(SP)
      021116 012746 000001      MOV      @1, -(SP)
      021122 010600          MOV      SP, R0
      021124 104417          TRAP     C$PNTF
      021126 062706 000004      ADD      @4, SP
2716 021132 000207      40$:    RTS      PC          ; RETURN
2717                                     .SBTTL  KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS
2718                                     ;*
2719                                     ;
2720                                     ; ROUTINE TO INIT KT-11
2721                                     ;
2722                                     ; -
2723
2724 021134          KTINIT:
2725 021134 005037 003124      CLR      KTFLG          ; INIT >28K MEMORY FLAG
2726 021140 005037 003126      CLR      KTENABLE       ; INIT TEST >28K FLAG
2727 021144 023727 002120 001577  CMP      L$HIME, @1577  ; GOT ENOUGH MEMORY (>28K)?
2728 021152 101444          BLOS     9$              ; NO.
2729 021154 013700 000004      MOV      @@ERRVEC, R0   ; SAVE OLD ERR VEC PTR.
2730 021160 012737 021252 000004  MOV      @2$, @@ERRVEC  ; SET ERR VEC PTR.
2731 021166 005737 177572      TST     @@SRO          ; GOT KT11?
2732 021172 000240          NOP                     ; (TRAP IF NO).
2733 021174 013737 002120 003124  MOV      L$HIME, KTFLG  ; YES. SET KT FLAG.
2734 021202 042737 000177 003124  BIC      @177, KTFLG    ;
2735 021210 010037 000004      MOV      R0, @@ERRVEC  ; RESTORE OLD ERR VEC PTR.
2736 021214 005000          CLR      R0              ; R0 = AR DATA.
2737 021216 012701 172340      MOV      @KIPAR0, R1   ; R1 = KI REGS PTR.
2738 021222 012761 077406 177740 1$:  MOV      @77406, -40(R1) ; SET DESCRIPTOR REG.
2739 021230 010021          MOV      R0, (R1)+     ; SET KIPAR REG.
2740 021232 062700 000200      ADD      @200, R0      ; BUMP AR DATA BY "4K".
2741 021236 020027 002000      CMP      R0, @2000    ; AT "I/O"?
2742 021242 001367          BNE     1$              ; NO.
2743 021244 012741 177600      MOV      @177600, -(R1) ; YES. SET KTPAR7 FOR I/O.
2744 021250 000405          BR      9$              ;
2745
2746 021252 012716 021260      2$:    MOV      @6$, (SP)     ; SET UP RETURN
2747 021256 000002          RTI                     ; RTI TO NEXT LOCATION
2748
2749 021260 010037 000004      6$:    MOV      R0, @@ERRVEC  ; RESTORE OLD ERR VEC PTR.
2750
2751 021264 000207      9$:    RTS      PC
2752                                     ;*
2753                                     ; SUBROUTINE TO SET EXTENDED FEATURES SWITCH
2754                                     ;
2755                                     ; Requires that SOFINIT and WRTCHR have been done previous to call.
2756                                     ;
2757                                     ;
2758                                     ; INPUTS:
2759                                     ; R5      CURRENT UNIT NUMBER
2760                                     ; OUTPUTS:
2761                                     ; The Extended Features Switch is set.
2762                                     ;
2763                                     ; -
2764
2765 021266          INVERT::
2766

```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

2767 021266 005737 002220          TST      EXTFEA          ; IS SWITCH SET?
2768 021272 001020          BNE      1$            ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
2769 021274 012737 100206 021340    MOV      #100206,CMDPKT ; WRT SUB-SYS MEM CMD
2770 021302 012737 021350 021342    MOV      #WSMBK,CMDPKT+2 ; MSG BUF ADDR
2771 021310 012737 000006 021346    MOV      #6,CMDPKT+6    ; BYTE COUNT
2772 021316 012737 100010 021350    MOV      #100010,WSMBK ; INVERT THE SWITCH
2773 021324 012704 021340          MOV      #CMDPKT,R4    ; SET CMDPKT INTO R4
2774 021330 004737 010742          JSR      PC,WRTCHR     ; DO IT
2775 021334 000207          1$:     RTS      PC    ; RETURN
2776
2777          ;          COMMAND PACKET.
2778
2779          021340          .          =          <.,+3>E177774 ;MUST BE ON MOD 4 BOUNDRY.
2780
2781 021340 000000          CMDPKT:: 0          ;1ST WORD IS TS05 COMMAND.
2782 021342 000000          0          ;2ND WORD IS THE BUFFER LOW ADDRESS.
2783 021344 000000          0          ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2784 021346 000000          0          ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2785
2786          ;          WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
2787
2788 021350 000000          WSMBK:: 0          ;1ST WORD:: SEL 0
2789 021352 000000          0          ;2ND WORD:: SEL 2
2790 021354 000000          0          ;3RD WORD:: SEL 4
2791          .EVEN
2792          ;
2793          ;          SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
2794          ;
2795          ;
2796          ;INPUTS:
2797          ;OUTPUTS:
2798          ;          The NXMFLG is set if we can test.
2799          ;          The NXMLO and NXMHI addresses are setup.
2800          ;
2801          ;
2802 021356          MEMCK::
2803
2804 021356          SAVREG          ;SAVE THE REGISTERS
2805 021362 005037 003130          CLR      NXMFLG        ;CLEAR THE FLAG
2806 021366 005037 003132          CLR      NXMLO        ;CLEAR THE TEST ADDRESS LO
2807 021372 005037 003134          CLR      NXMHI        ;CLEAR THE TEST ADDRESS HI
2808 021376 005737 003140          TST      T23B         ;IS IT A 11/23B?
2809 021402 001407          BEQ      1$           ;NO
2810 021404 023727 002120 007777    CMP      L$HIME,#7777  ; GREATER THAN 128K
2811 021412 103406          BLO     2$           ; NO
2812 021414 004737 021532          JSR      PC,NXMTST    ;SETUP THE ADDRESS
2813 021420 000427          BR      13$          ;SET THE FLAG AND EXIT
2814 021422 005737 003136          1$:     TST      T23A         ;IS IT A 11/23A?
2815 021426 001413          BEQ      4$           ;NO
2816 021430 023727 002120 005777    2$:     CMP      L$HIME,#5777 ;GREATER THAN 96K
2817 021436 101023          BHI     14$          ;YES,23A/23B WITH 128K MEMORY
2818 021440 023727 002120 003777    CMP      L$HIME,#3777 ;GREATER THAN 64K BUT LESS THAN 92K?
2819 021446 103403          BLO     4$           ;NO, CHECK 24K
2820 021450 004737 021532          JSR      PC,NXMTST    ;SETUP THE ADDRESS
2821 021454 000411          BR      13$          ;SET THE FLAG AND EXIT
2822 021456 023727 002120 001577    4$:     CMP      L$HIME,#1577 ;GREATER THAN 24K BUT LESS THAN 64K?
2823 021464 103410          BLO     14$          ;NO, TELL THEM AND EXIT WITH FLAG CLEAR

```

E7

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

2824 021466 004737 021532          JSR    PC,NXMTST          ;SETUP THE ADDRESS
2825 021472 062737 000077 003134  ADD    #77,NXMHI        ;FOOL THE 11/02 & 11/03
2826 021500 005237 003130          INC    NXMFLG          ;SET THE FLAG
2827 021504 000411                   BR     15$             ;EXIT
2828 021506 000410          14$:  BR     15$             ;NOP FOR PRINTOUT
2829 021510                   PRINTF #NOMEM          ;TELL THEM & EXIT ***NO PRINT*****
      021510 012746 005454          MOV    #NOMEM,-(SP)
      021514 012746 000001          MOV    #1,-(SP)
      021520 010600                   MOV    SP,R0
      021522 104417                   TRAP  C$PNTF
      021524 062706 000004          ADD    #4,SP
2830 021530 000207          15$:  RTS    PC          ;RETURN
2831
2832
2833          ;*
2834          ; SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
2835          ;
2836          ; OUTPUTS: NXMLO, NXMHI          ; SETUP WITH NXM ADDRESS
2837          ;
2838          ;-
2839 021532 013701 002120  NXMTST: MOV    L$HIME,R1      ;GET TOP OF MEMORY
2840 021536 062701 000200          ADD    #200,R1         ;MAKE IT I/O BLOCK OR OTHER NXM
2841 021542 042701 000177          BIC    #177,R1
2842 021546 010102                   MOV    R1,R2          ;RESAVE RESULTS
2843          000006          .REPT 6
2844          ASL    R1          ;PUT IN PLACE FOR XFER
2845          .ENDR
2846 021564 010137 003132          MOV    R1,NXMLO       ;SAVE TEST ADDRESS LOW
2847          000012          .REPT 10
2848          ASR    R2          ;PUT IN PLACE FOR XFER
2849          .ENDR
2850 021614 042702 177700          BIC    #177700,R2     ;DON'T WANT ILA!
2851 021620 010237 003134          MOV    R2,NXMHI       ;SAVE TEST ADDRESS HIGH
2852 021624 000207          RTS    PC             ;RETURN
2853
2854
2855
2856 021626          ENDMOD

```

F7

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

7
8
9 021626
021626
10
16

.TITLE TSV4 - MISCELLANEOUS SECTIONS

BGNMOD TSV4

TSV4::

PROTECTION TABLE

18

19 021626

021626

20 021626 177777 177777 177777

21 021636

22

.SBTTL PROTECTION TABLE

BGNPROT

L\$PROT::

.WORD -1. -1. -1. -1

ENDPROT

;NO DEVICE PROTECTION REQUIRED.

INITIALIZE SECTION

```

24          .SBTTL  INITIALIZE SECTION
25
26          ;**
27          ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
28          ;AT THE BEGINNING OF EACH PASS.
29          ;
30          ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
31          ;IF "CONTINUE", NOTHING IS REQUIRED.
32          ;
33          ;--
34          ;*
35          ;INSERT TEMPORARY JUMP TO ODT
36          ;-
37 021636          BGNINIT
38 021636          L$INIT::
39 021636          SETVEC  #140,#170000,#340          ;ODT ROM ADDRESS          ;JB REV A-0
021636          MOV      #340,-(SP)
021642          MOV      #170000,-(SP)
021646          MOV      #140,-(SP)
021652          MOV      #3,-(SP)
021656          TRAP    C$SVEC
021660          ADD     #10,SP
40
41 021664          005037 002220          40$:  CLR      EXTFEA
42 021670          005037 003130          CLR      NXMFLG
43 021674          012737 006354 002172  MOV      #EPRT1,EPRTSW          ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
44 021702          005037 003146          CLR      SIFLAG          ;CLEAR "SOFT INIT" FLAG
45 021706          005037 003126          CLR      KTENABLE          ;CLEAR TEST ABOVE 28K FLAG
46 021712          005037 002274          CLR      RAMSIZ          ;CLEAR RAM SIZE FOR RAMERR ROUTINE
47 021716          021716 012700 000036  READEF  #EF.CONTINUE
021716          MOV      #EF.CONTINUE,R0
021722          104447          TRAP    C$REFG
48 021724          BNCOMPLETE 1$
021724          103023          BCC     1$
49 021726          023737 002174 002012  CMP      UNITN,L$UNIT          ;UNIT IN RANGE?
50 021734          103070          BHIS    4$          ;BR IF NO.
51 021736          005737 003104          TST     DUFLG          ;DROPPED UNIT?
52 021742          100472          BMI     NXTU          ;BR IF YES
53 021744          013701 002174          MOV     UNITN,R1
54 021750          006301          ASL    R1
55 021752          005761 003170          TST    ERTABL(R1)
56 021756          001516          BEQ    SETU
57 021760          032761 040000 003170  BIT     #BIT14,ERTABL(R1)          ;DROPPED?
58 021766          001060          BNE    NXTU
59 021770          EXIT          ;DO NOTHING IF "CONTINUE".
021770          104432          TRAP   C$EXIT
021772          000416          .WORD  L10030-.
60 021774          021774 012700 000035  1$:  READEF  #EF.NEW
021774          MOV     #EF.NEW,R0
022000          104447          TRAP   C$REFG
61 022002          BNCOMPLETE NXTU          ;TAKE NEXT UNIT IF NOT NEW PASS.
022002          103052          BCC    NXTU
62 022004          022004 012700 000040  READEF  #EF.START
022010          104447          MOV     #EF.START,R0
63 022012          022010 104447          TRAP   C$REFG
          BCOMPLETE 2$

```

INITIALIZE SECTION

```

022012 103404          BCS      2$
64 022014          READEF   #EF.RESTART
022014 012700 000037  MOV      #EF.RESTART,RO
022020 104447          TRAP    C$REFG
65 022022          BNCOMPLETE 31$
022022 103031          BCC     31$
66 022024          2$:
67 022024          BRESET
022024 104433          TRAP    C$RESET
68 022026 005037 002206  CLR      TSTCNT
69 022032 005037 002214  CLR      FATFLG
70 022036 005037 003136  CLR      T23A
71 022042 005037 003140  CLR      T23B
72          :
73          : MOV     #340,-(SP)
74          : MOV     #20,-(SP)
75 022046 005037 003372  JMP     0.ODT
76 022052          CLR     SKIPT
77 022052 012737 177777 002176  MOV     #-1,QVP
78 022060 004737 020710          JSR    PC,ENVIRN
79 022064 004737 021134          JSR    PC,KTINIT
80 022070 012700 003170          MOV     #ERTABL,RO
81 022074 005020          CLR     (RO)+
82 022076 020027 003370          CMP    RO,#ERTABE
83 022102 103774          BLO    30$
84 022104 000404          BR     4$
85 022106 005037 002176          CLR     QVP
86 022112 000137 022162          JMP    PASRPT
87
88 022116          4$:
89 022116 012737 177777 002174  NEWPAS: MOV    #-1,UNITN
90 022124 005037 002212          CLR    DEVCNT
91 022130          NXTU:
022130 104422          BREAK
92 022132 005237 002174          TRAP   C$BRK
93 022136 023737 002174 002012  INC     UNITN
94 022144 103423          CMP    UNITN,L$UNIT
95 022146 012737 177777 003104  BLO    SETU
96 022154 000401          MOV    #-1,DUFLG
97 022156          BR     11$
022156 104444          DOCLN
98 022160 000240          TRAP   C$DCLN
99 022162          NOP
100 022162 023727 002012 000001  PASRPT: CMP    L$UNIT,#1
101 022170 101752          BLOS   NEWPAS
102 022172 005737 002212          TST    DEVCNT
103 022176 001747          BEQ    NEWPAS
104 022200          RFLAGS  RO
022200 104421          TRAP   C$RFLA
105 022202 032700 000100          BIT    #ISR,RO
106 022206 001343          BNE    NEWPAS
107
108 022210          DORPT
022210 104424          TRAP   C$DRPT
109 022212 000741          BR     NEWPAS
110 022214          10$:
111

```

```

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

```


INITIALIZE SECTION

```

112 022214          SETU:  GPWARD  UNITN,R0          ;GET UNIT N P-TABLE POINTER.
    022214 013700 002174  MOV      UNITN,R0
    022220 104442  TRAP    C$GPWRD
113 022222          BNCOMPLETE NXTU          ;BR IF UNIT NOT AVAILABLE.
    022222 103342  BCC     NXTU
114 022224 005037 003104  CLR     DUFLG          ;CLEAR "DROPPED" FLAG.
115 022230 005237 002212  INC     DEVCNT
116 022234 012001          MOV      (R0)+,R1          ;GET 1ST REGISTER ADDRESS.
117 022236 010137 002200  MOV      R1,CSRADDR    ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
118
119 022242 012001          MOV      (R0)+,R1          ;GET VECTOR ADDRESS.
120          ;MOV     (R0),R2          ;GET INTERRUPT PRIORITY
121          ;MOV     R2,IPRI          ;SET INTERRUPT PRIORITY.
122 022244 010137 002202  MOV      R1,IVEC       ;SET INTERRUPT VECTOR POINTER...
123 022250 012721 016276  MOV      @INTR,(R1)+   ;...VECTOR...
124 022254 013721 002204  MOV      IPRI,(R1)+   ;...AND PRIORITY.
125
126 022260          1$:
127          ; TST     QVP          ;1ST PASS ??
128          ; BEQ    5$          ;NO. SKIP THE PASS 1 STUFF.
129
130
131          ;
132          ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
133          ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
134 022260 013701 002174  ;
135 022264 006301          MOV      UNITN,R1
136 022266 052761 100000 003170  ASL     R1
137 022274 005037 005766          BIS     @BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
138 022300 023727 002012 000001  CLR     EXTA          ;CLEAR ERROR EXTENSION FLAG.
139 022306 101416          CMP     L$UNIT,#1      ;ARE WE TESTING MULTIPLE UNITS?
140 022310          BLOS   10$          ;BR IF NO.
    022310 104421  RFLAGS  R0          ;YES -- GET OPERATOR FLAGS.
    022312 032700 001000  TRAP    C$RFLA
141 022312 032700 001000  BIT     @PNT,R0        ;SHOULD WE PRINT UNIT #?
142 022316 001412          BEQ    10$          ;BR IF NOT.
143 022320          PRINTF @PUNIT,UNITN ;PRINT THE UNIT #
    022320 013746 002174  MOV     UNITN,-(SP)
    022324 012746 022412  MOV     @PUNIT,-(SP)
    022330 012746 000002  MOV     @2,-(SP)
    022334 010600          MOV     SP,R0
    022336 104417  TRAP    C$PNTF
    022340 062706 000006  ADD     @6,SP
144 022344          10$:
145 022344 005037 003106  CLR     NODEV
146 022350 013701 002200  MOV     CSRADDR,R1    ;ADDRESS OF FIRST REGISTER
147 022354 010102          MOV     R1,R2          ;START OF REGISTERS
148 022356 062702 000002  ADD     @TSSR,R2      ;ADDRESS OF TSSR REGISTER
149 022362 004737 016456  JSR    PC,XNXM        ;TEST BOTH CONTROLLER REGISTERS...
150 022366 103005          BCC    2$          ;...AND BR IF ALL OK.
151 022370 010137 003106  MOV     R1,NODEV      ;FLAG DEVICE AS NON-EXISTENT
152 022374 012737 177777 003104  MOV     @-1,DUFLG    ;DROP THIS UNIT.
153 022402          2$:
154          ;
155          ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
156          ;
157 022402          5$:  SETPRI  @PRI00          ;ENABLE INTERRUPTS.
    022402 012700 000000  MOV     @PRI00,R0

```

K7

INITIALIZE SECTION

```

022406 104441          TRAP  C$SPRI
158 022410          ENDINIT
    022410          L10030:
    022410 104411          TRAP  C$INIT
159
160 022412    045    116    045 PUNIT: .ASCIZ /#N#N#A***** TESTING UNIT #D2#A *****/
161                                .EVEN

```

ADD AND DROP UNITS SECTIONS

```

163                                     .SBTTL  ADD AND DROP UNITS SECTIONS
164
165                                     ;**
166                                     ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
167                                     ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
168                                     ; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
169                                     ;--
170 022460                                BGNAU
171 022460                                L$AU::
172 022460 010001                          MOV     R0,R1                ; GET UNIT TO BE ADDED (R0)
173 022464 006301                          ASL     R1                    ; MAKE IT A WORD INDEX
174 022472 052761 100000 003170           BIS     #100000,ERTABL(R1)   ; SET THE "ACTIVE" BIT
175 022472 042761 040000 003170           BIC     #40000,ERTABL(R1)   ; CLEAR THE "DROPPED" BIT
176 022500                                PRINTF  #1$,R0
177 022500 010046                          MOV     R0,-(SP)
178 022502 012746 022526                  MOV     #1$,-(SP)
179 022506 012746 000002                  MOV     #2,-(SP)
180 022512 010600                          MOV     SP,R0
181 022514 104417                          TRAP   C$PNTF
182 022516 062706 000006                  ADD     #6,SP
183 022522                                EXIT   AU
184 022522 000167                          .WORD  J$JMP
185 022524 000026                          .WORD  L10031-2-.
186 022526 045 116 045 1$:                .ASCIZ /#N#A UNIT #D#A ADDED/
187                                     .EVEN
188
189 022554                                ENDAU                        ; UNUSED.
190 022554                                L10031:
191 022554 104452                          TRAP   C$AU
192
193                                     ;**
194                                     ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
195                                     ; TO BE REMOVED FROM THE TEST LIST.
196                                     ;
197                                     ; SUPVSR DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN.
198                                     ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADC"
199                                     ; COMMAND. OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
200                                     ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
201                                     ; WHICH ARE STILL ACTIVE.
202                                     ; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.
203
204 022556                                BGNDU
205 022556                                L$DU::
206 022556 012737 177777 003104           MOV     #-1,DUFLG
207 022564 010001                          MOV     R0,R1
208 022566 006301                          ASL     R1
209 022570 052761 140000 003170           BIS     #140000,ERTABL(R1)   ; SAY DROPPED
210 022576 000240 000240 000240           240,240,240                ; ??????????
211 022604                                PRINTF  #1$,R0
212 022604 010046                          MOV     R0,-(SP)
213 022606 012746 022632                  MOV     #1$,-(SP)
214 022612 012746 000002                  MOV     #2,-(SP)
215 022616 010600                          MOV     SP,R0
216 022620 104417                          TRAP   C$PNTF
217 022622 062706 000006                  ADD     #6,SP
218 022626                                EXIT   DU
219 022626 000167                          .WORD  J$JMP
220 022630 000030                          .WORD  L10032-2-.

```

ADD AND DROP UNITS SECTIONS

```

200 022632      045      116      045 1$:      .ASCIZ  /%N%A  UNIT %D%A DROPPED/
201              .EVEN
202 022662              ENDDU
      022662              L10032:
      022662 104453      TRAP    C$DU
203              : **
204              ; AUTO-DROP CODE SECTION.
205              : --
206 022664              BGNAUTO
      022664              L$AUTO::
207 022664 013705 002200      MOV     CSRADDR,R5      ;POINT TO DEVICE REGISTER
208 022670 012703 000550      MOV     #360.,R3      ;ENOUGH TIME FOR 2400' REEL TO REWIND
209 022674 004737 016330      JSR    PC,WAITF      ;WAIT FOR SSR TO SET
210 022700 103420      BCS    20$           ;LEAVE WHEN SSR IS SET
211 022702              DELAY  250.        ;WAIT FOR .25 SECONDS
      022702 012727 000372      MOV     #250.,(PC)+
      022706 000000      .WORD  0
      022710 013727 002116      MOV     L$DLY,(PC)+
      022714 000000      .WORD  0
      022716 005367 177772      DEC     -6(PC)
      022722 001375      BNE     .-4
      022724 005367 177756      DEC     -22(PC)
      022730 001367      BNE     .-20
212 022732 005303      DEC     R3           ;BUMP COUNTER DOWN
213 022734 001357      BNE     10$         ;KEEP GOING
214 022736 004737 017262      JSR    PC,CKDROP    ;TRY AND DROP UNIT
215 022742              20$:
216 022742              ENDAUTO      ; UNUSED.
      022742              L10033:
      022742 104461      TRAP    C$AUTO

```

CLEAN-UP AND REPORT CODING SECTIONS

```

218 .SBTTL CLEAN-UP AND REPORT CODING SECTIONS
219
220 ;**
221 ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
222 ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
223 ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
224 ;--
225 022744 BGNCLN
      022744 L$CLEAN::
226 022744 013705 002200      MOV      CSRADDR,R5      ;POINT TO DEVICE REGISTER
227 022750 005737 003104      TST      DUFLG          ;"DROPPED" FLAG IS SET ON...
228 022754 100405      BMI      1$          ;...AND GROSS CONTROLLER FAULT...
229      ;...DON'T TRY TO XCT CLEANUP CODE.
230
231 022756 012765 000000 000002      MOV      #0,TSSR(R5)    ;DO SOFT INIT
232 022764 004737 016330      JSR      PC,WAITF
233 022770      1$:
234 022770      2$:      ENDCLN
      022770 L10034:
      022770 104412      TRAP     C$CLEAN
235
236 ;**
237 ; THE REPORT CODING SECTION CONTAINS THE
238 ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
239 ;--
239 022772 BGNRPT
      022772 L$RPT::
240 022772 PRINTS  #DEVSUM
      022772 012746 023234      MOV      #DEVSUM,-(SP)
      022776 012746 000001      MOV      #1,-(SP)
      023002 010600      MOV      SP,R0
      023004 104416      TRAP     C$PNTS
      023006 062706 000004      ADD      #4,SP
241 023012 010246      MOV      R2,-(SP)
242 023014 010346      MOV      R3,-(SP)
243 023016 010446      MOV      R4,-(SP)
244 023020 012704 003170      MOV      #ERTABL,R4    ; GET START OF ERROR TABLE.
245 023024 005003      CLR      R3            ; CLEAR UNIT NUMBER
246 023026 011402      1$:      MOV      (R4),R2      ; GET ERROR TABLE ENTRY & TEST IT.
247 023030 001467      BEQ      4$            ; ZERO IF UNIT NOT RUN
248 023032 100066      BPL      4$
249 023034 032702 040000      BIT      #BIT14,R2     ; WAS UNIT DROPPED?
250 023040 001015      BNE      2$            ; BR IF YES
251 023042 042702 170000      BIC      #C7777,R2     ; GET ERROR COUNT FIELD
252 023046 PRINTS  #DEVONL,R3,R2    ; PRINT
      023046 010246      MOV      R2,-(SP)
      023050 010346      MOV      R3,-(SP)
      023052 012746 023271      MOV      #DEVONL,-(SP)
      023056 012746 000003      MOV      #3,-(SP)
      023062 010600      MOV      SP,R0
      023064 104416      TRAP     C$PNTS
      023066 062706 000010      ADD      #10,SP
253 023072 000446      BR       4$
254 023074 020227 160000      2$:      CMP      R2,#160000    ; WAS UNIT NON-EXISTENT?
255 023100 001012      BNE      3$            ; BR IF NO
256 023102 PRINTS  #DEVNXR,R3
      023102 010346      MOV      R3,-(SP)
      023104 012746 023341      MOV      #DEVNXR,-(SP)

```

CLEAN-UP AND REPORT CODING SECTIONS

```

023110 012746 000002      MOV      #2,-(SP)
023114 010600      MOV      SP,R0
023116 104416      TRAP     C:PNTS
023120 062706 000006      ADD      #6,SP
257 023124 000431      BR       4$
258 023126 020227 160001      3$:    CMP      R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
259 023132 001012      BNE     30$              ; BR IF NO.
260 023134      PRINTS  #DEVNRD,R3
023134 010346      MOV      R3,-(SP)
023136 012746 023423      MOV      #DEVNRD,-(SP)
023142 012746 000002      MOV      #2,-(SP)
023146 010600      MOV      SP,R0
023150 104416      TRAP     C:PNTS
023152 062706 000006      ADD      #6,SP
261 023156 000414      BR       4$
262 023160 042702 170000      30$:   BIC      #C7777,R2
263 023164      PRINTS  #DEVDR0,R3,R2
023164 010246      MOV      R2,-(SP)
023166 010346      MOV      R3,-(SP)
023170 012746 023504      MOV      #DEVDR0,-(SP)
023174 012746 000003      MOV      #3,-(SP)
023200 010600      MOV      SP,R0
023202 104416      TRAP     C:PNTS
023204 062706 000010      ADD      #10,SP
264 023210 062704 000002      4$:    ADD      #2,R4
265 023214 005203      INC      R3
266 023216 020427 003370      CMP      R4,#ERTABE
267 023222 103701      BLO     1$
268 023224 012604      MOV      (SP),R4
269 023226 012603      MOV      (SP),R3
270 023230 012602      MOV      (SP),R2
271 023232      ENDRPT      ; UNUSED.
023232      L10035:
023232 104425      TRAP     C:RPT
272
273
274 023234      045      116      045  DEVSUM: .ASCIZ  /#N#ADEVICE STATUS SUMMARY:#N/
275 023271      045      101      040  DEVONL: .ASCIZ  /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
276 023341      045      101      040  DEVNXR: .ASCIZ  /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
277 023423      045      101      040  DEVNRD: .ASCIZ  /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
278 023504      045      101      040  DEVDR0: .ASCIZ  /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
279
280
281 023554      ENDMOD
282
283

```

CLEAN-UP AND REPORT CODING SECTIONS

1
2
3
10
11
17

.TITLE TEST 1 - HARDWARE TEST 1-8 TESTS

023554
023554

TSV7B:: BGNMOD TSV7B

TEST 1: WRITE TAPE MARK RETRY

313	025124	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND	
314	025130	103411			BCS	30\$;BR, IF NO PROBLEM	
315	025132	016501	000002		MOV	TSSR(R5),R1		;GET TSSR	
316	025136	010004			MOV	R0,R4		;SAVE PACKET ADDRESS	
317	025140	005237	002214		INC	FATFLG		;ERROR COUNT	
321	025144				ERRHRD	ERRNO,T29RWN,PKTSSR		;REWIND NOT ACCEPTED	
	025144	104456						TRAP	C\$ERHRD
	025146	000166						.WORD	118
	025150	030335						.WORD	T29RWN
	025152	012126						.WORD	PKTSSR
322	025154			30\$:	CKLOOP			;LOOP IF SELECTED	
	025154	104406						TRAP	C\$CLP1
323	025156	013701	026400		MOV	T29BFR+6,R1		;PICK UP XSTO	
324	025162	010102			MOV	R1,R2		;SET UP EXPECTED	
325	025164	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED	
326	025170	020102			CMP	R1,R2		;DOES EXP = REC'D	
327	025172	001406			BEQ	40\$;BR, IF EQUAL (OK)	
328	025174	005237	002214		INC	FATFLG		;ERROR COUNT	
332	025200				ERRHRD	ERRNO,T29BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND	
	025200	104456						TRAP	C\$ERHRD
	025202	000167						.WORD	119
	025204	030026						.WORD	T29BOT
	025206	015554						.WORD	EXPREC
333	025210			40\$:	CKLOOP			;LOOP IF SELECTED	
	025210	104406						TRAP	C\$CLP1
334	025212	012737	140011	026470	MOV	#140011,T29PK3		;WRITE TAPE MARK,ACK,CVC=1 COMMAND	
335	025220	012704	026470		MOV	#T29PK3,R4		;SET UP R4 WITH PACKET ADDRESS	
336	025224	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND	
337	025230	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET	
338	025234	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS	
339	025240	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED	
340	025244	020102			CMP	R1,R2		;ARE THEY EQUAL	
341	025246	001406			BEQ	70\$;BR, IF OK	
342	025250	005237	002214		INC	FATFLG		;ERROR COUNT	
346	025254				ERRHRD	ERRNO,T29WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE TAPE MARK	
	025254	104456						TRAP	C\$ERHRD
	025256	000170						.WORD	120
	025260	030727						.WORD	T29WDC
	025262	012126						.WORD	PKTSSR
347	025264			70\$:	CKLOOP			;LOOP IF SELECTED	
	025264	104406						TRAP	C\$CLP1
348	025266	012703	000001		MOV	#1.,R3		;NUMBER OF RECORDS TO WRITE TM	
349	025272	012737	141011	026470	MOV	#141011,T29PK3		;WRITE TAPE MARK RETRY,ACK,CVC=1 COMMAND	
350	025300	012704	026470		MOV	#T29PK3,R4		;SET UP R4 WITH PACKET ADDRESS	
351	025304	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND	
352	025310	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET	
353	025314	016501	000002		MOV	TSSR(R5),R1		;PICK UP TSSR	
354	025320	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED (SSR ONLY)	
355	025324	020102			CMP	R1,R2		;WAS STATUS GOOD	
356	025326	001406			BEQ	165\$;BR, IF TERMINATION WAS GOOD	
357	025330	005237	002214		INC	FATFLG		;ERROR COUNT	
361	025334				ERRHRD	ERRNO,T29WDC,PKTSSR		;TSSR NOT CORRECT AFTER WRT TAPE M.	
	025334	104456						TRAP	C\$ERHRD
	025336	000171						.WORD	121
	025340	030727						.WORD	T29WDC
	025342	012126						.WORD	PKTSSR
362	025344			165\$:	CKLOOP			;LOOP IF SELECTED	

TEST 1: WRITE TAPE MARK RCTRY

453	025736	012737	140011	026470	MOV	#140011,T29PK3	;WRITE TAPE MARK,ACK,CVC=1 COMMAND
454	025744	012704	026470		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
455	025750	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND
456	025754	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
457	025760	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS
458	025764	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED
459	025770	020102			CMP	R1,R2	;ARE THEY EQUAL
460	025772	001406			BEQ	70\$;BR, IF OK
461	025774	005237	002214		INC	FATFLG	;ERROR COUNT
465	026000				ERRHRD	ERRNO,T29WDC,PKTSSR	;TSSR INCORRECT AFTER WRITE TAPE MARK
	026000	104456					TRAP C\$ERHRD
	026002	000200					.WORD 128
	026004	030727					.WORD T29WDC
	026006	012126					.WORD PKTSSR
466	026010			70\$:	CKLOOP		;LOOP IF SELECTED
	026010	104406					TRAP C\$CLP1
467	026012	012703	000012		150\$:	MOV #10.,R3	;NUMBER OF RECORDS TO WRITE TM
468	026016	012737	000001	026472	MOV	#1,T29RB	;SET UP PACKET
469	026024	012737	141011	026470	MOV	#141011,T29PK3	;WRITE TAPE MARK RETRY,ACK,CVC=1 COMMAND
470	026032	012704	026470		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
471	026036	010465	000000		155\$:	MOV R4,TSDB(R5)	;ISSUE COMMAND
472	026042	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
473	026046	016501	000002		MOV	TSSR(R5),R1	;PICK UP TSSR
474	026052	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED (SSR ONLY)
475	026056	020102			CMP	R1,R2	;WAS STATUS GOOD
476	026060	001406			BEQ	165\$;BR, IF TERMINATION WAS GOOD
477	026062	005237	002214		INC	FATFLG	;ERROR COUNT
481	026066				ERRHRD	ERRNO,T29WDC,PKTSSR	;TSSR NOT CORRECT AFTER WRT TAPE M.
	026066	104456					TRAP C\$ERHRD
	026070	000201					.WORD 129
	026072	030727					.WORD T29WDC
	026074	012126					.WORD PKTSSR
482	026076			165\$:	CKLOOP		;LOOP IF SELECTED
	026076	104406					TRAP C\$CLP1
483	026100	005303			DEC	R3	;BUMP COUNTER DOWN
484	026102	001355			BNE	155\$;BR, IF LESS THAN 10 TAPE MARKS
485	026104	012737	140410	026470	MOV	#140410,T29PK3	;SPACE REVERSE,ACK,CVC=1, COMMAND
486	026112	012737	000001	026472	MOV	#1,T29RB	;NUMBER OF RECORDS TO SPACE BACK
487	026120	012704	026470		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
488	026124	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND
489	026130	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
490	026134	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS
491	026140	012702	100204		MOV	#SSR!SC!BIT2,R2	;SET UP EXPECTED
492	026144	020102			CMP	R1,R2	;ARE THEY EQUAL
493	026146	001406			BEQ	222\$;BR, IF OK
494	026150	005237	002214		INC	FATFLG	;ERROR COUNT
498	026154				ERRHRD	ERRNO,T29WDE,PKTSSR	;TSSR INCORRECT AFTER SPACE CMD.
	026154	104456					TRAP C\$ERHRD
	026156	000202					.WORD 130
	026160	027612					.WORD T29WDE
	026162	012126					.WORD PKTSSR
499	026164			222\$:	CKLOOP		;LOOP IF SELECTED
	026164	104406					TRAP C\$CLP1
500	026166	012737	100410	026470	MOV	#100410,T29PK3	;SPACE REVERSE,ACK, COMMAND
501	026174	012737	000005	026472	MOV	#5,T29RB	;NUMBER OF RECORDS TO SPACE BACK
502	026202	012704	026470		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
503	026206	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND

TEST 1: WRITE TAPE MARK RETRY

```

554 026366 000000          .WORD 0
555 026370 000000 T29DSW: .WORD 0          ;SELECT DRIVE 0
556 026372          T29BFR: .BLKW 25.      ;MESSAGE BUFFER
557          ;
558          ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
559          ;
561          026460
563 026460          T29PK2: .-<..10>&177770
564 026460 100006          .WORD 100006          ;WRITE SUB SYS MEM COMMAND, AND ACK
565 026462 026500          .WORD T29BF2          ;ADDRESS OF SELECT BLOCK DATA
566 026464 000000          .WORD 0
567 026466 000006          .WORD 6.            ;SIZE OF DATA PACKET
568          ;
572 026470          T29PK3:          ;
573 026470 140005          .WORD 140005          ;WRITE TAPE MARK RETRY COMMAND, CVC=1 AND ACK
574 026472          T29RB:          ;
575 026472 003116 T29WB: .WORD FREE          ;ADDRESS OF WRITE BUFFER
576 026474 000000          .WORD 0
577 026476 000000 T29SZ: .WORD 0          ;SIZE OF BUFFER (EXTENT)
578          .EVEN
579          ;
580          ;
581          ;
582 026500          T29BF2:          ;
583 026500          010 T29BS0: .BYTE 10          ;BSELO AREA
584 026501          200 T29BS1: .BYTE 200          ;BSEL1 AREA
585 026502 000000 T29S2: .WORD 0          ;SEL 2 AREA
586 026504 000000 T29S3: .WORD 0          ;DATA AREA
587          ;
588          ;
589          .EVEN
590          ;TAPE MOTION PACKET COMMAND VALUES
591          ;
592 026506 140001 T29RN: .WORD 140001          ;READ DATA
593 026510 140401 T29WDR: .WORD 140401          ;READ DATA REVERSE
594 026512 141001 T29CON: .WORD 141001          ;READ PREVIOUS OPP=0
595 026514 161001          .WORD 161001          ;READ PREVIOUS OPP=1
596 026516 141401          .WORD 141401          ;WRITE TAPE MARK RETRY NEXT OPP=0
597 026520 161401          .WORD 161401          ;WRITE TAPE MARK RETRY NEXT OPP=1
598 026522 177777          .WORD 177777          ;END OF DATA
599          ;
600          ;
601 026524 000000 T29CNT: .WORD 0          ;TAPE RECORD COUNTER STORAGE AREA
602          ;
603 026526 000000 T29RSZ: .WORD 0          ;RECORD STORAGE SIZE AREA
604 026530 000000 T29DLY: .WORD          ;DELAY COUNTER STORAGE AREA
605          ;*
606          ;LOCAL TEXT MESSAGES FOR TEST
607          ;-
608          ;
609 026532          104 162 151 T29OFL: .ASCIZ 'Drive is OFFLINE'
610 026553          124 141 160 T29WNG: .ASCIZ 'Tape Position Incorrect After WRITE TAPE MARK RETRY Previous (OPP=1)'
611 026660          127 122 111 T29NEF: .ASCIZ 'WRITE TAPE MARK RETRY, At BOT, Failed To Set NEF (XSTO)'
612 026750          124 123 123 T29RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
613 027017          127 122 111 T29RRF: .ASCIZ 'WRITE TAPE MARK RETRY Previous (Space Reverse, Read Forward) Command Failed
614 027133          127 122 111 T29RRG: .ASCIZ 'WRITE TAPE MARK RETRY Previous (Read Forward, Space Reverse) Command Failed
615 027247          120 117 123 T29SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'

```

TEST 1: WRITE TAPE MARK RCTRY

616	027331	122	111	102	T29LOR: .ASCIZ	'RIB NOT SET AFTER READ REVERSE INTO BOT'
617	027401	124	123	123	T29WDF: .ASCIZ	'TSSR Not Correct After Illegal Mode Bits Set'
618	027456	111	154	154	T29LOQ: .ASCIZ	'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
619	027537	127	122	111	T29SSR: .ASCIZ	'WRITE TAPE MARK RETRY COMMAND Not Accepted'
620	027612	124	123	123	T29WDE: .ASCIZ	'TSSR Not Correct After SPACE REVERSE DATA Command'
621	027674	052	052	052	T29WLK: .ASCIZ	'*****TAPE IS WRITE-LOCKED AND WILL CAUSE ERRORS*****'
622	027761	124	123	123	T29WRT: .ASCIZ	'TSSR Not Correct After WRITE Command'
623	030026	124	141	160	T29BOT: .ASCIZ	'Tape Not At BOT After REWIND Command'
624	030073	104	141	164	T29DTA: .ASCIZ	'Data Written To Tape Not Equal To Data Read From Tape'
625	030161	127	122	111	T29EOT: .ASCIZ	'WRITE TAPE MARK RETRY DATA OVER EOT GAVE NO TAPE STATUS ALERT'
626	030257	124	123	123	T29TM: .ASCIZ	'TSSR Not Correct After SPACE REVERSE Into BOT'
627	030335	122	145	167	T29RWN: .ASCIZ	'Rewind (POSITION) Command Not Accepted'
628	030404	122	101	115	T29RNC: .ASCIZ	'RAM Error, Correct Data Pattern Not In Ram'
629	030457	124	123	123	T29AM3: .ASCIZ	'TSSR Init. Failed After WRITE TAPE MARK RETRY COMMAND'
630	030545	104	162	151	T29OF7: .ASCIZ	'Drive 7 Select Failed To Set "OFL" In TSSR'
631	030620	124	123	123	T29WDD: .ASCIZ	'TSSR Not Correct After WRITE TAPE MARK RETRY DATA Command, SWB Bit Set'
632	030727	124	123	123	T29WDC: .ASCIZ	'TSSR Not Correct After WRITE TAPE MARK RETRY DATA Command'
633	031021	103	126	103	T29VCK: .ASCIZ	'CVC Set, Didn't Reset VCK In Message Buffer'
634	031074	124	123	102	T298A: .ASCIZ	'TSBA Not Correct After WRITE TAPE MARK RETRY DATA Command'
635	031166	127	122	111	T29WSS: .ASCIZ	'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
636	031255	122	145	141	T29LON: .ASCIZ	'Reading Long Record Failed To Set RLL Bit In XST0'
637	031337	122	145	141	T29LOP: .ASCIZ	'Reading Long Record Failed To Set RLS Bit In XST0'
638	031421	122	145	163	T29PBP: .ASCIZ	'Residual Byte Count Incorrect After Short Record Read'
639	031507	122	145	141	T29TRL: .ASCIZ	'Reading Long Record Failed To Give Tape Status Alert'
640	031575	104	141	164	T29NEQ: .ASCIZ	'Data WRITE TAPE MARK RETRY From Tape Not Correct, After SWB=1'
641	031673	124	123	123	T29RDG: .ASCIZ	'TSSR Incorrect After READ REVERSE Into Tape Mark'
642	031754	127	122	111	T29RIB: .ASCIZ	'WRITE TAPE MARK RETRY At First Record, Failed To Set RIB (XST3)'
643	032054	124	115	113	T29RRN: .ASCIZ	'TMK (XST0) Failed To Set After READ REVERSE Into Tape Mark'
644	032147	127	162	151	TST29ID: .ASCIZ	'Write Tape Mark Retry'
645					.EVEN	
646					;	
647					;	
648					;	
649					ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES	
650					WRITE SUBSYSTEM MEMORY COMMAND	
651					;	
652					;	
653	032176				T29REST:	
654	032176				SAVREG	SAVE THE REGISTERS
655	032202	012701	026350		MOV #T29PACKET,R1	START OF THE PACKET
656	032206	012721	140004		MOV #140004,(R1)	WRITE SUBSYSTEM MEM. WITH ACK, CVC=1
657	032212	012721	026360		MOV #T29DATA,(R1)	ADDRESS OF CHARAISTICS DATA BLOCK
658	032216	005021			CLR (R1)	EXTENDED ADDRESS
659	032220	012721	000012		MOV #10,(R1)	SIZE OF DATA BLOCK IN BYTES
660	032224	012721	026372		MOV #T29BFR,(R1)	ADDRESS OF MESSAGE BUFFER
661	032230	005021			CLR (R1)	
662	032232	012721	000024		MOV #20,(R1)	LENGTH OF MESSAGE BUFFER
663	032236	005021			CLR (R1)	
664	032240	012711	000000		MOV #0,(R1)	SELECT DRIVE ZERO (0)
665	032244	012702	000030		MOV #24,R2	NUMBER OF LOCATIONS TO BE CLEARED
666	032250	012762	177777	026372 64\$:	MOV #177777,T29BFR(R2)	ALL ONES TO MESSAGE BUFFER
667	032256	005742			TST -(R2)	NEXT LOCATION
668	032260	020227	000000		CMP R2,#0	CHECK FOR END OF LOOP
669	032264	001371			BNE 64\$	KEEP GOING UNTIL DONE
670	032266	000207			RTS PC	RETURN
671						
672	032270				T29RT2:	

TEST 1: WRITE TAPE MARK RETRY

