

TSV05

TSV05 CTRL LT4
CVTSDA0

AH-T100A-MC
FICHE 1 OF 2

SEP 1982
COPYRIGHT © 1982
MADE IN USA



TSV05

TSV05 CTRL LT4
CVTSDA0

AH-T100A-MC
FICHE 2 OF 2

SEP 1982
COPYRIGHT© 1982
MADE IN USA



.REM_
IDENTIFICATION

PRODUCT ID: AC-T099A-MC
PRODUCT TITLE: CVTSDAO TSV05 CTRL LT4
AUTHOR: DICK GORDON
MAINTAINER: SCOTT SNOWDON
DATE: MARCH 08, 1982

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

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

COPYRIGHT (C) 1982,1982 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

TABLE OF CONTENTS

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

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

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

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

1.2 SYSTEM REQUIREMENTS

PDP-11/23 PROCESSOR AND MEMORY
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE AND 4K RESERVED FOR I/O PAGE)
TSV05 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)
CONSOLE TERMINAL
PDP-11 DIAGNOSTIC SUPERVISOR (HSAAA.SYS VERSION 34 OR LATER)
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. CHQUS XXDP+ USERS MANUAL; DOCUMENT NUMBER AC-F348E-MC
DATE: 14 JULY 1980.
2. TSV05 TRANSPORT SUBSYSTEM USE?'S GUIDE; DOCUMENT NUMBER EK-TSV05-UG-001
DATE: AUGUST 1982
3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSV05-TM-001
DATE: AUGUST 1982
4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSV05-IN-001
DATE: AUGUST 1982

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

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

1.5 ASSUMPTIONS

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

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.1.1 OPERATOR COMMANDS

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

BOOT THE DIAGNOSTIC XXDP MEDIA

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


2.2 SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

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

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

FLAG	EFFECT
----	-----
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX 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 = 172520, VECTOR = 224

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

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

UNIT 0

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

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

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE '# UNITS?' QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING AS FOLLOWS:
UP TO 4 TSV05 CONTROLLERS PER 11/23 AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES

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

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

UNITS (D) ? 8<CR>

UNIT 1

CSR ADDRESS (O) ? 160000<CR>

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

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

UNIT 2

CSR ADDRESS (O) ? 160000<CR>

SUB-DEVICE # (O) ? 1<CR>

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

UNIT 3

CSR ADDRESS (O) ? 160000<CR>

SUB-DEVICE # (O) ? 2<CR>

Q-FACTOR (O) 0 ? <CR>

UNIT 4

CSR ADDRESS (O) ? 160000<CR>

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

Q-FACTOR (O) 0 ? <CR>

UNIT 5

CSR ADDRESS (O) ? 160000<CR>

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

Q-FACTOR (O) 0 ? <CR>

UNIT 6

CSR ADDRESS (O) ? 160000<CR>

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

Q-FACTOR (O) 0 ? <CR>

UNIT 7

CSR ADDRESS (O) ? 160000<CR>

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

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

UNIT 8

CSR ADDRESS (O) 160000<CR>

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

Q-FACTOR (O) 1 ? <CR>

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

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

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

UNITS (D) ? 8<CR>

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

UNIT 3
CSR ADDRESS (D) ? 160000<CR>
SUB-DEVICE # (D) ? 2-5<CR>
Q-FACTOR (D) 0 ? 0<CR>

UNIT 7
CSR ADDRESS (D) ? 160000<CR>
SUB-DEVICE # (D) ? 6,7<CR>
Q-FACTOR (D) 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 (D) ? 160000<CR>
SUB-DEVICE # (D) ? 0-7<CR>
Q-FACTOR (D) 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
CVTSD 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.

CVTSD 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.

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

4.0 PERFORMANCE AND PROGRESS REPORTS

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

SUCCESSFUL RUN EXAMPLE (PDP-11/23)

DR>STA/FLA:PNT:HOE

UNITS (D) ? 1

UNIT 0

DEVICE ADDRESS (O) 172520 ? <CR>

VECTOR (O) 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 PDP-11/23 PROCESSOR WITH A LA34 CONSOLE.

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

TEST NUMBER	N/I SECS.	NUMBER ITER	DEF SECS.
1	1	2	1
2	1	1	0
3	1	1	0
4	1	1	0
5	1	1	0
6	1	1	0
7	1	1	0
8	1	1	0

THE TIMES REQUIRED TO RUN TESTS 1 THROUGH 37 IN ONE COMMAND:

Q.V. 15 SECONDS
DEFAULT 16 SECONDS

5.0 DEVICE INFORMATION TABLES

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

CHANGE HW (L) ?

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

UNIT 0

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

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

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

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

CHANGE SW (L) ?

INHIBIT ITERATIONS (L) N ?

6.0 TEST SUMMARIES

TEST 1: 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 AVAILABLE 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

```

2      .TITLE  TSV2 - PROGRAM HEADER
3      .SBTTL  PROGRAM HEADER
4
10     .MCALL  SVC
11 000000 SVC                ; INITIALIZE SUPERVISOR MACROS
12     .ENABLE LC
13     .NLIST  BEX,CND
19 000000 .ENABL  ABS,AMA
20     .=2000
21 002000 BGNMOD  TSV2
    002000
22
23     ;++
24     ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
25     ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
26     ;--
27
28
29 002000 POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
30 002000 HEADER CVTSD,A,0,655.,0
    002000 L$NAME::          ;DIAGNOSTIC NAME
    002000 103 .ASCII /C/
    002001 126 .ASCII /V/
    002002 124 .ASCII /T/
    002003 123 .ASCII /S/
    002004 104 .ASCII /D/
    002005 000 .BYTE 0
    002006 000 .BYTE 0
    002007 000 .BYTE 0
    002010 L$REV::          ;REVISION LEVEL
    002010 101 .ASCII /A/
    002011 L$DEPO::        ;0
    002011 060 .ASCII /O/
    002012 L$UNIT::        ;NUMBER OF UNITS
    002012 000000 .WORD 0
    002014 L$TIML::        ;LONGEST TEST TIME
    002014 001217 .WORD 655.
    002016 L$HPCP::        ;PTR. TO H.W. QUES.
    002016 105556 .WORD L$HARD
    002020 L$SPCP::        ;PTR. TO S.W. QUES.
    002020 105710 .WORD L$SOFT
    002022 L$HPTP::        ;PTR. TO DEF. H.W. PTABLE
    002022 002150 .WORD L$HW
    002024 L$SPTP::        ;PTR. TO S.W. PTABLE
    002024 002160 .WORD L$SW
    002026 L$LADP::        ;DIAG. END ADDRESS
    002026 106404 .WORD L$LAST
    002030 L$STA::        ;RESERVED FOR APT STATS
    002030 000000 .WORD 0
    002032 L$CO::         .WORD 0
    002032 000000 .WORD 0
    002034 L$DTYP::        ;DIAGNOSTIC TYPE
    002034 000000 .WORD 0
    002036 L$APT::         ;APT EXPANSION
    002036 000000 .WORD 0
    002040 L$DTP::        ;PTR. TO DISPATCH TABLE
    002040 002124 .WORD L$DISPATCH

```

002042		LSPRIO::		;DIAGNOSTIC RUN PRIORITY
002042	000000		.WORD 0	
002044		LSENV1::		;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000		.WORD 0	
002046		LSEXP1::		;EXPANSION WORD
002046	000000		.WORD 0	
002050		LSMREV::		;SVC REV AND EDIT #
002050	003		.BYTE CSREVISION	
002051	003		.BYTE CSEDIT	
002052		LSEF::		;DIAG. EVENT FLAGS
002052	000000		.WORD 0	
002054	000000		.WORD 0	
002056		LSSPC::		
002056	000000		.WORD 0	
002060		LSDEVP::		; POINTER TO DEVICE TYPE LIST
002060	003374		.WORD LSDVTYP	
002062		LSREPP::		;PTR. TO REPORT CODE
002062	022744		.WORD LSRPT	
002064		LSEXP4::		
002064	000000		.WORD 0	
002066		LSEXP5::		
002066	000000		.WORD 0	
002070		LSAUT::		;PTR. TO ADD UNIT CODE
002070	022432		.WORD LSAU	
002072		LSDUT::		;PTR. TO DROP UNIT CODE
002072	022530		.WORD LSDU	
002074		LSLUN::		;LUN FOR EXERCISERS TO FILL
002074	000000		.WORD 0	
002076		LSDESP::		;POINTER TO DIAG. DESCRIPTION
002076	003402		.WORD LSDESC	
002100		LSLOAD::		;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		EMT ESLOAD	
002102		LSETP::		;POINTER TO ERR_TBL
002102	000000		.WORD 0	
002104		LSICP::		;PTR. TO INIT CODE
002104	021636		.WORD LSINIT	
002106		LSCCP::		;PTR. TO CLEAN-UP CODE
002106	022716		.WORD LSCLEAN	
002110		LSACP::		;PTR. TO AUTO CODE
002110	022636		.WORD LSAUTO	
002112		LSPRT::		;PTR. TO PROTECT TABLE
002112	021626		.WORD LSPROT	
002114		LSTEST::		;TEST NUMBER
002114	000000		.WORD 0	
002116		LSDLY::		;DELAY COUNT
002116	000000		.WORD 0	
002120		LSHIME::		;PTR. TO HIGH MEM
002120	000000		.WORD 0	

.SBTTL DISPATCH TABLE

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

DISPATCH 9
.WORD 9
LSDISPATCH::
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9

32
33
34
35
36
37
38
39 002122
002122 000011
002124
002124 023526
002126 032334
002130 041432
002132 046770
002134 053046
002136 056042
002140 063414
002142 073344
002144 101132
40

```

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$HW/2
    002150                                L$HW::
    002150                                DFPTBL::
50
51 002150 172520                      .WORD  172520      ; 1ST (OF 2) REGISTERS.
52 002152 000224                      .WORD  224        ; INTERRUPT VECTOR
53 002154 000200                      .WORD  PRI04      ; INTERRUPT PRIORITY.
54 002156                                ENDPW
    002156                                L10000:

```

```

56                                     .SBTTL  SOFTWARE P-TABLE
57
58                                     :++
59                                     : THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
60                                     : PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
61                                     :--
62 002156                                BGNSW  SFPTBL
    002156 000004                        .WORD  L10001-L$SW/2
    002160
    002160
63
64 002160 000000                        TRANSTST:: .WORD  0      : ENABLE TEST OF TRANSPORT(S) IF =1
65 002162 000000                        NOITS::   .WORD  0      : INHIBIT ITERATION OPTION.
66
67                                     : ... 0 = ITERATE.
68 002164 000017                        LERRMAX::  .WORD  15.    : ...NZ = INHIBIT ITERATE.
69 002166 000310                        GERRMAX::  .WORD  200.   : LOCAL (PER TEST) ERROR LIMIT
70 002170                                ENDSW      : GLOBAL (PER UNIT) ERROR LIMIT
    002170
71
72 002170                                L10001:
                                     ENDMOD

```

7
8
13
19
20 002170
002170
21
22
23
24
25
26
27
28
29
33 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 DEFINITIONS
:

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

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

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

000040	EF.START== 32.	: START COMMAND WAS ISSUED
000037	EF.RESTART== 31.	: RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE== 30.	: CONTINUE COMMAND WAS ISSUED
000035	EF.NEW== 29.	: A NEW PASS HAS BEEN STARTED
000034	EF.PWR== 28.	: A POWER-FAIL/POWER-UP OCCURRED

:
:

; PRIORITY LEVEL DEFINITIONS

000340	PRI07==	340
000300	PRI06==	300
000240	PRI05==	240
000200	PRI04==	200
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	PR1==	2000
004000	IXE==	4000
010000	IBE==	10000
020000	IER==	20000
040000	LOE==	40000
100000	HOE==	100000

34
35 002170KT11
; SBTTL MEMORY MANAGEMENT DEFINITIONS

; DEFINE MEMORY MANAGEMENT REGISTERS

000250	*KT11 VECTOR ADDRESS
	MMVEC= 250
	*KT11 STATUS REGISTER ADDRESSES
177572	SR0= 177572
177574	SR1= 177574
177576	SR2= 177576
172516	SR3= 172516

.IF NB

; *USER 'I' PAGE DESCRIPTOR REGISTERS

UIPDR0=	177600
UIPDR1=	177602
UIPDR2=	177604
UIPDR3=	177606
UIPDR4=	177610
UIPDR5=	177612
UIPDR6=	177614
UIPDR7=	177616

.IF NB

; *USER 'D' PAGE DESCRIPTOR REGISTERS

UDPDR0=	177620
UDPDR1=	177622
UDPDR2=	177624
UDPDR3=	177626
UDPDR4=	177630
UDPDR5=	177632
UDPDR6=	177634
UDPDR7=	177636


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

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

.SBTTL TSV05 REGISTER AND PACKET DEFINITIONS

:
: SOME GENERAL EQUATES.
:

46	000004	ERRVEC==	4	: POINTER TO ERROR VECTOR FOR BUS TIME OUT.
47	000060	TTIVEC==	60	: INTERRUPT VECTOR FOR CONSOLE INPUT
48	177560	TTICSR==	177560	: BUS ADDRESS OF CONSOLE INPUT
49	177562	TTIBFR==	177562	: CONSOLE INPUT DATA BUFFER
50	177520	BDVPCR==	177520	: BDV11 PAGE CONTROL REGISTER

:+
:BIT DEFINITIONS FOR TSSR REGISTER
:-

56	100000	SC=	BIT15	:SPECIAL CONDITION
57	040000	BIE=	BIT14	:BUS INTERFACE ERROR
58	020000	SCE=	BIT13	:SANITY CHECK ERROR
59	010000	RMR=	BIT12	:MODIFICATION REFUSED
60	004000	NXM=	BIT11	:NONEXISTANT MEMORY ERROR
61	002000	NBA=	BIT10	:NEED BUFFER ADDRESS
62	001400	HIADDR=	BIT9!BIT8	:EXTENDED ADDRESS BITS
63	000200	SSR=	BIT7	:SUB SYSTEM READY
64	000100	OFL=	BIT6	:OFF LINE BIT
65	000060	FATERR=	BIT4!BIT5	:FATAL TERMINATION ERROR CODES
66	000016	TERCLS=	BIT3!BIT2!BIT1	:TERMINATION CODES

:+
:BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
:(XST0)
:-

76	100000	XSOTMK=	BIT15	:TAPE MARK DETECTED
77	040000	XSORLS=	BIT14	:RECORD LENGTH SHORT
78	020000	XSOLET=	BIT13	:LOGICAL END OF TAPE
79	010000	XSORLL=	BIT12	:RECORD LENGTH LONG
80	004000	XSOWLE=	BIT11	:WRITE LOCK ERROR
81	002000	XSONEF=	BIT10	:NON EXECUTABLE FUNCTION
82	001000	XSOILC=	BIT9	:ILLEGAL COMMAND
83	000400	XSOILA=	BIT8	:ILLEGAL ADDRESS
84	000200	XSOMOT=	BIT7	:TAPE IN MOTION
85	000100	XSOONL=	BIT6	:TRANSPORT ON LINE
86	000040	XSOIE=	BIT5	:INTERRUPT ENABLE
87	000020	XSOVCK=	BIT4	:VOLUME CHECK BIT
88	000010	XCOPE=	BIT3	:PHASE ENCODED DRIVE
89	000004	XSOWLK=	BIT2	:WRITE LOCKED
90	000002	XSOBOT=	BIT1	:BEGINNING OF TAPE
91	000001	XSOEOT=	BIT0	:END OF TAPE

:+
:BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
:(XST1)

```

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

```

```

154      :-
155      :-
156      :-
157      000000      TSBA== 0
158      000000      TSDB== 0      ;TSDB/TSBA REGISTER
159      000001      TSBAH== 1
160      000001      TSDBH== 1      ;TSDB/TSBA REGISTER HIGH BYTE
161      000002      TSSR== 2      ;TSSR REGISTER
162      000003      TSSRH== 3      ;TSSR REGISTER HIGH BYTE
163
164      ;+
165      ; TSDB ADDRESS BIT DEFINITIONS
166      :-
167      000003      A1716 = BIT1+BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
168
169      ;+
170      ; COMMAND DEFINITIONS
171      :-
172      000017      P.GETSTAT      = 17      ;GET STATUS
173      000013      P.INIT      = 13      ;INITIALIZE
174      000012      P.CONTROL      = 12      ;CONTROL COMMANDS
175      000011      P.FORMAT      = 11      ;FORMAT
176      000010      P.POSITION      = 10      ;POSITION
177      000006      P.WRTSUB      = 6      ;SUBSYSTEM WRITE
178      000005      P.WRITE      = 5      ;WRITE
179      000004      P.WRTCHAR      = 4      ;WRITE CHARACTERISTICS
180      000001      P.READ      = 1      ;READ
181
182      ;+
183      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
184      :-
185      100000      P.ACK      = BIT15      ;BUFFER AVAIL FOR CONTROLLER
186      040000      P.CVC      = BIT14      ;CLEAR VOLUME CHECK
187      020000      P.OPP      = BIT13      ;REVERSE SEQUENCE OF DATA BITS
188      010000      P.SWB      = BIT12      ;SWAP BYTES IN MEMORY
189      007400      P.MODE      = BIT11:BIT10:BIT9:BIT8 ;EXTENDED COMMAND MODE FIELD
190      000200      P.IE      = BIT7      ;INTERRUPT ENABLE
191      000140      P.FMT      = BIT6:BIT5      ;PACKET HEADER TYPE (ALWAYS=0)
192      000037      P.CMD      = 37      ;MAJOR COMMAND FIELD
193
194      ;+
195      ; CONTROL COMMAND MODE CODES
196      :-
196      000000      PC.RELEASE      = 0*256.      ;RELEASE BUFFER
197      000400      PC.REWIND      = 1*256.      ;REWIND
198      001000      PC.NOOP      = 2*256.      ;NO-OP
199      002000      PC.IEREW      = 4*256.      ;REWIND IMMEDIATE INTERRUPT
200      002400      PC.ERASE      = 5*256.      ;SECURITY ERASE
201
202      ;+
203      ; CONTROLLER RAM DEFINITIONS
204      :-
205      000167      RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
206      000200      RMCHEND = 200      ;CHARACTERISTICS IO DATA END RAM ADDRESS
207      000201      RMPKTBEG= 201      ;COMMAND PACKET BEGIN RAM ADDRESS
208      000210      RMPKTEND= 210      ;COMMAND PACKET END RAM ADDRESS
209      000215      RMMSGBEG= 215      ;MESSAGE BUFFER BEGIN RAM ADDRESS
210      000234      RMMSGEND= 234      ;MESSAGE BUFFER END RAM ADDRESS

```



```

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

```

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

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

```

340 .SBTTL SPECIAL MACROS AND OPDEFS.
341
342
343 :+
344 :SAVE GENERAL REGS 1 TO 5
345 :-
346
347 .MACRO SAVREG
348 JSR R5,REGSAV
349 .ENDM
350
351 :+
352 : MACRO TO FORCE AN ERROR
353 :-
354 .MACRO FORCERROR TAG,NOTSSR
355 .NLIST
356 .IIF NDF LISTALL, .NLIST
357 .LIST
358 .IF B NOTSSR
359 MOV TSSR(R5),R1 ;READ TSSR
360 .ENDC
361 MOV FORCER,FORCER ;IS FORCER SET? (LEAVE C BIT ALONE)
362 BNE TAG ;BR IF YES
363 .NLIST
364 .IIF NDF LISTALL, .LIST
365 .LIST
366 .ENDM
367
368 :+
369 : MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
370 : WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
371 : SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
372 : FORCER TO 177777
373 : TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
374 :-
375 .MACRO FORCEEXIT TAG
376 .NLIST
377 .IIF NDF LISTALL, .NLIST
378 .LIST
379 MOV FORCER,FORCER ;IS FORCER NEGATIVE?
380 BMI TAG ;BR IF YES
381 .NLIST
382 .IIF NDF LISTALL, .LIST
383 .LIST
384 .ENDM
385 :+
386 : MACRO TO INCREMENT ERROR COUNTS
387 :-
388 .MACRO NEXT.ERRNO
389 .NLIST
390 :...IIF NDF LISTALL, .NLIST
391 ERRNO=ERRNO+1
392 :...IIF NDF LISTALL, .LIST
393 .LIST
394 .ENDM
395
396 :+

```

```

397          ;MACRO TO PERFORM XOR
398          :-
399
400          .MACRO XOR      A,B
401          MOV      A,-(SP)
402          BIC      B,(SP)
403          BIC      A,B
404          BIS      (SP)+,B
405          .ENDM
406
407          000000          EN=0          ; INITIALIZE ERROR NUMBER
408          .SBTTL  FORCER - FORCE ERROR FLAG
409
410          :
411          : THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
412          : TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
413          :
414
415 002170 000000  FORCER::      0          ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
416                                ; - BY THE MACRO 'IFERROR'). AN ERROR NEED NOT -
417                                ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
418
419
420

```


.SBTTL GLOBAL DATA SECTION

```
422
423
424
425
426
427
428
429
430
431
432
433 002172 000000
434 002174 000000
435 002176 000000
436 002200 000000
437 002202 000224
438 002204 000200
439 002206 000000
440 002210 000000
441 002212 000000
442 002214 000000
443 002216 000J00
444 002220 000000
445 002222 000000
446 002224 000000
447 002226 000000
448 002230 000000
449 002232 000000
450 002234
451 002274 000000
452 002276 000000
453 002300 000000
454 002302 000000
455 002304 000000
456 002306 000000
457 002310 000000
458 002312 000000
459 002314
460 002460
461 002624

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

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

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

```

463 .SBTTL TSTBLK - TEST DATA TABLE
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478

```

```

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

```

```

479 002744
480 002744 000000
481 002746 177777
482 002750 000001
483 002752 000002
484 002754 000004
485 002756 000010
486 002760 000020
487 002762 000040
488 002764 000100
489 002766 000200
490 002770 000400
491 002772 001000
492 002774 002000
493 002776 004000
494 003000 010000
495 003002 020000
496 003004 040000
497 003006 100000
498 003010 177776
499 003012 177775
500 003014 177773
501 003016 177767
502 003020 177757
503 003022 177737
504 003024 177677
505 003026 177577
506 003030 177377
507 003032 176777
508 003034 175777
509 003036 173777
510 003040 167777
511 003042 157777
512 003044 137777
513 003046 077777
514 003050 125252
515 003052 052525
516 003054

```

```

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

TBLEND==.

```

```

518                                     .SBTTL GLOBAL ENVIRONMENT STORAGE
519
520                                     : STORAGE FOR DEVICE REGISTERS
521
522 003054 000000 100000 000000 DUMMY: 0,100000,0,0 ; DUMMY DEVICE REGISTERS...
523 003064 000000 000000 000000 0,0,0,0,0,0,0,0 ; ...FOR MULTI-UNIT CHECKOUT.
524
525
526
527 003104 000000 DUFLG:: .WORD 0 ; 'DROPPED UNIT' FLAG.
528                                     ; INHIBITS CODE IN 'CLEAN-UP'.
529 003106 000000 NODEV:: .WORD 0 ; FLAG TO SAY NO DEVICE.
530
531 003110 000000 TEMP1:: .WORD 0 ; SOME TEMP LOCATIONS.
532 003112 000000 TEMP2:: .WORD 0
533 003114 000000 XXCOMM:: .WORD 0 ; XXDP+ COMM BLOCK POINTER.
534 003116 000000 FREE:: .WORD 0 ; 1ST FREE MEMORY ADDRESS...
535 003120 000000 FRESIZ:: .WORD 0 ; ...AND SIZE (IN WORDS).
536 003122 000000 FREEHI: .WORD 0 ; LAST WORD IN FREE SPACE
537 003124 000000 KTFLG:: .WORD 0 ; KT11, MEM AVAIL FLAG -
538                                     ; - .WORD 0 = <24K OR NO KT -
539                                     ; - NZ = >24K AND KT.
540 003126 000000 KTENABLE:: .WORD 0 ; SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
541 003130 000000 NXMFLG:: .WORD 0 ; SET IF WE CAN TEST CLEARED OTHERWISE
542 003132 000000 NXMLO:: .WORD 0 ; NXM LO ADDRESS BITS
543 003134 000000 NXMHI:: .WORD 0 ; NXM HI ADDRESS BITS FOR DAL'S 16-21
544 003136 000000 T23A:: .WORD 0 ; 11/23A FLAG
545 003140 000000 T23B:: .WORD 0 ; 11/23B FLAG
546 003142 000000 T3BFLG:: .WORD 0 ; TEST 3B FLAG ^0
547 003144 002000 PST32W:: .WORD 2000 ; 32W BLOCK ADDRESS FOR 32K START
548 003146 000000 SIFLAG:: .WORD 0
549 003150 000000 BADDAT:: .WORD 0 ; ACTUAL DATA
550 003152 000000 GDDAT:: .WORD 0 ; EXPECTED DATA
551 003154 000000 LOOPFL:: .WORD 0
552 003156 CTAB:: .WORD 0 ; CONFIGURATION TABLES.
553 003156 000000 CTABM:: .WORD 0 ; CONFIG WORK.
554 003160 000000 .WORD 0
555 003162 000000 .WORD 0
556 003164 000000 .WORD 0
557 003166 177777 .WORD -1 ; END OF MEM TABLE.
558 003170
559 CTABE::
560 : ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
561 :
562 : 0 = UNIT NOT TESTED
563 : 100000 = UNIT ONLINE, NO ERRORS
564 : 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
565 : 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
566 : 160001 = UNIT DROPPED, NOT IDLE AT START
567 : 14XXXX = UNIT DROPPED, ENCOUNTERED .XXXX ERRORS
568 003170 ERTABL: .BLKW 64.
569 003370 000000 ERTABE: .WORD 0
570
571 003372 000000 SKIPT: .WORD 0 ; 1=SKIP SUBTEST 0=NO SKIP OF SUBTEST

```

```
573 .SBTTL GLOBAL TEXT MESSAGES
574
575 :++
576 : THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
577 : MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
578 : MORE THAN ONE TEST.
579 :--
580
581
582 :+
583 : NAMES OF DEVICES SUPPORTED
584 :-
585
586 003374 DEVTYP <TSV05>
003374 LSDVTYP::
003374 124 123 126 .ASCIZ /TSV05/
.EVEN

587
608 :+
609 : TEST DESCRIPTION
610 :-
611 003402 DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - CHECK TRANSPORT IF ERROR ****>
003402 LSDESC::
003402 052 052 052 .ASCIZ /**** TSV05 LOGIC DIAGNOSTIC - CHECK TRANSPORT IF ERROR ****/
.EVEN

613
614
615
616 :+
617 : BIT TO ASCII CONVERSION FOR TSSR REGISTER
618 :-
619
620 003476 003536 003541 003545 TSSRBIT:: .WORD 1$,2$,3$,4$,5$,6$,7$,8$
621 003516 003577 003603 003607 .WORD 9$,10$,11$,12$,13$,14$,15$,16$
622 003536 123 103 000 1$: .ASCIZ 'SC'
623 003541 102 111 105 2$: .ASCIZ 'BIE'
624 003545 123 103 105 3$: .ASCIZ 'SCE'
625 003551 122 115 122 4$: .ASCIZ 'RMR'
626 003555 116 130 115 5$: .ASCIZ 'NXM'
627 003561 116 102 101 6$: .ASCIZ 'NBA'
628 003565 102 111 124 7$: .ASCIZ 'BIT9'
629 003572 102 111 124 8$: .ASCIZ 'BIT8'
630 003577 123 123 122 9$: .ASCIZ 'SSR'
631 003603 117 106 114 10$: .ASCIZ 'OFL'
632 003607 102 111 124 11$: .ASCIZ 'BIT5'
633 003614 102 111 124 12$: .ASCIZ 'BIT4'
634 003621 102 111 124 13$: .ASCIZ 'BIT3'
635 003626 102 111 124 14$: .ASCIZ 'BIT2'
636 003633 102 111 124 15$: .ASCIZ 'BIT1'
637 003640 102 111 124 16$: .ASCIZ 'BIT0'
638 .EVEN
639 003646 124 123 123 SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
640 003701 124 123 123 SFHERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
641 003734 040 040 116 NXR: .ASCIZ / NON-EXISTANT DEVICE REGISTER/
642 003773 045 101 040 NXR: .ASCIZ /% ADDRESS: %06/
643 004014 045 101 040 TSSX: .ASCIZ /% TSBA,TSSR EXP'D: %06%,%06%/
644 004054 045 101 040 .ASCIZ /% TSBA,TSSR REC'D: %06%,%06/
```

```
645 004113 045 116 045 FUSI: .ASCII /%N%/
646 004117 040 040 125 USI: .ASCII / UNEXPECTED INTERRUPT/
647 004146 040 040 111 NSI: .ASCII / INTERRUPT EXPECTED, NOT RECEIVED/
648 004211 045 116 045 FNOINTR: .ASCII /%N%/
649 004215 040 040 116 NOINTR: .ASCII / NO INTERRUPT WAS GENERATED/
650 004252 040 040 111 IFAULT: .ASCII / INTERRUPT FAULT/
651 004274 045 101 040 INTX: .ASCII /%A CPU PC: %06%A TSBA: %06/
652 004331 040 040 042 NOINIT: .ASCII / 'BUS-INIT' DIDN'T INITIALIZE CONTROLLER/
653 004403 040 040 042 NSINIT: .ASCII / 'SOFT-INIT' DIDN'T INITIALIZE THE DPU/
654 004453 040 040 042 BRINIT: .ASCII / 'BUS-RESET' DIDN'T INITIALIZE THE DPU/
655
656 004523 000 NUL: .ASCII //
657 004524 045 116 000 NULCR: .ASCII /%N/
658 004527 045 101 040 EXPGOT: .ASCII /%A EXP'D: %06%A, REC'D: %06/
659 004563 045 116 045 EXPGT2: .ASCII /%N%A EXP'D: %06%A, %06%N%A REC'D: %0%A, %06/
660 004637 045 101 040 DUAD12: .ASCII /%A REG(W) WRITTEN TO: %06%A REG(R) READ: EXP'D: %06%A, REC'D: %06/
661 004741 122 101 115 PKTRAM: .ASCII 'RAM Contents Do Not Match Packet Sent'
662 005007 040 040 103 SCME: .ASCII / CONFIG DOESN'T MATCH MFG. MASTER/
663 005052 127 122 111 WRTMSG: .ASCII 'WRITE CHARACTERISTICS Failed'
664 005107 124 123 123 WRTERR: .ASCII 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
665 005202 124 123 123 RDERR: .ASCII 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
666 005274 106 101 124 SCHERR: .ASCII 'FATAL ERROR IN SUBTEST - CHECK TAPE, CABLES, TRANSPORT etc.'
667 005366 105 122 122 RETERR: .ASCII 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
668 005454 045 116 045 NOMEM: .ASCII '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****%N'
669 005550 045 116 045 M8186: .ASCII '%N%A ***** 11/23A SYSTEM *****%N'
670 005641 045 116 045 M8189: .ASCII '%N%A ***** 11/23B SYSTEM *****%N'
671 .EVEN
672
673
674
```



```

676                                     .SBTTL GLOBAL ERROR REPORT SECTION
677
678                                     ;++
679                                     ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
680                                     ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
681                                     ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
682                                     ;--
683
684 005732      BGNMSG  NXRERR              ;NON-EXISTANT DEVICE REGISTER.
        005732
685 005732      NXRERR::
        005732      PRINTX  #NXRX,NODEV    ;NODEV = NEXM ADDRESS.
        005732      MOV     NODEV,-(SP)
        005736      MOV     #NXRX,-(SP)
        005742      MOV     #2,-(SP)
        005746      MOV     SP,R0
        005750      TRAP    C$PNTX
        005752      ADD     #6,SP
        005756      JSR     PC,EXTEND      ; PRINT EXTENSION IF REQUIRED.
        004737      005764
        005762
        005762      L10002:
        005762      TRAP    C$MSG
        104423
688
689
690
691                                     ;
692                                     ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
693                                     ; TO ANY OF THE ABOVE ERROR SIGNATURES.
694
695 EXTEND: TST      (PC)+
696 EXTA:  0
697        BEQ      1$
698        JSR      PC,@EXTA
        005776      PRINTX  #NULCR
        005776      MOV     #NULCR,-(SP)
        006002      MOV     #1,-(SP)
        006006      MOV     SP,R0
        006010      TRAP    C$PNTX
        006012      ADD     #4,SP
        006016      RTS     PC
        000207
        000000
        001402
        004777      177770
        012746      004524
        012746      000001
        010600
        104415
        062706      000004
        000207

```

```

702                                .SBTTL PRITSSR - PRINT TSSR CONTENTS
703
704                                :+
705                                :ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
706                                :THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
707                                :BY A MESSAGE PRINTING ROUTINE
708
709                                :INPUTS:
710
711                                :      R1      CONTENTS OF TSSR
712
713                                :SUBORDINATE ROUTINES:
714
715                                :      CHKAMB  CHECK FOR AMBIGUOUS CONTENTS
716
717                                :-
718
719
720 PRITSSR:
721     SAVREG                      :SAVE GENERAL REGISTERS
722     MOV      R1,R4              :SAVE THE TSSR CONTENTS
723     PRINTB   #TSSRFOR,R4       :PRINT THE CONTENTS OF TSSR
724     MOV      R4,-(SP)
725     MOV      #TSSRFOR,-(SP)
726     MOV      #2,-(SP)
727     MOV      SP,R0
728     TRAP     C$PNTB
729     ADD      #6,SP
730     MOV      R4,R0              :GET TSSR BACK FOR CHKAMB
731     JSR      PC,CHKAMB          :ARE CONTENTS AMBIGUOUS ?
732     BCS      5$                 :BRANCH IF NOT
733     PRINTX   #AMBTSSR          :SHOW CONTENTS ARE AMBIGUOUS
734     MOV      #AMBTSSR,-(SP)
735     MOV      #1,-(SP)
736     MOV      SP,R0
737     TRAP     C$PNTX
738     ADD      #4,SP
739     5$: MOV      R4,R3           :CONTENTS OF TSSR
740     BIC      #HIADDR!FATERR!TERCLS,R3 :CLEAR ALL MULTIPLE BIT FIELDS
741     BEQ      20$
742     MOV      #TMPBFR,R2        :TEMPORARY ASCII BUFFER
743     MOV      #TSSRBIT,R1       :ASCII EQUIVALENT OF BITS
744     10$: TST      R3            :REMAINING BITS TO CONVERT
745     BEQ      15$               :BRANCH WHEN ALL ARE DONE
746     CLC
747     ROL      R3                :CLEAR CARRY FOR SHIFT
748     BCC      13$               :SHIFT NEXT BIT TO CARRY
749     MOV      (R1),R0            :BRANCH IF BIT NOT SET
750     11$: MOV      (R0)+,(R2)+   :POINTER TO BIT DEFINITION
751     BNE      11$               :MOVE ASCII TO BUFFER
752     MOV      #',-1(R2)         :MOVE ALL BITS
753     13$: TST      (R1)+         :INSERT A COMMA TO TERMINATE
754     BR       10$               :POINT TO NEXT DESCRIPTION
755     15$: CLRB      -(R2)        :GET THE REMAINING BITS
756     PRINTX   #TSSDEF,#TMPBFR   :TERMINATE THE LINE
757     MOV      #TMPBFR,-(SP)     :PRINT THE BIT DEFINITIONS
758     MOV      #TSSDEF,-(SP)

```

```
006164 012746 000002      MOV    #2,-(SP)
006170 010600      MOV    SP,R0
006172 104415      TRAP   C$PNTX
006174 062706 000006      ADD    #6,SP
746
747 006200 010403      20$:  MOV    R4,R3      ;GET THE TSSR CONTENTS
748 006202 042703 177761 BIC    #^CTERCLS,R3 ;CLEAR ALL BUT TERMINATION
749 006206 016303 006754 MOV    TCOCOD(R3),R3 ;GET THE TERMINATION CODE MEANING
750 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$PNTX
      006230 062706 000006 ADD    #6,SP
751 006234 010403      MOV    R4,R3      ;TSSR CONTENTS AGAIN
752 006236 042703 177717 BIC    #^CFATERR,R3 ;CLEAR ALL BUT FATAL TERMINATION
753 006242 001416      BEQ     25$      ;DON'T PRINT IF ZERO
754 006244 006203      ASR     R3
755 006246 006203      ASR     R3
756 006250 006203      ASR     R3      ;ALINE TERMINATION CODE FOR INDEX
757 006252 016303 007314 MOV    TSFCOD(R3),R3 ;GET THE FATAL TERMINATION CODE
758 006256      PRINTX  #TFCASC,R3 ;PRINT THE FATAL TERMINATION CODE
      006256 010346      MOV    R3,-(SP)
      006260 012746 006615 MOV    #TFCASC,-(SP)
      006264 012746 000002 MOV    #2,-(SP)
      006270 010600      MOV    SP,R0
      006272 104415      TRAP   C$PNTX
      006274 062706 000006 ADD    #6,SP
759 006300 042704 176377      25$: BIC    #^CHIADDR,R4 ;CLEAR ALL BUT EXTENDED ADDRESS
760 006304 001411      BEQ     30$      ;DON'T PRINT IF ZERO
761 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$PNTX
      006324 062706 000006 ADD    #6,SP
762 006330 013703 002172      30$: MOV    EPRTSW,R3      ;PRINT MEASGE BUFFER ADDRESS
763 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$PNTX
      006346 062706 000004 ADD    #4,SP
764 006352 000207      RTS     PC      ;RETURN TO CALLER
765
780 006354      045      116      045 EPRT1: .ASCIZ 'XNXA *****CHECK TRANSPORT*****'
781 006413      045      116      045 EPRT2: .ASCIZ 'XNXA *****CHECK PARITY SWITCH IN TRANSPORT*****'
783 006473      045      116      045 TSSRFOR: .ASCIZ 'XNXA TSSR = %06'
784 006513      045      116      045 TEXASC: .ASCIZ 'XNXA Extended Address Bits = %06'
785 006554      045      116      045 TCOASC: .ASCIZ 'XNXA Termination Class Code = %T'
786 006615      045      116      045 TFCASC: .ASCIZ 'XNXA Fatal Termination Class Code = %T'
787 006664      045      116      045 TSSDEF: .ASCIZ 'XNXA TSSR Bits Set: %T'
788 006713      045      116      045 AMBTSSR: .ASCIZ 'XNXA TSSR Contents Are Ambiguous'
789
790 006754 006774 007017 007045 TCOCOD: .EVEN
      .WORD 1$,2$,3$,4$,5$,6$,7$,8$
```

791	006774	116	157	162	1\$:	.ASCIZ	'Normal Termination'
792	007017	124	145	162	2\$:	.ASCIZ	'Termination Condition'
793	007045	124	141	160	3\$:	.ASCIZ	'Tape Status Alert'
794	007067	106	165	156	4\$:	.ASCIZ	'Function Reject'
795	007107	122	145	143	5\$:	.ASCIZ	'Recoverable Error - Tape Position One Record Down'
796	007171	122	145	143	6\$:	.ASCIZ	'Recoverable Error - Tape Was Not Moved'
797	007240	125	156	162	7\$:	.ASCIZ	'Unrecoverable Error'
798	007264	106	141	164	8\$:	.ASCIZ	'Fatal Controller Error'
799						.EVEN	
800							
801	007314	007324	007360	007371	TSFCOD:	.WORD	1\$,2\$,3\$,4\$
802	007324	111	156	164	1\$:	.ASCIZ	'Internal Diagnostic Failure'
803	007360	122	145	163	2\$:	.ASCIZ	'Reserved'
804	007371	102	165	163	3\$:	.ASCIZ	'Bus Interface or Sanity Check Error'
805	007435	122	145	163	4\$:	.ASCIZ	'Reserved'
806						.EVEN	

```

808 .SBTTL PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
809
810
811 ;+
812 ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
813 ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
814
815 ;INPUT:
816
817 R0      NUMBER OF WORDS IN PACKET
818 R3      HIGH ORDER COMMAND PACKET ADDRESS
819 R4      ADDRESS OF COMMAND PACKET
820
821 ;-
822
823 PRIPKT::
824 SAVREG      ;SAVE THE REGISTERS
825 MOV R0,R5   ;SAVE NO. OF WORDS IN PACKET
826 TST KTENABLE ;ABOVE 28K UNDER TEST?
827 BNE 10$    ;BR IF YES
828 CLR R3     ;SET HIGH ORDER ADDRESS TO 0
829 MOV R3,R1  ;COPY HIGH ORDER ADDRESS
830 MOV R4,R0  ;GET LOWER ADDRESS
831 ROL R0     ;SHIFT BIT 15 INTO C BIT
832 ROL R1     ;AND INTO HIGH ORDER.
833 PRINTB #PKTADD,R1,R4 ;PRINT PACKET ADDRESS
      MOV R4,-(SP)
      MOV R1,-(SP)
      MOV #PKTADD,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C$PNTB
      ADD #10,SP
834 15$: MOV R3,R0 ;GET HIGH ORDER ADDRESS
835 BEQ 20$    ;BR IF NOT ABOVE 28K.
836 MOV R4,R1  ;GET LOW ORDER ADDRESS
837 JSR PC,SETMAP ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
838 MOV R0,R4  ;GET RETURNED PAR6 ADDRESS BIAS
839 20$: CLR R1 ;SAVE WORD NUMBER
840 25$: MOV (R4)+,R2 ;GET PACKET CONTENTS
841 PRINTB #PKTFRM,R1,R2 ;PRINT THE DATA
      MOV R2,-(SP)
      MOV R1,-(SP)
      MOV #PKTFRM,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C$PNTB
      ADD #10,SP
842 INC R1     ;NEXT WORD NUMBER
843 CMP R1,R5  ;DONE ALL PACKET WORDS?
844 BLT 25$   ;LOOP TILL ALL DONE
845 RTS PC    ;RETURN
846
847 007574 045 116 045 PKTFRM: .ASCIIZ '%N% Packet Word #%D1% = %06'
848 007632 045 116 045 PKTADD: .ASCIIZ '%N% Packet Address = %01%05'
849 .EVEN
850

```

```

852                                     .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR BYTE
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870 007670
871 007670
872 007674 010203
873 007676
874 007706 012700 177400
875 007712 040001
876 007714 040002
877 007716 040003
878 007720
    007720 010346
    007722 010146
    007724 010246
    007726 012746 007752
    007732 012746 000004
    007736 010600
    007740 104414
    007742 062706 000012
879 007746 010300
880 007750 000207
881
882 007752 045 116 045 XORBFOR: .ASCIZ '%X%A EXPD: %03XA RECV: %03XA XOR: %03'
883
884

```

```

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

```



```

886                                     .SBTTL PRI XOR - PRINT EXPD, RECV AND XOR
887
888
889
890                                     :+
891                                     :PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
892                                     :THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
893                                     :INPUTS:
894                                     :
895                                     :R1      RECEIVED DATA
896                                     :R2      EXPECTED DATA
897
898                                     :OUTPUT:
899                                     :
900                                     :R0      XOR OF EXPECTED/RECEIVED DATA
901                                     :
902                                     :-
903
904 010020                               PRI XOR::
905 010020                               SAVREG                               ;SAVE THE REGISTERS
906 010024 010203                       MOV      R2,R3                       ;EXPECTED DATA
907 010026                               XOR      R1,R3                       ;FORM THE EXCLUSIVE OR
908 010036                               PRINTB   #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
909                                     MOV      R3,-(SP)
910                                     MOV      R1,-(SP)
911                                     MOV      R2,-(SP)
912                                     MOV      #XORFOR,-(SP)
913                                     MOV      #4,-(SP)
914                                     MOV      SP,R0
915                                     TRAP     C$PNTB
916                                     ADD      #12,SP
917                                     MOV      R3,R0                               ;R0 HAS XOR ON RETURN
918                                     RTS      PC                               ;RETURN TO CALLER
919
920 010070 045 116 045 XORFOR: .ASCIZ  '%N% EXPD: %06% RECV: %06% XOR: %06%'
921                                     .EVEN
922
923

```

```

915 .SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
916
917 ;+
918 ;
919 ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
920 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
921 ;
922 ;INPUTS:
923 ;
924 ;      R0      OCTAL VALUE TO CONVERT
925 ;      R1      TABLE OF POINTERS TO ASCII EQUIVALENT
926 ;
927 ;-
928
929 PRIEQU:
930 010136 SAVREG
931 010136 RTS PC ;SAVE THE REGISTERS
932 010142 000207 ;RETURN TO CALLER
933
934
935 .SBTTL PRIRAM - PRINT RAM ADDRESS
936
937 ;+
938 ;
939 ;PRINT CONTROLLER RAM ADDRESS.
940 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
941 ;
942 ;INPUTS:
943 ;
944 ;      R4      RAM ADDRESS
945 ;
946 ;-
947 PRIRAM:
948 010144 SAVREG
949 010144 PRINTB #RAMFOR,R4 ;SAVE R1-R5 UNTIL NEXT RETURN
010150 010446 MOV R4,-(SP) ;PRINT RAM ADDRESS IN ERROR
010152 012746 010174 MOV #RAMFOR,-(SP)
010156 012746 000002 MOV #2,-(SP)
010162 010600 MOV SP,R0
010164 104414 TRAP C$PNTB
010166 062706 000006 ADD #6,SP
950 010172 000207 RTS PC ;RETURN
951
952 010174 045 116 045 RAMFOR: .ASCIZ '%N%A CONTROLLER RAM ADDRESS = %06'
953 .EVEN
954
955 .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
956
957 ;+
958 ;
959 ;PRINT MEMORY ADDRESS
960 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
961 ;
962 ;IMPLICIT INPUTS
963 ;
964 ;      ERRHI - HIGH ORDER ADDRESS
965 ;      ERRLO - LOW ORDER ADDRESS

```

```

966
967
968 010236
969 010236
970 010242 013700 002230
971 010246 013701 002232
972 010252 010102
973 010254 006101
974 010256 006100
975 010260
    010260 010246
    010262 010046
    010264 012746 010306
    010270 012746 000003
    010274 010600
    010276 104414
    010300 062706 000010
976 010304 000207
977
978 010306 045 116 045 PRIA0: .ASCIZ '%N% MEMORY ERROR ADDRESS = %01%05%'
979 .EVEN
980
981
982 .SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
983
984
985 :PRINT MEMORY ADDRESS
986 :THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
987
988 :IMPLICIT INPUTS
989
990 :ERRHI - HIGH ORDER ADDRESS
991 :ERRLO - LOW ORDER ADDRESS
992
993
994 010352
995 010352
996 010356 013702 002230
997 010362 013701 002232
998
999
1000
1001 010366
    010366 010146
    010370 012746 010434
    010374 012746 000002
    010400 010600
    010402 104414
    010404 062706 000006
1002 010410
    010410 010246
    010412 012746 010477
    010416 012746 000002
    010422 010600
    010424 104414
    010426 062706 000006
1003 010432 000207

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

    :RETURN

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

.SBTTL PRITADD - PRINT MEMORY TEST ADDRESS

:PRINT MEMORY ADDRESS
:THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.

:IMPLICIT INPUTS

:ERRHI - HIGH ORDER ADDRESS
:ERRLO - LOW ORDER ADDRESS

PRITADD:
    SAVREG
    MOV ERRHI,R2
    MOV ERRLO,R1
    :MOV R1,R2
    :ROL R1
    :ROL R0
    PRINTB #PRIT0,R1
    MOV R1,-(SP)
    MOV #PRIT0,-(SP)
    MOV #2,-(SP)
    MOV SP,R0
    TRAP C$PNTB
    ADD #6,SP
    PRINTB #PRIT1,R2
    MOV R2,-(SP)
    MOV #PRIT1,-(SP)
    MOV #2,-(SP)
    MOV SP,R0
    TRAP C$PNTB
    ADD #6,SP
    RTS PC
    :SAVE R1-R5 UNTIL NEXT RETURN
    :GET HIGH ADDRESS
    :GET LOW ADDRESS
    :COPY LOW ADDRESS
    :SHIFT BIT 15 TO C BIT
    :SHIFT INTO HIGH ORDER
    :PRINT MEMORY ADDRESS LOW IN ERROR

    :PRINT MEMORY ADDRESS HIGH IN ERROR

    :RETURN

```


1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062

.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

IMPLICIT OUTPUT:

TAPE HAS BEEN MOVED

SIDE EFFECTS:

-

SPACE::

SAVREG

MOV #500,SDELAY
MOV #140010,80\$
TST R3
BMI 5\$
MOV R3,90\$
BR 10\$
5\$: BIC #BIT15,R3
MOV R3,90\$
91\$ #BIT8,80\$
10\$: V #80\$,R4
V R4,TSDB(R5)
15\$: JSR PC,WAITF
BCS 20\$
DELAY 250
MOV #250,(PC)+
.WORD 0
MOV L\$DLY,(PC)+
.WORD 0
DEC -6(PC)
BNE -/

;SAVE THE GENERAL REGISTERS
;SET UP DELAY
;SET UP COMMAND, SPACE FORWARD
;CHECK FOR DIRECTION
;BR, IF REVERSE INDICATED
;LOAD UP NUMBER OF RECORDS TO SPACE
;GO DO COMMAND
;CLEAR DIRECTION BIT
;LOAD UP NUMBER OF RECORDS TO SPACE
;SET REVERSE BIT IN COMMAND PACKET
;SET UP R4 WITH PACKET ADDRESS
;SEND OUT COMMAND
;WAIT FOR SSR
;BR, IF SSR IS SET AND OK
;DELAY ABOUT .25 SECONDS

010544
010544
010550 012737 000764 010740
010556 012737 140010 010730
010564 005703
010566 100403
010570 010337 010732
010574 000407
010576 042703 100000
010602 010337 010732
010606 052737 000400 010730
010614 012704 010730
010620 010465 000000
010624 004737 016330
010630 103420
010632 012727 000250
010636 000000
010640 013727 002116
010644 000000
010646 005367 177772
010652 001375

	010654	005367	177756		DEC	-22(PC)	
	010660	001367			BNE	.-20	
1063	010662	005337	010740		DEC	SDELAY	:BUMP DELAY COUNTER DOWN
1064	010666	001356			BNE	15\$:BR, IF MORE DELAY
1065	010670	000411			BR	60\$:BR IF TROUBLE CARRY = CLEAR
1066	010672	016501	000002	20\$:	MOV	TSSR(R5),R1	:READ TSSR
1067	010676	012702	000200		MOV	#SSR,R2	:SET UP EXPECTED
1068	010702	020201		25\$:	CMP	R2,R1	:ARE THEY OK
1069	010704	001401			BEQ	40\$:BR, IF EQUAL = OK
1070	010706	000402			BR	60\$:TROUBLE EXIT
1071	010710	000261		40\$:	SEC		:SET CARRY NO TROUBLE
1072	010712	000401			BR	70\$:EXIT
1073	010714	000241		60\$:	CLC		:CARRY CLEAR = ERROR
1074	010716			70\$:			
1075	010716	010400			MOV	R4,R0	:PASS PACKET ADDRESS
1076	010720	000207			RTS	PC	:RETURN

```
1078  
1079  
1080  
1081  
1082  
1084      010730  
1086  
1087  
1088 010730 000000  
1089  
1090 010732 000000  
1091 010734 000000  
1092 010736 000000  
1093 010740 000000  
1094
```

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

```

1096 .SBTTL WRTCHR - WRITE CHARACTERISTICS COMMAND
1097
1098
1099
1100 :ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1101 :COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1102
1103 :INPUT:
1104
1105 :R4 ADDRESS OF PACKET FROM TEST
1106 :R5 FIRST DEVICE UNIBUS ADDRESS
1107 :REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1108
1109 :OUTPUT:
1110
1111 :R0 TSSR CONTENTS
1112 :CARRY SET - WRITE CHARACTERISTICS COMMAND OK
1113 :CLR - WRITE CHARACTERISTICS FAILED
1114
1115 :IMPLICIT OUTPUT:
1116
1117 :MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
1118 :SOFTWARE SWITCHES SET AS FOLLOWS:
1119 :EXTFEA = EXTENDED FEATURES PRESENT
1120 :BENBSW = BUFFER ENABLE SWITCH ON OR OFF
1121
1122 :SIDE EFFECTS:
1123
1124
1125
1126
1127

```

```

1128 010742 WRTCHR::
1129 010742 :SAVREG ;SAVE THE GENERAL REGISTERS
1130 010746 005037 002222 CLR BENBSW ;CLEAR BUFFER ENABLE SWITCH
1131 010752 005037 002220 CLR EXTFEA ;CLEAR EXTENDED FEATURES SW SWITCH
1132 010756 010465 000000 10$: MOV R4,TSDB(R5) ;SEND OUT COMMAND
1133 010762 004737 016416 JSR PC,CHKTSSR ;WAIT FOR SSR
1134 010766 103401 BCS 20$ ;BR, IF SSR IS SET AND OK
1135 010770 000435 BR 60$ ;BR IF TROUBLE CARRY = CLEAR
1136 010772 016501 000002 20$: MOV TSSR(R5),R1 ;READ TSSR
1137 010776 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
1138 011002 032701 000100 BIT #OFL,R1 ;WAS OFF LINE SET IN TSSR
1139 011006 001402 BEQ 25$ ;BR, IF NO OFL SET
1140 011010 052702 000100 BIS #OFL,R2 ;MAKE THEM LOOK ALIKE
1141 011014 020201 25$: CMP R2,R1 ;ARE THEY OK
1142 011016 CC1401 BEQ 40$ ;BR, IF EQUAL = OK
1143 011020 003421 BR 60$ ;TROUBLE EXIT
1144 011022 062704 000010 40$: ADD #8,R4 ;POINT TO WRT CHARA DATA PACKET
1145 011026 011403 MOV (R4),R3 ;GET ADDRESS OF MESSAGE BUFFER
1146 011030 032763 000200 000012 BIT #X2.EXTF,XST2(R3) ;EXTENDED FEATURES BIT SET?
1147 011036 001402 BEQ 45$ ;BR IF NO
1148 011040 005237 002220 INC EXTFEA ;SET EXTENDED FEATURES SW SWITCH
1149 011044 45$:
1150 011044 032763 000100 000012 BIT #X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
1151 011052 001402 BEQ 50$ ;BR, IF SWITCH NOT SET
1152 011054 005237 002222 INC BENBSW ;SET SOFTWARE SWITCH FOR ENABLED

```


SEQ 0052

1153	011060		50\$:			
1154	011060	000261		SEC		:SET CARRY NO TROUBLE
1155	011062	000401		BR	70\$:EXIT
1156	011064	000241	60\$:	CLC		:CARRY CLEAR = ERROR
1157	011066	016500 000002	70\$:	MOV	TSSR(R5),R0	:RETURN TSSR CONTENTS
1158	011072	000207		RTS	PC	:RETURN
1159						
1160						

```

1162 .SBTTL REWIND - POSITION TAPE (REWIND) COMMAND
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190 011074
1191 011074
1192 011100 012704 011170
1193 011104 010465 000000
1194 011110 012703 000550
1195 011114 004737 016330
1196 011120 103417
1197 011122
      011122 012727 000372
      011126 000000
      011130 013727 002116
      011134 000000
      011136 005367 177772
      011142 001375
      011144 005367 177756
      011150 001367
1198 011152 005303
1199 011154 001357
1200 011156 000241
1201 011160 010400
1202 011162 000207
1203
1204
1206 011170
1208 011170
1209 011170 102010
1210 011172 000000
1211
1212
  
```

```

      THIS ROUTINE WILL REWIND THE SELECTED TAPE.

      CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
                TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
                SSR TO SET IN THE TSSR

      CALLING SEQUENCE:

      DO A SOFT INIT
      DO A WRITE CHARACTERISTICS
      JSR PC,REWIND

      INPUT:

      R5      FIRST DEVICE UNIBUS ADDRESS

      OUTPUT

      R0      THE CONTENTS OF R4 IS PASSED TO R0

      REWIND::
      SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
      MOV #RWPACK,R4                        ;GET PACKET ADDRESS
      MOV R4,TSDB(R5)                       ;SEND PACKET ADDRESS TO EXECUTE
      MOV #360,R3                            ;ENOUGH TIME FOR 2400' REEL TO REWIND
10$: JSR PC,WAITF                            ;WAIT FOR SSR TO SET
      BCS 20$                               ;LEAVE WHEN SSR IS SET
      DELAY 250.                            ;WAIT FOR .25 SECONDS
      MOV #250.,(PC)+
      .WORD 0
      MOV L$DLY,(PC)+
      .WORD 0
      DEC -6(PC)
      BNE -4
      DEC -22(PC)
      BNE -20
      DEC R3                                ;BUMP COUNTER DOWN
      BNE 10$                              ;KEEP GOING
      CLC                                  ;CLEAR CARRY TO SET ERROR
20$: MOV R4,R0                            ;PASS THE PACKET ADDRESS
      RTS PC                               ;RETURN

      RWPACK: .=<.+10>8177770
      .WORD 102010                        ;POSITION COMMAND (REWIND)
      .WORD 0                            ;NOT USED
  
```

TSV3 - GLOBAL AREAS MACRO M1113 25-MAY-82 08:43 PAGE 30-1 C 5
REWIND - POSITION TAPE (REWIND) COMMAND

SEQ 0054

1213
1214
1215

T
M

.SBTTL CKRAM - COMPARE RAM TO I/O PACKET

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

: INPUT:

: R4 ADDRESS OF THE COMMAND PACKET
: R5 FIRST DEVICE UNIBUS ADDRESS

: OUTPUT:

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

: IMPLICIT OUTPUT:

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

: SIDE EFFECTS:

: THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE

: -

CKRAM::

SAVREG

MOV #RAMDATA,R1

MOV #RMPKTBEGR2

CLR R3

JSR PC,CHKTSSR

MOVB #0,TSDB(R5)

10\$:

JSR PC,CHKTSSR

MOV R2,TSDB(R5)

JSR PC,CHKTSSR

MOVB TSBA(R5),(R1)

CMPB (R1)+,(R4)+

BEQ 20\$

INC R3

20\$:

INC R2

CMP R2,#RMPKTEND

BLE 10\$

TST R3

BEQ 30\$

CLC

BR 50\$

30\$:

SEC

50\$:

MOV #8.,RAMSIZ

RTS PC

: SAVE THE GENERAL REGISTERS

: ADDRESS TO SAVE THE RAM DATA

: BYTE ADDRESS OF FIRST RAM DATA

: CLEAR THE ERROR FLAG

: WAIT FOR SSR

: SET MAINTENANCE MODE

: WAIT FOR SSR TO SET

: SELECT NEXT RAM ADDRESS

: WAIT FOR SSR TO SET

: READ THE RAM DATA

: COMPARE TO EXPECTED

: BRANCH IF OK

: SET ERROR FLAG

: ADDRESS OF NEXT RAM LOCATION

: REACHED END YET ?

: BRANCH TILL ALL READ

: WAS AN ERROR FOUND ?

: BRANCH IF NOT

: CLEAR CARRY TO SHOW ERROR

: AND EXIT

: SHOW GOOD COMPARE

: SETUP RAMSIZ FOR PRAMPKT ROUTINE

: RETURN

1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246 011174
1247 011174
1248 011200 012701 002234
1249 011204 012702 000201
1250 011210 005003
1251 011212 004737 016416
1252 011216 112765 000000 000000
1253 011224 004737 016416
1254 011234 010265 000000
1255 011234 004737 016416
1256 011240 116511 000000
1257 011244 122124
1258 011246 001401
1259 011250 005203
1260 011252 005202
1261 011254 020227 000210
1262 011260 003761
1263 011262 005703
1264 011264 001402
1265 011266 000241
1266 011270 000401
1267 011272 000261
1268 011274 012737 000010 002274
1269 011302 000207
1270

```

1272 .SBTTL CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA
1273
1274
1275 :ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
1276 :MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
1277
1278 :INPUT:
1279
1280 R4 ADDRESS OF THE CHARACTERISTICS DATA
1281 R5 FIRST DEVICE UNIBUS ADDRESS
1282
1283 :OUTPUT:
1284
1285 CARRY SET - RAM MATCHES PACKET
1286 CLR - RAM DOES NOT MATCH PACKET
1287
1288 :IMPLICIT OUTPUT:
1289
1290 THE TABLE RAMDATA IS FILLED WITH THE
1291 DATA HELD IN RAM.
1292 RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
1293
1294 :SIDE EFFECTS:
1295
1296 THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1297
1298 :-
1299
1300 CKRAM2::
1301 SAVREG
1302 MOV #RAMDATA,R1 ;SAVE THE GENERAL REGISTERS
1303 MOV #RMCHBEG,R2 ;ADDRESS TO SAVE THE RAM DATA
1304 CLR R3 ;BYTE ADDRESS OF FIRST RAM DATA
1305 JSR PC,CHKTSSR ;CLEAR THE ERROR FLAG
1306 MOV #0,TSDB(R5) ;WAIT FOR SSR
1307 JSR PC,CHKTSSR ;SET MAINTENANCE MODE
1308 MOV R2,TSDB(R5) ;WAIT FOR SSR TO SET
1309 JSR PC,CHKTSSR ;SELECT NEXT RAM ADDRESS
1310 MOV TSBA(R5),(R1) ;WAIT FOR SSR TO SET
1311 CMPB (R1)+,(R4)+ ;READ THE RAM DATA
1312 BEQ 20$ ;COMPARE TO EXPECTED
1313 INC R3 ;BRANCH IF OK
1314 INC R2 ;SET ERROR FLAG
1315 MOV #8,,RAMSIZ ;ADDRESS OF NEXT RAM LOCATION
1316 TST EXTFEA ;ASSUME EXTFEA NOT SET
1317 BEQ 25$ ;IS THE SOFTWARE EXTENDED FEATURES SET
1318 MOV #10,,RAMSIZ ;BR, IF NOT SET
1319 CMP R2,#RMCHEND ;SET RAMSIZ FOR EXTEND FEATURES
1320 BLE 10$ ;AT END OF EXTENDED BUFFER
1321 BR 27$ ;BR, IF NOT AT END YET
1322 CMP R2,#RMCHEND-2 ;AT END BRANCH
1323 BLE 10$ ;REACHED END YET ?
1324 TST R3 ;BRANCH TILL ALL READ
1325 BEQ 30$ ;WAS AN ERROR FOUND ?
1326 CLC ;BRANCH IF NOT
1327 BR 50$ ;CLEAR CARRY TO SHOW ERROR
1328 SEC ;AND EXIT
          ;SHOW GOOD COMPARE

```

TSV3 - GLOBAL AREAS MACRO M1113 25-MAY-82 08:43 PAGE 32-1 F 5
CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

SEQ 0057

1329 011436 000207
1330

50\$: RTS PC

;RETURN

```

1332 .SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS
1333
1334
1335 :ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
1336 :BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1337 :ERROR PRINT ROUTINES.
1338
1339 :INPUT:
1340
1341 R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1342 R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
1343 R2 EXPD MESSAGE BUFFER ADDRESS
1344
1345 :OUTPUT:
1346
1347 CARRY SET - MESSAGE BUFFERS MATCH
1348 CLR -MESSAGE BUFFERS DON'T MATCH
1349
1350 :IMPLICIT OUTPUT:
1351
1352 EXPMSG BUFFER IS SET TO EXPD DATA
1353 RECVMSG BUFFER IS SET TO RECV DATA
1354 RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1355 RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
1356
1357 01:40 CKMSG::
1358 01:44^ SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1359 011444 010037 002276 MOV R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
1360 011450 010137 002300 MOV R1,RCVLOAD ;SAVE RECV LOW ADDRESS
1361 011454 005737 003126 TST KTENABLE ;TESTING ABOVE 28K?
1362 011460 001403 BEQ 10$ ;BR IF NO
1363 011462 004737 017376 JSR PC,SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN R0
1364 011466 010001 MOV R0,R1 ;GET RETURNED ADDRESS BIASED TO PAR6
1365 011470 005004 10$: CLR R4 ;WORD IN BUFFER
1366 011472 005003 CLR R3 ;CLEAR ERROR SEEN FLAG
1367 011474 010205 MOV R2,R5 ;GET EXPD BUFFER ADDRESS
1368 011476 011264 002314 15$: MOV (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1369 011502 011164 002460 MOV (R1),RCVMSG(R4) ;SAVE RECV FOR ERROR REPORT
1370 011506 022221 CMP (R2)+,(R1)+ ;EXPD EQUAL RECV?
1371 011510 001401 BEQ 25$ ;BR IF YES
1372 011512 005203 INC R3 ;SET ERROR SEEN FLAG
1373 011514 062704 000002 25$: ADD #2,R4 ;POINT TO NEXT WORD ADDRESS
1374 011520 020427 000014 CMP R4,#14 ;DONE FIRST 7 WORDS?
1375 011524 003764 BLE 15$ ;BR IF NO
1376 011526 032765 000200 000012 BIT #X2.EXTF,XST2(R5) ;IS EXTENDED FEATURES SET IN EXPD?
1377 011534 001403 BEQ 50$ ;BR IF NO
1378 011536 020427 000016 CMP R4,#16 ;DONE EXTENDED FEATURES WORD?
1379 011542 003755 BLE 15$ ;BR IF NO
1380 011544 005703 50$: TST R3 ;ANY ERRORS SEEN?
1381 011546 001402 BEQ 55$ ;BR IF NO
1382 011550 000241 CLC ;SET FAILURE
1383 011552 000401 BR 60$
1384 011554 000261 55$: SEC ;SET SUCCESS
1385 011556 000207 60$: RTS PC ;RETURN
1386

```

```

1388 .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
1389
1390
1391
1392 ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
1393 BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1394 ERROR PRINT ROUTINES.
1395
1396 INPUT:
1397 R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1398 R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
1399 R2 EXPD MESSAGE BUFFER ADDRESS
1400 R3 NUMBER OF BYTES TO COMPARE
1401
1402 OUTPUT:
1403
1404 CARRY SET - MESSAGE BUFFERS MATCH
1405 CLR - MESSAGE BUFFERS DON'T MATCH
1406
1407 IMPLICIT OUTPUT:
1408
1409 EXPMSG BUFFER IS SET TO EXPD DATA
1410 RECVMSG BUFFER IS SET TO RECV DATA
1411 RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1412 RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
1413
1414
1415 CKMSG2::
1416 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1417 CMP R3,#RECVMSG-EXPMSG;00D IS COUNT ABOVE MAX ALLOWED?
1418 BLE 5$ ;00D BP IF NO
1419 MOV #RECVMSG-EXPMSG,R3;00D
1420 PRINTF #DEBUGMSG ;00D
1421 MOV #DEBUGMSG,-(SP)
1422 MOV #1,-(SP)
1423 MOV SP,R0
1424 TRAP C$PNTF
1425 ADD #4,SP
1426 MOV R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
1427 MOV R1,RCVLOADD ;SAVE RECV LOW ADDRESS
1428 TST KTENABLE ;TESTING ABOVE 28K?
1429 BEQ 10$ ;BR IF NO
1430 JSR PC,SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN R0
1431 MOV R0,R1 ;GET RETURNED ADDRESS BIASED TO PAR6
1432 CLR R4 ;WORD IN BUFFER
1433 CLR R5 ;CLEAR ERROR SEEN FLAG
1434 MOV# (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1435 MOV# (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
1436 CMPB (R2)+,(R1)+ ;EXPD EQUAL RECV?
1437 BEQ 25$ ;BR IF YES
1438 INC R5 ;SET ERROR SEEN FLAG
1439 ADD #1,R4 ;POINT TO NEXT BYTE
1440 CMP R4,R3 ;DONE ALL BYTES?
1441 BGE 50$ ;BR IF YES
1442 BR 15$ ;DO NEXT BYTE
1443 50$: TST R5 ;ANY ERRORS SEEN?
1444 BEQ 55$ ;BR IF NO

```



```

1440 011702 000241          CLC          ;SET FAILURE
1441 011704 000401          BR          60$      ;
1442 011706 000261          55$: SEC          ;SET SUCCESS
1443 011710 000207          60$: RTS          PC      ;RETURN
1444
1445 011712      120      122      117 DEBUGMSG: .ASCIIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';@@D
1446 012002      045      116      045 FERCM: .ASCII /%N% ***/
1447 012013      040      040      124 ERCM: .ASCIIZ / TSSR ERROR CODE REC'D = /
1448 012046      056      056      056 SIMSG: .ASCIIZ /.... AFTEP DOING SOFT INIT/
1449 012101      124      105      123 TINERR: .ASCIIZ /TEST: .../
1450
          .EVEN
  
```

```

1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468 012114
      012114
1469 012114 004737 006020
1470 012120 004737 017262
1471 012124
      012124
      012124 104423
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484 012126
      012126
1485 012126 004737 006020
1486 012132 012700 000004
1487 012136 004737 007446
1488 012142
      012142
      012142 104423
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501 012144
      012144

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

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

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

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

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

      BGNMSG  PKTGETS
PKTGETS::

```

```

1502 012144 004737 006020      JSR    PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1503 012150 012700 000002      MOV     #2,R0        ;NO. OF WORDS IN GET STATUS PACKET
1504 012154 004737 007446      JSR     PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1505 012160
      012160
      012160 104423      L10005:  TRAP    C$MSG

1506
1507
1508
1509      ;+
1510      ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1511      ;
1512      ;INPUTS:
1513      ;
1514      ;      R1      TSSR CONTENTS
1515      ;      R4      ADDRESS OF COMMAND PACKET
1516      ;-
1517 012162      BGNMSG  SFFMSG
      012162
1518 012162 004737 006020      SFFMSG:: JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1519 012166
      012166
      012166 104423      L10006:  TRAP    C$MSG

1520
1521
1522      .SBTTL  PKTMES - PRINT TSSR AND MESSAGE BUFFER
1523      ;+
1524      ;
1525      ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
1526      ;BUFFER FOR ERROR REPORTS
1527      ;
1528      ;INPUTS:
1529      ;
1530      ;      R1      CONTENTS OF TSSR
1531      ;      R2      LOW ORDER MESSAGE BUFFER
1532      ;      R3      HIGH ORDER MESSAGE BUFFER ADDRESS
1533      ;      NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
1534      ;-
1535 012170      BGNMSG  PKTMES
      012170
1536 012170 004737 006020      PKTMES:: JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR
1537 012174 010200
1538 012176 010301
1539 012200 004737 014322      MOV     R2,R0        ;LOW ORDER ADDRESS
1540 012204
      012204
      012204 104423      MOV     R3,R1        ;HIGH ORDER ADDRESS
1541
      L10007:  JSR     PC,PRMESS      ;PRINT THE MESSAGE BUFFER
      TRAP    C$MSG
  
```

```

1543 .SBTTL ADDSSR - PRINT TEST ADDRESS AND TSSR
1544 :+
1545 :PRINT ROUTINE TO PRINT THE CONTENTS OF
1546 :TSSR AND A MEMORY TEST ADDRESS
1547 :
1548 :INPUTS:
1549 :
1550 :      R5      FIRST DEVICE UNIBUS ADDRESS
1551 :      ERRHI   HIGH ORDER MEMORY TEST ADDRESS
1552 :      ERRLO   LOW ORDER MEMORY TEST ADDRESS
1553 :-
1554 :
1555 012206      BGNMSG  ADDSSR
1556 012206      ADDSSR::
1557 012206 004737 010352      JSR      PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
1558 012212 016501 000002      MOV      TSSR(R5),R1      ;GET CURRENT TSSR
1559 012216 004737 006020      JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1560 012222      ENDMSG
1561 012222      L10010:
1562 012222 104423      TRAP      C$MSG
1563 :
1564 .SBTTL MSGEXP - PRINT WRITE CHAR. EXPD-RCV MESSAGE BUFFERS
1565 :+
1566 :PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
1567 :
1568 :IMPLICIT INPUTS:
1569 :
1570 :      EXPMSG - EXPECTED MESSAGE BUFFER
1571 :      RECMSG - RECEIVED MESSAGE BUFFER
1572 :      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1573 :      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1574 :-
1575 012224      BGNMSG  MSGEXP
1576 012224      MSGEXP::
1577 012224 012700 000007      MOV      #7,R0      ;ASSUME NO EXT FEATURES
1578 012230 005737 002220      TST      EXTFEA      ;EXT FEATURES SET?
1579 012234 001402      BEQ      S$      ;BR IF NO
1580 012236 012700 000010      MOV      #8,R0      ;EXT FEATURE BUFFER IS 8 WORDS
1581 012242 004737 014632      JSR      PC,PRMSGEXP      ;PRINT EXPD/RCV MESSAGE BUFFERS
1582 012246      ENDMSG
1583 012246      L10011:
1584 012246 104423      TRAP      C$MSG

```

```

1584 .SBTTL FIFEXP - PRINT FIFO EXP/RECV DATA
1585
1586
1587 :PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
1588
1589 R1 - BYTE COUNT
1590
1591 :IMPLICIT INPUTS:
1592
1593 EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
1594 RECMSG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
1595
1596 BGNMSG FIFEXP
1597 FIFEXP::
1598 PRINTX #FIF1MSG,R1 ;PRINT BYTES TRANSFERRED
1599 MOV R1,-(SP)
1600 MOV #FIF1MSG,-(SP)
1601 MOV #2,-(SP)
1602 MOV SP,R0
1603 TRAP C$PNTX
1604 ADD #6,SP
1605 PRINTX #FIF2MSG ;PRINT HEADER MSG
1606 MOV #FIF2MSG,-(SP)
1607 MOV #1,-(SP)
1608 MOV SP,R0
1609 TRAP C$PNTX
1610 ADD #4,SP
1611 MOV R1,R0 ;GET BYTE COUNT
1612 JSR PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
1613 ENDMMSG
1614
1615 L10012:
1616 TRAP C$MSG
1617 .ASCIZ 'XNXA NUMBER OF BYTES TRANSFERRED = %d2'
1618 .ASCIZ 'XNXA FIFO DATA BYTES IN ERROR:'
1619 .EVEN
1620
1621
1622
1623
1624
1625

```

```

1607          .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
1608          :+
1609          :PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1610          :
1611          :IMPLICIT INPUTS:
1612          :
1613          :EXPMSG - EXPECTED MESSAGE BUFFER
1614          :RCMSG - RECEIVED MESSAGE BUFFER
1615          :RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1616          :RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1617          :-
1618
1619          BGNMSG MSGSTAT
1620          MSGSTAT:
1621          MOV #STATCOD,R1 ;ASCII ADDRESS TABLE
1622          10$: MOV (R1)+,R0 ;DONE ALL MSG LINES?
1623          BEQ 20$ ;BR IF YES
1624          PRINTX R0 ;PRINT STATUS BIT NAMES
1625          MOV R0,-(SP)
1626          MOV #1,-(SP)
1627          MOV SP,R0
1628          TRAP C$PNTX
1629          ADD #4,SP
1630          BR 10$ ;DO ANOTHER MSG LINE
1631          20$: MOV #10,R0 ;NUMBER OF WORDS IN A READ STATUS BUFFER
1632          JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
1633          ENDMSG
1634          L10013:
1635          TRAP C$MSG
1636
1637          STATCOD: .WORD 1$,2$,3$,4$,5$,6$,0
1638          1$: .ASCIIZ 'NNA Tape Bus Signals in Word #8:'
1639          2$: .ASCIIZ 'NNA PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
1640          3$: .ASCIIZ 'NNA IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
1641          4$: .ASCIIZ 'NNA IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
1642          5$: .ASCIIZ 'NNA Tape Bus Signals in Word #9:'
1643          6$: .ASCIIZ 'NNA DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
1644          .EVEN
1645
1646          .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
1647          :+
1648          :PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1649          :
1650          :IMPLICIT INPUTS:
1651          :
1652          :EXPMSG - EXPECTED MESSAGE BUFFER
1653          :RCMSG - RECEIVED MESSAGE BUFFER
1654          :RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1655          :RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1656          :-
1657
1658          BGNMSG MSGLOOP
1659          MSGLOOP:
1660          MOV #LOOPCOD,R1 ;ASCII ADDRESS TABLE

```

```

1655 013150 012100      10$:  MOV      (R1)+,R0      ;DONE ALL MSG LINES?
1656 013152 001410      BEQ      20$              ;BR IF YES
1657 013154      PRINTX  R0              ;PRINT STATUS BIT NAMES
      013154 010046      MOV      R0,-(SP)
      013156 012746 000001  MOV      #1,-(SP)
      013162 010600      MOV      SP,R0
      013164 104415      TRAP     C$PNTX
      013166 062706 000004  ADD      #4,SP
1658 013172 000766      BR       10$              ;DO ANOTHER MSG LINE
1659 013174 012700 000012 20$:  MOV      #10.,R0      ;NUMBER OF WORDS IN A READ STATUS BUFFER
1660 013200 004737 014632  JSR      PC,PRMSGEXP      ;PRINT EXPD/RECV MESSAGE BUFFERS
1661 013204      ENDMSG
      013204      L10014:
      013204 104423      TRAP     C$MSG
1662
1663 013206 013226 013301 013400 LOOPCOD: .WORD 1$,2$,3$,4$,5$,6$,7$,0
1664 013226      045      116      045 1$: .ASCIIZ '%N% Tape Bus Loopback Signals in Word #8:'
1665 013301      045      116      045 2$: .ASCIIZ '%N% PARERR<15> IRESV2<14> IRESV1<13>'
1666 013400      045      116      045 3$: .ASCIIZ '%N% IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
1667 013477      045      116      045 4$: .ASCIIZ '%N% IWFM =>IFMK<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
1668 013576      045      116      045 5$: .ASCIIZ '%N% ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDPA <04>'
1669 013675      045      116      045 6$: .ASCIIZ '%N% IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
1670 013774      045      116      045 7$: .ASCIIZ '%N% IGO =>IFPT<00>'
1671
1672      .EVEN

```

```

1674 .SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
1675
1676
1677 PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1678
1679
1680 IMPLICIT INPUTS:
1681
1682 EXPMSG - EXPECTED MESSAGE BUFFER
1683 RECMSG - RECEIVED MESSAGE BUFFER
1684 RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1685 RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1686
1687 014022 BGNMSG MSGSUB
      014022 MSGSUB::
1688 014022 012700 000012 MOV #10,R0 ;SIZE OF WRITE SUBSYSTEM BUFFER
1689 014026 004737 014632 JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
1690 014032 ENDMMSG
      014032 L10015:
      014032 104423 TRAP CSMSG

1691
1692
1693
1694
1695
1696 .SBTTL MEMADD - PRINT MEMORY ADDRESS DATA ERROR
1697
1698
1699 PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
1700
1701
1702 IMPLICIT INPUTS:
1703
1704 ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
1705 ERRLO - MEMORY ERROR LOW ORDER ADDRESS
1706 EXP - EXPECTED DATA
1707 RECV - RECEIVED DATA
1708
1708 014034 BGNMSG MEMADD
      014034 MEMADD::
1709 014034 004737 010236 JSR PC,PRIADD ;PRINT MEMORY ADDRESS IN ERROR
1710 014040 013701 002224 MOV EXPD,R1 ;GET EXPD DATA
1711 014044 013702 002226 MOV RECV,R2 ;GET RECEIVED DATA
1712 014050 004737 010020 JSR PC,PRI XOR ;PRINT EXPD/RCV
1713 014054 ENDMMSG
      014054 L10016:
      014054 104423 TRAP CSMSG
1714

```



```

1716 .SBTTL PRAMPKT - PRINT RAM AND PACKET DATA
1717
1718
1719 :PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1720 :WHEN THE RAM DATA DOES NOT MATCH.
1721
1722 :INPUTS:
1723
1724 :R4 POINTER TO COMMAND PACKET
1725
1726 :IMPLICIT INPUTS:
1727
1728 :RAMDATA DATA AS READ FROM THE RAM
1729 :RAMSIZ NUMBER OF BYTES IN PACKET
1730 :IF RAMSIZ=0 THEN DEFAULT TO 8.
1731
1732 :IMPLICIT OUTPUTS:
1733
1734 :RAMSIZ SET TO 0
1735
1736 :-
1737 PRAMPKT:
1738 SAVREG
1739 MOV #RAMDATA,R1 ;SAVE R1-R5 UNTIL NEXT RETURN
1740 CLR R2 ;DATA FROM THE RAM
1741 5$: CMPB (R1)+,(R4)+ ;INIT BYTE NUMBER
1742 BNE 7$ ;COMPARE EXPECTED, RECEIVED
1743 FORCERRR 7$,NOTSSR ;BR IF NO MATCH
1744 BR 10$
1745 7$: MOVB -1(R1),R5 ;GET RECV RAM DATA
1746 MOVB -1(R4),R3 ;GET EXPD PACKET DATA
1747 XOR R5,R3 ;XOR EXPD/RECV
1748 BIC #177400,R3 ;LOW BYTE ONLY
1749 MOVB -1(R1),RECV ;GET RECEIVED RAM DATA
1750 MOVB -1(R4),EXPD ;GET EXPECTED RAM DATA
1751 PRINTB #RAMASC,R2,RECV,EXPD,R3
1752 MOV R3,-(SP)
1753 MOV EXPD,-(SP)
1754 MOV RECV,-(SP)
1755 MOV R2,-(SP)
1756 MOV #RAMASC,-(SP)
1757 MOV #5,-(SP)
1758 MOV SP,R0
1759 TRAP C$PNTB
1760 ADD #14,SP
1761 10$: INC R2 ;UPDATE BYTE COUNT
1762 TST RAMSIZ ;DEFAULT TO 8.?
1763 BEQ 15$ ;BR IF YES
1764 CMP R2,RAMSIZ ;DONE ALL BYTES?
1765 BLE 5$ ;BR IF NO
1766 BR 25$
1767 15$: CMP R2,#8. ;DONE DEFAULT NUMBER OF BYTES?
1768 20$: BLT 5$ ;BR IF NO
1769 25$: CLR RAMSIZ ;SET DEFAULT RAMSIZ
1770 RTS PC ;RETURN
1771
1772 045 RAMASC: .ASCII 'XNXA BYTE: XD2XA RAM: X03XA Packet: X03XA XOR:X03'

```

TSV3 - GLOBAL AREAS MACRO M1113 25-MAY-82 08:43 PAGE 40-1 E 6
PRAMPKT - PRINT RAM AND PACKET DATA

SEQ 0069

1764
1765
1766

.EVEN

```

1768 .SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785 014322
1786 014322
1787 014326 010005 003126
1788 014330 005737
1789 014334 001001
1790 014336 005001
1791 014340 010103
1792 014342 006100
1793 014344 006101
1794 014346
    014346 010546
    014350 010146
    014352 012746 014500
    014356 012746 000003
    014362 010600
    014364 104415
    014366 062706 000010
1795 014372
    014372 012746 014545
    014376 012746 000001
    014402 010600
    014404 104415
    014406 062706 000004
1796 014412 005004
1797 014414 010501
1798 014416 010300
1799 014420 001403
1800 014422 004737 017376
1801 014426 010005
1802 014430
    014430 012546
    014432 010446
    014434 012746 014603
    014440 012746 000003
    014444 010600
    014446 104415
    014450 062706 000010
1803 014454 005204
1804 014456 020427 000007
1805 014462 003005

    THIS ROUTINE PRINTS THE CONTENTS OF
    THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE
    TSV-05.
    INPUT:
    R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
    R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
    NOTE: R1 'S IGNORED IF KTENABLE FLAG IS CLEAR
    THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
    -
PRMESS:
    SAVREG
    MOV     R0,R5      ;SAVE THE REGISTERS
    TST     KTENABLE   ;SAVE LOW ORDER ADDRESS
    BNE     10$        ;ADDRESS ABOVE 28K?
    CLR     R1         ;BR IF YES
    MOV     R1,R3      ;SET HIGH ORDER ADDRESS TO 0
    ROL     R0         ;SAVE HIGH ORDER ADDRESS
    ROL     R1         ;SHIFT BIT15 TO C BIT
    PRINTX  #PROASC,R1,R5 ;SHIFT TO HIGH ORDER FOR PRINTOUT
    MOV     R5,-(SP)   ;PRINT MESSAGE BUFFER ADDRESS
    MOV     R1,-(SP)
    MOV     #PROASC,-(SP)
    MOV     #3,-(SP)
    MOV     SP,R0
    TRAP    C$PNTX
    ADD     #10,SP
    PRINTX  #PRIASC
    MOV     #PRIASC,-(SP) ;PRINT HEADER FOR CONTENTS
    MOV     #1,-(SP)
    MOV     SP,R0
    TRAP    C$PNTX
    ADD     #4,SP
    CLR     R4
    MOV     R5,R1      ;NUMBER OF THE NEXT WORD
    MOV     R3,R0      ;COPY LOW ORDER ADDRESS
    BEQ     20$        ;COPY HIGH ORDER ADDRESS
    JSR     PC,SETMAP   ;BR IF NOT ABOVE 28K
    MOV     R0,R5      ;SETUP PAR ADDRESS IN R0
    PRINTX  #PRASC,R4,(R5)+ ;GET PAR FORMAT ADDRESS ABOVE 28K
    MOV     (R5)+,-(SP) ;PRINT THE CONTENTS OF MEMORY BUFFER
    MOV     R4,-(SP)
    MOV     #PRASC,-(SP)
    MOV     #3,-(SP)
    MOV     SP,R0
    TRAP    C$PNTX
    ADD     #10,SP
    INC     R4
    CMP     R4,#7      ;NUMBER OF THE NEXT
    BGT     50$        ;DONE ALL YET ?
    ;BRANCH IF ALL DONE
  
```

1806	014464	002761				BLT	20\$;PRINT FIRST 7 WORDS
1807	014466	032763	000200	000012		BIT	#X2.EXTF,XST2(R3)	;EXTENDED FEATUTES ON ?
1808	014474	001355				BNE	20\$;PRINT EXTENDED STATUS WORD
1809	014476	000207			50\$:	RTS	PC	;RETURN
1810								
1811	014500	045	116	045	PROASC:	.ASCIIZ	'%N% Message Buffer Address = %01%05'	
1812	014545	045	116	045	PR1ASC:	.ASCIIZ	'%N% Message Buffer Contents:'	
1813	014603	045	116	045	PRASC:	.ASCIIZ	'%N% Word%D1%A: %0'	
1814						.EVEN		

```

1816 .SBTTL PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830 014632
1831 014632
1832 014636 010005 002300
1833 014640 013700
1834 014644 010004
1835 014646 013701 002276
1836 014652 006100
1837 014654 006101
1838 014656
1839 014702
1840 014722 005004
1841 014724 012701 002314
1842 014730 012702 002460
1843 014734 011100
1844 014736 011203
1845 014740
1846 014750
1847 015000 005204
1848 015002 020405
1849 015004 002001
1850 015006 000752
1851 015010 000207

ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
RO - NUMBER OF WORDS IN BUFFER

IMPLICIT INPUTS:
EXPMSG - EXPECTED MESSAGE BUFFER
RECMMSG - RECEIVED MESSAGE BUFFER
RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS

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

20$:
50$:

```

```

1852
1853 015012      045      116      045  PRMSG0: .ASCIIZ  '%N% Message Buffer Address = %01%05'
1854 015057      045      116      045  PRMSG1: .ASCIIZ  '%N% Message Buffer Contents:'
1855 015115      045      116      045  PRMSG2: .ASCIIZ  '%N%      WORD #%D2%      EXPD: %06%      RECV: %06%      XOR: %06'
1856                                     .EVEN
1857

```

```

1859 .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
1860
1861
1862 ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
1863 ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
1864
1865 R0 - NUMBER OF BYTES IN BUFFER
1866
1867 IMPLICIT INPUTS:
1868
1869 EXPMSG - EXPECTED MESSAGE BUFFER
1870 RECMSG - RECEIVED MESSAGE BUFFER
1871
1872 PRBYTEXP::
1873 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1874 MOV R0,R5 ;SAVE NUMBER OF BYTES
1875 CLR PRMNO ;INIT ERROR COUNT
1876 CLR R4 ;NUMBER OF THE CURRENT BYTE
1877 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
1878 MOV #RECMSG,R2 ;GET RECV BUFFER ADDRESS
1879 20$: MOV (R1),R0 ;GET EXPD BYTE
1880 BIC #C<377>,R0 ;CLEAR UPPER BYTE
1881 MOV R0,PRBEXP ;SAVE FOR ERROR REPORT
1882 MOV (R2),R3 ;GET RECV BYTE
1883 BIC #C<377>,R3 ;CLEAR UPPER BYTE
1884 MOV R3,PRBREC ;FOR ERROR REPORT
1885 XOR R0,R3 ;XOR EXPD/RECV
1886 CMPB (R1)+,(R2)+ ;EXPD = RECV?
1887 BEQ 30$ ;BR IF YES
1888 INC PRMNO ;UPDATE ERROR COUNT
1889 000010 CMP PRMNO,#8. ;PRINTED 8?
1890 BHI 30$ ;BR IF YES
1891 27$: PRINTX #PRBMSG,R4,PRBEXP,PRBREC,R3
1892 MOV R3,-(SP)
1893 MOV PRBREC,-(SP)
1894 MOV PRBEXP,-(SP)
1895 MOV R4,-(SP)
1896 MOV #PRBMSG,-(SP)
1897 MOV #5,-(SP)
1898 MOV SP,R0
1899 TRAP C$PNTX
1900 ADD #14,SP
1901 30$: FORCEEXIT 50$ ;@@D
1902 BR 35$ ;@@D
1903 30$: FORCERROR 27$,NOTSSR ;@@D
1904 35$: ;@@D
1905 INC R4 ;NUMBER OF THE NEXT
1906 CMP R4,R5 ;DONE ALL YET?
1907 BGE 50$ ;BR IF YES
1908 BR 20$ ;DO ANOTHER
1909 50$: PRINTX #PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
1910 MOV PRMNO,-(SP)
1911 MOV #PRBTOT,-(SP)
1912 MOV #2,-(SP)
1913 MOV SP,R0
1914 TRAP C$PNTX

```

1902	015410	062706	000006		ADD	#6,SP	
1903	015414	000207			RTS	PC	;RETURN
1904	015416	045	116	045	PRBMjG:	.ASCIZ	'%N% BYTE #D2% EXPD: %03% RECV: %03% XOR: %03'
1905	015503	045	116	045	PRBTOT:	.ASCIZ	'%N% NUMBER OF BYTES IN ERROR = %D2'
1906						.EVEN	
1907	015550	000000			PRBEXP:	.WORD	0 ;EXPD
1908	015552	000000			PRBREC:	.WORD	0 ;RECV
1909							

1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927

015554
015554
015554 004737 010020
015560
015560
015560 104423

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

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

015562
015562
004737 007670
015566
015566
015566 104423
015570
015570
004737 014056
015574
015574
015574 104423

```
.SBTTL EXPBREC - PRINT EXPD/RECV BYTE DATA
:
:PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
:
:INPUTS:
:      R1      RECEIVED DATA BYTE
:      R2      EXPECTED DATA BYTE
:-
EXPBREC: BGNMSG EXPBREC
:      JSR      PC,PRIBXOR      :PRINT THE DATA
:      ENDMSG
L10020: TRAP      C$MSG

.SBTTL RAMERR - PRINT RAM AND PACKET DATA
:
:PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
:
:INPUTS:
:      R4      POINTER TO COMMAND PACKET
:
:IMPLICIT INPUTS:
:      RAMDATA      DATA AS READ FROM THE RAM
:      RAMSIZ        NUMBER OF BYTES IN PACKET
:                   IF RAMSIZ=0 THEN DEFAULT TO 8.
:
:IMPLICIT OUTPUTS:
:      RAMSIZ      SET TO 0
:-
RAMERR: BGNMSG RAMERR
:      JSR      PC,PRAMPKT      :PRINT RAM/PACKET DATA
:      ENDMSG
L10021: TRAP      C$MSG

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

```

1980
1981      R4      POINTER TO COMMAND PACKET
1982
1983      IMPLICIT INPUTS:
1984
1985      RAMDATA   DATA AS READ FROM THE RAM
1986      RAMSIZ    NUMBER OF BYTES IN PACKET
1987              IF RAMSIZ=0 THEN DEFAULT TO 8.
1988      ERRHI     HIGH ORDER TEST ADDRESS
1989      ERRLO     LOW ORDER TEST ADDRESS
1990
1991      IMPLICIT OUTPUTS:
1992
1993      RAMSIZ    SET TO 0
1994
1995
1996      BGNMSG    RAMTADD
1997      RAMTADD:: JSR      PC,PRITADD      ;PRINT TEST ADDRESS
1998                JSR      PC,PRAMPKT     ;PRINT RAM/PACKET DATA
1999                ENDMMSG
2000
2001      L10022:    TRAP      C$MSG
2002
2003      .SBTTL    RAMEXP - PRINT RAM EXPD/RECV DATA
2004
2005      PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2006
2007      INPUTS:
2008
2009      R1        RECEIVED DATA
2010      R2        EXPECTED DATA
2011      R4        CONTROLLER RAM ADDRESS
2012
2013
2014      BGNMSG    RAMEXP
2015      RAMEXP::  BIC      #^C<377>,R1      ;SAVE EXPD RAM DATA BYTE
2016                BIC      #^C<377>,R2      ;SAVE EXPD RAM DATA BYTE
2017                JSR      PC,PRIRAM        ;PRINT THE RAM ADDRESS
2018                JSR      PC,PRIXOR        ;PRINT THE DATA
2019                ENDMMSG
2020      L10023:    TRAP      C$MSG
2021
2022      .SBTTL    TIMEXP - PRINT TIMER A,B AND EXP/REC
2023
2024      PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2025      AND TIMER A,B HEADER MESSAGE
2026
2027      INPUTS:
2028
2029      R1        RECEIVED DATA
2030      R2        EXPECTED DATA
  
```

```

2031      ;:-
2032
2033      015632      BGNMSG  TIMEXP
          015632      TIMEXP::
2034      015632      PRINTX  #TIMSG0      ;PRINT HEADER
          015632      012746  015660      MOV      #TIMSG0,-(SP)
          015636      012746  000001      MOV      #1,-(SP)
          015642      010600      MOV      SP,R0
          015644      104415      TRAP     C$PNTX
          015646      062706  000004      ADD      #4,SP
2035      015652      004737  010020      JSR      PC,PRIXOR      ;PRINT THE DATA
2036      015656      L10024:
          015656      104423      TRAP     C$MSG
2037
2038
2039      015660      045      116      045  TIMSG0: .ASCIZ  '%N%A TIMER A STATUS IS IN BIT 3%N%A TIMER B STATUS IS IN BIT 2'
2040      .EVEN
  
```

```

2042                                .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
2043
2044                                :+
2045                                :PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
2046                                :
2047                                :INPUTS:
2048                                :
2049                                :
2050                                :      R1      CONTENTS OF TSSR
2051                                :      R2      DATA WRITTEN (8 BITS)
2052                                :
2053                                :-
2054
2055                                BGNMSG  BADSSR
2056                                BADSSR:
2057                                MOV      R2,-(SP)                ;SAVE DATA TRANSFERRED
2058                                BIC      #177400,R2            ;GET JUST ONE BYTE
2059                                PRINTB   #XFERASC,R2
2060                                MOV      R2,-(SP)
2061                                MOV      #XFERASC,-(SP)
2062                                MOV      #2,-(SP)
2063                                MOV      SP,R0
2064                                TRAP     C$PNTB
2065                                ADD      #6,SP
2066                                MOV      (SP)+,R2                ;RESTORE R2
2067                                JSR      PC,PRITSSR              ;DECODE TSSR CONTENTS
2068                                ENDMSG
2069
2070                                L10025:
2071                                TRAP     C$MSG
2072                                .ASCIZ   '%N% Data Transferred = %03'
2073
2074                                045  XFERASC:

```

2065
2066
2067
2068
2069
2070
2071

.SBTTL GLOBAL SUBROUTINES SECTION

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

```

2073 .SBTTL SOFINIT - SOFT INITIALIZE OF CONTROLLER
2074
2075 ;+
2076 ;ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
2077 ;BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
2078 ;THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
2079 ;DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
2080
2081 ;INPUTS:
2082
2083 R5 ADDRESS OF FIRST REGISTER
2084
2085 ;OUTPUTS:
2086
2087 R0 CONTENTS OF TSSR, IF ERROR
2088 CARRY SET IF INIT WAS OKAY
2089 CLEAR IF FATAL ERROR
2090
2091 ;CALLING SEQUENCE:
2092
2093 MOV #ADDRESS,R5
2094 JSR PC,SOFINIT
2095 BCS CONTINUE
2096 ERRDF ;REPORT FATAL ERROR
2097
2098 ;-
2099
2100 SOFINIT::
2101 016054 SAVREG ; SAVE THE REGISTERS
2102 016054 MOV #0,TSSR(R5) ; DO THE INIT.
2103 016060 012765 000000 000002 JSR PC,WAITF ; WAIT FOR SSR
2104 016066 004737 016330 MOV TSSR(R5),R0 ; GET THE TSSR REGISTER
2105 016072 016500 000002 MOV R0,R4 ; TSSR CONTENTS
2106 016076 010004 BIC #^C<HIADDR!OFL>,R4
2107 016100 042704 176277 BIS #SSR!NBA,R4 ; R4 HAS EXPECTED CONTENTS
2108 016104 052704 002200 CMP R4,R0 ; ONLY EXPECTED BITS SET ?
2109 016110 020400 BEQ S$ ; BRANCH IF OKAY
2110 016112 001402 CLC ; CLEAR THE CARRY FOR ERROR
2111 016114 000241 BR 10$ ; GO TO EXIT
2112 016116 000401 S$: SEC ; SET THE CARRY BIT
2113 016120 000261 10$: RTS ; RETURN TO CALLER
2114 016122 000207 PC

```

```

2116 .SBTTL CHKAMB - CHECK TSSR FOR AMBIGUITY
2117
2118 :+
2119 :THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
2120 :FOR AMBIGUITY
2121 :
2122 :INPUT:
2123 :
2124 :      R0      CONTENTS OF TSSR
2125 :
2126 :OUTPUT:
2127 :
2128 :      R0      CONTENTS OF TSSR
2129 :
2130 :      CARRY   SET - NO AMBIGUITY
2131 :             CLR - AMBIGUOUS CONTENTS
2132 :
2133 :-
2134
2135
2136 CHKAMB: SAVREG          ;SAVE THE GENERAL REGISTERS
2137          MOV R0,R4      ;CONTENTS OF TSSR
2138          BIT #SC,R0     ;IS BIT 15 SET ?
2139          BNE 5$         ;BRANCH IF YES
2140          BIT #C<NBA!OFL!SSR!HIADDR>,R0 ;ANY OTHER BITS SET ?
2141          BNE 40$        ;MUST BE AN ERROR
2142          BR 45$         ;RETURN WITH SUCCESS
2143          5$: BIT #SSR,R0 ;IS READY BIT SET ?
2144          BNE 10$        ;BRANCH IF READY BIT IS SET.
2145          BIT #BITS,R0   ;IS FATAL ERROR BIT SET ?
2146          BEQ 40$        ;ERROR IF NOT
2147          BIC #^CTERCLS,R4 ;CLEAR ALL BUT TERMINATION CODE
2148          CMP R4,#16     ;ALL THREE BITS MUST BE SET
2149          BNE 40$        ;ERROR IF NOT SET
2150          BR 45$         ;OK IF ALL ARE SET
2151          10$: BIT #BITS,R0 ;IS FATAL ERROR BIT SET ?
2152          BEQ 45$        ;ERROR IF BIT IS SET WITH SSR
2153          BIT #BIT2!BIT1,R0 ;IS THIS A FUNCTION REJECT
2154          BNE 45$        ;BR, IF TSSR IS OK
2155          40$: CLC        ;AMBIGUOUS CONTENTS
2156          BR 50$
2157          45$: SEC        ;SHOW SUCCESS - NO AMBIGUITY
2158          50$: RTS PC    ;RETURN TO CALLER
2159
2160
2161 016124
2162 016124
2163 016130 010004
2164 016132 032700 100000
2165 016136 001004
2166 016140 032700 174077
2167 016144 001023
2168 016146 000424
2169 016150 032700 000200
2170 016154 001011
2171 016156 032700 000040
2172 016162 001414
2173 016164 042704 177761
2174 016170 020427 000016
2175 016174 001007
2176 016176 000410
2177 016200 032700 000040
2178 016204 001405
2179 016206 032700 000006
2180 016212 001002
2181 016214 000241
2182 016216 000401
2183 016220 000261
2184 016222 000207
2185

```



```

2162          .SBTTL ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
2163          :
2164          : DEFAULT DISPLAY INTERRUPT HANDLERS.
2165          : IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2166          : OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
2167          :
2168          :
2169          : BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2170          :
2171          000200          IOKCKIN=BIT7      ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
2172          000001          IOKSTP=BIT0      ; EXPECT "STOP" INTERRUPT.
2173          :
2174          : INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
2175          016224          000          INTMASK: .BYTE 0
2176          : INTERRUPT FLAG -- SAYS WE GOT 0: (IF POSITIVE)
2177          016225          000          INTFLAG: .BYTE 0
2178          :
2179          : SAVED INTERRUPT VECTOR:
2180          016226          000000          INTVEC: .WORD 0
2181          : SAVE CPU PC
2182          016230          000000          INTCPC: .WORD 0
2183          :
2184          : SUBROUTINE TO ENABLE INTERRUPTS:
2185          016232          010046          ENAINT: MOV      R0,-(SP)          ;SAVE R0
2186          016234          013700          002202          MOV      IVEC,R0          ;GET POINTER TO VECTORS
2187          016240          012720          016276          MOV      #INTR,(R0)+      ;SET UP INTERRUPT VECTOR
2188          016244          012720          000340          MOV      #PRI07,(R0)+
2189          016250          012600          MOV      (SP)+,R0          ;RESTORE R0
2190          016252          011646          MOV      (SP),-(SP)
2191          016254          012766          000000          000002          MOV      #0,2(SP)          ;SET CPU TO LEVEL 0
2192          016262          000002          RTI
2193          :
2194          : SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2195          016264          011646          DSBINT: MOV      (SP),-(SP)
2196          016266          012766          000340          000002          MOV      #PRI07,2(SP)
2197          016274          000002          RTI
2198

```

```

2200 .SBTTL INTR - INTERRUPT HANDLERS
2201
2202 016276 BGNSRV INTR ;DEFINE INTERRUPT ENTRY
    016276 INTR::
2203 016276 012737 000001 002216 MOV #1,INTRECV ;SET FLAG TO SHOW INTERRUPT RECEIVED
2204 016304 105037 016225 CLRB INTFLAG ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2205 016310 132737 000001 016224 BITB #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2206 016316 001003 BNE 1$ ;BR IF YES
2207 016320 152737 000001 016225 BISB #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2208
2209 ;SAVE REGISTERS, MSG BUFFER, ETC.
2210 016326 1$:
2211 016326 ENDSRV
    016326 L10026:
    016326 000002 RTI
2212
2213
    
```

```

2215 .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
2216 :
2217 : SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2218 :
2219 : INPUTS:
2220 :
2221 : R5 ADDRESS OF FIRST DEVICE REGISTER
2222 :
2223 : OUTPUTS:
2224 :
2225 : (C) CONTENTS OF LAST TSSR READ
2226 : CARRY SET - READY BIT SET
2227 : CLR - TIMEOUT WAITING FOR READY
2228 :
2229 016330 000401 WAITF:: BR 1$ ;NOP WHEN SUPER FIXED
2230 016332 104422 BREAK ; DO A SUPVSR BREAK FIRST.
2231 016334 012746 003000 TRAP C$BRK ;300 MSEC TIMER
2232 016340 016500 000002 1$: MOV #3000,-(SP) ;READ THE TSSR REGISTER
2233 016344 105700 2$: MOV TSSR(R5),R0 ;TEST FOR READY BIT SET
2234 TSTB R0
2235 016346 100420 BMI 3$ ; EXIT ON STOP FLAG.
2236 016350 DELAY 1 ; WAIT 100 USEC
2237 016350 012727 000001 MOV #1,(PC)+
2238 016354 000000 .WORD 0
2239 016356 013727 002116 MOV LSDLY,(PC)+
2240 016362 000000 .WORD 0
2241 016364 005367 177772 DEC -6(PC)
2242 016370 001375 BNE -4
2243 016372 005367 177756 DEC -22(PC)
2244 016376 001367 BNE -20
2245 016400 005316 DEC (SP) ;REDUCE DELAY COUNT
2246 016402 001356 BNE 2$ ;RETRY UNTIL TIMER EXPIRES
2247 016404 000241 CLC ; C = 0, CONTROLLER STILL RUNNING...
2248 016406 000401 BR 4$ ;...OR HUNG-UP AFTER 300 MSEC.
2249 016410 000261 3$: SEC ; C = 1, CONTROLLER IS STOPPED.
2250 016412 005326 4$: DEC (SP)+ ;RESTORE STACK WITHOUT CHANGING CARRY BIT
2251 016414 000207 RTS PC
  
```

2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263

.SBTTL CHKTSSR - CHECK TSSR FOR READY

```

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

```

2264 016416
2265 016416 004737 016330
2266 016422 103014
2267 016424 004737 016124
2268 016430 103006
2269 016432 032700 100000
2270 016436 001405
2271 016440 032700 074000
2272 016444 001402
2273 016446 000241
2274 016450 000401
2275 016452 000261
2276 016454 000207

```

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

```

```

2278 .SBTTL XNXM - CHECK FOR NONEXISTENT MEMORY
2279
2280 ;+
2281 ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
2282 ; ON RETURN, IF 'C' = 1, (R1) = NEXM ADDRESS.
2283 ; 'C' = 0, ALL ADDRESSES OK.
2284
2285 ;CALL: MOV ADR1,R1
2286 ; MOV ADR2,R2
2287 ; JSR PC,NXM
2288 ; RETURN ;TEST 'C' AND PROCEED.
2289 016456 012737 016510 000004 XNXM: MOV #2$,@#4 ; SET BUSERR VECTOR.
2290 016464 012737 000200 000006 MOV #PRI04,@#6
2291 016472 005003 CLR R3 ; FLAG.
2292 016474 005711 1$: TST (R1) ; TEST THE ADDRESS(ES).
2293 ; IF ANY TRAP, CONTINUE AT 2$.
2294 016476 020102 CMP R1,R2 ; OTHERWISE, CONTINUE HERE.
2295 016500 001407 BEQ 3$ ; BR IF FINISHED (NO NEXM'S).
2296 016502 062701 000002 ADD #2,R1 ; SET NEXT ADDRESS...
2297 016506 000772 BR 1$ ; ...AND CONTINUE.
2298
2299 016510 005103 2$: COM R3 ; GOT ONE, SET FLAG...
2300 016512 012716 016520 MOV #3$, (SP)
2301 016516 000002 RTI ; ...AND DISMISS INTERRUPT...
2302 016520 012700 000004 3$: CLRVEC #4 ; ...AND GIVE BACK THE VECTOR.
2303 016526 005703 MOV #4,R0
2304 016530 001401 TRAP C$CVEC
2305 016532 000261 TST R3 ; DID WE CATCH ONE ??
2306 016534 000207 BEQ .+4 ; NO, 'C' = 0, SKIP NEXT.
2307 ; YES, 'C' = 1, (R1) = NEXM ADDR.
2308
2309
2310
2311 .SBTTL TSTLOOP - CHECK ITERATION COUNT
2312
2313 ;+
2314 ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
2315 ; EXIT WITH 'C' SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
2316 ; LOOP COUNTER IS SET BY 'BEGIN.TEST' MACRO.
2317
2318 ; CALL: LOOPTO ARG
2319
2320 016536 005737 002162 TSTLOOP: TST NOITS ; ITERATIONS INHIBITED?
2321 016542 001006 BNE 1$ ; YES.
2322 016544 005737 002176 TST QVP ; NO.
2323 016550 100403 BMI 1$ ; LOOPS DISALLOWED IN QUICK PASS.
2324 016552 005337 002210 DEC LOOPCNT ; BUMP LOOP COUNTER.
2325 016556 001002 BNE 2$
2326 016560 000241 1$: CLC ; LOOP DISALLOWED, OR DONE.
2327 016562 000401 BR 3$
2328 016564 000261 2$: SEC ; LOOP ENABLED.
2329 016566 000207 3$: RTS PC

```

2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358

.SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
INCREMENT 'TESTK' TO INDICATE THE NUMBER OF TESTS
IN THE CURRENT RUN SEQUENCE.
CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.