```

673 032270          SAVREG          ;SAVE THE REGISTERS
674 032274 012701 026460      MOV      #T29PK2,R1          ;START OF THE PACKET
675 032300 012721 140006      MOV      #140006,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1.
676 032304 012721 026500      MOV      #T29BF2,(R1)+      ;ADDRESS OF DATA BLOCK
677 032310 005021          CLR      (R1)+          ;EXTENDED ADDRESS
678 032312 012721 000006      MOV      #6,(R1)+          ;SIZE OF DATA BLOCK IN BYTES
679 032316 005021          CLR      (R1)+
680 032320 012701 026500      MOV      #T29BF2,R1          ;POINT TO DATA SEL AREA
681 032324 005021          CLR      (R1)+
682 032326 005011          CLR      (R1)+
683 032330 000207          RTS      PC          ;RETURN
684 032332          T29RT3:
685 032332          SAVREG          ;SAVE THE REGISTERS
686 032336 012701 026470      MOV      #T29PK3,R1          ;START OF THE PACKET
687 032342 012721 000000      MOV      #0,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK.
688 032346 012721 000000      MOV      #0,(R1)+      ;ADDRESS OF DATA BLOCK
689 032352 005021          CLR      (R1)+          ;EXTENDED ADDRESS
690 032354 012711 000000      MOV      #0,(R1)          ;SIZE OF DATA BLOCK IN BYTES
691 032360 000207          RTS      PC          ;RETURN
692 032362          ENDTST
        032362          L10036:
        032362 104401          TRAP      C#ETST

```

```

693          .SBTTL TEST 2: SKIP TAPE MARKS
694          ;*
695          ;
696          ;THIS TEST VERIFIES PROPER OPERATION OF THE SKIP TAPE MARKS
697          ;FORWARD AND SKIP TAPE MARKS REVERSE COMMANDS. PROPER OPERATION
698          ;UNDER CONTROL OF ALL COMBINATIONS OF THE ENABLE SKIP TAPE MARKS
699          ;STOP (ESS) AND ENABLE TAPE MARKS STOP OFF BOT (ENB) BITS SPECIFIED
700          ;BY THE WRITE CHARACTERISTICS COMMAND. THE TEST CONSISTS OF THE
701          ;FOLLOWING SUBTESTS (FOR EACH SUBTEST, THE TAPE IS FIRST WRITTEN
702          ;WITH AN APPROPRIATE SERIES OF DATA RECORDS AND/OR TAPE MARKS
703          ;AND/OR DOUBLE TAPE MARKS.
704          ;
705          ;
706          ;THE TEST CONSISTS OF THE FOLLOWING 11 SUBTESTS
707          ;
708          ;
709          ;
710          ;-
711          BGNTST
712 032364 012737 006354 002172      MOV      #EPRT1,EPRTSW      ;PRIMARY ERROR MESSAGE
717 032372 012700 041261          MOV      #TST30ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
718 032376 004737 016570          JSR      PC,TSTSETUP      ;DO INITIAL TEST SETUP
719 032402 012737 000005 002210      MOV      #5,LOOPCNT      ;PERFORM 5 ITERATIONS
720          ;*
721          ;
722          ;TEST 2, SUBTEST 1
723          ;
724          ;
725          ;VERIFIES THAT A SKIP TAPE MARKS FORWARD COMMAND WITH
726          ;A TAPE MARK COUNT OF 1 OPERATES PROPERLY. THE TAPE
727          ;IS FIRST REWOUND, THEN WRITTEN WITH SEVERAL "FILES";
728          ;EACH FILE CONSISTS OF A NUMBER OF DATA RECORDS
729          ;FOLLOWED BY A TAPE MARK. THE FINAL FILE IS
730          ;TERMINATED BY A DOUBLE TAPE MARK. EACH DATA RECORD

```


TEST 2: SKIP TAPE MARKS

```

777 032542 005237 002214          INC      FATFLG          ;ERROR COUNT
781 032546 010001          MOV      RO,R1          ;SAVE CONTENTS OF TSSR
782 032550          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      032550 104456          TRAP    C$ERHRD
      032552 000312          .WORD  202
      032554 005052          .WORD  WRTMSG
      032556 012114          .WORD  SFIMSG
783 032560          23$:   CKLOOP          ;LOOP IF SELECTED
      032560 104406          TRAP    C$CLP1
784
785          ;*****
786          ;
787          ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
788          ;
789          ;*****
790
791 032562 004737 011074          JSR     PC,REWIND      ;CALL TAPE REWIND COMMAND
792 032566 103411          BCS    30$            ;BR, IF NO PROBLEM
793 032570 010004          MOV     RO,R4          ;GET PACKET ADDRESS
794 032572 016501 000002          MOV     TSSR(R5),R1   ;GET STATUS REGISTER
795 032576 005237 002214          INC     FATFLG          ;ERROR COUNT
799 032602          ERRHRD  ERRNO,T3ORWN,PKTSSR ;REWIND NOT ACCEPTED
      032602 104456          TRAP    C$ERHRD
      032604 000313          .WORD  203
      032606 040270          .WORD  T3ORWN
      032610 012126          .WORD  PKTSSR
800 032612          30$:   CKLOOP          ;LOOP IF SELECTED
      032612 104406          TRAP    C$CLP1
801
802          ;*****
803          ;
804          ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
805          ;
806          ;*****
807
808 032614 013701 036560          MOV     T3OBF+6,R1    ;PICK UP XSTO
809 032620 010102          MOV     R1,R2          ;SET UP EXPECTED
810 032622 052702 000002          BIS    #BIT1,R2       ;SET BOT BIT IN EXPECTED
811 032626 020102          CMP     R1,R2          ;DOES EXP = REC'D
812 032630 001406          BEQ    40$            ;BR, IF EQUAL (OK)
813 032632 005237 002214          INC     FATFLG          ;ERROR COUNT
817 032636          ERRHRD  ERRNO,T3OBOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      032636 104456          TRAP    C$ERHRD
      032640 000314          .WORD  204
      032642 040071          .WORD  T3OBOT
      032644 015554          .WORD  EXPREC
818 032646          40$:   CKLOOP          ;LOOP IF SELECTED
      032646 104406          TRAP    C$CLP1
819 032650 012737 000001 036704          MOV     #1.,T30FCN    ;SET "FILE" COUNTER AT 1 DECIMAL
820 032656 012703 000001          64$:   MOV     #1,R3    ;ONE RECORD PER "FILE"
821 032662 013737 003116 036652          65$:   MOV     FREE,T30WB ;SET UP PACKETS'S WRITE BUFFER
822 032670 012737 003720 036656          MOV     #2000.,T30SZ ;SET RECORD SIZE AT 2000 BYTES
823
824          ;*****
825          ;
826          ;WRITE DATA,ACK,CVC=1 COMMAND
827          ;

```

TEST 2: SKIP TAPE MARKS

```

828 ;*****
829
830 032676 012737 140005 036650      MOV      #140005,T30PK3      ;WRITE DATA,ACK,CVC=1 COMMAND
831 032704 012704 036650              MOV      #T30PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
832 032710 013702 036704              MOV      T30FCN,R2        ;GET FILE COUNTER
833 032714 000302                      SWAB     R2                ;MOVE TO UPPER BYTE
834 032716 010301                      MOV      R3,R1            ;GET RECORD COUNTER
835 032720 060201                      ADD      R2,R1            ;FILE COUNTER IN UPPER, RECORD # LOW
836 032722 010177 150170              MOV      R1,#FREE        ;MOV TO OUT PUT BUFFER
837 032726 010465 000000              MOV      R4,TSD8(R5)     ;ISSUE COMMAND
838 032732 004737 016330              JSR      PC,WAITF        ;WAIT FOR SSR TO SET
839 032736 016501 000002              MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
840 032742 012702 000200              MOV      #SSR,R2        ;SET UP EXPECTED
841 032746 020102                      CMP      R1,R2           ;ARE THEY EQUAL
842 032750 001406                      BEQ      70$             ;BR, IF OK
843 032752 005237 002214              INC      FATFLG          ;ERROR COUNT
847 032756                      ERRHRD  ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP  C$ERHRD
                                .WORD 205
                                .WORD T30WDD
                                .WORD PKTSSR
                                TRAP  C$CLP1
848 032766 104456                      70$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP  C$CLP1
849 032770 005203                      INC      R3              ;COUNT THE RECORD COUNTER DOWN
850 032772 020327 000021              CMP      R3,#21         ;AT 20 YET
851 032776 001331                      BNE     65$             ;BR, IF NOT AT 20 RECORDS WRITTEN
852
853 ;*****
854 ;
855 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
856 ;
857 ;*****
858
859 033000 012737 141011 036650      MOV      #141011,T30PK3  ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
860 033006 012704 036650              MOV      #T30PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
861 033012 010465 000000              MOV      R4,TSD8(R5)   ;ISSUE COMMAND
862 033016 004737 016330              JSR      PC,WAITF      ;WAIT FOR SSR TO SET
863 033022 016501 000002              MOV      TSSR(R5),R1  ;PICK UP TSSR
864 033026 012702 000200              MOV      #SSR,R2      ;SET UP EXPECTED (SSR ONLY)
865 033032 020102                      CMP      R1,R2        ;WAS STATUS GOOD
866 033034 001406                      BEQ      160$         ;BR, IF TERMINATION WAS GOOD
867 033036 005237 002214              INC      FATFLG       ;ERROR COUNT
871 033042                      ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
                                TRAP  C$ERHRD
                                .WORD 206
                                .WORD T30WDC
                                .WORD PKTSSR
                                TRAP  C$CLP1
872 033052 104406                      160$: CKLOOP              ;LOOP IF SELECTED
873 033054 005237 036704              INC      T30FCN        ;COUNT THE "FILE" COUNTER DOWN
874 033060 023727 036704 000006      CMP      T30FCN,#6     ;WRITE 5 FILE TO TAPE
875 033066 001273                      BNE     64$           ;BR, IF NOT AT 5 FILES WRITTEN
876
877 ;*****
878 ;
879 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
880 ;

```


TEST 2: SKIP TAPE MARKS

```

881
882
883 033070 012737 141011 036650      MOV      #141011,T30PK3      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
884 033076 012704 036650              MOV      #T30PK3,R4         ;SET UP R4 WITH PACKET ADDRESS
885 033102 010465 000000              MOV      R4,TSDB(R5)        ;ISSUE COMMAND
886 033106 004737 016330              JSR      PC,WAITF           ;WAIT FOR SSR TO SET
887 033112 016501 000002              MOV      TSSR(R5),R1        ;PICK UP TSSR
888 033116 012702 000200              MOV      #SSR,R2           ;SET UP EXPECTED (SSR ONLY)
889 033122 020102                      CMP      R1,R2              ;WAS STATUS GOOD
890 033124 001406                      BEQ      165$               ;BR, IF TERMINATION WAS GOOD
891 033126 005237 002214              INC      FATFLG              ;ERROR COUNT
895 033132                      ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
                                TRAP    C$ERHRD
                                .WORD   207
                                .WORD   T30WDC
                                .WORD   PKTSSR
033132 104456
033134 000317
033136 040412
033140 012126
896 033142 165$: CKLOOP                  ;LOOP IF SELECTED
033142 104406                          TRAP    C$CLP1
897
898
899
900
901
902
903
904 033144 004737 011074              JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
905 033150 103411                      BCS      170$               ;BR, IF NO PROBLEM
906 033152 010004                      MOV      R0,R4              ;GET PACKET ADDRESS
907 033154 016501 000002              MOV      TSSR(R5),R1        ;GET STATUS REGISTER
908 033160 005237 002214              INC      FATFLG              ;ERROR COUNT
912 033164                      ERRHRD  ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD   208
                                .WORD   T30RWN
                                .WORD   PKTSSR
033164 104456
033166 000320
033170 040270
033172 012126
913 033174 170$: CKLOOP                  ;LOOP IF SELECTED
033174 104406                          TRAP    C$CLP1
914
915
916
917
918
919
920
921 033176 013701 036560              MOV      T30BFR+6,R1        ;PICK UP XSTO
922 033202 010102                      MOV      R1,R2              ;SET UP EXPECTED
923 033204 052702 000002              BIS      #BIT1,R2           ;SET BOT BIT IN EXPECTED
924 033210 020102                      CMP      R1,R2              ;DOES EXP = REC'D
925 033212 001406                      BEQ      180$               ;BR, IF EQUAL (OK)
926 033214 005237 002214              INC      FATFLG              ;ERROR COUNT
930 033220                      ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD   209
                                .WORD   T30BOT
                                .WORD   EXPREC
033220 104456
033222 000321
033224 040071
033226 015554
931 033230 180$: CKLOOP                  ;LOOP IF SELECTED
033230 104406                          TRAP    C$CLP1

```

TEST 2: SKIP TAPE MARKS

```

932 033232 012703 036666          MOV    #T30IMV,R3          ;SET UP POINTER TO COMMAND TABLE
933 033236 013737 002174 036550  MOV    UNITN,T30DSW      ;SET UP UNIT NUMBER
934 033244 011337 036546          MOV    (R3),T30ETM      ;GET NEXT COMMAND
935 033250 012704 036530          MOV    #T30PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
936
937
938
939
940
941
942
943 033254 004737 010742          JSR    PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
944 033260 103407          BCS    188$            ;BR, IF COMMAND ISSUED OK
945 033262 005237 002214          INC    FATFLG          ;ERROR COUNT
949 033266 010001          MOV    R0,R1           ;SAVE CONTENTS OF TSSR
950 033270          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
          033270 104456          TRAP   C$ERHRD
          033272 000322          .WORD 210
          033274 005052          .WORD WRTMSG
          033276 012114          .WORD SFIMSG
951 033300          188$:  CKLOOP          ;LOOP IF SELECTED
          033300 104406          TRAP   C$CLP1
952
953
954
955
956
957
958
959 033302 012737 141010 036650      MOV    #141010,T30PK3   ;SKIP TAPE MARK,ACK,CVC=1 COMMAND
960 033310 012737 000001 036652      MOV    #1,T30RB        ;SET UP NUMBER TO SKIP
961 033316 012704 036650          MOV    #T30PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
962 033322 010465 000000          MOV    R4,TSDB(R5)     ;ISSUE COMMAND
963 033326 012737 176750 036706      MOV    #65000.,T30DLY  ;SET UP DELAY COUNTER
964 033334 004737 016330          190$: JSR    PC,WAITF     ;WAIT FOR SSR TO SET
965 033340 016501 000002          MOV    TSSR(R5),R1     ;PICK UP TSSR
966 033344 032701 000200          BIT    #SSR,R1         ;IS SSR SET YET
967 033350 001017          BNE    191$            ;BR, IF SSR IS SET
968 033352          DELAY 250             ;CALL DELAY ROUTINE
          033352 012727 000250          MOV    #250,(PC)+
          033356 000000          .WORD 0
          033360 013727 002116          MOV    L$DLY,(PC)+
          033364 000000          .WORD 0
          033366 005367 177772          DEC    -6(PC)
          033372 001375          BNE    -4
          033374 005367 177756          DEC    -22(PC)
          033400 001367          BNE    -20
969 033402 005337 036706          DEC    T30DLY          ;BUMP DELAY ROUTINE
970 033406 001352          BNE    190$            ;BR, IF MORE DELAY TO GO
971 033410 012702 000200          191$: MOV    #SSR,R2      ;SET UP EXPECTED (SSR ONLY)
972 033414 020102          CMP    R1,R2           ;WAS STATUS GOOD
973 033416 001406          BEQ    192$            ;BR, IF TERMINATION WAS GOOD
974 033420 005237 002214          INC    FATFLG          ;ERROR COUNT
978 033424          ERRHRD  ERRNO,T30SKM,PKTSSR ;TSSR NOT CORRECT AFTER SKIP TAPE M.
          033424 104456          TRAP   C$ERHRD
          033426 000323          .WORD 211
          033430 037144          .WORD T30SKM

```

TEST 2: SKIP TAPE MARKS

```

979 033432 012126
033434 104406
980
981
982
983
984
985
986
987 033436 013701 036560
988 033442 010102
989 033444 052702 100000
990 033450 020102
991 033452 001406
992 033454 005237 002214
996 033460
033460 104456
033462 000324
033464 040544
033466 015554
997 033470
033470 104406
998 033472 012700 177777
999 033476 004737 017502
1000 033502 013737 003116 036652
1001
1002
1003
1004
1005
1006
1007
1008 033510 012737 140001 036650
1009 033516 012704 036650
1010 033522 012737 003720 036656
1011 033530 010465 000000
1012 033534 004737 016330
1013 033540 016501 000002
1014 033544 012702 000200
1015 033550 020102
1016 033552 001406
1017 033554 005237 002214
1021 033560
033560 104456
033562 000325
033564 037443
033566 012126
1022 033570
033570 104406
1023 033572 017701 147320
1024 033576 012702 177777
1025 033602 020102
1026 033604 001006
1027 033606 005237 002214
1031 033612
033612 104456

192$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
TRAP C$CLP1

;*****
;
;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
;
;*****
MOV T30BFR+6,R1 ;PICK UP XSTO
MOV R1,R2 ;SET UP EXPECTED
BIS #BIT15,R2 ;SET TMK BIT IN EXPECTED
CMP R1,R2 ;DOES EXP = REC'D
BEQ 195$ ;BR, IF EQUAL (OK)
INC FATFLG ;ERROR COUNT
ERRHRD ERRNO,T30TMK,EXPREC ;TMK NOT SET AFTER WRT TAPE MARK
TRAP C$ERHRD
.WORD 212
.WORD T30TMK
.WORD EXPREC

195$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
MOV #177777,R0 ;VALUE TO WRITTEN TO MEMORY
JSR PC,FILLMEM ;FILL MEM WITH ALL ONES
MOV FREE,T30RB ;STARTING READ BUFFER ADDRESS

;*****
;
;READ FORWARD,ACK,CVC=1 COMMAND
;
;*****
MOV #140001,T30PK3 ;READ FORWARD,ACK,CVC=1 COMMAND
MOV #T30PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
MOV #2000.,T30SZ ;SET UP RECORD SIZE IN PACKET
MOV R4,TSD8(R5) ;ISSUE COMMAND
JSR PC,WAITF ;WAIT FOR SSR TO SET
MOV TSSR(R5),R1 ;GET TSSR CONTENTS
MOV #SSR,R2 ;SET UP EXPECTED
CMP R1,R2 ;ARE THEY EQUAL
BEQ 200$ ;BR, IF OK
INC FATFLG ;ERROR COUNT
ERRHRD ERRNO,T30RDF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
TRAP C$ERHRD
.WORD 213
.WORD T30RDF
.WORD PKTSSR

200$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
MOV #FREE,R1 ;FIRST LOC IN READ BUFFER
MOV #177777,R2 ;EXPECTED IF NO DATA TRANS.
CMP R1,R2 ;DID ANY DATA GET TRANSFERRED
BNE 220$ ;BR, IF NO DATA TRANS (GOOD)
INC FATFLG ;ERROR COUNT
ERRHRD ERRNO,T30DTR,EXPREC ;DATA TRANSFERRED ON READ TAPE MARK
TRAP C$ERHRD

```

TEST 2: SKIP TAPE MARKS

```

033614 000326 .WORD 214
033616 041120 .WORD T30DTR
033620 015554 .WORD EXPREC
1032 033622 220$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
033622 104406 ;SET UP RECORD NUMBER EXPECTED (FILE 2)
1033 033624 012702 001001 MOV #1001,R2
1034 033630 017701 147262 MOV @FREE,R1 ;GET INFO FROM BUFFER
1035 033634 020201 CMP R2,R1 ;ARE THEY EQUAL
1036 033636 001406 BEQ 228$ ;BR, IF EQUAL (OK)
1037 033640 005237 002214 INC FATFLG ;ERROR COUNT
1041 033644 ERRHRD ERRNO,T30PTB,EXPREC ;RECORD POSITION WAS NOT CORRECT
033644 104456 TRAP C$ERHRD
033646 000327 .WORD 215
033650 037272 .WORD T30PTB
033652 015554 .WORD EXPREC
1042 033654 228$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
033654 104406
1043
1044 ;*****
1045 ;
1046 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1047 ;
1048 ;*****
1049
1050 033656 004737 011074 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
1051 033662 103411 BCS 230$ ;BR, IF NO PROBLEM
1052 033664 010004 MOV R0,R4 ;SAVE PACKET ADDRESS
1053 033666 016501 000002 MOV TSSR(R5),R1 ;GET TSSR STATUS
1054 033672 005237 002214 INC FATFLG ;ERROR COUNT
1058 033676 ERRHRD ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
033676 104456 TRAP C$ERHRD
033700 000330 .WORD 216
033702 040270 .WORD T30RWN
033704 012126 .WORD PKTSSR
1059 033706 230$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
033706 104406
1060
1061 ;*****
1062 ;
1063 ;GET EXTENDED STATUS REGISTER ZERO (XST0) FROM MESSAGE BUFFER
1064 ;
1065 ;*****
1066
1067 033710 013701 036560 MOV T30BFR+6,R1 ;PICK UP XST0
1068 033714 010102 MOV R1,R2 ;SET UP EXPECTED
1069 033716 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
1070 033722 020102 CMP R1,R2 ;DOES EXP = REC'D
1071 033724 001406 BEQ 240$ ;BR, IF EQUAL (OK)
1072 033726 005237 002214 INC FATFLG ;ERROR COUNT
1076 033732 ERRHRD ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
033732 104456 TRAP C$ERHRD
033734 000331 .WORD 217
033736 040071 .WORD T30BOT
033740 015554 .WORD EXPREC
1077 033742 240$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
033742 104406
1078 033744 005723 TST (R3)+ ;POINT TO NEXT POSITION

```


TEST 2: SKIP TAPE MARKS

```

1122
1123
1124
1125
1126
1127
1128
1129 034126 004737 010742          JSR    PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
1130 034132 103407                  BCS    23$                ;BR, IF COMMAND ISSUED OK
1131 034134 005237 002214          INC    FATFLG             ;ERROR COUNT
1135 034140 010001                  MOV    R0,R1              ;SAVE CONTENTS OF TSSR
1136 034142                  ERRHRD ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTIISC FAILED
                                TRAP    C$ERHRD
                                .WORD   219
                                .WORD   WRTMSG
                                .WORD   SFMSG
                                034142 104456
                                034144 000333
                                034146 005052
                                034150 012114
1137 034152                  23$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                034152 104406

1138
1139
1140
1141
1142
1143
1144
1145 034154 004737 011074          JSR    PC,REWIND          ;CALL TAPE REWIND COMMAND
1146 034160 103411                  BCS    30$                ;BR, IF NO PROBLEM
1147 034162 010004                  MOV    R0,R4              ;GET PACKET ADDRESS
1148 034164 016501 000002          MOV    TSSR(R5),R1        ;GET STATUS REGISTER
1149 034170 005237 002214          INC    FATFLG             ;ERROR COUNT
1153 034174                  ERRHRD ERRNO,T3ORWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD   220
                                .WORD   T3ORWN
                                .WORD   PKTSSR
                                034174 104456
                                034176 000334
                                034200 040270
                                034202 012126
1154 034204                  30$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                034204 104406

1155
1156
1157
1158
1159
1160
1161
1162 034206 013701 036560          MOV    T30BFR+6,R1        ;PICK UP XSTO
1163 034212 010102                  MOV    R1,R2              ;SET UP EXPECTED
1164 034214 052702 000002          BIS    #BIT1,R2           ;SET BOT BIT IN EXPECTED
1165 034220 020102                  CMP    R1,R2              ;DOES EXP = REC'D
1166 034222 001406                  BEQ    40$                ;BR, IF EQUAL (OK)
1167 034224 005237 002214          INC    FATFLG             ;ERROR COUNT
1171 034230                  ERRHRD ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD   221
                                .WORD   T30BOT
                                .WORD   EXPREC
                                034230 104456
                                034232 000335
                                034234 040071
                                034236 015554
1172 034240                  40$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                034240 104406

```

TEST 2: SKIP TAPE MARKS

```

1173 034242 012737 000001 036704      MOV      #1.,T30FCN      ;SET "FILE" COUNTER AT 1 DECIMAL
1174 034250 012703 000001      64$:    MOV      #1,R3      ;ONE RECORD PER "FILE"
1175 034254 013737 003116 036652 65$:    MOV      FREE,T30WB     ;SET UP PACKETS'S WRITE BUFFER
1176 034262 012737 000024 036656      MOV      #20.,T30SZ     ;SET RECORD SIZE AT 2000 BYTES
1177
1178      ;*****
1179      ;
1180      ;WRITE DATA,ACK,CVC=1 COMMAND
1181      ;
1182      ;*****
1183
1184 034270 012737 140005 036650      MOV      #140005,T30PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
1185 034276 012704 036650      MOV      #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
1186 034302 013702 036704      MOV      T30FCN,R2     ;GET FILE COUNTER
1187 034306 000302      SWAB     R2             ;MOVE TO UPPER BYTE
1188 034310 010301      MOV      R3,R1         ;GET RECORD COUNTER
1189 034312 060201      ADD      R2,R1         ;FILE COUNTER IN UPPER, RECORD # LOW
1190 034314 010177 146576      MOV      R1,#FREE      ;MOV TO OUT PUT BUFFER
1191 034320 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
1192 034324 004737 016330      JSR     PC,WAITF       ;WAIT FOR SSR TO SET
1193 034330 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
1194 034334 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
1195 034340 020102      CMP      R1,R2        ;ARE THEY EQUAL
1196 034342 001406      BEQ     70$           ;BR, IF OK
1197 034344 005237 002214      INC      FATFLG        ;ERROR COUNT
1201 034350      ERRHRD  ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      034350 104456      TRAP    C$ERHRD
      034352 000336      .WORD   222
      034354 037220      .WORD   T30WDD
      034356 012126      .WORD   PKTSSR
1202 034360      70$:    CKLOOP           ;LOOP IF SELECTED
      034360 104406      TRAP    C$CLP1
1203 034362 005203      INC      R3           ;COUNT THE RECORD COUNTER DOWN
1204 034364 020327 000021      CMP      R3,#21       ;AT 20 YET
1205 034370 001331      BNE     65$           ;BR, IF NOT AT 20 RECORDS WRITTEN
1206
1207      ;*****
1208      ;
1209      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1210      ;
1211      ;*****
1212
1213 034372 012737 141011 036650      MOV      #141011,T30PK3 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1214 034400 012704 036650      MOV      #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
1215 034404 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
1216 034410 004737 016330      JSR     PC,WAITF       ;WAIT FOR SSR TO SET
1217 034414 016501 000002      MOV      TSSR(R5),R1   ;PICK UP TSSR
1218 034420 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED (SSR ONLY)
1219 034424 020102      CMP      R1,R2        ;WAS STATUS GOOD
1220 034426 001406      BEQ     160$          ;BR, IF TERMINATION WAS GOOD
1221 034430 005237 002214      INC      FATFLG        ;ERROR COUNT
1225 034434      ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
      034434 104456      TRAP    C$ERHRD
      034436 000337      .WORD   223
      034440 040412      .WORD   T30WDC
      034442 012126      .WORD   PKTSSR
1226 034444      160$:  CKLOOP           ;LOOP IF SELECTED

```

TEST 2: SKIP TAPE MARKS

```

1227 034444 104406                                TRAP C$CLP1
1228 034446 005237 036704 000031                INC T30FCN ;COUNT THE "FILE" COUNTER DOWN
1229 034460 001273 036704 000031                CMP T30FCN,#25. ;WRITE 25 FILES TO TAPE
1230 034460 001273 036704 000031                BNE 64$ ;BR, IF NOT AT 25 FILES WRITTEN
1231 ;*****
1232 ;
1233 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1234 ;
1235 ;*****
1236
1237 034462 012737 141011 036650                MOV #141011,T30PK3 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1238 034470 012704 036650 036650                MOV #T30PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
1239 034474 010465 000000 036650                MOV R4,T5DB(R5) ;ISSUE COMMAND
1240 034500 004737 016330 036650                JSR PC,WAITF ;WAIT FOR SSR TO SET
1241 034504 016501 000002 036650                MOV T5SR(R5),R1 ;PICK UP T5SR
1242 034510 012702 000200 036650                MOV #5SR,R2 ;SET UP EXPECTED (SSR ONLY)
1243 034514 020102 000200 036650                CMP R1,R2 ;WAS STATUS GOOD
1244 034516 001406 000200 036650                BEQ 165$ ;BR, IF TERMINATION WAS GOOD
1245 034520 005237 002214 036650                INC FATFLG ;ERROR COUNT
1249 034524 005237 002214 036650                ERRHRD ERRNO,T30WDC,PKT5SR ;T5SR NOT CORRECT AFTER WRT TAPE M.
                                TRAP C$ERRHRD
                                .WORD 224
                                .WORD T30WDC
                                .WORD PKT5SR
1250 034534 165$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
034534 104406
1251 ;*****
1252 ;
1253 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1254 ;
1255 ;*****
1256
1257
1258 034536 004737 011074 036650                JSR PC,REWIND ;CALL TAPE REWIND COMMAND
1259 034542 103411 011074 036650                BCS 170$ ;BR, IF NO PROBLEM
1260 034544 010004 011074 036650                MOV R0,R4 ;GET PACKET ADDRESS
1261 034546 016501 000002 036650                MOV T5SR(R5),R1 ;GET STATUS REGISTER
1262 034552 005237 002214 036650                INC FATFLG ;ERROR COUNT
1266 034556 005237 002214 036650                ERRHRD ERRNO,T30RWN,PKT5SR ;REWIND NOT ACCEPTED
                                TRAP C$ERRHRD
                                .WORD 225
                                .WORD T30RWN
                                .WORD PKT5SR
1267 034566 170$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
034566 104406
1268 ;*****
1269 ;
1270 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1271 ;
1272 ;*****
1273
1274
1275 034570 013701 036560 036560                MOV T30BFR+6,R1 ;PICK UP XSTO
1276 034574 010102 036560 036560                MOV R1,R2 ;SET UP EXPECTED
1277 034576 052702 000002 036560                BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
1278 034602 020102 000002 036560                CMP R1,R2 ;DOES EXP = REC'D

```


TEST 2: SKIP TAPE MARKS

```

1279 034604 001406          BEQ     180$          ;BR, IF EQUAL (OK)
1280 034606 005237 002214    INC     FATFLG        ;ERROR COUNT
1284 034612          ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP     C$ERHRD
                                .WORD    226
                                .WORD    T30BOT
                                .WORD    EXPREC
1285 034622          180$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP1
1286 034624 012737 000002 036704    MOV     #2,T30FCN     ;SET TO NUMBER OF SKIP "FILES"
1287 034632 012703 036666          MOV     #T30IMV,R3    ;SET UP POINTER TO COMMAND TABLE
1288 034636 013737 002174 036550    MOV     UNITN,T30DSW  ;SET UP UNIT NUMBER
1289 034644 011337 036546          182$:  MOV     (R3),T30ETM ;GET NEXT COMMAND
1290 034650 012704 036530          MOV     #T30PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
1291
1292          ;*****
1293          ;
1294          ;ISSUE WRITE CHARACTERISTICS COMMAND
1295          ;
1296          ;*****
1297
1298 034654 004737 010742          JSR     PC,WRTCHR     ;ISSUE WRITE CHARACTERISTICS
1299 034660 103407          BCS     188$          ;BR, IF COMMAND ISSUED OK
1300 034662 005237 002214    INC     FATFLG        ;ERROR COUNT
1304 034666 010001          MOV     R0,R1         ;SAVE CONTENTS OF TSSR
1305 034670          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP     C$ERHRD
                                .WORD    227
                                .WORD    WRTMSG
                                .WORD    SFIMSG
1306 034700          188$:  CKLOOP          ;LOOP IF SELECTED
1307 034700 104406          TRAP     C$CLP1
1308
1309          ;*****
1310          ;
1311          ;SKIP TAPE MARK,ACK,CVC=1 COMMAND
1312          ;
1313          ;*****
1314 034702 012737 141010 036650    MOV     #141010,T30PK3 ;SKIP TAPE MARK,ACK,CVC=1 COMMAND
1315 034710 013737 036704 036652    MOV     T30FCN,T30RB  ;SET UP NUMBER TO SKIP
1316 034716 012704 036650          MOV     #T30PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
1317 034722 010465 000000          189$:  MOV     R4,T30DB(R5) ;ISSUE COMMAND
1318 034726 012737 176750 036706    MOV     #65000.,T30DLY ;SET UP DELAY COUNTER
1319 034734 004737 016330          190$:  JSR     PC,WAITF    ;WAIT FOR SSR TO SET
1320 034740 016501 000002          MOV     TSSR(R5),R1   ;PICK UP TSSR
1321 034744 032701 000200          BIT     #SSR,R1       ;IS SSR SET YET
1322 034750 001017          BNE     191$          ;BR, IF SSR IS SET
1323 034752          DELAY  250        ;CALL DELAY ROUTINE
                                MOV     #250,(PC)+
                                .WORD    0
                                MOV     L$DLY,(PC)+
                                .WORD    0
                                DEC     -6(PC)
                                BNE     -.4
                                DEC     -22(PC)
                                BNE     .-20
034752 012727 000250
034756 000000
034760 013727 002116
034764 000000
034766 005367 177772
034772 001375
034774 005367 177756
035000 001367

```

TEST 2: SKIP TAPE MARKS