INPUT:

R0 POINTER TO TEST ID ASCIZ STRING

OUTPUT:

R5 ADDRESS OF FIRST DEVICE REGISTER

IMPLICIT OUTPUTS:

TSTCNT UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART

SIDE EFFECTS:

INTERRUPT LEVEL IS RASIED TO LEVEL OF
THE DEVICE UNDER TEST

TSTSETUP::

2359 016570
2360 016570 010046
2361 016572 005037 003146
2362 016576 005037 017036
2363 016602 005037 005766
2364 016606 105037 016224
2365 016612 013700 002174
2366 016616 006300
2367 016620 005737 003106
2368 016624 001430
2369 016626 100010
2370 016630 052760 160000 003170
2371 016636
016636 104455
016640 000001
016642 003734
016644 005732
2372 016646 000407
2373 016650 052760 160001 003170 3\$:
2374 016656
016656 104455
016660 000002
016662 004331
016664 000000
2375 016666 012737 177777 003104 2\$:
2376 016674
016674 013700 002174
016700 104451
2377 016702

```

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

```

2378	016702	104444		TRAP	C\$DCLN	
2379	016704	000423		BR	5\$	
2380	016706		4\$:	RFLAGS	R0	: GET THE OPERATOR FLAGS.
	016706	104421		TRAP	C\$RFLA	
2381	016710	032700	001000	BIT	#PNT,R0	: PRINT THE TEST NUMBERS?
2382	016714	001412		BEQ	1\$: BR IF NO
2383	016716	011600		MOV	(SP),R0	:GET THE ID MESSAGE
2384	016720			PRINTF	#TNAM,R0	:DISPLAY THE TEST ID
	016720	010046		MOV	R0,-(SP)	
	016722	012746	016764	MOV	#TNAM,-(SP)	
	016726	012746	000002	MOV	#2,-(SP)	
	016732	010600		MOV	SP,R0	
	016734	104417		TRAP	C\$PNTF	
	016736	062706	000006	ADD	#6,SP	
2385	016742	005237	002206	1\$:	INC	TSTCNT
2386	016746			SETPRI	IPRI	: BUMP TEST COUNTER.
	016746	013700	002204	MOV	IPRI,R0	:PRIORITY THAT OF DEVICE
	016752	104441		TRAP	C\$SPRI	
2387	016754	005726		5\$:	TST	(SP)+
2388	016756	013705	002200	MOV	CSRADDR,R5	:FIX UP THE STACK
2389	016762	000207		RTS	PC	: ADDRESS OF TSV REGISTERS ON UNIBUS
2390	016764	045	123	045	TNAM:	.ASCIZ '%SXTXA Test'
2391						.EVEN

```
2393 .SBTTL TSTEND - PRINT ERRORS RECEIVED
2394
2395 : AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2396 : IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2397
2398 TSTEND: RFLAGS R0
2399 017000 104421 TRAP CSRFLA
2400 017002 030027 020000 BIT R0,#IER
2401 017006 001412 BEQ 1$ ; BR IF "IER" NOT SET.
2402 017010 013746 017036 PRINTF #ESUM,ERRK ; PRINT ERROR COUNT.
2403 017014 012746 017040 MOV ERRK,-(SP)
2404 017020 012746 000002 MOV #ESUM,-(SP)
2405 017024 010600 MOV #2,-(SP)
2406 017026 104417 MOV SP,R0
2407 017030 062706 000006 TRAP C$PNTF
2408 017034 000207 1$: ADD #6,SP
2409 RTS PC
2410
2411 ERRK: 0 ; LOCAL ERROR COUNT.
2412 ESUM: .ASCIZ /%A %D%A ERRORS/
2413 EMAXDU: .ASCIZ /ERROR LIMIT REACHED -- DROPPING UNIT/
2414 .EVEN
2415
2416 .SBTTL INCERK - INCREMENT LOCAL ERROR COUNT
2417
2418 :+
2419 : ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
2420 : -
2421 INCERK: INC ERRK ; INCREMENT LOCAL ERROR COUNT
2422 MOV R0,-(SP) ; SAVE R0
2423 MOV UNITN,R0 ; GET UNIT NUMBER,
2424 ASL R0 ; ... AND MAKE IT A WORD OFFSET.
2425 ADD #ERTABL,R0 ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
2426 INC (R0) ; INCREMENT THE DEVICE ERROR COUNT
2427 BIT #7777,(R0) ; DID WE OVERFLOW THE FIELD?
2428 BNE 1$ ; BR IF NO.
2429 DEC (R0) ; YES -- BACK IT UP TO 7777.
2430 1$: MOV (SP)+,R0 ; RESTORE R0
2431 RTS PC ; RETURN TO CALLER.
2432
2433 CKEMAX: MOV R0,-(SP) ; SAVE R0
2434 MOV UNITN,R0 ; GET UNIT NUMBER
2435 ASL R0 ; ... AND MAKE IT A WORD OFFSET
2436 MOV ERTABL(R0),R0 ; GET ERROR TABLE ENTRY
2437 BIC #170000,R0 ; EXTRACT ERROR COUNT FIELD
2438 CMP R0,GERRMAX ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
2439 BHIS 1$ ; BR IF YES
2440 CMP ERRK,LERRMAX ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2441 BLO 2$ ; BR IF NO
2442 1$: RFLAGS R0 ; GET OPERATOR FLAGS
2443 TRAP CSRFLA
2444 BIT #IDU,R0 ; IS DROPPING INHIBITED?
2445 BNE 2$ ; BR IF YES.
2446 MOV #-1,DUFLG ; NO -- DROP THE UNIT
2447 ERRDF 4,EMAXDU
2448 TRAP C$ERDF
2449 .WORD 4
2450 .WORD EMAXDU
```


2439 017244 000000
017246
017246 013700 002174
017252 104451
2440 017254
017254 104444
2441 017256 012600
2442 017260 000207
2443

2\$:

.WORD 0
DODU UNITN
MOV UNITN,RO
TRAP C\$DODU
DOCLN
TRAP C\$DCLN
MOV (SP)+,RO
RTS PC

: RESTORE RO
: RETURN TO CALLER

```

2445 .SBTTL CKDROP - CHECK IF UNIT SHOULD BE DROPPED
2446
2447 :+ CHECK IF UNIT SHOULD BE DROPPED
2448 :-
2449 CKDROP: MOV R0,-(SP)
2450 FORCERROR 1$,NOTSSR
2451 RFLAGS R0
2452 TRAP C$RFLA
2453 BIT #IDU,R0
2454 BNE 1$
2455 MOV (SP),R0
2456 MOV #-1,DUFLG
2457 DODU UNITN
2458 MOV UNITN,R0
2459 TRAP C$DODU
2460
2461 ;ABORT THE PASS
2462
2463 1$: TRAP C$DCLN
2464 MOV (SP)+,R0
2465 RTS PC
2466
2467 .SBTTL CONFIG - DETERMINE CONFIGURATION OF SYSTEM
2468
2469 : SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
2470 :
2471 CONFIG: JSR PC,SOFINIT
2472 RTS PC
2473

```

```
2475                                .SBTTL KTON,KTOFF    - ENABLE/DISABLE MEMORY MANAGEMENT
2476                                :
2477                                : SUBROUTINE - ENABLE MEM MGT.
2478                                :
2479 017336 005737 003124    KTON:    TST        KTFLG            ; GOT KT?
2480 017342 001403                BEQ        1$                ; NO.
2481 017344 012737 000001 177572    MOV        #1,SRO           ; YES. ENABLE KT11.
2482 017352 000207                1$:        RTS        PC
2483
2484
2485
2486                                :
2487                                : SUBROUTINE - DISABLE MEM MGT.
2488                                :
2489 017354 005737 003124    KTOFF:    TST        KTFLG            ; GOT KT11?
2490 017360 001405                BEQ        1$                ; NO.
2491 017362 000240                NOP
2492 017364 000240                NOP
2493 017366 012737 000000 177572    MOV        #0,SRO           ; DISABLE KT.
2494 017374 000207                1$:        RTS        PC
2495
2496
```

```

2498 .SBTTL SETMAP - SETUP PAR6 MAPPING
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517 017376
2518 017376
2519 017402 005737 003124
2520 017406 001433
2521 017410 010102
2522 000006
2523
2524
2525
2526 017442 042701 000177
2527 017446 020137 003124
2528 017452 103011
2529 017454 010137 172354
2530 017460 042702 160000
2531 017464 062702 140000
2532 017470 010200
2533 017472 000261
2534 017474 000401
2535 017476 000241
2536 017500 000207
2537

      THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
      AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
      IS RETURNED BIASED TO PAR6.

      INPUTS:
      R0      HIGH ORDER ADDRESS BITS
      R1      LOW ORDER ADDRESS BITS

      OUTPUTS:
      R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
      CARRY   SET IF SUCCESS
             CLR IF ERROR

      SETMAP:
      SAVREG      ;SAVE R1-R4 UNTIL NEXT RETURN
      TST         ;SYSTEM HAVE ABOVE 28K?
      BEQ 10$     ;BR IF NO
      MOV R1,R2   ;SAVE LOW ORDER BITS
      .REPT 6
      ASR R0      ;CONVERT WORD ADDRESS TO 32W BLOCKS
      ROR R1      ;MAKE IT DOUBLE PRECISION
      .ENDR
      BIC #177,R1 ;ALINE FOR LOWER 4K BOUNDARY
      CMP R1,KTFLG ;HIGHER THAN EXISTING MEMORY?
      BHIS 10$    ;BR IF YES
      MOV R1,#KIPAR6 ;SETUP MAPPING REGISTER PAR6
      BIC #160000,R2 ;SETUP DISPLACEMENT IN PAGE
      ADD #140000,R2 ;ADD IN PAR6 BIAS
      MOV R2,R0    ;RETURN IN R0
      SEC         ;SET SUCCESS
      BR 15$
      10$: CLC     ;SET FAILURE
      15$: RTS    ;RETURN
      PC

```

```

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

```

TSV3 - GLOBAL AREAS MACRO M1113 25-MAY-82 08:43 PAGE 60-1 G 8
FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN

SEQ 0097

2596
2597

T
K

```

2599 .SBTTL CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
2600
2601 :+
2602 : COMPARE MEMORY WITH A BACKGROUND PATTERN
2603 :
2604 : INPUTS:
2605 :
2606 :     R0 = BACKGROUND PATTERN
2607 :     FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2608 :     KTFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2609 :
2610 : OUTPUTS:
2611 :
2612 :     CARRY - SET IF NO ERROR
2613 :     CARRY - CLR IF ERROR
2614 :
2615 : IMPLICIT OUTPUTS:
2616 :
2617 :     ERRHI - ERROR HIGH ADDRESS
2618 :     ERRLO - ERROR LOW ADDRESS
2619 :     EXPD  - EXPECTED DATA
2620 :     RECV  - RECEIVED DATA
2621 :
2622 : CMPMEM:
2623 :     SAVREG
2624 :     MOV R0,R3 ;SAVE R1-R5 UNTIL NEXT RETURN
2625 :     JSR PC,KTOFF ;COPY TEST PATTERN
2626 :     MOV FREE,R1 ;DISABLE KT.
2627 :     MOV FRESIZ,R2 ;GET FIRST FREE LOCATION
2628 :     CMP R3,(R1) ;SIZE OF FREE SPACE BELOW 28K.
2629 :     BEQ 15$ ;FREE SPACE LOCATION EQUAL TO EXPD?
2630 :     MOV R1,ERRLO ;BR IF YES
2631 :     CLR ERRHI ;SAVE ADDRESS IN ERROR
2632 :     MOV R3,EXPD ;NO HIGH ADDRESS
2633 :     MOV (R1),RECV ;SAVE EXPD FOR ERROR REPORT
2634 :     BR 50$ ;SAVE RECV FOR ERROR REPORT
2635 :
2636 : 10$: TST (R1)+ ;POINT TO NEXT ADDRESS
2637 :     DEC R2 ;DONE ALL MEMORY IN FREE SPACE?
2638 :     BGT 10$ ;BR IF NO
2639 :     TST KTFLG ;GOT KT?
2640 :     BEQ 55$ ;NO. GET OUT.
2641 :     JSR PC,KTON ;YES. ENABLE KT.
2642 :     CLR R0 ;HIGH ORDER ADDRESS START
2643 :     MOV PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
2644 :     .REPT 6
2645 :     ROL R1 ;CONVERT BLOCKS TO WORDS
2646 :     ROL R0 ;MAKE IT DOUBLE PRECISION
2647 :     .ENDR
2648 :     BIC #177,R1 ;ALINE 4K BOUNDARY
2649 :     MOV R0,-(SP) ;SAVE HIGH ORDER
2650 :     MOV R1,-(SP) ;SAVE LOW ORDER
2651 :     JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
2652 :     MOV R0,R4 ;COPY ADDRESS BIASED TO PAR6
2653 :     MOV (SP)+,R1 ;RESTORE LOW ORDER IN NON PAR6 FORMAT
2654 :     MOV (SP)+,R0 ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
2655 :     CMP R3,(R4) ;ABOVE 28K LOCATION EQUAL EXPD?
2656 :     BEQ 32$ ;BR IF YES
2657 :     MOV R0,ERRHI ;SAVE HIGH ORDER IN ERROR

```

2656	020120	010137	002232		MOV	R1,ERRLO	;SAVE LOW ORDER IN ERROR
2657	020124	010337	002224		MOV	R3,EXPD	;SAVE EXPD FOR ERROR REPORT
2658	020130	011437	002226		MOV	(R4),RECV	;SAVE RECV FOR ERROR REPORT
2659	020134	000421			BR	50\$	
2660	020136	062701	000002	32\$:	ADD	#2,R1	;UPDATE NON PAR6 ADDRESS
2661	020142	005500			ADC	R0	;MAKE IT DOUBLE PRECISION ADD
2662	020144	062704	000002		ADD	#2,R4	;UPDATE PAR FORMAT ADDRESS
2663	020150	020427	160000		CMP	R4,#160000	;END OF PAR6 MAPPING AREA?
2664	020154	103755			BLO	30\$;BR IF NO
2665	020156	162704	020000		SUB	#20000,R4	;BACKUP INTO PAR6 MAPPING BEGIN
2666	020162	062737	000200	172354	ADD	#200,@#KIPAR6	;POINT TO NEXT 4K BLOCK >28K.
2667	020170	023737	172354	003124	CMP	@#KIPAR6,KTFLG	;END OF MEMORY?
2668	020176	101744			BLOS	30\$;BR IF NO
2669	020200	004737	017354	50\$:	JSR	PC,KTOFF	;TURN OFF MEMORY MAPPING
2670	020204	000241			CLC		;SET FAILURE
2671	020206	000403			BR	60\$	
2672	020210	004737	017354	55\$:	JSR	PC,KTOFF	;TURN OFF MEMORY MAPPING
2673	020214	000261			SEC		;SET SUCCESS
2674	020216	000207		60\$:	RTS	PC	
2675							


```

2677                                     .SBTTL  REGSAV - SAVE R1-R5 ON STACK
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697 020220
2698 020220 010446
2699 020222 010346
2700 020224 010246
2701 020226 010146
2702 020230 010546
2703 020232 016605 000012
2704 020236 004736
2705 020240 012601
2706 020242 012602
2707 020244 012603
2708 020246 012604
2709 020250 012605
2710 020252 000207
2711
  
```

```

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

```

2713 .SBTTL GETPAT - GET 8 BIT PATTERN FROM OPERATOR
2714
2715 :+
2716 :ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
2717 :
2718 :INPUTS:
2719 :
2720 :       NONE.
2721 :
2722 :OUTPUTS:
2723 :
2724 :       R0       OCTAL NUMBER FROM THE OPERATOR
2725 :
2726 :CALLING SEQUENCE:
2727 :
2728 :       JSR       PC,GETPAT
2729 :
2730 :-
2731
2732 GETPAT::
2733 1$: SAVREG          ;SAVE THE GENERAL REGISTERS
2734   GMANID DATASC,PATDAT,0,377,0,377,NO
2735   TRAP   CS$GMAN
2736   BR     10000$
2737   .WORD  PATDAT
2738   .WORD  T$CODE
2739   .WORD  DATASC
2740   .WORD  377
2741   .WORD  T$LOLIM
2742   .WORD  T$HILIM
2743 10000$: BNCOMPLETE 1$ ;RETRY IF ERROR
2744   BCC 1$
2745   MOV PATDAT,R0 ;DATA PATTERN FROM OPERATOR
2746   RTS PC ;RETURN TO CALLER
2747
2748 :+
2749 :LOCAL DATA AREA
2750 :-
2751
2752 PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
2753 DATASC: .ASCIZ 'ENTER DATA PATTERN'
2754 .EVEN

```

```

2747 .SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
2748
2749
2750 :ROUTINE TO ISSUE A MENU AND GET
2751 :THE OPERATOR'S RESPONSE.
2752
2753 :INPUTS:
2754
2755 :      R0      ADDRESS OF ASCIZ STRING OF MENU
2756 :      R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
2757
2758 :OUTPUTS:
2759
2760 :      R0      NUMBER OF THE OPERATOR'S SELECTION
2761
2762 :-
2763
2764 GETSEL::
2765     SAVREG                                ;SAVE GENERAL REGISTERS
2766     MOV     R0,R2                        ;SAVE THE MENU ADDRESS
2767     MOV     R2,R3                        ;START OF MENU STRING
2768     TST     (R3)                        ;END OF ASCII ?
2769     BEQ     3$                          ;BRANCH IF ALL LINES DISPLAYED
2770     PRINTF  #SELASC,(R3)+                ;DISPLAY THE MENU
2771     MOV     (R3)+,-(SP)
2772     MOV     #SELASC, -(SP)
2773     MOV     #2, -(SP)
2774     MOV     SP,R0
2775     TRAP    C$PNTF
2776     ADD     #6,SP
2777     BR      2$
2778     3$:     GMANID  MENASC,MENRES,D,-1,0,-1,NO
2779     TRAP    C$GMAN
2780     BR      10001$
2781     .WORD   MENRES
2782     .WORD   T$CODE
2783     .WORD   MENASC
2784     .WORD   -1
2785     .WORD   T$LOLIM
2786     .WORD   T$HILIM
2787     10001$: BNCOMPLETE 1$                ;RETRY IF ERROR
2788     BCC     1$
2789     MOV     MENRES,R0                    ;GET THE OPERATOR'S REPLY
2790     CMP     R0,R1                        ;COMPARE TO MAXIMUM ALLOWED
2791     BLOS    5$                          ;BRANCH IF OK
2792     PRINTF  #MENERR                      ;DISPLAY ERROR MESSAGE
2793     MOV     #MENERR, -(SP)
2794     MOV     #1, -(SP)
2795     MOV     SP,R0
2796     TRAP    C$PNTF
2797     ADD     #4,SP
2798     BR      1$
2799     5$:     RTS     PC                    ;RETRY
2800           ;RETURN TO CALLER
2801     045     MENERR: .ASCIZ 'XNZA *** Menu Selection Too Large ***'
2802     045     SELASC: .ASCIZ 'XNXT'
2803     164     MENASC: .ASCIZ 'Enter Menu Selection: '
  
```

TSV3 - GLOBAL AREAS MACRO M1113 25-MAY-82 08:43 PAGE 64-1 M 8
GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

SEQ 0103

2783
2784 020556 000000 MENRES: .EVEN .WORD 0

```

2786 .SBTTL CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808 020560
2809 020560
2810 020564 104450
2811 020566 103411
2812 020570 012746 020614
      020574 012746 000001
      020600 010600
      020602 104417
      020604 062706 000004
2813 020610 000241
2814 020612 000207
2815
2816 020614 045 116 045
2817
      ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
      INPUT:
      NONE.
      OUTPUT:
      CARRY 0 MANUAL INTERVENTION NOT ALLOWED
      1 MANUAL INTERVENTION IS OK
      SIDE EFFECTS:
      A MESSAGE IS DISPLAYED WARNING THAT TEST IS
      NOT EXECUTED IF MANUAL INTERVENTION IS NOT
      ALLOWED.
      -
      CHKMAN::
      SAVREG ;SAVE THE REGISTERS
      MANUAL ;SEE IF MANUAL INTERVENTION OK
      TRAP C$MANI
      BCCMPLETE 1$ ;BRANCH IF ALLOWED
      BCS 1$
      PRINTF #NOMAN ;PRINT THE WARNING MESSAGE
      MOV #NOMAN, -(SP)
      MOV #1, -(SP)
      MOV SP, R0
      TRAP C$PNTF
      ADD #4, SP
      CLC ;CLEAR CARRY FOR ERROR
      RTS PC ;RETURN
      1$:
      NOMAN: .ASCII '*** Manual Intervention not Allowed - Test Aborted ***'
      .even
  
```

```
2819 .SBTTL ENVIRN - SETUP FREE DIAGNOSTIC SPACE
2820
2821 : SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
2822 :
2823 ENVIRN: MEMORY R0
2824 020710 104431 TRAP C$MEM
2825 020712 010037 003116 MOV R0,FREE ; GET 1ST FREE ADDRESS...
2826 020724 011037 003120 ADD #2,FREE ;...AND WORD COUNT.
2827 020730 162737 000004 003120 MOV (R0),FRESIZ
2828 020736 013702 002012 MOV L$UNIT,R2 ; GET NUMBER OF UNITS
2829 020742 162737 000007 003120 10$: SUB #7,FRESIZ ; TAKE AWAY 7 WORDS PER UNIT
2830 020750 005302 DEC R2
2831 020752 001377 BNE 10$
2832 020754 013700 003116 MOV FREE,R0 ;GET FIRST FREE ADDRESS
2833 020760 063700 003120 ADD FRESIZ,R0 ;POINT TO LAST FREE ADDRESS
2834 020764 162700 000002 SUB #2,R0 ;BACKUP 1 WORD
2835 020770 010037 003122 MOV R0,FREEHI ;STORE LAST FREE ADDRESS
2836 020774 000240 NOP *****
2837 020776 012701 177520 MOV #BDVPCR,R1 ;GET BDV11 PCR ADDRESS
2838 021002 010102 MOV R1,R2 ;COPY TO R2
2839 021004 062702 000002 ADD #2,R2 ;SET THE RANGE
2840 021010 004737 016456 JSR PC,XNXM ;SEE IF WE HAVE ONE
2841 021014 103001 BCC 15$ ;OK TO SET FLAGS
2842 021016 000445 BR 40$ ;RETURN WITH FLAGS CLEAR
2843 021020 013701 177520 15$: MOV BDVPCR,R1 ;SAVE PCR CONTENTS
2844 021024 062701 000001 ADD #1,R1 ;ADD ONE TO IT
2845 021030 012702 177520 MOV #BDVPCR,R2 ;GET BDV11 PCR ADDRESS
2846 021034 005212 INC (R2) ;TRY TO WRITE TO IT
2847 021036 013703 177520 MOV BDVPCR,R3 ;GET RESULTS
2848 021042 020103 CMP R1,R3 ;DID IT CHANGE?
2849 021044 001017 BNE 20$ ;NO, MUST BE 11/23B
2850 021046 005237 003136 INC T23A ;SET THE FLAG
2851 021052 042737 170000 002120 BIC #170000,L$HIME ;SUPERVISOR COULD BE WRONG
2852 021060 000240 NOP ;BR 40$ FOR RELEASE
2853 021062 PRINTF #M8186 ;TELL THE SYSTEM TYPE
2854 021062 012746 005550 MOV #M8186,-(SP)
2855 021066 012746 000001 MOV #1,-(SP)
2856 021072 010600 MOV SP,R0
2857 021074 104417 TRAP C$PNTF
2858 021076 062706 000004 ADD #4,SP
2859 021102 000413 BR 40$ ;RETURN
2859 021104 005237 003140 20$: INC T23B ;SET THE FLAG
2859 021110 000240 NOP ;BR 40$ FOR RELEASE
2859 021112 PRINTF #M8189 ;TELL THE SYSTEM TYPE
2859 021112 012746 005641 MOV #M8189,-(SP)
2859 021116 012746 000001 MOV #1,-(SP)
2859 021122 010600 MOV SP,R0
2859 021124 104417 TRAP C$PNTF
2859 021126 062706 000004 ADD #4,SP
2859 021132 000207 40$: RTS PC ;RETURN
```

```

2861                                     .SBTTL KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS
2862
2863                                     :+
2864                                     :ROUTINE TO INIT KT-11
2865                                     :-
2866
2867
2868 021134 KTINIT:
2869 021134 005037 003124 CLR KTFLG ; INIT >28K MEMORY FLAG
2870 021140 005037 003126 CLR KTENABLE ; INIT TEST >28K FLAG
2871 021144 023727 002120 001577 CMP L$HIME,#1577 ; GOT ENOUGH MEMORY (>28K)?
2872 021152 101444 BLOS 9$ ; NO.
2873 021154 013700 000004 MOV @#ERRVEC,R0 ; SAVE OLD ERR VEC PTR.
2874 021160 012737 021252 000004 MOV #2$,@#ERRVEC ; SET ERR VEC PTR.
2875 021166 005737 177572 TST @#SRO ; GOT KT11?
2876 021172 000240 NOP ; (TRAP IF NO).
2877 021174 013737 002120 003124 MOV L$HIME,KTFLG ; YES. SET KT FLAG.
2878 021202 042737 000177 003124 BIC #177,KTFLG
2879 021210 010037 000004 MOV R0,@#ERRVEC ; RESTORE OLD ERR VEC PTR.
2880 021214 005000 CLR R0 ; R0 = AR DATA.
2881 021216 012701 172340 MOV #KIPAR0,R1 ; R1 = KI REGS PTR.
2882 021222 012761 077406 177740 1$: MOV #77406,-40(R1) ; SET DESCRIPTOR REG.
2883 021230 010021 MOV R0,(R1)+ ; SET KIPAR REG.
2884 021232 062700 000200 ADD #200,R0 ; BUMP AR DATA BY '4k'.
2885 021236 020027 002000 CMP R0,#2000 ; AT 'I/O'?
2886 021242 001367 BNE 1$ ; NO.
2887 021244 012741 177600 MOV #177600,-(R1) ; YES. SET KIPAR7 FOR I/O.
2888 021250 000405 BR 9$
2889
2890 021252 012716 021260 2$: MOV #6$,(SP) ; SET UP RETURN
2891 021256 000002 RTI ; RTI TO NEXT LOCATION
2892
2893 021260 010037 000004 6$: MOV R0,@#ERRVEC ; RESTORE OLD ERR VEC PTR.
2894
2895 021264 000207 9$: RTS PC
2896

```

```

2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911 021266
2912
2913 021266 005737 002220
2914 021272 001020
2915 021274 012737 100206 021340
2916 021302 012737 021350 021342
2917 021310 012737 000006 021346
2918 021316 012737 100010 021350
2919 021324 012704 021340
2920 021330 004737 010742
2921 021334 000207
2922
2923
2924
2925
2926
2927
2928 021340 000000
2929 021342 000000
2930 021344 000000
2931 021346 000000
2932
2933
2934
2935
2936 021350 000000
2937 021352 000000
2938 021354 000000
2939
2940

: +
: SUBROUTINE TO SET EXTENDED FEATURES SWITCH
: Requires that SOFINIT and WRTCHR have been done previous to call.
:
: INPUTS:
: R5 CURRENT UNIT NUMBER
: OUTPUTS:
: The Extended Features Switch is set.
:
: -
:
: INVERT::
:
: TST EXTFEA ; IS SWITCH SET?
: BNE 1$ ; YES, EXIT STAGE RIGHT! (or the next one outa town!)
: MOV #100206, CMDPKT ; WRT SUB-SYS MEM CMD
: MOV #WSMBK, CMDPKT+2 ; MSG BUF ADDR
: MOV #6, CMDPKT+6 ; BYTE COUNT
: MOV #100010, WSMBK ; INVERT THE SWITCH
: MOV #CMDPKT, R4 ; SET CMDPKT INTO R4
: JSR PC, WRTCHR ; DO IT
: RTS PC ; RETURN
:
: COMMAND PACKET.
:
: = <.+3>&177774 ; MUST BE ON MOD 4 BOUNDARY.
:
: CMDPKT:: 0 ; 1ST WORD IS TS05 COMMAND.
: 0 ; 2ND WORD IS THE BUFFER LOW ADDRESS.
: 0 ; 3RD WORD IS THE BUFFER HIGH ADDRESS.
: 0 ; 4TH WORD IS THE BYTE/RECORD/FILE COUNT.
:
: WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
:
: WSMBK:: 0 ; 1ST WORD:: SEL 0
: 0 ; 2ND WORD:: SEL 2
: 0 ; 3RD WORD:: SEL 4
: .EVEN

```



```

2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952 021356
2953
2954 021356
2955 021362 005037 003130
2956 021366 005037 003132
2957 021372 005037 003134
2958 021376 005737 003140
2959 021402 001407
2960 021404 023727 002120 007777
2961 021412 103406
2962 021414 004737 021532
2963 021420 000427
2964 021422 005737 003136 1$:
2965 021426 001413
2966 021430 023727 002120 005777 2$:
2967 021436 101023
2968 021440 023727 002120 003777
2969 021446 103403
2970 021450 004737 021532
2971 021454 000411
2972 021456 023727 002120 001577 4$:
2973 021464 103410
2974 021466 004737 021532
2975 021472 062737 000077 003134
2976 021500 005237 003130 13$:
2977 021504 000411
2978 021506 000410 14$:
2979 021510
      021510 012746 005454
      021514 012746 000001
      021520 010600
      021522 104417
      021524 062706 000004
2980 021530 000207 15$:
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990 021532 013701 002120
2991 021536 062701 000200
2992 021542 042701 000177
2993 021546 010102

      :+
      :
      : SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
      :
      : INPUTS:
      : OUTPUTS:
      : The NXMFLG is set if we can test.
      : The NXMLO and NXMHI addresses are setup.
      :
      : -
      :
      : MEMCK::
      :
      : SAVREG
      : CLR NXMFLG
      : CLR NXMLO
      : CLR NXMHI
      : TST T23B
      : BEQ 1$
      : CMP L$HIME,#7777
      : BLO 2$
      : JSR PC,NXMTST
      : BR 13$
      : TST T23A
      : BEQ 4$
      : CMP L$HIME,#5777
      : BHI 14$
      : CMP L$HIME,#3777
      : BLO 4$
      : JSR PC,NXMTST
      : BR 13$
      : CMP L$HIME,#1577
      : BLO 14$
      : JSR PC,NXMTST
      : ADD #77,NXMHI
      : INC NXMFLG
      : BR 15$
      : BR 15$
      : PRINTF #NOMEM
      : MOV #NOMEM,-(SP)
      : MOV #1,-(SP)
      : MOV SP,R0
      : TRAP C$PNTF
      : ADD #4,SP
      : RTS PC
      :
      : ;SAVE THE REGISTERS
      : ;CLEAR THE FLAG
      : ;CLEAR THE TEST ADDRESS LO
      : ;CLEAR THE TEST ADDRESS HI
      : ;IS IT A 11/23B?
      : ;NO
      : ; GREATER THAN 128K
      : ; NO
      : ;SETUP THE ADDRESS
      : ;SET THE FLAG AND EXIT
      : ;IS IT A 11/23A?
      : ;NO
      : ;GREATER THAN 96K
      : ;YES,23A/23B WITH 128K MEMORY
      : ;GREATER THAN 64K BUT LESS THAN 92K?
      : ;NO, CHECK 24K
      : ;SETUP THE ADDRESS
      : ;SET THE FLAG AND EXIT
      : ;GREATER THAN 24K BUT LESS THAN 64K?
      : ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
      : ;SETUP THE ADDRESS
      : ;FOOL THE 11/02 & 11/03
      : ;SET THE FLAG
      : ;EXIT
      : ;NOP FOR PRINTOUT
      : ;TELL THEM & EXIT ***NO PRINT*****
      :
      : ;RETURN

      :+
      :
      : SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
      :
      : OUTPUTS:NXMLO,NXMHI
      : ;SETUP WITH NXM ADDRESS
      :
      : -
      :
      : NXMTST: MOV L$HIME,R1
      : ADD #200,R1
      : BIC #177,R1
      : MOV R1,R2
      :
      : ;GET TOP OF MEMORY
      : ;MAKE IT I/O BLOCK OR OTHER NXM
      :
      : ;RESAVE RESULTS

```

SEQ 0109

2994		000006		.REPT	6	
2995				ASL	R1	;PUT IN PLACE FOR XFER
2996				.ENDR		
2997	021564	010137	003132	MOV	R1,NXMLO	;SAVE TEST ADDRESS LOW
2998		000012		.REPT	10.	
2999				ASR	R2	;PUT IN PLACE FOR XFER
3000				.ENDR		
3001	021614	042702	177700	BIC	#177700,R2	;DON'T WANT ILA!
3002	021620	010237	003134	MOV	R2,NXMHI	;SAVE TEST ADDRESS HIGH
3003	021624	000207		RTS	PC	;RETURN
3004						
3005						
3006						
3007						
3008	021626			ENDMOD		

TSV4 - MISCELLANEOUS SECTIONS MACRO M1113 25-MAY-82 08:43 PAGE 71
KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

SEQ 0110

7
8
9 021626
10 021626
16

.TITLE TSV4 - MISCELLANEOUS SECTIONS
BGNMOD TSV4
TSV4::

TT

```

18                                     .SBTTL  PROTECTION TABLE
19 021626                             BGNPROT
    021626
20 021626 177777 177777 177777  L$PROT:: .WORD  -1, -1, -1, -1
21 021636                             ENDPROT
22

```

;NO DEVICE PROTECTION REQUIRED.

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

.SBTTL INITIALIZE SECTION

```

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

```

BGNINIT

LSINIT::
40\$:

```

CLR      EXTFEA
CLR      NXMFLG
MOV      #EPRT1,EPRTSW
CLR      SIFLAG
CLR      KTENABLE
CLR      RAMSIZ
READEF   #EF.CONTINUE
MOV      #EF.CONTINUE,R0
TRAP     CSREFG
BNCOMPLETE 1$
BCC      1$
CMP      UNITN,LSUNIT
BHS      4$
TST      DUFLG
BMI      NXTU
MOV      UNITN,R1
ASL      R1
TST      ERTABL(R1)
BEQ      SETU
BIT      #BIT14,ERTABL(R1)
BNE      NXTU
EXIT     INIT
TRAP     CSEXIT
.WORD    L10030-.
READEF   #EF.NEW
MOV      #EF.NEW,R0
TRAP     CSREFG
BNCOMPLETE NXTU
BCC      NXTU
READEF   #EF.START
MOV      #EF.START,R0
TRAP     CSREFG
BCOMPLETE 2$
BCS      2$
READEF   #EF.RESTART
MOV      #EF.RESTART,R0
TRAP     CSREFG
BNCOMPLETE 31$
BCC      31$
BRESET
TRAP     CSRESET

```

```

:SET UP PRIMARY MESSAGE FOR REPLACEMENT
:CLEAR 'SOFT INIT' FLAG
:CLEAR TEST ABOVE 28K FLAG
:CLEAR RAM SIZE FOR RAMERR ROUTINE

```

```

:UNIT IN RANGE?
:BR IF NO.
:DROPPED UNIT?
:BR IF YES

```

:DROPPED?

:DO NOTHING IF 'CONTINUE'.

:TAKE NEXT UNIT IF NOT NEW PASS.

```

:1ST PASS, BUS-INIT...
:BUS RESET.

```

1\$:

2\$:

002172

002012

003170

000035

000040

000037

104433

```
65 022000 005037 002206      CLR      TSTCNT      ;NUMBER OF TESTS RUN IN PASS
66 022004 005037 002214      CLR      FATFLG      ;CLEAR FATAL ERROR COUNT
67 022010 005037 003136      CLR      T23A       ;CLEAR 11/23A FLAG
68 022014 005037 003140      CLR      T23B       ;CLEAR 11/23B FLAG
69      :      MOV      #340,-(SP)
70      :      MOV      #20$,-(SP)      ;RETURN TO DEBUGGER
71      :      JMP      0.ODT      ;:ENTER THE DEBUGGER
72 022020 005037 003372      CLR      SKIPT      ;CLEAR THE SUBTEST 'SKIPPER'
73 022024      :      20$:
74 022024 012737 177777 002176  MOV      #-1,QVP      ;...QUICK VERIFY...
75 022032 004737 020710      JSR      PC,ENVIRN      ;SET ENVIRONMENT.
76 022036 004737 021134      JSR      PC,KTINIT      ;INITIALIZE KT MEMORY MANAGEMENT
77 022042 012700 003170      MOV      #ERTABL,R0
78 022046 005020 30$:      CLR      (R0)+      ;CLEAR THE ERROR TABLE
79 022050 020027 003370      CMP      R0,#ERTABE
80 022054 103774      BLO      30$
81 022056 000404      BR      4$
82 022060 005037 002176 31$:      CLR      QVP
83 022064 000137 022134      JMP      PASRPT      ;GO REPORT THE STATUS
84
85 022070      4$:
86 022070 012737 177777 002174  NEWPAS: MOV      #-1,UNITN      ;INIT UNIT NUMBER...
87 022076 005037 002212      CLR      DEVCNT      ;CLEAR COUNT OF DEVICES RUNNING
88 022102      NXTU:      BREAK
89 022102 104422      TRAP      CSBRK
90 022104 005237 002174      INC      UNITN      ;...AND SET NEXT UNIT NUMBER.
91 022110 023737 002174 002012  CMP      UNITN,LSUNIT
92 022116 103423      BLO      SETU
93 022120 012737 177777 003104  MOV      #-1,DUFLG
94 022126 000401      BR      11$
95 022130      DOCLN      ;ABORT, NO MORE UNITS.
96 022132 104444      TRAP      C$DCLN
97 022134 000240      NOP
98 022134 023727 002012 000001 11$:      PASRPT:      CMP      LSUNIT,#1      ;HOW MANY UNITS SELECTED?
99 022142 101752      BLOS      NEWPAS      ;BR IF ONLY 1
100 022144 005737 002212      TST      DEVCNT      ;ARE ANY STILL RUNNING?
101 022150 001747      BEQ      NEWPAS      ;BR IF NO
102 022152      RFLAGS      R0
103 022154 104421      TRAP      C$RFLA
104 022154 032700 000100      BIT      #ISR,R0      ;SHOULD WE PRINT STATISTICS
105 022160 001343      BNE      NEWPAS      ;BR IF NO
106
107 022162      DORPT
108 022162 104424      TRAP      C$DRPT
109 022164 000741      BR      NEWPAS
110
111 022166      10$:
112 022166 013700 002174      SETU:      GPWARD      UNITN,R0      ;GET UNIT N P-TABLE POINTER.
113 022172 104442      MOV      UNITN,R0
114 022174      TRAP      C$GPHRD
115 022174 103342      BNCOMPLETE NXTU      ;BR IF UNIT NOT AVAILABLE.
116 022176 005037 003104      BCC      NXTU
117 022202 005237 002212      CLR      DUFLG      ;CLEAR 'DROPPED' FLAG.
118 022206 012001      INC      DEVCNT
119 022210 010137 002200      MOV      (R0)+,R1      ;GET 1ST REGISTER ADDRESS.
120      MOV      R1,CSRADDR      ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
```

```

115
116 022214 012001          MOV      (R0)+,R1          ;GET VECTOR ADDRESS.
117                      ;MOV      (R0),R2          ;GET INTERRUPT PRIORITY
118                      ;MOV      R2,IPRI          ;SET INTERRUPT PRIORITY.
119 022216 010137 002202    MOV      R1,IVEC          ;SET INTERRUPT VECTOR POINTER...
120 022222 012721 016276    MOV      #INTR,(R1)+      ;...VECTOR...
121 022226 013721 002204    MOV      IPRI,(R1)+      ;...AND PRIORITY.
122
123 022232                1$:
124                      ;TST      QVP              ;1ST PASS ??
125                      ;BEQ      5$              ;NO, SKIP THE PASS 1 STUFF.
126
127                      ;
128                      ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
129                      ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
130                      ;
131 022232 013701 002174      MOV      UNITN,R1
132 022236 006301            ASL      R1
133 022240 052761 100000 003170  BIS      #BIT15,ERTABL(R1)      ;SAY DEVICE RUNNING
134 022246 005037 005766      CLR      EXTA          ;CLEAR ERROR EXTENSION FLAG.
135 022252 023727 002012 000001  CMP      L$UNIT,#1      ;ARE WE TESTING MULTIPLE UNITS?
136 022260 101416            BLOS     10$          ;BR IF NO.
137 022262            RFLAGS     RO              ;YES -- GET OPERATOR FLAGS.
138 022264 104421            TRAP     CSRFLA
139 022270 001412            BIT      #PNT,RO
140 022272            BEQ      10$          ;SHOULD WE PRINT UNIT #?
141 022272 013746 002174      PRINTF  #PUNIT,UNITN      ;PRINT THE UNIT #
142 022276 012746 022364      MOV      UNITN,-(SP)
143 022302 012746 000002      MOV      #PUNIT,-(SP)
144 022306 010600            MOV      #2,-(SP)
145 022310 104417            MOV      SP,RO
146 022312 062706 000006      TRAP     C$PNTF
147 022316            ADD      #6,SP
148 022316 005037 003106      10$:      CLR      NODEV
149 022322 013701 002200      MOV      CSRADDR,R1      ;ADDRESS OF FIRST REGISTER
150 022326 010102            MOV      R1,R2          ;START OF REGISTERS
151 022330 062702 000002      ADD      #TSSR,R2      ;ADDRESS OF TSSR REGISTER
152 022334 004737 016456      JSR      PC,XNXM      ;TEST BOTH CONTROLLER REGISTERS...
153 022340 103005            BCC      2$          ;...AND BR IF ALL OK.
154 022342 010137 003106      MOV      R1,NODEV      ;FLAG DEVICE AS NON-EXISTENT
155 022346 012737 177777 003104  MOV      #-1,DUFLG      ;DROP THIS UNIT.
156 022354            2$:
157                      ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
158                      ;
159 022354            5$:      SETPRI  #PRI00          ;ENABLE INTERRUPTS.
160 022354 012700 000000      MOV      #PRI00,RO
161 022360 104441            TRAP     C$SPRI
162 022362            ENDINIT
163 022362            L10030:
164 022362 104411            TRAP     C$INIT
165
166 022364 045 116 045 PUNIT: .ASCIZ  /XNXNXA***** TESTING UNIT %D2XA *****/
167                      .EVEN

```

```
160                                     .SBTTL  ADD AND DROP UNITS SECTIONS
161
162
163      :++
164      : THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
165      : TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
166      : OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
167      :--
168      BGNAU
169      L$AU::
170      MOV      R0,R1                      ; GET UNIT TO BE ADDED (R0)
171      ASL      R1                          ; MAKE IT A WORD INDEX
172      BIS      #100000,ERTABL(R1)         ; SET THE 'ACTIVE' BIT
173      BIC      #40000,ERTABL(R1)         ; CLEAR THE 'DROPPED' BIT
174      PRINTF   #1$,R0
175      MOV      R0,-(SP)
176      MOV      #1$,-(SP)
177      MOV      #2,-(SP)
178      MOV      SP,R0
179      TRAP     C$PNTF
180      ADD      #6,SP
181      EXIT     AU
182      .WORD    JSJMP
183      .WORD    L10031-2-.
184      .ASCIZ   /XN% UNIT XD% ADDED/
185      .EVEN
186
187      ENDAU                                ; UNUSED.
188      L10031:
189      TRAP     C$AU
190
191      :++
192      : THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
193      : TO BE REMOVED FROM THE TEST LIST.
194      :
195      : SUPVSR DOES THE 'DROPPING'. THIS IS JUST TO TELL THE MAN.
196      : 'DROPPED' UNITS ARE RE-SELECTED ON OPERATOR 'STA' OR 'ADD'.
197      : COMMAND, OTHERWISE REMAIN INACTIVE. THE 'DISPLAY' COMMAND
198      : WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
199      : WHICH ARE STILL ACTIVE.
200      : UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.
201
202      BGNDU
203      L$DU::
204      MOV      #-1,DUFLG
205      MOV      R0,R1
206      ASL      R1
207      BIS      #140000,ERTABL(R1)         ; SAY DROPPED
208      240,240,240                          ; ?????????
209      PRINTF   #1$,R0
210      MOV      R0,-(SP)
211      MOV      #1$,-(SP)
212      MOV      #2,-(SP)
213      MOV      SP,R0
214      TRAP     C$PNTF
215      ADD      #6,SP
216      EXIT     DU
217      .WORD    JSJMP
218      .WORD    L10032-2-.
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
```


TRAP CS AUTO

```

215 .SBTTL CLEAN-UP AND REPORT CODING SECTIONS
216
217
218 :++
219 : THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
220 : EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
221 : USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
222 :--
222 022716 BGNCLN
223 022716 L$CLEAN::
224 022716 013705 002200 MOV CSRADDR,R5 ;POINT TO DEVICE REGISTER
225 022722 005737 003104 TST DUFLG ;'DROPPED' FLAG IS SET ON...
226 022726 100405 BMI 1$ ;...AND GROSS CONTROLLER FAULT...
227 ;...DON'T TRY TO XCT CLEANUP CODE.
228 022730 012765 000000 000002 MOV #0,TSSR(R5) ;DO SOFT INIT
229 022736 004737 016330 JSR PC,WAITF
230 022742 1$:
231 022742 2$: ENDCLN
232 022742 104412 L10034: TRAP C$CLEAN
233 :++
234 : THE REPORT CODING SECTION CONTAINS THE
235 : 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
236 :--
236 022744 BGNRPT
237 022744 L$RPT::
238 022744 012746 023206 PRINTS #DEVSUM
239 022750 012746 000001 MOV #DEVSUM,-(SP)
240 022754 010600 MOV #1,-(SP)
241 022756 104416 MOV SP,R0
242 022760 062706 000004 TRAP C$PNTS
243 022764 010246 ADD #4,SP
244 022766 010346 MOV R2,-(SP)
245 022770 010446 MOV R3,-(SP)
246 022772 012704 003170 MOV R4,-(SP)
247 022776 005003 MOV #ERTABL,R4 ; GET START OF ERROR TABLE.
248 023000 011402 1$: MOV (R4),R2 ; CLEAR UNIT NUMBER
249 023002 001467 BEQ 4$ ; GET ERROR TABLE ENTRY & TEST IT.
250 023004 100066 BPL 4$ ; ZERO IF UNIT NOT RUN
251 023006 032702 040000 BIT #BIT14,R2 ; WAS UNIT DROPPED?
252 023012 001015 BNE 2$ ; BR IF YES
253 023014 042702 170000 BIC #^C7777,R2 ; GET ERROR COUNT FIELD
254 023020 PRINTS #DEVONL,R3,R2 ; PRINT
255 023020 010246 MOV R2,-(SP)
256 023022 010346 MOV R3,-(SP)
257 023024 012746 023243 MOV #DEVONL,-(SP)
258 023030 012746 000003 MOV #3,-(SP)
259 023034 010600 MOV SP,R0
260 023036 104416 TRAP C$PNTS
261 023040 062706 000010 ADD #10,SP
262 023044 000446 BR 4$
263 023046 020227 160000 2$: CMP R2,#160000 ; WAS UNIT NON-EXISTENT?
264 023052 001012 BNE 3$ ; BR IF NO
265 023054 PRINTS #DEVNXR,R3
266 023054 010346 MOV R3,-(SP)
267 023056 012746 023313 MOV #DEVNXR,-(SP)

```

```

023062 012746 000002      MOV      #2,-(SP)
023066 010600      MOV      SP,R0
023070 104416      TRAP     C$PNTS
023072 062706 000006      ADD      #6,SP
254 023076 000431      BR      4$
255 023100 020227 160001      3$:    CMP      R2,#160001      : WAS UNIT NOT READY AT STARTUP?
256 023104 001012      BNE     30$      : BR IF NO.
257 023106      PRINTS   #DEVNRD,R3
023106 010346      MOV      R3,-(SP)
023110 012746 023375      MOV      #DEVNRD,-(SP)
023114 012746 000002      MOV      #2,-(SP)
023120 010600      MOV      SP,R0
023122 104416      TRAP     C$PNTS
023124 062706 000006      ADD      #6,SP
258 023130 000414      BR      4$
259 023132 042702 170000      30$:   BIC      #^C7777,R2
260 023136      PRINTS   #DEVDRD,R3,R2
023136 010246      MOV      R2,-(SP)
023140 010346      MOV      R3,-(SP)
023142 012746 023456      MOV      #DEVDRD,-(SP)
023146 012746 000003      MOV      #3,-(SP)
023152 010600      MOV      SP,R0
023154 104416      TRAP     C$PNTS
023156 062706 000010      ADD      #10,SP
261 023162 062704 000002      4$:    ADD      #2,R4
262 023166 005203      INC      R3
263 023170 020427 003370      CMP      R4,#ERTABE
264 023174 103701      BLO     1$
265 023176 012604      MOV      (SP)+,R4
266 023200 012603      MOV      (SP)+,R3
267 023202 012602      MOV      (SP)+,R2
268 023204      ENDRPT      : UNUSED.
023204      L10035:
023204 104425      TRAP     C$RPT
269
270
271 023206      045      116      045 DEVSUM: .ASCIZ /%N%ADEVICE STATUS SUMMARY:%N/
272 023243      045      101      040 DEVONL: .ASCIZ /%A UNIT %D3%A ONLINE, ERRORS = %D%N/
273 023313      045      101      040 DEVNXR: .ASCIZ /%A UNIT %D3%A DROPPED, NON-EXISTENT REGISTER%N/
274 023375      045      101      040 DEVNRD: .ASCIZ /%A UNIT %D3%A DROPPED, NOT READY AT STARTUP%N/
275 023456      045      101      040 DEVDRD: .ASCIZ /%A UNIT %D3%A DROPPED, ERRORS = %D%N/
276      .EVEN
277
278 023526      ENDMOD
279
280

```

1
2
3
10
11 023526
17 023526

.TITLE TEST 1 - HARDWARE TEST 1-8 TESTS

TSV7B:: BGNMOD TSV7B

```
27
28
29
30
31
32
33
34
35
36
37
38
39 023526
    023526
40 023526 012737 006354 002172
45 023534 012700 032117
46 023540 004737 016570
47 023544 012737 000005 002210
48 023552 005037 026474
49 023556

          .SBTTL TEST 1: WRITE TAPE MARK RETRY
          :+
          :THIS TEST VERIFIES PROPER OPERATION OF THE WRITE TAPE MARK RETRY COMMAND (SPACE
          :REVERSE, ERASE, WRITE TAPE MARK). SUBTESTS ARE AS FOLLOWS:
          :
          :THE TEST CONSISTS OF THE FOLLOWING 4 SUBTESTS
          :
          :-
          BGNTST
          MOV #EPR11,EPR1SW
          MOV #TST29ID,R0
          JSR PC,TSTSETUP
          MOV #5,LOOPCNT
          CLR T29CNT
          T29LOOP:
          T1::
          :PRIMARY ERROR MESSAGE
          :ASCII MESSAGE TO IDENTIFY TEST
          :DO INITIAL TEST SETUP
          :PERFORM 5 ITERATIONS
          :CLEAR TAPE RECORD COUNTER
```


98	023742			ERRDF	ERRNO,T29OFL,EXPREC	:DRIVE IS OFF LINE		
	023742	104455					TRAP	C\$ERDF
	023744	000147					.WORD	103
	023746	026502					.WORD	T29OFL
	023750	015554					.WORD	EXPREC
99	023752	004737	017262			:TRY AND DROP DRIVE		
100	023756	004737	011074	26\$:	JSR PC,CKDROP	:CALL TAPE REWIND COMMAND		
101	023762	016501	000002		JSR PC,REWIND	:GET TSSR		
102	023766	012702	000200		MOV TSSR(R5),R1	:SET UP EXPECTED TSSR		
103	023772	103407			MOV #SSR,R2	:BR, IF NO PROBLEM		
104	023774	010004			BCS 30\$:PACKET ADDRESS SET UP		
105	023776	005237	002214		MOV R0,R4	:ERROR COUNT		
109	024002				INC FATFLG	:REWIND NOT ACCEPTED		
	024002	104456			ERRHRD ERRNO,T29RWN,PKTSSR		TRAP	C\$ERHRD
	024004	000150					.WORD	104
	024006	030305					.WORD	T29RWN
	024010	012126					.WORD	PKTSSR
110	024012			30\$:	CKLOOP	:LOOP IF SELECTED		
	024012	104406					TRAP	C\$CLP1
111	024014	013701	026350		MOV T29BFR+6,R1	:PICK UP XSTO		
112	024020	010102			MOV R1,R2	:SET UP EXPECTED		
113	024022	052702	000002		BIS #BIT1,R2	:SET BOT BIT IN EXPECTED		
114	024026	020102			CMP R1,R2	:DOES EXP = REC'D		
115	024030	001406			BEQ 40\$:BR, IF EQUAL (OK)		
116	024032	005237	002214		INC FATFLG	:ERROR COUNT		
120	024036				ERRHRD ERRNO,T29BOT,EXPREC	:TAPE NOT AT BOT AFTER REWIND		
	024036	104456					TRAP	C\$ERHRD
	024040	000151					.WORD	105
	024042	027776					.WORD	T29BOT
	024044	015554					.WORD	EXPREC
121	024046			40\$:	CKLOOP	:LOOP IF SELECTED		
	024046	104406					TRAP	C\$CLP1
122	024050	013737	003116	026442	MOV FREE,T29RB	:ADDRESS OF READ BUFFER		
123	024056	012737	141011	026440	MOV #141011,T29PK3	:WRITE TAPE MARK RETRY,CVC=1,ACK COMMAND		
124	024064	012704	026440		MOV #T29PK3,R4	:SET UP R4 WITH PACKET ADDRESS		
125	024070	010465	000000		MOV R4,TSDB(R5)	:ISSUE COMMAND		
126	024074	004737	016330		JSR PC,WAITF	:WAIT FOR SSR TO SET		
127	024100	016501	000002		MOV TSSR(R5),R1	:GET TSSR CONTENTS		
128	024104	012702	100206		MOV #SSR!SC!BIT1!BIT2,R2	:SET UP EXPECTED		
129	024110	020102			CMP R1,R2	:ARE THEY EQUAL		
130	024112	001406			BEQ 75\$:BR, IF OK		
131	024114	005237	002214		INC FATFLG	:ERROR COUNT		
135	024120				ERRHRD ERRNO,T29WDE,PKTSSR	:TSSR INCORRECT AFTER READ DATA		
	024120	104456					TRAP	C\$ERHRD
	024122	000152					.WORD	106
	024124	027562					.WORD	T29WDE
	024126	012126					.WORD	PKTSSR
136	024130			75\$:	CKLOOP	:LOOP IF SELECTED		
	024130	104406					TRAP	C\$CLP1
137	024132	013701	026350		MOV T29BFR+6,R1	:GET XSTO STATUS WORD		
138	024136	010102			MOV R1,R2	:SET UP EXPECTED		
139	024140	052702	002000		BIS #BIT10,R2	:SET THE NEF BIT		
140	024144	020102			CMP R1,R2	:ARE THEY EQUAL		
141	024146	001406			BEQ 170\$:BR, IF EQUAL (GOOD)		
142	024150	005237	002214		INC FATFLG	:ERROR COUNT		
146	024154				ERRHRD ERRNO,T29NEF,EXPREC	:NEF SHOULD BE SET		
	024154	104456					TRAP	C\$ERHRD

SEQ 0123

024156 000153
024160 026630
024162 015554
147 024164 170\$:
148 024164 005103
149 024166 001273
150 024170
024170
024170 104403
151 024172 023727 002214 000017
152 024200 103402
153 024202 004737 017262
154 024206 999\$:

COM R3
BNF 26\$
ENDCUB

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

;RESET THE SWITCH
;BR, IF FIRST TIME THROUGH HERE

L10037:

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

.WORD 107
.WORD T29NEF
.WORD EXPREC

206	024360	020102				CMP	R1,R2		:DOES EXP = REC'D
207	024362	001406				BEQ	40\$:BR, IF EQUAL (OK)
208	024364	005237	002214			INC	FATFLG		:ERROR COUNT
212	024370					ERRHRD	ERRNO,T29BOT,EXPREC		:TAPE NOT AT BOT AFTER REWIND
	024370	104456							TRAP C\$ERHRD
	024372	000157							.WORD 111
	024374	027776							.WORD T29BOT
	024376	015554							.WORD EXPREC
213	024400	012737	000001	026442	40\$:	MOV	#1,T29RB		:NUMBER OF RECORDS TO SPACE OVER
214	024406	012737	000400	026446		MOV	#256,T29SZ		:SET UP RECORD SIZE
215	024414	012737	140005	026440		MOV	#140005,T29PK3		:WRITE FORWARD,CVC=1,ACK COMMAND
216	024422	012704	026440			MOV	#T29PK3,R4		:SET UP R4 WITH PACKET ADDRESS
217	024426	010465	000000			MOV	R4,TSDB(R5)		:ISSUE COMMAND
218	024432	004737	016330			JSR	PC,WAITF		:WAIT FOR SSR TO SET
219	024436	016501	000002			MOV	TSSR(R5),R1		:GET TSSR CONTENTS
220	024442	012702	000200			MOV	#SSR,R2		:SET UP EXPECTED
221	024446	020102				CMP	R1,R2		:ARE THEY EQUAL
222	024450	001420				BEQ	75\$:BR, IF OK
223	024452	013703	026350			MOV	T29BFR+6,R3		:PICK UP XT50
224	024456	032703	000004			BIT	#4,R3		:IS UNIT WRITE-LOCKED?
225	024462	001405				BEQ	41\$:NO,PROCEED WITH NORMAL ERROR
226	024464					ERRDF	ERRNO,T29WLK,SFIMSG		:TAPE IS WRITE LOCKED
	024464	104455							TRAP C\$ERDF
	024466	000157							.WORD 111
	024470	027644							.WORD T29WLK
	024472	012114							.WORD SFIMSG
227	024474					DOCLN			:DROP IT
	024474	104444							TRAP C\$DCLN
228	024476	005237	002214		41\$:	INC	FATFLG		:ERROR COUNT
232	024502					ERRHRD	ERRNO,T29WRT,PKTSSR		:TSSR INCORRECT AFTER WRITE DATA
	024502	104456							TRAP C\$ERHRD
	024504	000160							.WORD 112
	024506	027731							.WORD T29WRT
	024510	012126							.WORD PKTSSR
233	024512				75\$:	CKLOOP			:LOOP IF SELECTED
	024512	104406							TRAP C\$CLP1
234	024514	012737	000001	026442		MOV	#1,T29RB		:NUMBER OF RECORDS TO SPACE OVER
235	024522	012737	140410	026440		MOV	#140410,T29PK3		:SET UP COMMAND IN APCKET
236	024530	012704	026440			MOV	#T29PK3,R4		:SET UP R4 WITH PACKET ADDRESS
237	024534	010465	000000			MOV	R4,TSDB(R5)		:ISSUE COMMAND
238	024540	004737	016330			JSR	PC,WAITF		:WAIT FOR SSR TO SET
239	024544	016501	000002			MOV	TSSR(R5),R1		:GET TSSR CONTENTS
240	024550	012702	000200			MOV	#SSR,R2		:SET UP EXPECTED
241	024554	020102				CMP	R1,R2		:ARE THEY EQUAL
242	024556	001406				BEQ	175\$:BR, IF OK
243	024560	005237	002214			INC	FATFLG		:ERROR COUNT
247	024564					ERRHRD	ERRNO,T29WDE,PKTSSR		:TSSR INCORRECT AFTER READ DATA
	024564	104456							TRAP C\$ERHRD
	024566	000161							.WORD 113
	024570	027562							.WORD T29WDE
	024572	012126							.WORD PKTSSR
248	024574				175\$:	CKLOOP			:LOOP IF SELECTED
	024574	104406							TRAP C\$CLP1
249	024576	013737	003116	026442		MOV	FREE,T29RB		:ADDRESS OF BUFFER
250	024604	012737	141011	026440		MOV	#141011,T29PK3		:WRITE TAPE MARK RETRY,ACK,CVC=1 CMD.
251	024612	012704	026440			MOV	#T29PK3,R4		:SET UP R4 WITH PACKET ADDRESS
252	024616	010465	000000			MOV	R4,TSDB(R5)		:ISSUE COMMAND

:SET

253	024622	004737	016330	JSR	PC, WAITF	:WAIT FOR SSR TO SET	
254	024626	016501	000002	MOV	TSSR(R5), R1	:GET TSSR CONTENTS	
255	024632	012702	100204	MOV	#SSR!SC!BIT2, R2	:SET UP EXPECTED	
256	024636	020102		CMP	R1, R2	:ARE THEY EQUAL	
257	024640	001406		BEQ	180\$:BR, IF OK	
258	024642	005237	002214	INC	FATFLG	:ERROR COUNT	
262	024646			ERRHRD	ERRNO, T29WDE, PKTSSR	:TSSR INCORRECT AFTER READ DATA	
	024646	104456				TRAP	C\$ERHRD
	024650	000162				.WORD	114
	024652	027562				.WORD	T29WDE
	024654	012126				.WORD	PKTSSR
263	024656			180\$: CKLOOP		:LOOP IF SELECTED	
	024656	104406				TRAP	C\$CLP1
264	024660	013701	026356	MOV	T29BFR+14, R1	:GET XST3 STATUS WORD	
265	024664	010102		MOV	R1, R2	:SET UP EXPECTED	
266	024666	052702	000001	BIS	#BIT0, R2	:SET THE RIB BIT	
267	024672	020102		CMP	R1, R2	:ARE THEY EQUAL	
268	024674	001406		BEQ	190\$:BR, IF EQUAL (GOOD)	
269	024676	005237	002214	INC	FATFLG	:ERROR COUNT	
273	024702			ERRHRD	ERRNO, T29RIB, EXPREC	:NEF SHOULD BE SET	
	024702	104456				TRAP	C\$ERHRD
	024704	000163				.WORD	115
	024706	031724				.WORD	T29RIB
	024710	015554				.WORD	EXPREC
274	024712			190\$:			
275	024712			ENDSUB		:>>>>>>>>> END SUBTEST >>>>>>>>>	
	024712					L10040:	
	024712	104403				TRAP	C\$ESUB
276	024714	023727	002214 000017	CMP	FATFLG, #15.	:IS ERROR COUNT AT 25	
277	024722	103402		BLO	999\$:BR, IF LESS THAN 25	
278	024724	004737	017262	JSR	PC, CKDROP	:TRY TO DROP THE UNIT	
279	024730			999\$:			

Line	Address	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464
------	---------	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Line	Address	Offset	Label	Operation	Comments	Register/Value	Control	Trap	Word	Value
326	025122	030305							.WORD	T29RWN
	025124	012126							.WORD	PKTSSR
	025126	104406							TRAP	C\$CLP1
327	025130	013701	026350	MOV	T29BFR+6,R1					
328	025134	010102		MOV	R1,R2					
329	025136	052702	000002	BIS	#BIT1,R2					
330	025142	020102		CMP	R1,R2					
331	025144	001406		BEQ	40\$					
332	025146	005237	002214	INC	FATFLG					
336	025152			ERRHRD	ERRNO,T29BOT,EXPREC					
	025152	104456							TRAP	C\$ERHRD
	025154	000167							.WORD	119
	025156	027776							.WORD	T29BOT
	025160	015554							.WORD	EXPREC
337	025162									
	025162	104406								
338	025164	012737	140011	MOV	#140011,T29PK3				TRAP	C\$CLP1
339	025172	012704	026440	MOV	#T29PK3,R4					
340	025176	010465	000000	MOV	R4,TSDB(R5)					
341	025202	004737	016330	JSR	PC,WAITF					
342	025206	016501	000002	MOV	TSSR(R5),R1					
343	025212	012702	000200	MOV	#SSR,R2					
344	025216	020102		CMP	R1,R2					
345	025220	001406		BEQ	70\$					
346	025222	005237	002214	INC	FATFLG					
350	025226			ERRHRD	ERRNO,T29WDC,PKTSSR					
	025226	104456							TRAP	C\$ERHRD
	025230	000170							.WORD	120
	025232	030677							.WORD	T29WDC
	025234	012126							.WORD	PKTSSR
351	025236									
	025236	104406								
352	025240	012703	000001						TRAP	C\$CLP1
353	025244	012737	141011	MOV	#141011,T29PK3					
354	025252	012704	026440	MOV	#T29PK3,R4					
355	025256	010465	000000	MOV	R4,TSDB(R5)					
356	025262	004737	016330	JSR	PC,WAITF					
357	025266	016501	000002	MOV	TSSR(R5),R1					
358	025272	012702	000200	MOV	#SSR,R2					
359	025276	020102		CMP	R1,R2					
360	025300	001406		BEQ	165\$					
361	025302	005237	002214	INC	FATFLG					
365	025306			ERRHRD	ERRNO,T29WDC,PKTSSR					
	025306	104456							TRAP	C\$ERHRD
	025310	000171							.WORD	121
	025312	030677							.WORD	T29WDC
	025314	012126							.WORD	PKTSSR
366	025316									
	025316	104406								
367	025320	012737	140401	MOV	#140401,T29PK3				TRAP	C\$CLP1
368	025326	013737	003116	MOV	FREE,T29RB					
369	025334	012704	026440	MOV	#T29PK3,R4					
370	025340	0104								

374	025360	020102		CMP	R1,R2	:ARE THEY EQUAL		
375	025362	001406		BEQ	222\$:BR, IF OK		
376	025364	005237	002214	INC	FATFLG	:ERROR COUNT		
380	025370			ERRHRD	ERRNO,T29RDG,PKTSSR	:TSSR INCORRECT AFTER SPACE CMD.		
	025370	104456					TRAP	C\$ERHRD
	025372	000172					.WORD	122
	025374	031643					.WORD	T29RDG
	025376	012126					.WORD	PKTSSR
381	025400			222\$: CKLOOP		:LOOP IF SELECTED		
	025400	104406					TRAP	C\$CLP1
382	025402	013701	026350	MOV	T29BFR+6,R1	:PICK UP XSTO		
383	025406	010102		MOV	R1,R2	:SET UP EXPECTED		
384	025410	052702	100000	BIS	#BIT15,R2	:TMK SHOULD BE SET		
385	025414	020102		CMP	R1,R2	:IS TMK SET		
386	025416	001406		BEQ	226\$:BR, IF TMK WAS SET (GOOD)		
387	025420	005237	002214	INC	FATFLG	:ERROR COUNT		
391	025424			ERRHRD	ERRNO,T29RRN,EXPREC	:TMK NOT SET AFTER READ REV		
	025424	104456					TRAP	C\$ERHRD
	025426	000173					.WORD	123
	025430	032024					.WORD	T29RRN
	025432	015554					.WORD	EXPREC
392	025434			226\$: CKLOOP		:LOOP IF SELECTED		
	025434	104406					TRAP	C\$CLP1
393	025436			ENDSUB		:<<<<<<<<< END SUBTEST >>>>>>>>		
	025436					L10041:		
	025436	104403					TRAP	C\$ESUB
394	025440	023727	002214 000017	CMP	FATFLG,#15.	:IS ERROR COUNT AT 25		
395	025446	103402		BLO	999\$:BR, IF LESS THAN 25		
396	025450	004737	017262	JSR	PC,CKDROP	:TRY TO DROP THE UNIT		
397	025454			999\$:				

```

: +
:
: TEST 1, SUBTEST 4
:
: VERIFIES THAT THE SPACE-REVERSE PORTION OF THE WRITE TAPE MARK
: RETRY OPERATION IS PERFORMED BY REWINDING THE TAPE, ISSUING SEVERAL
: WRITE TAPE MARK RETRY COMMANDS IN SUCCESSION, THEN ISSUING TWO SPACE
: RECORDS REVERSE COMMANDS IN SUCCESSION.  THE SECOND SPACE RECORDS REVERSE
: COMMAND SHOULD TERMINATE WITH REVERSE INTO BOT (RIB) STATUS SET.

```

[illegible]

Line	Address	Offset	Label	Instruction	Comment	Trap	Word
	025642	104456				TRAP	C\$ERHRD
	025644	000176				.WORD	126
	025646	030305				.WORD	T29RWN
	025650	012126				.WORD	PKTSSR
446	025652		30\$:	CKLOOP	;LOOP IF SELECTED		
	025652	104406				TRAP	C\$CLP1
447	025654	013701	026350	MOV	T29BFR+6,R1		
448	025660	010102		MOV	R1,R2		
449	025662	052702	000002	BIS	#BIT1,R2		
450	025666	020102		CMP	R1,R2		
451	025670	001406		BEQ	40\$		
452	025672	005237	002214	INC	FATFLG		
456	025676			ERRHRD	ERRNO,T29BOT,EXPREC		
	025676	104456					
	025700	000177				TRAP	C\$ERHRD
	025702	027776				.WORD	127
	025704	015554				.WORD	T29BOT
						.WORD	EXPREC
457	025706		40\$:	CKLOOP	;LOOP IF SELECTED		
	025706	104406				TRAP	C\$CLP1
458	025710	012737	140011	MOV	#140011,T29PK3		
459	025716	012704	026440	MOV	#T29PK3,R4		
460	025722	010465	000000	MOV	R4,TSDB(R5)		
461	025726	004737	016330	JSR	PC,WAITF		
462	025732	016501	000002	MOV	TSSR(R5),R1		
463	025736	012702	000200	MOV	#SSR,R2		
464	025742	020102		CMP	R1,R2		
465	025744	001406		BEQ	70\$		
466	025746	005237	002214	INC	FATFLG		
470	025752			ERRHRD	ERRNO,T29WDC,PKTSSR		
	025752	104456					
	025754	000200				TRAP	C\$ERHRD
	025756	030677				.WORD	128
	025760	012126				.WORD	T29WDC
						.WORD	PKTSSR
471	025762		70\$:	CKLOOP	;LOOP IF SELECTED		
	025762	104406				TRAP	C\$CLP1
472	025764	012703	000012	MOV	#10,R3		
473	025770	012737	000001	MOV	#1,T29RB		
474	025776	012737	141011	MOV	#141011,T29PK3		
475	026004	012704	026440	MOV	#T29PK3,R4		
476	026010	010465	000000	MOV	R4,TSDB(R5)		
477	026014	004737	016330	JSR	PC,WAITF		
478	026020	016501	000002	MOV	TSSR(R5),R1		
479	026024	012702	000200	MOV	#SSR,R2		
480	026030	020102		CMP	R1,R2		
481	026032	001406		BEQ	165\$		
482	026034	005237	002214	INC	FATFLG		
486	026040			ERRHRD	ERRNO,T29WDC,PKTSSR		
	026040	104456					
	026042	000201				TRAP	C\$ERHRD
	026044	030677				.WORD	129
	026046	012126				.WORD	T29WDC
						.WORD	PKTSSR
487	026050		165\$:	CKLOOP	;LOOP IF SELECTED		
	026050	104406				TRAP	C\$CLP1
488	026052	005303		DEC	R3		
489	026054	001355		BNE	155\$		
490	026056	012737	140410	MOV	#140410,T29PK3		
491	026064	012737	000001	MOV	#1,T29RB		

492	026072	012704	026440	MOV	#T29PK3,R4	:SET UP R4 WITH PACKET ADDRESS
493	026076	010465	000000	MOV	R4,TSDB(R5)	:ISSUE COMMAND
494	026102	004737	016330	JSR	PC,WAITF	:WAIT FOR SSR TO SET
495	026106	016501	000002	MOV	TSSR(R5),R1	:GET TSSR CONTENTS
496	026112	012702	100204	MOV	#SSR!SC!BIT2,R2	:SET UP EXPECTED
497	026116	020102		CMP	R1,R2	:ARE THEY EQUAL
498	026120	001406		BEQ	222\$:BR, IF OK
499	026122	005237	002214	INC	FATFLG	:ERROR COUNT
503	026126			ERRHRD	ERRNO,T29WDE,PKTSSR	:TSSR INCORRECT AFTER SPACE CMD.
	026126	104456				TRAP C\$ERHRD
	026130	000202				.WORD 130
	026132	027562				.WORD T29WDE
	026134	012126				.WORD PKTSSR
504	026136			222\$: CKLOOP		:LOOP IF SELECTED
	026136	104406				TRAP C\$CLP1
505	026140	012737	100410 026440	MOV	#100410,T29PK3	:SPACE REVERSE,ACK, COMMAND
506	026146	012737	000005 026442	MOV	#5,T29RB	:NUMBER OF RECORDS TO SPACE BACK
507	026154	012704	026440	MOV	#T29PK3,R4	:SET UP R4 WITH PACKET ADDRESS
508	026160	010465	000000	MOV	R4,TSDB(R5)	:ISSUE COMMAND
509	026164	004737	016330	JSR	PC,WAITF	:WAIT FOR SSR TO SET
510	026170	016501	000002	MOV	TSSR(R5),R1	:GET TSSR CONTENTS
511	026174	012702	100204	MOV	#SSR!SC!BIT2,R2	:SET UP EXPECTED
512	026200	020102		CMP	R1,R2	:ARE THEY EQUAL
513	026202	001406		BEQ	260\$:BR, IF OK
514	026204	005237	002214	INC	FATFLG	:ERROR COUNT
518	026210			ERRHRD	ERRNO,T29RDG,PKTSSR	:TSSR INCORRECT AFTER SPACE REV CMD.
	026210	104456				TRAP C\$ERHRD
	026212	000203				.WORD 131
	026214	031643				.WORD T29RDG
	026216	012126				.WORD PKTSSR
519	026220			260\$: CKLOOP		:LOOP IF SELECTED
	026220	104406				TRAP C\$CLP1
520	026222	013701	026356	MOV	T29BFR+14,R1	:PICK UP XST3
521	026226	010102		MOV	R1,R2	:SET UP EXPECTED
522	026230	052702	000001	BIS	#BIT0,R2	:RIB SHOULD BE SET
523	026234	020102		CMP	R1,R2	:IS RIB SET
524	026236	001406		BEQ	270\$:BR, IF RIB WAS SET (GOOD)
525	026240	005237	002214	INC	FATFLG	:ERROR COUNT
529	026244			ERRHRD	ERRNO,T29RIB,EXPREC	:TMK NOT SET AFTER READ REV
	026244	104456				TRAP C\$ERHRD
	026246	000204				.WORD 132
	026250	031724				.WORD T29RIB
	026252	015554				.WORD EXPREC
530	026254			270\$: CKLOOP		:LOOP IF SELECTED
	026254	104406				TRAP C\$CLP1
531	026256			330\$: CKLOOP		:LOOP IF SELECTED
	026256	104406				TRAP C\$CLP1
532	026260			ENDSUB		:<<<<<<<<< END SUBTEST >>>>>>>>>
	026260					L10042:
	026260	104403				TRAP C\$ESUB
533	026262	023727	002214 000017	CMP	FATFLG,#15.	:IS ERROR COUNT AT 25
534	026270	103402		BLO	999\$:BR, IF LESS THAN 25
535	026272	004737	017262	JSR	PC,CKDROP	:TRY TO DROP THE UNIT
536	026276			999\$:		

538
539
540
541
542
543
544
545
026310
026312

026276 004737 016536
026302 103002
026304 000137 023556
026310 104432
026312 004020

..
..
..

JSR PC,TSTLOOP
BCC 163\$
JMP T29LOOP
163\$: EXIT TST

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

TRAP C\$EXIT
.WORD L10036-.

547		;	+			
548		;	LOCAL STORAGE FOR THIS TEST			
549		;	-			
551	026320			.=<.+10>8177770		
553	026320		T29PACKET:			;COMMAND PACKET FOR TEST
554	026320	014004		.WORD 14004		;WRITE CHARACTERISTICS COMMAND, WITH CVC=1, ACK
555	026322	026330		.WORD T29DATA		;ADDRESS OF CHARACTERISTICS BLOCK
556	026324	000000		.WORD 0		
557	026326	000012		.WORD 10.		;STARTING VALUE OF BLOCK SIZE
558	026330		T29DATA:			;CHARACTERISTICS DATA BLOCK
559	026330	026342		.WORD T29BFR		;ADDRESS OF MESSAGE BUFFER
560	026332	000000		.WORD 0		
561	026334	000024		.WORD 20.		;LENGTH OF MESSAGE BUFFER
562	026336	000000		.WORD 0		
563	026340	000000	T29DSW:	.WORD 0		;SELECT DRIVE 0
564	026342		T29BFR:	.BLKW 25.		;MESSAGE BUFFER
565			;			
566			;	WRITE SUBSYSTEM MEMORY COMMAND PACKET		
567			;			
569	026430			.=<.+10>8177770		
571	026430		T29PK2:			
572	026430	100006		.WORD 100006		;WRITE SUB SYS MEM COMMAND, AND ACK
573	026432	026450		.WORD T29BF2		;ADDRESS OF SELECT BLOCK DATA
574	026434	000000		.WORD 0		
575	026436	000006		.WORD 6.		;SIZE OF DATA PACKET
576						
580	026440		T29PK3:			
581	026440	140005		.WORD 140005		;WRITE TAPE MARK RETRY COMMAND, CVC=1 AND ACK
582	026442		T29RB:			
583	026442	003116	T29WB:	.WORD FREE		;ADDRESS OF WRITE BUFFER
584	026444	000000		.WORD 0		
585	026446	000000	T29SZ:	.WORD 0		;SIZE OF BUFFER (EXTENT)
586				.EVEN		
587			;			
588			;			
589			;			
590	026450		T29BF2:			
591	026450	010	T29BS0:	.BYTE 10		;BSEL0 AREA
592	026451	200	T29BS1:	.BYTE 200		;BSEL1 AREA
593	026452	000000	T29S2:	.WORD 0		;SEL 2 AREA
594	026454	000000	T29S3:	.WORD 0		;DATA AREA
595			;			
596			;			
597				.EVEN		
598			;	TAPE MOTION PACKET COMMAND VALUES		
599						
600	026456	140001	T29RN:	.WORD 140001		;READ DATA
601	026460	140401	T29WDR:	.WORD 140401		;READ DATA REVERSE
602	026462	141001	T29CON:	.WORD 141001		;READ PREVIOUS OPP=0
603	026464	161001		.WORD 161001		;READ PREVIOUS OPP=1
604	026466	141401		.WORD 141401		;WRITE TAPE MARK RETRY NEXT OPP=0
605	026470	161401		.WORD 161401		;WRITE TAPE MARK RETRY NEXT OPP=1
606	026472	177777		.WORD 177777		;END OF DATA
607						
608			;			
609	026474	000000	T29CNT:	.WORD 0		;TAPE RECORD COUNTER STORAGE AREA
610						

F 11
TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 25-MAY-82 08:43 PAGE 85-1
TEST 1: WRITE TAPE MARK RETRY

SEQ 0135

611 026476 000000
612 026500 000000
613

T29RSZ: .WORD 0
T29DLY: .WORD

:RECORD STORAGE SIZE AREA
:DELAY COUNTER STORAGE AREA

```

615
616
617      ;+
618      ;LOCAL TEXT MESSAGES FOR TEST
619      ;--
620
621 026502      104      162      151  T290FL: .ASCIIZ 'Drive is OFFLINE'
622 026523      124      141      160  T29WNG: .ASCIIZ 'Tape Position Incorrect After WRITE TAPE MARK RETRY Previous (OPP=1)'
623 026630      127      122      111  T29NEF: .ASCIIZ 'WRITE TAPE MARK RETRY, At BOT, Failed To Set NEF (XST0)'
624 026720      124      123      123  T29RDF: .ASCIIZ 'TSSR Incorrect After READ DATA Command'
625 026767      127      122      111  T29RRF: .ASCIIZ 'WRITE TAPE MARK RETRY Previous (Space Reverse, Read Forward) Command Failed
626 027103      127      122      111  T29RRG: .ASCIIZ 'WRITE TAPE MARK RETRY Previous (Read Forward, Space Reverse) Command Failed
627 027217      120      117      123  T29SC: .ASCIIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
628 027301      122      111      102  T29LOR: .ASCIIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
629 027351      124      123      123  T29WDF: .ASCIIZ 'TSSR Not Correct After Illegal Mode Bits Set'
630 027426      111      154      154  T29LOQ: .ASCIIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
631 027507      127      122      111  T29SSR: .ASCIIZ 'WRITE TAPE MARK RETRY COMMAND Not Accepted'
632 027562      124      123      123  T29WDE: .ASCIIZ 'TSSR Not Correct After SPACE REVERSE DATA Command'
633 027644      052      052      052  T29WLK: .ASCIIZ '*****TAPE IS WRITE-LOCKED AND WILL CAUSE ERRORS*****'
634 027731      124      123      123  T29WRT: .ASCIIZ 'TSSR Not Correct After WRITE Command'
635 027776      124      141      160  T29BOT: .ASCIIZ 'Tape Not At BOT After REWIND Command'
636 030043      104      141      164  T29DTA: .ASCIIZ 'Data Written To Tape Not Equal To Data Read From Tape'
637 030131      127      122      111  T29EOT: .ASCIIZ 'WRITE TAPE MARK RETRY DATA OVER EOT GAVE NO TAPE STATUS ALERT'
638 030227      124      123      123  T29TM: .ASCIIZ 'TSSR Not Correct After SPACE REVERSE Into BOT'
639 030305      122      145      167  T29RWN: .ASCIIZ 'Rewind (POSITION) Command Not Accepted'
640 030354      122      101      115  T29RNC: .ASCIIZ 'RAM Error, Correct Data Pattern Not In Ram'
641 030427      124      123      123  T29AM3: .ASCIIZ 'TSSR Init. Failed After WRITE TAPE MARK RETRY COMMAND'
642 030515      104      162      151  T29OF7: .ASCIIZ 'Drive 7 select Failed To Set 'OFL' In TSSR'
643 030570      124      123      123  T29WDD: .ASCIIZ 'TSSR Not Correct After WRITE TAPE MARK RETRY DATA Command, SWB Bit Set'
644 030677      124      123      123  T29WDC: .ASCIIZ 'TSSR Not Correct After WRITE TAPE MARK RETRY DATA Command'
645 030771      103      126      103  T29VCK: .ASCIIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
646 031044      124      123      102  T29BA: .ASCIIZ 'TSBA Not Correct After WRITE TAPE MARK RETRY DATA Command'
647 031136      127      122      111  T29WSS: .ASCIIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
648 031225      122      145      141  T29LON: .ASCIIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
649 031307      122      145      141  T29LOP: .ASCIIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
650 031371      122      145      163  T29PBP: .ASCIIZ 'Residual Byte Count Incorrect After Short Record Read'
651 031457      122      145      141  T29TRL: .ASCIIZ 'Reading Long Record Failed To Give Tape Status Alert'
652 031545      104      141      164  T29NEQ: .ASCIIZ 'Data WRITE TAPE MARK RETRY From Tape Not Correct, After SWB=1'
653 031643      124      123      123  T29RDG: .ASCIIZ 'TSSR Incorrect After READ REVERSE Into Tape Mark'
654 031724      127      122      111  T29RIB: .ASCIIZ 'WRITE TAPE MARK RETRY At First Record, Failed To Set RIB (XST3)'
655 032024      124      115      113  T29RRN: .ASCIIZ 'TMK (XST0) Failed To Set After READ REVERSE Into Tape Mark'
656 032117      127      162      151  T29ID: .ASCIIZ 'Write Tape Mark Retry'
657
658      .EVEN
659
660      ;+
661      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
662      ;WRITE SUBSYSTEM MEMORY COMMAND
663      ;--
664
665 032146
666 032146
667 032152      012701      026320
668 032156      012721      140004
669 032162      012721      026330
670 032166      005021
671 032170      012721      000012

T29REST:
      SAVREG
      MOV      #T29PACKET,R1
      MOV      #140004,(R1)+
      MOV      #T29DATA,(R1)+
      CLR      (R1)+
      MOV      #10.,(R1)+
      ;SAVE THE REGISTERS
      ;START OF THE PACKET
      ;WRITE SUBSYSTEM MEM. WITH ACK, CVC=1
      ;ADDRESS OF CHARAISTICS DATA BLOCK
      ;EXTENDED ADDRESS
      ;SIZE OF DATA BLOCK IN BYTES
  
```

672	032174	012721	026342	MOV	#T29BFR,(R1)+	:ADDRESS OF MESSAGE BUFFER
673	032200	005021		CLR	(R1)+	
674	032202	012721	000024	MOV	#20,(R1)+	:LENGTH OF MESSAGE BUFFER
675	032206	005021		CLR	(R1)+	
676	032210	012711	000000	MOV	#0,(R1)	:SELECT DRIVE ZERO (0)
677	032214	012702	000030	MOV	#24,R2	:NUMBER OF LOCATIONS TO BE CLEARED
678	032220	012762	177777 026342 64\$:	MOV	#177777,T29BFR(R2)	:ALL ONES TO MESSAGE BUFFER
679	032226	005742		TST	-(R2)	:NEXT LOCATION
680	032230	00227	000000	CMP	R2,#0	:CHECK FOR END OF LOOP
681	032234	001371		BNE	64\$:KEEP GOING UNTIL DONE
682	032236	000207		RTS	PC	:RETURN
683						
684						
685	032240			T29RT2:	SAVREG	:SAVE THE REGISTERS
686	032240			MOV	#T29PK2,R1	:START OF THE PACKET
687	032244	012701	026430	MOV	#140006,(R1)+	:WRITE SUBSYSTEM MEM. WITH ACK,CVC=1,
688	032250	012721	140006	MOV	#T29BF2,(R1)+	:ADDRESS OF DATA BLOCK
689	032254	012721	026450	CLR	(R1)+	:EXTENDED ADDRESS
690	032260	005021		MOV	#6,(R1)+	:SIZE OF DATA BLOCK IN BYTES
691	032262	012721	000006	CLR	(R1)+	
692	032266	005021		MOV	#T29BF2,R1	:POINT TO DATA SEL AREA
693	032270	012701	026450	CLR	(R1)+	
694	032274	005021		CLR	(R1)	
695	032276	005011		RTS	PC	:RETURN
696	032300	000207				
697	032302			T29RT3:	SAVREG	:SAVE THE REGISTERS
698	032302			MOV	#T29PK3,R1	:START OF THE PACKET
699	032306	012701	026440	MOV	#0,(R1)+	:WRITE SUBSYSTEM MEM. WITH ACK,
700	032312	012721	000000	MOV	#0,(R1)+	:ADDRESS OF DATA BLOCK
701	032316	012721	000000	CLR	(R1)+	:EXTENDED ADDRESS
702	032322	005021		MOV	#0,(R1)	:SIZE OF DATA BLOCK IN BYTES
703	032324	012711	000000	RTS	PC	:RETURN
704	032330	000207		ENDTST		
705	032332					
	032332					
	032332	104401				

L10036: TRAP CSETST


```

766 032372 004737 041344      JSR      PC,T30RT2      ;SET UP OTHER COMMAND PACKET
767 032376 004737 041406      JSR      PC,T30RT3      ;SET UP OTHER COMMAND PACKET
768 032402 012737 176750 036656 10$: MOV      #65000.,T30DLY    ;SET UP DELAY COUNTER
769 032410 004737 016054      JSR      PC,SOFINIT    ;DO INITIALIZE ON CONTROLLER
770 032414 103426      BCS      20$      ;BR IF INIT WAS OK
771 032416      DELAY      250      ;DELAY ROUTINE CALL
      032416 012727 000250      MOV      #250,(PC)+
      032422 000000      .WORD      0
      032424 013727 002116      MOV      L$DLY,(PC)+
      032430 000000      .WORD      0
      032432 005367 177772      DEC      -6(PC)
      032436 001375      BNE      -4
      032440 005367 177756      DEC      -22(PC)
      032444 001367      BNE      -20
772 032446 005337 036656      DEC      T30DLY      ;BUMP COUNTER
773 032452 001356      BNE      10$      ;BR, IF MORE COUNTING TO DO
774 032454 005237 002214      INC      FATFLG      ;ERROR COUNT
778 032460 010001      MOV      R0,R1      ;CONTENTS OF TSSR REGISTER
779 032462      ERRDF      ERRNO,SFIERR,SFIMSG    ;FATAL ERROR TSSR WAS NOT OK
      032462 104455      TRAP      C$ERDF
      032464 000311      .WORD      201
      032466 003646      .WORD      SFIERR
      032470 012114      .WORD      SFIMSG
780 032472      20$:
781 032472 013737 002174 036520 MOV      UNITN,T30DSW      ;SET UP UNIT NUMBER
782 032500 012704 036500      MOV      #T30PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
783
784      ;*****
785      ;ISSUE WRITE CHARACTERISTICS COMMAND
786      ;*****
787
788
789
790 032504 004737 010742      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
791 032510 103407      BCS      23$      ;BR, IF COMMAND ISSUED OK
792 032512 005237 002214      INC      FATFLG      ;ERROR COUNT
796 032516 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
797 032520      ERRHRD      ERRNO,WRTMSG,SFIMSG    ;WRITE CHARACTERISTICS FAILED
      032520 104456      TRAP      C$ERHRD
      032522 000312      .WORD      202
      032524 005052      .WORD      WRTMSG
      032526 012114      .WORD      SFIMSG
798 032530      23$: CKLOOP      ;LOOP IF SELECTED
      032530 104406      TRAP      C$CLP1
799
800      ;*****
801      ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
802      ;*****
803
804
805
806 032532 004737 011074      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
807 032536 103411      BCS      30$      ;BR, IF NO PROBLEM
808 032540 010004      MOV      R0,R4      ;GET PACKET ADDRESS
809 032542 016501 000002      MOV      TSSR(R5),R1    ;GET STATUS REGISTER
810 032546 005237 002214      INC      FATFLG      ;ERROR COUNT
814 032552      ERRHRD      ERRNO,T30RWN,PKTSSR    ;REWIND NOT ACCEPTED

```


Line	Address	Hex	Hex	Hex	Hex	Instruction	Comment	Trap	Trap Data
	032552	104456						TRAP	C\$ERHRD
	032554	000313						.WORD	203
	032556	040240						.WORD	T30RWN
	032560	012126						.WORD	PKTSSR
815	032562				30\$:	CKLOOP	:LOOP IF SELECTED		
	032562	104406						TRAP	C\$CLP1
816									
817									
818									
819									
820									
821									
822									
823	032564	013701	036530			MOV T30BFR+6,R1	:PICK UP XSTO		
824	032570	010102				MOV R1,R2	:SET UP EXPECTED		
825	032572	052702	000002			BIS #BIT1,R2	:SET BOT BIT IN EXPECTED		
826	032576	020102				CMP R1,R2	:DOES EXP = REC'D		
827	032600	001406				BEQ 40\$:BR, IF EQUAL (OK)		
828	032602	005237	002214			INC FATFLG	:ERROR COUNT		
832	032606					ERRHRD ERRNO,T30BOT,EXPREC	:TAPE NOT AT BOT AFTER REWIND		
	032606	104456						TRAP	C\$ERHRD
	032610	000314						.WORD	204
	032612	040041						.WORD	T30BOT
	032614	015554						.WORD	EXPREC
833	032616				40\$:	CKLOOP	:LOOP IF SELECTED		
	032616	104406						TRAP	C\$CLP1
834	032620	012737	000001	036654		MOV #1.,T30FCN	:SET 'FILE' COUNTER AT 1 DECIMAL		
835	032626	012703	000001		64\$:	MOV #1,R3	:ONE RECORD PER 'FILE'		
836	032632	013737	003116	036622	65\$:	MOV FREE,T30WB	:SET UP PACKETS'S WRITE BUFFER		
837	032640	012737	003720	036626		MOV #2000.,T30SZ	:SET RECORD SIZE AT 2000 BYTES		
838									
839									
840									
841									
842									
843									
844									
845	032646	012737	140005	036620		MOV #140005,T30PK3	:WRITE DATA,ACK,CVC=1 COMMAND		
846	032654	012704	036620			MOV #T30PK3,R4	:SET UP R4 WITH PACKET ADDRESS		
847	032660	013702	036654			MOV T30FCN,R2	:GET FILE COUNTER		
848	032664	000302				SWAB R2	:MOVE TO UPPER BYTE		
849	032666	010301				MOV R3,R1	:GET RECORD COUNTER		
850	032670	060201				ADD R2,R1	:FILE COUNTER IN UPPER, RECORD # LOW		
851	032672	010177	150220			MOV R1,@FREE	:MOV TO OUT PUT BUFFER		
852	032676	010465	000000			MOV R4,TSD8(R5)	:ISSUE COMMAND		
853	032702	004737	016330			JSR PC,WAITF	:WAIT FOR SSR TO SET		
854	032706	016501	000002			MOV TSSR(R5),R1	:GET TSSR CONTENTS		
855	032712	012702	000200			MOV #SSR,R2	:SET UP EXPECTED		
856	032716	020102				CMP R1,R2	:ARE THEY EQUAL		
857	032720	001406				BEQ 70\$:BR, IF OK		
858	032722	005237	002214			INC FATFLG	:ERROR COUNT		
862	032726					ERRHRD ERRNO,T30WDD,PKTSSR	:TSSR INCORRECT AFTER WRITE DATA		
	032726	104456						TRAP	C\$ERHRD
	032730	000315						.WORD	205
	032732	037170						.WORD	T30WDD
	032734	012126						.WORD	PKTSSR
863	032736				70\$:	CKLOOP			

```

864 032736 104406
865 032740 005203
866 032742 020327 000021
867 032746 001331
868
869
870
871
872
873
874 032750 012737 141011 036620
875 032756 012704 036620
876 032762 010465 000000
877 032766 004737 016330
878 032772 016501 000002
879 032776 012702 000200
880 033002 020102
881 033004 001406
882 033006 005237 002214
886 033012
      033012 104456
      033014 000316
      033016 040362
      033020 012126
887 033022
      033022 104406
888 033024 005237 036654
889 033030 023727 036654 000006
890 033036 001273
891
892
893
894
895
896
897
898 033040 012737 141011 036620
899 033046 012704 036620
900 033052 010465 000000
901 033056 004737 016330
902 033062 016501 000002
903 033066 012702 000200
904 033072 020102
905 033074 001406
906 033076 005237 002214
910 033102
      033102 104456
      033104 000317
      033106 040362
      033110 012126
911 033112
      033112 104406
912
913
914
915

      INC R3
      CMP R3,#21
      BNE 65$
      TRAP C$CLP1
      ;COUNT THE RECORD COUNTER DOWN
      ;AT 20 YET
      ;BR, IF NOT AT 20 RECORDS WRITTEN

      *****
      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
      *****

      MOV #141011,T30PK3
      MOV #T30PK3,R4
      MOV R4,TSDB(R5)
      JSR PC,WAITF
      MOV TSSR(R5),R1
      MOV #SSR,R2
      CMP R1,R2
      BEQ 160$
      INC FATFLG
      ERRHRD ERRNO,T30WDC,PKTSSR
      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
      ;SET UP R4 WITH PACKET ADDRESS
      ;ISSUE COMMAND
      ;WAIT FOR SSR TO SET
      ;PICK UP TSSR
      ;SET UP EXPECTED (SSR ONLY)
      ;WAS STATUS GOOD
      ;BR, IF TERMINATION WAS GOOD
      ;ERROR COUNT
      ;TSSR NOT CORRECT AFTER WRT TAPE M.
      TRAP C$ERHRD
      .WORD 206
      .WORD T30WDC
      .WORD PKTSSR

160$: CKLOOP
      ;LOOP IF SELECTED
      TRAP C$CLP1
      ;COUNT THE 'FILE' COUNTER DOWN
      ;WRITE 5 FILE TO TAPE
      ;BR, IF NOT AT 5 FILES WRITTEN

      *****
      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
      *****

      MOV #141011,T30PK3
      MOV #T30PK3,R4
      MOV R4,TSDB(R5)
      JSR PC,WAITF
      MOV TSSR(R5),R1
      MOV #SSR,R2
      CMP R1,R2
      BEQ 165$
      INC FATFLG
      ERRHRD ERRNO,T30WDC,PKTSSR
      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
      ;SET UP R4 WITH PACKET ADDRESS
      ;ISSUE COMMAND
      ;WAIT FOR SSR TO SET
      ;PICK UP TSSR
      ;SET UP EXPECTED (SSR ONLY)
      ;WAS STATUS GOOD
      ;BR, IF TERMINATION WAS GOOD
      ;ERROR COUNT
      ;TSSR NOT CORRECT AFTER WRT TAPE M.
      TRAP C$ERHRD
      .WORD 207
      .WORD T30WDC
      .WORD PKTSSR

165$: CKLOOP
      ;LOOP IF SELECTED
      TRAP C$CLP1

      *****
      ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
  
```

```

916
917
918
919 033114 004737 011074      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
920 033120 103411      BCS      170$      ;BR, IF NO PROBLEM
921 033122 010004      MOV      R0,R4      ;GET PACKET ADDRESS
922 033124 016501 000002      MOV      TSSR(R5),R1      ;GET STATUS REGISTER
923 033130 005237 002214      INC      FATFLG      ;ERROR COUNT
927 033134      ERRHRD  ERRNO,T3ORWN,PKTSSR      ;REWIND NOT ACCEPTED
      033134 104456
      033136 000320      TRAP      C$ERHRD
      033140 040240      .WORD    208
      033142 012126      .WORD    T3ORWN
928 033144      170$:  CKLOOP      ;LOOP IF SELECTED      .WORD    PKTSSR
      033144 104406      TRAP      C$CLP1
929
930
931
932
933      ;*****
934      ;GET EXTENDED STATUS REGISTER ZERO (XST0) FROM MESSAGE BUFFER
935      ;*****
936 033146 013701 036530      MOV      T30BFR+6,R1      ;PICK UP XST0
937 033152 010102      MOV      R1,R2      ;SET UP EXPECTED
938 033154 052702 000002      BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
939 033160 020102      CMP      R1,R2      ;DOES EXP = REC'D
940 033162 001406      BEQ      180$      ;BR, IF EQUAL (OK)
941 033164 005237 002214      INC      FATFLG      ;ERROR COUNT
945 033170      ERRHRD  ERRNO,T30BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      033170 104456      TRAP      C$ERHRD
      033172 000321      .WORD    209
      033174 040041      .WORD    T30BOT
      033176 015554      .WORD    EXPREC
946 033200      180$:  CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
      033200 104406
947 033202 012703 036636      MOV      #T30IMV,R3      ;SET UP POINTER TO COMMAND TABLE
948 033206 013737 002174 036520      MOV      UNITN,T30DSW      ;SET UP UNIT NUMBER
949 033214 011337 036516      182$:  MOV      (R3),T30ETM      ;GET NEXT COMMAND
950 033220 012704 036500      MOV      #T30PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
951
952
953
954      ;*****
955      ;ISSUE WRITE CHARACTERISTICS COMMAND
956      ;*****
957
958 033224 004737 010742      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
959 033230 103407      BCS      188$      ;BR, IF COMMAND ISSUED OK
960 033232 005237 002214      INC      FATFLG      ;ERROR COUNT
964 033236 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
965 033240      ERRHRD  ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTICS FAILED
      033240 104456      .RAP      C$ERHRD
      033242 000322      .WORD    210
      033244 005052      .WORD    WRTMSG
      033246 012114      .WORD    SFIMSG
966 033250      188$:  CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
      033250 104406
  
```

```

967
968
969
970
971
972
973
974 033252 012737 141010 036620      MOV      #141010,T30PK3      ;SKIP TAPE MARK,ACK,CVC=1 COMMAND
975 033260 012737 000001 036622      MOV      #1,T30RB      ;SET UP NUMBER TO SKIP
976 033266 012704 036620      MOV      #T30PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
977 033272 010465 000000      MOV      R4,T30DB(R5)      ;ISSUE COMMAND
978 033276 012737 176750 036656      MOV      #65000,T30DLY      ;SET UP DELAY COUNTER
979 033304 004737 016330      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
980 033310 016501 000002      MOV      TSSR(R5),R1      ;PICK UP TSSR
981 033314 032701 000200      BIT      #SSR,R1      ;IS SSR SET YET
982 033320 001017      BNE      191$      ;BR, IF SSR IS SET
983 033322      DELAY      250      ;CALL DELAY ROUTINE
      MOV      #250,(PC)+
      .WORD      0
      MOV      L$DLY,(PC)+
      .WORD      0
      DEC      -6(PC)
      BNE      -4
      DEC      -22(PC)
      BNE      -20
984 033352 005337 036656      DEC      T30DLY      ;BUMP DELAY ROUTINE
985 033356 001352      BNE      190$      ;BR, IF MORE DELAY TO GO
986 033360 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED (SSR ONLY)
987 033364 020102      CMP      R1,R2      ;WAS STATUS GOOD
988 033366 001406      BEQ      192$      ;BR, IF TERMINATION WAS GOOD
989 033370 005237 002214      INC      FATFLG      ;ERROR COUNT
993 033374      ERRHRD      ERRNO,T30SKM,PKTSSR      ;TSSR NOT CORRECT AFTER
      SKIP TAPE M.
      TRAP      C$ERHRD
      .WORD      211
      .WORD      T30SKM
      .WORD      PKTSSR
994 033404      CKLOOP      ;LOOP IF SELECTED
      TRAP      C$CLP1
995
996
997
998
999
1000
1001
1002 033406 013701 036530      MOV      T30BFR+6,R1      ;PICK UP XST0
1003 033412 010102      MOV      R1,R2      ;SET UP EXPECTED
1004 033414 052702 100000      BIS      #BIT15,R2      ;SET TMK BIT IN EXPECTED
1005 033420 020102      CMP      R1,R2      ;DOES EXP = REC'D
1006 033422 001406      BEQ      195$      ;BR, IF EQUAL (OK)
1007 033424 005237 002214      INC      FATFLG      ;ERROR COUNT
1011 033430      ERRHRD      ERRNO,T30TMK,EXPREC      ;TMK NOT SET AFTER WRT TAPE MARK
      TRAP      C$ERHRD
      .WORD      212
      .WORD      T30TMK
      .WORD      EXPREC
1012 033440      CKLOOP      ;LOOP IF SELECTED
  
```