```

1324 035002 005337 036706          DEC      T30DLY          ;BUMP DELAY ROUTINE
1325 035006 001352                  BNE      190$          ;BR, IF MORE DELAY TO GO
1326 035010 012702 000200          191$:  MOV     #SSR,R2    ;SET UP EXPECTED (SSR ONLY)
1327 035014 020102                  CMP     R1,R2         ;WAS STATUS GOOD
1328 035016 001406                  BEQ     192$          ;BR, IF TERMINATION WAS GOOD
1329 035020 005237 002214          INC     FATFLG        ;ERROR COUNT
1333 035024                  ERRHRD  ERRNO,T30SKM,PKTSSR ;TSSR NOT CORRECT AFTER SKIP TAPE M.
                                TRAP  C$ERHRD
                                .WORD 228
                                .WORD T30SKM
                                .WORD PKTSSR
                                TRAP  C$CLP1
1334 035034 104456          192$:  CKLOOP          ;LOOP IF SELECTED
035024 104456
035026 000344
035030 037144
035032 012126
1334 035034 104406          TRAP  C$CLP1
035034 104406
1335
1336          ;*****
1337          ;
1338          ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1339          ;
1340          ;*****
1341
1342 035036 013701 036560          MOV     T30BFR+6,R1   ;PICK UP XSTO
1343 035042 010102          MOV     R1,R2         ;SET UP EXPECTED
1344 035044 052702 100000          BIS     #BIT15,R2    ;SET TMK BIT IN EXPECTED
1345 035050 020102          CMP     R1,R2         ;DOES EXP = REC'D
1346 035052 001406          BEQ     195$          ;BR, IF EQUAL (OK)
1347 035054 005237 002214          INC     FATFLG        ;ERROR COUNT
1351 035060          ERRHRD  ERRNO,T30TMK,EXPREC ;TMK NOT SET AFTER WRT TAPE MARK
                                TRAP  C$ERHRD
                                .WORD 229
                                .WORD T30TMK
                                .WORD EXPREC
                                TRAP  C$CLP1
1352 035070 104406          195$:  CKLOOP          ;LOOP IF SELECTED
035070 104406
1353 035072 012700 177777          MOV     #177777,R0    ;VALUE TO WRITTEN TO MEMORY
1354 035076 004737 017502          JSR     PC,FILLMEM    ;FILL MEM WITH ALL ONES
1355 035102 013737 003116 036652          MOV     FREE,T30RB    ;STARTING READ BUFFER ADDRESS
1356
1357          ;*****
1358          ;
1359          ;READ FORWARD,ACK,CVC=1 COMMAND
1360          ;
1361          ;*****
1362
1363 035110 012737 140001 036650          MOV     #140001,T30PK3 ;READ FORWARD,ACK,CVC=1 COMMAND
1364 035116 012704 036650          MOV     #T30PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
1365 035122 012737 000024 036656          MOV     #20.,T30SZ    ;SET UP RECORD SIZE IN PACKET
1366 035130 010465 000000          MOV     R4,TSDR(R5)   ;ISSUE COMMAND
1367 035134 004737 016330          JSR     PC,WAITF      ;WAIT FOR SSR TO SET
1368 035140 016501 000002          MOV     TSSR(R5),R1   ;GET TSSR CONTENTS
1369 035144 012702 000200          MOV     #SSR,R2       ;SET UP EXPECTED
1370 035150 020102          CMP     R1,R2         ;ARE THEY EQUAL
1371 035152 001406          BEQ     200$          ;BR, IF OK
1372 035154 005237 002214          INC     FATFLG        ;ERROR COUNT
1376 035160          ERRHRD  ERRNO,T30RDF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP  C$ERHRD
                                .WORD 230
                                .WORD T30RDF
035160 104456
035162 000346
035164 037443

```

TEST 2: SKIP TAPE MARKS

1377	035166	012126		200\$:	CKLOOP		;LOOP IF SELECTED	.WORD	PKTSSR
	035170	104406						TRAP	C\$CLP1
1378	035172	017701	145720		MOV	@FREE,R1	;FIRST LOC IN READ BUFFER		
1379	035176	012702	177777		MOV	@177777,R2	;EXPECTED IF NO DATA TRANS.		
1380	035202	020102			CMP	R1,R2	;DID ANY DATA GET TRANSFERRED		
1381	035204	001006			BNE	220\$;BR, IF NO DATA TRANS (GOOD)		
1382	035206	005237	002214		INC	FATFLG	;ERROR COUNT		
1386	035212				ERRHRD	ERRNO,T30DTR,EXPREC	;DATA TRANSFERRED ON READ TAPE MARK		
	035212	104456						TRAP	C\$ERHRD
	035214	000347						.WORD	231
	035216	041120						.WORD	T30DTR
	035220	015554						.WORD	EXPREC
1387	035222			220\$:	CKLOOP		;LOOP IF SELECTED		
	035222	104406						TRAP	C\$CLP1
1388	035224	013702	036704		MOV	T30FCN,R2	;GET NUMBER OF SKIPS		
1389	035230	005202			INC	R2	;SET TO CORRECT FILE VALUE		
1390	035232	000302			SWAB	R2	;SWAP BYTE HALVES		
1391	035234	052702	000001		BIS	@BIT0,R2	;SET FOR RECORD #1		
1392	035240	017701	145652		MOV	@FREE,R1	;GET INFO FROM BUFFER		
1393	035244	020201			CMP	R2,R1	;ARE THEY EQUAL		
1394	035246	001406			BEQ	228\$;BR, IF EQUAL (OK)		
1395	035250	005237	002214		INC	FATFLG	;ERROR COUNT		
1399	035254				ERRHRD	ERRNO,T30PTB,EXPREC	;RECORD POSITION WAS NOT CORRECT		
	035254	104456						TRAP	C\$ERHRD
	035256	000350						.WORD	232
	035260	037272						.WORD	T30PTB
	035262	015554						.WORD	EXPREC
1400	035264			228\$:	CKLOOP		;LOOP IF SELECTED		
	035264	104406						TRAP	C\$CLP1
1401									
1402									
1403									
1404									
1405									
1406									
1407									
1408	035266	004737	011074		JSR	PC,REWIND	;CALL TAPE REWIND COMMAND		
1409	035272	103411			BCS	230\$;BR, IF NO PROBLEM		
1410	035274	010004			MOV	R0,R4	;SAVE PACKET ADDRESS		
1411	035276	016501	000002		MOV	TSSR(R5),R1	;GET TSSR STATUS		
1412	035302	005237	002214		INC	FATFLG	;ERROR COUNT		
1416	035306				ERRHRD	ERRNO,T30RWN,PKTSSR	;REWIND NOT ACCEPTED		
	035306	104456						TRAP	C\$ERHRD
	035310	000351						.WORD	233
	035312	040270						.WORD	T30RWN
	035314	012126						.WORD	PKTSSR
1417	035316			230\$:	CKLOOP		;LOOP IF SELECTED		
	035316	104406						TRAP	C\$CLP1
1418									
1419									
1420									
1421									
1422									
1423									
1424									
1425	035320	013701	036560		MOV	T30BFR+6,R1	;PICK UP XSTO		

TEST 2: SKIP TAPE MARKS

Address	Command	Hex	Hex	Hex	Hex	Description	Trap	Trap Label
1426	MOV	035324	010102			MOV R1,R2		
1427	BIS	035326	052702	000002		BIS #BIT1,R2		
1428	CMP	035332	020102			CMP R1,R2		
1429	BEQ	035334	001406			BEQ 240\$		
1430	INC	035336	005237	002214		INC FATFLG		
1434	ERRHRD	035342				ERRHRD ERRNO,T30BOT,EXPREC		
		035342	104456				TRAP	C\$ERHRD
		035344	000352				.WORD	234
		035346	040071				.WORD	T30BOT
		035350	015554				.WORD	EXPREC
1435		035352			240\$:	CKLOOP		
		035352	104406					
1436		035354	005723			TST (R3)+		
1437	MOV	035356	011301			MOV (R3),R1		
1438	CMP	035360	020127	177777		CMP R1,#177777		
1439	BEQ	035364	001410			BEQ 330\$		
1440	MOV	035366	013701	036704		MOV T30FCN,R1		
1441	CLC	035372	000241			CLC		
1442	ROL	035374	006101			ROL R1		
1443	MOV	035376	010137	036704		MOV R1,T30FCN		
1444	JMP	035402	000137	034644		JMP 182\$		
1445		035406			330\$:	CKLOOP		
		035406	104406					
1446	ENDSUB	035410				ENDSUB		
		035410	104403				TRAP	C\$CLP1
1447	CMP	035412	023727	002214	000017	CMP FATFLG,#15.		
1448	BLO	035420	103402			BLO 999\$		
1449	JSR	035422	004737	017262		JSR PC,CKDROP		
1450		035426			999\$:			
1451					:			
1452					:			
1453					:	TEST 2, SUBTEST 3		
1454					:			
1455					:			
1456					:	VERIFIES THAT A SKIP TAPE MARKS REVERSE COMMAND		
1457					:	ISSUED WHILE THE TAPE IS POSITIONED AT BOT CAUSES		
1458					:	FUNCTION REJECT TERMINATION, WITH THE NON-EXECUTABLE		
1459					:	FUNCTION (NEF) ERROR BIT SET.		
1460					:			
1461					:			
1462					:			
1463					:			
1464					:			
1465					:			
1466	BGNSUB	035426				BGNSUB		
		035426	104402					
1467	JSR	035430	004737	041302		JSR PC,T30REST		
1468	CLR	035434	005037	036704		CLR T30FCN		
1469	JSR	035440	004737	041374		JSR PC,T30RT2		
1470	JSR	035444	004737	041436		JSR PC,T30RT3		
1471	MOV	035450	012737	176750	036706	MOV #65000.,T30DLY		
1472	JSR	035456	004737	016054	10\$:	JSR PC,SOFINIT		
1473	BCS	035462	103426			BCS 20\$		
1474	DELAY	035464	012727	000250		DELAY 250		
		035464					MOV	#250.(PC)+

TEST 2: SKIP TAPE MARKS

```

035470 000000
035472 013727 002116
035476 000000
035500 005367 177772
035504 001375
035506 005367 177756
035512 001367
1475 035514 005337 036706
1476 035520 001356
1477 035522 005237 002214
1481 035526 010001
1482 035530
035530 104455
035532 000353
035534 003646
035536 012114
1483 035540
1484 035540 013737 002174 036550
1485 035546 012704 036530
1486
1487
1488
1489
1490
1491
1492
1493 035552 004737 010742
1494 035556 103407
1495 035560 005237 002214
1499 035564 010001
1500 035566
035566 104456
035570 000354
035572 005052
035574 012114
1501 035576
035576 104406
1502
1503
1504
1505
1506
1507
1508
1509 035600 004737 011074
1510 035604 103411
1511 035606 010004
1512 035610 016501 000002
1513 035614 005237 002214
1517 035620
035620 104456
035622 000355
035624 040270
035626 012126
1518 035630
035630 104406
1519

```

```

                                .WORD 0
                                MOV L$DLY,(PC)
                                .WORD 0
                                DEC -6(PC)
                                BNE -.4
                                DEC -22(PC)
                                BNE -.20
                                DEC T3ODLY ;BUMP COUNTER
                                BNE 10$ ;BR, IF MORE COUNTING TO DO
                                INC FATFLG ;ERROR COUNT
                                MOV R0,R1 ;CONTENTS OF TSSR REGISTER
                                ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                TRAP C$ERDF
                                .WORD 235
                                .WORD SFIERR
                                .WORD SFIMSG
20$: MOV UNITN,T3ODSW ;SET UP UNIT NUMBER
      MOV @T3OPACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
      ;*****
      ;
      ;ISSUE WRITE CHARACTERISTICS COMMAND
      ;
      ;*****
      JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
      BCS 23$ ;BR, IF COMMAND ISSUED OK
      INC FATFLG ;ERROR COUNT
      MOV R0,R1 ;SAVE CONTENTS OF TSSR
      ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      TRAP C$ERHRD
      .WORD 236
      .WORD WRTMSG
      .WORD SFIMSG
1501 23$: CKLOOP ;LOOP IF SELECTED
      TRAP C$CLP1
      ;*****
      ;
      ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
      ;
      ;*****
      JSR PC,REWIND ;CALL TAPE REWIND COMMAND
      BCS 30$ ;BR, IF NO PROBLEM
      MOV R0,R4 ;GET PACKET ADDRESS
      MOV TSSR(R5),R1 ;GET STATUS REGISTER
      INC FATFLG ;ERROR COUNT
      ERRHRD ERRNO,T3ORWN,PKTSSR ;REWIND NOT ACCEPTED
      TRAP C$ERHRD
      .WORD 237
      .WORD T3ORWN
      .WORD PKTSSR
1518 30$: CKLOOP ;LOOP IF SELECTED
      TRAP C$CLP1

```

TEST 2: SKIP TAPE MARKS

```

1520 ;*****
1521 ;
1522 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1523 ;
1524 ;*****
1525
1526 035632 013701 036560      MOV      T30BFR+6,R1      ;PICK UP XSTO
1527 035636 010102      MOV      R1,R2           ;SET UP EXPECTED
1528 035640 052702 000002      BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
1529 035644 020102      CMP      R1,R2           ;DOES EXP = REC'D
1530 035646 001406      BEQ      40$             ;BR, IF EQUAL (OK)
1531 035650 005237 002214      INC      FATFLG          ;ERROR COUNT
1535 035654      ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    238
                                .WORD    T30BOT
                                .WORD    EXPREC
1536 035664      4C$: CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
1537 035666 012737 000001 036652  MOV      #1,T30WB        ;SET # OF TM TO SKIP
1538 ;
1539 ;*****
1540 ;
1541 ;SKIP TAPE MARK REVERSE,ACK,CVC=1 COMMAND
1542 ;
1543 ;*****
1544
1545 035674 012737 141410 036650  MOV      #141410,T30PK3  ;SKIP TAPE MARK REVERSE,ACK,CVC=1 CMD
1546 035702 012704 036650      MOV      #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
1547 035706 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
1548 035712 004737 016330      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
1549 035716 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
1550 035722 012702 100206      MOV      #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
1551 035726 020102      CMP      R1,R2           ;ARE THEY EQUAL
1552 035730 001406      BEQ      70$             ;BR, IF OK
1553 035732 005237 002214      INC      FATFLG          ;ERROR COUNT
1557 035736      ERRHRD  ERRNO,T30IBT,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    239
                                .WORD    T30IBT
                                .WORD    PKTSSR
1558 035746      70$: CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
1559 035746 104406
1560 ;
1561 ;*****
1562 ;
1563 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1564 ;
1565 ;*****
1566 035750 013701 036560      MOV      T30BFR+6,R1    ;PICK UP XSTO
1567 035754 010102      MOV      R1,R2           ;SET UP EXPECTED
1568 035756 052702 002000      BIS      #BIT10,R2      ;SET NEF BIT IN EXPECTED
1569 035762 020102      CMP      R1,R2           ;DOES EXP = REC'D
1570 035764 001406      BEQ      180$           ;BR, IF EQUAL (OK)
1571 035766 005237 002214      INC      FATFLG          ;ERROR COUNT
1575 035772      ERRHRD  ERRNO,T3ONEF,EXPREC ;TAPE NOT AT NEF

```

TEST 2: SKIP TAPE MARKS

	035772	104456					TRAP	C\$ERHRD
	035774	000360					.WORD	240
	035776	040626					.WORD	T3ONEF
	036000	015554					.WORD	EXPREC
1576	036002		180\$:	CKLOOP				;LOOP IF SELECTED
	036002	104406						TRAP
1577	036004			ENDSUB				C\$CLP1
	036004							; <<<<<<<<<<<<< END SUBTEST >>>>>>>>>>>>
	036004	104403						L10046:
1578	036006	023727	002214	000017	CMP	FATFLG,#15.		TRAP
1579	036014	103402			BLO	999\$		C\$ESUB
1580	036016	004737	017262		JSR	PC,CKDROP		;IS ERROR COUNT AT 25
1581	036022							;BR, IF LESS THAN 25
1582				999\$:				;TRY TO DROP THE UNIT
1583				;				
1584				;				
1585				;TEST 2, SUBTEST 4				
1586				;				
1587				;				
1588				;VERIFIES THAT A SKIP TAPE MARKS REVERSE COMMAND				
1589				;ISSUED WHILE THE TAPE IS POSITIONED JUST BEFORE THE				
1590				;FIRST RECORD ON ON TAPE (BUT NOT AT BOT) CAUSES TAPE				
1591				;STATUS ALERT TERMINATION, WITH THE REVERSE INTO BOT				
1592				;(RIB) STATUS BIT SET.				
1593				;				
1594				;				
1595				;				
1596				;				
1597				;				
1598	036022			;-				
	036022				BGNSUB			; >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
	036022	104402						T2.4:
1599	036024	004737	041302		JSR	PC,T3OREST		TRAP
1600	036030	005037	036704		CLR	T30FCN		C\$BSUB
1601	036034	004737	041374		JSR	PC,T3ORT2		;SET COMMAND PACKET
1602	036040	004737	041436		JSR	PC,T3ORT3		;CLEAR FILE COUNTER
1603	036044	012737	176750	036706	MOV	#65000.,T3ODLY		;SET UP OTHER COMMAND PACKET
1604	036052	004737	016054	10\$:	JSR	PC,SOFINIT		;SET UP OTHER COMMAND PACKET
1605	036056	103426			BCS	20\$;SET UP DELAY COUNTER
1606	036060				DELAY	250		;DO INITIALIZE ON CONTROLLER
	036060	012727	000250					;BR IF INIT WAS OK
	036064	000000						;DELAY ROUTINE CALL
	036066	013727	002116					MOV #250.(PC)+
	036072	000000						.WORD 0
	036074	005367	177772					MOV L\$DLY.(PC)+
	036100	001375						.WORD 0
	036102	005367	177756					DEC -6(PC)
	036106	001367						BNE .-4
1607	036110	005337	036706		DEC	T3ODLY		DEC -22(PC)
1608	036114	001356			BNE	10\$		BNE .-20
1609	036116	005237	002214		INC	FATFLG		;BUMP COUNTER
1613	036122	010001			MOV	R0,R1		;BR, IF MORE COUNTING TO DO
1614	036124				ERRDF	ERRNO,SFIERR,SFIMSG		;ERROR COUNT
	036124	104455						;CONTENTS OF TSSR REGISTER
	036126	000361						;FATAL ERROR TSSR WAS NOT OK
	036130	003646						TRAP
	036132	012114						C\$ERDF

TEST 2: SKIP TAPE MARKS

```

1615 036134
1616 036134 013737 002174 036550 20$: MOV UNITN,T30DSW ;SET UP UNIT NUMBER
1617 036142 012704 036530 MOV #T30PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
1618
1619 ;*****
1620 ;
1621 ;ISSUE WRITE CHARACTERISTICS COMMAND
1622 ;
1623 ;*****
1624
1625 036146 004737 010742 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
1626 036152 103407 BCS 23$ ;BR, IF COMMAND ISSUED OK
1627 036154 005237 002214 INC FATFLG ;ERROR COUNT
1631 036160 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
1632 036162 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
036162 104456 TRAP C$ERHRD
036164 00C362 .WORD 242
036166 005052 .WORD WRTMSG
036170 012114 .WORD SFIMSG
1633 036172 23$: CKLOOP ;LOOP IF SELECTED
036172 104406 TRAP C$CLP1
1634
1635 ;*****
1636 ;
1637 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1638 ;
1639 ;*****
1640
1641 036174 004737 011074 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
1642 036200 103411 BCS 30$ ;BR, IF NO PROBLEM
1643 036202 010004 MOV RO,R4 ;GET PACKET ADDRESS
1644 036204 016501 000002 MOV TSSR(R5),R1 ;GET STATUS REGISTER
1645 036210 005237 002214 INC FATFLG ;ERROR COUNT
1649 036214 ERRHRD ERRNO,T3ORWN,PKTSSR ;REWIND NOT ACCEPTED
036214 104456 TRAP C$ERHRD
036216 000363 .WORD 243
036220 040270 .WORD T3ORWN
036222 012126 .WORD PKTSSR
1650 036224 30$: CKLOOP ;LOOP IF SELECTED
036224 104406 TRAP C$CLP1
1651
1652 ;*****
1653 ;
1654 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1655 ;
1656 ;*****
1657
1658 036226 013701 036560 MOV T30BFR+6,R1 ;PICK UP XSTO
1659 036232 010102 MOV R1,R2 ;SET UP EXPECTED
1660 036234 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
1661 036240 020102 CMP R1,R2 ;DOES EXP = REC'D
1662 036242 001406 BEQ 40$ ;BR, IF EQUAL (OK)
1663 036244 005237 002214 INC FATFLG ;ERROR COUNT
1667 036250 ERRHRD ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
036250 104456 TRAP C$ERHRD
036252 000364 .WORD 244
036254 040071 .WORD T30BOT

```


TEST 2: SKIP TAPE MARKS

```

036256 015554
1668 036260 104406 40$: CKLOOP ;LOOP IF SELECTED .WORD EXPREC
036260 104406 TRAP C$CLP1
1669 036262 013737 003116 036652 MOV FREE,T30WB ;SET UP GOOD WRITE BUFFER
1670 036270 012737 000400 036656 MOV #256.,T30SZ ;SET UP SIZE
1671
1672 ;*****
1673 ;
1674 ;WRITE DATA,ACK,CVC=1 COMMAND
1675 ;
1676 ;*****
1677
1678 036276 012737 140005 036650 MOV #140005,T30PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
1679 036304 012704 036650 MOV #T30PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
1680 036310 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
1681 036314 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
1682 036320 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
1683 036324 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
1684 036330 020102 CMP R1,R2 ;ARE THEY EQUAL
1685 036332 001406 BEQ 70$ ;BR, IF OK
1686 036334 005237 002214 INC FATFLG ;ERROR COUNT
1690 036340 ERRHRD ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
036340 104456 TRAP C$ERHRD
036342 000365 .WORD 245
036344 037220 .WORD T30WDD
036346 012126 .WORD PKTSSR
1691 036350 70$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
036350 104406
1692
1693 ;*****
1694 ;
1695 ;SKIP TAPE MARK REVERSE,ACK,CVC=1 COMMAND
1696 ;
1697 ;*****
1698
1699 036352 012737 000001 036652 MOV #1,T30WB ;# OF TM TO SKIP
1700 036360 012737 141410 036650 MOV #141410,T30PK3 ;SKIP TAPE MARK REVERSE,ACK,CVC=1 CMD
1701 036366 012704 036650 MOV #T30PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
1702 036372 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
1703 036376 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
1704 036402 016501 000002 MOV TSSR(R5),R1 ;PICK UP TSSR
1705 036406 012702 100204 MOV #SSR!BIT2!SC,R2 ;SET UP EXPECTED (SSR AND SC ONLY)
1706 036412 020102 CMP R1,R2 ;WAS STATUS GOOD
1707 036414 001406 BEQ 160$ ;BR, IF TERMINATION WAS GOOD
1708 036416 005237 002214 INC FATFLG ;ERROR COUNT
1712 036422 ERRHRD ERRNO,T30IBU,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
036422 104456 TRAP C$ERHRD
036424 000366 .WORD 246
036426 036710 .WORD T30IBU
036430 012126 .WORD PKTSSR
1713 036432 160$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
036432 104406
1714
1715 ;*****
1716 ;
1717 ;GET EXTENDED STATUS REGISTER ZERO (XST3) FROM MESSAGE BUFFER
1718 ;

```


TEST 2: SKIP TAPE MARKS

1777	036650				T30PK3:			
1778	036650	100205				.WORD	100205	;REREAD COMMAND, IE AND ACK
1779	036652				T30RB:			
1780	036652	003116			T30WB:	.WORD	FREE	;ADDRESS OF WRITE BUFFER
1781	036654	000000				.WORD	0	
1782	036656	000000			T30SZ:	.WORD	0	;SIZE OF BUFFER (EXTENT)
1783						.EVEN		
1784					:			
1785					:			
1786					:			
1787	036660				T30BF2:			
1788	036660	010			T30BS0:	.BYTE	10	;BSELO AREA
1789	036661	200			T30BS1:	.BYTE	200	;BSEL1 AREA
1790	036662	000000			T30S2:	.WORD	0	;SEL 2 AREA
1791	036664	000000			T30S3:	.WORD	0	;DATA AREA
1792					:			
1793					:			
1794						.EVEN		
1795					;TAPE MOTION PACKET COMMAND VALUES			
1796								
1797	036666				T30IMV:			
1798	036666				T30RN:			
1799	036666	000000				.WORD	000000	;NEITHER EWB NOR ESS
1800	036670	000100				.WORD	000100	;EWB SET
1801	036672	000200				.WORD	000200	;ESS SET
1802	036674	000300				.WORD	000300	;BOTH EWB AND ESS SET
1803	036676	177777				.WORD	177777	;END OF DATA
1804								
1805					:			
1806	036700	000000			T30CNT:	.WORD	0	;TAPE TIMER COUNTER STORAGE AREA
1807	036702	000000			T30CNU:	.WORD	0	;TAPE TIMER COUNTER STORAGE AREA
1808	036704	000000			T30FCN:	.WORD	0	;FILE NUMBER COUNTER
1809	036706	000000			T30DLY:	.WORD	0	;DELAY COUNTER STORAGE
1810					:			
1811					;LOCAL TEXT MESSAGES FOR TEST			
1812					:			
1813					:			
1814	036710	124	123	123	T30IBU:	.ASCIZ	'TSSR Incorrect After SKIP TAPE MARK REVERSE Into BOT'	
1815	036775	122	111	102	T30RIB:	.ASCIZ	'RIB Bit (XST3) Failed To Set After Reverse Into BOT'	
1816	037061	124	123	123	T30IBT:	.ASCIZ	'TSSR Incorrect After SKIP TAPE MARK REVERSE At BOT'	
1817	037144	124	123	123	T30SKM:	.ASCIZ	'TSSR Incorrect After SKIP TAPE MARK Command'	
1818	037220	124	123	123	T30WDD:	.ASCIZ	'TSSR Not Correct After WRITE DATA Command'	
1819	037272	124	141	160	T30PTB:	.ASCIZ	'Tape Not Positioned On Correct Record After READ REVERSE'	
1820	037363	124	141	160	T30TPB:	.ASCIZ	'Tape Not Positioned On Second File First Record'	
1821	037443	124	123	123	T30RDF:	.ASCIZ	'TSSR Incorrect After READ FORWARD Into "File"'	
1822	037521	124	123	123	T30RDG:	.ASCIZ	'TSSR Incorrect After SPACE Command Into TAPE MARK'	
1823	037603	124	123	123	T30WDF:	.ASCIZ	'TSSR Not Correct After Illegal Mode Bits Set'	
1824	037660	111	154	154	T30LOQ:	.ASCIZ	'Illegal Mode Bits, Failed To Set ILC Bit In XST0'	
1825	037741	127	122	111	T30SSR:	.ASCIZ	'WRITE MISCELLANEOUS Command Not Accepted'	
1826	040012	124	123	123	T30WDE:	.ASCIZ	'TSSR Not Correct After SKIP TAPE MARKS, At BOT'	
1827	040071	124	141	160	T30BOT:	.ASCIZ	'Tape Not At BOT After REWIND Command'	
1828	040136	124	123	123	T30TM:	.ASCIZ	'TSSR Not Correct After SPACE FORWARD Command'	
1829	040213	124	123	123	T30TM2:	.ASCIZ	'TSSR Not Correct After SPACE REVERSE Command'	
1830	040270	122	145	167	T30RWN:	.ASCIZ	'Rewind (POSITION) Command Not Accepted'	
1831	040337	104	162	151	T30OFL:	.ASCIZ	'Drive 7 Select Failed To Set "OFL" In TSSR'	
1832	040412	124	123	123	T30WDC:	.ASCIZ	'TSSR Not Correct After WRITE TAPE MARK Command'	
1833	040471	103	126	103	T30VCK:	.ASCIZ	'CVC Set, Didn't Reset VCK In Message Buffer'	

TEST 2: SKIP TAPE MARKS

```

1834 040544      124      115      113 T30TMK: .ASCIZ 'TMK Not Set After WRITE TAPE MARK (RETRY) Command'
1835 040626      123      113      111 T30NEF: .ASCIZ 'SKIP TAPE MARKS, At BOT, Failed To Set NEF Bit'
1836 040705      124      115      113 T30RRM: .ASCIZ 'TMK Not Set After READ REVERSE Into TAPE MARK'
1837 040763      124      115      113 T30RRN: .ASCIZ 'TMK Not Set After SPACE REVERSE Into TAPE MARK'
1838 041042      124      115      113 T30RRP: .ASCIZ 'TMK Not Set After READ FORWARD Into TAPE MARK'
1839 041120      116      117      040 T30DTR: .ASCIZ 'NO Data Transferred On READ FORWARD'
1840 041164      104      141      164 T30DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
1841 041261      123      153      151 TST30ID: .ASCIZ 'Skip Tape Marks'
1842
1843
1844
1845
1846
1847
1848
1849
1850 041302
1851 041302
1852 041306      012701 036530
1853 041312      012721 100004
1854 041316      012721 036540
1855 041322      005021
1856 041324      012721 000012
1857 041330      012721 036552
1858 041334      005021
1859 041336      012721 000024
1860 041342      005021
1861 041344      012711 000000
1862 041350      012702 000030
1863 041354      012762 177777 036552 64$:
1864 041362      005742
1865 041364      022702 000000
1866 041370      001371
1867 041372      000207
1868
1869 041374
1870 041374
1871 041400      012701 036640
1872 041404      012721 100006
1873 041410      012721 036660
1874 041414      005021
1875 041416      012721 000006
1876 041422      005021
1877 041424      012701 036660
1878 041430      005021
1879 041432      005011
1880 041434      000207
1881 041436
1882 041436
1883 041442      012701 036650
1884 041446      005021
1885 041450      005021
1886 041452      005021
1887 041454      005011
1888 041456      000207
1889 041460
      041460

```

```

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

T30REST:
  SAVREG
  MOV #T30PACKET,R1 ;SAVE THE REGISTERS
  MOV #100004,(R1)+ ;START OF THE PACKET
  MOV #T30DATA,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
  CLR (R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
  MOV #10,(R1)+ ;EXTENDED ADDRESS
  MOV #T30BFR,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
  CLR (R1)+ ;ADDRESS OF MESSAGE BUFFER
  MOV #20,(R1)+ ;LENGTH OF MESSAGE BUFFER
  CLR (R1)+
  MOV #0,(R1) ;SELECT DRIVE ZERO
  MOV #24,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
  MOV #177777,T30BFR(R2) ;ALL ONES TO MESSAGE BUFFER
  TST -(R2) ;NEXT LOCATION
  CMP #0,R2 ;CHECK R2 FOR DONE
  BNE 64$ ;KEEP GOING UNTIL DONE
  RTS PC ;RETURN

T30RT2:
  SAVREG
  MOV #T30PK2,R1 ;SAVE THE REGISTERS
  MOV #100006,(R1)+ ;START OF THE PACKET
  MOV #T30BF2,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
  CLR (R1)+ ;ADDRESS OF DATA BLOCK
  MOV #6,(R1)+ ;EXTENDED ADDRESS
  CLR (R1)+ ;SIZE OF DATA BLOCK IN BYTES
  MOV #T30BF2,R1 ;POINT TO DATA SEL AREA
  CLR (R1)+
  CLR (R1)
  RTS PC ;RETURN

T30RT3:
  SAVREG
  MOV #T30PK3,R1 ;SAVE REGISTERS
  CLR (R1)+ ;SET UP POINTER ADDRESS
  CLR (R1)+ ;COMMAND SPACE
  CLR (R1)+ ;ADDRESS OF DATA BLOCK
  CLR (R1)+ ;EXTENDED ADDRESS
  RTS PC ;SIZE OF DATA TRANSFER BLOCK
  ENDTST ;RETURN

```

L10043:

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

	041552	013727	002116						MOV	LSDLY,(PC)	
	041556	000000							.WORD	0	
	041560	005367	177772						DEC	-6(PC)	
	041564	001375							BNE	..4	
	041566	005367	177756						DEC	-22(PC)	
	041572	001367							BNE	..-20	
1945	041574	005337	043372		DEC	T31DLY					
1946	041600	001356			BNE	10\$					
1947	041602	005237	002214		INC	FATFLG					
1951	041606	010001			MOV	RO,R1					
1952	041610				ERRDF	ERRNO,SFIERR,SFIMSG					
	041610	104455									
	041612	000455							TRAP	C\$ERDF	
	041614	003646							.WORD	301	
	041616	012114							.WORD	SFIERR	
									.WORD	SFIMSG	
1953	041620	013737	002174	043240	20\$:	MOV	UNITN,T31DSW				
1954	041626	012704	043220		MOV	@T31PACKET,R4					
1955	041632	004737	010742		JSR	PC,WRTCHR					
1956	041636	103407			BCS	23\$					
1957	041640	005237	002214		INC	FATFLG					
1961	041644	010001			MOV	RO,R1					
1962	041646				ERRHRD	ERRNO,WRTMSG,SFIMSG					
	041646	104456									
	041650	000456							TRAP	C\$ERHRD	
	041652	005052							.WORD	302	
	041654	012114							.WORD	WRTMSG	
									.WORD	SFIMSG	
1963	041656			23\$:	CKLOOP						
	041656	104406									
1964	041660	004737	011074		JSR	PC,REWIND			TRAP	C\$CLP1	
1965	041664	103407			BCS	30\$					
1966	041666	010004			MOV	RO,R4					
1967	041670	005237	002214		INC	FATFLG					
1971	041674				ERRHRD	ERRNO,T31RWN,PKTSSR					
	041674	104456									
	041676	000457							TRAP	C\$ERHRD	
	041700	044724							.WORD	303	
	041702	012126							.WORD	T31RWN	
1972	041704			30\$:	CKLOOP				.WORD	PKTSSR	
	041704	104406									
1973	041706	013701	043250		MOV	T31BFR+6,R1			TRAP	C\$CLP1	
1974	041712	010102			MOV	R1,R2					
1975	041714	052702	000002		BIS	@BIT1,R2					
1976	041720	020102			CMP	R1,R2					
1977	041722	001406			BEQ	40\$					
1978	041724	005237	002214		INC	FATFLG					
1982	041730				ERRHRD	ERRNO,T31BOT,EXPREC					
	041730	104456									
	041732	000460							TRAP	C\$ERHRD	
	041734	044375							.WORD	304	
	041736	015554							.WORD	T31BOT	
1983	041740			40\$:	CKLOOP				.WORD	EXPREC	
	041740	104406									
1984	041742	013737	003116	043342	MOV	FREE,T31WB			TRAP	C\$CLP1	
1985	041750	012737	140005	043340	65\$:	MOV	@140005,T31PK3				
1986	041756	012704	043340		MOV	@T31PK3,R4					
1987	041762	012700	000144		MOV	@100.,RO					
1988	041766	004737	017502		JSR	PC,FILLMEM					

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

1989	041772	012737	000144	043346	MOV	#100.,T31SZ	;SET UP RECORD SIZE IN PACKET		
1990	042000	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND		
1991	042004	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
1992	042010	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
1993	042014	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED		
1994	042020	020102			CMP	R1,R2	;ARE THEY EQUAL		
1995	042022	001406			BEQ	80\$;BR, IF OK		
1996	042024	005237	002214		INC	FATFLG	;ERROR COUNT		
2000	042030				ERRHRD	ERRNO,T31WDC,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA		
	042030	104456					TRAP	C\$ERHRD	
	042032	000461					.WORD	305	
	042034	045260					.WORD	T31WDC	
	042036	012126					.WORD	PKTSSR	
2001	042040			80\$:	CKLOOP		;LOOP IF SELECTED		
	042040	104406					TRAP	C\$CLP1	
2002	042042	004737	011074		JSR	PC,REWIND	;CALL TAPE REWIND COMMAND		
2003	042046	103407			BCS	230\$;BR, IF NO PROBLEM		
2004	042050	010001			MOV	R0,R1	;SAVE TSSR		
2005	042052	005237	002214		INC	FATFLG	;ERROR COUNT		
2009	042056				ERRHRD	ERRNO,T31RWN,EXPREC	;REWIND NOT ACCEPTED		
	042056	104456					TRAP	C\$ERHRD	
	042060	000462					.WORD	306	
	042062	044724					.WORD	T31RWN	
	042064	015554					.WORD	EXPREC	
2010	042066			230\$:	CKLOOP		;LOOP IF SELECTED		
	042066	104406					TRAP	C\$CLP1	
2011	042070	013701	043250		MOV	T31BFR+6,R1	;PICK UP XSTO		
2012	042074	010102			MOV	R1,R2	;SET UP EXPECTED		
2013	042076	052702	000002		BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED		
2014	042102	020102			CMP	R1,R2	;DOES EXP = REC'D		
2015	042104	001406			BEQ	240\$;BR, IF EQUAL (OK)		
2016	042106	005237	002214		INC	FATFLG	;ERROR COUNT		
2020	042112				ERRHRD	ERRNO,T31BOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	042112	104456					TRAP	C\$ERHRD	
	042114	000463					.WORD	307	
	042116	044375					.WORD	T31BOT	
	042120	015554					.WORD	EXPREC	
2021	042122			240\$:	CKLOOP		;LOOP IF SELECTED		
	042122	104406					TRAP	C\$CLP1	
2022	042124	012737	041012	043340	265\$:	MOV	#041012,T31PK3	;NO-OP,CVC=1 COMMAND	
2023	042132	012704	043340		MOV	#T31PK3,R4	;SET UP R4 WITH PACKET ADDRESS		
2024	042136	010337	043346		MOV	R3,T31SZ	;SET UP RECORD SIZE IN PACKET		
2025	042142	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND		
2026	042146	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
2027	042152	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
2028	042156	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED		
2029	042162	020102			CMP	R1,R2	;ARE THEY EQUAL		
2030	042164	001406			BEQ	280\$;BR, IF OK		
2031	042166	005237	002214		INC	FATFLG	;ERROR COUNT		
2035	042172				ERRHRD	ERRNO,T31RDF,PKTSSR	;TSSR INCORRECT AFTER READ DATA		
	042172	104456					TRAP	C\$ERHRD	
	042174	000464					.WORD	308	
	042176	043573					.WORD	T31RDF	
	042200	012126					.WORD	PKTSSR	
2036	042202			280\$:	CKLOOP		;LOOP IF SELECTED		
	042202	104406					TRAP	C\$CLP1	
2037	042204	013701	043250		MOV	T31BFR+6,R1	;PICK UP XSTO		

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

```

2038 042210 010102          MOV      R1,R2          ;SET UP EXPECTED
2039 042212 052702 000002  BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
2040 042216 020102          CMP      R1,R2          ;DOES EXP = REC'D
2041 042220 001406          BEQ      285$          ;BR, IF EQUAL (OK)
2042 042222 005237 002214  INC      FATFLG         ;ERROR COUNT
2046 042226          ERRHRD  ERRNO,T31BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                  TRAP      C$ERHRD
                                  .WORD     309
                                  .WORD     T31BOT
                                  .WORD     EXPREC
                                  TRAP      C$CLP1
042226 104456
042230 000465
042232 044375
042234 015554
2047 042236          285$:  CKLOOP          ;LOOP IF SELECTED
                                  TRAP      C$CLP1
042236 104406
2048 042240 012737 140001 043340  MOV      #140001,T31PK3 ;READ,ACK,CVC=1 COMMAND
2049 042246 012704 043340  MOV      #T31PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
2050 042252 012737 000144 043346  MOV      #100.,T31SZ    ;SET UP RECORD SIZE IN PACKET
2051 042260 010465 000000  MOV      R4,TSDB(R5)    ;ISSUE COMMAND
2052 042264 004737 016330  JSR      PC,WAITF       ;WAIT FOR SSR TO SET
2053 042270 016501 000002  MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
2054 042274 012702 000200  MOV      #SSR,R2       ;SET UP EXPECTED
2055 042300 020102          CMP      R1,R2          ;ARE THEY EQUAL
2056 042302 001406          BEQ      290$          ;BR, IF OK
2057 042304 005237 002214  INC      FATFLG         ;ERROR COUNT
2061 042310          ERRHRD  ERRNO,T31RDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                  TRAP      C$ERHRD
                                  .WORD     310
                                  .WORD     T31RDE
                                  .WORD     PKTSSR
042310 104456
042312 000466
042314 043374
042316 012126
2062 042320          290$:  CKLOOP          ;LOOP IF SELECTED
                                  TRAP      C$CLP1
042320 104406
2063 042322 017701 140570  MOV      #FREE,R1       ;GET DATA READ
2064 042326 012702 000144  MOV      #100.,R2       ;READ EXPECTED
2065 042332 020102          CMP      R1,R2          ;DID TAPE STAY POSITIONED
2066 042334 001406          BEQ      330$          ;BR, IF EXPD = RECD
2067 042336 005237 002214  INC      FATFLG         ;ERROR COUNT
2071 042342          ERRHRD  ERRNO,T31WNG,EXPREC ;TAPE DATA NOT CORRECT
                                  TRAP      C$ERHRD
                                  .WORD     311
                                  .WORD     T31WNG
                                  .WORD     EXPREC
042342 104456
042344 000467
042346 043521
042350 015554
2072 042352          330$:  ENDSUB          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
2073 042352          ENDSUB          ;>>>>>>>>>>>> L10051:
                                  TRAP      C$ESUB
042352 104403
2074 042354 023727 002214 000017  CMP      FATFLG,#15.    ;IS ERROR COUNT AT 25
2075 042362 103402          BLO      999$          ;BR, IF LESS THAN 25
2076 042364 004737 017262  JSR      PC,CKDROP     ;TRY TO DROP THE UNIT
2077 042370          999$:
2078 ;
2079 ;
2080 ;TEST 3, SUBTEST 2
2081 ;
2082 ;
2083 ;
2084 ;
2085 ;
2086 ;
2087 ;
VERIFIES THAT THE INITIALIZE COMMAND OPERATES AS A NO-OP,
ASSUMING NO MICRODIAGNOSTIC ERRORS ARE PRESENT (THEY WOULD
ALREADY HAVE BEEN DETECTED IN PREVIOUS TESTS). THE TEST
SEQUENCE IS SIMILAR TO THAT USED IN SUBTEST 1.

```


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

2136	042570	012704	043340		MOV	#T31PK3,R4	;SET UP R4 WITH PACKET ADDRESS		
2137	042574	012700	000144		MOV	#100.,R0	;SET PATTERN IN CORRECT REGISTER		
2138	042600	004737	017502		JSR	PC,FILLMEM	;FILL MEMORY WITH RECORD SIZE		
2139	042604	012737	000144	043346	MOV	#100.,T31SZ	;SET UP RECORD SIZE IN PACKET		
2140	042612	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND		
2141	042616	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
2142	042622	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
2143	042626	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED		
2144	042632	020102			CMP	R1,R2	;ARE THEY EQUAL		
2145	042634	001406			BEQ	80\$;BR, IF OK		
2146	042636	005237	002214		INC	FATFLG	;ERROR COUNT		
2150	042642				ERRHRD	ERRNO,T31WDC,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA		
	042642	104456					TRAP	C\$ERHRD	
	042644	000474					.WORD	316	
	042646	045260					.WORD	T31WDC	
	042650	012126					.WORD	PKTSSR	
2151	042652			80\$:	CKLOOP		;LOOP IF SELECTED		
	042652	104406					TRAP	C\$CLP1	
2152	042654	004737	011074		JSR	PC,REWIND	;CALL TAPE REWIND COMMAND		
2153	042660	103407			BCS	230\$;BR, IF NO PROBLEM		
2154	042662	010001			MOV	RO,R1	;SAVE TSSR		
2155	042664	005237	002214		INC	FATFLG	;ERROR COUNT		
2159	042670				ERRHRD	ERRNO,T31RWN,EXPREC	;REWIND NOT ACCEPTED		
	042670	104456					TRAP	C\$ERHRD	
	042672	000475					.WORD	317	
	042674	044724					.WORD	T31RWN	
	042676	015554					.WORD	EXPREC	
2160	042700			230\$:	CKLOOP		;LOOP IF SELECTED		
	042700	104406					TRAP	C\$CLP1	
2161	042702	013701	043250		MOV	T31BFR+6,R1	;PICK UP XSTO		
2162	042706	010102			MOV	R1,R2	;SET UP EXPECTED		
2163	042710	052702	000002		BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED		
2164	042714	020102			CMP	R1,R2	;DOES EXP = REC'D		
2165	042716	001406			BEQ	240\$;BR, IF EQUAL (OK)		
2166	042720	005237	002214		INC	FATFLG	;ERROR COUNT		
2170	042724				ERRHRD	ERRNO,T31BOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	042724	104456					TRAP	C\$ERHRD	
	042726	000476					.WORD	318	
	042730	044375					.WORD	T31BOT	
	042732	015554					.WORD	EXPREC	
2171	042734			240\$:	CKLOOP		;LOOP IF SELECTED		
	042734	104406					TRAP	C\$CLP1	
2172	042736	012737	041012	043340	265\$:	MOV	#041012,T31PK3	;INITIALIZE,CVC=1 COMMAND	
2173	042744	012704	043340		MOV	#T31PK3,R4	;SET UP R4 WITH PACKET ADDRESS		
2174	042750	010337	043346		MOV	R3,T31SZ	;SET UP RECORD SIZE IN PACKET		
2175	042754	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND		
2176	042760	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
2177	042764	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
2178	042770	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED		
2179	042774	020102			CMP	R1,R2	;ARE THEY EQUAL		
2180	042776	001406			BEQ	280\$;BR, IF OK		
2181	043000	005237	002214		INC	FATFLG	;ERROR COUNT		
2185	043004				ERRHRD	ERRNO,T31RDF,PKTSSR	;TSSR INCORRECT AFTER READ DATA		
	043004	104456					TRAP	C\$ERHRD	
	043006	000477					.WORD	319	
	043010	043573					.WORD	T31RDF	
	043012	012126					.WORD	PKTSSR	

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

```

043214 104432
043216 003600
2235
2236
2237
2241 043220
2242 043220 100004
2243 043222 043230
2244 043224 000000
2245 043226 000012
2246 043230
2247 043230 043242
2248 043232 000000
2249 043234 000024
2250 043236 000000
2251 043240 000000
2252 043242
2253
2254
2255
2257 043330
2259 043330
2260 043330 100006
2261 043332 043350
2262 043334 000000
2263 043336 000006
2264
2268 043340
2269 043340 100005
2270 043342
2271 043342 003116
2272 043344 000000
2273 043346 000000
2274
2275
2276
2277
2278 043350
2279 043350 010
2280 043351 200
2281 043352 000000
2282 043354 000000
2283
2284
2285
2286
2287
2288 043356 100205
2289 043360 100605
2290 043362 102205
2291 043364 177777
2292
2293
2294 043366 000000
2295 043370 000000
2296 043372 000000
2297

;+
;LOCAL STORAGE FOR THIS TEST
;
T31PACKET:
;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH . ACK
;ADDRESS OF CHARACTERISTICS BLOCK
        .WORD 100004
        .WORD T31DATA
        .WORD 0
        .WORD 10.
T31DATA:
;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
        .WORD T31BFR
        .WORD 0
        .WORD 20.
        .WORD 0
T31DSW: .WORD 0
T31BFR: .BLKW 25.
;SELECT DRIVE 0
;MESSAGE BUFFER
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
        .=<.+10>E177770
T31PK2:
;WRITE SUB SYS MEM COMMAND, AND ACK
;ADDRESS OF SELECT BLOCK DATA
        .WORD 100006
        .WORD T31BF2
        .WORD 0
        .WORD 6.
;SIZE OF DATA PACKET
T31PK3:
;REREAD COMMAND, AND ACK
T31RB:
;ADDRESS OF WRITE BUFFER
T31WB: .WORD FREE
        .WORD 0
T31SZ: .WORD 0
        .EVEN
;SIZE OF BUFFER (EXTENT)
;
;
T31BF2:
T31BS0: .BYTE 10
T31BS1: .BYTE 200
T31S2: .WORD 0
T31S3: .WORD 0
;BSEL0 AREA
;BSEL1 AREA
;SEL 2 AREA
;DATA AREA
;
;
        .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T31RN: .WORD 100205
T31WDR: .WORD 100605
T31CON: .WORD 102205
        .WORD 177777
;REREAD DATA (NEXT)
;REREAD DATA RETRY
;WRITE CONTINOUS
;END OF DATA
;
;
T31CNT: .WORD 0
T31CNU: .WORD 0
T31DLY: .WORD 0
;TAPE TIMER COUNTER STORAGE AREA
;TAPE TIMER COUNTER STORAGE AREA
;DELAY COUNTER
;+

```

TRAP C\$EXIT
.WORD L10050-

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

```

2298          ;LOCAL TEXT MESSAGES FOR TEST
2299          ;
2300
2301 043374      124      123      123  T31RDE: .ASCIZ  'TSSR Not Correct After READ Command'
2302 043440      124      141      160  T31WNH: .ASCIZ  'Tape Position Incorrect After INITIALIZE Command'
2303 043521      124      141      160  T31WNG: .ASCIZ  'Tape Position Incorrect After NOP Command'
2304 043573      124      123      123  T31RDF: .ASCIZ  'TSSR Incorrect After READ DATA Command'
2305 043642      122      105      122  T31RRF: .ASCIZ  'REREAD Previous (Space Reverse, Read Forward) Command Failed'
2306 043737      120      117      123  T31SC: .ASCIZ   'POSITION (Space Command) Failed, TSSR Not Correct'
2307 044021      122      111      102  T31LOR: .ASCIZ  'RIB NOT SET AFTER READ REVERSE INTO BOT'
2308 044071      124      123      123  T31WDF: .ASCIZ  'TSSR Not Correct After Illegal Mode Bits Set'
2309 044146      111      154      154  T31LOQ: .ASCIZ  'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
2310 044227      122      105      122  T31SSR: .ASCIZ  'REREAD COMMAND Not Accepted'
2311 044263      124      123      123  T31WDE: .ASCIZ  'TSSR Not Correct After NO-OP ("CLEAN TAPE") AND INITIALIZE Command,At BOT'
2312 044375      124      141      160  T31BOT: .ASCIZ  'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
2313 044470      116      117      055  T31TIM: .ASCIZ  'NO-OP ("CLEAN TAPE") AND INITIALIZE'S Erase Tape Not Long Enough'
2314 044570      122      105      122  T31EOT: .ASCIZ  'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
2315 044647      124      123      123  T31TM: .ASCIZ   'TSSR Not Correct After REREAD COMMAND Reject'
2316 044724      122      145      167  T31RMN: .ASCIZ  'Rewind (POSITION) Command Not Accepted'
2317 044773      122      101      115  T31RNC: .ASCIZ  'RAM Error, Correct Data Pattern Not In Ram'
2318 045046      124      123      123  T31AM3: .ASCIZ  'TSSR Init. Failed After REREAD COMMAND'
2319 045115      104      162      151  T31OFL: .ASCIZ  'Drive 7 Select Failed To Set "OFL" In TSSR'
2320 045170      124      123      123  T31WDD: .ASCIZ  'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
2321 045260      124      123      123  T31WDC: .ASCIZ  'TSSR Not Correct After REREAD DATA Command'
2322 045333      103      126      103  T31VCK: .ASCIZ  'CVC Set, Didn't Reset VCK In Message Buffer'
2323 045406      124      123      102  T31BA: .ASCIZ   'TSBA Not Correct After REREAD DATA Command'
2324 045461      127      122      111  T31WSS: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
2325 045550      122      145      141  T31LON: .ASCIZ  'Reading Long Record Failed To Set RLL Bit In XST0'
2326 045632      122      145      141  T31LOP: .ASCIZ  'Reading Long Record Failed To Set RLS Bit In XST0'
2327 045714      122      145      163  T31PBP: .ASCIZ  'Residual Byte Count Incorrect After Short Record Read'
2328 046002      122      145      141  T31TRL: .ASCIZ  'Reading Long Record Failed To Give Tape Status Alert'
2329 046070      116      117      055  T31NEF: .ASCIZ  'NO-OP ("CLEAN TAPE") AND INITIALIZE, At First Record, Failed To Set RIB Bit
X
2330 046211      124      123      123  T31SCF: .ASCIZ  'TSSR Not Correct After SPACE RECORDS Command'
2331 046266      124      123      123  T31TSA: .ASCIZ  'TSSR Not Correct After NO-OP ("CLEAN TAPE") AND INITIALIZE, Into BOT'
2332 046373      124      123      123  T31WRF: .ASCIZ  'TSSR Not Correct After NO-OP ("CLEAN TAPE") AND INITIALIZE Command'
2333 046476      104      141      164  T31DTA: .ASCIZ  'Data Compare Error, Data Read From Tape Not Equal To Written'
2334 046573      116      117      055  TST31ID: .ASCIZ  'NO-OP ("Clean Tape") And INITIALIZE'
2335
2336          ;EVEN
2337
2338          ;
2339          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
2340          ;WRITE SUBSYSTEM MEMORY COMMAND
2341
2342          ;
2343
2344          T31REST:
2345 046640      012701  043220  SAVREG
2346 046644      012701  100004  MOV      #T31PACKET,R1      ;SAVE THE REGISTERS
2347 046654      012721  043230  MOV      #100004,(R1)+     ;START OF THE PACKET
2348 046660      005021  CLR      (R1)+             ;WRITE SUBSYSTEM MEM. WITH ACK.
2349 046662      012721  000012  MOV      #T31DATA,(R1)+   ;ADDRESS OF CHARAISTICS DATA BLOCK
2350 046666      012721  043242  CLR      (R1)+             ;EXTENDED ADDRESS
2351 046672      005021  MOV      #10.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
2352 046674      012721  000024  MOV      #T31BFR,(R1)+   ;ADDRESS OF MESSAGE BUFFER
2353 046700      005021  CLR      (R1)+             ;LENGTH OF MESSAGE BUFFER
2354 046702      012711  000000  MOV      #0,(R1)         ;SELECT DRIVE ZERO

```

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

```

2355 046706 012702 000030      MOV    #24.,R2          ;NUMBER OF LOCATIONS TO BE CLEARED
2356 046712 012762 177777 043242 64$:  MOV    #177777,T31BFR(R2) ;ALL ONES TO MESSAGE BUFFER
2357 046720 005742              TST    -(R2)           ;NEXT LOCATION
2358 046722 022702 000000      CMP    #0,R2          ;AT END OF LOOP YET
2359 046726 001371              BNE    64$            ;KEEP GOING UNTIL DONE
2360 046730 000207              RTS    PC              ;RETURN

```

```

2361
2362 046732          T31RT2:
2363 046732          SAVREG          ;SAVE THE REGISTERS
2364 046736 012701 043330      MOV    #T31PK2,R1      ;START OF THE PACKET
2365 046742 012721 100006      MOV    #100006,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK.
2366 046746 012721 043350      MOV    #T31BF2,(R1)+  ;ADDRESS OF DATA BLOCK
2367 046752 005021              CLR    (R1)+           ;EXTENDED ADDRESS
2368 046754 012721 000006      MOV    #6.,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
2369 046760 005021              CLR    (R1)+
2370 046762 012701 043350      MOV    #T31BF2,R1     ;POINT TO DATA SEL AREA
2371 046766 005021              CLR    (R1)+
2372 046770 005011              CLR    (R1)
2373 046772 000207              RTS    PC              ;RETURN

```

```

2374 046774          T31RT3:
2375 046774          SAVREG          ;SAVE REGISTERS
2376 047000 012701 043340      MOV    #T31PK3,R1     ;SET UP POINTER ADDRESS
2377 047004 005021              CLR    (R1)+           ;COMMAND SPACE
2378 047006 005021              CLR    (R1)+           ;ADDRESS OF DATA BLOCK
2379 047010 005021              CLR    (R1)+           ;EXTENDED ADDRESS
2380 047012 005011              CLR    (R1)           ;SIZE OF DATA TRANSFER BLOCK
2381 047014 000207              RTS    PC              ;RETURN
2382 047016          ENDTST

```

L10050: TRAP C\$ETST

```

2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409

```

.SBTTL TEST 4: Erase And Operation Incomplete

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

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

TEST 4: ERASE AND OPERATION INCOMPLETE

	047152	003646						.WORD	SFIERR
	047154	012114						.WORD	SFIMSG
2461	047156	013737	002174	051400	20\$:	MOV	UNITN,T32DSW		;SET UP DRIVE NUMBER
2462	047164	012704	051360			MOV	#T32PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS
2463	047170	004737	010742			JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS
2464	047174	103407				BCS	25\$;BR, IF COMMAND ISSUED OK
2465	047176	005237	002214			INC	FATFLG		;ERROR COUNT
2469	047202	010001				MOV	RO,R1		;SAVE CONTENTS OF TSSR
2470	047204					ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTISC FAILED
	047204	104456						TRAP	C\$ERHRD
	047206	000622						.WORD	402
	047210	005052						.WORD	WRTMSG
	047212	012114						.WORD	SFIMSG
2471	047214				25\$:	CKLOOP			;LOOP IF SELECTED
	047214	104406						TRAP	C\$CLP1
2472	047216	004737	011074			JSR	PC,REWIND		;CALL TAPE REWIND COMMAND
2473	047222	103411				BCS	26\$;BR, IF NO PROBLEM
2474	047224	010004				MOV	RO,R4		;SET UP REWIND PACKET ADDRESS
2475	047226	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS
2476	047232	005237	002214			INC	FATFLG		;ERROR COUNT
2480	047236					ERRHRD	ERRNO,T32RWN,PKTSSR		;REWIND NOT ACCEPTED
	047236	104456						TRAP	C\$ERHRD
	047240	000623						.WORD	403
	047242	051730						.WORD	T32RWN
	047244	012126						.WORD	PKTSSR
2481	047246				26\$:	CKLOOP			;LOOP IF SELECTED
	047246	104406						TRAP	C\$CLP1
2482	047250	012703	000400			MOV	#256.,R3		;STARTING RECORD SIZE
2483	047254	013737	003116	051502		MOV	FREE,T32WB		;STARTING WRITE BUFFER ADDRESS
2484	047262	012737	140005	051500		MOV	#140005,T32PK3		;WRITE DATA,CVC-1,ACK COMMAND
2485	047270	012704	051500			MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS
2486	047274	010337	051506		27\$:	MOV	R3,T32SZ		;SET UP RECORD SIZE IN PACKET
2487	047300	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND
2488	047304	004737	016330			JSR	PC,WAITF		;WAIT FOR SSR TO SET
2489	047310	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS
2490	047314	012702	000200			MOV	#SSR,R2		;SET UP EXPECTED
2491	047320	020102				CMP	R1,R2		;ARE THEY EQUAL
2492	047322	001406				BEQ	28\$;BR, IF OK
2493	047324	005237	002214			INC	FATFLG		;ERROR COUNT
2497	047330					ERRHRD	ERRNO,T32WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA
	047330	104456						TRAP	C\$ERHRD
	047332	000624						.WORD	404
	047334	052566						.WORD	T32WDC
	047336	012126						.WORD	PKTSSR
2498	047340				28\$:	CKLOOP			;LOOP IF SELECTED
	047340	104406						TRAP	C\$CLP1
2499	047342	005723				TST	(R3)+		;BUMP RECORD COUNTER
2500	047344	020327	001002			CMP	R3,#514.		;AT MAX SIZE YET
2501	047350	001351				BNE	27\$;BR, IF NOT AT END OF LOOP
2502	047352	004737	011074			JSR	PC,REWIND		;CALL TAPE REWIND COMMAND
2503	047356	103411				BCS	30\$;BR, IF NO PROBLEM
2504	047360	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS
2505	047364	010004				MOV	RO,R4		;SET UP REWIND PACKET ADDRESS
2506	047366	005237	002214			INC	FATFLG		;ERROR COUNT
2510	047372					ERRHRD	ERRNO,T32RWN,PKTSSR		;REWIND NOT ACCEPTED
	047372	104456						TRAP	C\$ERHRD
	047374	000625						.WORD	405

TEST 4: ERASE AND OPERATION INCOMPLETE

	047376	051730					.WORD	T32RWN
	047400	012126					.WORD	PKTSSR
2511	047402		30\$:	CKLOOP		;LOOP IF SELECTED		
	047402	104406					TRAP	C\$CLP1
2512	047404	013701	051410	MOV	T32BFR+6,R1	;PICK UP XSTO		
2513	047410	010102		MOV	R1,R2	;SET UP EXPECTED		
2514	047412	052702	000002	BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED		
2515	047416	020102		CMP	R1,R2	;DOES EXP = REC'D		
2516	047420	001406		BEQ	40\$;BR, IF EQUAL (OK)		
2517	047422	005237	002214	INC	FATFLG	;ERROR COUNT		
2521	047426			ERRHRD	ERRNO,T32BOE,EXPREC	;TAPE AT BOT AFTER ERASE		
	047426	104456					TRAP	C\$ERHRD
	047430	000626					.WORD	406
	047432	052416					.WORD	T32BOE
	047434	015554					.WORD	EXPREC
2522	047436		40\$:	CKLOOP		;LOOP IF SELECTED		
	047436	104406					TRAP	C\$CLP1
2523	047440	012737	140411	MOV	#140411,T32PK3	;ERASE TAPE,CVC=1,ACK COMMAND		
2524	047446	012704	051500	MOV	#T32PK3,R4	;SET UP R4 WITH PACKET ADDRESS		
2525	047452	010465	000000	MOV	R4,TSDB(R5)	;ISSUE COMMAND		
2526	047456	004737	016330	JSR	PC,WAITF	;WAIT FOR SSR TO SET		
2527	047462	016501	000002	MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
2528	047466	012702	000200	MOV	#SSR,R2	;SET UP EXPECTED		
2529	047472	020102		CMP	R1,R2	;ARE THEY EQUAL		
2530	047474	001406		BEQ	50\$;BR, IF OK		
2531	047476	005237	002214	INC	FATFLG	;ERROR COUNT		
2535	047502			ERRHRD	ERRNO,T32ERA,PKTSSR	;TSSR INCORRECT AFTER ERASE DATA		
	047502	104456					TRAP	C\$ERHRD
	047504	000627					.WORD	407
	047506	052046					.WORD	T32ERA
	047510	012126					.WORD	PKTSSR
2536	047512		50\$:	CKLOOP		;LOOP IF SELECTED		
	047512	104406					TRAP	C\$CLP1
2537	047514	013701	051410	MOV	T32BFR+6,R1	;PICK UP XSTO		
2538	047520	010102		MOV	R1,R2	;SET UP EXPECTED		
2539	047522	042702	000002	BIC	#BIT1,R2	;SET BOT BIT IN EXPECTED		
2540	047526	020102		CMP	R1,R2	;DOES EXP = REC'D		
2541	047530	001406		BEQ	55\$;BR, IF EQUAL (OK)		
2542	047532	005237	002214	INC	FATFLG	;ERROR COUNT		
2546	047536			ERRHRD	ERRNO,T32BOE,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	047536	104456					TRAP	C\$ERHRD
	047540	000630					.WORD	408
	047542	052416					.WORD	T32BOE
	047544	015554					.WORD	EXPREC
2547	047546		55\$:	CKLOOP		;LOOP IF SELECTED		
	047546	104406					TRAP	C\$CLP1
2548	047550	013737	003116	MOV	FREE,T32RB	;ADDRESS OF BUFFER		
2549	047556	012737	140401	MOV	#140401,T32PK3	;READ REVERSE,ACK,CVC=1 COMMAND		
2550	047564	012737	000400	MOV	#256.,T32SZ	;SET UP THE SIZE OF RECORD		
2551	047572	012704	051500	MOV	#T32PK3,R4	;SET UP R4 WITH PACKET ADDRESS		
2552	047576	010465	000000	MOV	R4,TSDB(R5)	;ISSUE COMMAND		
2553	047602	004737	016330	JSR	PC,WAITF	;WAIT FOR SSR TO SET		
2554	047606	016501	000002	MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
2555	047612	012702	100204	MOV	#SSR!SC!BIT2,R2	;SET UP EXPECTED TAPE STATUS ALERT		
2556	047616	020102		CMP	R1,R2	;ARE THEY EQUAL		
2557	047620	001406		BEQ	180\$;BR, IF OK		
2558	047622	005237	002214	INC	FATFLG	;ERROR COUNT		

TEST 4: ERASE AND OPERATION INCOMPLETE

2659	050102	004737	017502			JSR	PC,FILLMEM		;CALL MEMORY FILLER
2660	050106	013737	003116	051502		MOV	FREE,T32WB		;STARTING WRITE BUFFER ADDRESS
2661	050114	012737	140005	051500	65\$:	MOV	#140005,T32PK3		;WRITE DATA,CVC=1,ACK COMMAND
2662	050122	012704	051500			MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS
2663	050126	010300				MOV	R3,R0		;SET PATTERN IN CORRECT REGISTER
2664	050130	004737	017502			JSR	PC,FILLMEM		;FILL MEMORY WITH RECORD SIZE
2665	050134	010337	051506			MOV	R3,T32SZ		;SET UP RECORD SIZE IN PACKET
2666	050140	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND
2667	050144	004737	016330			JSR	PC,WAITF		;WAIT FOR SSR TO SET
2668	050150	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS
2669	050154	012702	000200			MOV	#SSR,R2		;SET UP EXPECTED
2670	050160	020102				CMP	R1,R2		;ARE THEY EQUAL
2671	050162	001406				BEQ	80\$;BR, IF OK
2672	050164	005237	002214			INC	FATFLG		;ERROR COUNT
2676	050170					ERRHRD	ERRNO,T32WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA
	050170	104456						TRAP	C\$ERHRD
	050172	000637						.WORD	415
	050174	052566						.WORD	T32WDC
	050176	012126						.WORD	PKTSSR
2677	050200				80\$:	CKLOOP			;LOOP IF SELECTED
	050200	104406						TRAP	C\$CLP1
2678	050202	005723				TST	(R3)+		;BUMP RECORD SIZE COUNTER
2679	050204	020327	000156			CMP	R3,#110.		;AT 160 SIZE YET
2680	050210	001341				BNE	65\$;BR, IF MORE RECORDS TO WRITE
2681	050212	004737	011074			JSR	PC,REWIND		;CALL TAPE REWIND COMMAND
2682	050216	103407				BCS	230\$;BR, IF NO PROBLEM
2683	050220	010001				MOV	R0,R1		;SAVE TSSR
2684	050222	005237	002214			INC	FATFLG		;ERROR COUNT
2688	050226					ERRHRD	ERRNO,T32RWN,EXPREC		;REWIND NOT ACCEPTED
	050226	104456						TRAP	C\$ERHRD
	050230	000640						.WORD	416
	050232	051730						.WORD	T32RWN
	050234	015554						.WORD	EXPREC
2689	050236				230\$:	CKLOOP			;LOOP IF SELECTED
	050236	104406						TRAP	C\$CLP1
2690	050240	013701	051410			MOV	T32BFR+6,R1		;PICK UP XSTO
2691	050244	010102				MOV	R1,R2		;SET UP EXPECTED
2692	050246	052702	000002			BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED
2693	050252	020102				CMP	R1,R2		;DOES EXP = REC'D
2694	050254	001406				BEQ	240\$;BR, IF EQUAL (OK)
2695	050256	005237	002214			INC	FATFLG		;ERROR COUNT
2699	050262					ERRHRD	ERRNO,T32BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND
	050262	104456						TRAP	C\$ERHRD
	050264	000641						.WORD	417
	050266	051546						.WORD	T32BOT
	050270	015554						.WORD	EXPREC
2700	050272				240\$:	CKLOOP			;LOOP IF SELECTED
	050272	104406						TRAP	C\$CLP1
2701	050274	012703	000001			MOV	#1,R3		;SET UP FOR SPACE COMMAND
2702	050300	004737	010544			JSR	PC,SPACE		;ISSUE SPACE COMMAND 1 FORWARD
2703	050304	012737	140411	051500	265\$:	MOV	#140411,T32PK3		;ERASE DATA,ACK COMMAND
2704	050312	012704	051500			MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS
2705	050316	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND
2706	050322	004737	016330			JSR	PC,WAITF		;WAIT FOR SSR TO SET
2707	050326	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS
2708	050332	012702	000200			MOV	#SSR,R2		;SET UP EXPECTED
2709	050336	020102				CMP	R1,R2		;ARE THEY EQUAL

TEST 4: ERASE AND OPERATION INCOMPLETE

2805	050642	012704	051360		MOV	#T32PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS	
2806	050646	004737	010742		JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS	
2807	050652	103407			BCS	23\$;BR, IF COMMAND ISSUED OK	
2808	050654	005237	002214		INC	FATFLG		;ERROR COUNT	
2812	050660	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
2813	050662				ERRHRD	ERRNO,WRTMSG,SFMSG		;WRITE CHARACTERISTICS FAILED	
	050662	104456						TRAP	C\$ERHRD
	050664	000646						.WORD	422
	050666	005052						.WORD	WRTMSG
	050670	012114						.WORD	SFMSG
2814	050672			23\$:	CKLOOP			;LOOP IF SELECTED	
	050672	104406						TRAP	C\$CLP1
2815	050674	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND	
2816	050700	103411			BCS	30\$;BR, IF NO PROBLEM	
2817	050702	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS	
2818	050706	010004			MOV	R0,R4		;GET PACKET ADDRESS	
2819	050710	005237	002214		INC	FATFLG		;ERROR COUNT	
2823	050714				ERRHRD	ERRNO,T32RWN,PKTSSR		;REWIND NOT ACCEPTED	
	050714	104456						TRAP	C\$ERHRD
	050716	000647						.WORD	423
	050720	051730						.WORD	T32RWN
	050722	012126						.WORD	PKTSSR
2824	050724			30\$:	CKLOOP			;LOOP IF SELECTED	
	050724	104406						TRAP	C\$CLP1
2825	050726	013701	051410		MOV	T32BFR+6,R1		;PICK UP XSTO	
2826	050732	010102			MOV	R1,R2		;SET UP EXPECTED	
2827	050734	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED	
2828	050740	020102			CMP	R1,R2		;DOES EXP = REC'D	
2829	050742	001406			BEQ	40\$;BR, IF EQUAL (OK)	
2830	050744	005237	002214		INC	FATFLG		;ERROR COUNT	
2834	050750				ERRHRD	ERRNO,T32BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND	
	050750	104456						TRAP	C\$ERHRD
	050752	000650						.WORD	424
	050754	051546						.WORD	T32BOT
	050756	015554						.WORD	EXPREC
2835	050760			40\$:	CKLOOP			;LOOP IF SELECTED	
	050760	104406						TRAP	C\$CLP1
2836	050762	012737	140411	051500	65\$:	MOV	#140411,T32PK3	;ERASE DATA,CVC=1,ACK COMMAND	
2837	050770	012704	051500		MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS	
2838	050774	010337	051506		MOV	R3,T32SZ		;SET UP RECORD SIZE IN PACKET	
2839	051000	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND	
2840	051004	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET	
2841	051010	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS	
2842	051014	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED	
2843	051020	020102			CMP	R1,R2		;ARE THEY EQUAL	
2844	051022	001757			BEQ	65\$;BR, IF OK	
2845	051024	032701	000004		BIT	#BIT2,R1		;CHECK FOR TAPE STATUS ALERT	
2846	051030	001006			BNE	80\$;BR, IF TAPE STATUS ALERT SET	
2847	051032	005237	002214		INC	FATFLG		;ERROR COUNT	
2851	051036				ERRHRD	ERRNO,T32WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA	
	051036	104456						TRAP	C\$ERHRD
	051040	000651						.WORD	425
	051042	052566						.WORD	T32WDC
	051044	012126						.WORD	PKTSSR
2852	051046			80\$:	CKLOOP			;LOOP IF SELECTED	
	051046	104406						TRAP	C\$CLP1
2853	051050	013701	051410		MOV	T32BFR+6,R1		;PICK UP XSTO	

TEST 4: ERASE AND OPERATION INCOMPLETE

```

2854 051054 010102          MOV     R1,R2          ;SET UP EXPECTED
2855 051056 052702 000001  BIS     #BIT0,R2      ;SET EOT BIT IN EXPECTED
2856 051062 020102          CMP     R1,R2          ;DOES EXP = REC'D
2857 051064 001406          BEQ     240$          ;BR, IF EQUAL (OK)
2858 051066 005237 002214  INC     FATFLG        ;ERROR COUNT
2862 051072          ERRHRD  ERRNO,T32EOT,EXPREC ;TAPE NOT AT EOT AFTER ERASE COMMANDS
      051072 104456          TRAP   C$ERHRD
      051074 000652          .WORD  426
      051076 051641          .WORD  T32EOT
      051100 015554          .WORD  EXPREC
2863 051102          240$: CKLOOP        ;LOOP IF SELECTED
      051102 104406          TRAP   C$CLP1
2864 051104 012703 051510  MOV     #T32CMD,R3    ;STARTING RECORD SIZE
2865 051110 013737 003116 051502  MOV     FREE,T32RB    ;STARTING READ BUFFER ADDRESS
2866 051116 011337 051500 265$:  MOV     (R3),T32PK3   ;READ DATA,ACK COMMAND
2867 051122 012704 051500  MOV     #T32PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
2868 051126 012700 177777  MOV     #177777,R0    ;SET PATTERN IN CORRECT REGISTER
2869 051132 004737 017502  JSR     PC,FILLMEM    ;FILL MEMORY WITH ALL ONES
2870 051136 012737 000144 051506  MOV     #100.,T32SZ   ;SET UP RECORD SIZE IN PACKET
2871 051144 010465 000000  MOV     R4,TSDB(R5)   ;ISSUE COMMAND
2872 051150 012737 000062 051544  MOV     #50.,T32DLY   ;SET UP DELAY COUNTER
2873 051156 004737 016330 270$:  JSR     PC,WAITF      ;WAIT FOR SSR TO SET
2874 051162 016501 000002  MOV     TSSR(R5),R1   ;GET TSSR CONTENTS
2875 051166 012702 100214  MOV     #SSR!SC!BIT2!BIT3,R2 ;SET UP EXPECTED
2876 051172 020102          CMP     R1,R2          ;ARE THEY EQUAL
2877 051174 001425          BEQ     280$          ;BR, IF OK
2878 051176          DELAY  250          ;DELAY FOR SSR TO BE SET
      051176 012727 000250  MOV     #250,(PC)+
      051202 000000          .WORD  0
      051204 013727 002116  MOV     L$DLY,(PC)+
      051210 000000          .WORD  0
      051212 005367 177772  DEC     -6(PC)
      051216 001375          BNE     -.4
      051220 005367 177756  DEC     -22(PC)
      051224 001367          BNE     .-20
2879 051226 005337 051544  DEC     T32DLY        ;COUNT DELAY ROUTINE DOWN
2880 051232 001351          BNE     270$          ;BR, IF DELAY HAS NOT ENDED
2881 051234 005237 002214  INC     FATFLG        ;ERROR COUNT
2885 051240          ERRHRD  ERRNO,T32ECF,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      051240 104456          TRAP   C$ERHRD
      051242 000653          .WORD  427
      051244 052505          .WORD  T32ECF
      051246 012126          .WORD  PKTSSR
2886 051250          280$: CKLOOP        ;LOOP IF SELECTED
      051250 104406          TRAP   C$CLP1
2887 051252 013701 051416  MOV     T32BFR+14,R1  ;PICK UP XST3
2888 051256 010102          MOV     R1,R2          ;SET UP EXPECTED
2889 051260 052702 000100  BIS     #BIT6,R2      ;SET OPI BIT IN EXPECTED
2890 051264 020102          CMP     R1,R2          ;IS OPI BIT SET
2891 051266 001406          BEQ     290$          ;BR, IF BIT IS SET
2892 051270 005237 002214  INC     FATFLG        ;ERROR COUNT
2896 051274          ERRHRD  ERRNO,T32OPI,EXPREC ;OPI BIT NOT SET
      051274 104456          TRAP   C$ERHRD
      051276 000654          .WORD  428
      051300 052633          .WORD  T32OPI
      051302 015554          .WORD  EXPREC
2897 051304          290$: CKLOOP        ;LOOP IF SELECTED

```


TEST 4: ERASE AND OPERATION INCOMPLETE

```

2957
2958
2959
2960
2961 051510
2962 051510 140410
2963 051512 141410
2964 051514 140401
2965 051516 141001
2966 051520 161401
2967 051522 161001
2968 051524 141401
2969 051526 140001
2970 051530 141410
2971 051532 141010
2972 051534 141005
2973 051536 177777
2974
2975
2976 051540 000000
2977 051542 000000
2978 051544 000000
2979
2980
2981
2982
2983 051546 124 141 160
2984 051641 124 141 160
2985 051730 122 145 167
2986 051777 124 123 123
2987 052046 124 123 123
2988 052113 124 123 102
2989 052166 122 105 101
2990 052264 124 123 123
2991 052341 124 123 123
2992 052416 102 117 124
2993 052505 105 122 101
2994 052566 124 123 123
2995 052633 117 120 111
2996 052670 105 162 141
2997
2998
2999
3000
3001
3002
3003
3004
3005 052730
3006 052730
3007 052734 012701 051360
3008 052740 012721 100004
3009 052744 012721 051370
3010 052750 005021
3011 052752 012721 000012
3012 052756 012721 051402
3013 052762 005021

```

```

;
; .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T32CMD:
;SPACE RECORDS REVERSE
;SKIP TAPE MARKS REVERSE
;READ REVERSE
;REREAD PREVIOUS (OPP=0)
;REREAD NEXT (OPP=1)
;REREAD PREVIOUS (OPP=1)
;REREAD NEXT (OPP=0)
;READ NEXT
;SKIP TAPE MARKS REVERSE
;SKIP RECORDS FORWARD
;WRITE DATA RETRY
;END OF DATA
;
T32CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T32CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T32DLY: .WORD 0 ;DELAY COUNTER
;
; *
;LOCAL TEXT MESSAGES FOR TEST
;-
T32BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
T32EOT: .ASCIZ 'Tape Status Alert During Erase To EOT, But EOT Not Set'
T32RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
T32AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
T32ERA: .ASCIZ 'TSSR Not Correct After ERASE Command'
T32BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
T32RIB: .ASCIZ 'READ REVERSE, After ERASE From BOT, Failed To Set RIB In XST3'
T32SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
T32TSA: .ASCIZ 'TSSR Not Correct After READ REVERSE Into BOT'
T32BOE: .ASCIZ 'BOT (XST0) Still Set After Erase From Tape's BOT Marker'
T32ECF: .ASCIZ 'ERASE Failed To Clear Tape (Erase) Tape Properly'
T32WDC: .ASCIZ 'TSSR Not Correct After ERASE Command'
T32OPI: .ASCIZ 'OPI Bit (XST3) Failed To Set'
TST32ID: .ASCIZ 'Erase And Operation Incomplete'
; .EVEN
;
; *
;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
;WRITE SUBSYSTEM MEMORY COMMAND
;-
T32REST:
;SAVE THE REGISTERS
;START OF THE PACKET
;WRITE SUBSYSTEM MEM. WITH ACK.
;ADDRESS OF CHARAISTICS DATA BLOCK
;EXTENDED ADDRESS
;SIZE OF DATA BLOCK IN BYTES
;ADDRESS OF MESSAGE BUFFER
SAVREG
MOV #T32PACKET,R1
MOV #100004,(R1)+
MOV #T32DATA,(R1)+
CLR (R1)+
MOV #10,(R1)+
MOV #T32BFR,(R1)+
CLR (R1)+

```

TEST 4: ERASE AND OPERATION INCOMPLETE

```

3014 052764 012721 000024      MOV    #20.,(R1)+      ;LENGTH OF MESSAGE BUFFER
3015 052770 005021              CLR    (R1)+
3016 052772 012711 000000      MOV    #0,(R1)        ;SELECT DRIVE ZERO
3017 052776 012702 000030      MOV    #24.,R2        ;NUMBER OF LOCATIONS TO BE CLEARED
3018 053002 012762 177777 051402 64$: MOV    #177777,T32BFR(R2) ;ALL ONES TO MESSAGE BUFFER
3019 053010 005742              TST    -(R2)          ;NEXT LOCATION
3020 053012 022702 000000      CMP    #0,R2          ;AT END OF LOOP YET
3021 053016 001371              BNE    64$            ;KEEP GOING UNTIL DONE
3022 053020 000207              RTS    PC              ;RETURN
3023
3024 053022              T32RT2:
3025 053022              SAVREG                ;SAVE THE REGISTERS
3026 053026 012701 051470      MOV    #T32PK2,R1     ;START OF THE PACKET
3027 053032 012721 100006      MOV    #100006,(R1)+  ;WRITE SUBSYSTEM MEM. WITH ACK.
3028 053036 005021              CLR    (R1)+          ;ADDRESS OF DATA BLOCK
3029 053040 005021              CLR    (R1)+          ;EXTENDED ADDRESS
3030 053042 012721 000006      MOV    #6.,(R1)+     ;SIZE OF DATA BLOCK IN BYTES
3031 053046 005021              CLR    (R1)+
3032 053050 000207              RTS    PC              ;RETURN
3033 053052              T32RT3:
3034 053052              SAVREG                ;SAVE REGISTERS
3035 053056 012701 051500      MOV    #T32PK3,R1     ;SET UP POINTER ADDRESS
3036 053062 005021              CLR    (R1)+          ;COMMAND SPACE
3037 053064 005021              CLR    (R1)+          ;ADDRESS OF DATA BLOCK
3038 053066 005021              CLR    (R1)+          ;EXTENDED ADDRESS
3039 053070 005011              CLR    (R1)           ;SIZE OF DATA TRANSFER BLOCK
3040 053072 000207              RTS    PC              ;RETURN
3041 053074              ENDTST
      053074
      053074 104401

```

L10053: TRAP C\$ETST

3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068

```

.SBTTL TEST 5: DATA PARITY TEST
;+
;
;
;
;
;TEST 5 -- Data Parity Test
;
;This test verifies that the data parity circuitry in both the controller and the
;transport is operating properly by forcing data records with wrong parity to be
;written onto tape and checking the results obtained when the data is read. The
;following test sequence is performed:
;
; 1. A Write Characteristics command is issued and the resulting status is
;    examined to determine the states of the Extended Features and Buffering
;    Enable switches on the controller module. If buffering is disabled, no
;    further actions need be taken in this step and the program proceeds to
;    the next step. If buffering is enabled, it is disabled via the Buffer
;    Control field in the extended characteristics data word supplied by a
;    Write Characteristics command. (The module must be in Extended mode,
;    so if it is not already, a Write Subsystem Memory command is issued to
;    change the logical sense of the Extended Features switch.)
;
; 2. The Write Subsystem Memory command is used to set the Force Wrong
;    Parity control flip-flop.
;

```

TEST 5: DATA PARITY TEST

- 3069 ;
 - 3070 ;
 - 3071 ;
 - 3072 ;
 - 3073 ;
 - 3074 ;
 - 3075 ;
 - 3076 ;
 - 3077 ;
 - 3078 ;
 - 3079 ;
 - 3080 ;
 - 3081 ;
 - 3082 ;
 - 3083 ;
 - 3084 ;
 - 3085 ;
 - 3086 ;
 - 3087 ;
 - 3088 ;
 - 3089 ;
 - 3090 ;
 - 3091 ;
 - 3092 ;
 - 3093 ;
 - 3094 ;
 - 3095 ;
 - 3096 ;
 - 3097 ;
 - 3098 ;
 - 3099 ;
 - 3100 ;
 - 3101 ;
 - 3102 ;
 - 3103 ;
 - 3104 ;
 - 3105 ;
 - 3106 ;
3. The tape is rewound.
 4. A Write Data command is issued to write a data record containing all 0's. It is verified that this command results in Recoverable Error termination (TC=4) and that the Uncorrectable Data Error (UNC) error bit is set.
 5. The previous step is repeated for each data value 2 through 377 (octal).
 6. The tape is rewound.
 7. A Read Next command is issued to read a record with faulty parity. It is verified that this command results in Recoverable Error termination (TC=4) and that both the Uncorrectable Data (UNC) and Read Bus Parity (RBP) error bits are set. It is also verified that the data actually read is correct.
 8. A Read Reverse command with OPP=1 is issued to read, in reverse, the same record with faulty parity as read in the previous step. It is verified that this command results in Recoverable Error termination (TC=4) and that both the Uncorrectable Data (UNC) and Read Bus Parity (RBP) error bits are set. It is also verified that the data actually read is correct.
 9. Tape is spaced forward one record.
 10. The previous three steps are executed for each test record originally written.
 11. The controller is initialized to clear the special test conditions previously set up.