```

1013 033440 104406
1014 033442 012700 177777      MOV    #177777,R0      ;VALUE TO WRITTEN TO MEMORY
1015 033446 004737 017502      JSR     PC,FILLMEM    ;FILL MEM WITH ALL ONES
1016 033452 013737 003116 036622  MOV    FREE,T30RB    ;STARTING READ BUFFER ADDRESS
1017
1018 :*****
1019 :READ FORWARD,ACK,CVC=1 COMMAND
1020 :*****
1021
1022
1023 033460 012737 140001 036620      MOV    #140001,T30PK3  ;READ FORWARD,ACK,CVC=1 COMMAND
1024 033466 012704 036620      MOV    #T30PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
1025 033472 012737 003720 036626      MOV    #2000.,T30SZ  ;SET UP RECORD SIZE IN PACKET
1026 033500 010465 000000      MOV    R4,TSD8(R5)   ;ISSUE COMMAND
1027 033504 004737 016330      JSR     PC,WAITF     ;WAIT FOR SSR TO SET
1028 033510 016501 000002      MOV    TSSR(R5),R1  ;GET TSSR CONTENTS
1029 033514 012702 000200      MOV    #SSR,R2     ;SET UP EXPECTED
1030 033520 020102      CMP    R1,R2       ;ARE THEY EQUAL
1031 033522 001406      BEQ    200$        ;BR, IF OK
1032 033524 005237 002214      INC    FATFLG      ;ERROR COUNT
1036 033530      ERRHRD  ERRNO,T30RDF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
1037 033530 104456      TRAP   C$ERHRD
1038 033532 000325      .WORD  213
1039 033534 037413      .WORD  T30RDF
1040 033536 012126      .WORD  PKTSSR
1041 033540      200$:  CKLOOP      ;LOOP IF SELECTED
1042 033540 104406      TRAP   C$CLP1
1043 033542 017701 147350      MOV    @FREE,R1     ;FIRST LOC IN READ BUFFER
1044 033546 012702 177777      MOV    #177777,R2  ;EXPECTED IF NO DATA TRANS.
1045 033552 020102      CMP    R1,R2       ;DID ANY DATA GET TRANSFERRED
1046 033554 001006      BNE    220$        ;BR, IF NO DATA TRANS (GOOD)
1047 033556 005237 002214      INC    FATFLG      ;ERROR COUNT
1048 033562      ERRHRD  ERRNO,T30DTR,EXPREC ;DATA TRANSFERRED ON READ TAPE MARK
1049 033562 104456      TRAP   C$ERHRD
1050 033564 000326      .WORD  214
1051 033566 041070      .WORD  T30DTR
1052 033570 015554      .WORD  EXPREC
1053 033572      220$:  CKLOOP      ;LOOP IF SELECTED
1054 033572 104406      TRAP   C$CLP1
1055 033574 012702 001001      MOV    #1001,R2     ;SET UP RECORD NUMBER EXPECTED (FILE 2)
1056 033600 017701 147312      MOV    @FREE,R1     ;GET INFO FROM BUFFER
1057 033604 020201      CMP    R2,R1       ;ARE THEY EQUAL
1058 033606 001406      BEQ    228$        ;BR, IF EQUAL (OK)
1059 033610 005237 002214      INC    FATFLG      ;ERROR COUNT
1060 033614      ERRHRD  ERRNO,T30PTB,EXPREC ;RECORD POSITION WAS NOT CORRECT
1061 033614 104456      TRAP   C$ERHRD
1062 033616 000327      .WORD  215
1063 033620 037242      .WORD  T30PTB
1064 033622 015554      .WORD  EXPREC
1065 033624      228$:  CKLOOP      ;LOOP IF SELECTED
1066 033624 104406      TRAP   C$CLP1
1067
1068 :*****
1069 :ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1070 :

```

1063				;*****					
1064									
1065	033626	004737	011074	JSR	PC,REWIND			;CALL TAPE REWIND COMMAND	
1066	033632	103411		BCS	230\$;BR, IF NO PROBLEM	
1067	033634	010004		MOV	R0,R4			;SAVE PACKET ADDRESS	
1068	033636	016501	000002	MOV	TSSR(R5),R1			;GET TSSR STATUS	
1069	033642	005237	002214	INC	FATFLG			;ERROR COUNT	
1073	033646			ERRHRD	ERRNO,T3ORWN,PKTSSR			;REWIND NOT ACCEPTED	
	033646	104456							TRAP C\$ERHRD
	033650	000330							.WORD 216
	033652	040240							.WORD T3ORWN
	033654	012126							.WORD PKTSSR
1074	033656			230\$: CKLOOP				;LOOP IF SELECTED	
	033656	104406							TRAP C\$CLP1
1075									
1076				;*****					
1077				:					
1078				;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER					
1079				:					
1080				;*****					
1081									
1082	033660	013701	036530	MOV	T3OBFR+6,R1			;PICK UP XSTO	
1083	033664	010102		MOV	R1,R2			;SET UP EXPECTED	
1084	033666	052702	000002	BIS	#BIT1,R2			;SET BOT BIT IN EXPECTED	
1085	033672	020102		CMP	R1,R2			;DOES EXP = REC'D	
1086	033674	001406		BEQ	240\$;BR, IF EQUAL (OK)	
1087	033676	005237	002214	INC	FATFLG			;ERROR COUNT	
1091	033702			ERRHRD	ERRNO,T3OBOT,EXPREC			;TAPE NOT AT BOT AFTER REWIND	
	033702	104456							TRAP C\$ERHRD
	033704	000331							.WORD 217
	033706	040041							.WORD T3OBOT
	033710	015554							.WORD EXPREC
1092	033712			240\$: CKLOOP				;LOOP IF SELECTED	
	033712	104406							TRAP C\$CLP1
1093	033714	005723		TST	(R3)+			;POINT TO NEXT POSITION	
1094	033716	011301		MOV	(R3),R1			;GET NEXT COMMAND ETC.	
1095	033720	020127	177777	CMP	R1,#177777			;END OF TABLE MARKER	
1096	033724	001402		BEQ	330\$;BR, IF AT END OF TABLE	
1097	033726	000137	033214	JMP	182\$;JUMP TO MORE COMMANDS TO DO	
1098	033732			330\$: CKLOOP				;LOOP IF SELECTED	
	033732	104406							TRAP C\$CLP1
1099	033734			ENDSUB				;<<<<<<<< END SUBTEST >>>>>>>>	
	033734							L10044:	
	033734	104403							TRAP C\$ESUB
1100	033736	023727	002214 000017	CMP	FATFLG,#15.			;IS ERROR COUNT AT 25	
1101	033744	103402		BLO	999\$;BR, IF LESS THAN 25	
1102	033746	004737	017262	JSR	PC,CKDROP			;TRY TO DROP THE UNIT	
1103	033752			999\$:					

Line	Address	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464	Op46
------	---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

PC	OP	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14	OP15	OP16	OP17	OP18	OP19	OP20	OP21	OP22	OP23	OP24	OP25	OP26	OP27	OP28	OP29	OP30	OP31	OP32	OP33	OP34	OP35	OP36	OP37	OP38	OP39	OP40	OP41	OP42	OP43	OP44	OP45	OP46	OP47	OP48	OP49	OP50	OP51	OP52	OP53	OP54	OP55	OP56	OP57	OP58	OP59	OP60	OP61	OP62	OP63	OP64	OP65	OP66	OP67	OP68	OP69	OP70	OP71	OP72	OP73	OP74	OP75	OP76	OP77	OP78	OP79	OP80	OP81	OP82	OP83	OP84	OP85	OP86	OP87	OP88	OP89	OP90	OP91	OP92	OP93	OP94	OP95	OP96	OP97	OP98	OP99	OP100	OP101	OP102	OP103	OP104	OP105	OP106	OP107	OP108	OP109	OP110	OP111	OP112	OP113	OP114	OP115	OP116	OP117	OP118	OP119	OP120	OP121	OP122	OP123	OP124	OP125	OP126	OP127	OP128	OP129	OP130	OP131	OP132	OP133	OP134	OP135	OP136	OP137	OP138	OP139	OP140	OP141	OP142	OP143	OP144	OP145	OP146	OP147	OP148	OP149	OP150	OP151	OP152	OP153	OP154	OP155	OP156	OP157	OP158	OP159	OP160	OP161	OP162	OP163	OP164	OP165	OP166	OP167	OP168	OP169	OP170	OP171	OP172	OP173	OP174	OP175	OP176	OP177	OP178	OP179	OP180	OP181	OP182	OP183	OP184	OP185	OP186	OP187	OP188	OP189	OP190	OP191	OP192	OP193	OP194	OP195	OP196	OP197	OP198	OP199	OP200	OP201	OP202	OP203	OP204	OP205	OP206	OP207	OP208	OP209	OP210	OP211	OP212	OP213	OP214	OP215	OP216	OP217	OP218	OP219	OP220	OP221	OP222	OP223	OP224	OP225	OP226	OP227	OP228	OP229	OP230	OP231	OP232	OP233	OP234	OP235	OP236	OP237	OP238	OP239	OP240	OP241	OP242	OP243	OP244	OP245	OP246	OP247	OP248	OP249	OP250	OP251	OP252	OP253	OP254	OP255	OP256	OP257	OP258	OP259	OP260	OP261	OP262	OP263	OP264	OP265	OP266	OP267	OP268	OP269	OP270	OP271	OP272	OP273	OP274	OP275	OP276	OP277	OP278	OP279	OP280	OP281	OP282	OP283	OP284	OP285	OP286	OP287	OP288	OP289	OP290	OP291	OP292	OP293	OP294	OP295	OP296	OP297	OP298	OP299	OP300	OP301	OP302	OP303	OP304	OP305	OP306	OP307	OP308	OP309	OP310	OP311	OP312	OP313	OP314	OP315	OP316	OP317	OP318	OP319	OP320	OP321	OP322	OP323	OP324	OP325	OP326	OP327	OP328	OP329	OP330	OP331	OP332	OP333	OP334	OP335	OP336	OP337	OP338	OP339	OP340	OP341	OP342	OP343	OP344	OP345	OP346	OP347	OP348	OP349	OP350	OP351	OP352	OP353	OP354	OP355	OP356	OP357	OP358	OP359	OP360	OP361	OP362	OP363	OP364	OP365	OP366	OP367	OP368	OP369	OP370	OP371	OP372	OP373	OP374	OP375	OP376	OP377	OP378	OP379	OP380	OP381	OP382	OP383	OP384	OP385	OP386	OP387	OP388	OP389	OP390	OP391	OP392	OP393	OP394	OP395	OP396	OP397	OP398	OP399	OP400	OP401	OP402	OP403	OP404	OP405	OP406	OP407	OP408	OP409	OP410	OP411	OP412	OP413	OP414	OP415	OP416	OP417	OP418	OP419
----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

1202	034252	013702	036654	MOV	T30FCN,R2	:GET FILE COUNTER
1203	034256	000302		SWAB	R2	:MOVE TO UPPER BYTE
1204	034260	010301		MOV	R3,R1	:GET RECORD COUNTER
1205	034262	060201		ADD	R2,R1	:FILE COUNTER IN UPPER, RECORD # LOW
1206	034264	010177	146626	MOV	R1,@FREE	:MOV TO OUT PUT BUFFER
1207	034270	010465	000000	MOV	R4,TSDB(R5)	:ISSUE COMMAND
1208	034274	004737	016330	JSR	PC,WAITF	:WAIT FOR SSR TO SET
1209	034300	016501	000002	MOV	TSSR(R5),R1	:GET TSSR CONTENTS
1210	034304	012702	000200	MOV	#SSR,R2	:SET UP EXPECTED
1211	034310	020102		CMP	R1,R2	:ARE THEY EQUAL
1212	034312	001406		BEQ	70\$:BR, IF OK
1213	034314	005237	002214	INC	FATFLG	:ERROR COUNT
1217	034320			ERRHRD	ERRNO,T30WDD,PKTSSR	:TSSR INCORRECT AFTER WRITE DATA
	034320	104456				TRAP CSERHRD
	034322	000336				.WORD 222
	034324	037170				.WORD T30WDD
	034326	012126				.WORD PKTSSR
1218	034330			70\$: CKLOOP		:LOOP IF SELECTED
	034330	104406				TRAP CSCLP1
1219	034332	005203		INC	R3	:COUNT THE RECORD COUNTER DOWN
1220	034334	020327	000021	CMP	R3,#21	:AT 20 YET
1221	034340	001331		BNE	65\$:BR, IF NOT AT 20 RECORDS WRITTEN
1222				:*****		
1223				:WRITE TAPE MARK,ACK,CVC=1 COMMAND		
1224				:*****		
1225						
1226						
1227						
1228						
1229	034342	012737	141011 036620	MOV	#141011,T30PK3	:WRITE TAPE MARK,ACK,CVC=1 COMMAND
1230	034350	012704	036620	MOV	#T30PK3,R4	:SET UP R4 WITH PACKET ADDRESS
1231	034354	010465	000000	MOV	R4,TSDB(R5)	:ISSUE COMMAND
1232	034360	004737	016330	JSR	PC,WAITF	:WAIT FOR SSR TO SET
1233	034364	016501	000002	MOV	TSSR(R5),R1	:PICK UP TSSR
1234	034370	012702	000200	MOV	#SSR,R2	:SET UP EXPECTED (SSR ONLY)
1235	034374	020102		CMP	R1,R2	:WAS STATUS GOOD
1236	034376	001406		BEQ	160\$:BR, IF TERMINATION WAS GOOD
1237	034400	005237	002214	INC	FATFLG	:ERROR COUNT
1241	034404			ERRHRD	ERRNO,T30WDC,PKTSSR	:TSSR NOT CORRECT AFTER WRT TAPE M.
	034404	104456				TRAP CSERHRD
	034406	000337				.WORD 223
	034410	040362				.WORD T30WDC
	034412	012126				.WORD PKTSSR
1242	034414			160\$: CKLOOP		:LOOP IF SELECTED
	034414	104406				TRAP CSCLP1
1243	034416	005237	036654	INC	T30FCN	:COUNT THE 'FILE' COUNTER DOWN
1244	034422	023727	036654 000031	CMP	T30FCN,#25.	:WRITE 25 FILES TO TAPE
1245	034430	001273		BNE	64.	:BR, IF NOT AT 25 FILES WRITTEN
1246				:*****		
1247				:WRITE TAPE MARK,ACK,CVC=1 COMMAND		
1248				:*****		
1249						
1250						
1251						
1252						
1253	034432	012737	141011 036620	MOV	#141011,T30PK3	:WRITE TAPE MARK,ACK,CVC=1 COMMAND
1254	034440	012704	036620	MOV	#T30PK3,R4	:SET UP R4 WITH PACKET ADDRESS

1255	034444	010465	000000	MOV	R4,TSDB(R5)	:ISSUE COMMAND	
1256	034450	004737	016330	JSR	PC,WAITF	:WAIT FOR SSR TO SET	
1257	034454	016501	000002	MOV	TSSR(R5),R1	:PICK UP TSSR	
1258	034460	012702	000200	MOV	#SSR,R2	:SET UP EXPECTED (SSR ONLY)	
1259	034464	020102		CMP	R1,R2	:WAS STATUS GOOD	
1260	034466	001406		BEQ	165\$:BR, IF TERMINATION WAS GOOD	
1261	034470	005237	002214	INC	FATFLG	:ERROR COUNT	
1265	034474			ERRHRD	ERRNO,T30WDC,PKTSSR	:TSSR NOT CORRECT AFTER WRT TAPE M.	
	034474	104456				TRAP	C\$ERHRD
	034476	000340				.WORD	224
	034500	040362				.WORD	T30WDC
	034502	012126				.WORD	PKTSSR
1266	034504			165\$: CKLOOP		:LOOP IF SELECTED	
	034504	104406				TRAP	C\$CLP1
1267							
1268							
1269							
1270							
1271							
1272							
1273							
1274	034506	004737	011074	JSR	PC,REWIND	:CALL TAPE REWIND COMMAND	
1275	034512	103411		BCS	170\$:BR, IF NO PROBLEM	
1276	034514	010004		MOV	R0,R4	:GET PACKET ADDRESS	
1277	034516	016501	000002	MOV	TSSR(R5),R1	:GET STATUS REGISTER	
1278	034522	005237	002214	INC	FATFLG	:ERROR COUNT	
1282	034526			ERRHRD	ERRNO,T30RWN,PKTSSR	:REWIND NOT ACCEPTED	
	034526	104456				TRAP	C\$ERHRD
	034530	000341				.WORD	225
	034532	040240				.WORD	T30RWN
	034534	012126				.WORD	PKTSSR
1283	034536			170\$: CKLOOP		:LOOP IF SELECTED	
	034536	104406				TRAP	C\$CLP1
1284							
1285							
1286							
1287							
1288							
1289							
1290							
1291	034540	013701	036530	MOV	T30BFR+6,R1	:PICK UP XSTO	
1292	034544	010102		MOV	R1,R2	:SET UP EXPECTED	
1293	034546	052702	000002	BIS	#BIT1,R2	:SET BOT BIT IN EXPECTED	
1294	034552	020102		CMP	R1,R2	:DOES EXP = REC'D	
1295	034554	001406		BEQ	180\$:BR, IF EQUAL (OK)	
1296	034556	005237	002214	INC	FATFLG	:ERROR COUNT	
1300	034562			ERRHRD	ERRNO,T30BOT,EXPREC	:TAPE NOT AT BOT AFTER REWIND	
	034562	104456				TRAP	C\$ERHRD
	034564	000342				.WORD	226
	034566	040041				.WORD	T30BOT
	034570	015554				.WORD	EXPREC
1301	034572			180\$: CKLOOP		:LOOP IF SELECTED	
	034572	104406				TRAP	C\$CLP1
1302	034574	012737	000002	MOV	#2,T30FCN	:SET TO NUMBER OF SKIP 'FILES'	
1303	034602	012703	036636	MOV	#T30IMV,R3	:SET UP POINTER TO COMMAND TABLE	
1304	034606	013737	002174	MOV	UNITN,T30DSW	:SET UP UNIT NUMBER	
1305	034614	011337	036516	MOV	(R3),T30ETM	:GET NEXT COMMAND	
				182\$:			

```

1306 034620 012704 036500      MOV      #T30PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
1307
1308      ;*****
1309      ;ISSUE WRITE CHARACTERISTICS COMMAND
1310      ;*****
1311
1312
1313
1314 034624 004737 010742      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
1315 034630 103407      BCS      188$      ;BR, IF COMMAND ISSUED OK
1316 034632 005237 002214      INC      FATFLG      ;ERROR COUNT
1320 034636 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1321 034640      ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      034640 104456      TRAP      C$ERHRD
      034642 000343      .WORD    227
      034644 005052      .WORD    WRTMSG
      034646 012114      .WORD    SFIMSG
1322 034650      188$: CKLOOP      ;LOOP IF SELECTED
      034650 104406      TRAP      C$CLP1
1323
1324      ;*****
1325      ;SKIP TAPE MARK,ACK,CVC=1 COMMAND
1326      ;*****
1327
1328
1329
1330 034652 012737 141010 036620      MOV      #141010,T30PK3      ;SKIP TAPE MARK,ACK,CVC=1 COMMAND
1331 034660 013737 036654 036622      MOV      T30FCN,T30RB      ;SET UP NUMBER TO SKIP
1332 034666 012704 036620      MOV      #T30PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
1333 034672 010465 000000      189$: MOV      R4,TSDR(R5)      ;ISSUE COMMAND
1334 034676 012737 176750 036656      MOV      #65000.,T30DLY      ;SET UP DELAY COUNTER
1335 034704 004737 016330      190$: JSR      PC,WAITF      ;WAIT FOR SSR TO SET
1336 034710 016501 000002      MOV      TSSR(R5),R1      ;PICK UP TSSR
1337 034714 032701 000200      BIT      #SSR,R1      ;IS SSR SET YET
1338 034720 001017      BNE      191$      ;BR, IF SSR IS SET
1339 034722      DELAY      250      ;CALL DELAY ROUTINE
      034722 012727 000250      MOV      #250,(PC)+
      034726 000000      .WORD    0
      034730 013727 002116      MOV      L$DLY,(PC)+
      034734 000000      .WORD    0
      034736 005367 177772      DEC      -6(PC)
      034742 001375      BNE      -4
      034744 005367 177756      DEC      -22(PC)
      034750 001367      BNE      -20
1340 034752 005337 036656      DEC      T30DLY      ;BUMP DELAY ROUTINE
1341 034756 001352      BNE      190$      ;BR, IF MORE DELAY TO GO
1342 034760 012702 000200      191$: MOV      #SSR,R2      ;SET UP EXPECTED (SSR ONLY)
1343 034764 020102      CMP      R1,R2      ;WAS STATUS GOOD
1344 034766 001406      BEQ      192$      ;BR, IF TERMINATION WAS GOOD
1345 034770 005237 002214      INC      FATFLG      ;ERROR COUNT
1349 034774      ERRHRD ERRNO,T30SKM,PKTSSR ;TSSR NOT CORRECT AFTER SKIP TAPE M.
      034774 104456      TRAP      C$ERHRD
      034776 000344      .WORD    228
      035000 037114      .WORD    T30SKM
      035002 012126      .WORD    PKTSSR
1350 035004      192$: CKLOOP      ;LOOP IF SELECTED
      035004 104406      TRAP      C$CLP1

```

```

1351
1352
1353
1354
1355
1356
1357
1358 035006 013701 036530      MOV      T30BFR+6,R1      ;PICK UP XST0
1359 035012 010102      MOV      R1,R2      ;SET UP EXPECTED
1360 035014 052702 100000      BIS      #BIT15,R2      ;SET TMK BIT IN EXPECTED
1361 035020 020102      CMP      R1,R2      ;DOES EXP = REC'D
1362 035022 001406      BEQ      195$      ;BR, IF EQUAL (OK)
1363 035024 005237 002214      INC      FATFLG      ;ERROR COUNT
1367 035030      ERRHRD  ERRNO,T30TMK,EXPREC ;TMK NOT SET AFTER WRT TAPE MARK
                                TRAP      C$ERHRD
                                .WORD    229
                                .WORD    T30TMK
                                .WORD    EXPREC
                                TRAP      C$CLP1
1368 035040      195$: CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
1369 035042 012700 177777      MOV      #177777,R0      ;VALUE TO WRITTEN TO MEMORY
1370 035046 004737 017502      JSR      PC,FILLMEM      ;FILL MEM WITH ALL ONES
1371 035052 013737 003116 036622 MOV      FREE,T30RB      ;STARTING READ BUFFER ADDRESS
1372
1373
1374
1375
1376
1377
1378
1379 035060 012737 140001 036620      MOV      #140001,T30PK3      ;READ FORWARD,ACK,CVC=1 COMMAND
1380 035066 012704 036620      MOV      #T30PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
1381 035072 012737 000024 036626      MOV      #20,T30SZ      ;SET UP RECORD SIZE IN PACKET
1382 035100 010465 000000      MOV      R4,T30SDB(R5)      ;ISSUE COMMAND
1383 035104 004737 016330      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
1384 035110 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
1385 035114 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
1386 035120 020102      CMP      R1,R2      ;ARE THEY EQUAL
1387 035122 001406      BEQ      200$      ;BR, IF OK
1388 035124 005237 002214      INC      FATFLG      ;ERROR COUNT
1392 035130      ERRHRD  ERRNO,T30RDF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    230
                                .WORD    T30RDF
                                .WORD    PKTSSR
                                TRAP      C$CLP1
1393 035140      200$: CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
1394 035142 017701 145750      MOV      @FREE,R1      ;FIRST LOC IN READ BUFFER
1395 035146 012702 177777      MOV      #177777,R2      ;EXPECTED IF NO DATA TRANS.
1396 035152 020102      CMP      R1,R2      ;DID ANY DATA GET TRANSFERRED
1397 035154 001006      BNE      220$      ;BR, IF NO DATA TRANS (GOOD)
1398 035156 005237 002214      INC      FATFLG      ;ERROR COUNT
1402 035162      ERRHRD  ERRNO,T30DTR,EXPREC ;DATA TRANSFERRED ON READ TAPE MARK
                                TRAP      C$ERHRD
                                .WORD    231
                                .WORD    T30DTR
                                .WORD    EXPREC
1403 035162 104456
1404 035164 000347
1405 035166 041070
1406 035170 015554

```

```

1403 035172      220$:  CKLOOP                      ;LOOP IF SELECTED
      035172 104406                                TRAP  C$CLP1
1404 035174 013702 036654      MOV  T30FCN,R2      ;GET NUMBER OF SKIPS
1405 035200 005202      INC  R2                    ;SET TO CORRECT FILE VALUE
1406 035202 000302      SWAB R2                     ;SWAP BYTE HALVES
1407 035204 052702 000001      BIS  #BIT0,R2        ;SET FOR RECORD #1
1408 035210 017701 145702      MOV  @FREE,R1        ;GET INFO FROM BUFFER
1409 035214 020201      CMP  R2,R1                  ;ARE THEY EQUAL
1410 035216 001406      BEQ  228$                    ;BR, IF EQUAL (OK)
1411 035220 005237 002214      INC  FATFLG           ;ERROR COUNT
1415 035224      ERRHRD  ERRNO,T30PTB,EXPREC      ;RECORD POSITION WAS NOT CORRECT
      035224 104456                                TRAP  C$ERHRD
      035226 000350                                .WORD  232
      035230 037242                                .WORD  T30PTB
      035232 015554                                .WORD  EXPREC

1416 035234      228$:  CKLOOP                      ;LOOP IF SELECTED
      035234 104406                                TRAP  C$CLP1

1417
1418      ;*****
1419      ;
1420      ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1421      ;
1422      ;*****
1423
1424 035236 004737 011074      JSR  PC,REWIND        ;CALL TAPE REWIND COMMAND
1425 035242 103411      BCS  230$                    ;BR, IF NO PROBLEM
1426 035244 010004      MOV  R0,R4                    ;SAVE PACKET ADDRESS
1427 035246 016501 000002      MOV  TSSR(R5),R1      ;GET TSSR STATUS
1428 035252 005237 002214      INC  FATFLG           ;ERROR COUNT
1432 035256      ERRHRD  ERRNO,T30RWN,PKTSSR      ;REWIND NOT ACCEPTED
      035256 104456                                TRAP  C$ERHRD
      035260 000351                                .WORD  233
      035262 040240                                .WORD  T30RWN
      035264 012126                                .WORD  PKTSSR

1433 035266      230$:  CKLOOP                      ;LOOP IF SELECTED
      035266 104406                                TRAP  C$CLP1

1434
1435      ;*****
1436      ;
1437      ;GET EXTENDED STATUS REGISTER ZERO (XST0) FROM MESSAGE BUFFER
1438      ;
1439      ;*****
1440
1441 035270 013701 036530      MOV  T30BFR+6,R1      ;PICK UP XST0
1442 035274 010102      MOV  R1,R2                    ;SET UP EXPECTED
1443 035276 052702 000002      BIS  #BIT1,R2        ;SET BOT BIT IN EXPECTED
1444 035302 020102      CMP  R1,R2                    ;DOES EXP = REC'D
1445 035304 001406      BEQ  240$                    ;BR, IF EQUAL (OK)
1446 035306 005237 002214      INC  FATFLG           ;ERROR COUNT
1450 035312      ERRHRD  ERRNO,T30BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      035312 104456                                TRAP  C$ERHRD
      035314 000352                                .WORD  234
      035316 040041                                .WORD  T30BOT
      035320 015554                                .WORD  EXPREC

1451 035322      240$:  CKLOOP                      ;LOOP IF SELECTED
      035322 104406                                TRAP  C$CLP1
1452 035324 005723      TST  (R3)+                  ;POINT TO NEXT POSITION
  
```

330S:

999\$:

FATFLG.#15.

```
;GET NEXT COMMAND ETC.  
;END OF TABLE MARKER  
;BR, IF AT END OF TABLE  
;GET NUMBER OF SKIPS  
;CLEAR THE CARRY BIT  
;PUSH OVER ONE POSITION  
;PUT BACK IN COUNTER  
;JUMP TO MORE COMMANDS TO DO  
;LOOP IF SELECTED  
  
                                TRAP          C$CLP1  
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<  
                                L10045:  
  
                                TRAP          C$ESUB  
      ;IS ERROR COUNT AT 25  
      ;BR, IF LESS THAN 25  
      ;TRY TO DROP THE UNIT
```

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 25-MAY-82 08:43 PAGE L 12
TEST 2: SKIP TAPE MARKS 91

1469

SEQ 0154

1
1

1471							:*
1472							:
1473							:TEST 2, SUBTEST 3
1474							:
1475							:
1476							:VERIFIES THAT A SKIP TAPE MARKS REVERSE COMMAND
1477							:ISSUED WHILE THE TAPE IS POSITIONED AT BOT CAUSES
1478							:FUNCTION REJECT TERMINATION, WITH THE NON-EXECUTABLE
1479							:FUNCTION (NEF) ERROR BIT SET.
1480							:
1481							:
1482							:
1483							:
1484							:
1485							:
1486	035376			BGNSUB		;>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>	
	035376					T2.3:	
	035376	104402				TRAP C\$BSUB	
1487	035400	004737	041252	JSR PC,T3OREST		:SET COMMAND PACKET	
1488	035404	005037	036654	CLR T30FCN		:CLEAR FILE COUNTER	
1489	035410	004737	041344	JSR PC,T3ORT2		:SET UP OTHER COMMAND PACKET	
1490	035414	004737	041406	JSR PC,T3ORT3		:SET UP OTHER COMMAND PACKET	
1491	035420	012737	176750	MOV #65000.,T3ODLY		:SET UP DELAY COUNTER	
1492	035426	004737	016054	JSR PC,SOFINIT		:DO INITIALIZE ON CONTROLLER	
1493	035432	103426		BCS 20\$:BR IF INIT WAS OK	
1494	035434			DELAY 250		:DELAY ROUTINE CALL	
	035434	012727	000250			MOV #250,(PC)+	
	035440	000000				.WORD 0	
	035442	013727	002116			MOV LSDLY,(PC)+	
	035446	000000				.WORD 0	
	035450	005367	177772			DEC -6(PC)	
	035454	001375				BNE .-4	
	035456	005367	177756			DEC -22(PC)	
	035462	001367				BNE .-20	
1495	035464	005337	036656	DEC T3ODLY		:BUMP COUNTER	
1496	035470	001356		BNE 10\$:BR, IF MORE COUNTING TO DO	
1497	035472	005237	002214	INC FATFLG		:ERROR COUNT	
1501	035476	010001		MOV R0,R1		:CONTENTS OF TSSR REGISTER	
1502	035500			ERRDF ERRNO,SFIERR,SFIMSG		:FATAL ERROR TSSR WAS NOT OK	
	035500	104455				TRAP C\$ERDF	
	035502	000353				.WORD 235	
	035504	003646				.WORD SFIERR	
	035506	012114				.WORD SFIMSG	
1503	035510						
1504	035510	013737	002174	MOV UNITN,T3ODSW		:SET UP UNIT NUMBER	
1505	035516	012704	036500	MOV #T3OPACKET,R4		:SUBROUTINE NEEDS PACKET ADDRESS	
1506							
1507							
1508							
1509							
1510							
1511							
1512							
1513	035522	004737	010742	JSR PC,WRTCHR		:ISSUE WRITE CHARACTERISTICS	
1514	035526	103407		BCS 23\$:BR, IF COMMAND ISSUED OK	
1515	035530	005237	002214	INC FATFLG		:ERROR COUNT	
1519	035534	010001		MOV R0,R1		:SAVE CONTENTS OF TSSR	


```

1520 035536          ERRHRD  ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTICS FAILED
      035536 104456          TRAP          C$ERHRD
      035540 000354          .WORD          236
      035542 005052          .WORD          WRTMSG
      035544 012114          .WORD          SFIMSG
1521 035546          23$:    CKLOOP                  ;LOOP IF SELECTED
      035546 104406          TRAP          C$CLP1
1522
1523          ;*****
1524          ;
1525          ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1526          ;
1527          ;*****
1528
1529 035550 004737 011074      JSR          PC,REWIND      ;CALL TAPE REWIND COMMAND
1530 035554 103411          BCS          30$            ;BR, IF NO PROBLEM
1531 035556 010004          MOV          R0,R4            ;GET PACKET ADDRESS
1532 035560 016501 000002      MOV          TSSR(R5),R1    ;GET STATUS REGISTER
1533 035564 005237 002214      INC          FATFLG          ;ERROR COUNT
1537 035570          ERRHRD  ERRNO,T3ORWN,PKTSSR      ;REWIND NOT ACCEPTED
      035570 104456          TRAP          C$ERHRD
      035572 000355          .WORD          237
      035574 040240          .WORD          T3ORWN
      035576 012126          .WORD          PKTSSR
1538 035600          30$:    CKLOOP                  ;LOOP IF SELECTED
      035600 104406          TRAP          C$CLP1
1539
1540          ;*****
1541          ;
1542          ;GET EXTENDED STATUS REGISTER ZERO (XST0) FROM MESSAGE BUFFER
1543          ;
1544          ;*****
1545
1546 035602 013701 036530      MOV          T30BFR+6,R1    ;PICK UP XST0
1547 035606 010102          MOV          R1,R2            ;SET UP EXPECTED
1548 035610 052702 000002      BIS          #BIT1,R2      ;SET BOT BIT IN EXPECTED
1549 035614 020102          CMP          R1,R2            ;DOES EXP = REC'D
1550 035616 001406          BEQ          40$            ;BR, IF EQUAL (OK)
1551 035620 005237 002214      INC          FATFLG          ;ERROR COUNT
1555 035624          ERRHRD  ERRNO,T30BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      035624 104456          TRAP          C$ERHRD
      035626 000356          .WORD          238
      035630 040041          .WORD          T30BOT
      035632 015554          .WORD          EXPREC
1556 035634          40$:    CKLOOP                  ;LOOP IF SELECTED
      035634 104406          TRAP          C$CLP1
1557 035636 012737 000001 036622  MOV          #1,T30WB    ;SET # OF TM TO SKIP
1558
1559          ;*****
1560          ;
1561          ;SKIP TAPE MARK REVERSE,ACK,CVC=1 COMMAND
1562          ;
1563          ;*****
1564
1565 035644 012737 141410 036620  MOV          #141410,T30PK3  ;SKIP TAPE MARK REVERSE,ACK,CVC=1 CMD
1566 035652 012704 036620      MOV          #T30PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
1567 035656 010465 000000      MOV          R4,TSDB(R5)    ;ISSUE COMMAND
  
```

Line	Address	Hex	Label	Instruction	Comment	Trap	Value
1568	035662	004737	016330	JSR	PC, WAITF		
1569	035666	016501	000002	MOV	TSSR(R5), R1		
1570	035672	012702	100206	MOV	#SSR!SC!BIT1!BIT2, R2		
1571	035676	020102		CMP	R1, R2		
1572	035700	001406		BEQ	70\$		
1573	035702	005237	002214	INC	FATFLG		
1577	035706			ERRHRD	ERRNO, T30IBT, PKTSSR		
	035706	104456					
	035710	000357				TRAP	C\$ERHRD
	035712	037031				.WORD	239
	035714	012126				.WORD	T30IBT
1578	035716			70\$: CKLOOP			
	035716	104406					
1579							
1580							
1581							
1582							
1583							
1584							
1585							
1586	035720	013701	036530	MOV	T30BFR+6, R1		
1587	035724	010102		MOV	R1, R2		
1588	035726	052702	002000	BIS	#BIT10, R2		
1589	035732	020102		CMP	R1, R2		
1590	035734	001406		BEQ	180\$		
1591	035736	005237	002214	INC	FATFLG		
1595	035742			ERRHRD	ERRNO, T30NEF, EXPREC		
	035742	104456					
	035744	000360				TRAP	C\$ERHRD
	035746	040576				.WORD	240
	035750	015554				.WORD	T30NEF
1596	035752			180\$: CKLOOP			
	035752	104406					
1597	035754			ENDSUB			
	035754						
	035754	104403					
1598	035756	023727	002214	CMP	FATFLG, #15.		
1599	035764	103402	000017	BLO	999\$		
1600	035766	004737	017262	JSR	PC, CKDROP		
1601	035772			999\$:			

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 25-MAY-82 08:43 C 13
TEST 2: SKIP TAPE MARKS PAGE 93

1603

SEQ 0158

1
1

Line	Address	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464	Op46
------	---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

```

1654 036130 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
1655 036132          ERRHRD  ERRNO,WRTMSG,SFIMSG    ;WRITE CHARACTERISTICS
                                TRAP    C$ERHRD
                                .WORD   242
                                .WORD   WRTMSG
                                .WORD   SFIMSG
1656 036142          23$:   CKLOOP                  ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
1657
1658
1659
1660
1661
1662
1663
1664 036144 004737 011074    JSR     PC,REWIND        ;CALL TAPE REWIND COMMAND
1665 036150 103411          BCS     30$             ;BR, IF NO PROBLEM
1666 036152 010004          MOV     R0,R4           ;GET PACKET ADDRESS
1667 036154 016501 000002    MOV     TSSR(R5),R1      ;GET STATUS REGISTER
1668 036160 005237 002214    INC     FATFLG         ;ERROR COUNT
1672 036164          ERRHRD  ERRNO,T3ORWN,PKTSSR    ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD   243
                                .WORD   T3ORWN
                                .WORD   PKTSSR
1673 036174          30$:   CKLOOP                  ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
1674
1675
1676
1677
1678
1679
1680
1681 036176 013701 036530    MOV     T30BFR+6,R1      ;PICK UP XST0
1682 036202 010102          MOV     R1,R2           ;SET UP EXPECTED
1683 036204 052702 000002    BIS     #BIT1,R2       ;SET BOT BIT IN EXPECTED
1684 036210 020102          CMP     R1,R2           ;DOES EXP = REC'D
1685 036212 001406          BEQ     40$             ;BR, IF EQUAL (OK)
1686 036214 005237 002214    INC     FATFLG         ;ERROR COUNT
1690 036220          ERRHRD  ERRNO,T30BOT,EXPREC    ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD   244
                                .WORD   T30BOT
                                .WORD   EXPREC
1691 036230          40$:   CKLOOP                  ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
1692 036232 013737 003116 036622  MOV     FREE,T30WB      ;SET UP GOOD WRITE BUFFER
1693 036240 012737 000400 036626  MOV     #256.,T30SZ    ;SET UP SIZE
1694
1695
1696
1697
1698
1699
1700
1701 036246 012737 140005 036620  MOV     #140005,T30PK3    ;WRITE DATA,ACK,CVC=1 COMMAND
  
```

1702	036254	012704	036620	MOV	#T30PK3,R4	;SET UP R4 WITH PACKET ADDRESS	
1703	036260	010465	000000	MOV	R4,TSDB(R5)	;ISSUE COMMAND	
1704	036264	004737	016330	JSR	PC,WAITF	;WAIT FOR SSR TO SET	
1705	036270	016501	000002	MOV	TSSR(R5),R1	;GET TSSR CONTENTS	
1706	036274	012702	000200	MOV	#SSR,R2	;SET UP EXPECTED	
1707	036300	020102		CMP	R1,R2	;ARE THEY EQUAL	
1708	036302	001406		BEQ	70\$;BR, IF OK	
1709	036304	005237	002214	INC	FATFLG	;ERROR COUNT	
1713	036310			ERRHRD	ERRNO,T30WDD,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA	
	036310	104456				TRAP	C\$ERHRD
	036312	000365				.WORD	245
	036314	037170				.WORD	T30WDD
	036316	012126				.WORD	PKTSSR
1714	036320			70\$: CKLOOP		;LOOP IF SELECTED	
	036320	104406				TRAP	C\$CLP1
1715							
1716							
1717							
1718							
1719							
1720							
1721							
1722	036322	012737	000001	MOV	#1,T30WB	;# OF TM TO SKIP	
1723	036330	012737	141410	MOV	#141410,T30PK3	;SKIP TAPE MARK REVERSE,ACK,CVC=1 CMD	
1724	036336	012704	036620	MOV	#T30PK3,R4	;SET UP R4 WITH PACKET ADDRESS	
1725	036342	010465	000000	MOV	R4,TSDB(R5)	;ISSUE COMMAND	
1726	036346	004737	016330	JSR	PC,WAITF	;WAIT FOR SSR TO SET	
1727	036352	016501	000002	MOV	TSSR(R5),R1	;PICK UP TSSR	
1728	036356	012702	100204	MOV	#SSR!BIT2!SC,R2	;SET UP EXPECTED (SSR AND SC ONLY)	
1729	036362	020102		CMP	R1,R2	;WAS STATUS GOOD	
1730	036364	001406		BEQ	160\$;BR, IF TERMINATION WAS GOOD	
1731	036366	005237	002214	INC	FATFLG	;ERROR COUNT	
1735	036372			ERRHRD	ERRNO,T30IBU,PKTSSR	;TSSR NOT CORRECT AFTER WRT TAPE M.	
	036372	104456				TRAP	C\$ERHRD
	036374	000366				.WORD	246
	036376	036660				.WORD	T30IBU
	036400	012126				.WORD	PKTSSR
1736	036402			160\$: CKLOOP		;LOOP IF SELECTED	
	036402	104406				TRAP	C\$CLP1
1737							
1738							
1739							
1740							
1741							
1742							
1743							
1744	036404	013701	036536	MOV	T30BFR+14,R1	;PICK UP XST3	
1745	036410	010102		MOV	R1,R2	;SET UP EXPECTED	
1746	036412	052702	000001	BIS	#BIT0,R2	;SET RIB BIT IN EXPECTED	
1747	036416	020102		CMP	R1,R2	;DOES EXP = REC'D	
1748	036420	001406		BEQ	170\$;BR, IF EQUAL (OK)	
1749	036422	005237	002214	INC	FATFLG	;ERROR COUNT	
1753	036426			ERRHRD	ERRNO,T30RIB,EXPREC	;TAPE NOT AT RIB	
	036426	104456				TRAP	C\$ERHRD
	036430	000367				.WORD	247
	036432	036745				.WORD	T30RIB
	036434	015554				.WORD	EXPREC

1761
1762
1763
1764

:SUBTEST END

1765 036456 004737 016536
1766 036462 103002
1767 036464 000137 032360
1768 036470
036470 104432
036472 002736

JSR PC,TSTLOOP
BCC 400\$
JMP T30LOOP
400\$: EXIT TST

:DO WE NEED TO ITERATE TEST
:BR, IF NO LOOP REQUIRED
:EXECUTE AGAIN
:ALL DONE THIS TEST

TRAP C\$EXIT
.WORD L10043-

1770		;	+			
1771		;	LOCAL STORAGE FOR THIS TEST			
1772		;	-			
1774						
1776	036500					
1777	036500	100004	T30PACKET:			
1778	036502	036510				
1779	036504	000000				
1780	036506	000012				
1781	036510		T30DATA:			
1782	036510	036522				
1783	036512	000000				
1784	036514	000024				
1785	036516	000000	T30ETM:			
1786	036520	000000	T30DSW:			
1787	036522		T30BFR:			
1788						
1789			;			
1790			WRITE SUBSYSTEM MEMORY COMMAND PACKET			
1792		036610	;			
1794	036610					
1795	036610	100006	T30PK2:			
1796	036612	036630				
1797	036614	000000				
1798	036616	000006				
1799						
1803	036620		T30PK3:			
1804	036620	100205				
1805	036622		T30RB:			
1806	036622	003116	T30WB:			
1807	036624	000000				
1808	036626	000000	T30SZ:			
1809						
1810						
1811						
1812						
1813	036630		T30BF2:			
1814	036630	010	T30BS0:			
1815	036631	200	T30BS1:			
1816	036632	000000	T30S2:			
1817	036634	000000	T30S3:			
1818						
1819						
1820						
1821						
1822						
1823	036636					
1824	036636		T30IMV:			
1825	036636	000000	T30RN:			
1826	036640	000100				
1827	036642	000200				
1828	036644	000300				
1829	036646	177777				
1830						
1831						
1832	036650	000000	T30CNT:			
1833	036652	000000	T30CNU:			

;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
 ;SKIP TAPE MARK CONTROL
 ;SELECT DRIVE 0
 ;MESSAGE BUFFER

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

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

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

;NEITHER EWB NOR ESS
 ;EWB SET
 ;ESS SET
 ;BOTH EWB AND ESS SET
 ;END OF DATA

;TAPE TIMER COUNTER STORAGE AREA
 ;TAPE TIMER COUNTER STORAGE AREA

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 25-MAY-82 08:43 J 13
TEST 2: SKIP TAPE MARKS PAGE 96-1

1834 036654 000000
1835 036656 000000

T30FCN: .WORD 0
T30DLY: .WORD 0

;FILE NUMBER COUNTER
;DELAY COUNTER STORAGE

SEQ 0165

1837
 1838
 1839
 1840
 1841
 1842

:+
 :LOCAL TEXT MESSAGES FOR TEST
 :-

1843	036660	124	123	123	T30IBU: .ASCIIZ	'TSSR Incorrect After SKIP TAPE MARK REVERSE Into BOT'
1844	036745	122	111	102	T30RIB: .ASCIIZ	'RIB Bit (XST3) Failed To Set After Reverse Into BOT'
1845	037031	124	123	123	T30IBT: .ASCIIZ	'TSSR Incorrect After SKIP TAPE MARK REVERSE At BOT'
1846	037114	124	123	123	T30SKM: .ASCIIZ	'TSSR Incorrect After SKIP TAPE MARK Command'
1847	037170	124	123	123	T30WDD: .ASCIIZ	'TSSR Not Correct After WRITE DATA Command'
1848	037242	124	141	160	T30PTB: .ASCIIZ	'Tape Not Positioned On Correct Record After READ REVERSE'
1849	037333	124	141	160	T30TPB: .ASCIIZ	'Tape Not Positioned On Second File First Record'
1850	037413	124	123	123	T30RDF: .ASCIIZ	'TSSR Incorrect After READ FORWARD Into 'File''
1851	037471	124	123	123	T30RDG: .ASCIIZ	'TSSR Incorrect After SPACE Command Into TAPE MARK'
1852	037553	124	123	123	T30WDF: .ASCIIZ	'TSSR Not Correct After Illegal Mode Bits Set'
1853	037630	111	154	154	T30LOQ: .ASCIIZ	'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
1854	037711	127	122	111	T30SSR: .ASCIIZ	'WRITE MISCELLANEOUS Command Not Accepted'
1855	037762	124	123	123	T30WDE: .ASCIIZ	'TSSR Not Correct After SKIP TAPE MARKS, At BOT'
1856	040041	124	141	160	T30BOT: .ASCIIZ	'Tape Not At BOT After REWIND Command'
1857	040106	124	123	123	T30TM: .ASCIIZ	'TSSR Not Correct After SPACE FORWARD Command'
1858	040163	124	123	123	T30TM2: .ASCIIZ	'TSSR Not Correct After SPACE REVERSE Command'
1859	040240	122	145	167	T30RWN: .ASCIIZ	'Rewind (POSITION) Command Not Accepted'
1860	040307	104	162	151	T30OFL: .ASCIIZ	'Drive 7 Select Failed To Set 'OFL' In 'SSR'
1861	040362	124	123	123	T30WDC: .ASCIIZ	'TSSR Not Correct After WRITE TAPE MARK Command'
1862	040441	103	126	103	T30VCK: .ASCIIZ	'CVC Set, Didn't Reset VCK In Message Buffer'
1863	040514	124	115	113	T30TMK: .ASCIIZ	'TMK Not Set After WRITE TAPE MARK (RETRY) Command'
1864	040576	123	113	111	T30NEF: .ASCIIZ	'SKIP TAPE MARKS, At BOT, Failed To Set NEF Bit'
1865	040655	124	115	113	T30RRM: .ASCIIZ	'TMK Not Set After READ REVERSE Into TAPE MARK'
1866	040733	124	115	113	T30RRN: .ASCIIZ	'TMK Not Set After SPACE REVERSE Into TAPE MARK'
1867	041012	124	115	113	T30RRP: .ASCIIZ	'TMK Not Set After READ FORWARD Into TAPE MARK'
1868	041070	116	117	040	T30DTR: .ASCIIZ	'NO Data Transferred On READ FORWARD'
1869	041134	104	141	164	T30DTA: .ASCIIZ	'Data Compare Error, Data Read From Tape Not Equal To Written'
1870	041231	123	153	151	TST30ID: .ASCIIZ	'Skip Tape Marks'

.EVEN

1871
 1872
 1873
 1874
 1875
 1876
 1877
 1878

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

1879 041252
 1880 041252
 1881 041256 012701 036500
 1882 041262 012721 100004
 1883 041266 012721 036510
 1884 041272 005021
 1885 041274 012721 000012
 1886 041300 012721 036522
 1887 041304 005021
 1888 041306 012721 000024
 1889 041312 005021
 1890 041314 012711 000000
 1891 041320 012702 000030
 1892 041324 012762 177777 036522 64\$:
 1893 041332 005742

T30REST:

SAVREG

MOV #T30PACKET,R1
 MOV #100004,(R1)+
 MOV #T30DATA,(R1)+
 CLR (R1)+
 MOV #10,(R1)+
 MOV #T30BFR,(R1)+
 CLR (R1)+
 MOV #20,(R1)+
 CLR (R1)+
 MOV #0,(R1)
 MOV #24,R2
 MOV #177777,T30BFR(R2)
 TST -(R2)

;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
 ;LENGTH OF MESSAGE BUFFER
 ;SELECT DRIVE ZERO
 ;NUMBER OF LOCATIONS TO BE CLEARED
 ;ALL ONES TO MESSAGE BUFFER
 ;NEXT LOCATION

1894 041334 022702 000000
 1895 041340 001371
 1896 041342 000207
 1897
 1898
 1899 041344
 1900 041344
 1901 041350 012701 036610
 1902 041354 012721 100006
 1903 041360 012721 036630
 1904 041364 005021
 1905 041366 012721 000006
 1906 041372 005021
 1907 041374 012701 036630
 1908 041400 005021
 1909 041402 005011
 1910 041404 000207
 1911 041406
 1912 041406
 1913 041412 012701 036620
 1914 041416 005021
 1915 041420 005021
 1916 041422 005021
 1917 041424 005011
 1918 041426 000207
 1919 041430
 041430
 041430 104401

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

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

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

L10043: TRAP C\$ETST

```
1921                                     .SBTTL TEST 3: NO-OP ('CLEAN TAPE') AND INITIALIZE
1922                                     :+
1923                                     :
1924                                     :THIS TEST VERIFIES PROPER OPERATION OF THE NO-OP ('CLEAN TAPE') AND INITIALIZE
1925                                     :COMMAND (SPACE REVERSE, ERASE, WRITE DATA)
1926                                     :
1927                                     :
1928                                     :THE TEST CONSISTS OF THE FOLLOWING 2 SUBTESTS
1929                                     :
1930                                     :
1931                                     :
1932                                     :-
1933 041432                               BGNTST
1934 041432                               T3::
1939 041440 012737 006354 002172          MOV    #EPRT1,EPRTSW          ;PRIMARY ERROR MESSAGE
1940 041444 004737 016570                  MOV    #TST31ID,R0          ;ASCII MESSAGE TO IDENTIFY TEST
1941 041450 012737 000005 002210          JSR     PC,TSTSETUP          ;DO INITIAL TEST SETUP
1942 041456 005037 043336                  MOV    #5,LOOPCNT          ;PERFORM 5 ITERATIONS
1943                                     CLR     T31CNT          ;CLEAR TAPE RECORD COUNTER
1944                                     :
1945                                     :-
1946 041462                               T31LOOP:
```

;
;
;
;TEST 3, SUBTEST 1

VERIFIES THAT THE NO-OP COMMAND (CORRESPONDS TO THE CLEAN TAPE COMMAND) TERMINATES PROPERLY (NORMAL TERMINATION), STORES PROPER STATUS IN THE MESSAGE BUFFER (LIKE THE GET STATUS COMMAND), AND INDEED DOES NOT MOVE TAPE. THE TAPE IS FIRST REWOUND AND WRITTEN WITH SEQUENCED TEST RECORDS. IT IS THEN REWOUND AGAIN AND THE NO-OP COMMAND IS ISSUED. IT IS VERIFIED THAT THE TAPE IS STILL AT BOT AND THAT PROPER STATUS IS STORED. THE FIRST RECORD ON TAPE IS READ AND VERIFIED (TO CHECK TAPE POSITION AND VERIFY THAT DATA WAS NOT CHANGED), THEN THE NO-OP COMMAND IS ISSUED AGAIN AND STATUS AND POSITION VERIFIED.

1968				BGNSUB		:>>>>>>>>> BEGIN SUBTEST >>>>>>>>>			
1969	041462					T3.1:			
	041462								
	041462	104402							
1970	041464	004737	046610		JSR	PC,T31REST	:	SET COMMAND PACKET	TRAP CSBSUB
1971	041470	004737	046702		JSR	PC,T31RT2	:	SET UP OTHER COMMAND PACKET	
1972	041474	004737	046744		JSR	PC,T31RT3	:	SET UP OTHER COMMAND PACKET	
1973	041500	012737	176750	043342	MOV	#65000.,T31DLY	:	SET UP DELAY COUNTER	
1974	041506	004737	016054		JSR	PC,SOFINIT	:	DO INITIALIZE ON CONTROLLER	
1975	041512	103426		10\$:	BCS	20\$:	BR IF INIT WAS OK	
1976	041514				DELAY	250	:	DELAY ABOUT .25 SEC	
	041514	012727	000250					MOV	#250,(PC)+
	041520	000000						.WORD	0
	041522	013727	002116					MOV	LSDLY,(PC)+
	041526	000000						.WORD	0
	041530	005367	177772					DEC	-6(PC)
	041534	001375						BNE	.-4
	041536	005367	177756					DEC	-22(PC)
	041542	001367						BNE	.-20
1977	041544	005337	043342		DEC	T31DLY	:	BUMP COUNTER	
1978	041550	001356			BNE	10\$:	BR, IF COUNTER NOT DONE	
1979	041552	005237	002214		INC	FATFLG	:	ERROR COUNT	
1983	041556	010001			MOV	R0,R1	:	CONTENTS OF TSSR REGISTER	
1984	041560				ERRDF	ERRNO,SFIERR,SFIMSG	:	FATAL ERROR TSSR WAS NOT OK	
	041560	104455						TRAP	C\$ERDF
	041562	000455						.WORD	301
	041564	003646						.WORD	SFIERR
	041566	012114						.WORD	SFIMSG
1985	041570	013737	002174	043210	20\$: MOV	UNITN,T31DSW	:	SET UP UNIT NUMBER IN PACKET	
1986	041576	012704	043170		MOV	#T31PACKET,R4	:	SUBROUTINE NEEDS PACKET ADDRESS	
1987	041602	004737	010742		JSR	PC,WRTCHR	:	ISSUE WRITE CHARACTERISTICS	
1988	041606	103407			BCS	23\$:	BR, IF COMMAND ISSUED OK	
1989	041610	005237	002214		INC	FATFLG	:	ERROR COUNT	
1993	041614	010001			MOV	R0,R1	:	SAVE CONTENTS OF TSSR	
1994	041616				ERRHRD	ERRNO,WRTMSG,SFIMSG	:	WRITE CHARACTERISTICSC FAILED	
	041616	104456						TRAP	C\$ERHRD
	041620	000456						.WORD	302

Year	Card	Address	Label	Operation	Operand	Comment	Register
1995	041622	005052				.WORD	WRTMSG
	041624	012114				.WORD	SFIMSG
	041626	104406	23\$:	CKLOOP		;LOOP IF SELECTED	
1996	041630	004737	011074	JSR	PC,REWIND	;CALL TAPE REWIND COMMAND	TRAP
1997	041634	103407		BCS	30\$;BR, IF NO PROBLEM	C\$CLP1
1998	041636	010004		MOV	R0,R4	;SET UP REWIND PACKET ADDRESS	
1999	041640	005237	002214	INC	FATFLG	;ERROR COUNT	
2003	041644			ERRHRD	ERRNO,T31RWN,PKTSSR	;REWIND NOT ACCEPTED	
	041644	104456				TRAP	C\$ERHRD
	041646	000457				.WORD	303
	041650	044674				.WORD	T31RWN
	041652	012126				.WORD	PKTSSR
2004	041654			30\$:	CKLOOP	;LOOP IF SELECTED	
	041654	104406				TRAP	C\$CLP1
2005	041656	013701	043220	MOV	T31BFR+6,R1	;PICK UP XSTO	
2006	041662	010102		MOV	R1,R2	;SET UP EXPECTED	
2007	041664	052702	000002	BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED	
2008	041670	020102		CMP	R1,R2	;DOES EXP = REC'D	
2009	041672	001406		BEQ	40\$;BR, IF EQUAL (OK)	
2010	041674	005237	002214	INC	FATFLG	;ERROR COUNT	
2014	041700			ERRHRD	ERRNO,T31BOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND	
	041700	104456				TRAP	C\$ERHRD
	041702	000460				.WORD	304
	041704	044345				.WORD	T31BOT
	041706	015554				.WORD	EXPREC
2015	041710			40\$:	CKLOOP	;LOOP IF SELECTED	
	041710	104406				TRAP	C\$CLP1
2016	041712	013737	003116	MOV	FREE,T31WB	;STARTING WRITE BUFFER ADDRESS	
2017	041720	012737	140005	MOV	#140005,T31PK3	;WRITE DATA,CVC=1,ACK COMMAND	
2018	041726	012704	043310	MOV	#T31PK3,R4	;SET UP R4 WITH PACKET ADDRESS	
2019	041732	012700	000144	MOV	#100.,R0	;SET PATTERN IN CORRECT REGISTER	
2020	041736	004737	017502	JSR	PC,FILLMEM	;FILL MEMORY WITH RECORD SIZE	
2021	041742	012737	000144	MOV	#100.,T31SZ	;SET UP RECORD SIZE IN PACKET	
2022	041750	010465	000000	MOV	R4,T31WDC(R5)	;ISSUE COMMAND	
2023	041754	004737	016330	JSR	PC,WAITF	;WAIT FOR SSR TO SET	
2024	041760	016501	000002	MOV	TSSR(R5),R1	;GET TSSR CONTENTS	
2025	041764	012702	000200	MOV	#SSR,R2	;SET UP EXPECTED	
2026	041770	020102		CMP	R1,R2	;ARE THEY EQUAL	
2027	041772	001406		BEQ	80\$;BR, IF OK	
2028	041774	005237	002214	INC	FATFLG	;ERROR COUNT	
2032	042000			ERRHRD	ERRNO,T31WDC,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA	
	042000	104456				TRAP	C\$ERHRD
	042002	000461				.WORD	305
	042004	045230				.WORD	T31WDC
	042006	012126				.WORD	PKTSSR
2033	042010			80\$:	CKLOOP	;LOOP IF SELECTED	
	042010	104406				TRAP	C\$CLP1
2034	042012	004737	011074	JSR	PC,REWIND	;CALL TAPE REWIND COMMAND	
2035	042016	103407		BCS	230\$;BR, IF NO PROBLEM	
2036	042020	010001		MOV	R0,R1	;SAVE TSSR	
2037	042022	005237	002214	INC	FATFLG	;ERROR COUNT	
2041	042026	</					

2042	042036			230\$:	CKLOOP		:LOOP IF SELECTED	
	042036	104406					TRAP	C\$CLP1
2043	042040	013701	043220		MOV	T31BFR+6,R1	:PICK UP XSTO	
2044	042044	010102			MOV	R1,R2	:SET UP EXPECTED	
2045	042046	052702	000002		BIS	#BIT1,R2	:SET BOT BIT IN EXPECTED	
2046	042052	020102			CMP	R1,R2	:DOES EXP = REC'D	
2047	042054	001406			BEQ	240\$:BR, IF EQUAL (OK)	
2048	042056	005237	002214		INC	FATFLG	:ERROR COUNT	
2052	042062				ERRHRD	ERRNO,T31BOT,EXPREC	:TAPE NOT AT BOT AFTER REWIND	
	042062	104456					TRAP	C\$ERHRD
	042064	000463					.WORD	307
	042066	044345					.WORD	T31BOT
	042070	015554					.WORD	EXPREC
2053	042072			240\$:	CKLOOP		:LOOP IF SELECTED	
	042072	104406					TRAP	C\$CLP1
2054	042074	012737	041012	043310	265\$:	MOV	#041012,T31PK3	:NO-OP,CVC=1 COMMAND
2055	042102	012704	043310		MOV	#T31PK3,R4	:SET UP R4 WITH PACKET ADDRESS	
2056	042106	010337	043316		MOV	R3,T31SZ	:SET UP RECORD SIZE IN PACKET	
2057	042112	010465	000000		MOV	R4,TSD8(R5)	:ISSUE COMMAND	
2058	042116	004737	016330		JSR	PC,WAITF	:WAIT FOR SSR TO SET	
2059	042122	016501	000002		MOV	TSSR(R5),R1	:GET TSSR CONTENTS	
2060	042126	012702	000200		MOV	#SSR,R2	:SET UP EXPECTED	
2061	042132	020102			CMP	R1,R2	:ARE THEY EQUAL	
2062	042134	001406			BEQ	280\$:BR, IF OK	
2063	042136	005237	002214		INC	FATFLG	:ERROR COUNT	
2067	042142				ERRHRD	ERRNO,T31RDF,PKTSSR	:TSSR INCORRECT AFTER READ DATA	
	042142	104456					TRAP	C\$ERHRD
	042144	000464					.WORD	308
	042146	043543					.WORD	T31RDF
	042150	012126					.WORD	PKTSSR
2068	042152			280\$:	CKLOOP		:LOOP IF SELECTED	
	042152	104406					TRAP	C\$CLP1
2069	042154	013701	043220		MOV	T31BFR+6,R1	:PICK UP XSTO	
2070	042160	010102			MOV	R1,R2	:SET UP EXPECTED	
2071	042162	052702	000002		BIS	#BIT1,R2	:SET BOT BIT IN EXPECTED	
2072	042166	020102			CMP	R1,R2	:DOES EXP = REC'D	
2073	042170	001406			BEQ	285\$:BR, IF EQUAL (OK)	
2074	042172	005237	002214		INC	FATFLG	:ERROR COUNT	
2078	042176				ERRHRD	ERRNO,T31BOT,EXPREC	:TAPE NOT AT BOT AFTER REWIND	
	042176	104456					TRAP	C\$ERHRD
	042200	000465					.WORD	309
	042202	044345					.WORD	T31BOT
	042204	015554					.WORD	EXPREC
2079	042206			285\$:	CKLOOP		:LOOP IF SELECTED	
	042206	104406					TRAP	C\$CLP1
2080	042210	012737	140001	043310	MOV	#140001,T31PK3	:READ,ACK,CVC=1 COMMAND	
2081	042216	012704	043310		MOV	#T31PK3,R4	:SET UP R4 WITH PACKET ADDRESS	
2082	042222	012737	000144	043316	MOV	#100.,T31SZ	:SET UP RECORD SIZE IN PACKET	
2083	042230	010465	000000		MOV	R4,TSD8(R5)	:ISSUE COMMAND	
2084	042234	004737	016330		JSR	PC,WAITF	:WAIT FOR SSR TO SET	
2085	042240	016501	000002		MOV	TSSR(R5),R1	:GET TSSR CONTENTS	
2086	042244	012702	000200		MOV	#SSR,R2	:SET UP EXPECTED	
2087	042250	020102			CMP	R1,R2	:ARE THEY EQUAL	
2088	042252	001406			BEQ	290\$:BR, IF OK	
2089	042254	005237	002214		INC	FATFLG	:ERROR COUNT	
2093	042260				ERRHRD	ERRNO,T31RDE,PKTSSR	:TSSR INCORRECT AFTER READ DATA	
	042260	104456					TRAP	C\$ERHRD

PC	PC+4	PC+8	PC+12	PC+16	PC+20	PC+24	PC+28	PC+32	PC+36	PC+40	PC+44	PC+48	PC+52	PC+56	PC+60	PC+64	PC+68	PC+72	PC+76	PC+80	PC+84	PC+88	PC+92	PC+96	PC+100	PC+104	PC+108	PC+112	PC+116	PC+120	PC+124	PC+128	PC+132	PC+136	PC+140	PC+144	PC+148	PC+152	PC+156	PC+160	PC+164	PC+168	PC+172	PC+176	PC+180	PC+184	PC+188	PC+192	PC+196	PC+200	PC+204	PC+208	PC+212	PC+216	PC+220	PC+224	PC+228	PC+232	PC+236	PC+240	PC+244	PC+248	PC+252	PC+256	PC+260	PC+264	PC+268	PC+272	PC+276	PC+280	PC+284	PC+288	PC+292	PC+296	PC+300	PC+304	PC+308	PC+312	PC+316	PC+320	PC+324	PC+328	PC+332	PC+336	PC+340	PC+344	PC+348	PC+352	PC+356	PC+360	PC+364	PC+368	PC+372	PC+376	PC+380	PC+384	PC+388	PC+392	PC+396	PC+400	PC+404	PC+408	PC+412	PC+416	PC+420	PC+424	PC+428	PC+432	PC+436	PC+440	PC+444	PC+448	PC+452	PC+456	PC+460	PC+464	PC+468	PC+472	PC+476	PC+480	PC+484	PC+488	PC+492	PC+496	PC+500	PC+504	PC+508	PC+512	PC+516	PC+520	PC+524	PC+528	PC+532	PC+536	PC+540	PC+544	PC+548	PC+552	PC+556	PC+560	PC+564	PC+568	PC+572	PC+576	PC+580	PC+584	PC+588	PC+592	PC+596	PC+600	PC+604	PC+608	PC+612	PC+616	PC+620	PC+624	PC+628	PC+632	PC+636	PC+640	PC+644	PC+648	PC+652	PC+656	PC+660	PC+664	PC+668	PC+672	PC+676	PC+680	PC+684	PC+688	PC+692	PC+696	PC+700	PC+704	PC+708	PC+712	PC+716	PC+720	PC+724	PC+728	PC+732	PC+736	PC+740	PC+744	PC+748	PC+752	PC+756	PC+760	PC+764	PC+768	PC+772	PC+776	PC+780	PC+784	PC+788	PC+792	PC+796	PC+800	PC+804	PC+808	PC+812	PC+816	PC+820	PC+824	PC+828	PC+832	PC+836	PC+840	PC+844	PC+848	PC+852	PC+856	PC+860	PC+864	PC+868	PC+872	PC+876	PC+880	PC+884	PC+888	PC+892	PC+896	PC+900	PC+904	PC+908	PC+912	PC+916	PC+920	PC+924	PC+928	PC+932	PC+936	PC+940	PC+944	PC+948	PC+952	PC+956	PC+960	PC+964	PC+968	PC+972	PC+976	PC+980	PC+984	PC+988	PC+992	PC+996	PC+1000	PC+1004	PC+1008	PC+1012	PC+1016	PC+1020	PC+1024	PC+1028	PC+1032	PC+1036	PC+1040	PC+1044	PC+1048	PC+1052	PC+1056	PC+1060	PC+1064	PC+1068	PC+1072	PC+1076	PC+1080	PC+1084	PC+1088	PC+1092	PC+1096	PC+1100	PC+1104	PC+1108	PC+1112	PC+1116	PC+1120	PC+1124	PC+1128	PC+1132	PC+1136	PC+1140	PC+1144	PC+1148	PC+1152	PC+1156	PC+1160	PC+1164	PC+1168	PC+1172	PC+1176	PC+1180	PC+1184	PC+1188	PC+1192	PC+1196	PC+1200	PC+1204	PC+1208	PC+1212	PC+1216	PC+1220	PC+1224	PC+1228	PC+1232	PC+1236	PC+1240	PC+1244	PC+1248	PC+1252	PC+1256	PC+1260	PC+1264	PC+1268	PC+1272	PC+1276	PC+1280	PC+1284	PC+1288	PC+1292	PC+1296	PC+1300	PC+1304	PC+1308	PC+1312	PC+1316	PC+1320	PC+1324	PC+1328	PC+1332	PC+1336	PC+1340	PC+1344	PC+1348	PC+1352	PC+1356	PC+1360	PC+1364	PC+1368	PC+1372	PC+1376	PC+1380	PC+1384	PC+1388	PC+1392	PC+1396	PC+1400	PC+1404	PC+1408	PC+1412	PC+1416	PC+1420	PC+1424	PC+1428	PC+1432	PC+1436	PC+1440	PC+1444	PC+1448	PC+1452
----	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

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.

2124	042340		BGNSUB			; >>>>>>>>> BEGIN SUBTEST >>>>>>>>>			
	042340					T3.2:			
	042340	104402					TRAP	C\$BSUB	
2125	042342	004737	046610			:SET COMMAND PACKET			
2126	042346	004737	046702			:SET UP OTHER COMMAND PACKET			
2127	042352	004737	046744			:SET UP OTHER COMMAND PACKET			
2128	042356	004737	016054			:DO INITIALIZE ON CONTROLLER			
2129	042362	103407				:BR IF INIT WAS OK			
2130	042364	005237	002214			:ERROR COUNT			
2134	042370	010001				:CONTENTS OF TSSR REGISTER			
2135	042372					:FATAL ERROR TSSR WAS NOT OK			
	042372	104455					TRAP	C\$ERDF	
	042374	000470					.WORD	312	
	042376	003646					.WORD	SFIERR	
	042400	012114					.WORD	SFIMSG	
2136	042402	013737	002174	043210	20\$:	:SET UP UNIT NUMBER IN PACKET			
2137	042410	012704	043170			:SUBROUTINE NEEDS PACKET ADDRESS			
2138	042414	004737	010742			:ISSUE WRITE CHARACTERISTICS			
2139	042420	103407				:BR, IF COMMAND ISSUED OK			
2140	042422	005237	002214			:ERROR COUNT			
2144	042426	010001				:SAVE CONTENTS OF TSSR			
2145	042430					:WRITE CHARACTERISTISC FAILED			
	042430	104456					TRAP	C\$ERHRD	
	042432	000471					.WORD	313	
	042434	005052					.WORD	WRTMSG	
	042436	012114					.WORD	SFIMSG	
2146	042440				23\$:	:LOOP IF SELECTED			
	042440	104406					TRAP	(C\$CLP1	
2147	042442	004737	011074			:CALL TAPE REWIND COMMAND			
2148	042446	103407				:BR, IF NO PROBLEM			
2149	042450	010004				:SET UP REWIND PACKET ADDRESS			
2150	042452	005237	002214			:ERROR COUNT			
2154	042456					:REWIND NOT ACCEPTED			
	042456	104456					TRAP	C\$ERHRD	
	042460	000472					.WORD	314	
	042462	044674					.WORD	T31RWn	
	042464	012126					.WORD	PKTSSR	
2155	042466				30\$:	:LOOP IF SELECTED			
	042466	104406					TRAP	C\$CLP1	
2156	042470	013701	043220			:PICK JP XSTO			
2157	042474	010102				:SET UP EXPECTED			
2158	042476	052702	000002			:SET BOT BIT IN EXPECTED			
2159	042502	020102				:DOES EXP = REC'D			
2160	042504	001406				:BR, IF EQUAL (OK)			

Line	Address	Offset	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	
------	---------	--------	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

2210	042734	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
2211	042740	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED		
2212	042744	020102			CMP	R1,R2	;ARE THEY EQUAL		
2213	042746	001406			BEG	280\$;BR, IF OK		
2214	042750	005237	002214		INC	FATFLG	;ERROR COUNT		
2218	042754				ERRHRD	ERRNO,T31RDF,PKTSSR	;TSSR INCORRECT AFTER READ DATA		
	042754	104456					TRAP	\$SERHRD	
	042756	000477					.WORD	319	
	042760	043543					.WORD	T31RDF	
	042762	012126					.WORD	PKTSSR	
2219	042764			280\$: CKLOOP			;LOOP IF SELECTED		
	042764	104406					TRAP	\$CLP1	
2220	042766	013701	043220		MOV	T31BFR+6,R1	;PICK UP XSTO		
2221	042772	010102			MOV	R1,R2	;SET UP EXPECTED		
2222	042774	052702	000002		BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED		
2223	043000	020102			CMP	R1,R2	;DOES EXP = REC'D		
2224	043002	001406			BEG	285\$;BR, IF EQUAL (OK)		
2225	043004	005237	002214		INC	FATFLG	;ERROR COUNT		
2229	043010				ERRHRD	ERRNO,T31BOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	043010	104456					TRAP	\$SERHRD	
	043012	000500					.WORD	320	
	043014	044345					.WORD	T31BOT	
	043016	015554					.WORD	EXPREC	
2230	043020			285\$: CKLOOP			;LOOP IF SELECTED		
	043020	104406					TRAP	\$CLPI	
2231	043022	012737	140001	043310	MOV	#14000!,T31PK3	;READ,ACK,CVC=1 COMMAND		
2232	043030	012704	043310		MOV	#T31PK3,R4	;SET UP R4 WITH PACKET ADDRESS		
2233	043034	012737	000144	043316	MOV	#100.,T31SZ	;SET UP RECORD SIZE IN PACKET		
2234	043042	010465	000000		MOV	R4,TSD(BR5)	;ISSUE COMMAND		
2235	043046	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
2236	043052	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
2237	043056	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED		
2238	043062	020102			CMP	R1,R2	;ARE THEY EQUAL		
2239	043064	001406			BEG	290\$;BR, IF OK		
2240	043066	005237	002214		INC	FATFLG	;ERROR COUNT		
2244	043072				ERRHRD	ERRNO,T31RDE,PKTSSR	;TSSR INCORRECT AFTER READ DATA		
	043072	104456					TRAP	\$SERHRD	
	043074	000501					.WORD	321	
	043076	043344					.WORD	T31RDE	
	043100	012126					.WORD	PKTSSR	
2245	043102			290\$: CKLOOP			;LOOP IF SELECTED		
	043102	104406					TRAP	\$CLP1	
2246	043104	017701	140006		MOV	@FREE,R1	;GET DATA READ		
2247	043110	012702	000144		MOV	#100.,R2	;READ EXPECTED		
2248	043114	020102			CMP	R1,R2	;DID TAPE STAY POSITIONED		
2249	043116	001406			BEG	330\$;BR, IF EXPD = RECD		
2250	043120	005237	002214		INC	FATFLG	;ERROR COUNT		
2254	043124				ERRHRD	ERRNO,T31WNH,EXPREC	;TAPE POSITION NOT CORRECT AFTER INIT		
	043124	104456					TRAP	\$SERHRD	
	043126	000502					.WORD	322	
	043130	043410					.WORD	T31WNH	
	043132	015554					.WORD	EXPREC	
2255	043134			330\$: -	-				
2256	043134			ENDSUB			>>>>>>>>> END SUBTEST >>>>>>>>		

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 25-MAY-82 08:43 H 14
TEST 3: NO-OP ('CLEAN TAPE') AND INITIALIZE PAGE 100-3

SEQ 0176

2258 043144 103402
2259 043146 004737 017262
2260 043152
2261
2262
2263
2264 043152 004737 016536
2265 043156 103002
2266 043160 000137 041462
2267 043164
043164 104432
043166 003600

BLO 999\$
JSR PC,CKDROP
999\$:
:
:
:
JSR PC,TSTLOOP
BCC 163\$
JMP T31LOOP
163\$: EXIT TST

;BR, IF LESS THAN 25
;TRY TO DROP THE UNIT

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

TRAP C\$EXIT
.WORD L10050-

```

2269
2270      ;+
2271      ;LOCAL STORAGE FOR THIS TEST
2272      ;-
2275 043170      T31PACKET:      ;COMMAND PACKET FOR TEST
2276 043170      .WORD 100004      ;WRITE CHARACTERISTICS COMMAND, WITH , ACK
2277 043172      .WORD T31DATA      ;ADDRESS OF CHARACTERISTICS BLOCK
2278 043174      .WORD 0
2279 043176      .WORD 10.      ;STARTING VALUE OF BLOCK SIZE
2280 043200      T31DATA:      ;CHARACTERISTICS DATA BLOCK
2281 043200      .WORD T31BFR      ;ADDRESS OF MESSAGE BUFFER
2282 043202      .WORD 0
2283 043204      .WORD 20.      ;LENGTH OF MESSAGE BUFFER
2284 043206      .WORD 0
2285 043210      T31DSW: .WORD 0      ;SELECT DRIVE 0
2286 043212      T31BFR: .BLKW 25.      ;MESSAGE BUFFER
2287
2288      ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
2289      ;
2291      .=<.+10>&177770
2293 043300      T31PK2:      ;WRITE SUB SYS MEM COMMAND, AND ACK
2294 043300      .WORD 100006      ;ADDRESS OF SELECT BLOCK DATA
2295 043302      .WORD T31BF2
2296 043304      .WORD 0
2297 043306      .WORD 6.      ;SIZE OF DATA PACKET
2298
2302 043310      T31PK3:      ;REREAD COMMAND, AND ACK
2303 043310      .WORD 100005
2304 043312      T31RB:      ;ADDRESS OF WRITE BUFFER
2305 043312      T31WB: .WORD FREE
2306 043314      .WORD 0
2307 043316      T31SZ: .WORD 0      ;SIZE OF BUFFER (EXTENT)
2308      .EVEN
2309      ;
2310      ;
2311      ;
2312 043320      T31BF2:
2313 043320      T31BS0: .BYTE 10      ;BSEL0 AREA
2314 043321      T31BS1: .BYTE 200      ;BSEL1 AREA
2315 043322      T31S2: .WORD 0      ;SEL 2 AREA
2316 043324      T31S3: .WORD 0      ;DATA AREA
2317      ;
2318      ;
2319      .EVEN
2320      ;TAPE MOTION PACKET COMMAND VALUES
2321
2322 043326      T31RN: .WORD 100205      ;REREAD DATA (NEXT)
2323 043330      T31WDR: .WORD 100605      ;REREAD DATA RETRY
2324 043332      T31CON: .WORD 102205      ;WRITE CONTINUOUS
2325 043334      .WORD 177777      ;END OF DATA
2326
2327      ;
2328 043336      T31CNT: .WORD 0      ;TAPE TIMER COUNTER STORAGE AREA
2329 043340      T31CNU: .WORD 0      ;TAPE TIMER COUNTER STORAGE AREA
2330 043342      T31DLY: .WORD 0      ;DELAY COUNTER
2331

```

```

2333
2334
2335      ;+
2336      ;LOCAL TEXT MESSAGES FOR TEST
2337      ;+
2338
2339
2340 043344      124      123      123      T31RDE: .ASCIIZ      'TSSR Not Correct After READ Command'
2341 043410      124      141      160      T31WNH: .ASCIIZ      'Tape Position Incorrect After INITIALIZE Command'
2342 043471      124      141      160      T31WNG: .ASCIIZ      'Tape Position Incorrect After NOP Command'
2343 043543      124      123      123      T31RDF: .ASCIIZ      'TSSR Incorrect After READ DATA Command'
2344 043612      122      105      122      T31RRF: .ASCIIZ      'REREAD Previous (Space Reverse, Read Forward) Command Failed'
2345 043707      120      117      123      T31SC: .ASCIIZ      'POSITION (Space Command) Failed, TSSR Not Correct'
2346 043771      122      111      102      T31LOR: .ASCIIZ      'RIB NOT SET AFTER READ REVERSE INTO BOT'
2347 044041      124      123      123      T31WDF: .ASCIIZ      'TSSR Not Correct After Illegal Mode Bits Set'
2348 044116      111      154      154      T31LOQ: .ASCIIZ      'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
2349 044177      122      105      122      T31SSR: .ASCIIZ      'REREAD COMMAND Not Accepted'
2350 044233      124      123      123      T31WDE: .ASCIIZ      'TSSR Not Correct After NO-OP ('CLEAN TAPE') AND INITIALIZE Command, At BOT'
2351 044345      124      141      160      T31BOT: .ASCIIZ      'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
2352 044440      116      117      055      T31TIM: .ASCIIZ      'NO-OP ('CLEAN TAPE') AND INITIALIZE'S Erase Tape Not Long Enough'
2353 044540      122      105      122      T31EOT: .ASCIIZ      'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
2354 044617      124      123      123      T31TM: .ASCIIZ      'TSSR Not Correct After REREAD COMMAND Reject'
2355 044674      122      145      167      T31RWN: .ASCIIZ      'Rewind (POSITION) Command Not Accepted'
2356 044743      122      101      115      T31RNC: .ASCIIZ      'RAM Error, Correct Data Pattern Not In Ram'
2357 045016      124      123      123      T31AM3: .ASCIIZ      'TSSR Init. Failed After REREAD COMMAND'
2358 045065      104      162      151      T31OFL: .ASCIIZ      'Drive 7 Select Failed To Set 'OFL' In TSSR'
2359 045140      124      123      123      T31WDD: .ASCIIZ      'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
2360 045230      124      123      123      T31WDC: .ASCIIZ      'TSSR Not Correct After REREAD DATA Command'
2361 045303      103      126      103      T31VCK: .ASCIIZ      'CVC Set, Didn't Reset VCX In Message Buffer'
2362 045356      124      123      102      T31BA: .ASCIIZ      'TSBA Not Correct After REREAD DATA Command'
2363 045431      127      122      111      T31WSS: .ASCIIZ      'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
2364 045520      122      145      141      T31LON: .ASCIIZ      'Reading Long Record Failed To Set RLL Bit In XST0'
2365 045602      122      145      141      T31LOP: .ASCIIZ      'Reading Long Record Failed To Set RLS Bit In XST0'
2366 045664      122      145      163      T31PBP: .ASCIIZ      'Residual Byte Count Incorrect After Short Record Read'
2367 045752      122      145      141      T31TRL: .ASCIIZ      'Reading Long Record Failed To Give Tape Status Alert'
2368 046040      116      117      055      T31NEF: .ASCIIZ      'NO-OP ('CLEAN TAPE') AND INITIALIZE, At First Record, Failed To Set RIB Bit
2369 046161      124      123      123      T31SCF: .ASCIIZ      'TSSR Not Correct After SPACE RECORDS Command'
2370 046236      124      123      123      T31TSA: .ASCIIZ      'TSSR Not Correct After NO-OP ('CLEAN TAPE') AND INITIALIZE, Into BOT'
2371 046343      124      123      123      T31WRF: .ASCIIZ      'TSSR Not Correct After NO-OP ('CLEAN TAPE') AND INITIALIZE Command'
2372 046446      104      141      164      T31DTA: .ASCIIZ      'Data Compare Error, Data Read From Tape Not Equal To Written'
2373 046543      116      117      055      T31IDA: .ASCIIZ      'NO-OP ('Clean Tape') And INITIALIZE'
2374
2375      .EVEN
2376
2377      ;+
2378      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
2379      ;WRITE SUBSYSTEM MEMORY COMMAND
2380      ;+
2381
2382
2383
2384 046610      012701      043170
2385 046610      012721      100004
2386 046614      012721      043200
2387 046620      005021
2388 046632      012721      000012
2389 046636      012721      043212

      T31REST:
      SAVREG
      MOV      #T31PACKET,R1      ;SAVE THE REGISTERS
      MOV      #100004,(R1)+      ;START OF THE PACKET
      MOV      #T31DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,
      CLR      (R1)+              ;ADDRESS OF CHARACTERISTICS DATA BLOCK
      MOV      #10,(R1)+          ;EXTENDED ADDRESS
      MOV      #T31BFR,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
      ;ADDRESS OF MESSAGE BUFFER
  
```

2390	046642	005021		CLR	(R1)+	
2391	046644	012721	000024	MOV	#20.,(R1)+	:LENGTH OF MESSAGE BUFFER
2392	046650	005021		CLR	(R1)+	
2393	046652	012711	000000	MOV	#0,(R1)	:SELECT DRIVE ZERO
2394	046656	012702	000030	MOV	#24.,R2	:NUMBER OF LOCATIONS TO BE CLEARED
2395	046662	012762	177777 043212 64\$:	MOV	#177777,T31BFR(R2)	:ALL ONES TO MESSAGE BUFFER
2396	046670	005742		TST	-(R2)	:NEXT LOCATION
2397	046672	022702	000000	CMP	#0,R2	:AT END OF LOOP YET
2398	046676	001371		BNE	64\$:KEEP GOING UNTIL DONE
2399	046700	000207		RTS	PC	:RETURN
2400						
2401						
2402	046702			T31RT2:		
2403	046702			SAVREG		:SAVE THE REGISTERS
2404	046706	012701	043300	MOV	#T31PK2,R1	:START OF THE PACKET
2405	046712	012721	100006	MOV	#100006,(R1)+	:WRITE SUBSYSTEM MEM. WITH ACK.
2406	046716	012721	043320	MOV	#T31BF2,(R1)+	:ADDRESS OF DATA BLOCK
2407	046722	005021		CLR	(R1)+	:EXTENDED ADDRESS
2408	046724	012721	000006	MOV	#6.,(R1)+	:SIZE OF DATA BLOCK IN BYTES
2409	046730	005021		CLR	(R1)+	
2410	046732	012701	043320	MOV	#T31BF2,R1	:POINT TO DATA SEL AREA
2411	046736	005021		CLR	(R1)+	
2412	046740	005011		CLR	(R1)	
2413	046742	000207		RTS	PC	:RETURN
2414	046744			T31RT3:		
2415	046744			SAVREG		:SAVE REGISTERS
2416	046750	012701	043310	MOV	#T31PK3,R1	:SET UP POINTER ADDRESS
2417	046754	005021		CLR	(R1)+	:COMMAND SPACE
2418	046756	005021		CLR	(R1)+	:ADDRESS OF DATA BLOCK
2419	046760	005021		CLR	(R1)+	:EXTENDED ADDRESS
2420	046762	005011		CLR	(R1)	:SIZE OF DATA TRANSFER BLOCK
2421	046764	000207		RTS	PC	:RETURN
2422	046766			ENDTST		
	046766					
	046766	104401				

L10050: TRAP CSETST

2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483

.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.

THE TEST CONSISTS OF THE FOLLOWING 3 SUBTESTS

BGNTST

MOV #EPRT1,EPRTSW
 MOV #TST32ID,R0
 JSR PC,TSTSETUP
 MOV #5,LOOPCNT
 CLR T32CNT

T4::
 ;PRIMARY ERROR MESSAGE
 ;ASCII MESSAGE TO IDENTIFY TEST
 ;DO INITIAL TEST SETUP
 ;PERFORM 5 ITERATIONS
 ;CLEAR TAPE RECORD COUNTER

TEST 4, SUBTEST 1

VERIFIES THAT A Erase And Operation Incomplete COMMAND ISSUED WHILE
 THE TAPE IS POSITIONED AT BOT CAUSES FUNCTION REJECT
 TERMINATION, WITH THE NON-EXECUTABLE FUNCTION (NEF)
 ERROR BIT SET.

T32LOOP:

046770
 046770
 046770 012737 006354 002172
 046776 012700 052640
 047002 004737 016570
 047006 012737 000005 002210
 047014 005037 051510
 047020

[illegible]

2527	047232	012737	140005	051450	MOV	#140005,T32PK3	:WRITE DATA,CVC=1,ACK COMMAND	
2528	047240	012704	051450		MOV	#T32PK3,R4	:SET UP R4 WITH PACKET ADDRESS	
2529	047244	010337	051456	27\$:	MOV	R3,T32SZ	:SET UP RECORD SIZE IN PACKET	
2530	047250	010465	000000		MOV	R4,TSD8(R5)	:ISSUE COMMAND	
2531	047254	004737	016330		JSR	PC,WAITF	:WAIT FOR SSR TO SET	
2532	047260	016501	000002		MOV	TSSR(R5),R1	:GET TSSR CONTENTS	
2533	047264	012702	000200		MOV	#SSR,R2	:SET UP EXPECTED	
2534	047270	020102			CMP	R1,R2	:ARE THEY EQUAL	
2535	047272	001406			BEQ	28\$:BR, IF OK	
2536	047274	005237	002214		INC	FATFLG	:ERROR COUNT	
2540	047300				ERRHRD	ERRNO,T32WDC,PKTSSR	:TSSR INCORRECT AFTER WRITE DATA	
	047300	104456					TRAP	C\$ERHRD
	047302	000624					.WORD	404
	047304	052536					.WORD	T32WDC
	047306	012126					.WORD	PKTSSR
2541	047310			28\$:	CKLOOP		:LOOP IF SELECTED	
	047310	104406					TRAP	C\$CLP1
2542	047312	005723			TST	(R3)+	:BUMP RECORD COUNTER	
2543	047314	020327	001002		CMP	R3,#514.	:AT MAX SIZE YET	
2544	047320	001351			BNE	27\$:BR, IF NOT AT END OF LOOP	
2545	047322	004737	011074		JSR	PC,REWIND	:CALL TAPE REWIND COMMAND	
2546	047326	103411			BCS	30\$:BR, IF NO PROBLEM	
2547	047330	016501	000002		MOV	TSSR(R5),R1	:GET TSSR CONTENTS	
2548	047334	010004			MOV	R0,R4	:SET UP REWIND PACKET ADDRESS	
2549	047336	005237	002214		INC	FATFLG	:ERROR COUNT	
2553	047342				ERRHRD	ERRNO,T32RWN,PKTSSR	:REWIND NOT ACCEPTED	
	047342	104456					TRAP	C\$ERHRD
	047344	000625					.WORD	405
	047346	051700					.WORD	T32RWN
	047350	012126					.WORD	PKTSSR
2554	047352			30\$:	CKLOOP		:LOOP IF SELECTED	
	047352	104406					TRAP	C\$CLP1
2555	047354	013701	051360		MOV	T32BFR+6,R1	:PICK UP XSTO	
2556	047360	010102			MOV	R1,R2	:SET UP EXPECTED	
2557	047362	052702	000002		BIS	#BIT1,R2	:SET BOT BIT IN EXPECTED	
2558	047366	020102			CMP	R1,R2	:DOES EXP = REC'D	
2559	047370	001406			BEQ	40\$:BR, IF EQUAL (OK)	
2560	047372	005237	002214		INC	FATFLG	:ERROR COUNT	
2564	047376				ERRHRD	ERRNO,T32BOE,EXPREC	:TAPE AT BOT AFTER ERASE	
	047376	104456					TRAP	C\$ERHRD
	047400	000626					.WORD	406
	047402	052366					.WORD	T32BOE
	047404	015554					.WORD	EXPREC
2565	047406			40\$:	CKLOOP		:LOOP IF SELECTED	
	047406	104406					TRAP	C\$CLP1
2566	047410	012737	140411	051450	MOV	#140411,T32PK3	:ERASE TAPE,CVC=1,ACK COMMAND	
2567	047416	012704	051450		MOV	#T32PK3,R4	:SET UP R4 WITH PACKET ADDRESS	
2568	047422	010465	000000		MOV	R4,TSD8(R5)	:ISSUE COMMAND	
2569	047426	004737	016330		JSR	PC,WAITF	:WAIT FOR SSR TO SET	
2570	047432	016501	000002		MOV	TSSR(R5),R1	:GET TSSR CONTENTS	
2571	047436	012702	000200		MOV	#SSR,R2	:SET UP EXPECTED	
2572	047442	020102			CMP	R1,R2	:ARE THEY EQUAL	
2573	047444	001406			BEQ	50\$:BR, IF OK	
2574	047446	005237	002214		INC	FATFLG	:ERROR COUNT	
2578	047452				ERRHRD	ERRNO,T32ERA,PKTSSR	:TSSR INCORRECT AFTER ERASE DATA	
	047452	104456					TRAP	C\$ERHRD
	047454	000627					.WORD	407

	047456	052016					.WORD	T32ERA
	047460	012126					.WORD	PKTSSR
2579	047462		50\$:	CKLOOP		;LOOP IF SELECTED		
	047462	104406					TRAP	C\$CLP1
2580	047464	013701	051360	MOV	T32BFR+6,R1	;PICK UP XSTO		
2581	047470	010102		MOV	R1,R2	;SET UP EXPECTED		
2582	047472	042702	000002	BIC	#BIT1,R2	;SET BOT BIT IN EXPECTED		
2583	047476	020102		CMP	R1,R2	;DOES EXP = REC'D		
2584	047500	001406		BEQ	55\$;BR, IF EQUAL (OK)		
2585	047502	005237	002214	INC	FATFLG	;ERROR COUNT		
2589	047506			ERRHRD	ERRNO,T32BOE,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	047506	104456					TRAP	C\$ERHRD
	047510	000630					.WORD	408
	047512	052366					.WORD	T32BOE
	047514	015554					.WORD	EXPREC
2590	047516		55\$:	CKLOOP		;LOOP IF SELECTED		
	047516	104406					TRAP	C\$CLP1
2591	047520	013737	003116	MOV	FREE,T32RB	;ADDRESS OF BUFFER		
2592	047526	012737	140401	MOV	#140401,T32PK3	;READ REVERSE,ACK,CVC=1 COMMAND		
2593	047534	012737	000400	MOV	#256.,T32SZ	;SET UP THE SIZE OF RECORD		
2594	047542	012704	051450	MOV	#T32PK3,R4	;SET UP R4 WITH PACKET ADDRESS		
2595	047546	010465	000000	MOV	R4,TSD(B(R5))	;ISSUE COMMAND		
2596	047552	004737	016330	JSR	PC,WAITF	;WAIT FOR SSR TO SET		
2597	047556	016501	000002	MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
2598	047562	012702	100204	MOV	#SSR!SC!BIT2,R2	;SET UP EXPECTED TAPE STATUS ALERT		
2599	047566	020102		CMP	R1,R2	;ARE THEY EQUAL.		
2600	047570	001406		BEQ	180\$;BR, IF OK		
2601	047572	005237	002214	INC	FATFLG	;ERROR COUNT		
2605	047576			ERRHRD	ERRNO,T32TSA,PKTSSR	;TSSR INCORRECT AFTER READ DATA		
	047576	104456					TRAP	C\$ERHRD
	047600	000631					.WORD	409
	047602	052311					.WORD	T32TSA
	047604	012126					.WORD	PKTSSR
2606	047606		180\$:	CKLOOP		;LOOP IF SELECTED		
	047606	104406					TRAP	C\$CLP1
2607	047610	013701	051366	MOV	T32BFR+14,R1	;GET XST3 STATUS WORD		
2608	047614	010102		MOV	R1,R2	;SET UP EXPECTED		
2609	047616	052702	000001	BIS	#BIT0,R2	;SET THE RIB BIT		
2610	047622	020102		CMP	R1,R2	;ARE THEY EQUAL		
2611	047624	001406		BEQ	190\$;BR, IF EQUAL (GOOD)		
2612	047626	005237	002214	INC	FATFLG	;ERROR COUNT		
2616	047632			ERRHRD	ERRNO,T32RIB,EXPREC	;RIB SHOULD BE SET		
	047632	104456					TRAP	C\$ERHRD
	047634	000632					.WORD	410
	047636	052136					.WORD	T32RIB
	047640	015554					.WORD	EXPREC
2617	047642		190\$:					
2618	047642			ENDSUB		;>>>>>>>>>> END SUBTEST >>>>>>>>>		
	047642					L100>4:		
	047642	104403					TRAP	C\$ESUB
2619	047644	023727	002214	CMP	FATFLG,#15.	;IS ERROR COUNT AT 25		
2620	047652	103402	000017	BLO	999\$;BR, IF LESS THAN 25		
2621	047654	004737	017262	JSR	PC,CKDROP	;TRY TO DROP THE UNIT		
2622	047660		999\$:					

Address	Instruction	Operand	Comment	Trap	Trap Address
047752	000634			.WORD	412
047754	005052			.WORD	WRTMSG
047756	012114			.WORD	SFIMSG
2680 047760	23\$: CKLOOP		:LOOP IF SELECTED	TRAP	C\$CLP1
047760	104406				
2681 047762	004737	011074	JSR PC,REWIND		
2682 047766	103407		BCS 30\$		
2683 047770	010004		MOV R0,R4		
2684 047772	005237	002214	INC FATFLG		
2688 047776	ERRHRD		ERRNO,T32RWN,PKTSSR		
047776	104456		:REWIND NOT ACCEPTED	TRAP	C\$ERHRD
050000	000635			.WORD	413
050002	051700			.WORD	T32RWN
050004	012126			.WORD	PKTSSR
2689 050006	30\$: CKLOOP		:LOOP IF SELECTED	TRAP	C\$CLP1
050006	104406				
2690 050010	013701	05,360	MOV T32BFR+6,R1		
2691 050014	010102		MOV R1,R2		
2692 050016	052702	000002	BIS #BIT1,R2		
2693 050022	020102		CMP R1,R2		
2694 050024	001406		BEQ 40\$		
2695 050026	005237	002214	INC FATFLG		
2699 050032	ERRHRD		ERRNO,T32BOT,EXPREC		
050032	104456		:TAPE NOT AT BOT AFTER REWIND	TRAP	C\$ERHRD
050034	000636			.WORD	414
050036	051516			.WORD	T32BOT
050040	015554			.WORD	EXPREC
2700 050042	40\$: CKLOOP		:LOOP IF SELECTED	TRAP	C\$CLP1
050042	104406				
2701 050044	012703	000144	MOV #100.,R3		
2702 050050	010300		MOV R3,R0		
2703 050052	004737	017502	JSR PC,FILLMEM		
2704 050056	013737	003116	MOV FREE,T32WB		
2705 050064	012737	140005	MOV #140005,T32PK3		
2706 050072	012704	051450	MOV #T32PK3,R4		
2707 050076	010300		MOV R3,R0		
2708 050100	004737	017502	JSR PC,FILLMEM		
2709 050104	010337	051456	MOV R3,T32SZ		
2710 050110	010465	000000	MOV R4,TSDB(R5)		
2711 050114	004737	016330	JSR PC,WAITF		
2712 050120	016501	000002	MOV TSSR(R5),R1		
2713 050124	012702	000200	MOV #SSR,R2		
2714 050130	020102		CMP R1,R2		
2715 050132	001406		BEQ 80\$		
2716 050134	005237	002214	INC FATFLG		
2720 050140	ERRHRD		ERRNO,T32WDC,PKTSSR		
050140	104456		:TSSR INCORRECT AFTER WRITE DATA	TRAP	C\$ERHRD
050142	000637			.WORD	415
050144	052536			.WORD	T32WDC
050146	012126			.WORD	PKTSSR
2721 050150	80\$: CKLOOP		:LOOP IF SELECTED	TRAP	C\$CLP1
050150	104406				
2722 050152	005723		TST (R3)+		
2723 050154	020327	000156	CMP R3,#110.		
2724 050160	001341		BNE 65\$		
2725 050162	004737	011074	JSR PC,REWIND		
2726 050166	103407		BCS 230\$		
			:BUMP RECORD SIZE COUNTER		
			:AT 160 SIZE YET		
			:BR, IF MORE RECORDS TO WRITE		
			:CALL TAPE REWIND COMMAND		
			:BR, IF NO PROBLEM		

2727	050170	010001			MOV	R0,R1		:SAVE TSSR		
2728	050172	005237	002214		INC	FATFLG		:ERROR COUNT		
2732	050176				ERRHRD	ERRNO,T32RWN,EXPREC		:REWIND NOT ACCEPTED		
	050176	104456							TRAP	C\$ERHRD
	050200	000640							.WORD	416
	050202	051700							.WORD	T32RWN
	050204	015554							.WORD	EXPREC
2733	050206			230\$:	CKLOOP			:LOOP IF SELECTED		
	050206	104406							TRAP	C\$CLP1
2734	050210	013701	051360		MOV	T32BFR+6,R1		:PICK UP XSTO		
2735	050214	010102			MOV	R1,R2		:SET UP EXPECTED		
2736	050216	052702	000002		BIS	#BIT1,R2		:SET BOT BIT IN EXPECTED		
2737	050222	020102			CMP	R1,R2		:DOES EXP = REC'D		
2738	050224	001406			BEQ	240\$:BR, IF EQUAL (OK)		
2739	050226	005237	002214		INC	FATFLG		:ERROR COUNT		
2743	050232				ERRHRD	ERRNO,T32BOT,EXPREC		:TAPE NOT AT BOT AFTER REWIND		
	050232	104456							TRAP	C\$ERHRD
	050234	000641							.WORD	417
	050236	051516							.WORD	T32BOT
	050240	015554							.WORD	EXPREC
2744	050242			240\$:	CKLOOP			:LOOP IF SELECTED		
	050242	104406							TRAP	C\$CLP1
2745	050244	012703	000001		MOV	#1,R3		:SET UP FOR SPACE COMMAND		
2746	050250	004737	010544		JSR	PC,SPACE		:ISSUE SPACE COMMAND 1 FORWARD		
2747	050254	012737	140411	051450	265\$:	MOV	#140411,T32PK3	:ERASE DATA,ACK COMMAND		
2748	050262	012704	051450		MOV	#T32PK3,R4		:SET UP R4 WITH PACKET ADDRESS		
2749	050266	010465	000000		MOV	R4,TSDB(R5)		:ISSUE COMMAND		
2750	050272	004737	016330		JSR	PC,WAITF		:WAIT FOR SSR TO SET		
2751	050276	016501	000002		MOV	TSSR(R5),R1		:GET TSSR CONTENTS		
2752	050302	012702	000200		MOV	#SSR,R2		:SET UP EXPECTED		
2753	050306	020102			CMP	R1,R2		:ARE THEY EQUAL		
2754	050310	001406			BEQ	280\$:BR, IF OK		
2755	050312	005237	002214		INC	FATFLG		:ERROR COUNT		
2759	050316				ERRHRD	ERRNO,T32ERA,PKTSSR		:TSSR INCORRECT AFTER READ DATA		
	050316	104456							TRAP	C\$ERHRD
	050320	000642							.WORD	418
	050322	052016							.WORD	T32ERA
	050324	012126							.WORD	PKTSSR
2760	050326			280\$:	CKLOOP			:LOOP IF SELECTED		
	050326	104406							TRAP	C\$CLP1
2761	050330	013737	003116	051452	MOV	FREE,T32RB		:ADDRESS OF BUFFER		
2762	050336	012737	140401	051450	MOV	#140401,T32PK3		:READ REVERSE,ACK,CVC=1 COMMAND		
2763	050344	012737	000144	051456	MOV	#100,T32SZ		:SET UP THE SIZE OF RECORD		
2764	050352	012704	051450		MOV	#T32PK3,R4		:SET UP R4 WITH PACKET ADDRESS		
2765	050356	010465	000000		MOV	R4,TSDB(R5)		:ISSUE COMMAND		
2766	050362	004737	016330		JSR	PC,WAITF		:WAIT FOR SSR TO SET		
2767	050366	016501	000002		MOV	TSSR(R5),R1		:GET TSSR CONTENTS		
2768	050372	012702	000200		MOV	#SSR,R2		:SET UP EXPECTED TAPE STATUS ALERT		
2769	050376	020102			CMP	R1,R2		:ARE THEY EQUAL		
2770	050400	001406			BEQ	290\$:BR, IF OK		
2771	050402	005237	002214		INC	FATFLG		:ERROR COUNT		
2775	050406				ERRHRD	ERRNO,T32TSA,PKTSSR		:TSSR INCORRECT AFTER READ DATA		
	050406	104456							TRAP	C\$ERHRD
	050410	000643							.WORD	419
	050412	052311							.WORD	T32TSA
	050414	012126							.WORD	PKTSSR
2776	050416			290\$:	CKLOOP			:LOOP IF SELECTED		

	050416	104406							TRAP	C\$CLP1
2777	050420	017701	132472		MOV	@FREE,R1		;GET DATA READ		
2778	050424	012702	000144		MOV	#100.,R2		;SHOULD BE 100		
2779	050430	020102			CMP	R1,R2		;CHECK'EM OUT		
2780	050432	001406			BEG	300\$;BR, IF OK		
2781	050434	005237	002214		INC	FATFLG		;ERROR COUNT		
2785	050440				ERRHRD	ERRNO,T32ECF,EXPREC		;ERASE COMMAND DIDN'T WORK		
	050440	104456							TRAP	C\$ERHRD
	050442	000644							.WORD	420
	050444	052455							.WORD	T32ECF
	050446	015554							.WORD	EXPREC
2786	050450		300\$:	CKLOOP				;LOOP IF SELECTED		
	050450	104406							TRAP	C\$CLP1
2787	050452		330\$:							
2788	050452			ENDSUB				;>>>>>>>>>>> END SUBTEST >>>>>>>>>>		
	050452							L10055:		
	050452	104403							TRAP	C\$ESUB
2789	050454	023727	002214	000017	CMP	FATFLG,#15.		;IS ERROR COUNT AT 25		
2790	050462	103402			BLO	999\$;BR, IF LESS THAN 25		
2791	050464	004737	017262		JSR	PC,CKDROP		;TRY TO DROP THE UNIT		
2792	050470		999\$:							

2841	050556	001356				BNE	10\$:BR, IF COUNTER NOT DONE	
2842	050560	005237	002214			INC	FATFLG		:ERROR COUNT	
2846	050564	010001				MOV	R0,R1		:CONTENTS OF TSSR REGISTER	
2847	050566					ERRDF	ERRNO,SFIERR,SFIMSG		:FATAL ERROR TSSR WAS NOT OK	
	050566	104455							TRAP	C\$ERDF
	050570	000645							.WORD	421
	050572	003646							.WORD	SFIERR
	050574	012114							.WORD	SFIMSG
2848	050576	013737	002174	051350	20\$:	MOV	UNITN,T32DSW		:SET UP UNIT (DRIVE) NUMBER	
2849	050604	052737	000040	051350		BIS	#BIT5,T32DSW		:TURN ON HIGH SPEED TO SAVE TIME	
2850	050612	012704	051330			MOV	#T32PACKET,R4		:SUBROUTINE NEEDS PACKET ADDRESS	
2851	050616	004737	010742			JSR	PC,WRTCHR		:ISSUE WRITE CHARACTERISTICS	
2852	050622	103407				BCS	23\$:BR, IF COMMAND ISSUED OK	
2853	050624	005237	002214			INC	FATFLG		:ERROR COUNT	
2857	050630	010001				MOV	R0,R1		:SAVE CONTENTS OF TSSR	
2858	050632					ERRHRD	ERRNO,WRTMSG,SFIMSG		:WRITE CHARACTERISTICS FAILED	
	050632	104456							TRAP	C\$ERHRD
	050634	000646							.WORD	422
	050636	005052							.WORD	WRTMSG
	050640	012114							.WORD	SFIMSG
2859	050642				23\$:	CKLOOP			:LOOP IF SELECTED	
	050642	104406							TRAP	C\$CLP1
2860	050644	004737	011074			JSR	PC,REWIND		:CALL TAPE REWIND COMMAND	
2861	050650	103411				BCS	30\$:BR, IF NO PROBLEM	
2862	050652	016501	000002			MOV	TSSR(R5),R1		:GET TSSR CONTENTS	
2863	050656	010004				MOV	R0,R4		:GET PACKET ADDRESS	
2864	050660	005237	002214			INC	FATFLG		:ERROR COUNT	
2868	050664					ERRHRD	ERRNO,T32RWN,PKTSSR		:REWIND NOT ACCEPTED	
	050664	104456							TRAP	C\$ERHRD
	050666	000647							.WORD	423
	050670	051700							.WORD	T32RWN
	050672	012126							.WORD	PKTSSR
2869	050674				30\$:	CKLOOP			:LOOP IF SELECTED	
	050674	104406							TRAP	C\$CLP1
2870	050676	013701	051360			MOV	T32BFR+6,R1		:PICK UP XSTO	
2871	050702	010102				MOV	R1,R2		:SET UP EXPECTED	
2872	050704	052702	000002			BIS	#BIT1,R2		:SET BOT BIT IN EXPECTED	
2873	050710	020102				CMP	R1,R2		:DOES EXP = REC'D	
2874	050712	001406				BEQ	40\$:BR, IF EQUAL (OK)	
2875	050714	005237	002214			INC	FATFLG		:ERROR COUNT	
2879	050720					ERRHRD	ERRNO,T32BOT,EXPREC		:TAPE NOT AT BOT AFTER REWIND	
	050720	104456							TRAP	C\$ERHRD
	050722	000650							.WORD	424
	050724	051516							.WORD	T32BOT
	050726	015554							.WORD	EXPREC
2880	050730				40\$:	CKLOOP			:LOOP IF SELECTED	
	050730	104406							TRAP	C\$CLP1
2881	050732	012737	140411	051450	65\$:	MOV	#140411,T32PK3		:ERASE DATA,CVC=1,ACK COMMAND	
2882	050740	012704	051450			MOV	#T32PK3,R4		:SET UP R4 WITH PACKET ADDRESS	
2883	050744	010337	051456			MOV	R3,T32SZ		:SET UP RECORD SIZE IN PACKET	
2884	050750	010465	000000			MOV	R4,T32DB(R5)		:ISSUE COMMAND	
2885	050754	004737	016330			JSR	PC,WAITF		:WAIT FOR SSR TO SET	
2886	050760	016501	000002			MOV	TSSR(R5),R1		:GET TSSR CONTENTS	
2887	050764	012702	000200			MOV	#SSR,R2		:SET UP EXPECTED	
2888	050770	020102				CMP	R1,R2		:ARE THEY EQUAL	
2889	050772	001757				BEQ	65\$:BR, IF OK	
2890	050774	032701	000004			BIT	#BIT2,R1		:CHECK FOR TAPE STATUS ALERT	

2891	051000	001006		BNE	80\$;BR, IF TAPE STATUS ALERT SET	
2892	051002	005237	002214	INC	FATFLG		;ERROR COUNT	
2896	051006			ERRHRD	ERRNO,T32WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA	
	051006	104456					TRAP	C\$ERHRD
	051010	000651					.WORD	425
	051012	052536					.WORD	T32WDC
	051014	012126					.WORD	PKTSSR
2897	051016			80\$: CKLOOP			;LOOP IF SELECTED	
	051016	104406					TRAP	C\$CLP1
2898	051020	013701	051360	MOV	T32BFR+6,R1		;PICK UP XST0	
2899	051024	010102		MOV	R1,R2		;SET UP EXPECTED	
2900	051026	052702	000001	BIS	#BIT0,R2		;SET EOT BIT IN EXPECTED	
2901	051032	020102		CMP	R1,R2		;DOES EXP = REC'D	
2902	051034	001406		BEQ	240\$;BR, IF EQUAL (OK)	
2903	051036	005237	002214	INC	FATFLG		;ERROR COUNT	
2907	051042			ERRHRD	ERRNO,T32EOT,EXPREC		;TAPE NOT AT EOT AFTER ERASE COMMANDS	
	051042	104456					TRAP	C\$ERHRD
	051044	000652					.WORD	426
	051046	051611					.WORD	T32EOT
	051050	015554					.WORD	EXPREC
2908	051052			240\$: CKLOOP			;LOOP IF SELECTED	
	051052	104406					TRAP	C\$CLP1
2909	051054	012703	051460	MOV	#T32CMD,R3		;STARTING RECORD SIZE	
2910	051060	013737	003116	MOV	FREE,T32RB		;STARTING READ BUFFER ADDRESS	
2911	051066	011337	051450	MOV	(R3),T32PK3		;READ DATA,ACK COMMAND	
2912	051072	012704	051450	MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS	
2913	051076	012700	177777	MOV	#177777,R0		;SET PATTERN IN CORRECT REGISTER	
2914	051102	004737	017502	JSR	PC,FILLMEM		;FILL MEMORY WITH ALL ONES	
2915	051106	012737	000144	MOV	#100.,T32SZ		;SET UP RECORD SIZE IN PACKET	
2916	051114	010465	000000	MOV	R4,T32DB(R5)		;ISSUE COMMAND	
2917	051120	012737	000062	MOV	#50.,T32DLY		;SET UP DELAY COUNTER	
2918	051126	004737	016330	JSR	PC,WAITF		;WAIT FOR SSR TO SET	
2919	051132	016501	000002	MOV	TSSR(R5),R1		;GET TSSR CONTENTS	
2920	051136	012702	100214	MOV	#SSR!SC!BIT2!BIT3,R2		;SET UP EXPECTED	
2921	051142	020102		CMP	R1,R2		;ARE THEY EQUAL	
2922	051144	001425		BEQ	280\$;BR, IF OK	
2923	051146			DELAY	250		;DELAY FOR SSR TO BE SET	
	051146	012727	000250				MOV	#250,(PC)+
	051152	000000					.WORD	0
	051154	013727	002116				MOV	L\$DLY,(PC)+
	051160	000000					.WORD	0
	051162	005367	177772				DEC	-6(PC)
	051166	001375					BNE	-4
	051170	005367	177756				DEC	-22(PC)
	051174	001367					BNE	-20
2924	051176	005337	051514	DEC	T32DLY		;COUNT DELAY ROUTINE DOWN	
2925	051202	001351		BNE	270\$;BR, IF DELAY HAS NOT ENDED	
2926	051204	005237	002214	INC	FATFLG		;ERROR COUNT	
2930	051210			ERRHRD	ERRNO,T32ECF,PKTSSR		;TSSR INCORRECT AFTER READ DATA	
	051210	104456					TRAP	C\$ERHRD
	051212	000653					.WORD	427
	051214	052455					.WORD	T32ECF
	051216	012126					.WORD	PKTSSR
2931	051220			280\$: CKLOOP			;LOOP IF SELECTED	
	051220	104406					TRAP	C\$CLP1
2932	051222	013701	051366	MOV	T32BFR+14,R1		;PICK UP XST3	
2933	051226	010102		MOV	R1,R2		;SET UP EXPECTED	

2934	051230	052702	000100		BIS	#BIT6,R2	;SET OPI BIT IN EXPECTED		
2935	051234	020102			CMP	R1,R2	;IS OPI BIT SET		
2936	051236	001406			BEQ	290\$;BR, IF BIT IS SET		
2937	051240	005237	002214		INC	FATFLG	;ERROR COUNT		
2941	051244				ERRHRD	ERRNO,T32OPI,EXPREC	;OPI BIT NOT SET		
	051244	104456						TRAP	C\$ERHRD
	051246	000654						.WORD	428
	051250	052603						.WORD	T32OPI
	051252	015554						.WORD	EXPREC
2942	051254			290\$:	CKLOOP		;LOOP IF SELECTED		
	051254	104406						TRAP	C\$CLP1
2943	051256	005723			TST	(R3)+	;BUMP COMMAND POINTER		
2944	051260	021327	177777		CMP	(R3),#177777	;AT END OF TABLE YET		
2945	051264	001300			BNE	265\$;BR, KEEP TRYING COMMANDS		
2946	051266				ENDSUB		>>>>>>>>>> END SUBTEST >>>>>>>>>>		
	051266						L10056:		
	051266	104403						TRAP	C\$ESUB
2947	051270	023727	002214	000017	CMP	FATFLG,#15.	;IS ERROR COUNT AT 25		
2948	051276	103402			BLO	999\$;BR, IF LESS THAN 25		
2949	051300	004737	017262		JSR	PC,CKDROP	;TRY TO DROP THE UNIT		
2950	051304			999\$:					

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 25-MAY-82 08:43 PAGE 108
TEST 4: ERASE AND OPERATION INCOMPLETE

K 15

SEQ 0192

2952

2953

2954

2955 051304 004737 016536

2956 051310 103002

2957 051312 000137 047020

2958 051316

051316 104432

051320 001524

:

163\$:

JSR

BCC

JMP

EXIT

PC,TSTLOOP

163\$

T32LOOP

TST

;DO WE NEED TO ITERATE TEST

;BR, IF NO LOOP REQUIRED

;EXECUTE AGAIN

;ALL DONE THIS TEST

TRAP
.WORD

C\$EXIT
L10053-.

2960			;		
2961			;	LOCAL STORAGE FOR THIS TEST	
2962			;		
2964		051330	;		
2966	051330		;		
2967	051330	100004	T32PACKET:		;
2968	051332	051340		.WORD 100004	;
2969	051334	000000		.WORD T32DATA	;
2970	051336	000012		.WORD 0	;
2971	051340			.WORD 10.	;
2972	051340	051352	T32DATA:		;
2973	051342	000000		.WORD T32BFR	;
2974	051344	000024		.WORD 0	;
2975	051346	000000		.WORD 20.	;
2976	051350	000000		.WORD 0	;
2977	051352		T32DSW:	.WORD 0	;
2978			T32BFR:	.BLKW 25.	;
2979			;		;
2980			;	WRITE SUBSYSTEM MEMORY COMMAND PACKET	
2982		051440	;		
2984	051440		;		
2985	051440	100006	T32PK2:		;
2986	051442	000000		.WORD 100006	;
2987	051444	000000		.WORD 0	;
2988	051446	000006		.WORD 0	;
2989				.WORD 6.	;
2993	051450		T32PK3:		;
2994	051450	100005		.WORD 100005	;
2995	051452		T32RB:		;
2996	051452	003116	T32WB:	.WORD FREE	;
2997	051454	000000		.WORD 0	;
2998	051456	000000	T32SZ:	.WORD 0	;
2999				.EVEN	;
3000			;		
3001			;		
3002			;		
3003			;		
3004			;		
3005			;		
3006			;	.EVEN	
3007			;	TAPE MOTION PACKET COMMAND VALUES	
3008	051460		T32CMD:		;
3009	051460	140410		.WORD 140410	;
3010	051462	141410		.WORD 141410	;
3011	051464	140401		.WORD 140401	;
3012	051466	141001		.WORD 141001	;
3013	051470	161401		.WORD 161401	;
3014	051472	161001		.WORD 161001	;
3015	051474	141401		.WORD 141401	;
3016	051476	140001		.WORD 140001	;
3017	051500	141410		.WORD 141410	;
3018	051502	141010		.WORD 141010	;
3019	051504	141005		.WORD 141005	;
3020	051506	177777		.WORD 177777	;
3021					;
3022			;		
3023	051510	000000	T32CNT:	.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)

;

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

;

TAPE TIMER COUNTER STORAGE AREA

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 25-MAY-82 08:43 PAGE 109-1
TEST 4: ERASE AND OPERATION INCOMPLETE

M 15

3024 051512 000000
3025 051514 000000
3026

T32CNU: .WORD 0
T32DLY: .WORD 0

;TAPE TIMER COUNTER STORAGE AREA
;DELAY COUNTER

SEQ 0194

```

3028
3029
3030      ;+
3031      ;LOCAL TEXT MESSAGES FOR TEST
3032      ;--
3033
3034 051516      124      141      160  T32BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
3035 051611      124      141      160  T32EOT: .ASCIZ 'Tape Status Alert During Erase To EOT, But EOT Not Set'
3036 051700      122      145      167  T32RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
3037 051747      124      123      123  T32AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
3038 052016      124      123      123  T32ERA: .ASCIZ 'TSSR Not Correct After ERASE Command'
3039 052063      124      123      102  T32BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
3040 052136      122      105      101  T32RIB: .ASCIZ 'READ REVERSE, After ERASE From BOT, Failed To Set RIB In XST3'
3041 052234      124      123      123  T32SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
3042 052311      124      123      123  T32TSA: .ASCIZ 'TSSR Not Correct After READ REVERSE Into BOT'
3043 052366      102      117      124  T32BOE: .ASCIZ 'BOT (XST0) Still Set After Erase From Tape's BOT Marker'
3044 052455      105      122      101  T32ECF: .ASCIZ 'ERASE Failed To Clear Tape (Erase) Tape Properly'
3045
3046 052536      124      123      123  T32WDC: .ASCIZ 'TSSR Not Correct After ERASE Command'
3047 052603      117      120      111  T32OPI: .ASCIZ 'OPI Bit (XST3) Failed To Set'
3048 052640      105      162      141  TST32ID: .ASCIZ 'Erase And Operation Incomplete'
3049
3050      ;+
3051      ;
3052      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3053      ;WRITE SUBSYSTEM MEMORY COMMAND
3054      ;
3055      ;--
3056
3057 052700      T32REST:
3058 052700      SAVREG
3059 052704      012701      051330      MOV      #T32PACKET,R1      ;SAVE THE REGISTERS
3060 052710      012721      100004      MOV      #100004,(R1)+      ;START OF THE PACKET
3061 052714      012721      051340      MOV      #T32DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK.
3062 052720      005021      CLR      (R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
3063 052722      012721      000012      MOV      #10,(R1)+      ;EXTENDED ADDRESS
3064 052726      012721      051352      MOV      #T32BFR,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
3065 052732      005021      CLR      (R1)+      ;ADDRESS OF MESSAGE BUFFER
3066 052734      012721      000024      MOV      #20,(R1)+      ;LENGTH OF MESSAGE BUFFER
3067 052740      005021      CLR      (R1)+
3068 052742      012711      000000      MOV      #0,(R1)      ;SELECT DRIVE ZERO
3069 052746      012702      000030      MOV      #24,R2      ;NUMBER OF LOCATIONS TO BE CLEARED
3070 052752      012762      177777      051352      64$      MOV      #177777,T32BFR(R2)      ;ALL ONES TO MESSAGE BUFFER
3071 052760      005742      TST      -(R2)      ;NEXT LOCATION
3072 052762      022702      000000      CMP      #0,R2      ;AT END OF LOOP YET
3073 052766      001371      BNE      64$      ;KEEP GOING UNTIL DONE
3074 052770      000207      RTS      PC      ;RETURN
3075
3076
3077 052772      T32PT2:
3078 052772      SAVREG
3079 052776      012701      051440      MOV      #T32PK2,R1      ;SAVE THE REGISTERS
3080 053002      012721      100006      MOV      #100006,(R1)+      ;START OF THE PACKET
3081 053006      005021      CLR      (R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK.
3082 053010      005021      CLR      (R1)+      ;ADDRESS OF DATA BLOCK
3083 053012      012721      000006      MOV      #6,(R1)+      ;EXTENDED ADDRESS
3084 053016      005021      CLR      (R1)+      ;SIZE OF DATA BLOCK IN BYTES

```


3085 053020 000207
3086 053022
3087 053022
3088 053026 012701 051450
3089 053032 005021
3090 053034 005021
3091 053036 005021
3092 053040 005011
3093 053042 000207
3094 053044
053044
053044 104401

T32RT3: RTS PC
SAVREG
MOV #T32PK3,R1
CLR (R1)+
CLR (R1)+
CLR (R1)+
CLR (R1)
RTS PC
ENDTST

;RETURN
;SAVE REGISTERS
;SET UP POINTER ADDRESS
;COMMAND SPACE
;ADDRESS OF DATA BLOCK
;EXTENDED ADDRESS
;SIZE OF DATA TRANSFER BLOCK
;RETURN

L10053: TRAP CSETST

3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153

.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.
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.