```

3106 053076          BGNTST
      053076
3107 053076 012737 006413 002172      MOV    #EPRT2,EPRTSW      ;SECONDARY ERROR MESSAGE
3112 053104 012700 055675              MOV    #TST33ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
3113 053110 004737 016570              JSR    PC,TSTSETUP      ;DO INITIAL TEST SETUP
3114 053114 012737 000005 002210      MOV    #5,LOOPCNT      ;PERFORM 5 ITERATIONS
3115 053122 005037 054746              CLR    T33CNT          ;CLEAR TAPE RECORD COUNTER
3116
3117 ;+
3118 053126          T33LOOP:
3119 053126          BGNSUB
      053126
      053126 104402
3120 053130 005037 002216              CLR    INTRECV         ;INTERRUPT INDICATOR
3121 053134 005037 054746              CLR    T33CNT         ;TIMER FOR WRITE DATA SPACING
3122 053140 005037 054750              CLR    T33CNU        ;TIMER FOR WRITE DATA RETRY SPACING
3123 053144 004737 055712              JSR    PC,T33REST     ;SET COMMAND PACKET
3124 053150 004737 056004              JSR    PC,T33RT2     ;SET UP OTHER COMMAND PACKET
3125 053154 004737 056046              JSR    PC,T33RT3     ;SET UP OTHER COMMAND PACKET
3126 053160 012737 176750 054752      MOV    #65000.,T33DLY ;SET UP DELAY COUNTER

```


TEST 5: DATA PARITY TEST

```

3170 053374          40$:  CKLOOP          ;LOOP IF SELECTED
      053374 104406          TRAP  C$CLP1
3171 053376 005737 002220  42$:  TST  EXTFEA          ;CHECK FOR EXTENDED FEATURES SW SWITCH
3172 053402 001025          BNE  55$          ;BR IF SWITCH IS ON
3173 053404 112737 000200 054731  MOVB #200,T33BS1      ;WRITE MISCELLANEOUS CONT/READ STATUS
3174 053412 112737 000010 054730  MOVB #10,T33BS0      ;FUNC. SEL. BIT (TURN ON EXTFEA SWITCH)
3175 053420 012704 054710  MOV  #T33PK2,R4      ;WRITE SUBSYS MEM PACKET
3176 053424 010465 000000  MOV  R4,TSDB(R5)     ;ISSUE COMMAND
3177 053430 004737 016416  JSR  PC,CHKTSSR      ;WAIT FOR SSR
3178 053434 103407          BCS  50$          ;BR, IF NO ERROR
3179 053436 010001          MOV  R0,R1          ;ERROR, SAVE TSSR
3180 053440 005237 002214  INC  FATFLG          ;ERROR COUNT
3184 053444          ERRHRD ERRNO,T33SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      053444 104456          TRAP  C$ERHRD
      053446 000771          .WORD 505
      053450 055271          .WORD T33SSR
      053452 012126          .WORD PKTSSR
3185 053454          50$:  CKLOOP          ;LOOP IF SELECTED
      053454 104406          TRAP  C$CLP1
3186 053456 005737 002222  55$:  TST  BENBSW          ;CHECK FOR BUFFER ENABLED
3187 053462 001426          BEQ  70$          ;BR, IF BUFFERING NOT ENABLED
3188 053464 013737 002174 054620  MOV  UNITN,T33DSW   ;SET UP UNIT NUMBER
3189 053472 042737 000020 054620  BIC  #BIT4,T33DSW   ;BUFFER DISABLE
3190 053500 052737 000010 054620  BIS  #BIT3,T33DSW   ;BUFFER DISABLE SEND 01 TO BITS 4 AND 3
3191 053506 012704 054600  MOV  #T33PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
3192 053512 004737 010742  JSR  PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
3193 053516 103407          BCS  60$          ;BR, IF COMMAND ISSUED OK
3194 053520 005237 002214  INC  FATFLG          ;ERROR COUNT
3198 053524 010001          MOV  R0,R1          ;SAVE CONTENTS OF TSSR
3199 053526          ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      053526 104456          TRAP  C$ERHRD
      053530 000772          .WORD 506
      053532 005052          .WORD WRTMSG
      053534 012114          .WORD SFIMSG
3200 053536          60$:  CKLOOP          ;LOOP IF SELECTED
      053536 104406          TRAP  C$CLP1
3201 053540          70$:
3202 053540 112737 000100 054731  MOVB #100,T33BS1    ;WRITE MISCELLANEOUS CONT/READ STATUS
3203 053546 112737 000011 054730  MOVB #11,T33BS0     ;FUNC. SEL. BIT (SET WRONG PARITY)
3204 053554 012704 054710  MOV  #T33PK2,R4     ;WRITE SUBSYS MEM PACKET
3205 053560 010465 000000  MOV  R4,TSDB(R5)   ;ISSUE COMMAND
3206 053564 004737 016416  JSR  PC,CHKTSSR    ;WAIT FOR SSR
3207 053570 103407          BCS  80$          ;BR, IF NO ERROR
3208 053572 010001          MOV  R0,R1          ;ERROR, SAVE TSSR
3209 053574 005237 002214  INC  FATFLG          ;ERROR COUNT
3213 053600          ERRHRD ERRNO,T33SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      053600 104456          TRAP  C$ERHRD
      053602 000773          .WORD 507
      053604 055271          .WORD T33SSR
      053606 012126          .WORD PKTSSR
3214 053610          80$:  CKLOOP          ;LOOP IF SELECTED
      053610 104406          TRAP  C$CLP1
3215 053612 012703 000026          MOV  #22.,R3          ;NUMBER OF RECORDS TO BE WRITTEN
3216 053616 013737 003116 054722  MOV  FREE,T33WB     ;STARTING WRITE BUFFER ADDRESS
3217 053624 005037 054750          CLR  T33CNU         ;MAKE SURE ITS CLEAR
3218 053630 012737 140005 054720 110$:  MOV  #140005,T33PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
3219 053636 012704 054720          MOV  #T33PK3,R4     ;SET UP R4 WITH PACKET ADDRESS

```

TEST 5: DATA PARITY TEST

3220	053642	012737	000024	054726	MOV	#20.,T33SZ	;SET UP RECORD SIZE IN PACKET		
3221	053650	013777	054750	127240	MOV	T33CNU,#FREE	;MEMORY FILLED WITH DATA IN RECORD		
3222	053656	005237	054750		INC	T33CNU	;READY FOR NEXT RECORD		
3223	053662	010465	000000		MOV	R4,T33DB(R5)	;ISSUE COMMAND		
3224	053666	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
3225	053672	016501	000002		MOV	T33R(R5),R1	;GET T33R CONTENTS		
3226	053676	012702	100210		MOV	#SSR!SC!BIT3,R2	;SET UP EXPECTED		
3227	053702	020102			CMP	R1,R2	;ARE THEY EQUAL		
3228	053704	001406			BEQ	120\$;BR, IF OK		
3229	053706	005237	002214		INC	FATFLG	;ERROR COUNT		
3233	053712				ERRHRD	ERRNO,T33WPW,PKTSSR	;T33R INCORRECT AFTER WRITE DATA		
	053712	104456					TRAP	C\$ERHRD	
	053714	000774					.WORD	508	
	053716	055032					.WORD	T33WPW	
	053720	012126					.WORD	PKTSSR	
3234	053722				120\$:	CKLOOP	;LOOP IF SELECTED		
	053722	104406					TRAP	C\$CLP1	
3235	053724	013701	054632		MOV	T33BFR+10,R1	;PICK UP XST1		
3236	053730	010102			MOV	R1,R2	;SET UP EXPECTED		
3237	053732	052702	000002		BIS	#BIT1,R2	;SET UNC BIT IN EXPECTED		
3238	053736	020102			CMP	R1,R2	;DOES EXP = REC'D		
3239	053740	001406			BEQ	130\$;BR, IF EQUAL (OK)		
3240	053742	005237	002214		INC	FATFLG	;ERROR COUNT		
3244	053746				ERRHRD	ERRNO,T33UNC,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	053746	104456					TRAP	C\$ERHRD	
	053750	000775					.WORD	509	
	053752	055112					.WORD	T33UNC	
	053754	015554					.WORD	EXPREC	
3245	053756				130\$:	CKLOOP	;LOOP IF SELECTED		
	053756	104406					TRAP	C\$CLP1	
3246	053760	005303			DEC	R3	;DEC RECORD COUNTER		
3247	053762	001322			BNE	110\$;BR, IF MORE RECORDS TO WRITE		
3248	053764	004737	011074		JSR	PC,REWIND	;CALL TAPE REWIND COMMAND		
3249	053770	103411			BCS	140\$;BR, IF NO PROBLEM		
3250	053772	016501	000002		MOV	T33R(R5),R1	;GET T33R CONTENTS		
3251	053776	C10004			MOV	R0,R4	;GET PACKET ADDRESS		
3252	054000	005237	002214		INC	FATFLG	;ERROR COUNT		
3256	054004				ERRHRD	ERRNO,T33RWN,PKTSSR	;REWIND NOT ACCEPTED		
	054004	104456					TRAP	C\$ERHRD	
	054006	000776					.WORD	510	
	054010	055450					.WORD	T33RWN	
	054012	012126					.WORD	PKTSSR	
3257	054014				140\$:	CKLOOP	;LOOP IF SELECTED		
	054014	104406					TRAP	C\$CLP1	
3258	054016	013701	054630		MOV	T33BFR+6,R1	;PICK UP XST0		
3259	054022	010102			MOV	R1,R2	;SET UP EXPECTED		
3260	054024	052702	000002		BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED		
3261	054030	020102			CMP	R1,R2	;DOES EXP = REC'D		
3262	054032	001406			BEQ	150\$;BR, IF EQUAL (OK)		
3263	054034	005237	002214		INC	FATFLG	;ERROR COUNT		
3267	054040				ERRHRD	ERRNO,T33BOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	054040	104456					TRAP	C\$ERHRD	
	054042	000777					.WORD	511	
	054044	055355					.WORD	T33BOT	
	054046	015554					.WORD	EXPREC	
3268	054050				150\$:	CKLOOP	;LOOP IF SELECTED		
	054050	104406					TRAP	C\$CLP1	

TEST 5: DATA PARITY TEST

```

3269 054052 005037 054750          CLR      T33CNU          ;CLEAR DATA VALUE IN RECORD
3270 054056 012703 000024          MOV      @20.,R3        ;RECORD SIZE
3271 054062 013737 003116 054722 155$: MOV      FREE,T33RB     ;STARTING WRITE BUFFER ADDRESS
3272 054070 012737 140001 054720   MOV      @140001,T33PK3 ;READ DATA,CVC=1,ACK COMMAND
3273 054076 012704 054720          MOV      @T33PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
3274 054102 012737 000024 054726   MOV      @20.,T33SZ    ;SET UP RECORD SIZE IN PACKET
3275 054110 010465 000000          MOV      R4,T33DB(R5)  ;ISSUE COMMAND
3276 054114 004737 016330          JSR      PC,WAITF      ;WAIT FOR SSR TO SET
3277 054120 016501 000002          MOV      T33R(R5),R1  ;GET T33R CONTENTS
3278 054124 012702 100210          MOV      @SSR!SC!BIT3,R2 ;SET UP EXPECTED
3279 054130 020102                CMP      R1,R2        ;ARE THEY EQUAL
3280 054132 001406                BEQ      160$         ;BR, IF OK
3281 054134 005237 002214          INC      FATFLG       ;ERROR COUNT
3285 054140          ERRHRD  ERRNO,T33WDC,PKTSSR ;T33R INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    512
                                .WORD    T33WDC
                                .WORD    PKTSSR
                                054140 104456
                                054142 001000
                                054144 055517
                                054146 012126
3286          054150          160$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                054150 104406
3287 054152 013701 054632          MOV      T33BFR+10,R1 ;PICK UP XST1
3288 054156 010102                MOV      R1,R2        ;SET UP EXPECTED
3289 054160 052702 000002          BIS      @BIT1,R2     ;SET UNC BIT IN EXPECTED
3290 054164 020102                CMP      R1,R2        ;DOES EXP = REC'D
3291 054166 001406                BEQ      170$         ;BR, IF EQUAL (OK)
3292 054170 005237 002214          INC      FATFLG       ;ERROR COUNT
3296 054174          ERRHRD  ERRNO,T33UND,EXPREC ;UNC BIT NOT SET AFTER READ CMD.
                                TRAP      C$ERHRD
                                .WORD    513
                                .WORD    T33UND
                                .WORD    EXPREC
                                054174 104456
                                054176 001001
                                054200 055202
                                054202 015554
3297          054204          170$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                054204 104406
3298 054206 013701 054632          MOV      T33BFR+10,R1 ;PICK UP XST1
3299 054212 010102                MOV      R1,R2        ;SET UP EXPECTED
3300 054214 052702 000400          BIS      @BIT8,R2     ;SET RBP BIT IN EXPECTED
3301 054220 020102                CMP      R1,R2        ;DOES EXP = REC'D
3302 054222 001406                BEQ      180$         ;BR, IF EQUAL (OK)
3303 054224 005237 002214          INC      FATFLG       ;ERROR COUNT
3307 054230          ERRHRD  ERRNO,T33RBP,EXPREC ;READ BUS PARITY ERROR BIT NOT SET
                                TRAP      C$ERHRD
                                .WORD    514
                                .WORD    T33RBP
                                .WORD    EXPREC
                                054230 104456
                                054232 001002
                                054234 054754
                                054236 015554
3308          054240          180$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                054240 104406
3309 054242 017701 126650          MOV      @FREE,R1     ;GET DATA READ
3310 054246 013702 054750          MOV      T33CNU,R2   ;GET PATTERN
3311 054252 020102                CMP      R1,R2        ;ARE THEY EQUAL
3312 054254 001406                BEQ      182$         ;BR, IF OK
3313 054256 005237 002214          INC      FATFLG       ;ERROR COUNT
3317 054262          ERRHRD  ERRNO,T33DTA,EXPREC ;DATA NOT CORRECT
                                TRAP      C$ERHRD
                                .WORD    515
                                .WORD    T33DTA
                                .WORD    EXPREC
                                054262 104456
                                054264 001003
                                054266 055600
                                054270 015554
3318          054272          182$: CKLOOP          ;LOOP IF SELECTED

```


TEST 5: DATA PARITY TEST

3319	054272	013737	003116	054722	MOV	FREE,T33WB	; STARTING WRITE BUFFER ADDRESS	TRAP	C\$CLP1
3320	054302	012737	140401	054720	195\$:	MOV	#140401,T33PK3	; READ REVERSE DATA RETRY,ACK COMMAND	
3321	054310	012704	054720			MOV	#T33PK3,R4	; SET UP R4 WITH PACKET ADDRESS	
3322	054314	012737	000024	054726		MOV	#20.,T33SZ	; SET UP RECORD SIZE IN PACKET	
3323	054322	010465	000000			MOV	R4,TSDB(R5)	; ISSUE COMMAND	
3324	054326	004737	016330			JSR	PC,WAITF	; WAIT FOR SSR TO SET	
3325	054332	016501	000002			MOV	TSSR(R5),R1	; GET TSSR CONTENTS	
3326	054336	012702	100210			MOV	#SC!SSR!BIT3,R2	; SET UP EXPECTED	
3327	054342	020102				CMP	R1,R2	; ARE THEY EQUAL	
3328	054344	001406				BEQ	190\$; BR, IF OK	
3329	054346	005237	002214			INC	FATFLG	; ERROR COUNT	
3333	054352					ERRHRD	ERRNO,T33WDC,PKTSSR	; TSSR INCORRECT AFTER WRITE DATA	
	054352	104456						TRAP	C\$ERHRD
	054354	001004						.WORD	516
	054356	055517						.WORD	T33WDC
	054360	012126						.WORD	PKTSSR
3334	054362			190\$:	CKLOOP		; LOOP IF SELECTED		
	054362	104406						TRAP	C\$CLP1
3335	054364	013701	054632			MOV	T33BFR+10,R1	; PICK UP XST1	
3336	054370	010102				MOV	R1,R2	; SET UP EXPECTED	
3337	054372	052702	000002			BIS	#BIT1,R2	; SET UNC BIT IN EXPECTED	
3338	054376	020102				CMP	R1,R2	; DOES EXP = REC'D	
3339	054400	001406				BEQ	200\$; BR, IF EQUAL (OK)	
3340	054402	005237	002214			INC	FATFLG	; ERROR COUNT	
3344	054406					ERRHRD	ERRNO,T33UND,EXPREC	; TAPE NOT AT BOT AFTER REWIND	
	054406	104456						TRAP	C\$ERHRD
	054410	001005						.WORD	517
	054412	055202						.WORD	T33UND
	054414	015554						.WORD	EXPREC
3345	054416			200\$:	CKLOOP		; LOOP IF SELECTED		
	054416	104406						TRAP	C\$CLP1
3346	054420	013701	054632			MOV	T33BFR+10,R1	; PICK UP XST0	
3347	054424	010102				MOV	R1,R2	; SET UP EXPECTED	
3348	054426	052702	000400			BIS	#BIT8,R2	; SET RBP BIT IN EXPECTED	
3349	054432	020102				CMP	R1,R2	; DOES EXP = REC'D	
3350	054434	001406				BEQ	210\$; BR, IF EQUAL (OK)	
3351	054436	005237	002214			INC	FATFLG	; ERROR COUNT	
3355	054442					ERRHRD	ERRNO,T33RBP,EXPREC	; READ BUS PARITY ERROR BIT NOT SET	
	054442	104456						TRAP	C\$ERHRD
	054444	001006						.WORD	518
	054446	054754						.WORD	T33RBP
	054450	015554						.WORD	EXPREC
3356	054452			210\$:	CKLOOP		; LOOP IF SELECTED		
	054452	104406						TRAP	C\$CLP1
3357	054454	017701	126436			MOV	#FREE,R1	; GET DATA READ	
3358	054460	013702	054750			MOV	T33CNU,R2	; GET PATTERN	
3359	054464	020102				CMP	R1,R2	; ARE THEY EQUAL	
3360	054466	001406				BEQ	215\$; BR, IF OK	
3361	054470	005237	002214			INC	FATFLG	; ERROR COUNT	
3365	054474					ERRHRD	ERRNO,T33DTA,EXPREC	; DATA NOT CORRECT	
	054474	104456						TRAP	C\$ERHRD
	054476	001007						.WORD	519
	054500	055600						.WORD	T33DTA
	054502	015554						.WORD	EXPREC
3366	054504			215\$:	CKLOOP		; LOOP IF SELECTED		
	054504	104406						TRAP	C\$CLP1

TEST 5: DATA PARITY TEST

```

3427                                     .EVEN
3428                                     ;
3429                                     ;
3430                                     ;
3431 054730 T33BF2:
3432 054730 010 T33BS0: .BYTE 10 ;BSELO AREA
3433 054731 200 T33BS1: .BYTE 200 ;BSEL1 AREA
3434 054732 000000 T33S2: .WORD 0 ;SEL 2 AREA
3435 054734 000000 T33S3: .WORD 0 ;DATA AREA
3436                                     ;
3437                                     ;
3438                                     .EVEN
3439 ;TAPE MOTION PACKET COMMAND VALUES
3440
3441 054736 100205 T33RN: .WORD 100205 ;REREAD DATA (NEXT)
3442 054740 100605 T33WDR: .WORD 100605 ;REREAD DATA RETRY
3443 054742 102205 T33CON: .WORD 102205 ;WRITE CONTINOUS
3444 054744 177777 .WORD 177777 ;END OF DATA
3445
3446                                     ;
3447 054746 000000 T33CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
3448 054750 000000 T33CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
3449 054752 000000 T33DLY: .WORD 0 ;DELAY COUNTER
3450
3451 ;*
3452 ;LOCAL TEXT MESSAGES FOR TEST
3453 ;-
3454 054754 122 145 141 T33RBP: .ASCIZ 'Read Bus Parity Bit Not Set (XST1), Should Be'
3455 055032 124 123 123 T33WPW: .ASCIZ 'TSSR Incorrect After Wrong Parity Write Command'
3456 055112 125 116 103 T33UNC: .ASCIZ 'UNC Bit (XST1) Not Set After Wrong Parity WRITE Command'
3457 055202 125 116 103 T33UND: .ASCIZ 'UNC Bit (XST1) Not Set After Wrong Parity READ Command'
3458 055271 127 122 111 T33SSR: .ASCIZ 'WRITE MISSCELLANEOUS CONT/READ COMMAND Not Accepted'
3459 055355 124 141 160 T33BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
3460 055450 122 145 167 T33RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
3461 055517 124 123 123 T33WDC: .ASCIZ 'TSSR Not Correct After READ Wrong Parity Command'
3462 055600 104 141 164 T33DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
3463 055675 104 141 164 TST33ID: .ASCIZ 'Data Parity'
3464                                     .EVEN
3465                                     ;
3466                                     ;
3467 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3468 ;WRITE SUBSYSTEM MEMORY COMMAND
3469                                     ;
3470 ;-
3471
3472 055712 T33REST:
3473 055712 SAVREG
3474 055716 012701 054600 MOV #T33PACKET,R1 ;SAVE THE REGISTERS
3475 055722 012721 100004 MOV #100004,(R1)+ ;START OF THE PACKET
3476 055726 012721 054610 MOV #T33DATA,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK.
3477 055732 005021 CLR (R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
3478 055734 012721 000012 MOV #10.,(R1)+ ;EXTENDED ADDRESS
3479 055740 012721 054622 MOV #T33BFR,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
3480 055744 005021 CLR (R1)+ ;ADDRESS OF MESSAGE BUFFER
3481 055746 012721 000024 MOV #20.,(R1)+ ;LENGTH OF MESSAGE BUFFER
3482 055752 005021 CLR (R1)+
3483 055754 012711 000000 MOV #0,(R1) ;SELECT DRIVE ZERO

```

TEST 5: DATA PARITY TEST

```

3484 055760 012702 000030      MOV      #24.,R2          ;NUMBER OF LOCATIONS TO BE CLEARED
3485 055764 012762 177777 054622 64$:  MOV      #177777,T33BFR(R2) ;ALL ONES TO MESSAGE BUFFER
3486 055772 005742              TST      -(R2)           ;NEXT LOCATION
3487 055774 022702 000000      CMP      #0,R2          ;AT END OF LOOP YET
3488 056000 001371              BNE      64$            ;KEEP GOING UNTIL DONE
3489 056002 000207              RTS      PC              ;RETURN
3490
3491 056004              T33RT2:
3492 056004              SAVREG                ;SAVE THE REGISTERS
3493 056010 012701 054710      MOV      #T33PK2,R1     ;START OF THE PACKET
3494 056014 012721 100006      MOV      #100006,(R1)+  ;WRITE SUBSYSTEM MEM. WITH ACK.
3495 056020 012721 054730      MOV      #T33BF2,(R1)+ ;ADDRESS OF DATA BLOCK
3496 056024 005021              CLR      (R1)+          ;EXTENDED ADDRESS
3497 056026 012721 000006      MOV      #6.,(R1)+     ;SIZE OF DATA BLOCK IN BYTES
3498 056032 005021              CLR      (R1)+
3499 056034 012701 054730      MOV      #T33BF2,R1     ;POINT TO DATA SEL AREA
3500 056040 005021              CLR      (R1)+
3501 056042 005011              CLR      (R1)
3502 056044 000207              RTS      PC              ;RETURN
3503 056046              T33RT3:
3504 056046              SAVREG                ;SAVE REGISTERS
3505 056052 012701 054720      MOV      #T33PK3,R1     ;SET UP POINTER ADDRESS
3506 056056 005021              CLR      (R1)+          ;COMMAND SPACE
3507 056060 005021              CLR      (R1)+          ;ADDRESS OF DATA BLOCK
3508 056062 005021              CLR      (R1)+          ;EXTENDED ADDRESS
3509 056064 005011              CLR      (R1)           ;SIZE OF DATA TRANSFER BLOCK
3510 056066 000207              RTS      PC              ;RETURN
3511 056070              ENDTST

```

L10057: TRAP C\$ETST

3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541

056070 104401
056072
056072 012737 006354 002172
056100 012700 063237
056104 004737 016570
056110 012737 000005 002210
056116 005037 060722

```

.SBTTL TEST 6: OPERATIONS AT EOT
; *
; THIS TEST VERIFIES PROPER OPERATION OF THE WRITE DATA RETRY
; COMMAND (SPACE REVERSE, ERASE, WRITE DATA)
;
; THE TEST CONSISTS OF THE FOLLOWING 1 SUBTEST
;
; -
BGNTST
; *
; TEST 6, SUBTEST 1
;
; THIS TEST VERIFIES THAT THE EOT STATUS IS HANDLED PROPERLY BY
; THE VARIOUS TAPE MOTION COMMANDS. THE FOLLOWING TEST SEQUENCE
; IS PERFORMED:

```

```

MOV      #EPRT1,EPRTSW ;PRIMARY ERROR MESSAGE
MOV      #TST34ID,R0   ;ASCII MESSAGE TO IDENTIFY TEST
JSR      PC,TSTSETUP   ;DO INITIAL TEST SETUP
MOV      #5,LOOPCNT    ;PERFORM 5 ITERATIONS
CLR      T34CNT        ;CLEAR TAPE RECORD COUNTER

```


TEST 6: OPERATIONS AT EOT

```

3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621 056122                BGNSUB                                ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>
        056122                T6.1:                                TRAP      C$BSUB
        056122 104402                JSR      PC,T34REST                ;SET COMMAND PACKET
3622 056124 004737 063262            JSR      PC,T34RT3                ;RESTORE PACKET
3623 056130 004737 063416            JSR      PC,T34RT2                ;SET UP OTHER COMMAND PACKET
3624 056134 004737 063354                MOV     #65000.,T34DLY            ;SET UP COUNTER
3625 056140 012737 176750 060724    10$:  JSR      PC,SOFINIT            ;DO INITIALIZE ON CONTROLLER
3626 056146 004737 016054                BCS    20$                        ;BR IF INIT WAS OK
3627 056152 103433                DELAY   250                        ;DELAY A WHILE
3628 056154 012727 000250                MOV     #250.(PC)+               MOV     #250.(PC)+
        056160 000000                .WORD 0                           .WORD 0
        056162 013727 002116                MOV     L$DLY.(PC)+             MOV     L$DLY.(PC)+
        056166 000000                .WORD 0                           .WORD 0
        056170 005367 177772                DEC     -6(PC)                    DEC     -6(PC)
        056174 001375                BNE     -4                          BNE     -4
        056176 005367 177756                DEC     -22(PC)                   DEC     -22(PC)
        056202 001367                BNE     -20                          BNE     -20
3629 056204 016501 000002            MOV     TSSR(R5),R1               ;GET TSSR STATUS
3630 056210 032701 000200            BIT     #SSR,R1                   ;CHECK FOR SSR SET
3631 056214 001012                BNE    20$                          ;BR, WHEN SSR IS SET
3632 056216 005337 060724            DEC     T34DLY                     ;BUMP COUNTER DOWN
3633 056222 001351                BNE    10$                          ;BR, IF MORE DELAY REQUIRED
3634 056224 005237 002214                INC     FATFLG                     ;ERROR COUNT
3638 056230 010001                MOV     R0,R1                       ;CONTENTS OF TSSR REGISTER
3639 056232                ERRDF  ERRNO,SFIERR,SFIMSG        ;FATAL ERROR TSSR WAS NOT OK
        056232 104455                TRAP   C$ERDF                      TRAP   C$ERDF
        056234 001131                .WORD 601                           .WORD 601
        056236 003646                .WORD SFIERR                       .WORD SFIERR
        056240 012114                .WORD SFIMSG                        .WORD SFIMSG
3640 056242                20$: CKLOOP                            ;LOOP IF SELECTED
        056242 104406                TRAP   C$CLP1                      TRAP   C$CLP1
3641 056244 013737 002174 060610    MOV     UNITN,T34DSW              ;SET UP DRIVE NUMBER
3642 056252 052737 000040 060610    BIS     #BITS,T34DSW              ;TURN ON HIGH SPEED TO SAVE TIME
3643 056260 012704 060570                MOV     #T34PACKET,R4            ;SUBROUTINE NEEDS PACKET ADDRESS

```

TEST 6: OPERATIONS AT EOT

```

3644 056264 004737 010742      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
3645 056270 103407      BCS      30$           ;BR, IF COMMAND ISSUED OK
3646 056272 005237 002214      INC      FATFLG        ;ERROR COUNT
3650 056276 010001      MOV      R0,R1         ;SAVE CONTENTS OF TSSR
3651 056300      ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP      C$ERHRD
                                .WORD    602
                                .WORD    WRTMSG
                                .WORD    SFIMSG
3652 056310      30$:   CKLOOP        ;LOOP IF SELECTED
                                TRAP      C$CLP1
3653 056312 004737 011074      JSR      PC,REWIND     ;REWIND CALL
3654 056316 103411      BCS      35$           ;BR, IF TSSR IS OK (GOOD)
3655 056320 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR
3656 056324 010004      MOV      R0,R4         ;SET UP PACKET
3657 056326 005237 002214      INC      FATFLG        ;ERROR COUNT
3661 056332      ERRHRD  ERRNO,T34RWN,PKTSSR ;TSSR IS INCORRECT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    603
                                .WORD    T34RWN
                                .WORD    PKTSSR
3662 056342      35$:   CKLOOP        ;LOOP IF SELECTED
                                TRAP      C$CLP1
3663 056344 012737 140005 060710      MOV      #140005,T34PK3 ;WRITE DATA, ACK, CVC=1
3664 056352 012703 176750      MOV      #65000.,R3    ;SET MAX NUMBER OF WRITES
3665 056356 013737 003116 060712      MOV      FREE,T34WB    ;SET UP WRITE BUFFER ADDRESS
3666 056364 012737 006654 060716      MOV      #3500.,T34SZ  ;SET UP BUFFER SIZE (4K BYTES)
3667 056372 012704 060710      MOV      #T34PK3,R4   ;R4 = POINTER TO PACKET
3668 056376 010465 000000      40$:   MOV      R4,T34RWN ;ISSUE COMMAND
3669 056402 004737 016330      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
3670 056406 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
3671 056412 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
3672 056416 020102      CMP      R1,R2        ;ARE THEY EQUAL
3673 056420 001010      BNE      50$          ;BR, IT MIGHT BE END OF TAPE
3674 056422 005303      DEC      R3           ;DEC RECORD COUNTER
3675 056424 001364      BNE      40$          ;BR, IF MORE TO GO
3676 056426 005237 002214      INC      FATFLG        ;ERROR COUNT
3680 056432      ERRDF  ERRNO,T34ET,PKTSSR ;EOT NOT FOUND (USE SHORTER TAPE?)
                                TRAP      C$ERDF
                                .WORD    604
                                .WORD    T34ET
                                .WORD    PKTSSR
3681 056442 032701 000004      50$:   BIT      #BIT2,R1 ;CHECK FOR TAPE STATUS ALERT
3682 056446 001001      BNE      60$          ;BR, IF SET
3683 056450 000752      BR       40$          ;KEEP GOING
3684 056452 013701 060620      60$:   MOV      T34BFR+6,R1 ;PICK UP XSTO
3685 056456 010102      MOV      R1,R2        ;SET UP EXPECTED
3686 056460 052702 000001      BIS      #BIT0,R2     ;SET THE EOT BIT ON IN EXPECTED
3687 056464 020102      CMP      R1,R2        ;WAS THE BIT ON
3688 056466 001402      BEQ      80$          ;BR, IF EOT WAS FOUND
3689 056470 000137 056376      JMP      40$          ;KEEP LOOKING
3690 056474      80$:   CKLOOP        ;LOOP IF SELECTED
                                TRAP      C$CLP1
3691 056476 012737 140005 060710      MOV      #140005,T34PK3 ;WRITE DATA, ACK, CVC=1
3692 056504 013737 003116 060712      MOV      FREE,T34WB    ;SET UP WRITE BUFFER ADDRESS
3693 056512 012737 006654 060716      MOV      #3500.,T34SZ  ;SET UP BUFFER SIZE (4K BYTES)
3694 056520 012704 060710      MOV      #T34PK3,R4   ;R4 = POINTER TO PACKET

```

TEST 6: OPERATIONS AT EOT

```

3695 056524 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
3696 056530 004737 016330      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
3697 056534 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
3698 056540 012702 100204      MOV      #SC!SSR!BIT2,R2 ;SET UP EXPECTED
3699 056544 020102                CMP      R1,R2           ;ARE THEY EQUAL
3700 056546 001406                BEQ      90$             ;BR, IF THEY ARE OK
3701 056550 005237 002214      INC      FATFLG          ;ERROR COUNT
3705 056554                ERRHRD  ERRNO,T34ET2,PKTSSR ;WRITE TAPE AT EOT FAILED TO SET TSA
                                TRAP      C$ERHRD
                                .WORD    605
                                .WORD    T34ET2
                                .WORD    PKTSSR
                                056554 104456
                                056556 001135
                                056560 061417
                                056562 012126
3706 056564                90$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                056564 104406
3707 056566 013701 060620      MOV      T34BFR+6,R1     ;PICK UP XSTO
3708 056572 010102                MOV      R1,R2           ;SET UP EXPECTED
3709 056574 052702 000001      BIS      #BIT0,R2        ;SET THE EOT BIT ON IN EXPECTED
3710 056600 020102                CMP      R1,R2           ;WAS THE BIT ON
3711 056602 001406                BEQ      100$            ;BR, IF EOT WAS FOUND
3712 056604 005237 002214      INC      FATFLG          ;ERROR COUNT
3716 056610                ERRHRD  ERRNO,T34ETN,EXPREC ;EOT BIT (XSTO) NOT SET
                                TRAP      C$ERHRD
                                .WORD    606
                                .WORD    T34ETN
                                .WORD    EXPREC
                                056610 104456
                                056612 001136
                                056614 061501
                                056616 015554
3717 056620                100$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                056620 104406
3718 056622 012737 140011 060710      MOV      #140011,T34PK3  ;WRITE TAPE MARK, ACK, CVC=1 COMMAND
3719 056630 012704 060710      MOV      #T34PK3,R4      ;R4 = POINTER TO PACKET
3720 056634 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
3721 056640 004737 016330      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
3722 056644 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
3723 056650 012702 100204      MOV      #SC!SSR!BIT2,R2 ;SET UP EXPECTED
3724 056654 020102                CMP      R1,R2           ;ARE THEY EQUAL
3725 056656 001406                BEQ      110$            ;BR, IF STATUS IS GOOD (OK)
3726 056660 005237 002214      INC      FATFLG          ;ERROR COUNT
3730 056664                ERRHRD  ERRNO,T34WTM,PKTSSR ;EOT NOT FOUND (USE SHORTER TAPE?)
                                TRAP      C$ERHRD
                                .WORD    607
                                .WORD    T34WTM
                                .WORD    PKTSSR
                                056664 104456
                                056666 001137
                                056670 061330
                                056672 012126
3731 056674                110$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                056674 104406
3732 056676 013701 060620      MOV      T34BFR+6,R1     ;PICK UP XSTO
3733 056702 010102                MOV      R1,R2           ;SET UP EXPECTED
3734 056704 052702 000001      BIS      #BIT0,R2        ;SET THE EOT BIT ON IN EXPECTED
3735 056710 020102                CMP      R1,R2           ;WAS THE BIT ON
3736 056712 001406                BEQ      120$            ;BR, IF EOT WAS FOUND
3737 056714 005237 002214      INC      FATFLG          ;ERROR COUNT
3741 056720                ERRHRD  ERRNO,T34ETO,EXPREC ;EOT BIT (XSTO) NOT SET
                                TRAP      C$ERHRD
                                .WORD    608
                                .WORD    T34ETO
                                .WORD    EXPREC
                                056720 104456
                                056722 001140
                                056724 061032
                                056726 015554
3742 056730                120$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                056730 104406
3743 056732 012737 141410 060710      MOV      #141410,T34PK3  ;SKIP TAPE MARK REVERSE ACK,CVC=1 COMMAND

```


TEST 6: OPERATIONS AT EOT

```

3744 056740 012737 000001 060712      MOV      #1,T34WB      ;SET NUMBER (1) OF TMS TO SKIP
3745 056746 012704 060710      MOV      #T34PK3,R4   ;R4 = POINTER TO PACKET
3746 056752 010465 000000      MOV      R4,TSDB(R5)  ;ISSUE COMMAND
3747 056756 004737 016330      JSR      PC,WAITF     ;WAIT FOR SSR TO SET
3748 056762 016501 000002      MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
3749 056766 012702 000200      MOV      #SSR,R2     ;SET UP EXPECTED
3750 056772 020102      CMP      R1,R2       ;ARE THEY EQUAL
3751 056774 001406      BEQ      130$        ;BR, IF STATUS IS GOOD (OK)
3752 056776 005237 002214      INC      FATFLG      ;ERROR COUNT
3756 057002      ERRHRD  ERRNO,T34STM,PKTSSR ;SKIP TAPE MARK REV. DIDN'T SET TSA
                                TRAP      C$ERHRD
                                .WORD    609
                                .WORD    T34STM
                                .WORD    PKTSSR
3757 057012      130$:  CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    609
3758 057014 013701 060620      MOV      T34BFR+6,R1 ;PICK UP XST0
3759 057020 010102      MOV      R1,R2       ;SET UP EXPECTED
3760 057022 052702 000001      BIS      #BIT0,R2    ;SET THE EOT BIT ON IN EXPECTED
3761 057026 020102      CMP      R1,R2       ;WAS THE BIT ON
3762 057030 001406      BEQ      140$        ;BR, IF EOT WAS FOUND
3763 057032 005237 002214      INC      FATFLG      ;ERROR COUNT
3767 057036      ERRHRD  ERRNO,T34ETN,EXPREC ;EOT BIT (XST0) NOT SET
                                TRAP      C$ERHRD
                                .WORD    610
                                .WORD    T34ETN
                                .WORD    EXPREC
3768 057046      140$:  CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    610
3769 057050 013701 060620      MOV      T34BFR+6,R1 ;PICK UP XST0
3770 057054 010102      MOV      R1,R2       ;SET UP EXPECTED
3771 057056 052702 100000      BIS      #BIT15,R2   ;SET THE TMK BIT ON IN EXPECTED
3772 057062 020102      CMP      R1,R2       ;WAS THE BIT ON
3773 057064 001406      BEQ      150$        ;BR, IF TMK WAS FOUND
3774 057066 005237 002214      INC      FATFLG      ;ERROR COUNT
3778 057072      ERRHRD  ERRNO,T34TMK,EXPREC ;EOT BIT (XST0) NOT SET
                                TRAP      C$ERHRD
                                .WORD    611
                                .WORD    T34TMK
                                .WORD    EXPREC
3779 057102      150$:  CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    611
3780 057104 012737 140410 060710      MOV      #140410,T34PK3 ;SPACE RECORDS REVERSE, ACK, CVC=1 CMD
3781 057112 012737 000001 060712      MOV      #1,T34WB     ;SPACE ONE RECORD REVERSE
3782 057120 012704 060710      MOV      #T34PK3,R4   ;R4 = POINTER TO PACKET
3783 057124 010465 000000      MOV      R4,TSDB(R5)  ;ISSUE COMMAND
3784 057130 004737 016330      JSR      PC,WAITF     ;WAIT FOR SSR TO SET
3785 057134 016501 000002      MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
3786 057140 012702 100204      MOV      #SC!SSR!BIT2,R2 ;SET UP EXPECTED
3787 057144 020102      CMP      R1,R2       ;ARE THEY EQUAL
3788 057146 001006      BNE     160$        ;BR, IT MIGHT BE END OF TAPE
3789 057150 005237 002214      INC      FATFLG      ;ERROR COUNT
3793 057154      ERRHRD  ERRNO,T34POS,PKTSSR ;EOT NOT FOUND (USE SHORTER TAPE?)
                                TRAP      C$ERHRD
                                .WORD    612
                                .WORD    T34POS
                                .WORD    PKTSSR
3793 057154 104456
3793 057156 001144
3793 057160 060744
3793 057162 012126

```


TEST 6: OPERATIONS AT EOT

3843	057410	016501	000002		MOV	TSSR(R5),R1				;GET TSSR CONTENTS		
3844	057414	012702	000200		MOV	#SSR,R2				;SET UP EXPECTED		
3845	057420	020102			CMP	R1,R2				;ARE THEY EQUAL		
3846	057422	001406			BEQ	190\$;BR, IT MIGHT BE END OF TAPE		
3847	057424	005237	002214		INC	FATFLG				;ERROR COUNT		
3851	057430				ERRHRD	ERRNO,T34POS,PKTSSR				;EOT NOT FOUND (USE SHORTER TAPE?)		
	057430	104456								TRAP	C\$ERHRD	
	057432	001150								.WORD	616	
	057434	060744								.WORD	T34POS	
	057436	012126								.WORD	PKTSSR	
3852	057440			190\$:	CKLOOP					;LOOP IF SELECTED		
	057440	104406								TRAP	C\$CLP1	
3853	057442	013701	060620		MOV	T34BFR+6,R1				;PICK UP XSTO		
3854	057446	010102			MOV	R1,R2				;SET UP EXPECTED		
3855	057450	052702	000001		BIS	#BIT0,R2				;SET THE EOT BIT ON IN EXPECTED		
3856	057454	020102			CMP	R1,R2				;WAS THE BIT ON		
3857	057456	001406			BEQ	200\$;BR, IF EOT WAS FOUND		
3858	057460	005237	002214		INC	FATFLG				;ERROR COUNT		
3862	057464				ERRHRD	ERRNO,T34ETS,EXPREC				;EOT BIT (XSTO) NOT SET		
	057464	104456								TRAP	C\$ERHRD	
	057466	001151								.WORD	617	
	057470	061560								.WORD	T34ETS	
	057472	015554								.WORD	EXPREC	
3863	057474			200\$:	CKLOOP					;LOOP IF SELECTED		
	057474	104406								TRAP	C\$CLP1	
3864	057476	012737	140401	060710	MOV	#140401,T34PK3				;READ REVERSE, ACK, CVC=1		
3865	057504	013737	003116	060712	MOV	FREE,T34RB				;SET UP WRITE BUFFER ADDRESS		
3866	057512	012704	060710		MOV	#T34PK3,R4				;R4 = POINTER TO PACKET		
3867	057516	010465	000000		MOV	R4,TSDB(R5)				;ISSUE COMMAND		
3868	057522	004737	016330		JSR	PC,WAITF				;WAIT FOR SSR TO SET		
3869	057526	016501	000002		MOV	TSSR(R5),R1				;GET TSSR CONTENTS		
3870	057532	012702	000200		MOV	#SSR,R2				;SET UP EXPECTED		
3871	057536	020102			CMP	R1,R2				;ARE THEY EQUAL		
3872	057540	001406			BEQ	205\$;BR, ONLY SSR IS SET		
3873	057542	005237	002214		INC	FATFLG				;ERROR COUNT		
3877	057546				ERRHRD	ERRNO,T34RRE,PKTSSR				;EOT NOT FOUND (USE SHORTER TAPE?)		
	057546	104456								TRAP	C\$ERHRD	
	057550	001152								.WORD	618	
	057552	061116								.WORD	T34RRE	
	057554	012126								.WORD	PKTSSR	
3878	057556			205\$:	CKLOOP					;LOOP IF SELECTED		
	057556	104406								TRAP	C\$CLP1	
3879	057560	012737	140401	060710	MOV	#140401,T34PK3				;READ REVERSE, ACK, CVC=1		
3880	057566	013737	003116	060712	MOV	FREE,T34RB				;SET UP WRITE BUFFER ADDRESS		
3881	057574	012704	060710		MOV	#T34PK3,R4				;R4 = POINTER TO PACKET		
3882	057600	010465	000000		MOV	R4,TSDB(R5)				;ISSUE COMMAND		
3883	057604	004737	016330		JSR	PC,WAITF				;WAIT FOR SSR TO SET		
3884	057610	016501	000002		MOV	TSSR(R5),R1				;GET TSSR CONTENTS		
3885	057614	012702	000200		MOV	#SSR,R2				;SET UP EXPECTED		
3886	057620	020102			CMP	R1,R2				;ARE THEY EQUAL		
3887	057622	001406			BEQ	210\$;BR, IT MIGHT BE END OF TAPE		
3888	057624	005237	002214		INC	FATFLG				;ERROR COUNT		
3892	057630				ERRHRD	ERRNO,T34RRE,PKTSSR				;EOT NOT FOUND (USE SHORTER TAPE?)		
	057630	104456								TRAP	C\$ERHRD	
	057632	001153								.WORD	619	
	057634	061116								.WORD	T34RRE	
	057636	012126								.WORD	PKTSSR	

TEST 6: OPERATIONS AT EOT

3893	057640				2108:	CKLOOP		:LOOP IF SELECTED		
	057640	104406							TRAP	C\$CLP1
3894	057642	012737	140001	060710		MOV	#140001,T34PK3	:READ DATA, ACK, CVC=1		
3895	057650	013737	003116	060712		MOV	FREE,T34RB	:SET UP WRITE BUFFER ADDRESS		
3896	057656	012737	006654	060716		MOV	#3500.,T34SZ	:SET UP BUFFER SIZE (4K BYTES)		
3897	057664	012704	060710			MOV	#T34PK3,R4	:R4 = POINTER TO PACKET		
3898	057670	010465	000000			MOV	R4,TSDB(R5)	:ISSUE COMMAND		
3899	057674	004737	016330			JSR	PC,WAITF	:WAIT FOR SSR TO SET		
3900	057700	016501	000002			MOV	TSSR(R5),R1	:GET TSSR CONTENTS		
3901	057704	012702	000200			MOV	#SSR,R2	:SET UP EXPECTED		
3902	057710	020102				CMP	R1,R2	:ARE THEY EQUAL		
3903	057712	001406				BEQ	2308	:BR, IT MIGHT BE END OF TAPE		
3904	057714	005237	002214			INC	FATFLG	:ERROR COUNT		
3908	057720					ERRHRD	ERRNO,T34RRE,PKTSSR	:EOT NOT FOUND (USE SHORTER TAPE?)		
	057720	104456							TRAP	C\$ERHRD
	057722	001154							.WORD	620
	057724	061116							.WORD	T34RRE
	057726	012126							.WORD	PKTSSR
3909	057730				2308:	CKLOOP		:LOOP IF SELECTED		
	057730	104406							TRAP	C\$CLP1
3910	057732	012737	140001	060710		MOV	#140001,T34PK3	:READ DATA, ACK, CVC=1		
3911	057740	013737	003116	060712		MOV	FREE,T34RB	:SET UP WRITE BUFFER ADDRESS		
3912	057746	012737	006654	060716		MOV	#3500.,T34SZ	:SET UP BUFFER SIZE (4K BYTES)		
3913	057754	012704	060710			MOV	#T34PK3,R4	:R4 = POINTER TO PACKET		
3914	057760	010465	000000			MOV	R4,TSDB(R5)	:ISSUE COMMAND		
3915	057764	004737	016330			JSR	PC,WAITF	:WAIT FOR SSR TO SET		
3916	057770	016501	000002			MOV	TSSR(R5),R1	:GET TSSR CONTENTS		
3917	057774	012702	000200			MOV	#SSR,R2	:SET UP EXPECTED		
3918	060000	020102				CMP	R1,R2	:ARE THEY EQUAL		
3919	060002	001406				BEQ	2358	:BR, IT MIGHT BE END OF TAPE		
3920	060004	005237	002214			INC	FATFLG	:ERROR COUNT		
3924	060010					ERRHRD	ERRNO,T34RRE,PKTSSR	:EOT NOT FOUND (USE SHORTER TAPE?)		
	060010	104456							TRAP	C\$ERHRD
	060012	001155							.WORD	621
	060014	061116							.WORD	T34RRE
	060016	012126							.WORD	PKTSSR
3925	060020				2358:	CKLOOP		:LOOP IF SELECTED		
	060020	104406							TRAP	C\$CLP1
3926	060022	013701	060620			MOV	T34BFR+6,R1	:PICK UP XSTO		
3927	060026	010102				MOV	R1,R2	:SET UP EXPECTED		
3928	060030	052702	000001			BIS	#BIT0,R2	:SET THE EOT BIT ON IN EXPECTED		
3929	060034	020102				CMP	R1,R2	:WAS THE BIT ON		
3930	060036	001406				BEQ	2408	:BR, IF EOT WAS FOUND		
3931	060040	005237	002214			INC	FATFLG	:ERROR COUNT		
3935	060044					ERRHRD	ERRNO,T34ETZ,EXPREC	:EOT BIT (XSTO) NOT SET		
	060044	104456							TRAP	C\$ERHRD
	060046	001156							.WORD	622
	060050	061652							.WORD	T34ETZ
	060052	015554							.WORD	EXPREC
3936	060054				2408:	CKLOOP		:LOOP IF SELECTED		
	060054	104406							TRAP	C\$CLP1
3937	060056	012737	140410	060710		MOV	#140410,T34PK3	:SPACE RECORDS REVERSE, ACK, CVC=1 CMD.		
3938	060064	012737	000005	060712		MOV	#5,T34RB	:NUMBER OF RECORDS TO SPACE		
3939	060072	012704	060710			MOV	#T34PK3,R4	:R4 = POINTER TO PACKET		
3940	060076	010465	000000			MOV	R4,TSDB(R5)	:ISSUE COMMAND		
3941	060102	004737	016330			JSR	PC,WAITF	:WAIT FOR SSR TO SET		
3942	060106	016501	000002			MOV	TSSR(R5),R1	:GET TSSR CONTENTS		

TEST 6: OPERATIONS AT EOT

```

3943 060112 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
3944 060116 020102             CMP      R1,R2           ;ARE THEY EQUAL
3945 060120 001406             BEQ     250$             ;BR, IT MIGHT BE END OF TAPE
3946 060122 005237 002214      INC     FATFLG          ;ERROR COUNT
3950 060126             ERRHRD  ERRNO,T34POS,PKTSSR ;POSITION COMMAND DIDN'T WORK
                                TRAP      C$ERHRD
                                .WORD    623
                                .WORD    T34POS
                                .WORD    PKTSSR
                                060126 104456
                                060130 001157
                                060132 060744
                                060134 012126
3951 060136             250$: CKLCOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                060136 104406
3952 060140 013701 060620      MOV     T34BFR-6,R1     ;PICK UP XSTO
3953 060144 010102             MOV     R1,R2           ;SET UP EXPECTED
3954 060146 042702 000001      BIC    #BIT0,R2        ;CLEAR THE EOT BIT ON IN EXPECTED
3955 060152 020102             CMP     R1,R2           ;WAS THE BIT ON
3956 060154 001406             BEQ    260$             ;BR, IF EOT WAS FOUND
3957 060156 005237 002214      INC    FATFLG          ;ERROR COUNT
3961 060162             ERRHRD  ERRNO,T34ETC,EXPREC ;EOT BIT (XSTO) NOT CLEAR
                                TRAP      C$ERHRD
                                .WORD    624
                                .WORD    T34ETC
                                .WORD    EXPREC
                                060162 104456
                                060164 001160
                                060166 061207
                                060170 015554
3962 060172             260$: CKLOOP        ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                060172 104406
3963 060174 012737 140010 060710  MOV     #140010,T34PK3   ;SPACE RECORDS FORWARD, ACK, CVC=1 CMD.
3964 060202 012737 000005 060712  MOV     #5,T34RB        ;NUMBER OF RECORDS TO SPACE
3965 060210 012704 060710      MOV     #T34PK3,R4     ;R4 = POINTER TO PACKET
3966 060214 010465 000000      MOV     R4,TSDB(R5)    ;ISSUE COMMAND
3967 060220 004737 016330      JSR    PC,WAITF       ;WAIT FOR SSR TO SET
3968 060224 016501 000002      MOV     TSSR(R5),R1    ;GET TSSR CONTENTS
3969 060230 012702 000200      MOV     #SSR,R2       ;SET UP EXPECTED
3970 060234 020102             CMP     R1,R2           ;ARE THEY EQUAL
3971 060236 001406             BEQ    270$             ;BR, IT MIGHT BE END OF TAPE
3972 060240 005237 002214      INC    FATFLG          ;ERROR COUNT
3976 060244             ERRHRD  ERRNO,T34ET,PKTSSR ;TSSR NOT CORRECT
                                TRAP      C$ERHRD
                                .WORD    625
                                .WORD    T34ET
                                .WORD    PKTSSR
                                060244 104456
                                060246 001161
                                060250 062146
                                060252 012126
3977 060254             270$: CKLOOP        ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                060254 104406
3978 060256 013701 060620      MOV     T34BFR-6,R1    ;PICK UP XSTO
3979 060262 010102             MOV     R1,R2           ;SET UP EXPECTED
3980 060264 052702 000001      BIS    #BIT0,R2        ;SET THE EOT BIT ON IN EXPECTED
3981 060270 020102             CMP     R1,R2           ;WAS THE BIT ON
3982 060272 001400             BEQ    280$             ;BR, IF EOT WAS FOUND
3983 060274             280$: CKLOOP        ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                060274 104406
3984 060276 012737 141410 060710  MOV     #141410,T34PK3  ;SKIP FILE MARKS REVERSE,ACK,CVC=1 COMMAND
3985 060304 012737 000003 060712  MOV     #3,T34RB        ;NUMBER OF FILE MARKS
3986 060312 012704 060710      MOV     #T34PK3,R4     ;R4 = POINTER TO PACKET
3987 060316 010465 000000      MOV     R4,TSDB(R5)    ;ISSUE COMMAND
3988 060322 012737 176750 060724  MOV     #65000,T34DLY   ;SET UP DELAY COUNTER
3989 060330 004737 016330      JSR    PC,WAITF       ;WAIT FOR SSR TO SET
3990 060334 016501 000002      MOV     TSSR(R5),R1    ;GET TSSR CONTENTS
3991 060340 032701 000200      BIT    #SSR,R1         ;CHECK FOR SSR SET
3992 060344 001017             BNE    286$             ;BR, WHEN SSR IS SET
                                060276 012737 141410 060710
                                060276 012737 000003 060712
                                060312 012704 060710
                                060316 010465 000000
                                060322 012737 176750 060724
                                060330 004737 016330
                                060334 016501 000002
                                060340 032701 000200
                                060344 001017

```

TEST 6: OPERATIONS AT EOT

```

3993 060346        DELAY    250                ;WAIT ABOUT .25 SECONDS
      060346 012727 000250                    MOV    #250,(PC)
      060352 000000                                .WORD 0
      060354 013727 002116                    MOV    L$DL,(PC)
      060360 000000                                .WORD 0
      060362 005367 177772                    DEC    -6(PC)
      060366 001375                                BNE    -4
      060370 005367 177756                    DEC    -22(PC)
      060374 001367                                BNE    -20
3994 060376 005337 060724          DEC    T34DL    ;BUMP COUNTER
3995 060402 001352                    BNE    285$    ;BR, IF MORE TO COUNT
3996 060404 012702 000200          286$: MOV    #SSR,R2    ;SET UP EXPECTED
3997 060410 020102                    CMP    R1,R2    ;ARE THEY EQUAL
3998 060412 001007                    BNE    290$    ;BR, IT MIGHT BE END OF TAPE
3999 060414 005303                    DEC    R3        ;DEC RECORD COUNTER
4000 060416 005237 002214          INC    FATFLG    ;ERROR COUNT
4004 060422          ERRHRD  ERRNO,T34ET,PKTSSR    ;EOT NOT FOUND (USE SHORTER TAPE?)
      060422 104456                                TRAP   C$ERHRD
      060424 001162                                .WORD 626
      060426 062146                                .WORD T34ET
      060430 012126                                .WORD PKTSSR
4005 060432 032701 000004          290$: BIT    #BIT2,R1    ;CHECK FOR TAPE STATUS ALERT
4006 060436 013701 060620          MOV    T34BFR+6,R1    ;PICK UP XSTO
4007 060442 010102                    MOV    R1,R2    ;SET UP EXPECTED
4008 060444 042702 000001          BIC    #BIT0,R2    ;CLEAR THE EOT BIT IN EXPECTED
4009 060450 020102                    CMP    R1,R2    ;WAS THE BIT ON
4010 060452 001406                    BEQ    300$    ;BR, IF EOT WAS FOUND
4011 060454 005237 002214          INC    FATFLG    ;ERROR COUNT
4015 060460          ERRHRD  ERRNO,T34ETC,EXPREC    ;EOT BIT (XSTO) NOT CLEAR
      060460 104456                                TRAP   C$ERHRD
      060462 001163                                .WORD 627
      060464 061207                                .WORD T34ETC
      060466 015554                                .WORD EXPREC
4016 060470          300$: CKLOOP                ;LOOP IF SELECTED
      060470 104406                                TRAP   C$CLP1
4017 060472 013701 060620          MOV    T34BFR+6,R1    ;PICK UP XSTO
4018 060476 010102                    MOV    R1,R2    ;SET UP EXPECTED
4019 060500 052702 000002          BIS    #BIT1,R2    ;SET THE BOT BIT ON IN EXPECTED
4020 060504 020102                    CMP    R1,R2    ;WAS THE BIT ON
4021 060506 001406                    BEQ    320$    ;BR, IF BOT WAS FOUND
4022 060510 005237 002214          INC    FATFLG    ;ERROR COUNT
4026 060514          ERRHRD  ERRNO,T34BOT,EXPREC    ;EOT BIT (XSTO) NOT CLEAR
      060514 104456                                TRAP   C$ERHRD
      060516 001164                                .WORD 628
      060520 061264                                .WORD T34BOT
      060522 015554                                .WORD EXPREC
4027 060524          320$: CKLOOP                ;LOOP IF SELECTED
      060524 104406                                TRAP   C$CLP1
4028 060526          600$:
4029 060526          ENDSUB                ;>>>>>>>>>> END SUBTEST >>>>>>>>>>
      060526 104403                                L10062:
4030 060530 023727 002214 000017    CMP    FATFLG,#15.    ;IS ERROR COUNT AT 25
4031 060536 103402                    BLO    999$    ;BR, IF LESS THAN 25
4032 060540 004737 017262          JSR    PC,CKDROP    ;TRY TO DROP THE UNIT
4033 060544          999$:
4034 060544 004737 016536          JSR    PC,TSTLOOP    ;DO WE NEED TO ITERATE TEST

```

TEST 6: OPERATIONS AT EOT

```

4035 060550 103002      BCC 163$
4036 060552 000137 056122 JMP T34LOOP
4037 060556      163$: EXIT TST
      060556 104432
      060560 002662
4038
4039
4040
4042      060570
4044 060570      .=<..+10>E177770
4045 060570 100004      T34PACKET: .WORD 100004
4046 060572 060600      .WORD T34DATA
4047 060574 000000      .WORD 0
4048 060576 000010      .WORD 8.
4049 060600      T34DATA:
4050 060600 060612      .WORD T34BFR
4051 060602 00C000      .WORD 0
4052 060604 000012      .WORD 10.
4053 060606 000000      .WORD 0
4054 060610 000000      T34DSW: .WORD 0
4055 060612      T34BFR: .BLKW 25.
4056
4057      ;
4058      ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
4060      ;
4062 060700      .=<..+10>E177770
4063 060700 100006      T34PK2: .WORD 100006
4064 060702 060726      .WORD T34BF2
4065 060704 000000      .WORD 0
4066 060706 000006      .WORD 6.
4067
4071 060710      T34PK3:
4072 060710 100005      .WORD 100005
4073 060712      T34RB:
4074 060712 000000      T34WB: .WORD 0
4075 060714 000000      .WORD 0
4076 060716 000000      T34SZ: .WORD 0
4077      .EVEN
4078      ;
4079 060720 000000      T34RSZ: .WORD 0
4080 060722 000000      T34CNT: .WORD 0
4081 060724 000000      T34DLY: .WORD 0
4082      ;
4083      ;
4084 060726      T34BF2:
4085 060726      010      T34BS0: .BYTE 10
4086 060727      200      T34BS1: .BYTE 200
4087 060730 000000      T34S2: .WORD 0
4088 060732 000000      T34S3: .WORD 0
4089      ;
4090      ;
4091      .EVEN
4092      ;TAPE MOTION PACKET COMMAND VALUES
4093
4094 060734 100005      T34WD: .WORD 100005
4095 060736 100405      T34WR: .WORD 100405
4096 060740 102005      T34CON: .WORD 102005

```

```

;BR, IF NO LOOP REQUIRED
;EXECUTE AGAIN
;ALL DONE THIS TEST
TRAP C$EXIT
.WORD L10061.

;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/READ BUFFER

;SIZE OF BUFFER (EXTENT)

;LARGEST TAPE RECORD IN BYTES
;TAPE RECORD COUNTER
;DELAY COUNTER

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

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

```

TEST 6: OPERATIONS AT EOT

```

4097 060742 177777          .WORD 177777          ;END OF DATA
4098
4099                      ;*
4100                      ;LOCAL TEXT MESSAGES FOR TEST
4101                      ;-
4102 060744      124      123      123  T34POS: .ASCIZ  'TSSR Incorrect After Position (SPACE RECORDS) Command'
4103 061032      127      122      111  T34ETO: .ASCIZ  'WRITE TAPE MARK At EOT Failed To Set EOT Bit (XSTO)'
4104 061116      122      105      101  T34RRE: .ASCIZ  'READ Command At EOT Didn't Give Normal Termination (TSSR)'
4105 061207      125      156      141  T34ETC: .ASCIZ  'Unable To Clear EOT Indication, (XSTO) Bit 0'
4106 061264      122      105      127  T34BOT: .ASCIZ  'REWIND Failed To Set BOT (XSTO) Bit'
4107 061330      127      122      111  T34WTM: .ASCIZ  'WRITE TAPE MARK At EOT Failed To Set Tape Status Alert'
4108 061417      127      122      111  T34ET2: .ASCIZ  'WRITE DATA At EOT Failed To Set Tape Status Alert'
4109 061501      127      122      111  T34ETN: .ASCIZ  'WRITE DATA At EOT Failed To Set EOT Bit (XSTO)'
4110 061560      123      120      101  T34ETS: .ASCIZ  'SPACE RECORDS FORWARD At EOT Failed To Set EOT Bit (XSTO)'
4111 061652      122      105      101  T34ETZ: .ASCIZ  'READ DATA At EOT Failed To Set EOT Bit (XSTO)'
4112 061730      124      123      123  T34STM: .ASCIZ  'TSSR Incorrect After SKIP TAPE MARK REVERSE At EOT'
4113 062013      120      117      123  T34TMK: .ASCIZ  'POSITION Command At EOT Onto Tape Mark Failed To Set TMK (XSTO)'
4114 062113      127      122      111  T34SSR: .ASCIZ  'WRITE Command Not Accepted'
4115 062146      105      117      124  T34ET: .ASCIZ  'EOT Not Found In 65000 3.5K Writes, (Use Shorter Tape)'
4116 062235      127      122      111  T34EOT: .ASCIZ  'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
4117 062313      124      123      123  T34TM: .ASCIZ  'TSSR Not Correct After WRITE Command Reject'
4118 062367      122      145      167  T34RWN: .ASCIZ  'Rewind (POSITION) Command Not Accepted'
4119 062436      122      101      115  T34RNC: .ASCIZ  'RAM Error, Correct Data Pattern Not In Ram'
4120 062511      124      123      123  T34AM3: .ASCIZ  'TSSR Init. Failed After WRITE Command'
4121 062557      104      162      151  T34OFL: .ASCIZ  'Drive 7 Select Failed To Set "OFL" In TSSR'
4122 062632      124      123      123  T34WDD: .ASCIZ  'TSSR Not Correct After WRITE DATA Command, SWB Bit Set'
4123 062721      124      123      123  T34WDC: .ASCIZ  'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
4124 063023      103      126      103  T34VCK: .ASCIZ  'CVC Set, Didn't Reset VCK In Message Buffer'
4125 063076      124      123      102  T34BA: .ASCIZ  'TSBA Not Correct After WRITE DATA Command'
4126 063150      127      122      111  T34WSS: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
4127 063237      117      160      145  T34ID: .ASCIZ  'Operations At EOT'
4128                      .EVEN
4129                      ;*
4130                      ;
4131                      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4132                      ;WRITE SUBSYSTEM MEMORY COMMAND
4133                      ;
4134                      ;-
4135
4136 063262          T34REST:
4137 063262          SAVREG          ;SAVE THE REGISTERS
4138 063266      012701  060570      MOV          @T34PACKET,R1          ;START OF THE PACKET
4139 063272      012721  100004      MOV          @100004,(R1)+        ;WRITE SUBSYSTEM MEM. WITH ACK
4140 063276      012721  060600      MOV          @T34DATA,(R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
4141 063302      005021              CLR          (R1)+                ;EXTENDED ADDRESS
4142 063304      012721  000012      MOV          @10.,(R1)+          ;SIZE OF DATA BLOCK IN BYTES
4143 063310      012721  060612      MOV          @T34BFR,(R1)+      ;ADDRESS OF MESSAGE BUFFER
4144 063314      005021              CLR          (R1)+
4145 063316      012721  000024      MOV          @20.,(R1)+         ;LENGTH OF MESSAGE BUFFER
4146 063322      005021              CLR          (R1)+
4147 063324      012711  000000      MOV          @0,(R1)            ;SELECT DRIVE ZERO
4148 063330      012702  000030      MOV          @24.,R2            ;NUMBER OF LOCATIONS TO BE CLEARED
4149 063334      012762  177777  060612  64$:  MOV          @177777,T34BFR(R2)   ;ALL ONES TO MESSAGE BUFFER
4150 063342      005742              TST          -(R2)               ;BUMP DOWN TO NEXT LOCATION
4151 063344      020227  000000      CMP          R2,@0              ;R2 AT ZERO YET
4152 063350      001371              BNE          64$                 ;KEEP GOING UNTIL DONE
4153 063352      000207              RTS          PC                   ;RETURN

```


TEST 6: OPERATIONS AT EOT

4154
4155 063354
4156 063354
4157 063360 012701 060700
4158 063364 012721 100006
4159 063370 012721 060726
4160 063374 005021
4161 063376 012721 000006
4162 063402 012701 060726
4163 063406 005021
4164 063410 005021
4165 063412 005011
4166 063414 000207
4167 063416
4168 063416
4169 063422 012701 060710
4170 063426 012721 100005
4171 063432 005021
4172 063434 005021
4173 063436 005011
4174 063440 000207
4175 063442
063442
063442 104401

```

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

```

```

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

```

L10061: TRAP C\$ETST

4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194

.SBTTL TEST 7: EXTENDED MODE FEATURES

```

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

```

REWIND WITH IMMEDIATE INTERRUPT

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

; THE TEST CONSISTS OF THE FOLLOWING 7 SUBTESTS

4195 063444
063444
4196 063444 012737 006354 002172
4201 063452 012700 073163
4202 063456 004737 016570
4203 063462 012737 000005 002210
4204 063470 005037 067566

```

BGNTST
T7::
MOV      #EPRT1,EPRTSW     ;PRIMARY ERROR MESSAGE
MOV      #TST35ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
JSR      PC,TSTSETUP      ;DO INITIAL TEST SETUP
MOV      #5,LOOPCNT       ;PERFORM 5 ITERATIONS
CLR      T35CNT           ;CLEAR TAPE RECORD COUNTER

```

4205
4206
4207
4208
4209
4210
4211

; TEST 7, SUBTEST 1

VERIFIES THAT A REWIND WITH IMMEDIATE INTERRUPT COMMAND, ISSUED WITH THE INTERRUPT ENABLE (IE) BIT CLEAR (0), CAUSES ALMOST

TEST 7: EXTENDED MODE FEATURES

```

4212                :               IMMEDIATE TERMINATION BUT NO INTERRUPT. STATUS IN THE MESSAGE
4213                :               BUFFER IS CHECKED TO VERIFY THAT THE MOTION (MOT) AND OPERATION
4214                :               IN PROGRESS (OPM) STATUS BITS ARE BOTH SET.
4215                :
4216                :
4217                :
4218                :
4219                :
4220                :
4221                :
4222                :
4222 063474          : T35LOOP:
4223 063474          : BGNSUB                               ; >>>>>>>>>> BEGIN SUBTEST >>>>>>>>>
                        :                                   T7.1:
                        :                                   TRAP      C$BSUB
4224 063474 104402  : JSR      PC,T35REST                     ;SET COMMAND PACKET
4225 063476 004737 073214  : CLR      INTRECV                        ;CLEAR INTERRUPT RECEIVED FLAG
4226 063502 005037 002216  : JSR      PC,T35RT2                       ;SET UP OTHER COMMAND PACKET
4227 063506 004737 073306  : JSR      PC,T35RT3                       ;SET UP OTHER COMMAND PACKET
4228 063512 004737 073350  : MOV      #65000.,T35DLY                  ;SET UP DELAY COUNTER
4229 063516 012737 176750 067572  : CLR      T35CNT                          ;CLEAR COUNTER
4230 063524 005037 067566  : JSR      PC,SOFINIT                       ;DO INITIALIZE ON CONTROLLER
4231 063530 004737 016054  : BCS      20$                             ;BR IF INIT WAS OK
4232 063536 103426          : DELAY   250                             ;DELAY ABOUT .25 SEC
4232 063536 012727 000250          : MOV      #250.(PC)+                       MOV      #250.(PC)+
4232 063542 000000          : .WORD   0                                 .WORD   0
4232 063544 013727 002116          : MOV      L$DLY.(PC)+                       MOV      L$DLY.(PC)+
4232 063550 000000          : .WORD   0                                 .WORD   0
4232 063552 005367 177772          : DEC      -6(PC)                           DEC      -6(PC)
4232 063556 001375          : BNE     -.4                               BNE     -.4
4232 063560 005367 177756          : DEC     -22(PC)                          DEC     -22(PC)
4232 063564 001367          : BNE     -.20                             BNE     -.20
4233 063566 005337 067572          : DEC      T35DLY                          ;BUMP COUNTER
4234 063572 001356          : BNE     10$                             ;BR, IF COUNTER NOT DONE
4235 063574 005237 002214          : INC      FATFLG                          ;ERROR COUNT
4239 063600 010001          : MOV      R0,R1                          ;CONTENTS OF TSSR REGISTER
4240 063602          : ERRDF  ERRNO,SFIERR,SFIMSG              ;FATAL ERROR TSSR WAS NOT OK
4240 063602 104455          : TRAP    C$ERDF                          TRAP    C$ERDF
4240 063604 001275          : .WORD   701                             .WORD   701
4240 063606 003646          : .WORD   SFIERR                          .WORD   SFIERR
4240 063610 012114          : .WORD   SFIMSG                          .WORD   SFIMSG
4241 063612 013737 002174 067440 20$: MOV      UNITN,T35DSW                    ;SET UP DRIVE NUMBER
4242 063620 012704 067420          : MOV      #T35PACKET,R4                   ;SUBROUTINE NEEDS PACKET ADDRESS
4243 063624 004737 010742          : JSR      PC,WRTCHR                       ;ISSUE WRITE CHARACTERISTICS
4244 063630 103407          : BCS     25$                             ;BR, IF COMMAND ISSUED OK
4245 063632 005237 002214          : INC      FATFLG                          ;ERROR COUNT
4249 063636 010001          : MOV      R0,R1                          ;SAVE CONTENTS OF TSSR
4250 063640          : ERRHRD ERRNO,WRTMSG,SFIMSG              ;WRITE CHARACTERISTICSC FAILED
4250 063640 104456          : TRAP    C$ERHRD                          TRAP    C$ERHRD
4250 063642 001276          : .WORD   702                             .WORD   702
4250 063644 005052          : .WORD   WRTMSG                          .WORD   WRTMSG
4250 063646 012114          : .WORD   SFIMSG                          .WORD   SFIMSG
4251 063650          : CKLOOP
4251 063650 104406          : 25$: LOOP IF SELECTED                          LOOP IF SELECTED
4251 063650          : TRAP    C$CLP1                          TRAP    C$CLP1
4252 063652 004737 011074          : JSR      PC,REWIND                       ;CALL TAPE REWIND COMMAND
4253 063656 103411          : BCS     30$                             ;BR, IF NO PROBLEM
4254 063660 010004          : MOV      R0,R4                          ;SET UP REWIND PACKET ADDRESS
4255 063662 016501 000002          : MOV      TSSR(R5),R1                     ;GET TSSR FOR PRINTOUT

```

TEST 7: EXTENDED MODE FEATURES

```

4256 063666 005237 002214          INC      FATFLG          ;ERROR COUNT
4260 063672          ERRHRD  ERRNO,T35RWN,PKTSSR ;REWIND NOT ACCEPTED
      063672 104456          TRAP      C$ERHRD
      063674 001277          .WORD    703
      063676 070674          .WORD    T35RWN
      063700 012126          .WORD    PKTSSR
4261 063702          30$:   CKLOOP          ;LOOP IF SELECTED
      063702 104406          TRAP      C$CLP1
4262 063704 013701 067450          MOV      T35BFR+6,R1    ;PICK UP XSTO
4263 063710 010102          MOV      R1,R2          ;SET UP EXPECTED
4264 063712 052702 000002          BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
4265 063716 020102          CMP      R1,R2          ;DOES EXP = REC'D
4266 063720 001406          BEQ      40$           ;BR, IF EQUAL (OK)
4267 063722 005237 002214          INC      FATFLG          ;ERROR COUNT
4271 063726          ERRHRD  ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      063726 104456          TRAP      C$ERHRD
      063730 001300          .WORD    704
      063732 070370          .WORD    T35BOT
      063734 015554          .WORD    EXPREC
4272 063736          40$:   CKLOOP          ;LOOP IF SELECTED
      063736 104406          TRAP      C$CLP1
4273 063740 012703 000024          MOV      #20.,R3        ;NUMBER OF RECORDS
4274 063744 012737 000400 067546          MOV      #256.,T35SZ    ;SET UP RECORD SIZE
4275 063752 013737 003116 067542          MOV      FREE,T35WB     ;ADDRESS OF WRITE BUFFER
4276
4277          ;*****
4278          ;
4279          ;WRITE DATA,ACK,CVC=1 COMMAND
4280          ;
4281          ;*****
4282
4283 063760 012737 140005 067540          MOV      #140005,T35PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
4284 063766 012704 067540          MOV      #T35PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
4285 063772 010465 000000          50$:   MOV      R4,TSDB(R5) ;ISSUE COMMAND
4286 063776 004737 016330          JSR      PC,WAITF      ;WAIT FOR SSR TO SET
4287 064002 016501 000002          MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
4288 064006 012702 000200          MOV      #SSR,R2       ;SET UP EXPECTED
4289 064012 020102          CMP      R1,R2         ;ARE THEY EQUAL
4290 064014 001406          BEQ      60$           ;BR, IF OK
4291 064016 005237 002214          INC      FATFLG          ;ERROR COUNT
4295 064022          ERRHRD  ERRNO,T35WDE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      064022 104456          TRAP      C$ERHRD
      064024 001301          .WORD    705
      064026 070316          .WORD    T35WDE
      064030 012126          .WORD    PKTSSR
4296 064032          60$:   CKLOOP          ;LOOP IF SELECTED
      064032 104406          TRAP      C$CLP1
4297 064034 005303          DEC      R3             ;BUMP RECORD COUNTER
4298 064036 001355          BNE      50$           ;BR, IF MORE RRECORDS TO COUNT
4299
4300          ;*****
4301          ;
4302          ;WAIT FOR TAPE TO STOP ALL MOTION
4303          ;
4304          ;*****
4305
4306 064040 012737 000012 067572          MOV      #10.,T35DLY   ;SET UP DELAY COUNTER

```

TEST 7: EXTENDED MODE FEATURES

4307	064046				70\$: DELAY 250			;WAIT ABOUT .25 SEC	MOV #250,(PC)
	064046	012727	000250						.WORD 0
	064052	000000							MOV L\$DLY,(PC)
	064054	013727	002116						.WORD 0
	064060	000000							DEC -6(PC)
	064062	005367	177772						BNE -.4
	064066	001375							DEC -22(PC)
	064070	005367	177756						BNE .-20
	064074	001367							
4308	064076	005337	067572			DEC T35DLY		;BUMP COUNTER DOWN	
4309	064102	001361				BNE 70\$;BR, IF MORE TO DELAY	
4310	064104	005737	002220			TST EXTFEA		;CHECK FOR EXTENDED FEATURES SW SWITCH	
4311	064110	001042				BNE 110\$;BR IF SWITCH IS ON	
4312	064112	112737	000200	067551		MOVB #200,T35BS1		;WRITE MISCELLANEOUS CONT/READ STATUS	
4313	064120	112737	000010	067550		MOVB #10,T35BS0		;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)	
4314	064126	012704	067530			MOV #T35PK2,R4		;WRITE SUBSYS MEM PACKET	
4315	064132	010465	000000			MOV R4,TSDB(R5)		;ISSUE COMMAND	
4316	064136	004737	016416			JSR PC,CHKTSSR		;WAIT FOR SSR	
4317	064142	103407				BCS 90\$;BR, IF NO ERROR	
4318	064144	010001				MOV R0,R1		;ERROR, SAVE TSSR	
4319	064146	005237	002214			INC FATFLG		;ERROR COUNT	
4323	064152					ERRHRD ERRNO,T35SSR,PKTSSR		;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS	
	064152	104456							TRAP C\$ERHRD
	064154	001302							.WORD 706
	064156	072452							.WORD T35SSR
	064160	012126							.WORD PKTSSR
4324	064162				90\$: CKLOOP			;LOOP IF SELECTED	
	064162	104406							TRAP C\$CLP1
4325	064164	012704	067420			MOV #T35PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS	
4326	064170	004737	010742			JSR PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS	
4327	064174	103407				BCS 100\$;BR, IF COMMAND ISSUED OK	
4328	064176	005237	002214			INC FATFLG		;ERROR COUNT	
4332	064202	010001				MOV R0,R1		;SAVE CONTENTS OF TSSR	
4333	064204					ERRHRD ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTICSC FAILED	
	064204	104456							TRAP C\$ERHRD
	064206	001303							.WORD 707
	064210	005052							.WORD WRTMSG
	064212	012114							.WORD SFIMSG
4334	064214				100\$: CKLOOP			;SCOPE LOOP	
	064214	104406							TRAP C\$CLP1
4335	064216	012737	176750	067572	110\$:	MOV #65000.,T35DLY		;SET UP DELAY COUNTER	
4336	064224	005037	067566			CLR T35CNT		;DELAY COUNTER	
4337									
4338									
4339									
4340									
4341									
4342									
4343									
4344	064230	012737	142012	067540		MOV #142012,T35PK3		;REWIND IMED. INTERRUPT,ACK,CVC=1,IE=0 COMMAND	
4345	064236	012704	067540			MOV #T35PK3,R4		;SET UP R4 WITH PACKET ADDRESS	
4346	064242	010465	000000			MOV R4,TSDB(R5)		;ISSUE COMMAND	
4347	064246	016501	000002		120\$:	MOV TSSR(R5),R1		;GET TSSR CONTENTS	
4348	064252	032701	000200			BIT #SSR,R1		;CHECK FOR SSR SET	
4349	064256	001021				BNE 130\$;BR, WHEN SSR IS SET	
4350	064260	005237	067566			INC T35CNT		;BUMP THE CYCLE COUNTER	
4351	064264					DELAY 1		;DELAY TO KEEP COUNTER DOWN	

TEST 7: EXTENDED MUDE FEATURES

```

064264 012727 000001                MOV    #1.(PC)+
064270 000000                .WORD 0
064272 013727 002116                MOV    L$DLY,(PC)+
064276 000000                .WORD 0
064300 005367 177772                DEC    -6(PC)
064304 001375                BNE    -4
064306 005367 177756                DEC    -22(PC)
064312 001367                BNE    -20
4352 064314 005337 067572          DEC    T35DLY          ;DROP DEAD TIMER BUMP DOWN
4353 064320 001352                BNE    120$           ;BR, IF MORE TIME TO GO
4354 064322 012702 000200          130$: MOV    #SSR,R2       ;SET UP EXPECTED
4355 064326 020102                CMP    R1,R2          ;ARE THEY EQUAL
4356 064330 001406                BEQ    140$           ;BR, IF OK
4357 064332 005237 002214          INC    FATFLG          ;ERROR COUNT
4361 064336                ERRHRD ERRNO,T35RWE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERHRD
                                .WORD 708
                                .WORD T35RWE
                                .WORD PKTSSR
                                TRAP    C$CLP1
064336 104456
064340 001304
064342 073020
064344 012126
4362 064346                140$: CKLOOP          ;LOOP IF SELECTED
064346 104406                TST    INTRECV        ;CHECK FOR INTERRUPTS
4363 064350 005737 002216          BEQ    150$           ;BR, IF NO INTERRUPTS DETECTED
4364 064354 001410                MOV    TSSR(R5),R1    ;GET TSSR STATUS FOR PRINTOUT
4365 064356 016501 000002          INC    FATFLG          ;ERROR COUNT
4366 064362 005237 002214          ERRHRD ERRNO,T35INT,PKTSSR ;INTERRUPT RECEIVED (BAD)
                                TRAP    C$ERHRD
                                .WORD 709
                                .WORD T35INT
                                .WORD PKTSSR
4370 064366 104456
064370 001305
064372 072631
064374 012126
4371 064376                150$: CKLOOP          ;LOOP IF SELECTED
064376 104406                TRAP    C$CLP1
4372
4373 ;*****
4374 ;
4375 ;NOW CHECK FOR THE MOTION BITS SET
4376 ;
4377 ;*****
4378
4379 064400 013701 067450          MOV    T35BFR+6,R1    ;PICK UP XST0
4380 064404 010102                MOV    R1,R2          ;SET UP EXPECTED
4381 064406 052702 000200          BIS    #BIT7,R2       ;SET MOT BIT IN EXPECTED
4382 064412 020102                CMP    R1,R2          ;DOES EXP = REC'D
4383 064414 001406                BEQ    160$           ;BR, IF EQUAL (OK)
4384 064416 005237 002214          INC    FATFLG          ;ERROR COUNT
4388 064422                ERRHRD ERRNO,T35MOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD 710
                                .WORD T35MOT
                                .WORD EXPREC
064422 104456
064424 001306
064426 072533
064430 015554
4389 064432                160$: CKLOOP          ;LOOP IF SELECTED
064432 104406                TRAP    C$CLP1
4390 064434 013701 067454          MOV    T35BFR+12,R1   ;PICK UP XST2
4391 064440 010102                MOV    R1,R2          ;SET UP EXPECTED
4392 064442 052702 100000          BIS    #BIT15,R2      ;SET OPM BIT IN EXPECTED
4393 064446 020102                CMP    R1,R2          ;DOES EXP = REC'D
4394 064450 001406                BEQ    170$           ;BR, IF EQUAL (OK)

```


TEST 7: EXTENDED MODE FEATURES