```

3154      :
3155      :      11. The controller is initialized to clear the special test conditions
3156      :      previously set up.
3157      :
3158      :
3159      :
3160      :
3161      :      BGNTST
3162      :      053046      053046      T5::
3163      :      053046      012737      006413      002172      MOV      #EPRT2,EPRTSW      ;SECONDARY ERROR MESSAGE
3164      :      053054      012700      055645      MOV      #TST33ID,RO      ;ASCII MESSAGE TO IDENTIFY TEST
3165      :      053060      004737      016570      JSR      PC,TSTSETUP      ;DO INITIAL TEST SETUP
3166      :      053064      012737      000005      002210      MOV      #5,LOOPCNT      ;PERFORM 5 ITERATIONS
3167      :      053072      005037      054716      CLR      T33CNT      ;CLEAR TAPE RECORD COUNTER
3168      :
3169      :
3170      :
3171      :
3172      :
3173      :      T33LOOP:

```

[illegible]

3217	053316	010102		MOV	R1,R2		;SET UP EXPECTED	
3218	053320	052702	000002	BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED	
3219	053324	020102		CMP	R1,R2		;DOES EXP = REC'D	
3220	053326	001406		BEQ	40\$;BR, IF EQUAL (OK)	
3221	053330	005237	002214	INC	FATFLG		;ERROR COUNT	
3225	053334			ERRHRD	ERRNO,T33BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND	
	053334	104456					TRAP	C\$ERHRD
	053336	000770					.WORD	504
	053340	055325					.WORD	T33BOT
	053342	015554					.WORD	EXPREC
3226	053344			40\$:	CKLOOP		;LOOP IF SELECTED	
	053344	104406					TRAP	C\$CLP1

3228	053346	005737	002220		42\$:	TST	EXTFEA		:CHECK FOR EXTENDED FEATURES SW SWITCH
3229	053352	001025				BNE	55\$:BR IF SWITCH IS ON
3230	053354	112737	000200	054701		MOVB	#200,T33BS1		:WRITE MISCELLANEOUS CONT/READ STATUS
3231	053362	112737	000010	054700		MOVB	#10,T33BS0		:FUNC. SEL. BIT (TURN ON EXTFEA SWITCH)
3232	053370	012704	054660			MOV	#T33PK2,R4		:WRITE SUBSYS MEM PACKET
3233	053374	010465	000000			MOV	R4,TSDB(R5)		:ISSUE COMMAND
3234	053400	004737	016416			JSR	PC,CHKTSSR		:WAIT FOR SSR
3235	053404	103407				BCS	50\$:BR, IF NO ERROR
3236	053406	010001				MOV	R0,R1		:ERROR, SAVE TSSR
3237	053410	005237	002214			INC	FATFLG		:ERROR COUNT
3241	053414					ERRHRD	ERRNO,T33SSR,PKTSSR		:TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
	053414	104456						TRAP	C\$ERHRD
	053416	000771						.WORD	505
	053420	055241						.WORD	T33SSR
	053422	012126						.WORD	PKTSSR
3242	053424				50\$:	CKLOOP			:LOOP IF SELECTED
	053424	104406						TRAP	C\$CLP1
3243	053426	005737	002222		55\$:	TST	BENBSW		:CHECK FOR BUFFER ENABLED
3244	053432	001426				BEQ	70\$:BR, IF BUFFERING NOT ENABLED
3245	053434	013737	002174	054570		MOV	UNITN,T33DSW		:SET UP UNIT NUMBER
3246	053442	042737	000020	054570		BIC	#BIT4,T33DSW		:BUFFER DISABLE
3247	053450	052737	000010	054570		BIS	#BIT3,T33DSW		:BUFFER DISABLE SEND 01 TO BITS 4 AND 3
3248	053456	012704	054550			MOV	#T33PACKET,R4		:SUBROUTINE NEEDS PACKET ADDRESS
3249	053462	004737	010742			JSR	PC,WRTCHR		:ISSUE WRITE CHARACTERISTICS
3250	053466	103407				BCS	60\$:BR, IF COMMAND ISSUED OK
3251	053470	005237	002214			INC	FATFLG		:ERROR COUNT
3255	053474	010001				MOV	R0,R1		:SAVE CONTENTS OF TSSR
3256	053476					ERRHRD	ERRNO,WRTMSG,SFMSG		:WRITE CHARACTERISTICS FAILED
	053476	104456						TRAP	C\$ERHRD
	053500	000772						.WORD	506
	053502	005052						.WORD	WRTMSG
	053504	012114						.WORD	SFMSG
3257	053506				60\$:	CKLOOP			:LOOP IF SELECTED
	053506	104406						TRAP	C\$CLP1
3258	053510				70\$:				
3259	053510	112737	000100	054701		MOVB	#100,T33BS1		:WRITE MISCELLANEOUS CONT/READ STATUS
3260	053516	112737	000011	054700		MOVB	#11,T33BS0		:FUNC. SEL. BIT (SET WRONG PARITY)
3261	053524	012704	054660			MOV	#T33PK2,R4		:WRITE SUBSYS MEM PACKET
3262	053530	010465	000000			MOV	R4,TSDB(R5)		:ISSUE COMMAND
3263	053534	004737	016416			JSR	PC,CHKTSSR		:WAIT FOR SSR
3264	053540	103407				BCS	80\$:BR, IF NO ERROR
3265	053542	010001				MOV	R0,R1		:ERROR, SAVE TSSR
3266	053544	005237	002214			INC	FATFLG		:ERROR COUNT
3270	053550					ERRHRD	ERRNO,T33SSR,PKTSSR		:TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
	053550	104456						TRAP	C\$ERHRD
	053552	000773						.WORD	507
	053554	055241						.WORD	T33SSR
	053556	012126						.WORD	PKTSSR
3271	053560				80\$:	CKLOOP			:LOOP IF SELECTED
	053560	104406						TRAP	C\$CLP1
3272	053562	012703	000026			MOV	#22.,R3		:NUMBER OF RECORDS TO BE WRITTEN
3273	053566	013737	003116	054672		MOV	FREE,T33WB		:STARTING WRITE BUFFER ADDRESS
3274	053574	005037	054720			CLR	T33CNU		:MAKE SURE ITS CLEAR
3275	053600	012737	140005	054670	110\$:	MOV	#140005,T33PK3		:WRITE DATA,ACK,CVC=1 COMMAND
3276	053606	012704	054670			MOV	#T33PK3,R4		:SET UP R4 WITH PACKET ADDRESS
3277	053612	012737	000024	054676		MOV	#20.,T33SZ		:SET UP RECORD SIZE IN PACKET
3278	053620	013777	054720	127270		MOV	T33CNU,@FREE		:MEMORY FILLED WITH DATA IN RECORD

3279	053626	005237	054720	INC	T33CNU	:READY FOR NEXT RECORD	
3280	053632	010465	000000	MOV	R4,TSD8(R5)	:ISSUE COMMAND	
3281	053636	004737	016330	JSR	PC,WAITF	:WAIT FOR SSR TO SET	
3282	053642	016501	000002	MOV	TSSR(R5),R1	:GET TSSR CONTENTS	
3283	053646	012702	100210	MOV	#SSR!SC!BIT3,R2	:SET UP EXPECTED	
3284	053652	020102		CMP	R1,R2	:ARE THEY EQUAL	
3285	053654	001406		BEQ	120\$:BR, IF OK	
3286	053656	005237	002214	INC	FATFLG	:ERROR COUNT	
3290	053662			ERRHRD	ERRNO,T33WPW,PKTSSR	:TSSR INCORRECT AFTER WRITE DATA	
	053662	104456				TRAP	C\$ERHRD
	053664	000774				.WORD	508
	053666	055002				.WORD	T33WPW
	053670	012126				.WORD	PKTSSR
3291	053672			120\$: CKLOOP		:LOOP IF SELECTED	
	053672	104406				TRAP	C\$CLP1
3292	053674	013701	054602	MOV	T33BFR+10,R1	:PICK UP XST1	
3293	053700	010102		MOV	R1,R2	:SET UP EXPECTED	
3294	053702	052702	000002	BIS	#BIT1,R2	:SET UNC BIT IN EXPECTED	
3295	053706	020102		CMP	R1,R2	:DOES EXP = REC'D	
3296	053710	001406		BEQ	130\$:BR, IF EQUAL (OK)	
3297	053712	005237	002214	INC	FATFLG	:ERROR COUNT	
3301	053716			ERRHRD	ERRNO,T33UNC,EXPREC	:TAPE NOT AT BOT AFTER REWIND	
	053716	104456				TRAP	C\$ERHRD
	053720	000775				.WORD	509
	053722	055062				.WORD	T33UNC
	053724	015554				.WORD	EXPREC
3302	053726			130\$: CKLOOP		:LOOP IF SELECTED	
	053726	104406				TRAP	C\$CLP1
3303	053730	005303		DEC	R3	:DEC RECORD COUNTER	
3304	053732	001322		BNE	110\$:BR, IF MORE RECORDS TO WRITE	
3305	053734	004737	011074	JSR	PC,REWIND	:CALL TAPE REWIND COMMAND	
3306	053740	103411		BCS	140\$:BR, IF NO PROBLEM	
3307	053742	016501	000002	MOV	TSSR(R5),R1	:GET TSSR CONTENTS	
3308	053746	010004		MOV	R0,R4	:GET PACKET ADDRESS	
3309	053750	005237	002214	INC	FATFLG	:ERROR COUNT	
3313	053754			ERRHRD	ERRNO,T33RWN,PKTSSR	:REWIND NOT ACCEPTED	
	053754	104456				TRAP	C\$ERHRD
	053756	000776				.WORD	510
	053760	055420				.WORD	T33RWN
	053762	012126				.WORD	PKTSSR
3314	053764			140\$: CKLOOP		:LOOP IF SELECTED	
	053764	104406				TRAP	C\$CLP1
3315	053766	013701	054600	MOV	T33BFR+6,R1	:PICK UP XST0	
3316	053772	010102		MOV	R1,R2	:SET UP EXPECTED	
3317	053774	052702	000002	BIS	#BIT1,R2	:SET BOT BIT IN EXPECTED	
3318	054000	020102		CMP	R1,R2	:DOES EXP = REC'D	
3319	054002	001406		BEQ	150\$:BR, IF EQUAL (OK)	
3320	054004	005237	002214	INC	FATFLG	:ERROR COUNT	
3324	054010			ERRHRD	ERRNO,T33BOT,EXPREC	:TAPE NOT AT BOT AFTER REWIND	
	054010	104456				TRAP	C\$ERHRD
	054012	000777				.WORD	511
	054014	055325				.WORD	T33BOT
	054016	015554				.WORD	EXPREC
3325	054020			150\$: CKLOOP		:LOOP IF SELECTED	
	054020	104406				TRAP	C\$CLP1
3326	054022	005037	054720	CLR	T33CNU	:CLEAR DATA VALUE IN RECORD	
3327	054026	012703	000024	MOV	#20.,R3	:RECORD SIZE	

3328	054032	013737	003116	054672	155\$:	MOV	FREE,T33RB		:STARTING WRITE BUFFER ADDRESS
3329	054040	012737	140001	054670		MOV	#140001,T33PK3		:READ DATA,CVC=1,ACK COMMAND
3330	054046	012704	054670			MOV	#T33PK3,R4		:SET UP R4 WITH PACKET ADDRESS
3331	054052	012737	000024	054676		MOV	#20.,T33SZ		:SET UP RECORD SIZE IN PACKET
3332	054060	010465	000000			MOV	R4,T33DB(R5)		:ISSUE COMMAND
3333	054064	004737	016330			JSR	PC,WAITF		:WAIT FOR SSR TO SET
3334	054070	016501	000002			MOV	T33R(R5),R1		:GET T33R CONTENTS
3335	054074	012702	100210			MOV	#SSR!SC!BIT3 R2		:SET UP EXPECTED
3336	054100	020102				CMP	R1,R2		:ARE THEY EQUAL
3337	054102	001406				BEQ	160\$:BR, IF OK
3338	054104	005237	002214			INC	FATFLG		:ERROR COUNT
3342	054110					ERRHRD	ERRNO,T33WDC,PKTSSR		:T33R INCORRECT AFTER WRITE DATA
	054110	104456						TRAP	C\$ERHRD
	054112	001000						.WORD	512
	054114	055467						.WORD	T33WDC
	054116	012126						.WORD	PKTSSR
3343	054120				160\$:	CKLOOP			:LOOP IF SELECTED
	054120	104406						TRAP	C\$CLP1
3344	054122	013701	054602			MOV	T33BFR+10,R1		:PICK UP XST1
3345	054126	010102				MOV	R1,R2		:SET UP EXPECTED
3346	054130	052702	000002			BIS	#BIT1,R2		:SET UNC BIT IN EXPECTED
3347	054134	020102				CMP	R1,R2		:DOES EXP = REC'D
3348	054136	001406				BEQ	170\$:BR, IF EQUAL (OK)
3349	054140	005237	002214			INC	FATFLG		:ERROR COUNT
3353	054144					ERRHRD	ERRNO,T33UND,EXPREC		:UNC BIT NOT SET AFTER READ CMD.
	054144	104456						TRAP	C\$ERHRD
	054146	001001						.WORD	513
	054150	055152						.WORD	T33UND
	054152	015554						.WORD	EXPREC
3354	054154				170\$:	CKLOOP			:LOOP IF SELECTED
	054154	104406						TRAP	C\$CLP1
3355	054156	013701	054602			MOV	T33BFR+10,R1		:PICK UP XST1
3356	054162	010102				MOV	R1,R2		:SET UP EXPECTED
3357	054164	052702	000400			BIS	#BIT8,R2		:SET RBP BIT IN EXPECTED
3358	054170	020102				CMP	R1,R2		:DOES EXP = REC'D
3359	054172	001406				BEQ	180\$:BR, IF EQUAL (OK)
3360	054174	005237	002214			INC	FATFLG		:ERROR COUNT
3364	054200					ERRHRD	ERRNO,T33RBP,EXPREC		:READ BUS PARITY ERROR BIT NOT SET
	054200	104456						TRAP	C\$ERHRD
	054202	001002						.WORD	514
	054204	054724						.WORD	T33RBP
	054206	015554						.WORD	EXPREC
3365	054210				180\$:	CKLOOP			:LOOP IF SELECTED
	054210	104406						TRAP	C\$CLP1
3366	054212	017701	126700			MOV	@FREE,R1		:GET DATA READ
3367	054216	013702	054720			MOV	T33CNU,R2		:GET PATTERN
3368	054222	020102				CMP	R1,R2		:ARE THEY EQUAL
3369	054224	001406				BEQ	182\$:BR, IF OK
3370	054226	005237	002214			INC	FATFLG		:ERROR COUNT
3374	054232					ERRHRD	ERRNO,T33DTA,EXPREC		:DATA NOT CORRECT
	054232	104456						TRAP	C\$ERHRD
	054234	001003						.WORD	515
	054236	055550						.WORD	T33DTA
	054240	015554						.WORD	EXPREC
3375	054242				182\$:	CKLOOP			:LOOP IF SELECTED
	054242	104406						TRAP	C\$CLP1
3376	054244	013737	003116	054672		MOV	FREE,T33WB		:STARTING WRITE BUFFER ADDRESS

3377	054252	012737	140401	054670	195\$:	MOV	#140401,T33PK3	;READ REVERSE DATA RETRY,ACK COMMAND
3378	054260	012704	054670			MOV	#T33PK3,R4	;SET UP R4 WITH PACKET ADDRESS
3379	054264	012737	000024	054676		MOV	#20,T33SZ	;SET UP RECORD SIZE IN PACKET
3380	054272	010465	000000			MOV	R4,TSD8(R5)	;ISSUE COMMAND
3381	054276	004737	016330			JSR	PC,WAITF	;WAIT FOR SSR TO SET
3382	054302	016501	000002			MOV	TSSR(R5),R1	;GET TSSR CONTENTS
3383	054306	012702	100210			MOV	#SC!SSR!BIT3,R2	;SET UP EXPECTED
3384	054312	020102				CMP	R1,R2	;ARE THEY EQUAL
3385	054314	001406				BEQ	190\$;BR, IF OK
3386	054316	005237	002214			INC	FATFLG	;ERROR COUNT
3390	054322					ERRHRD	ERRNO,T33WDC,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA
	054322	104456						TRAP C\$ERHRD
	054324	001004						.WORD 516
	054326	055467						.WORD T33WDC
	054330	012126						.WORD PKTSSR
3391	054332				190\$:	CKLOOP		;LOOP IF SELECTED
	054332	104406						TRAP C\$CLP1
3392	054334	013701	054602			MOV	T33BFR+10,R1	;PICK UP XST1
3393	054340	010102				MOV	R1,R2	;SET UP EXPECTED
3394	054342	052702	000002			BIS	#BIT1,R2	;SET UNC BIT IN EXPECTED
3395	054346	020102				CMP	R1,R2	;DOES EXP = REC'D
3396	054350	001406				BEQ	200\$;BR, IF EQUAL (OK)
3397	054352	005237	002214			INC	FATFLG	;ERROR COUNT
3401	054356					ERRHRD	ERRNO,T33UND,EXPREC	;TAPF NOT AT BOT AFTER REWIND
	054356	104456						TRAP C\$ERHRD
	054356	001005						.WORD 517
	054362	055152						.WORD T33UND
	054364	015554						.WORD EXPREC
3402	054366				200\$:	CKLOOP		;LOOP IF SELECTED
	054366	104406						TRAP C\$CLP1
3403	054370	013701	054602			MOV	T33BFR+10,R1	;PICK UP XST0
3404	054374	010102				MOV	R1,R2	;SET UP EXPECTED
3405	054376	052702	000400			BIS	#BIT8,R2	;SET RBP BIT IN EXPECTED
3406	054402	020102				CMP	R1,R2	;DOES EXP = REC'D
3407	054404	001406				BEQ	210\$;BR, IF EQUAL (OK)
3408	054406	005237	002214			INC	FATFLG	;ERROR COUNT
3412	054412					ERRHRD	ERRNO,T33RBP,EXPREC	;READ BUS PARITY ERROR BIT NOT SET
	054412	104456						TRAP C\$ERHRD
	054414	001006						.WORD 518
	054416	054724						.WORD T33RBP
	054420	015554						.WORD EXPREC
3413	054422				210\$:	CKLOOP		;LOOP IF SELECTED
	054422	104406						TRAP C\$CLP1
3414	054424	017701	126466			MOV	@FREE,R1	;GET DATA READ
3415	054430	013702	054720			MOV	T33CNU,R2	;GET PATTERN
3416	054434	020102				CMP	R1,R2	;ARE THEY EQUAL
3417	054436	001406				BEQ	215\$;BR, IF OK
3418	054440	005237	002214			INC	FATFLG	;ERROR COUNT
3422	054444					ERRHRD	ERRNO,T33DTA,EXPREC	;DATA NOT CORRECT
	054444	104456						TRAP C\$ERHRD
	054446	001007						.WORD 519
	054450	055550						.WORD T33DTA
	054452	015554						.WORD EXPREC
3423	054454				215\$:	CKLOOP		;LOOP IF SELECTED
	054454	104406						TRAP C\$CLP1
3424	054456	010302				MOV	R3,R2	;SAVE R3 FOR A MOMENT
3425	054460	012703	000001			MOV	#1,R3	;SPACE FORWARD ONE RECORD

220\$:

ENDSUB

```
;<<<<<<<<<< END SUBTEST >>>>>>>>>>
      L10060:
```

9995:

```
JSR      PC,TSTLOOP
BCC      230$
JMP      T33LOOP
EXIT     TST
```

```

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

```

230S:

```
JSR      PC,TSTLOOP
BCC      230$
JMP      T33LOOP
EXIT     TST
```

```

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

```

```
TRAP      C$EXIT
.WORD     L10057-.
```

3446			;		
3447			;	LOCAL STORAGE FOR THIS TEST	
3448			;		
3450		054550	;		
3452	054550		T33PACKET:	;<.+10>B177770	
3453	054550	100004		.WORD 100004	;COMMAND PACKET FOR TEST
3454	054552	054560		.WORD T33DATA	;WRITE CHARACTERISTICS COMMAND, WITH , ACK
3455	054554	000000		.WORD 0	;ADDRESS OF CHARACTERISTICS BLOCK
3456	054556	000012		.WORD 10.	;STARTING VALUE OF BLOCK SIZE
3457	054560		T33DATA:		;CHARACTERISTICS DATA BLOCK
3458	054560	054572		.WORD T33BFR	;ADDRESS OF MESSAGE BUFFER
3459	054562	000000		.WORD 0	
3460	054564	000024		.WORD 20.	;LENGTH OF MESSAGE BUFFER
3461	054566	000000		.WORD 0	
3462	054570	000000	T33DSW:	.WORD 0	;SELECT DRIVE 0
3463	054572		T33BFR:	.BLKW 25.	;MESSAGE BUFFER
3464			;		
3465			;	WRITE SUBSYSTEM MEMORY COMMAND PACKET	
3466			;		
3468		054660	;		
3470	054660		T33PK2:	;<.+10>B177770	
3471	054660	100006		.WORD 100006	;WRITE SUB SYS MEM COMMAND, AND ACK
3472	054662	054700		.WORD T33BF2	;ADDRESS OF SELECT BLOCK DATA
3473	054664	000000		.WORD 0	
3474	054666	000006		.WORD 6.	;SIZE OF DATA PACKET
3475					
3479	054670		T33PK3:		
3480	054670	100005		.WORD 100005	;REREAD COMMAND, AND ACK
3481	054672		T33RB:		
3482	054672	003116	T33WB:	.WORD FREE	;ADDRESS OF WRITE BUFFER
3483	054674	000000		.WORD 0	
3484	054676	000000	T33SZ:	.WORD 0	;SIZE OF BUFFER (EXTENT)
3485				.EVEN	
3486			;		
3487			;		
3488			;		
3489	054700		T33BF2:		
3490	054700	010	T33BS0:	.BYTE 10	;BSELO AREA
3491	054701	200	T33BS1:	.BYTE 200	;BSEL1 AREA
3492	054702	000000	T33S2:	.WORD 0	;SEL 2 AREA
3493	054704	000000	T33S3:	.WORD 0	;DATA AREA
3494			;		
3495			;		
3496			;		
3497			;		
3498			;		
3499	054706	100205	T33RN:	.WORD 100205	;REREAD DATA (NEXT)
3500	054710	100605	T33WDR:	.WORD 100605	;REREAD DATA RETRY
3501	054712	102205	T33CON:	.WORD 102205	;WRITE CONTINUOUS
3502	054714	177777		.WORD 177777	;END OF DATA
3503					
3504			;		
3505	054716	000000	T33CNT:	.WORD 0	;TAPE TIMER COUNTER STORAGE AREA
3506	054720	000000	T33CNU:	.WORD 0	;TAPE TIMER COUNTER STORAGE AREA
3507	054722	000000	T33DLY:	.WORD 0	;DELAY COUNTER
3508					

```

3510
3511
3512      ;+
3513      ;LOCAL TEXT MESSAGES FOR TEST
3514      ;-
3515
3516 054724      122      145      141  T33RBP: .ASCIIZ 'Read Bus Parity Bit Not Set (XST1), Should Be'
3517 055002      124      123      123  T33WPW: .ASCIIZ 'TSSR Incorrect After Wrong Parity Write Command'
3518 055062      125      116      103  T33UNC: .ASCIIZ 'UNC Bit (XST1) Not Set After Wrong Parity WRITE Command'
3519 055152      125      116      103  T33UND: .ASCIIZ 'UNC Bit (XST1) Not Set After Wrong Parity READ Command'
3520 055241      127      122      111  T33SSR: .ASCIIZ 'WRITE MISCELLANEOUS CONT/READ COMMAND Not Accepted'
3521 055325      124      141      160  T33BOT: .ASCIIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XSTG)'
3522 055420      122      145      167  T33RWN: .ASCIIZ 'Rewind (POSITION) Command Not Accepted'
3523 055467      124      123      123  T33WDC: .ASCIIZ 'TSSR Not Correct After READ Wrong Parity Command'
3524 055550      104      141      164  T33DTA: .ASCIIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
3525 055645      104      141      164  T33ID:  .ASCIIZ 'Data Parity'
3526                                     .EVEN
3527
3528      ;+
3529      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3530      ;WRITE SUBSYSTEM MEMORY COMMAND
3531      ;-
3532
3533
3534 055662      T33REST:
3535 055662      SAVREG
3536 055666      012701 054550      MOV      #T33PACKET,R1      ;SAVE THE REGISTERS
3537 055672      012721 100004      MOV      #100004,(R1)+      ;START OF THE PACKET
3538 055676      012721 054560      MOV      #T33DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,
3539 055702      005021      CLR      (R1)+      ;ADDRESS OF CHARACTERISTICS DATA BLOCK
3540 055704      012721 000012      MOV      #10,(R1)+      ;EXTENDED ADDRESS
3541 055710      012721 054572      MOV      #T33BFR,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
3542 055714      005021      CLR      (R1)+      ;ADDRESS OF MESSAGE BUFFER
3543 055716      012721 000024      MOV      #20,(R1)+      ;LENGTH OF MESSAGE BUFFER
3544 055722      005021      CLR      (R1)+
3545 055724      012711 000000      MOV      #0,(R1)      ;SELECT DRIVE ZERO
3546 055730      012702 000030      MOV      #24,R2      ;NUMBER OF LOCATIONS TO BE CLEARED
3547 055734      012762 177777 054572 64$: MOV      #177777,T33BFR(R2) ;ALL ONES TO MESSAGE BUFFER
3548 055742      005742      TST      -(R2)      ;NEXT LOCATION
3549 055744      022702 000000      CMP      #0,R2      ;AT END OF LOOP YES
3550 055750      001371      BNE      64$      ;KEEP GOING UNTIL DONE
3551 055752      000207      RTS      PC      ;RETURN
3552
3553
3554 055754      T33RT2:
3555 055754      SAVREG
3556 055760      012701 054660      MOV      #T33PK2,R1      ;SAVE THE REGISTERS
3557 055764      012721 100006      MOV      #100006,(R1)+      ;START OF THE PACKET
3558 055770      012721 054700      MOV      #T33BF2,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,
3559 055774      005021      CLR      (R1)+      ;ADDRESS OF DATA BLOCK
3560 055776      012721 000006      MOV      #6,(R1)+      ;EXTENDED ADDRESS
3561 056002      005021      CLR      (R1)+      ;SIZE OF DATA BLOCK IN BYTES
3562 056004      012701 054700      MOV      #T33BF2,R1      ;POINT TO DATA SEL AREA
3563 056010      005021      CLR      (R1)+
3564 056012      005011      CLR      (R1)
3565 056014      000207      RTS      PC
3566 056016      T33RT3:

```

3567 056016
3568 056022 012701 054670
3569 056026 005021
3570 056030 005021
3571 056032 005021
3572 056034 005011
3573 056036 000207
3574 056040
056040
056040 104401

SAVREG
MOV #T33PK3,R1
CLR (R1)+
CLR (R1)+
CLR (R1)+
CLR (R1)
RTS PC
ENDTST

;SAVE REGISTERS
;SET UP POINTER ADDRESS
;COMMAND SPACE
;ADDRESS OF DATA BLOCK
;EXTENDED ADDRESS
;SIZE OF DATA TRANSFER BLOCK
;RETURN

L10057:
TRAP CSETST

3577
 3578
 3579
 3580
 3581
 3582
 3583
 3584
 3585
 3586
 3587
 3588
 3589
 3590
 3595
 3596
 3597
 3598
 3599
 3600
 3601
 3602
 3603
 3604
 3605
 3606
 3607
 3608
 3609
 3610
 3611
 3612
 3613
 3614
 3615
 3616
 3617
 3618
 3619
 3620
 3621
 3622
 3623
 3624
 3625
 3626
 3627
 3628
 3629
 3630
 3631
 3632
 3633
 3634
 3635
 3636

.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

MOV	#EPR11,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, SUBTEST 1

THIS TEST VERIFIES THAT THE EOT STATUS IS HANDLED PROPERLY BY
 THE VARIOUS TAPE MOTION COMMANDS. THE FOLLOWING TEST SEQUENCE
 IS PERFORMED:

1. THE TAPE IS REWOUND.
2. WRITE DATA COMMANDS ARE REPEATEDLY ISSUED UNTIL TAPE
 STATUS ALERT TERMINATION IS SEEN WITH EOT=1. ERRORS
 OTHER THAN OCCASIONAL CORRECTABLE OR UNCORRECTABLE DATA
 ERRORS CAUSE A FATAL ERROR REPORT. RECORDS WITH DATA
 ERRORS ARE RETRIED, SO THE TAPE ENDS UP WITH GOOD DATA.
3. ANOTHER WRITE DATA COMMAND IS ISSUED, AND IT IS CHECKED
 THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=1.
4. A WRITE TAPE MARK COMMAND IS ISSUED, AND IT IS CHECKED
 THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=1.
5. A SKIP TAPE MARKS REVERSE COMMAND IS ISSUED, AND IT IS
 CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH
 EOT=1 AND TMK=1.
6. A SPACE RECORDS REVERSE COMMAND, WITH A RECORD COUNT OF
 1, IS ISSUED, AND IT IS CHECKED THAT TAPE STATUS ALERT
 TERMINATION OCCURS, WITH EOT=1 AND TMK=1.
7. A SPACE RECORDS REVERSE COMMAND, WITH A RECORD COUNT OF
 1, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION
 OCCURS, WITH EOT=1.
8. A SPACE RECORDS FORWARD COMMAND, WITH A RECORD COUNT OF
 1, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION
 OCCURS, WITH EOT=1.

.....

9. A READ REVERSE COMMAND IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
10. A READ FORWARD COMMAND IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
11. A SPACE RECORDS REVERSE COMMAND, WITH A RECORD COUNT OF 3, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=0.
12. A SPACE RECORDS FORWARD COMMAND, WITH A RECORD COUNT OF 3, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
13. A SKIP FILE MARKS REVERSE COMMAND IS ISSUED, WHICH SHOULD SKIP ALL THE WAY TO BOT, AND IT IS CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=0, BOT=1, AND RIB=1.

T34LOOP:

Line	Address	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464
------	---------	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

3709	056230	012704	060540		MOV	#T34PACKET,R4		:SUBROUTINE NEEDS PACKET ADDRESS
3710	056234	004737	010742		JSR	PC,WRTCHR		:ISSUE WRITE CHARACTERISTICS
3711	056240	103407			BCS	30\$:BR, IF COMMAND ISSUED OK
3712	056242	005237	002214		INC	FATFLG		:ERROR COUNT
3716	056246	010001			MOV	R0,R1		:SAVE CONTENTS OF TSSR
3717	056250				ERRHRD	ERRNO,WRTMSG,SFIMSG		:WRITE CHARACTERISTICS FAILED
	056250	104456					TRAP	C\$ERHRD
	056252	001132					.WORD	602
	056254	005052					.WORD	WRTMSG
	056256	012114					.WORD	SFIMSG
3718	056260			30\$:	CKLOOP			:LOOP IF SELECTED
	056260	104406					TRAP	C\$CLP1
3719	056262	004737	011074		JSR	PC,REWIND		:REWIND CALL
3720	056266	103411			BCS	35\$:BR, IF TSSR IS OK (GOOD)
3721	056270	016501	000002		MOV	TSSR(R5),R1		:GET TSSR
3722	056274	010004			MOV	R0,R4		:SET UP PACKET
3723	056276	005237	002214		INC	FATFLG		:ERROR COUNT
3727	056302				ERRHRD	ERRNO,T34RWN,PKTSSR		:TSSR IS INCORRECT AFTER REWIND
	056302	104456					TRAP	C\$ERHRD
	056304	001133					.WORD	603
	056306	062337					.WORD	T34RWN
	056310	012126					.WORD	PKTSSR
3728	056312			35\$:	CKLOOP			:LOOP IF SELECTED
	056312	104406					TRAP	C\$CLP1
3729	056314	012737	140005	060660	MOV	#140005,T34PK3		:WRITE DATA, ACK, CVC=1
3730	056322	012703	176750		MOV	#65000,R3		:SET MAX NUMBER OF WRITES
3731	056326	013737	003116	060662	MOV	FREE,T34WB		:SET UP WRITE BUFFER ADDRESS
3732	056334	012737	006654	060666	MOV	#3500,T34SZ		:SET UP BUFFER SIZE (4K BYTES)
3733	056342	012704	060660		MOV	#T34PK3,R4		:R4 = POINTER TO PACKET
3734	056346	010465	000000		MOV	R4,TSDR(R5)		:ISSUE COMMAND
3735	056352	004737	016330		JSR	PC,WAITF		:WAIT FOR SSR TO SET
3736	056356	016501	000002		MOV	TSSR(R5),R1		:GET TSSR CONTENTS
3737	056362	012702	000200		MOV	#SSR,R2		:SET UP EXPECTED
3738	056366	020102			CMP	R1,R2		:ARE THEY EQUAL
3739	056370	001010			BNE	50\$:BR, IT MIGHT BE END OF TAPE
3740	056372	005303			DEC	R3		:DEC RECORD COUNTER
3741	056374	001364			BNE	40\$:BR, IF MORE TO GO
3742	056376	005237	002214		INC	FATFLG		:ERROR COUNT
3746	056402				ERRDF	ERRNO,T34ET,PKTSSR		:EOT NOT FOUND (USE SHORTER TAPE?)
	056402	104455					TRAP	C\$ERDF
	056404	001134					.WORD	604
	056406	062116					.WORD	T34ET
	056410	012126					.WORD	PKTSSR
3747	056412	032701	000004		BIT	#BIT2,R1		:CHECK FOR TAPE STATUS ALERT
3748	056416	001001			BNE	60\$:BR, IF SET
3749	056420	000752			BR	40\$:KEEP GOING
3750	056422	013701	060570		MOV	T34BFR+6,R1		:PICK UP XST0
3751	056426	010102			MOV	R1,R2		:SET UP EXPECTED
3752	056430	052702	000001		BIS	#BIT0,R2		:SET THE EOT BIT ON IN EXPECTED
3753	056434	020102			CMP	R1,R2		:WAS THE BIT ON
3754	056436	001402			BEQ	80\$:BR, IF EOT WAS FOUND
3755	056440	000137	056346		JMP	40\$:KEEP LOOKING
3756	056444			80\$:	CKLOOP			:LOOP IF SELECTED
	056444	104406					TRAP	C\$CLP1
3757	056446	012737	140005	060660	MOV	#140005,T34PK3		:WRITE DATA, ACK, CVC=1
3758	056454	013737	003116	060662	MOV	FREE,T34WB		:SET UP WRITE BUFFER ADDRESS
3759	056462	012737	006654	060666	MOV	#3500,T34SZ		:SET UP BUFFER SIZE (4K BYTES)

3760	056470	012704	060660	MOV	#T34PK3,R4	;R4 = POINTER TO PACKET
3761	056474	010465	000000	MOV	R4,TSDB(R5)	;ISSUE COMMAND
3762	056500	004737	016330	JSR	PC,WAITF	;WAIT FOR SSR TO SET
3763	056504	016501	000002	MOV	TSSR(R5),R1	;GET TSSR CONTENTS
3764	056510	012702	100204	MOV	#SC!SSR!BIT2,R2	;SET UP EXPECTED
3765	056514	020102		CMP	R1,R2	;ARE THEY EQUAL
3766	056516	001406		BEQ	90\$;BR, IF THEY ARE OK
3767	056520	005237	002214	INC	FATFLG	;ERROR COUNT
3771	056524			ERRHRD	ERRNO,T34ET2,PKTSSR	;WRITE TAPE AT EOT FAILED TO SET TSA
	056524	104456				TRAP C\$ERHRD
	056526	001135				.WORD 605
	056530	061367				.WORD T34ET2
	056532	012126				.WORD PKTSSR
3772	056534			90\$:	CKLOOP	;LOOP IF SELECTED
	056534	104406				TRAP C\$CLP1
3773	056536	013701	060570	MOV	T34BFR+6,R1	;PICK UP XST0
3774	056542	010102		MOV	R1,R2	;SET UP EXPECTED
3775	056544	052702	000001	BIS	#BIT0,R2	;SET THE EOT BIT ON IN EXPECTED
3776	056550	020102		CMP	R1,R2	;WAS THE BIT ON
3777	056552	001406		BEQ	100\$;BR, IF EOT WAS FOUND
3778	056554	005237	002214	INC	FATFLG	;ERROR COUNT
3782	056560			ERRHRD	ERRNO,T34ETN,EXPREC	;EOT BIT (XST0) NOT SET
	056560	104456				TRAP C\$ERHRD
	056562	001136				.WORD 606
	056564	061451				.WORD T34ETN
	056566	015554				.WORD EXPREC
3783	056570			100\$:	CKLOOP	;LOOP IF SELECTED
	056570	104406				TRAP C\$CLP1
3784	056572	012737	140011 060660	MOV	#140011,T34PK3	;WRITE TAPE MARK, ACK, CVC=1 COMMAND
3785	056600	012704	060660	MOV	#T34PK3,R4	;R4 = POINTER TO PACKET
3786	056604	010465	000000	MOV	R4,TSDB(R5)	;ISSUE COMMAND
3787	056610	004737	016330	JSR	PC,WAITF	;WAIT FOR SSR TO SET
3788	056614	016501	000002	MOV	TSSR(R5),R1	;GET TSSR CONTENTS
3789	056620	012702	100204	MOV	#SC!SSR!BIT2,R2	;SET UP EXPECTED
3790	056624	020102		CMP	R1,R2	;ARE THEY EQUAL
3791	056626	001406		BEQ	110\$;BR, IF STATUS IS GOOD (OK)
3792	056630	005237	002214	INC	FATFLG	;ERROR COUNT
3796	056634			ERRHRD	ERRNO,T34WTM,PKTSSR	;EOT NOT FOUND (USE SHORTER TAPE?)
	056634	104456				TRAP C\$ERHRD
	056636	001137				.WORD 607
	056640	061300				.WORD T34WTM
	056642	012126				.WORD PKTSSR
3797	056644			110\$:	CKLOOP	;LOOP IF SELECTED
	056644	104406				TRAP C\$CLP1
3798	056646	013701	060570	MOV	T34BFR+6,R1	;PICK UP XST0
3799	056652	010102		MOV	R1,R2	;SET UP EXPECTED
3800	056654	052702	000001	BIS	#BIT0,R2	;SET THE EOT BIT ON IN EXPECTED
3801	056660	020102		CMP	R1,R2	;WAS THE BIT ON
3802	056662	001406		BEQ	120\$;BR, IF EOT WAS FOUND
3803	056664	005237	002214	INC	FATFLG	;ERROR COUNT
3807	056670			ERRHRD	ERRNO,T34ETO,EXPREC	;EOT BIT (XST0) NOT SET
	056670	104456				TRAP C\$ERHRD
	056672	001140				.WORD 608
	056674	061002				.WORD T34ETO
	056676	015554				.WORD EXPREC
3808	056700			120\$:	CKLOOP	;LOOP IF SELECTED
	056700	104406				TRAP C\$CLP1

3809	056702	012737	141410	060660	MOV	#141410,T34PK3	:SKIP TAPE MARK REVERSE ACK,CVC=1 COMMAND
3810	056710	012737	000001	060662	MOV	#1,T34WB	:SET NUMBER (1) OF TMS TO SKIP
3811	056716	012704	060660		MOV	#T34PK3,R4	:R4 = POINTER TO PACKET
3812	056722	010465	000000		MOV	R4,TSD8(R5)	:ISSUE COMMAND
3813	056726	004737	016330		JSR	PC,WAITF	:WAIT FOR SSR TO SET
3814	056732	016501	000002		MOV	TSSR(R5),R1	:GET TSSR CONTENTS
3815	056736	012702	000200		MOV	#SSR,R2	:SET UP EXPECTED
3816	056742	020102			CMP	R1,R2	:ARE THEY EQUAL
3817	056744	001406			BEQ	130\$:BR, IF STATUS IS GOOD (OK)
3818	056746	005237	002214		INC	FATFLG	:ERROR COUNT
3822	056752				ERRHRD	ERRNO,T34STM,PKTSSR	:SKIP TAPE MARK REV. DIDN'T SET TSA
	056752	104456					TRAP C\$ERHRD
	056754	001141					.WORD 609
	056756	061700					.WORD T34STM
	056760	012126					.WORD PKTSSR
3823	056762				130\$: CKLOOP		:LOOP IF SELECTED
	056762	104406					TRAP C\$CLP1
3824	056764	013701	060570		MOV	T34BFR+6,R1	:PICK UP XST0
3825	056770	010102			MOV	R1,R2	:SET UP EXPECTED
3826	056772	052702	000001		BIS	#BIT0,R2	:SET THE EOT BIT ON IN EXPECTED
3827	056776	020102			CMP	R1,R2	:WAS THE BIT ON
3828	057000	001406			BEQ	140\$:BR, IF EOT WAS FOUND
3829	057002	005237	002214		INC	FATFLG	:ERROR COUNT
3833	057006				ERRHRD	ERRNO,T34ETN,EXPREC	:EOT BIT (XST0) NOT SET
	057006	104456					TRAP C\$ERHRD
	057010	001142					.WORD 610
	057012	061451					.WORD T34ETN
	057014	015554					.WORD EXPREC
3834	057016				140\$: CKLOOP		:LOOP IF SELECTED
	057016	104406					TRAP C\$CLP1
3835	057020	013701	060570		MOV	T34BFR+6,R1	:PICK UP XST0
3836	057024	010102			MOV	R1,R2	:SET UP EXPECTED
3837	057026	052702	100000		BIS	#BIT15,R2	:SET THE TMK BIT ON IN EXPECTED
3838	057032	020102			CMP	R1,R2	:WAS THE BIT ON
3839	057034	001406			BEQ	150\$:BR, IF TMK WAS FOUND
3840	057036	005237	002214		INC	FATFLG	:ERROR COUNT
3844	057042				ERRHRD	ERRNO,T34TMK,EXPREC	:EOT BIT (XST0) NOT SET
	057042	104456					TRAP C\$ERHRD
	057044	001143					.WORD 611
	057046	061763					.WORD T34TMK
	057050	015554					.WORD EXPREC
3845	057052				150\$: CKLOOP		:LOOP IF SELECTED
	057052	104406					TRAP C\$CLP1
3846	057054	012737	140410	060660	MOV	#140410,T34PK3	:SPACE RECORDS REVERSE, ACK, CVC=1 CMD
3847	057062	012737	000001	060662	MOV	#1,T34WB	:SPACE ONE RECORD REVERSE
3848	057070	012704	060660		MOV	#T34PK3,R4	:R4 = POINTER TO PACKET
3849	057074	010465	000000		MOV	R4,TSD8(R5)	:ISSUE COMMAND
3850	057100	004737	016330		JSR	PC,WAITF	:WAIT FOR SSR TO SET
3851	057104	016501	000002		MOV	TSSR(R5),R1	:GET TSSR CONTENTS
3852	057110	012702	100204		MOV	#SC!SSR!BIT2,R2	:SET UP EXPECTED
3853	057114	020102			CMP	R1,R2	:ARE THEY EQUAL
3854	057116	001006			BNE	160\$:BR, IT MIGHT BE END OF TAPE
3855	057120	005237	002214		INC	FATFLG	:ERROR COUNT
3859	057124				ERRHRD	ERRNO,T34POS,PKTSSR	:EOT NOT FOUND (USE SHORTER TAPE?)
	057124	104456					TRAP C\$ERHRD
	057126	001144					.WORD 612
	057130	060714					.WORD T34POS

Address	Instruction	Comments	Labels	Registers	Operands	Flow Control	Trap	Word	PKTSSR
3860	057132	012126							
3861	057134	104406							
3862	057136	013701	060570						
3863	057142	010102							
3864	057144	052702	000001						
3865	057150	020102							
3866	057152	001406							
3867	057154	005237	002214						
3870	057160								
3871	057170								
3872	057172	013701	060570						
3873	057176	010102							
3874	057200	042702	100000						
3875	057204	020102							
3876	057206	001406							
3877	057210	005237	002214						
3881	057214								
3882	057224								
3883	057226	012737	140410	060660					
3884	057234	012737	000001	060662					
3885	057242	012704	060660						
3886	057246	010465	000000						
3887	057252	004737	016330						
3888	057256	016501	000002						
3889	057262	012702	000200						
3890	057266	020102							
3891	057270	001406							
3892	057272	005237	002214						
3896	057276								
3897	057306								
3898	057310	013701	060570						
3899	057314	010102							
3900	057316	042702	000001						
3901	057322	020102							
3902	057324	001406							
3903	057326								
3904	057330	012737	140010	060660					
3905	057336	012737	000002	060662					
3906	057344	012704	060660						
3907	057350	010465	000000						

3908	057354	004737	016330		JSR	PC, WAITF		:WAIT FOR SSR TO SET
3909	057360	016501	000002		MOV	TSSR(R5), R1		:GET TSSR CONTENTS
3910	057364	012702	000200		MOV	#SSR, R2		:SET UP EXPECTED
3911	057370	020102			CMP	R1, R2		:ARE THEY EQUAL
3912	057372	001406			BEQ	190\$:BR, IT MIGHT BE END OF TAPE
3913	057374	005237	002214		INC	FATFLG		:ERROR COUNT
3917	057400				ERRHRD	ERRNO, T34POS, PKTSSR		:EOT NOT FOUND (USE SHORTER TAPE?)
	057400	104456					TRAP	C\$ERHRD
	057402	001150					.WORD	616
	057404	060714					.WORD	T34POS
	057406	012126					.WORD	PKTSSR
3918	057410			190\$:	CKLOOP			:LOOP IF SELECTED
	057410	104406					TRAP	C\$CLP1
3919	057412	013701	060570		MOV	T34BFR+6, R1		:PICK UP XST0
3920	057416	010102			MOV	R1, R2		:SET UP EXPECTED
3921	057420	052702	000001		BIS	#BIT0, R2		:SET THE EOT BIT ON IN EXPECTED
3922	057424	020102			CMP	R1, R2		:WAS THE BIT ON
3923	057426	001406			BEQ	200\$:BR, IF EOT WAS FOUND
3924	057430	005237	002214		INC	FATFLG		:ERROR COUNT
3928	057434				ERRHRD	ERRNO, T34ETS, EXPREC		:EOT BIT (XST0) NOT SET
	057434	104456					TRAP	C\$ERHRD
	057436	001151					.WORD	617
	057440	061530					.WORD	T34ETS
	057442	015554					.WORD	EXPREC
3929	057444			200\$:	CKLOOP			:LOOP IF SELECTED
	057444	104406					TRAP	C\$CLP1
3930	057446	012737	140401	060660	MOV	#140401, T34PK3		:READ REVERSE, ACK, CVC=1
3931	057454	013737	003116	060662	MOV	FREE, T34RB		:SET UP WRITE BUFFER ADDRESS
3932	057462	012704	060660		MOV	#T34PK3, R4		:R4 = POINTER TO PACKET
3933	057466	010465	000000		MOV	R4, TSDB(R5)		:ISSUE COMMAND
3934	057472	004737	016330		JSR	PC, WAITF		:WAIT FOR SSR TO SET
3935	057476	016501	000002		MOV	TSSR(R5), R1		:GET TSSR CONTENTS
3936	057502	012702	000200		MOV	#SSR, R2		:SET UP EXPECTED
3937	057506	020102			CMP	R1, R2		:ARE THEY EQUAL
3938	057510	001406			BEQ	205\$:BR, ONLY SSR IS SET
3939	057512	005237	002214		INC	FATFLG		:ERROR COUNT
3943	057516				ERRHRD	ERRNO, T34RRE, PKTSSR		:EOT NOT FOUND (USE SHORTER TAPE?)
	057516	104456					TRAP	C\$ERHRD
	057520	001152					.WORD	618
	057522	061066					.WORD	T34RRE
	057524	012126					.WORD	PKTSSR
3944	057526			205\$:	CKLOOP			:LOOP IF SELECTED
	057526	104406					TRAP	C\$CLP1
3945	057530	012737	140401	060660	MOV	#140401, T34PK3		:READ REVERSE, ACK, CVC=1
3946	057536	013737	003116	060662	MOV	FREE, T34RB		:SET UP WRITE BUFFER ADDRESS
3947	057544	012704	060660		MOV	#T34PK3, R4		:R4 = POINTER TO PACKET
3948	057550	010465	000000		MOV	R4, TSDB(R5)		:ISSUE COMMAND
3949	057554	004737	016330		JSR	PC, WAITF		:WAIT FOR SSR TO SET
3950	057560	016501	000002		MOV	TSSR(R5), R1		:GET TSSR CONTENTS
3951	057564	012702	000200		MOV	#SSR, R2		:SET UP EXPECTED
3952	057570	020102			CMP	R1, R2		:ARE THEY EQUAL
3953	057572	001406			BEQ	210\$:BR, IT MIGHT BE END OF TAPE
3954	057574	005237	002214		INC	FATFLG		:ERROR COUNT
3958	057600				ERRHRD	ERRNO, T34RRE, PKTSSR		:EOT NOT FOUND (USE SHORTER TAPE?)
	057600	104456					TRAP	C\$ERHRD
	057602	001153					.WORD	619
	057604	061066					.WORD	T34RRE

Line	Address	Offset	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	
------	---------	--------	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

4008	060056	016501	000002		MOV	TSSR(R5),R1		:GET TSSR CONTENTS	
4009	060062	012702	000200		MOV	#SSR,R2		:SET UP EXPECTED	
4010	060066	020102			CMP	R1,R2		:ARE THEY EQUAL	
4011	060070	001406			BEQ	250\$:BR, IT MIGHT BE END OF TAPE	
4012	060072	005237	002214		INC	FATFLG		:ERROR COUNT	
4016	060076				ERRHRD	ERRNO,T34POS,PKTSSR		:POSITION COMMAND DIDN'T WORK	
	060076	104456						TRAP	C\$ERHRD
	060100	001157						.WORD	623
	060102	060714						.WORD	T34POS
	060104	012126						.WORD	PKTSSR
4017	060106			250\$:	CKLOOP			:LOOP IF SELECTED	
	060106	104406						TRAP	C\$CLP1
4018	060110	013701	060570		MOV	T34BFR+6,R1		:PICK UP XST0	
4019	060114	010102			MOV	R1,R2		:SET UP EXPECTED	
4020	060116	042702	000001		BIC	#BIT0,R2		:CLEAR THE EOT BIT ON IN EXPECTED	
4021	060122	020102			CMP	R1,R2		:WAS THE BIT ON	
4022	060124	001406			BEQ	260\$:BR, IF EOT WAS FOUND	
4023	060126	005237	002214		INC	FATFLG		:ERROR COUNT	
4027	060132				ERRHRD	ERRNO,T34ETC,EXPREC		:EOT BIT (XST0) NOT CLEAR	
	060132	104456						TRAP	C\$ERHRD
	060134	001160						.WORD	624
	060136	061157						.WORD	T34ETC
	060140	015554						.WORD	EXPREC
4028	060142			260\$:	CKLOOP			:LOOP IF SELECTED	
	060142	104406						TRAP	C\$CLP1
4029	060144	012737	140010	060660	MOV	#140010,T34PK3		:SPACE RECORDS FORWARD, ACK, CVC=1 CMD.	
4030	060152	012737	000005	060662	MOV	#5,T34RB		:NUMBER OF RECORDS TO SPACE	
4031	060160	012704	060660		MOV	#T34PK3,R4		:R4 = POINTER TO PACKET	
4032	060164	010465	000000		MOV	R4,TSDB(R5)		:ISSUE COMMAND	
4033	060170	004737	016330		JSR	PC,WAITF		:WAIT FOR SSR TO SET	
4034	060174	016501	000002		MOV	TSSR(R5),R1		:GET TSSR CONTENTS	
4035	060200	012702	000200		MOV	#SSR,R2		:SET UP EXPECTED	
4036	060204	020102			CMP	R1,R2		:ARE THEY EQUAL	
4037	060206	001406			BEQ	270\$:BR, IT MIGHT BE END OF TAPE	
4038	060210	005237	002214		INC	FATFLG		:ERROR COUNT	
4042	060214				ERRHRD	ERRNO,T34ET,PKTSSR		:TSSR NOT CORRECT	
	060214	104456						TRAP	C\$ERHRD
	060216	001161						.WORD	625
	060220	062116						.WORD	T34ET
	060222	012126						.WORD	PKTSSR
4043	060224			270\$:	CKLOOP			:LOOP IF SELECTED	
	060224	104406						TRAP	C\$CLP1
4044	060226	013701	060570		MOV	T34BFR+6,R1		:PICK UP XST0	
4045	060232	010102			MOV	R1,R2		:SET UP EXPECTED	
4046	060234	052702	000001		BIS	#BIT0,R2		:SET THE EOT BIT ON IN EXPECTED	
4047	060240	020102			CMP	R1,R2		:WAS THE BIT ON	
4048	060242	001400			BEQ	280\$:BR, IF EOT WAS FOUND	
4049	060244			280\$:	CKLOOP			:LOOP IF SELECTED	
	060244	104406						TRAP	C\$CLP1
4050	060246	012737	141410	060660	MOV	#141410,T34PK3		:SKIP FILE MARKS REVERSE,ACK,CVC=1 COMMAND	
4051	060254	012737	000003	060662	MOV	#3,T34RB		:NUMBER OF FILE MARKS	
4052	060262	012704	060660		MOV	#T34PK3,R4		:R4 = POINTER TO PACKET	
4053	060266	010465	000000		MOV	R4,TSDB(R5)		:ISSUE COMMAND	
4054	060272	012737	176750	060674	MOV	#65000,T34DLY		:SET UP DELAY COUNTER	
4055	060300	004737	016330		JSR	PC,WAITF		:WAIT FOR SSR TO SET	
4056	060304	016501	000002		MOV	TSSR(R5),R1		:GET TSSR CONTENTS	
4057	060310	032701	000200		BIT	#SSR,R1		:CHECK FOR SSR SET	

Address	Op Code	Label	Register	Value	Comment
4058	060314	001017			
4059	060316		BNE	286\$:BR, WHEN SSR IS SET
	060316		DELAY	250	:WAIT ABOUT .25 SECONDS
	060316	012727		000250	
	060322	000000			MOV #250, (PC)+
	060324	013727		002116	.WORD 0
	060330	000000			MOV LSDLY, (PC)+
	060332	005367		177772	.WORD 0
	060336	001375			DEC -6(PC)
	060340	005367		177756	BNE -4
	060344	001367			DEC -22(PC)
4060	060346	005337		060674	BNE -20
4061	060352	001352	DEC	T34DLY	:BUMP COUNTER
4062	060354	012702	BNE	285\$:BR, IF MORE TO COUNT
4063	060360	020102	MOV	#SSR, R2	:SET UP EXPECTED
4064	060362	001007	CMP	R1, R2	:ARE THEY EQUAL
4065	060364	005303	BNE	290\$:BR, IT MIGHT BE END OF TAPE
4066	060366	005237	DEC	R3	:DEC RECORD COUNTER
4070	060372		INC	FATFLG	:ERROR COUNT
	060372	104456	ERRHRD	ERRNO, T34ET, PKTSSR	:EOT NOT FOUND (USE SHORTER TAPE?)
	060374	001162			TRAP C\$ERHRD
	060376	062116			.WORD 626
	060400	012126			.WORD T34ET
4071	060402	032701		000004	.WORD PKTSSR
4072	060406	013701	290\$:	BIT #BIT2, R1	:CHECK FOR TAPE STATUS ALERT
4073	060412	010102	MOV	T34BFR+6, R1	:PICK UP XST0
4074	060414	042702	MOV	R1, R2	:SET UP EXPECTED
4075	060420	020102	BIC	#BIT0, R2	:CLEAR THE EOT BIT IN EXPECTED
4076	060422	001406	CMP	R1, R2	:WAS THE BIT ON
4077	060424	005237	BEQ	300\$:BR, IF EOT WAS FOUND
4081	060430		INC	FATFLG	:ERROR COUNT
	060430	104456	ERRHRD	ERRNO, T34ETC, EXPREC	:EOT BIT (XST0) NOT CLEAR
	060432	001163			TRAP C\$ERHRD
	060434	061157			.WORD 627
	060436	015554			.WORD T34ETC
4082	060440		300\$:	CKLOOP	.WORD EXPREC
	060440	104406			TRAP C\$CLP1
4083	060442	013701	MOV	T34BFR+6, R1	:PICK UP XST0
4084	060446	010102	MOV	R1, R2	:SET UP EXPECTED
4085	060450	052702	BIS	#BIT1, R2	:SET THE BOT BIT ON IN EXPECTED
4086	060454	020102	CMP	R1, R2	:WAS THE BIT ON
4087	060456	001406	BEQ	320\$:BR, IF BOT WAS FOUND
4088	060460	005237	INC	FATFLG	:ERROR COUNT
4092	060464		ERRHRD	ERRNO, T34BOT, EXPREC	:EOT BIT (XST0) NOT CLEAR
	060464	104456			TRAP C\$ERHRD
	060466	001164			.WORD 628
	060470	061234			.WORD T34BOT
	060472	015554			.WORD EXPREC
4093	060474		320\$:	CKLOOP	:LOOP IF SELECTED
	060474	104406			TRAP C\$CLP1
4094	060476		600\$:		
4095	060476		ENDSUB		:>>>>>>>>> END SUBTEST >>>>>>>>>
	060476	104403			L10062:
4096	060500	023727	CMP	FATFLG, #15.	TRAP C\$ESUB
4097	060506	103402	BLO	999\$:IS ERROR COUNT AT 25
4098	060510	004737	JSR	PC, CKDROP	:BR, IF LESS THAN 25
4099	060514		999\$:		:TRY TO DROP THE UNIT

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 25-MAY-82 08:43 PAGE 119-9
TEST 6: OPERATIONS AT EOT

SEQ 0220

4100 060514 004737 016536
4101 060520 103002
4102 060522 000137 056072
4103 060526
060526 104432
060530 002662

JSR PC,TSTLOOP
BCC 163\$
JMP T34LOOP
163\$: EXIT TST

:DO WE NEED TO ITERATE TEST
:BR, IF NO LOOP REQUIRED
:EXECUTE AGAIN
:ALL DONE THIS TEST

TRAP C\$EXIT
.WORD L10061-.

4105		;	+			
4106		;	LOCAL STORAGE FOR THIS TEST			
4107		;	-			
4109	060540			.=<. +10>8177770		
4111	060540		T34PACKET:			;COMMAND PACKET FOR TEST
4112	060540	100004		.WORD 100004		;WRITE CHARACTERISTICS COMMAND, WITH ACK
4113	060542	060550		.WORD T34DATA		;ADDRESS OF CHARACTERISTICS BLOCK
4114	060544	000000		.WORD 0		
4115	060546	000010		.WORD 8.		;STARTING VALUE OF BLOCK SIZE
4116	060550		T34DATA:			;CHARACTERISTICS DATA BLOCK
4117	060550	060562		.WORD T34BFR		;ADDRESS OF MESSAGE BUFFER
4118	060552	000000		.WORD 0		
4119	060554	000012		.WORD 10.		;LENGTH OF MESSAGE BUFFER
4120	060556	000000		.WORD 0		
4121	060560	000000	T34DSW:	.WORD 0		;SELECT DRIVE 0
4122	060562		T34BFR:	.BLKW 25.		;MESSAGE BUFFER
4123			;			
4124			;	WRITE SUBSYSTEM MEMORY COMMAND PACKET		
4125			;			
4127	060650			.=<. +10>8177770		
4129	060650		T34PK2:			
4130	060650	100006		.WORD 100006		;WRITE SUB SYS MEM COMMAND, AND ACK
4131	060652	060676		.WORD T34BF2		;ADDRESS OF SELECT BLOCK DATA
4132	060654	000000		.WORD 0		
4133	060656	000006		.WORD 6.		;SIZE OF DATA PACKET
4134						
4138	060660		T34PK3:			
4139	060660	100005		.WORD 100005		;WRITE COMMAND, AND ACK
4140	060662		T34RB:			
4141	060662	000000	T34WB:	.WORD 0		;ADDRESS OF WRITE/READ BUFFER
4142	060664	000000		.WORD 0		
4143	060666	000000	T34SZ:	.WORD 0		;SIZE OF BUFFER (EXTENT)
4144				.EVEN		
4145			;			
4146	060670	000000	T34RSZ:	.WORD 0		;LARGEST TAPE RECORD IN BYTES
4147	060672	000000	T34CNT:	.WORD 0		;TAPE RECORD COUNTER
4148	060674	000000	T34DLY:	.WORD 0		;DELAY COUNTER
4149			;			
4150			;			
4151	060676		T34BF2:			
4152	060676	010	T34BS0:	.BYTE 10		;BSEL0 AREA
4153	060677	200	T34BS1:	.BYTE 200		;BSEL1 AREA
4154	060700	000000	T34S2:	.WORD 0		;SEL 2 AREA
4155	050702	000000	T34S3:	.WORD 0		;DATA AREA
4156			;			
4157			;			
4158			;			
4159			.EVEN			
4160			;	TAPE MOTION PACKET COMMAND VALUES		
4161	060704	100005	T34WD:	.WORD 100005		;WRITE DATA (NEXT)
4162	060706	100405	T34WDR:	.WORD 100405		;WRITE DATA RETRY
4163	060710	102005	T34CON:	.WORD 102005		;WRITE CONTINUOUS
4164	060712	177777		.WORD 177777		;END OF DATA
4165						
4166						