```

4436 064622      DELAY 250      ;DELAY ABOUT .25 SEC
      064622 012727 000250      MOV      #250,(PC)+
      064626 000000      .WORD 0
      064630 013727 002116      MOV      L$DLY,(PC)+
      064634 000000      .WORD 0
      064636 005367 177772      DEC      -6(PC)
      064642 001375      BNE      -.4
      064644 005367 177756      DEC      -22(PC)
      064650 001367      BNE      -.20
4437 064652 005337 067572      DEC      T35DLY      ;BUMP COUNTER
4438 064656 001356      BNE      10$         ;BR, IF COUNTER NOT DONE
4439 064660 005237 002214      INC      FATFLG      ;ERROR COUNT
4443 064664 010001      MOV      R0,R1      ;CONTENTS OF TSSR REGISTER
4444 064666      ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      064666 104455      TRAP  C$ERDF
      064670 001310      .WORD 712
      064672 003646      .WORD SFIERR
      064674 012114      .WORD SFIMSG
4445 064676 013737 002174 067440 20$: MOV      UNITN,T35DSW ;SET UP DRIVE NUMBER
4446 064704 012704 067420      MOV      #T35PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4447 064710 004737 010742      JSR      PC,WRTCHR   ;ISSUE WRITE CHARACTERISTICS
4448 064714 103407      BCS      25$         ;BR, IF COMMAND ISSUED OK
4449 064716 005237 002214      INC      FATFLG      ;ERROR COUNT
4453 064722 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4454 064724      ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      064724 104456      TRAP  C$ERHRD
      064726 001311      .WORD 713
      064730 005052      .WORD WRTMSG
      064732 012114      .WORD SFIMSG
4455 064734      25$:  CKLOOP      ;LOOP IF SELECTED
      064734 104406      TRAP  C$CLP1
4456 064736 004737 011074      JSR      PC,REWIND  ;CALL TAPE REWIND COMMAND
4457 064742 103411      BCS      30$         ;BR, IF NO PROBLEM
4458 064744 010004      MOV      R0,R4      ;SET UP REWIND PACKET ADDRESS
4459 064746 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
4460 064752 005237 002214      INC      FATFLG      ;ERROR COUNT
4464 064756      ERRHRD ERRNO,T35RWN,PKTSSR ;REWIND NOT ACCEPTED
      064756 104456      TRAP  C$ERHRD
      064760 001312      .WORD 714
      064762 070674      .WORD T35RWN
      064764 012126      .WORD PKTSSR
4465 064766      30$:  CKLOOP      ;LOOP IF SELECTED
      064766 104406      TRAP  C$CLP1
4466 064770 013701 067450      MOV      T35BFR+6,R1 ;PICK UP XSTO
4467 064774 010102      MOV      R1,R2      ;SET UP EXPECTED
4468 064776 052702 000002      BIS      #BIT1,R2   ;SET BOT BIT IN EXPECTED
4469 065002 020102      CMP      R1,R2      ;DOES EXP = REC'D
4470 065004 001406      BEQ      40$         ;BR, IF EQUAL (OK)
4471 065006 005237 002214      INC      FATFLG      ;ERROR COUNT
4475 065012      ERRHRD ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      065012 104456      TRAP  C$ERHRD
      065014 001313      .WORD 715
      065016 070370      .WORD T35BOT
      065020 015554      .WORD EXPREC
4476 065022      40$:  CKLOOP      ;LOOP IF SELECTED
      065022 104406      TRAP  C$CLP1
4477 065024 012703 000024      MOV      #20.,R3    ;NUMBER OF RECORDS

```

TEST 7: EXTENDED MODE FEATURES

```

4478 065030 012737 000400 067546      MOV    #256.,T35SZ      ;SET UP RECORD SIZE
4479 065036 013737 003116 067542      MOV    FREE,T35WB      ;ADDRESS OF WRITE BUFFER
4480
4481      ;*****
4482      ;
4483      ;WRITE DATA,ACK,CVC=1 COMMAND
4484      ;
4485      ;*****
4486
4487 065044 012737 140005 067540      MOV    #140005,T35PK3  ;WRITE DATA,ACK,CVC=1 COMMAND
4488 065052 012704 067540                MOV    #T35PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4489 065056 010465 000000 50$:   MOV    R4,TSDB(R5)     ;ISSUE COMMAND
4490 065062 004737 016330                JSR    PC,WAITF        ;WAIT FOR SSR TO SET
4491 065066 016501 000002                MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
4492 065072 012702 000200                MOV    #SSR,R2        ;SET UP EXPECTED
4493 065076 020102                    CMP    R1,R2           ;ARE THEY EQUAL
4494 065100 001406                    BEQ    60$             ;BR, IF OK
4495 065102 005237 002214                INC    FATFLG          ;ERROR COUNT
4499 065106                    ERRHRD ERRNO,T35WDE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      065106 104456                                TRAP   C$ERHRD
      065110 001314                                .WORD 716
      065112 070316                                .WORD T35WDE
      065114 012126                                .WORD PKTSSR
4500 065116                    60$:   CKLOOP                ;LOOP IF SELECTED
      065116 104406                                TRAP   C$CLP1
4501
4502      ;*****
4503      ;
4504      ;WAIT FOR TAPE TO STOP ALL MOTION
4505      ;
4506      ;*****
4507
4508 065120 012737 000012 067572      MOV    #10.,T35DLY    ;SET UP DELAY COUNTER
4509 065126                    70$:   DELAY    250          ;WAIT ABOUT .25 SEC
      065126 012727 000250                                MOV    #250,(PC)+
      065132 000000                                .WORD 0
      065134 013727 002116                                MOV    L$DLY,(PC)+
      065140 000000                                .WORD 0
      065142 005367 177772                                DEC    -6(PC)
      065146 001375                                BNE    -.4
      065150 005367 177756                                DEC    -22(PC)
      065154 001367                                BNE    -.20
4510 065156 005337 067572                DEC    T35DLY          ;BUMP COUNTER DOWN
4511 065162 001361                    BNE    70$            ;BR, IF MORE TO DELAY
4512 065164 005737 002220                TST    EXTFEA         ;CHECK FOR EXTENDED FEATURES SW SWITCH
4513 065170 001042                    BNE    110$           ;BR IF SWITCH IS ON
4514 065172 112737 000200 067551      MOV    #200,T35BS1    ;WRITE MISCELLANEOUS CONT/READ STATUS
4515 065200 112737 000010 067550      MOV    #10,T35BS0     ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4516 065206 012704 067530                MOV    #T35PK2,R4     ;WRITE SUBSYS MEM PACKET
4517 065212 010465 000000                MOV    R4,TSDB(R5)    ;ISSUE COMMAND
4518 065216 004737 016416                JSR    PC,CHKTSSR     ;WAIT FOR SSR
4519 065222 103407                    BCS    90$            ;BR, IF NO ERROR
4520 065224 010001                    MOV    R0,R1          ;ERROR, SAVE TSSR
4521 065226 005237 002214                INC    FATFLG         ;ERROR COUNT
4525 065232                    ERRHRD ERRNO,T35SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      065232 104456                                TRAP   C$ERHRD
      065234 001315                                .WORD 717

```


TEST 7: EXTENDED MODE FEATURES

```

065236 072452                                     .WORD T35SSR
065240 012126                                     .WORD PKTSSR
4526 065242          90$: CKLOOP                     ;LOOP IF SELECTED          TRAP C$CLP1
065242 104406                                     ;SUBROUTINE NEEDS PACKET ADDRESS
4527 065244 012704 067420      MOV    #T35PACKET,R4      ;ISSUE WRITE CHARACTERISTICS
4528 065250 004737 010742      JSR    PC,WRTCHR      ;BR, IF COMMAND ISSUED OK
4529 065254 103407             BCS    100$          ;ERROR COUNT
4530 065256 005237 002214      INC    FATFLG        ;SAVE CONTENTS OF TSSR
4534 065262 010001             MOV    R0,R1        ;WRITE CHARACTERISTICS FAILED
4535 065264             ERRHRD ERRNO,WRTMSG,SFIMSG
065264 104456                                     TRAP C$ERHRD
065266 001316                                     .WORD 718
065270 005052                                     .WORD WRTMSG
065272 012114                                     .WORD SFIMSG
4536 065274          100$: CKLOOP                   ;SCOPE LOOP          TRAP C$CLP1
065274 104406                                     ;SET UP DELAY COUNTER
4537 065276 012737 176750 067572 110$: MOV    #65000.,T35DLY
4538 065304 005037 067566      CLR    T35CNT      ;DELAY COUNTER
4539
4540      ;*****
4541      ;
4542      ;REWIND IMMEDIATE,ACK,CVC=1,IE=1 COMMAND
4543      ;
4544      ;*****
4545
4546 065310 012737 142212 067540      MOV    #142212,T35PK3 ;REWIND IMMEDIATE,ACK,CVC=1,IE=1 COMMAND
4547 065316 012704 067540      MOV    #T35PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
4548 065322 010465 000000      MOV    R4,TSDB(R5) ;ISSUE COMMAND
4549 065326 016501 000002      MOV    TSSR(R5),R1 ;GET TSSR CONTENTS
4550 065332 032701 000200      BIT    #SSR,R1     ;CHECK FOR SSR SET
4551 065336 001021             BNE    130$        ;BR, WHEN SSR IS SET
4552 065340 005237 067566      INC    T35CNT      ;BUMP THE CYCLE COUNTER
4553 065344             DELAY    1         ;DELAY TO KEEP COUNTER DOWN
065344 012727 000001             MOV    #1,(PC)-
065350 000000             .WORD 0
065352 013727 002116             MOV    L$DLY,(PC)-
065356 000000             .WORD 0
065360 005367 177772             DEC    -6(PC)
065364 001375             BNE    -4
065366 005367 177756             DEC    -22(PC)
065372 001367             BNE    -20
4554 065374 005337 067572      DEC    T35DLY      ;DROP DEAD TIMER BUMP DOWN
4555 065400 001352             BNE    120$        ;BR, IF MORE TIME TO GO
4556 065402 012702 000200      MOV    #SSR,R2     ;SET UP EXPECTED
4557 065406 020102             CMP    R1,R2       ;ARE THEY EQUAL
4558 065410 001406             BEQ    140$        ;BR, IF OK
4559 065412 005237 002214      INC    FATFLG      ;ERROR COUNT
4563 065416             ERRHRD ERRNO,T35RWE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
065416 104456                                     TRAP C$ERHRD
065420 001317                                     .WORD 719
065422 073020                                     .WORD T35RWE
065424 012126                                     .WORD PKTSSR
4564 065426          140$: CKLOOP                   ;LOOP IF SELECTED          TRAP C$CLP1
065426 104406                                     ;CHECK FOR INTERRUPTS
4565 065430 005737 002216      TST    INTRECV      ;BR, IF INTERRUPTS DETECTED
4566 065434 001010             BNE    150$
4567 065436 016501 000002      MOV    TSSR(R5),R1 ;GET TSSR STATUS FOR PRINTOUT

```

TEST 7: EXTENDED MODE FEATURES

```

4568 065442 005237 002214          INC    FATFLG
4572 065446              ERRHRD  ERRNO,T35NIN,PKTSSR    ;ERROR COUNT
                                ; INTERRUPT NOT RECEIVED (BAD)
                                TRAP  C$ERHRD
                                .WORD 720
                                .WORD T35NIN
                                .WORD PKTSSR
                                TRAP  C$CLP1
      065446 104456
      065450 001320
      065452 073106
      065454 012126
4573 065456          150$: CKLOOP           ;LOOP IF SELECTED
      065456 104406
4574
4575
4576          ;*****
4577          ;
4578          ;NOW CHECK FOR THE MOTION BITS SET
4579          ;
4580          ;*****
4581 065460 013701 067450          MOV    T35BFR+6,R1        ;PICK UP XST0
4582 065464 010102          MOV    R1,R2            ;SET UP EXPECTED
4583 065466 052702 000200          BIS    #BIT7,R2        ;SET MOT BIT IN EXPECTED
4584 065472 020102          CMP    R1,R2            ;DOES EXP = REC'D
4585 065474 001406          BEQ    160$            ;BR, IF EQUAL (OK)
4586 065476 005237 002214          INC    FATFLG          ;ERROR COUNT
4590 065502          ERRHRD  ERRNO,T35MOT,EXPREC    ;TAPE NOT AT BOT AFTER REWIND
                                TRAP  C$ERHRD
                                .WORD 721
                                .WORD T35MOT
                                .WORD EXPREC
      065502 104456
      065504 001321
      065506 072533
      065510 015554
4591 065512          160$: CKLOOP           ;LOOP IF SELECTED
      065512 104406                                TRAP  C$CLP1
4592 065514 013701 067454          MOV    T35BFR+12,R1     ;PICK UP XST2
4593 065520 010102          MOV    R1,R2            ;SET UP EXPECTED
4594 065522 052702 100000          BIS    #BIT15,R2       ;SET OPM BIT IN EXPECTED
4595 065526 020102          CMP    R1,R2            ;DOES EXP = REC'D
4596 065530 001406          BEQ    170$            ;BR, IF EQUAL (OK)
4597 065532 005237 002214          INC    FATFLG          ;ERROR COUNT
4601 065536          ERRHRD  ERRNO,T35OPM,EXPREC    ;OPM BIT NOT SET
                                TRAP  C$ERHRD
                                .WORD 722
                                .WORD T35OPM
                                .WORD EXPREC
      065536 104456
      065540 001322
      065542 072722
      065544 015554
4602 065546          170$: CKLOOP           ;LOOP IF SELECTED
      065546 104406                                TRAP  C$CLP1
4603 065550 012737 000027 067572   MOV    #23.,T35DLY     ;SET UP DELAY COUNTER
4604 065556          175$: DELAY    250          ;START DELAY
      065556 012727 000250          MOV    #250,(PC)+
      065562 000000          .WORD 0
      065564 013727 002116          MOV    L$DLY,(PC)+
      065570 000000          .WORD 0
      065572 005367 177772          DEC    -6(PC)
      065576 001375          BNE    -.4
      065600 005367 177756          DEC    -22(PC)
      065604 001367          BNE    -.20
4605 065606 005337 067572          DEC    T35DLY          ;BUMP DELAY COUNTER
4606 065612 001361          BNE    175$            ;BR, IF MORE DELAY
4607 065614          ENDSUB
      065614
      065614 104403          L10065: TRAP  C$ESUB
4608 065616 023727 002214 000017   CMP    FATFLG,#15.     ;IS ERROR COUNT AT 25

```


TEST 7: EXTENDED MODE FEATURES

```

4712 066140 016501 000002          MOV   TSSR(R5),R1          ;GET TSSR CONTENTS
4713 066144 012702 000200          MOV   #SSR,R2           ;SET UP EXPECTED
4714 066150 020102                CMP   R1,R2             ;ARE THEY EQUAL
4715 066152 001406                BEQ   90$               ;BR, IF OK
4716 066154 005237 002214          INC   FATFLG            ;ERROR COUNT
4720 066160                ERRHRD  ERRNO,T35WRF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA RETRY
                                TRAP   C$ERHRD
                                .WORD  728
                                .WORD  T35WRF
                                .WORD  PKTSSR
4721 066170                90$:  CKLOOP            ;LOOP IF SELECTED
                                TRAP   C$CLP1
                                .WORD  728
4722 066172 005723                TST   (R3)+             ;BUMP RECORD SIZE COUNTER
4723 066174 020327 000052          CMP   R3,#42           ;AT 42 SIZE YET
4724 066200 001315                BNE   65$               ;BR, IF MORE RECORDS TO WRITE
4725 066202 004737 011074          JSR   PC,REWIND        ;CALL TAPE REWIND COMMAND
4726 066206 103411                BCS   230$              ;BR, IF NO PROBLEM
4727 066210 010001                MOV   R0,R1            ;SAVE TSSR
4728 066212 016501 000002          MOV   TSSR(R5),R1      ;GET TSSR CONTENTS
4729 066216 005237 002214          INC   FATFLG            ;ERROR COUNT
4733 066222                ERRHRD  ERRNO,T35RWN,EXPREC ;REWIND NOT ACCEPTED
                                TRAP   C$ERHRD
                                .WORD  729
                                .WORD  T35RWN
                                .WORD  EXPREC
4734 066232                230$: CKLOOP            ;LOOP IF SELECTED
                                TRAP   C$CLP1
                                .WORD  729
4735 066234 013701 067450          MOV   T35BFR+6,R1      ;PICK UP XSTO
4736 066240 010102                MOV   R1,R2            ;SET UP EXPECTED
4737 066242 052702 000002          BIS   #BIT1,R2         ;SET BOT BIT IN EXPECTED
4738 066246 020102                CMP   R1,R2            ;DOES EXP = REC'D
4739 066250 001406                BEQ   240$              ;BR, IF EQUAL (OK)
4740 066252 005237 002214          INC   FATFLG            ;ERROR COUNT
4744 066256                ERRHRD  ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP   C$ERHRD
                                .WORD  730
                                .WORD  T35BOT
                                .WORD  EXPREC
4745 066266                240$: CKLOOP            ;LOOP IF SELECTED
                                TRAP   C$CLP1
                                .WORD  730
4746 066270 012703 000024          MOV   #20.,R3           ;STARTING RECORD SIZE
4747 066274 013737 003116 067542  MOV   FREE,T35RB       ;STARTING READ BUFFER ADDRESS
4748
4749 ;*****
4750 ;
4751 ;READ DATA,ACK COMMAND
4752 ;
4753 ;*****
4754
4755 066302 012737 100001 067540 265$: MOV   #100001,T35PK3    ;READ DATA,ACK COMMAND
4756 066310 012704 067540          MOV   #T35PK3,R4       ;SET UP R4 WITH PACKET ADDRESS
4757 066314 012700 177777          MOV   #177777,R0       ;SET PATTERN IN CORRECT REGISTER
4758 066320 004737 017502          JSR   PC,FILLMEM       ;FILL MEMORY WITH RECORD SIZE
4759 066324 010337 067546          MOV   R3,T35SZ         ;SET UP RECORD SIZE IN PACKET
4760 066330 010465 000000          MOV   R4,TSDB(R5)     ;ISSUE COMMAND
4761 066334 004737 016330          JSR   PC,WAITF         ;WAIT FOR SSR TO SET
4762 066340 016501 000002          MOV   TSSR(R5),R1     ;GET TSSR CONTENTS

```


TEST 7: EXTENDED MODE FEATURES

```
                                T7.4:
066474                               TRAP     C$BSUB
066474 104402
4813 066476 004737 073214        JSR     PC,T35REST                ;SET COMMAND PACKET
4814 066502 004737 073306        JSR     PC,T35RT2                 ;SET UP OTHER COMMAND PACKET
4815 066506 004737 073350        JSR     PC,T35RT3                 ;SET UP OTHER COMMAND PACKET
4816 066512 012737 176750 067572  MOV     #65000.,T35DLY            ;SET UP DELAY COUNTER
4817 066520 004737 016054        JSR     PC,SOFINIT                ;DO INITIALIZE ON CONTROLLER
4818 066524 103426                BCS     20$                      ;BR IF INIT WAS OK
4819 066526                DELAY   250                    ;DELAY ABOUT .25 SEC

                                MOV     #250,(PC)+
                                .WORD  0
066526 012727 000250                MOV     L$DLY,(PC)+
066532 000000                .WORD  0
066534 013727 002116                MOV     .WORD  0
066540 000000                DEC     -6(PC)
066542 005367 177772                BNE    -.4
066546 001375                DEC     -22(PC)
066550 005367 177756                BNE    .-20
066554 001367
4820 066556 005337 067572        DEC     T35DLY                   ;BUMP COUNTER
4821 066562 001356                BNE    10$                       ;BR, IF COUNTER NOT DONE
4822 066564 005237 002214        INC     FATFLG                   ;ERROR COUNT
4826 066570 010001        MOV     R0,R1                    ;CONTENTS OF TSSR REGISTER
4827 066572        ERRDF  ERRNO,SFIERR,SFIMSG      ;FATAL ERROR TSSR WAS NOT OK
                                TRAP     C$ERDF
                                .WORD  733
                                .WORD  SFIERR
                                .WORD  SFIMSG
066572 104455
066574 001335
066576 003646
066600 012114
4828 066602 013737 002174 067440 20$: MOV     UNITN,T35DSW             ;SET UP UNIT (DRIVE) NUMBER
4829 066610 012704 067420        MOV     #T35PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
4830 066614 004737 010742        JSR     PC,WRTCHR                ;ISSUE WRITE CHARACTERISTICS
4831 066620 103407                BCS     23$                      ;BR, IF COMMAND ISSUED OK
4832 066622 005237 002214        INC     FATFLG                   ;ERROR COUNT
4836 066626 010001        MOV     R0,R1                    ;SAVE CONTENTS OF TSSR
4837 066630        ERRHRD ERRNO,WRTMSG,SFIMSG    ;WRITE CHARACTERISTICSC FAILED
                                TRAP     C$ERHRD
                                .WORD  734
                                .WORD  WRTMSG
                                .WORD  SFIMSG
066630 104456
066632 001336
066634 005052
066636 012114
4838 066640                CKLOOP                          ;LOOP IF SELECTED
                                TRAP     C$CLP1
066640 104406
4839 066642 004737 011074        JSR     PC,REWIND                ;CALL TAPE REWIND COMMAND
4840 066646 103411                BCS     30$                      ;BR, IF NO PROBLEM
4841 066650 016501 000002        MOV     TSSR(R5),R1              ;GET TSSR CONTENTS
4842 066654 010004        MOV     R0,R4                    ;GET PACKET ADDRESS
4843 066656 005237 002214        INC     FATFLG                   ;ERROR COUNT
4847 066662        ERRHRD ERRNO,T35RWN,PKTSSR    ;REWIND NOT ACCEPTED
                                TRAP     C$ERHRD
                                .WORD  735
                                .WORD  T35RWN
                                .WORD  PKTSSR
066662 104456
066664 001337
066666 070674
066670 012126
4848 066672                CKLOOP                          ;LOOP IF SELECTED
                                TRAP     C$CLP1
066672 104406
4849 066674 013701 067450        MOV     T35BFR+6,R1              ;PICK UP XSTO
4850 066700 010102        MOV     R1,R2                    ;SET UP EXPECTED
4851 066702 052702 000002        BIS     #BIT1,R2                 ;SET BOT BIT IN EXPECTED
4852 066706 020102        CMP     R1,R2                    ;DOES EXP = REC'D
4853 066710 001406                BEQ     40$                      ;BR, IF EQUAL (OK)
4854 066712 005237 002214        INC     FATFLG                   ;ERROR COUNT
```

TEST 7: EXTENDED MODE FEATURES

```

4858 066716          ERRHRD  ERRNO,T35BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
        066716 104456          TRAP           C$ERHRD
        066720 001340          .WORD         736
        066722 070370          .WORD         T35BOT
        066724 015554          .WORD         EXPREC
4859 066726          40$:   CKLOOP          ;LOOP IF SELECTED          TRAP           C$CLP1
        066726 104406
4860 066730 012703 000024      MOV      #20.,R3          ;STARTING RECORD SIZE
4861 066734 013737 003116 067542  MOV      FREE,T35WB      ;STARTING WRITE BUFFER ADDRESS
4862
4863      ;*****
4864      ;
4865      ;WRITE DATA,CVC=1,ACK COMMAND
4866      ;
4867      ;*****
4868
4869 066742 012737 140005 067540 65$:   MOV      #140005,T35PK3      ;WRITE DATA,CVC=1,ACK COMMAND
4870 066750 012704 067540      MOV      #T35PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4871 066754 010300          MOV      R3,R0           ;SET PATTERN IN CORRECT REGISTER
4872 066756 004737 017502      JSR      PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
4873 066762 010337 067546      MOV      R3,T35SZ       ;SET UP RECORD SIZE IN PACKET
4874 066766 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
4875 066772 004737 016330      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
4876 066776 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
4877 067002 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED
4878 067006 020102          CMP      R1,R2          ;ARE THEY EQUAL
4879 067010 001406          BEQ      80$           ;BR, IF OK
4880 067012 005237 002214      INC      FATFLG        ;ERROR COUNT
4884 067016          ERRHRD  ERRNO,T35WDC,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA
        067016 104456          TRAP           C$ERHRD
        067020 001341          .WORD         737
        067022 071230          .WORD         T35WDC
        067024 012126          .WORD         PKTSSR
4885 067026          80$:   CKLOOP          ;LOOP IF SELECTED          TRAP           C$CLP1
        067026 104406
4886
4887      ;*****
4888      ;
4889      ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
4890      ;
4891      ;*****
4892
4893 067030 012737 111005 067540      MOV      #111005,T35PK3  ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
4894 067036 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
4895 067042 004737 016330      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
4896 067046 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
4897 067052 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED
4898 067056 020102          CMP      R1,R2          ;ARE THEY EQUAL
4899 067060 001406          BEQ      90$           ;BR, IF OK
4900 067062 005237 002214      INC      FATFLG        ;ERROR COUNT
4904 067066          ERRHRD  ERRNO,T35WRF,EXPREC      ;TSSR INCORRECT AFTER WRITE DATA RETRY
        067066 104456          TRAP           C$ERHRD
        067070 001342          .WORD         738
        067072 072275          .WORD         T35WRF
        067074 015554          .WORD         EXPREC
4905 067076          90$:   CKLOOP          ;LOOP IF SELECTED          TRAP           C$CLP1
        067076 104406

```


TEST 7: EXTENDED MODE FEATURES

```

4906 067100 005723          TST      (R3)+          ;BUMP RECORD SIZE COUNTER
4907 067102 020327 000052   CMP      R3,#42.       ;AT 42 SIZE YET
4908 067106 001315          BNE      65$           ;BR, IF MORE RECORDS TO WRITE
4909 067110 004737 011074   JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
4910 067114 103411          BCS      230$         ;BR, IF NO PROBLEM
4911 067116 016501 000002   MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
4912 067122 010004          MOV      R0,R4        ;GET PACKET ADDRESS
4913 067124 005237 002214   INC      FATFLG        ;ERROR COUNT
4917 067130          ERRHRD  ERRNO,T35RWN,PKTSSR ;REWIND NOT ACCEPTED
        067130 104456          TRAP      C$ERHRD
        067132 001343          .WORD    739
        067134 070674          .WORD    T35RWN
        067136 012126          .WORD    PKTSSR
4918           230$: CKLOOP          ;LOOP IF SELECTED          TRAP      C$CLP1
        067140 104406          MOV      T35BFR+6,R1  ;PICK UP XSTO
4919 067142 013701 067450   MOV      R1,R2        ;SET UP EXPECTED
4920 067146 010102          BIS      #BIT1,R2     ;SET BOT BIT IN EXPECTED
4921 067150 052702 000002   CMP      R1,R2        ;DOES EXP = REC'D
4922 067154 020102          BEQ      240$         ;BR, IF EQUAL (OK)
4923 067156 001406          INC      FATFLG        ;ERROR COUNT
4924 067160 005237 002214   ERRHRD  ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
        067164 104456          TRAP      C$ERHRD
        067166 001344          .WORD    740
        067170 070370          .WORD    T35BOT
        067172 015554          .WORD    EXPREC
4929           240$: CKLOOP          ;LOOP IF SELECTED          TRAP      C$CLP1
        067174 104406          MOV      #20.,R3      ;STARTING RECORD SIZE
4930 067176 012703 000024   MOV      FREE,T35RB   ;STARTING READ BUFFER ADDRESS
4931 067202 013737 003116 067542
4932
4933          ;*****
4934          ;
4935          ;READ DATA,ACK COMMAND
4936          ;
4937          ;*****
4938
4939 067210 012737 100001 067540 265$: MOV      #100001,T35PK3 ;READ DATA,ACK COMMAND
4940 067216 012704 067540    MOV      #T35PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
4941 067222 010337 067546    MOV      R3,T35SZ    ;SET UP RECORD SIZE IN PACKET
4942 067226 010465 000000    MOV      R4,T5DB(R5) ;ISSUE COMMAND
4943 067232 004737 016330    JSR      PC,WAITF    ;WAIT FOR SSR TO SET
4944 067236 016501 000002    MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
4945 067242 012702 000200    MOV      #SSR,R2    ;SET UP EXPECTED
4946 067246 020102          CMP      R1,R2      ;ARE THEY EQUAL
4947 067250 001406          BEQ      280$         ;BR, IF OK
4948 067252 005237 002214   INC      FATFLG        ;ERROR COUNT
4952 067256          ERRHRD  ERRNO,T35RDF,PKTSSR ;TSSR INCORRECT AFTER READ DATA
        067256 104456          TRAP      C$ERHRD
        067260 001345          .WORD    741
        067262 067662          .WORD    T35RDF
        067264 012126          .WORD    PKTSSR
4953           280$: CKLOOP          ;LOOP IF SELECTED          TRAP      C$CLP1
        067266 104406          MOV      FREE,R2     ;GET BUFFER ADDRESS
4954 067270 013702 003116   MOV      R3,R4        ;GET RECORD SIZE
4955 067274 010304          SUB      #20.,R4     ;POINT BACK TO 1ST RECORD
4956 067276 162704 000024

```


TEST 7: EXTENDED MODE FEATURES

```

5009 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
5010 ;
5012 067530 .=<.10>E177770
5014 067530 T35PK2:
5015 067530 100006 ;WRITE SUB SYS MEM COMMAND, AND ACK
5016 067532 067550 ;ADDRESS OF SELECT BLOCK DATA
5017 067534 000000 .WORD 100006
5018 067536 000006 .WORD T35BF2
5019 ;WORD 0
5023 067540 T35PK3: ;WORD 6. ;SIZE OF DATA PACKET
5024 067540 100005 ;WORD 100005 ;REREAD COMMAND, AND ACK
5025 067542 T35RB:
5026 067542 003116 T35WB: .WORD FREE ;ADDRESS OF WRITE BUFFER
5027 067544 000000 .WORD 0
5028 067546 000000 T35SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
5029 ;.EVEN
5030 ;
5031 ;
5032 ;
5033 067550 T35BF2:
5034 067550 010 T35BS0: .BYTE 10 ;BSELO AREA
5035 067551 200 T35BS1: .BYTE 200 ;BSEL1 AREA
5036 067552 000000 T35S2: .WORD 0 ;SEL 2 AREA
5037 067554 000000 T35S3: .WORD 0 ;DATA AREA
5038 ;
5039 ;
5040 ;.EVEN
5041 ;TAPE MOTION PACKET COMMAND VALUES
5042 ;
5043 067556 100205 T35RN: .WORD 100205 ;REREAD DATA (NEXT)
5044 067560 100605 T35WDR: .WORD 100605 ;REREAD DATA RETRY
5045 067562 102205 T35CON: .WORD 102205 ;WRITE CONTINUOUS
5046 067564 177777 .WORD 177777 ;END OF DATA
5047 ;
5048 ;
5049 067566 000000 T35CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
5050 067570 000000 T35CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
5051 067572 000000 T35DLY: .WORD 0 ;DELAY COUNTER
5052 ;+
5053 ;LOCAL TEXT MESSAGES FOR TEST
5054 ;-
5055 ;
5056 067574 124 141 160 T35WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
5057 067662 124 123 123 T35RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
5058 067731 122 105 122 T35RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
5059 070026 120 117 123 T35SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
5060 070110 122 111 102 T35LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
5061 070160 124 123 123 T35WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
5062 070235 111 154 154 T35LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
5063 070316 124 123 123 T35WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
5064 070370 124 141 160 T35BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
5065 070463 127 122 111 T35TIM: .ASCIZ 'WRITE DATA RETRY'S Erase Tape Not Long Enough'
5066 070540 122 105 122 T35EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
5067 070617 124 123 123 T35TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
5068 070674 122 145 167 T35RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
5069 070743 122 101 115 T35RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
5070 071016 124 123 123 T35AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'

```


TEST 8: RECORD BUFFERING

```

5282 073730          50$:  CKLOOP      ;LOOP IF SELECTED
      073730 104406          TRAP    C$CLP1
5283 073732 012737 003720 075736  MOV    #2000.,T36SZ  ;SET UP RECORD SIZE
5284 073740 013737 003116 075732  MOV    FREE,T36WB  ;ADDRESS OF WRITE BUFFER
5285 073746 012737 140005 075730  MOV    #140005,T36PK3 ;WRITE DATA,ACK,CVC-1 COMMAND
5286 073754 012704 075730  MOV    #T36PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
5287 073760 010465 000000  MOV    R4,TSDB(R5) ;ISSUE COMMAND
5288 073764 004737 016330  JSR    PC,WAITF  ;WAIT FOR SSR TO SET
5289 073770 016501 000002  MOV    TSSR(R5),R1 ;GET TSSR CONTENTS
5290 073774 012702 000200  MOV    #SSR,R2    ;SET UP EXPECTED
5291 074000 020102  CMP    R1,R2      ;ARE THEY EQUAL
5292 074002 001406  BEQ    60$       ;BR, IF OK
5293 074004 005237 002214  INC    FATFLG    ;ERROR COUNT
5297 074010  ERRHRD  ERRNO,WRterr,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      074010 104456          TRAP    C$ERRHRD
      074012 001446          .WORD  806
      074014 005107          .WORD  WRERR
      074016 012126          .WORD  PKTSSR
5298 074020          60$:  CKLOOP      ;LOOP IF SELECTED
      074020 104406          TRAP    C$CLP1
5299 074022 012737 000005 075762  MOV    #05.,T36DLY ;25-APR-83 REV B - DELAY FOR TAPE TO STOP
5300 074030          70$:  DELAY    1  ;25-APR-83 REV B - DELAY ROUTINE CALL
      074030 012727 000001  MOV    #1.(PC)
      074034 000000          .WORD  0
      074036 013727 002116  MOV    L$DLY,(PC)
      074042 000000          .WORD  0
      074044 005367 177772  DEC    -6(PC)
      074050 001375          BNE    -.4
      074052 005367 177756  DEC    -22(PC)
      074056 001367          BNE    -.20
5301 074060 005337 075762  DEC    T36DLY      ;BUMP COUNTER DOWN
5302 074064 001361 70$ BNE    70$         ;BR, IF MORE DELAY TO GO
5303 074066 012737 006642 075736  MOV    #3490.,T36SZ ;SET SIZE OF TRANSFER
5304 074074 012737 140005 075730  MOV    #140005,T36PK3 ;WRITE DATA,ACK,CVC-1 COMMAND
5305 074102 012704 075730  MOV    #T36PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
5306 074106 005037 075756  CLR    T36CNT      ;CLEAR COUNTER
5307 074112 012737 001750 075762  MOV    #1000.,T36DLY ;SET DROP DEAD COUNTER VALUE
5308 074120 010465 000000  MOV    R4,TSDB(R5) ;ISSUE COMMAND
5309 074124 016501 000002  MOV    TSSR(R5),R1 ;GET TSSR CONTENTS
5310 074130 032701 000200 80$:  BIT    #SSR,R1  ;CHECK FOR SSR SET
5311 074134 001021 90$ BNE    90$       ;BR, IF SSR IS SET
5312 074136 005237 075756  INC    T36CNT      ;BUMP CYCLE COUNTER
5313 074142          DELAY    1  ;CUT NUMBER OF LOOPS DOWN
      074142 012727 000001  MOV    #1.(PC)
      074146 000000          .WORD  0
      074150 013727 002116  MOV    L$DLY,(PC)
      074154 000000          .WORD  0
      074156 005367 177772  DEC    -6(PC)
      074162 001375          BNE    -.4
      074164 005367 177756  DEC    -22(PC)
      074170 001367          BNE    -.20
5314 074172 005337 075762  DEC    T36DLY      ;BUMP DROP DEAD COUNTER
5315 074176 001352 80$ BNE    80$       ;BR, IF THERE IS STILL TIME
5316 074200 012702 000200 90$:  MOV    #SSR,R2  ;SET UP EXPECTED
5317 074204 020102  CMP    R1,R2      ;ARE THEY EQUAL
5318 074206 001406  BEQ    100$      ;BR, IF OK
5319 074210 005237 002214  INC    FATFLG    ;ERROR COUNT

```


TEST 8: RECORD BUFFERING

```

5323 074214           ERRHRD  ERRNO,T36WDE,PKTSSR   ;TSSR INCORRECT AFTER READ DATA
      074214 104456                                      TRAP  C$ERHRD
      074216 001447                                      .WORD 807
      074220 076613                                      .WORD T36WDE
      074222 012126                                      .WORD  PKTSSR
5324 074224           100$:  CKLOOP               ;LOOP IF SELECTED
      074224 104406                                      TRAP  C$CLP1
5325 074226 013737 002174 075630   MOV    UNITN,T36DSW   ;SET UP DRIVE NUMBER
5326 074234 052737 000010 075630   BIS    #BIT3,T36DSW  ;25-APR-83 REV B - TURN OFF BUFFERING
5327 074242 012704 075610           MOV    #T36PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
5328 074246 004737 010742           JSR    PC,WRTCHR     ;ISSUE WRITE CHARACTERISTICS
5329 074252 103407                     BCS    110$          ;BR, IF COMMAND ISSUED OK
5330 074254 005237 002214           INC    FATFLG        ;ERROR COUNT
5334 074260 010001           MOV    R0,R1        ;SAVE CONTENTS OF TSSR
5335 074262           ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      074262 104456                                      TRAP  C$ERHRD
      074264 001450                                      .WORD 808
      074266 005052                                      .WORD WRTMSG
      074270 012114                                      .WORD  SFIMSG
5336 074272           110$:  CKLOOP               ;LOOP IF SELECTED
      074272 104406                                      TRAP  C$CLP1
5337 074274 012737 006642 075736   MOV    #3490.,T36SZ  ;SET SIZE OF TRANSFER
5338 074302 012737 140005 075730   MOV    #140005,T36PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
5339 074310 012704 075730           MOV    #T36PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
5340 074314 005037 075760           CLR    T36CNU       ;CLEAR COUNTER
5341 074320 012737 001750 075762   MOV    #1000.,T36DLY ;SET DROP DEAD COUNTER VALUE
5342 074326 010465 000000           MOV    R4,TSDB(R5)  ;ISSUE COMMAND
5343 074332 016501 000002 120$:  MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
5344 074336 032701 000200           BIT    #SSR,R1      ;CHECK FOR SSR SET
5345 074342 001021                     BNE    130$         ;BR, IF SSR IS SET
5346 074344 005237 075760           INC    T36CNU       ;BUMP CYCLE COUNTER
5347 074350           DELAY    1           ;CUT NUMBER OF LOOPS DOWN
      074350 012727 000001           MOV    #1,(PC)+    ;
      074354 000000           .WORD  0           ;
      074356 013727 002116           MOV    L$DLY,(PC)+
      074362 000000           .WORD  0           ;
      074364 005367 177772           DEC    -6(PC)      ;
      074370 001375                    BNE    -.4         ;
      074372 005367 177756           DEC    -22(PC)    ;
      074376 001367                    BNE    -.20        ;
5348 074400 005337 075762           DEC    T36DLY       ;BUMP DROP DEAD COUNTER
5349 074404 001352                     BNE    120$        ;BR, IF THERE IS STILL TIME
5350 074406 012702 000200 130$:  MOV    #SSR,R2      ;SET UP EXPECTED
5351 074412 020102           CMP    R1,R2        ;ARE THEY EQUAL
5352 074414 001406           BEQ    140$         ;BR, IF OK
5353 074416 005237 002214           INC    FATFLG        ;ERROR COUNT
5357 074422           ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      074422 104456                                      TRAP  C$ERHRD
      074424 001451                                      .WORD 809
      074426 005107                                      .WORD WRTERR
      074430 012126                                      .WORD  PKTSSR
5358 074432           140$:  CKLOOP               ;LOOP IF SELECTED
      074432 104406                                      TRAP  C$CLP1
5359 074434 013701 075756           MOV    T36CNT,R1    ;GET FIRST COUNTER
5360 074440 013702 075760           MOV    T36CNU,R2    ;GET SECOND COUNTER
5361 074444 020102           CMP    R1,R2        ;25-APR-83 REV B - COMPARE EM
5362 074446 003406           BLE    300$         ;BR, IF VALUES ARE CORRECT (OK)

```

TEST 8: RECORD BUFFERING

```

5363 074450 005237 002214          INC   FATFLG           ;ERROR COUNT
5367 074454          ERRHRD  ERRNO,T36NAS,EXPREC ;TAPE NOT AT CORRECT SPEED
      074454  104456          TRAP   C$ERHRD
      074456  001452          .WORD  810
      074460  075764          .WORD  T36NAS
      074462  015554          .WORD  EXPREC
5368 074464          300$:  CKLOOP             ;LOOP IF SELECTED
      074464  104406          TRAP   C$CLP1
5369 074466          ENDSUB
      074466          L10071:
      074466  104403          TRAP   C$ESUB
5370 074470 023727 002214 000017    CMP   FATFLG,#15.       ;IS ERROR COUNT AT 25
5371 074476 103402          BLO   999$              ;BR, IF LESS THAN 25
5372 074500 004737 017262          JSR   PC,CKDROP        ;TRY TO DROP THE UNIT
5373 074504          999$:
5374          ;*
5375          ;
5376          ;TEST 8, SUBTEST 2
5377          ;
5378          ;
5379          ;
5380          ;      THIS TEST VERIFIES THAT RECORD BUFFERING, USED FOR WRITE DATA
5381          ;      AND READ NEXT COMMANDS, OPERATES PROPERLY AND IS PROPERLY
5382          ;      CONTROLLED BY THE EXTENDED CHARACTERISTICS DATA WORD. IF THE
5383          ;      M7196 CONTROLLER MODULE IS NOT ALREADY IN EXTENDED FEATURES MODE
5384          ;      (AS CONTROLLED BY THE DIP SWITCH ON THE MODULE), IT IS PLACED
5385          ;      INTO THAT MODE BY INVERTING THE SENSE OF THE SWITCH USING THE
5386          ;      WRITE SUBSYSTEM MEMORY COMMAND. NOTE THAT RECORD BUFFERING HAS
5387          ;      BEEN ENABLED IN PREVIOUS TESTS OF READ AND WRITE AND SO HAS BEEN
5388          ;      PARTIALLY TESTED ALREADY. THIS TEST VERIFIES THAT BUFFERING IS
5389          ;      ACTUALLY OPERATING. THE FOLLOWING SUBTESTS ARE PERFORMED:
5390          ;
5391          ;      VERIFIES THAT NORMAL BUFFERING ON WRITE DATA COMMANDS OPERATES
5392          ;      PROPERLY AT LOW TAPE SPEED. THE FOLLOWING SEQUENCE IS
5393          ;      PERFORMED:
5394          ;
5395          ;      1. THE TAPE IS REWOUND.
5396          ;
5397          ;      2. BUFFERING IS DISABLED AND LOW TAPE SPEED IS SELECTED
5398          ;      (VIA WRITE CHARACTERISTICS COMMAND).
5399          ;
5400          ;      3. AN INITIAL RECORD IS WRITTEN ONTO THE TAPE IN ORDER TO
5401          ;      MOVE THE TAPE OFF BOT.
5402          ;
5403          ;      4. THE PROGRAM DELAYS FOR A TIME SUFFICIENT TO ALLOW THE
5404          ;      TAPE TO REPOSITION AND COME TO REST.
5405          ;
5406          ;      5. A WRITE DATA COMMAND, WITH A BYTE COUNT LESS THAN 3.5K,
5407          ;      IS ISSUED, AND THE PROGRAM COUNTS, IN A WAIT LOOP, THE
5408          ;      TIME IT TAKES TO RECEIVE COMMAND TERMINATION. THIS
5409          ;      SHOULD BE A RELATIVELY LONG TIME SINCE BUFFERING IS
5410          ;      DISABLED.
5411          ;
5412          ;      6. BUFFERING IS ENABLED.
5413          ;
5414          ;      7. THE WRITE DATA COMMAND IS AGAIN ISSUED, WITH THE SAME
5415          ;      BYTE COUNT AS THAT USED PREVIOUSLY. THE TIME TO
          ;      COMPLETION IS AGAIN MEASURED.
    
```

TEST 8: RECORD BUFFERING

5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428

- 8. THE COMPLETION TIMES MEASURED FOR THE NON-BUFFERED AND BUFFERED CASES ARE COMPARED. IT IS VERIFIED THAT THE TIME MEASURED FOR THE NON-BUFFERED CASE IS MUCH LARGER THAN THAT MEASURED FOR THE BUFFERED CASE.
- 9. THE PREVIOUS STEPS, EXCEPT FOR REWINDING AND WRITING A RECORD OFF BOT, ARE REPEATED FOR VARIOUS BYTE COUNTS IN THE RANGE 20 THROUGH 3.5K.

```

5429 074504       BGNSUB                                ;>>>>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>>>>
      074504                                     T8.2:
      074504 104402                                     TRAP      C$BSUB
5430 074506 004737 100770     JSR      PC,T36REST        ;SET COMMAND PACKET
5431 074512 004737 101062     JSR      PC,T36RT2        ;SET UP OTHER COMMAND PACKET
5432 074516 004737 101124     JSR      PC,T36RT3        ;SET UP OTHER COMMAND PACKET
5433 074522 012737 176750 075762  MOV      #65000.,T36DLY    ;SET UP DELAY COUNTER
5434 074530 005037 075756     CLR      T36CNT          ;CLEAR COUNTER
5435 074534 004737 016054 10$: JSR      PC,SOFINIT       ;DO INITIALIZE ON CONTROLLER
5436 074540 103426                BCS      20$             ;BR IF INIT WAS OK
5437 074542                DELAY      250                 ;DELAY ABOUT .25 SEC
      074542 012727 000250                MOV      #250,(PC).
      074546 000000                .WORD   0
      074550 013727 002116                MOV      L$DLY,(PC).
      074554 000000                .WORD   0
      074556 005367 177772                DEC      -6(PC)
      074562 001375                BNE     .-4
      074564 005367 177756                DEC      -22(PC)
      074570 001367                BNE     .-20
5438 074572 005337 075762     DEC      T36DLY          ;BUMP COUNTER
5439 074576 001356                BNE     10$             ;BR, IF COUNTER NOT DONE
5440 074600 005237 002214     INC      FATFLG          ;ERROR COUNT
5444 074604 010001     MOV      R0,R1           ;CONTENTS OF TSSR REGISTER
5445 074606                ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      074606 104455                TRAP      C$ERDF
      074610 001453                .WORD   811
      074612 003646                .WORD   SFIERR
      074614 012114                .WORD   SFIMSG
5446 074616 013737 002174 075630 20$: MOV      UNITN,T36DSW     ;SET UP DRIVE NUMBER
5447 074624 052737 000040 075630  BIS     #BITS,T36DSW     ;TURN ON HIGH SPEED
5448 074632 012704 075610     MOV     #T36PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
5449 074636 004737 010742     JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
5450 074642 103407                BCS     25$             ;BR, IF COMMAND ISSUED OK
5451 074644 005237 002214     INC     FATFLG          ;ERROR COUNT
5455 074650 010001     MOV     R0,R1           ;SAVE CONTENTS OF TSSR
5456 074652                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      074652 104456                TRAP     C$ERHRD
      074654 001454                .WORD   812
      074656 005052                .WORD   WRTMSG
      074660 012114                .WORD   SFIMSG
5457 074662                25$: CKLOOP                    ;LOOP IF SELECTED
      074662 104406                TRAP     C$CLP1
5458 074664 004737 011074     JSR     PC,REWIND       ;CALL TAPE REWIND COMMAND
5459 074670 103407                BCS     30$             ;BR, IF NO PROBLEM

```


TEST 8: RECORD BUFFERING

```

5546 075372 012704 075730      MOV   #T36PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
5547 075376 005037 075760      CLR   T36CNU         ;CLEAR COUNTER
5548 075402 012737 001750 075762      MOV   #1000.,T36DLY  ;SET DROP DEAD COUNTER VALUE
5549 075410 010465 000000      MOV   R4,T36DLY     ;ISSUE COMMAND
5550 075414 016501 000002      MOV   TSSR(R5),R1   ;GET TSSR CONTENTS
5551 075420 032701 000200      BIT   #SSR,R1       ;CHECK FOR SSR SET
5552 075424 001021              BNE   130$          ;BR, IF SSR IS SET
5553 075426 005237 075760      INC   T36CNUJ       ;BUMP CYCLE COUNTER
5554 075432              DELAY 1             ;CUT NUMBER OF LOOPS DOWN
                    MOV   #1,(PC)+
                    .WORD 0
                    MOV   L$DLY,(PC)+
                    .WORD 0
                    DEC   -6(PC)
                    BNE   -4
                    DEC   -22(PC)
                    BNE   -20
5555 075462 005337 075762      DEC   T36DLY        ;BUMP DROP DEAD COUNTER
5556 075466 001352              BNE   120$          ;BR, IF THERE IS STILL TIME
5557 075470 012702 000200      MOV   #SSR,R2       ;SET UP EXPECTED
5558 075474 020102              CMP   R1,R2         ;ARE THEY EQUAL
5559 075476 001406              BEQ   140$          ;BR, IF OK
5560 075500 005237 002214      INC   FATFLG        ;ERROR COUNT
5564 075504      ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                    TRAP C$ERRHRD
                    .WORD 819
                    .WORD WRERR
                    .WORD PKTSSR
                    TRAP C$CLP1
5565 075514      140$: CKLOOP          ;LOOP IF SELECTED
                    TRAP C$CLP1
                    MOV   T36CNT,R1      ;GET FIRST COUNTER
5566 075516 013701 075756      MOV   T36CNU,R2     ;GET SECOND COUNTER
5567 075522 013702 075760      CMP   R1,R2         ;25-APR-83 REV B - COMPARE EM
5568 075526 020102              BLE   300$          ;BR, IF VALUES ARE CORRECT (OK)
5569 075530 003406              INC   FATFLG        ;ERROR COUNT
5570 075532 005237 002214      ERRHRD ERRNO,T36NAS,EXPREC ;TAPE NOT AT CORRECT SPEED
                    TRAP C$ERRHRD
                    .WORD 820
                    .WORD T36NAS
                    .WORD EXPREC
5574 075536      300$: CKLOOP          ;LOOP IF SELECTED
                    TRAP C$CLP1
5575 075546      300$: CKLOOP          ;LOOP IF SELECTED
                    TRAP C$CLP1
5576 075550      075550: ENDSUB
                    .WORD 0
                    L10072: TRAP C$ESUB
5577 075552 023727 002214 000017      CMP   FATFLG,#15.  ;IS ERROR COUNT AT 25
5578 075560 103402              BLO   999$          ;BR, IF LESS THAN 25
5579 075562 004737 017262              JSR   PC,CKDROP    ;TRY TO DROP THE UNIT
5580 075566      999$:
5581          ;
5582          ;
5583          ;
5584 075566 004737 016536      JSR   PC,TSTLOOP   ;DO WE NEED TO ITERATE TEST
5585 075572 103002              BCC   163$          ;BR, IF NO LOOP REQUIRED
5586 075574 000137 073430      JMP   T36LOOP      ;EXECUTE AGAIN
5587 075600      163$:
5588 075600      EXIT   TST       ;ALL DONE THIS TEST

```

TEST 8: RECORD BUFFERING

TRAP C\$EXIT
.WORD L10070-

075600 104432
075602 003344
5589
5590
5591
5593 075610
5595 075610
5596 075610 100004
5597 075612 075620
5598 075614 000000
5599 075616 000012
5600 075620
5601 075620 075632
5602 075622 000000
5603 075624 000024
5604 075626 000000
5605 075630 000000
5606 075632
5607
5608
5609
5611 075720
5613 075720
5614 075720 100006
5615 075722 075740
5616 075724 000000
5617 075726 000006
5618
5622 075730
5623 075730 100005
5624 075732
5625 075732 003116
5626 075734 000000
5627 075736 000000
5628
5629
5630
5631
5632 075740
5633 075740 010
5634 075741 200
5635 075742 000000
5636 075744 000000
5637
5638
5639
5640
5641
5642 075746 100205
5643 075750 100605
5644 075752 102205
5645 075754 177777
5646
5647
5648 075756 000000
5649 075760 000000
5650 075762 000000

```

;+
;LOCAL STORAGE FOR THIS TEST
;-
          .=<+.10>E177770
T36PACKET:
        .WORD 100004
        .WORD T36DATA
        .WORD 0
        .WORD 10.
T36DATA:
        .WORD T36BFR
        .WORD 0
        .WORD 20.
        .WORD 0
T36DSW: .WORD 0
T36BFR: .BLKW 25.
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
          .=<+.10>E177770
T36PK2:
        .WORD 100006
        .WORD T36BF2
        .WORD 0
        .WORD 6.
T36PK3:
        .WORD 100005
T36RB:
T36WB:  .WORD FREE
        .WORD 0
T36SZ:  .WORD 0
        .EVEN
;
;
T36BF2:
T36BS0: .BYTE 10
T36BS1: .BYTE 200
T36S2:  .WORD 0
T36S3:  .WORD 0
;
;
          .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T36RN:  .WORD 100205
T36WDR: .WORD 100605
T36CON: .WORD 102205
        .WORD 177777
;
T36CNT: .WORD 0
T36CNU: .WORD 0
T36DLY: .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)

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

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

;TAPE TIMER COUNTER STORAGE AREA
;TAPE TIMER COUNTER STORAGE AREA
;DELAY COUNTER
    
```

TEST 8: RECORD BUFFERING

```

5651
5652 ;+
5653 ;LOCAL TEXT MESSAGES FOR TEST
5654 ;-
5655 075764 111 155 160 T36NAS: .ASCIZ 'Improper Tape Controller Buffering Speed'
5656 076035 124 141 160 T36WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
5657 076123 124 123 123 T36RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
5658 076172 122 105 122 T36RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
5659 076267 120 117 123 T36SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
5660 076351 122 111 102 T36LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
5661 076421 124 123 123 T36WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
5662 076476 111 154 154 T36LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
5663 076557 122 105 122 T36SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
5664 076613 124 123 123 T36WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
5665 076665 124 141 160 T36BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XSTO)'
5666 076760 127 122 111 T36TIM: .ASCIZ 'WRITE DATA RETRY''S Erase Tape Not Long Enough'
5667 077035 122 105 122 T36EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
5668 077114 124 123 123 T36TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
5669 077171 122 145 167 T36RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
5670 077240 122 101 115 T36RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
5671 077313 124 123 123 T36AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
5672 077362 104 162 151 T36OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
5673 077435 124 123 123 T36WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
5674 077525 124 123 123 T36WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
5675 077600 103 126 103 T36VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
5676 077653 124 123 102 T36BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
5677 077726 127 122 111 T36WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
5678 100015 122 145 141 T36LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XSTO'
5679 100077 122 145 141 T36LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XSTO'
5680 100161 122 145 163 T36PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
5681 100247 122 145 141 T36TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
5682 100335 127 122 111 T36NEF: .ASCIZ 'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
5683 100433 124 123 123 T36SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
5684 100510 124 123 123 T36TSA: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
5685 100572 124 123 123 T36WRF: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command'
5686 100652 104 141 164 T36DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
5687 100747 122 145 143 T36ID: .ASCIZ 'Record Buffering'
5688 .EVEN
5689 ;+
5690 ;
5691 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5692 ;WRITE SUBSYSTEM MEMORY COMMAND
5693 ;
5694 ;-
5695
5696 100770 T36REST:
5697 100770 SAVREG
5698 100774 012701 075610 MOV #T36PACKET,R1 ;SAVE THE REGISTERS
5699 101000 012721 100004 MOV #100004,(R1)+ ;START OF THE PACKET
5700 101004 012721 075620 MOV #T36DATA,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK.
5701 101010 005021 CLR (R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
5702 101012 012721 000012 MOV #10,(R1)+ ;EXTENDED ADDRESS
5703 101016 012721 075632 MOV #T36BFR,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
5704 101022 005021 CLR (R1)+ ;ADDRESS OF MESSAGE BUFFER
5705 101024 012721 000024 MOV #20,(R1)+ ;LENGTH OF MESSAGE BUFFER
5706 101030 005021 CLR (R1)+
5707 101032 012711 000000 MOV #0,(R1) ;SELECT DRIVE ZERO

```


TEST 8: RECORD BUFFERING

```

5708 101036 012702 000030          MOV      #24.,R2          ;NUMBER OF LOCATIONS TO BE CLEARED
5709 101042 012762 177777 075632 64$: MOV      #177777,T36BFR(R2) ;ALL ONES TO MESSAGE BUFFER
5710 101050 005742                TST      -(R2)          ;NEXT LOCATION
5711 101052 022702 000000          CMP      #0,R2          ;AT END OF LOOP YET
5712 101056 001371                BNE      64$           ;KEEP GOING UNTIL DONE
5713 101060 000207                RTS      PC             ;RETURN
5714
5715 101062                T36RT2:
5716 101062                SAVREG          ;SAVE THE REGISTERS
5717 101066 012701 075720          MOV      #T36PK2,R1    ;START OF THE PACKET
5718 101072 012721 100006          MOV      #100006,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
5719 101076 012721 075740          MOV      #T36BF2,(R1)+ ;ADDRESS OF DATA BLOCK
5720 101102 005021                CLR      (R1)+         ;EXTENDED ADDRESS
5721 101104 012721 000006          MOV      #6.,(R1)+     ;SIZE OF DATA BLOCK IN BYTES
5722 101110 005021                CLR      (R1)+
5723 101112 012701 075740          MOV      #T36BF2,R1    ;POINT TO DATA SEL AREA
5724 101116 005021                CLR      (R1)+
5725 101120 005011                CLR      (R1)
5726 101122 000207                RTS      PC             ;RETURN
5727 101124                T36RT3:
5728 101124                SAVREG          ;SAVE REGISTERS
5729 101130 012701 075730          MOV      #T36PK3,R1    ;SET UP POINTER ADDRESS
5730 101134 005021                CLR      (R1)+         ;COMMAND SPACE
5731 101136 005021                CLR      (R1)+         ;ADDRESS OF DATA BLOCK
5732 101140 005021                CLR      (R1)+         ;EXTENDED ADDRESS
5733 101142 005011                CLR      (R1)          ;SIZE OF DATA TRANSFER BLOCK
5734 101144 000207                RTS      PC             ;RETURN
5735 101146
101146                                L10070: TRAP      C$ETST
101146 104401
5736                                .SBTTL TEST 9: FUNCTION TIMING
5737                                ;+
5738                                ;
5739                                ;THIS TEST VERIFIES THAT THE TAPE TRANSPORT SEEMS TO BE WRITING
5740                                ;RECORDS, GAPS, AND EXTENDED GAPS OF THE PROPER LENGTH. BOTH LOW
5741                                ;AND HIGH SPEED MODES ARE TESTED. IT IS ALSO VERIFIED THAT A
5742                                ;SPACE RECORDS COMMAND WITH A RECORD COUNT OF 80 OR MORE, AND A
5743                                ;SKIP TAPE MARKS COMMAND WITH A COUNT OF 2 OF MORE, OPERATE THE
5744                                ;TAPE IN HIGH-SPEED MODE. THIS TEST CAN ONLY BE RUN IF A
5745                                ;REAL-TIME CLOCK IS AVAILIABLE ON THE SYSTEM. THE TEST OPERATES BY
5746                                ;TIMING VARIOUS TAPE-MOTION OPERATIONS, USING A NUMBER OF
5747                                ;DIFFERENT TEST RECORD LENGTHS.
5748                                ;
5749                                ;
5750                                ;-
5751 101150                BGNTST
101150
5752 101150 012737 006354 002172          MOV      #EPRT1,EPRTSW ;PRIMARY ERROR MESSAGE
5753 101156 004737 017354                JSR      PC,KTOFF      ;TURN KT OFF
5758 101162 012700 105373                MOV      #TST37ID,R0   ;ASCII MESSAGE TO IDENTIFY TEST
5759 101166 004737 016570                JSR      PC,TSTSETUP   ;DO INITIAL TEST SETUP
5760 101172 012737 000005 002210          MOV      #5,LOOPCNT    ;PERFORM 5 ITERATIONS
5761 101200 005037 102436                CLR      T37CNT        ;CLEAR TAPE RECORD COUNTER
5762                                ;+
5763                                ;
5764                                ;TEST 9, SUBTEST 1
5765                                ;

```


TEST 9: FUNCTION TIMING

	101410	001607							.WORD	903
	101412	103615							.WORD	T37RWN
	101414	012126							.WORD	PKTSSR
5812	101416			30%:	CKLOOP			;LOOP IF SELECTED		
	101416	104406							TRAP	C\$CLP1
5813	101420	013701	102320		MOV	T37BFR+6,R1		;PICK UP XSTO		
5814	101424	010102			MOV	R1,R2		;SET UP EXPECTED		
5815	101426	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED		
5816	101432	020102			CMP	R1,R2		;DOES EXP = REC'D		
5817	101434	001406			BEQ	40\$;BR, IF EQUAL (OK)		
5818	101436	005237	002214		INC	FATFLG		;ERROR COUNT		
5822	101442				ERRHRD	ERRNO,T37BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND		
	101442	104456							TRAP	C\$ERHRD
	101444	001610							.WORD	904
	101446	103311							.WORD	T37BOT
	101450	015554							.WORD	EXPREC
5823	101452			40%:	CKLOOP			;LOOP IF SELECTED		
	101452	104406							TRAP	C\$CLP1
5824	101454	012703	000144		MOV	#100.,R3		;NUMBER OF RECORDS TO BE WRITTEN		
5825	101460	013737	003116	102412	MOV	FREE,T37WB		;STARTING WRITE BUFFER ADDRESS		
5826	101466	012737	140005	102410	65%:	MOV	#140005,T37PK3	;WRITE DATA,ACK,CVC=1 COMMAND		
5827	101474	012704	102410		MOV	#T37PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
5828	101500	012737	001130	102416	MOV	#600.,T37SZ		;SET UP RECORD SIZE IN PACKET		
5829	101506	010465	000000		MOV	R4,TSD8(R5)		;ISSUE COMMAND		
5830	101512	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET		
5831	101516	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
5832	101522	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED		
5833	101526	020102			CMP	R1,R2		;ARE THEY EQUAL		
5834	101530	001406			BEQ	70\$;BR, IF OK		
5835	101532	005237	002214		INC	FATFLG		;ERROR COUNT		
5839	101536				ERRHRD	ERRNO,T37WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA		
	101536	104456							TRAP	C\$ERHRD
	101540	001611							.WORD	905
	101542	104151							.WORD	T37WDC
	101544	012126							.WORD	PKTSSR
5840	101546			70%:	CKLOOP			;LOOP IF SELECTED		
	101546	104406							TRAP	C\$CLP1
5841	101550	005303			DEC	R3		;DEC RECORD COUNTER		
5842	101552	001345			BNE	65\$;BR, IF MORE RECORDS TO WRITE		
5843	101554	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND		
5844	101560	103411			BCS	130\$;BR, IF NO PROBLEM		
5845	101562	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
5846	101566	010004			MOV	R0,R4		;GET PACKET ADDRESS		
5847	101570	005237	002214		INC	FATFLG		;FRROR COUNT		
5851	101574				ERRHRD	ERRNO,T37RWN,PKTSSR		;REWIND NOT ACCEPTED		
	101574	104456							TRAP	C\$ERHRD
	101576	001612							.WORD	906
	101600	103615							.WORD	T37RWN
	101602	012126							.WORD	PKTSSR
5852	101604			130%:	CKLOOP			;LOOP IF SELECTED		
	101604	104406							TRAP	C\$CLP1
5853	101606	013701	102320		MOV	T37BFR+6,R1		;PICK UP XSTO		
5854	101612	010102			MOV	R1,R2		;SET UP EXPECTED		
5855	101614	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED		
5856	101620	020102			CMP	R1,R2		;DOES EXP = REC'D		
5857	101622	001406			BEQ	140\$;BR, IF EQUAL (OK)		
5858	101624	005237	002214		INC	FATFLG		;ERROR COUNT		

TEST 9: FUNCTION TIMING

5862	101630				ERRHRD	ERRNO,T37BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND		
	101630	104456						TRAP	C\$ERHRD	
	101632	001613						.WORD	907	
	101634	103311						.WORD	T37BOT	
	101636	015554						.WORD	EXPREC	
5863	101640			140\$:	CKLOOP			;LOOP IF SELECTED		
	101640	104406						TRAP	C\$CLP1	
5864	101642	012704	102410		MOV	#T37PK3,R4		;SET UP PACKET ADDRESS		
5865	101646	012737	000037	102412	MOV	#31.,T37RB		;SET UP RECORDS TO SPACE OVER		
5866	101654	012737	140010	102410	MOV	#140010,T37PK3		;ACK,CVC=1,SPACE FORWARD COMMAND		
5867	101662	010465	000000		150\$:	MOV	R4,T5DB(R5)	;ISSUE COMMAND		
5868	101666	005237	102436		152\$:	INC	T37CNT	;BUMP TIMER		
5869	101672				DELAY	1		;DELAY ABOUT 100US		
	101672	012727	000001					MOV	#1,(PC)+	
	101676	000000						.WORD	0	
	101700	013727	002116					MOV	L\$DL1,(PC)+	
	101704	006000						.WORD	0	
	101706	005367	177772					DEC	-6(PC)	
	101712	001375						BNE	.-4	
	101714	005367	177756					DEC	-22(PC)	
	101720	001367						BNE	.-20	
5870	101722	016501	000002		MOV	T5SR(R5),R1		;GET T5SR		
5871	101726	032701	000200		BIT	#5SR,R1		;CHECK FOR T5SR'S 5SR SET		
5872	101732	001755			BEQ	152\$;KEEP COUNTING UNTIL SET		
5873	101734	012702	000200		MOV	#5SR,R2		;SET UP EXPECTED		
5874	101740	020201			CMP	R2,R1		;WAS EVERYTHING OK		
5875	101742	001406			BEQ	160\$;BR, IF ALL IS WELL		
5876	101744	005237	002214		INC	FATFLG		;ERROR COUNT		
5880	101750				ERRHRD	ERRNO,T37SCF,PKT5SR		;SPACE FORWARD DIDN'T WORK OUT		
	101750	104456						TRAP	C\$ERHRD	
	101752	001614						.WORD	908	
	101754	105057						.WORD	T37SCF	
	101756	012126						.WORD	PKT5SR	
5881	101760			160\$:	CKLOOP			;LOOP IF SELECTED		
	101760	104406						TRAP	C\$CLP1	
5882	101762	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND		
5883	101766	103411			BCS	170\$;BR, IF NO PROBLEM		
5884	101770	010004			MOV	R0,R4		;GET PACKET ADDRESS		
5885	101772	016501	000002		MOV	T5SR(R5),R1		;GET STATUS FROM T5SR		
5886	101776	005237	002214		INC	FATFLG		;ERROR COUNT		
5890	102002				ERRHRD	ERRNO,T37RWN,PKT5SR		;REWIND NOT ACCEPTED		
	102002	104456						TRAP	C\$ERHRD	
	102004	001615						.WORD	909	
	102006	103615						.WORD	T37RWN	
	102010	012126						.WORD	PKT5SR	
5891	102012			170\$:	CKLOOP			;LOOP IF SELECTED		
	102012	104406						TRAP	C\$CLP1	
5892	102014	013701	102320		MOV	T37BFR+6,R1		;PICK UP XST0		
5893	102020	010102			MOV	R1,R2		;SET UP EXPECTED		
5894	102022	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED		
5895	102026	020102			CMP	R1,R2		;DOES EXP = REC'D		
5896	102030	001406			BEQ	175\$;BR, IF EQUAL (OK)		
5897	102032	005237	002214		INC	FATFLG		;ERROR COUNT		
5901	102036				ERRHRD	ERRNO,T37BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND		
	102036	104456						TRAP	C\$ERHRD	
	102040	001616						.WORD	910	
	102042	103311						.WORD	T37BOT	

TEST 9: FUNCTION TIMING

```

5943 102256 000137 101204      JMP      T37LOOP      ;EXECUTE AGAIN
5944 102262      163$:      EXIT      TST      ;ALL DONE THIS TEST
5945 102262      104432      TRAP      C$EXIT
      102264      003306      .WORD     L10073-.
5946
5947      ;*
5948      ;LOCAL STORAGE FOR THIS TEST
5949      ;-
5950      102270      .=<..+10>&177770
5952 102270      T37PACKET:          ;COMMAND PACKET FOR TEST
      100004      .WORD     100004      ;WRITE CHARACTERISTICS COMMAND, WITH , ACK
      102300      .WORD     T37DATA      ;ADDRESS OF CHARACTERISTICS BLOCK
      000000      .WORD     0
      000012      .WORD     10.      ;STARTING VALUE OF BLOCK SIZE
5957 102300      T37DATA:          ;CHARACTERISTICS DATA BLOCK
      102312      .WORD     T37BFR      ;ADDRESS OF MESSAGE BUFFER
      000000      .WORD     0
      000024      .WORD     20.      ;LENGTH OF MESSAGE BUFFER
5961 102306      .WORD     0
5962 102310      T37DSW: .WORD     0      ;SELECT DRIVE 0
5963 102312      T37BFR: .BLKW    25.      ;MESSAGE BUFFER
5964
5965      ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
5966      ;
5968      102400      .=<..+10>&177770
5970 102400      T37PK2:          ;WRITE SUB SYS MEM COMMAND, AND ACK
      100006      .WORD     100006      ;ADDRESS OF SELECT BLOCK DATA
      102420      .WORD     T37BF2
      000000      .WORD     0
      000006      .WORD     6.      ;SIZE OF DATA PACKET
5975
5979 102410      T37PK3:          ;REREAD COMMAND, AND ACK
      100005      .WORD     100005
      102412      T37RB:          ;ADDRESS OF WRITE BUFFER
      003116      T37WB: .WORD     FREE
      000000      .WORD     0
      000000      T37SZ: .WORD     0      ;SIZE OF BUFFER (EXTENT)
      .EVEN
5987
5988
5989 102420      T37BF2:
      102420      T37BS0: .BYTE     10      ;BSELO AREA
      102421      T37BS1: .BYTE     200      ;BSEL1 AREA
      000000      T37S2: .WORD     0      ;SEL 2 AREA
      000000      T37S3: .WORD     0      ;DATA AREA
5994
5995
5996      .EVEN
5997      ;TAPE MOTION PACKET COMMAND VALUES
5998
5999 102426      T37RN: .WORD     100205      ;REREAD DATA (NEXT)
6000 102430      T37WDR: .WORD     100605      ;REREAD DATA RETRY
6001 102432      T37CON: .WORD     102205      ;WRITE CONTINOUS
6002 102434      .WORD     177777      ;END OF DATA
6003
6004

```

TEST 9: FUNCTION TIMING

```

6005 102436 000000      T37CNT: .WORD 0      ;TAPE TIMER COUNTER STORAGE AREA
6006 102440 000000      T37CNU: .WORD 0      ;TAPE TIMER COUNTER STORAGE AREA
6007 102442 000000      T37DLY: .WORD 0      ;DELAY COUNTER
6008                      ;*
6009                      ;LOCAL TEXT MESSAGES FOR TEST
6010                      ;-
6011
6012 102444      124      141      160 T37WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
6013 102532      124      123      123 T37RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
6014 102601      122      105      122 T37RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
6015 102676      120      117      123 T37SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
6016 102760      122      111      102 T37LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
6017 103030      124      123      123 T37WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
6018 103105      111      154      154 T37LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
6019 103166      122      105      122 T37SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
6020 103222      124      123      123 T37WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command, At BOT'
6021 103311      124      141      160 T37BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
6022 103404      127      122      111 T37TIM: .ASCIZ 'WRITE DATA RETRY'S Erase Tape Not Long Enough'
6023 103461      122      105      122 T37EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
6024 103540      124      123      123 T37TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
6025 103615      122      145      167 T37RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
6026 103664      122      101      115 T37RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
6027 103737      124      123      123 T37AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
6028 104006      104      162      151 T37OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
6029 104061      124      123      123 T37WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
6030 104151      124      123      123 T37WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
6031 104224      103      126      103 T37VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
6032 104277      124      123      102 T37BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
6033 104352      127      122      111 T37WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
6034 104441      122      145      141 T37LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
6035 104523      122      145      141 T37LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
6036 104605      122      145      163 T37PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
6037 104673      122      145      141 T37TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
6038 104761      127      122      111 T37NEF: .ASCIZ 'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
6039 105057      124      123      123 T37SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
6040 105134      124      123      123 T37TSA: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
6041 105216      124      123      123 T37WRF: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command'
6042 105276      104      141      164 T37DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
6043 105373      106      165      156 TST37ID: .ASCIZ 'Function Timing'
6044                      .EVEN
6045                      ;*
6046                      ;
6047                      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
6048                      ;WRITE SUBSYSTEM MEMORY COMMAND
6049                      ;
6050                      ;-
6051
6052 105414      T37REST:
6053 105414      SAVREG
6054 105420      012701 102270      MOV #T37PACKET,R1      ;SAVE THE REGISTERS
6055 105424      012721 100004      MOV #100004,(R1)+      ;START OF THE PACKET
6056 105430      012721 102300      MOV #T37DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,
6057 105434      005021      CLR (R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
6058 105436      012721 000012      MOV #10.,(R1)+      ;EXTENDED ADDRESS
6059 105442      012721 102312      MOV #T37BFR,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
6060 105446      005021      CLR (R1)+      ;ADDRESS OF MESSAGE BUFFER
6061 105450      012721 000024      MOV #20.,(R1)+      ;LENGTH OF MESSAGE BUFFER

```

TEST 9: FUNCTION TIMING

```

6062 105454 005021          CLR      (R1)+
6063 105456 012711 000000    MOV      #0,(R1)          ;SELECT DRIVE ZERO
6064 105462 012702 000030    MOV      #24.,R2         ;NUMBER OF LOCATIONS TO BE CLEARED
6065 105466 012762 177777 102312 64$: MOV      #177777,T37BFR(R2) ;ALL ONES TO MESSAGE BUFFER
6066 105474 005742          TST      -(R2)           ;NEXT LOCATION
6067 105476 022702 000000    CMP      #0,R2           ;AT END OF LOOP YET
6068 105502 001371          BNE      64$             ;KEEP GOING UNTIL DONE
6069 105504 000207          RTS      PC              ;RETURN
6070
6071 105506          T37RT2:
6072 105506          SAVREG          ;SAVE THE REGISTERS
6073 105512 012701 102400    MOV      #T37PK2,R1      ;START OF THE PACKET
6074 105516 012721 100006    MOV      #100006,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK.
6075 105522 012721 102420    MOV      #T37BF2,(R1)+  ;ADDRESS OF DATA BLOCK
6076 105526 005021          CLR      (R1)+           ;EXTENDED ADDRESS
6077 105530 012721 000006    MOV      #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
6078 105534 005021          CLR      (R1)+
6079 105536 012701 102420    MOV      #T37BF2,R1      ;POINT TO DATA SEL AREA
6080 105542 005021          CLR      (R1)+
6081 105544 005011          CLR      (R1)
6082 105546 000207          RTS      PC              ;RETURN
6083 105550          T37RT3:
6084 105550          SAVREG          ;SAVE REGISTERS
6085 105554 012701 102410    MOV      #T37PK3,R1      ;SET UP POINTER ADDRESS
6086 105560 005021          CLR      (R1)+           ;COMMAND SPACE
6087 105562 005021          CLR      (R1)+           ;ADDRESS OF DATA BLOCK
6088 105564 005021          CLR      (R1)+           ;EXTENDED ADDRESS
6089 105566 005011          CLR      (R1)           ;SIZE OF DATA TRANSFER BLOCK
6090 105570 000207          RTS      PC              ;RETURN
6091 105572          ENDTST
        105572
        105572 104401
6092 105574          ENDMOD

```

L10073: TRAP C\$ETST

TEST 9: FUNCTION TIMING

```

1          .TITLE  TSV6 - PARAMETER CODING
7
12
18
19 105574   BGNMOD  TSV6
105574   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 105574   BGNHRD
105574   .WORD  L10075-L$HARD/2
105576   L$HARD::
32
33 105576   GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
105576   .WORD   T$CODE
105600   .WORD   HPM1
105602   .WORD   T$LOLIM
105604   .WORD   T$HILIM
34 105606   GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
105606   .WORD   T$CODE
105610   .WORD   HPM2
105612   .WORD   T$LOLIM
105614   .WORD   T$HILIM
35          ;GPRMD  HPM3,4,0,340,0,7,YES          ;GET INTERRUPT PRIORITY.
36 105616   ENDHRD
          .EVEN
          L10075:
37 105616   104     105     126   HPM1:  .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
38 105652   111     116     124   HPM2:  .ASCIZ  'INTERRUPT VECTOR '
39 105676   111     116     124   HPM3:  .ASCIZ  'INTERRUPT PRIORITY '
40          .EVEN
    
```

SOFTWARE PARAMETER CODING SECTION

```

42                                     .SBTTL SOFTWARE PARAMETER CODING SECTION
43
44                                     : **
45                                     : THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
46                                     : THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
47                                     : MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
48                                     : INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
49                                     : MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
50                                     : WITH THE OPERATOR.
51                                     : **
52 105726                                BGNSFT
105726 000003                          .WORD L10076-L$SOFT/2
105730
53                                     L$SOFT::
54 105730                                : GPRML SPM1,0,-1,YES ; GET TRANSPORT TEST FLAG.
105732 001130                          : GPRML SPM4,2,-1,YES ; GET ITERATION CONTROL.
105734 105766                          .WORD T$CODE
105734 177777                          .WORD SPM4
55                                     : GPRMD SPM6,4,D,7777,0,7777,YES ; GET LOCAL ERROR LIMIT
56                                     : GPRMD SPM7,6,D,7777,0,7777,YES ; GET GLOBAL ERROR LIMIT
57 105736                                ENDSFT
105736                                .EVEN
58                                     L10076:
59 105736                                105 116 101 SPM1: .ASCIZ 'ENABLE TRANSPORT TESTS '
60 105766                                111 116 110 SPM4: .ASCIZ 'INHIBIT ITERATIONS '
61 106016                                120 105 122 SPM6: .ASCIZ 'PER TEST ERROR LIMIT '
62 106046                                120 105 122 SPM7: .ASCIZ 'PER UNIT ERROR LIMIT '
63                                     .SBTTL PATCH AREA
64
65                                     ;
66                                     ; FINALLY A GENEROUS PATCH AREA.
67                                     ;
68                                     ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
69                                     ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
70                                     ;
71                                     ;
72 106076                                PATCH::
73                                     .BLKW 32.
74 106076
75                                     . = !377*1
77 106400 106400                          LASTAD ; SET LAST USED ADDRESS.
79 106400                                .EVEN
106400 000000                          .WORD 0
106402 000000                          .WORD 0
106404
80 106404                                L$LAST::
81 000001                                ENDMOD
                                         .END

```


SYMBOL TABLE

T37CNT 102436	T37SSR 103166	T7.4 066474	WSMBK 021350 G	X\$OFFS= 000400
T37CNU 102440	T37SZ 102416	T8 073374 G	XFERAS 016020	X\$TRUE= 000020
T37CON 102432	T37S2 102422	T8.1 073430	XNXM 016456	X1.COR= 020000
T37DAT 102300	T37S3 102424	T8.2 074504	XORBFO 007752	X1.DLT= 100000
T37DLY 102442	T37TIM 103404	T9 101150 G	XORFOR 010070	X1.MBZ= 017375
T37DSW 102310	T37TM 103540	T9.1 101204	XST0 = 000006 G	X1.RBP= 000400
T37DTA 105276	T37TRL 104673	UAM = 000200 G	XST1 = 000010 G	X1.SPA= 040000
T37EQT 103461	T37TSA 105134	UNITN 002174 G	XST2 = 000012 G	X1.UNC= 000002
T37LON 104441	T37VCK 104224	UNREC = 000006	XST3 = 000014 G	X2.BUF= 000100
T37LOO 101204	T37WB 102412	USI 004117	XST4 = 000016 G	X2.EXT= 000200
T37LOP 104523	T37WDC 104151	WAITF 016330 G	XSOBOT= 000002	X2.OPM= 100000
T37LOQ 103105	T37WDD 104061	WC.IFA= 000200	XSOEOT= 000001	X2.RCE= 040000
T37LOR 102760	T37WDE 103222	WC.IFE= 000002	XSOIE = 000040	X2.REV= 000077
T37NEF 104761	T37WDF 103030	WC.IGO= 000001	XSOILA= 000400	X2.SPA= 035400
T37OFL 104006	T37WDR 102430	WC.IRE= 000010	XSOILC= 001000	X2.UNI= 000007
T37PAC 102270	T37WNG 102444	WC.IRW= 000004	XSOLET= 020000	X2.WCF= 002000
T37PBP 104605	T37WRF 105216	WC.IOT= 000100	XSOMOT= 000200	X3.DCK= 000010
T37PK2 102400	T37WSS 104352	WC.IIT= 000040	XSONEF= 002000	X3.MBZ= 000006
T37PK3 102410	T4 047020 G	WC.ISR= 000020	XSOONL= 000100	X3.MDE= 177400
T37RB 102412	T4.1 047050	WF.IED= 000010	XSOPED= 000010	X3.OPI= 000100
T37RDF 102532	T4.2 047710	WF.IER= 000004	XSORLL= 010000	X3.REV= 000040
T37RES 105414	T4.3 050520	WF.IHI= 000200	XSORLS= 040000	X3.RIB= 000001
T37RN 102426	T5 053076 G	WF.IRE= 000040	XSOTMK= 100000	X3.SPA= 000200
T37RNC 103664	T5.1 053126	WF.IWF= 000020	XSOVCK= 000020	X3.TRF= 000020
T37RRF 102601	T6 056072 G	WF.IWR= 000100	XSOWLE= 004000	X4.HSP= 100000
T37RT2 105506	T6.1 056122	WF.I3R= 000002	XSOWLK= 000004	X4.MBZ= 017400
T37RT3 105550	T7 063444 G	WF.I4R= 000001	XXCOMM 003114 G	X4.RCE= 040000
T37RWN 103615	T7.1 063474	WRTCHR 010742 G	X\$ALWA= 000000	X4.TSM= 020000
T37SC 102676	T7.2 064552	WRTERR 005107	X\$FALS= 000040	X4.WRC= 000377
T37SCF 105057	T7.3 065632	WRTMSG 005052		

. ABS. 106404 000
 000000 001
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 31520 WORDS (124 PAGES)
 DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
 ELAPSED TIME: 00:08:39
 CNTSDAO.BIC,CNTSDAO.SEQ/-SP=SVC34/ML,TSV1D,TSV22D,TSV3B,TSV4,TSV7B,TSV6