```

4168
4169
4170      ;+
4171      ;LOCAL TEXT MESSAGES FOR TEST
4172      ;+
4173
4174 060714      124      123      123      T34POS: .ASCIZ 'TSSR Incorrect After Position (SPACE RECORDS) Command'
4175 061002      127      122      111      T34ETO: .ASCIZ 'WRITE TAPE MARK At EOT Failed To Set EOT Bit (XST0)'
4176 061066      122      105      101      T34RRE: .ASCIZ 'READ Command At EOT Didn't Give Normal Termination (TSSR)'
4177 061157      125      156      141      T34ETC: .ASCIZ 'Unable To Clear EOT Indication, (XST0) Bit 0'
4178 061234      122      105      127      T34BOT: .ASCIZ 'REWIND Failed To Set BOT (XST0) Bit'
4179 061300      127      122      111      T34WTM: .ASCIZ 'WRITE TAPE MARK At EOT Failed To Set Tape Status Alert'
4180 061367      127      122      111      T34ET2: .ASCIZ 'WRITE DATA At EOT Failed To Set Tape Status Alert'
4181 061451      127      122      111      T34ETN: .ASCIZ 'WRITE DATA At EOT Failed To Set EOT Bit (XST0)'
4182 061530      123      120      101      T34ETS: .ASCIZ 'SPACE RECORDS FORWARD At EOT Failed To Set EOT Bit (XST0)'
4183 061622      122      105      101      T34ETZ: .ASCIZ 'READ DATA At EOT Failed To Set EOT Bit (XST0)'
4184 061700      124      123      123      T34STM: .ASCIZ 'TSSR Incorrect After SKIP TAPE MARK REVERSE At EOT'
4185 061763      120      117      123      T34TMK: .ASCIZ 'POSITION Command At EOT Onto Tape Mark Failed To Set TMK (XST0)'
4186 062063      127      122      111      T34SSR: .ASCIZ 'WRITE Command Not Accepted'
4187 062116      105      117      124      T34ET: .ASCIZ 'EOT Not Found In 65000 3.5K Writes, (Use Shorter Tape)'
4188 062205      127      122      111      T34EOT: .ASCIZ 'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
4189 062263      124      123      123      T34TM: .ASCIZ 'TSSR Not Correct After WRITE Command Reject'
4190 062337      122      145      167      T34RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
4191 062406      122      101      115      T34RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
4192 062461      124      123      123      T34AM3: .ASCIZ 'TSSR Init. Failed After WRITE Command'
4193 062527      104      162      151      T34OFL: .ASCIZ 'Drive 7 Select Failed To Set 'OFL' In TSSR'
4194 062602      124      123      123      T34WDD: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, SWB Bit Set'
4195 062671      124      123      123      T34WDC: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
4196 062773      103      126      103      T34VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
4197 063046      124      123      102      T34BA: .ASCIZ 'TSBA Not Correct After WRITE DATA Command'
4198 063120      127      122      111      T34WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
4199 063207      117      160      145      T34ID: .ASCIZ 'Operations At EOT'
4200
4201      .EVEN
4202
4203      ;+
4204      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4205      ;WRITE SUBSYSTEM MEMORY COMMAND
4206      ;+
4207      ;-
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224

```

T34REST:

SAVREG

```

MOV #T34PACKET,R1
MOV #100004,(R1)+
MOV #T34DATA,(R1)+
CLR (R1)+
MOV #10.,(R1)+
MOV #T34BFR,(R1)+
CLR (R1)+
MOV #20.,(R1)+
CLR (R1)+
MOV #0,(R1)
MOV #24.,R2
MOV #177777,T34BFR(R2)
TST -(R2)
CMP R2,#0
BNE 64$

```

```

;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

;LENGTH OF MESSAGE BUFFER

;SELECT DRIVE ZERO
;NUMBER OF LOCATIONS TO BE CLEARED
;ALL ONES TO MESSAGE BUFFER
;BUMP DOWN TO NEXT LOCATION
;R2 AT ZERO YET
;KEEP GOING UNTIL DONE

```

060562 64\$:

```
4225 063322 000207          RTS      PC          ;RETURN
4226
4227
4228 063324          T34RT2:
4229 063324          SAVREG
4230 063330 012701 060650    MOV      #T34PK2,R1      ;SAVE THE REGISTERS
4231 063334 012721 100006    MOV      #100006,(R1)+    ;START OF THE PACKET
4232 063340 012721 060676    MOV      #T34BF2,(R1)+    ;WRITE SUBSYSTEM MEM. WITH ACK
4233 063344 005021          CLR      (R1)+            ;ADDRESS OF DATA BLOCK
4234 063346 012721 000006    MOV      #6,(R1)+        ;EXTENDED ADDRESS
4235 063352 012701 060676    MOV      #T34BF2,R1      ;SIZE OF DATA BLOCK IN BYTES
4236 063356 005021          CLR      (R1)+            ;POINT TO DATA SEL AREA
4237 063360 005021          CLR      (R1)+
4238 063362 005011          CLR      (R1)
4239 063364 000207          RTS      PC          ;RETURN
4240 063366          T34RT3:
4241 063366          SAVREG
4242 063372 012701 060660    MOV      #T34PK3,R1      ;SAVE THE REGISTERS
4243 063376 012721 100005    MOV      #100005,(R1)+    ;START OF THE PACKET
4244 063402 005021          CLR      (R1)+            ;WRITE TAPE. WITH ACK
4245 063404 005021          CLR      (R1)+            ;ADDRESS OF DATA BLOCK
4246 063406 005011          CLR      (R1)             ;EXTENDED ADDRESS
4247 063410 000207          RTS      PC          ;SIZE OF DATA BLOCK
4248 063412          ENDTST          ;RETURN
      063412
      063412 104401          L10061: TRAP      C$ETST
```

MOV #250.(PC)+

063506 012727 000250

Address	Offset	Hex	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	
---------	--------	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

```

4350
4351
4352
4353
4354
4355
4356
4357 063730 012737 140005 067510      MOV      #140005,T35PK3      ;WRITE DATA,ACK,CVC=1 COMMAND
4358 063736 012704 067510      MOV      #T35PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4359 063742 010465 000000      50$:    MOV      R4,TSD8(R5)      ;ISSUE COMMAND
4360 063746 004737 016330      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
4361 063752 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
4362 063756 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
4363 063762 020102      CMP      R1,R2      ;ARE THEY EQUAL
4364 063764 001406      BEQ      60$      ;BR, IF OK
4365 063766 005237 002214      INC      FATFLG      ;ERROR COUNT
4366 063772      ERRHRD  ERRNO,T35WDE,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA
4367 063772 104456      TRAP      C$ERHRD
4368 063774 001301      .WORD      705
4369 063776 070266      .WORD      T35WDE
4370 064000 012126      .WORD      PKTSSR
4371 064002      60$:    CKLOOP      ;LOOP IF SELECTED
4372 064002 104406      TRAP      C$CLP1
4373 064004 005303      DEC      R3      ;BUMP RECORD COUNTER
4374 064006 001355      BNE      50$      ;BR, IF MORE RRECORDS TO COUNT
4375
4376
4377
4378
4379
4380 064010 012737 000012 067542      MOV      #10.,T35DLV      ;SET UP DELAY COUNTER
4381 064016      70$:    DELAY      250      ;WAIT ABOUT .25 SEC
4382 064016 012727 000250      MOV      #250,(PC)+
4383 064022 000000      .WORD      0
4384 064024 013727 002116      MOV      L$DLV,(PC)+
4385 064030 000000      .WORD      0
4386 064032 005367 177772      DEC      -6(PC)
4387 064036 001375      BNE      -4
4388 064040 005367 177756      DEC      -22(PC)
4389 064044 001367      BNE      -20
4390 064046 005337 067542      DEC      T35DLV      ;BUMP COUNTER DOWN
4391 064052 001361      BNE      70$      ;BR, IF MORE TO DELAY
4392 064054 005737 002220      TST      EXTFEA      ;CHECK FOR EXTENDED FEATURES SW SWITCH
4393 064060 001042      BNE      110$      ;BR IF SWITCH IS ON
4394 064062 112737 000200 067521      MOV8     #200,T35BS1      ;WRITE MISCELLANEOUS CONT/READ STATUS
4395 064070 112737 000010 067520      MOV8     #10,T35BS0      ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4396 064076 012704 067500      MOV      #T35PK2,R4      ;WRITE SUBSYS MEM PACKET
4397 064102 010465 000000      MOV      R4,TSD8(R5)      ;ISSUE COMMAND
4398 064106 004737 016416      JSR      PC,CHKTSSR      ;WAIT FOR SSR
4399 064112 103407      BCS      90$      ;BR, IF NO ERROR
4400 064114 010001      MOV      R0,R1      ;ERROR, SAVE TSSR
4401 064116 005237 002214      INC      FATFLG      ;ERROR COUNT
4402 064122      ERRHRD  ERRNO,T35SSR,PKTSSR      ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
4403 064122 104456      TRAP      C$ERHRD
4404 064124 001302      .WORD      706

```

Line	Address	Hex	Label	Instruction	Comment	Register/Value
4398	064126	072422				
	064130	012126				
	064132	104406	90\$:	CKLOOP	:LOOP IF SELECTED	
4399	064134	012704	067370	MOV	#T3SPACKET,R4	
4400	064140	004737	010742	JSR	PC,WRTCHR	
4401	064144	103407		BCS	100\$	
4402	064146	005237	002214	INC	FATFLG	
4406	064152	010001		MOV	R0,R1	
4407	064154			ERRHRD	ERRNO,WRTMSG,SFIMSG	
	064154	104456				
	064156	001303				
	064160	005052				
	064162	012114				
4408	064164			100\$:	CKLOOP	:SCOPE LOOP
	064164	104406				
4409	064166	012737	176750	110\$:	MOV	#65000.,T35DLY
4410	064174	005037	067536	CLR	T35CNT	
4411						
4412						
4413						
4414						
4415						
4416						
4417						
4418	064200	012737	142012	067510	MOV	#142012,T35PK3
4419	064206	012704	067510		MOV	#T35PK3,R4
4420	064212	010465	000000		MOV	R4,T35DB(R5)
4421	064216	016501	000002	120\$:	MOV	TSSR(R5),R1
4422	064222	032701	000200		BIT	#SSR,R1
4423	064226	001021			BNE	130\$
4424	064230	005237	067536		INC	T35CNT
4425	064234				DELAY	1
	064234	012727	000001			
	064240	000000				
	064242	013727	002116			
	064246	000000				
	064250	005367	177772			
	064254	001375				
	064256	005367	177756			
	064262	001367				
4426	064264	005337	067542		DEC	T35DLY
4427	064270	001352			BNE	120\$
4428	064272	012702	000200	130\$:	MOV	#SSR,R2
4429	064276	020102			CMP	R1,R2
4430	064300	001406			BEQ	140\$
4431	064302	005237	002214		INC	FATFLG
4435	064306				ERRHRD	ERRNO,T35RWE,PKTSSR
	064306	104456				
	064310	001304				
	064312	072770				
	064314	012126				
4436	064316			140\$:	CKLOOP	:LOOP IF SELECTED
	064316	104406				
4437	064320	005737	002216		TST	INTRECV
4438	064324	001410			BEQ	150\$
4439	064326	016501	000002		MOV	TSSR(R5),R1


```

4440 064332 005237 002214          INC    FATFLG          ;ERROR COUNT
4444 064336          ERRHRD  ERRNO,T35INT,PKTSSR      ;INTERRUPT RECEIVED (BAD)
          064336 104456          TRAP    C$ERHRD
          064340 001305          .WORD   709
          064342 072601          .WORD   T35INT
          064344 012126          .WORD   PKTSSR
4445 064346          150$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
          064346 104406
4446
4447
4448
4449
4450
4451
4452
          :*****
          :NOW CHECK FOR THE MOTION BITS SET
          :*****
4453 064350 013701 067420          MOV    T35BFR+6,R1          ;PICK UP XST0
4454 064354 010102          MOV    R1,R2          ;SET UP EXPECTED
4455 064356 052702 000200          BIS    #BIT7,R2          ;SET MOT BIT IN EXPECTED
4456 064362 020102          CMP    R1,R2          ;DOES EXP = REC'D
4457 064364 001406          BEQ    160$          ;BR, IF EQUAL (OK)
4458 064366 005237 002214          INC    FATFLG          ;ERROR COUNT
4462 064372          ERRHRD  ERRNO,T35MOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
          064372 104456          TRAP    C$ERHRD
          064374 001306          .WORD   710
          064376 072503          .WORD   T35MOT
          064400 015554          .WORD   EXPREC
4463 064402          160$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
          064402 104406
4464 064404 013701 067424          MOV    T35BFR+12,R1        ;PICK UP XST2
4465 064410 010102          MOV    R1,R2          ;SET UP EXPECTED
4466 064412 052702 100000          BIS    #BIT15,R2        ;SET OPM BIT IN EXPECTED
4467 064416 020102          CMP    R1,R2          ;DOES EXP = REC'D
4468 064420 001406          BEQ    170$          ;BR, IF EQUAL (OK)
4469 064422 005237 002214          INC    FATFLG          ;ERROR COUNT
4473 064426          ERRHRD  ERRNO,T35OPM,EXPREC      ;OPM BIT NOT SET
          064426 104456          TRAP    C$ERHRD
          064430 001307          .WORD   711
          064432 072672          .WORD   T35OPM
          064434 015554          .WORD   EXPREC
4474 064436          170$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
          064436 104406
4475 064440 012737 000027 067542 175$:  MOV    #23.,T35DLY        ;SET UP DELAY COUNTER
4476 064446          DELAY  250          ;START DELAY
          064446 012727 000250          MOV    #250,(PC)+
          064452 000000          .WORD   0
          064454 013727 002116          MOV    L$DLY,(PC)+
          064460 000000          .WORD   0
          064462 005367 177772          DFC    -6(PC)
          064466 001375          BNE    -4
          064470 005367 177756          DEC    -22(PC)
          064474 001367          BNE    -20
4477 064476 005337 067542          DEC    T35DLY          ;BUMP DELAY COUNTER
4478 064502 001361          BNE    175$          ;BR, IF MORE DELAY
4479 064504          ENDSUB
          064504
          064504 104403          L10064:
4480 064506 023727 002214 000017          CMP    FATFLG,#15.          ;IS ERROR COUNT AT 25          TRAP    C$ESUB
  
```

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 25-MAY-82 08:43 PAGE 122-5
TEST 7: EXTENDED MODE FEATURES

4481 064514 103402
4482 064516 004737 017262
4483 064522

999\$: BLO 999\$
JSR PC,CKDROP

;BR, IF LESS THAN 25
;TRY TO DROP THE UNIT

SEQ 0229

T
T

TEST 7: SUBTEST 2

WITH THE INTERRUPT ENABLE (IE) BIT SET (1), CAUSES ALMOST IMMEDIATE TERMINATION AND AN INTERRUPT. STATUS IN THE MESSAGE BUFFER IS CHECKED TO VERIFY THAT THE MOTION (MOT) AND OPERATION IN PROGRESS (OPM) STATUS BITS ARE BOTH SET.

Address	Hex	Dec	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op41
---------	-----	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

```

064700 005052
064702 012114
4530 064704 104406 25$: CKLOOP ;LOOP IF SELECTED .WORD WRTMSG
064704 104406 ;CALL TAPE REWIND COMMAND .WORD SFIMSG
4531 064706 004737 011074 JSR PC,REWIND TRAP C$CLP1
4532 064712 103411 BCS 30$ ;BR, IF NO PROBLEM
4533 064714 010004 MOV R0,R4 ;SET UP REWIND PACKET ADDRESS
4534 064716 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4535 064722 005237 002214 INC FATFLG ;ERROR COUNT
4539 064726 ERRHRD ERRNO,T35RWN,PKTSSR ;REWIND NOT ACCEPTED
064726 104456 TRAP C$ERHRD
064730 001312 .WORD 714
064732 070644 .WORD T35RWN
064734 012126 .WORD PKTSSR
4540 064736 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
064736 104406 ;PICK UP XSTO
4541 064740 013701 067420 MOV T35BFR+6,R1 ;SET UP EXPECTED
4542 064744 010102 MOV R1,R2 ;SET BOT BIT IN EXPECTED
4543 064746 052702 000002 BIS #BIT1,R2 ;DOES EXP = REC'D
4544 064752 020102 CMP R1,R2 ;BR, IF EQUAL (OK)
4545 064754 001406 BEQ 40$ ;ERROR COUNT
4546 064756 005237 002214 INC FATFLG ;TAPE NOT AT BOT AFTER REWIND
4550 064762 ERRHRD ERRNO,T35BOT,EXPREC TRAP C$ERHRD
064762 104456 .WORD 715
064764 001313 .WORD T35BOT
064766 070340 .WORD EXPREC
064770 015554
4551 064772 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
064772 104406 ;NUMBER OF RECORDS
4552 064774 012703 000024 MOV #20.,R3 ;SET UP RECORD SIZE
4553 065000 012737 000450 067516 MOV #256.,T35SZ ;ADDRESS OF WRITE BUFFER
4554 065006 013737 003'16 067512 MOV FREE,T35WB
4555
4556 ;*****
4557 ;
4558 ;WRITE DATA,ACK,CVC=1 COMMAND
4559 ;
4560 ;*****
4561
4562 065014 012737 140005 067510 MOV #140005,T35PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
4563 065022 012704 067510 MOV #T35PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
4564 065026 010465 000000 50$: MOV R4,TSD8(R5) ;ISSUE COMMAND
4565 065032 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
4566 065036 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4567 065042 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
4568 065046 020102 CMP R1,R2 ;ARE THEY EQUAL
4569 065050 001406 BEQ 60$ ;BR, IF OK
4570 065052 005237 002214 INC FATFLG ;ERROR COUNT
4574 065056 ERRHRD ERRNO,T35WDE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
065056 104456 TRAP C$ERHRD
065060 001314 .WORD 716
065062 070266 .WORD T35WDE
065064 012126 .WORD PKTSSR
4575 065066 60$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
065066 104406
4576
4577 ;*****

```

```

4578
4579      ;WAIT FOR TAPE TO STOP ALL MOTION
4580      ;*****
4581
4582
4583 065070 012737 000012 067542      MOV    #10.,T35DLY      ;SET UP DELAY COUNTER
4584 065076      70$: DELAY    250      ;WAIT ABOUT .25 SEC
      065076 012727 000250      MOV    #250,(PC)+
      065102 000000      .WORD    0
      065104 013727 002116      MOV    L$DLY,(PC)+
      065110 000000      .WORD    0
      065112 005367 177772      DEC     -6(PC)
      065116 001375      BNE     -4
      065120 005367 177756      DEC     -22(PC)
      065124 001367      BNE     -20
4585 065126 005337 067542      DEC     T35DLY      ;BUMP COUNTER DOWN
4586 065132 001361      BNE     70$      ;BR, IF MORE TO DELAY
4587 065134 005737 002220      TST     EXTFEA      ;CHECK FOR EXTENDED FEATURES SW SWITCH
4588 065140 001042      BNE     110$      ;BR IF SWITCH IS ON
4589 065142 112737 000200 067521      MOVB   #200,T35BS1      ;WRITE MISCELLANEOUS CONT/READ STATUS
4590 065150 112737 000010 067520      MOVB   #10,T35BS0      ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4591 065156 012704 067500      MOV    #T35PK2,R4      ;WRITE SUBSYS MEM PACKET
4592 065162 010465 000000      MOV    R4,T$DB(R5)      ;ISSUE COMMAND
4593 065166 004737 016416      JSR    PC,CHKTSSR      ;WAIT FOR SSR
4594 065172 103407      BCS     90$      ;BR, IF NO ERROR
4595 065174 010001      MOV    R0,R1      ;ERROR, SAVE TSSR
4596 065176 005237 002214      INC     FATFLG      ;ERROR COUNT
4600 065202      ERRHRD  ERRNO,T35SSR,PKTSSR      ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      065202 104456      TRAP    C$ERHRD
      065204 001315      .WORD    717
      065206 072422      .WORD    T35SSR
      065210 012126      .WORD    PKTSSR
4601 065212      90$: CKLOOP      ;LOOP IF SELECTED
      065212 104406      TRAP    C$CLP1
4602 065214 012704 067370      MOV    #T35PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
4603 065220 004737 010742      JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
4604 065224 103407      BCS     100$      ;BR, IF COMMAND ISSUED OK
4605 065226 005237 002214      INC     FATFLG      ;ERROR COUNT
4609 065232 010001      MOV    R0,R1      ;SAVE CONTENTS OF TSSR
4610 065234      ERRHRD  ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTICS FAILED
      065234 104456      TRAP    C$ERHRD
      065236 001316      .WORD    718
      065240 005052      .WORD    WRTMSG
      065242 012114      .WORD    SFIMSG
4611 065244      100$: CKLOOP      ;SCOPE LOOP
      065244 104406      TRAP    C$CLP1
4612 065246 012737 176750 067542 110$: MOV    #65000.,T35DLY      ;SET UP DELAY COUNTER
4613 065254 005037 067536      CLR     T35CNT      ;DELAY COUNTER
4614
4615      ;*****
4616      ;REWIND IMMEDIATE,ACK,CVC=1,IE=1 COMMAND
4617
4618      ;*****
4619
4620
4621 065260 012737 142212 067510      MOV    #142212,T35PK3      ;REWIND IMMEDIATE,ACK,CVC=1,IE=1 COMMAND
4622 065266 012704 067510      MOV    #T35PK3,R4      ;SET UP R4 WITH PACKET ADDRESS

```

4623	065272	010465	000000		MOV	R4,TSD8(R5)		:ISSUE COMMAND	
4624	065276	016501	000002	120\$:	MOV	TSSR(R5),R1		:GET TSSR CONTENTS	
4625	065302	032701	000200		BIT	#SSR,R1		:CHECK FOR SSR SET	
4626	065306	001021			BNE	130\$:BR, WHEN SSR IS SET	
4627	065310	005237	067536		INC	T35CNT		:BUMP THE CYCLE COUNTER	
4628	065314				DELAY	1		:DELAY TO KEEP COUNTER DOWN	
	065314	012727	000001					MOV	#1,(PC)+
	065320	000000						.WORD	0
	065322	013727	002116					MOV	LSDLY,(PC)+
	065326	000000						.WORD	0
	065330	005367	177772					DEC	-6(PC)
	065334	001375						BNE	-4
	065336	005367	177756					DEC	-22(PC)
	065342	001367						BNE	-20
4629	065344	005337	067542		DEC	T35DLY		:DROP DEAD TIMER BUMP DOWN	
4630	065350	001352			BNE	120\$:BR, IF MORE TIME TO GO	
4631	065352	012702	000200	130\$:	MOV	#SSR,R2		:SET UP EXPECTED	
4632	065356	020102			CMP	R1,R2		:ARE THEY EQUAL	
4633	065360	001406			BEQ	140\$:BR, IF OK	
4634	065362	005237	002214		INC	FATFLG		:ERROR COUNT	
4638	065366				ERRHRD	ERRNO,T35RWE,PKTSSR		:TSSR INCORRECT AFTER WRITE DATA	
	065366	104456						TRAP	C\$ERHRD
	065370	001317						.WORD	719
	065372	072770						.WORD	T35RWE
	065374	012126						.WORD	PKTSSR
4639	065376			140\$:	CKLOOP			:LOOP IF SELECTED	
	065376	104406						TRAP	C\$CLP1
4640	065400	005737	002216		TST	INTRECV		:CHECK FOR INTERRUPTS	
4641	065404	001010			BNE	150\$:BR, IF INTERRUPTS DETECTED	
4642	065406	016501	000002		MOV	TSSR(R5),R1		:GET TSSR STATUS FOR PRINTOUT	
4643	065412	005237	002214		INC	FATFLG		:ERROR COUNT	
4647	065416				ERRHRD	ERRNO,T35NIN,PKTSSR		:INTERRUPT NOT RECEIVED (BAD)	
	065416	104456						TRAP	C\$ERHRD
	065420	001320						.WORD	720
	065422	073056						.WORD	T35NIN
	065424	012126						.WORD	PKTSSR
4648	065426			150\$:	CKLOOP			:LOOP IF SELECTED	
	065426	104406						TRAP	C\$CLP1
4649									
4650									
4651									
4652									
4653									
4654									
4655									
4656	065430	013701	067420		MOV	T35BFR+6,R1		:PICK UP XSTO	
4657	065434	010102			MOV	R1,R2		:SET UP EXPECTED	
4658	065436	052702	000200		BIS	#BIT7,R2		:SET MOT BIT IN EXPECTED	
4659	065442	020102			CMP	R1,R2		:DOES EXP = REC'D	
4660	065444	001406			BEQ	160\$:BR, IF EQUAL (OK)	
4661	065446	005237	002214		INC	FATFLG		:ERROR COUNT	
4665	065452				ERRHRD	ERRNO,T35MOT,EXPREC		:TAPE NOT AT BOT AFTER REWIND	
	065452	104456						TRAP	C\$ERHRD
	065454	001321						.WORD	721
	065456	072503						.WORD	T35MOT
	065460	015554						.WORD	EXPREC
4666	065462			160\$:	CKLOOP			:LOOP IF SELECTED	

 :NOW CHECK FOR THE MOTION BITS SET
 :*****

	065462	104406						TRAP	C\$CLP1
4667	065464	013701	067424		MOV	T35BFR+12,R1	:PICK UP XST2		
4668	065470	010102			MOV	R1,R2	:SET UP EXPECTED		
4669	065472	052702	100000		BIS	#BIT15,R2	:SET OPM BIT IN EXPECTED		
4670	065476	020102			CMP	R1,R2	:DOES EXP = REC'D		
4671	065500	001406			BEQ	170\$:BR, IF EQUAL (UK)		
4672	065502	005237	002214		INC	FATFLG	:ERROR COUNT		
4676	065506				ERRHRD	ERRNO,T35OPM,EXPREC	:OPM BIT NOT SET		
	065506	104456						TRAP	C\$ERHRD
	065510	001322						.WORD	722
	065512	072672						.WORD	T35OPM
	065514	015554						.WORD	EXPREC
4677	065516			170\$:	CKLOOP		:LOOP IF SELECTED		
	065516	104406						TRAP	C\$CLP1
4678	065520	012737	000027	067542	MOV	#23.,T35DLY	:SET UP DELAY COUNTER		
4679	065526			175\$:	DELAY	250	:START DELAY		
	065526	012727	000250					MOV	#250,(PC)+
	065532	000000						.WORD	0
	065534	013777	002116					MOV	L\$DLY,(PC)+
	065540	000000						.WORD	0
	065542	005367	177772					DEC	-6(PC)
	065546	001375						BNE	.-4
	065550	005367	177756					DEC	-22(PC)
	065554	001367						BNE	.-20
4680	065556	005337	067542		DEC	T35DLY	:BUMP DELAY COUNTER		
4681	065562	001361			BNE	175\$:BR, IF MORE DELAY		
4682	065564				ENDSUB				
	065564								
	065564	104403						L10065:	
4683	065566	023727	002214	000017	CMP	FATFLG,#15.	:IS ERROR COUNT AT 25	TRAP	C\$ESUB
4684	065574	103402			BLO	999\$:BR, IF LESS THAN 25		
4685	065576	004737	017262		JSR	PC,CKDROP	:TRY TO DROP THE UNIT		
4686	065602			999\$:					

TEST 7, SUBTEST 3

VERIFIES THAT THE NON-TAPE-MOTION COMMAND GET STATUS, ISSUED IMMEDIATELY AFTER TERMINATION OF A REWIND WITH IMMEDIATE INTERRUPT COMMAND, TERMINATES PROPERLY AND PROVIDES PROPER STATUS. FIRST, A NUMBER OF DATA RECORDS ARE WRITTEN FROM BOT. THEN THE REWIND WITH IMMEDIATE INTERRUPT COMMAND IS ISSUED AND TERMINATION VERIFIED. THEN THE GET STATUS COMMAND IS ISSUED. THE GET STATUS SHOULD TERMINATE ALMOST IMMEDIATELY AND SHOW MOT=1 AND OPM=1 IN THE MESSAGE BUFFER. AFTER A DELAY LONG ENOUGH TO ALLOW THE REWIND TO COMPLETE AND THE TAPE COME TO REST, GET STATUS IS AGAIN ISSUED AND THE STATUS CHECKED; BOTH MOT AND OPM SHOULD BE CLEAR.

4708	065602			BGNSUB		; >>>>>>>>> BEGIN SUBTEST >>>>>>>>>			
	065602					T7.3:			
	065602	104402					TRAP	C\$BSUB	
4709	065604	004737	073164	JSR	PC,T3SREST	;SET COMMAND PACKET			
4710	065610	004737	073256	JSR	PC,T3SRT2	;SET UP OTHER COMMAND PACKET			
4711	065614	004737	073320	JSR	PC,T3SRT3	;SET UP OTHER COMMAND PACKET			
4712	065620	004737	016054	JSR	PC,SOFINIT	;DO INITIALIZE ON CONTROLLER			
4713	065624	103407		BCS	20\$;BR IF INIT WAS OK			
4714	065626	005237	002214	INC	FATFLG	;ERROR COUNT			
4718	065632	010001		MOV	R0,R1	;CONTENTS OF TSSR REGISTER			
4719	065634			ERRDF	ERRNO,SFIERR,SFIMSG	;FATAL ERROR TSSR WAS NOT OK			
	065634	104455					TRAP	C\$ERDF	
	065636	001323					.WORD	723	
	065640	003646					.WORD	SFIERR	
	065642	012114					.WORD	SFIMSG	
4720	065644	013737	002174	MOV	UNITN,T3SDSW	;SET UP UNIT NUMBER IN PACKET			
4721	065652	012704	067370	MOV	#T3SPACKET,R4	;SUBROUTINE NEEDS PACKET ADDRESS			
4722	065656	004737	010742	JSR	PC,WRTCHR	;ISSUE WRITE CHARACTERISTICS			
4723	065662	103407		BCS	23\$;BR, IF COMMAND ISSUED OK			
4724	065664	005237	002214	INC	FATFLG	;ERROR COUNT			
4728	065670	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR			
4729	065672			ERRHRD	ERRNO,WRTMSG,SFIMSG	;WRITE CHARACTERISTIC FAILED			
	065672	104456					TRAP	C\$ERHRD	
	065674	001324					.WORD	724	
	065676	005052					.WORD	WRTMSG	
	065700	012114					.WORD	SFIMSG	
4730	065702		23\$:	CKLOOP		;LOOP IF SELECTED			
	065702	104406					TRAP	C\$CLP1	
4731	065704	004737	011074	JSR	PC,REWIND	;CALL TAPE REWIND COMMAND			
4732	065710	103411		BCS	30\$;BR, IF NO PROBLEM			
4733	065712	010004		MOV	R0,R4	;SET UP REWIND PACKET ADDRESS			
4734	065714	016501	000002	MOV	TSSR(R5),R1	;GET CONTENTS FOR CALL			
4735	065720	005237	002214	INC	FATFLG	;ERROR COUNT			
4739	065724			ERRHRD	ERRNO,T3SRWN,PKTTSSR	;REWIND NOT ACCEPTED			
	065724	104456					TRAP	C\$ERHRD	
	065726	001325					.WORD	725	
	065730	070644					.WORD	T3SRWN	

4740	065732	012126			30\$:	CKLOOP		:LOOP IF SELECTED	.WORD	PKTSSR
	065734	104406							TRAP	C\$CLP1
4741	065736	013701	067420			MOV	T35BFR+6,R1	:PICK UP XSTO		
4742	065742	010102				MOV	R1,R2	:SET UP EXPECTED		
4743	065744	052702	000002			BIS	#BIT1,R2	:SET BOT BIT IN EXPECTED		
4744	065750	020102				CMP	R1,R2	:DOES EXP = REC'D		
4745	065752	001406				BEQ	40\$:BR, IF EQUAL (OK)		
4746	065754	005237	002214			INC	FATFLG	:ERROR COUNT		
4750	065760					ERRHRD	ERRNO,T35BOT,EXPREC	:TAPE NOT AT BOT AFTER REWIND		
	065760	104456							TRAP	C\$ERHRD
	065762	001326							.WORD	726
	065764	070340							.WORD	T35BOT
	065766	015554							.WORD	EXPREC
4751	065770				40\$:	CKLOOP		:LOOP IF SELECTED		
	065770	104406							TRAP	C\$CLP1
4752	065772	012703	000024			MOV	#20.,R3	:STARTING RECORD SIZE		
4753	065776	013737	003116	067512		MOV	FREE,T35WB	:STARTING WRITE BUFFER ADDRESS		
4754										
4755										
4756										
4757										
4758										
4759										
4760										
4761	066004	012737	140005	067510	65\$:	MOV	#140005,T35PK3	:WRITE DATA,CVC=1,ACK COMMAND		
4762	066012	012704	067510			MOV	#T35PK3,R4	:SET UP R4 WITH PACKET ADDRESS		
4763	066016	010300				MOV	R3,R0	:SET PATTERN IN CORRECT REGISTER		
4764	066020	004737	017502			JSR	PC,FILLMEM	:FILL MEMORY WITH RECORD SIZE		
4765	066024	010337	067516			MOV	R3,T35SZ	:SET UP RECORD SIZE IN PACKET		
4766	066030	010465	000000			MOV	R4,TSDDB(R5)	:ISSUE COMMAND		
4767	066034	004737	016330			JSR	PC,WAITF	:WAIT FOR SSR TO SET		
4768	066040	016501	000002			MOV	TSSR(R5),R1	:GET TSSR CONTENTS		
4769	066044	012702	000200			MOV	#SSR,R2	:SET UP EXPECTED		
4770	066050	020102				CMP	R1,R2	:ARE THEY EQUAL		
4771	066052	001406				BEQ	80\$:BR, IF OK		
4772	066054	005237	002214			INC	FATFLG	:ERROR COUNT		
4776	066060					ERRHRD	ERRNO,T35WDC,PKTSSR	:TSSR INCORRECT AFTER WRITE DATA		
	066060	104456							TRAP	C\$ERHRD
	066062	001327							.WORD	727
	066064	071200							.WORD	T35WDC
	066066	012126							.WORD	PKTSSR
4777	066070				80\$:	CKLOOP		:LOOP IF SELECTED		
	066070	104406							TRAP	C\$CLP1
4778										
4779										
4780										
4781										
4782										
4783										
4784										
4785	066072	012737	141005	067510		MOV	#141005,T35PK3	:WRITE DATA RETRY,CVC=1,ACK COMMAND		
4786	066100	010465	000000			MOV	R4,TSDDB(R5)	:ISSUE COMMAND		
4787	066104	004737	016330			JSR	PC,WAITF	:WAIT FOR SSR TO SET		
4788	066110	016501	000002			MOV	TSSR(R5),R1	:GET TSSR CONTENTS		
4789	066114	012702	000200			MOV	#SSR,R2	:SET UP EXPECTED		
4790	066120	020102				CMP	R1,R2	:ARE THEY EQUAL		

```

4791 066122 001406          BEQ      90$          ;BR, IF OK
4792 066124 005237 002214    INC      FATFLG          ;ERROR COUNT
4796 066130          ERRHRD  ERRNO,T35WRF,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA RETRY
                                TRAP      C$ERHRD
                                .WORD     728
                                .WORD     T35WRF
                                .WORD     PKTSSR
4797 066140          90$:    CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD     C$CLP1
4798 066142 005723          TST      (R3)+          ;BUMP RECORD SIZE COUNTER
4799 066144 020327 000052    CMP      R3,#42.        ;AT 42 SIZE YET
4800 066150 001315          BNE      65$          ;BR, IF MORE RECORDS TO WRITE
4801 066152 004737 011074    JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
4802 066156 103411          BCS      230$          ;BR, IF NO PROBLEM
4803 066160 010001          MOV      R0,R1          ;SAVE TSSR
4804 066162 016501 000002    MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
4805 066166 005237 002214    INC      FATFLG          ;ERROR COUNT
4809 066172          ERRHRD  ERRNO,T35RWN,EXPREC      ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     729
                                .WORD     T35RWN
                                .WORD     EXPREC
4810 066202          230$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD     C$CLP1
4811 066204 013701 067420    MOV      T35BFR+6,R1      ;PICK UP XSTO
4812 066210 010102          MOV      R1,R2          ;SET UP EXPECTED
4813 066212 052702 000002    BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
4814 066216 020102          CMP      R1,R2          ;DOES EXP = REC'D
4815 066220 001406          BEQ      240$          ;BR, IF EQUAL (OK)
4816 066222 005237 002214    INC      FATFLG          ;ERROR COUNT
4820 066226          ERRHRD  ERRNO,T35BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD     730
                                .WORD     T35BOT
                                .WORD     EXPREC
4821 066236          240$:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD     C$CLP1
4822 066240 012703 000024    MOV      #20.,R3        ;STARTING RECORD SIZE
4823 066244 013737 003116 067512  MOV      FREE,T35RB      ;STARTING READ BUFFER ADDRESS
4824
4825 ;*****
4826 ;
4827 ;READ DATA,ACK COMMAND
4828 ;
4829 ;*****
4830
4831 066252 012737 100001 067510 265$:  MOV      #100001,T35PK3      ;READ DATA,ACK COMMAND
4832 066260 012704 067510      MOV      #T35PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4833 066264 012700 177777      MOV      #177777,R0        ;SET PATTERN IN CORRECT REGISTER
4834 066270 004737 017502      JSR      PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
4835 066274 010337 067516      MOV      R3,T35SZ        ;SET UP RECORD SIZE IN PACKET
4836 066300 010465 000000      MOV      R4,T35DB(R5)      ;ISSUE COMMAND
4837 066304 004737 016330      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
4838 066310 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
4839 066314 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
4840 066320 020102          CMP      R1,R2          ;ARE THEY EQUAL
4841 066322 001406          BEQ      280$          ;BR, IF OK

```

[illegible]

VERIFIES THAT A TAPE-MOTION COMMAND (READ, WRITE, POSITION), ISSUED IMMEDIATELY AFTER TERMINATION OF A REWIND WITH IMMEDIATE INTERRUPT COMMAND, IS "QUEUED" BY THE CONTROLLER AND THEN EXECUTES PROPERLY.

Address	Hex	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
---------	-----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

```

4924 066632          ERRHRD  ERRNO,T35RWN,PKTSSR      ;REWIND NOT ACCEPTED
      066632 104456
      066634 001337
      066636 070644
      066640 012126
4925 066642          30$:  CKLOOP                      ;LOOP IF SELECTED
      066642 104406
4926 066644 013701 067420      MOV      T35BFR+6,R1      ;PICK UP XST0
4927 066650 010102      MOV      R1,R2                  ;SET UP EXPECTED
4928 066652 052702 000002      BIS      #BIT1,R2          ;SET BOT BIT IN EXPECTED
4929 066656 020102      CMP      R1,R2                  ;DOES EXP = REC'D
4930 066660 001406      BEQ      40$                      ;BR, IF EQUAL (OK)
4931 066662 005237 002214      INC      FATFLG              ;ERROR COUNT
4935 066666          ERRHRD  ERRNO,T35BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      066666 104456
      066670 001340
      066672 070340
      066674 015554
4936 066676          40$:  CKLOOP                      ;LOOP IF SELECTED
      066676 104406
4937 066700 012703 000024      MOV      #20.,R3          ;STARTING RECORD SIZE
4938 066704 013737 003116 067512  MOV      FREE,T35WB      ;STARTING WRITE BUFFER ADDRESS
4939
4940      ;*****
4941      ;WRITE DATA,CVC=1,ACK COMMAND
4942      ;
4943      ;*****
4944
4945
4946 066712 012737 140005 067510 65$:  MOV      #140005,T35PK3      ;WRITE DATA,CVC=1,ACK COMMAND
4947 066720 012704 067510      MOV      #T35PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
4948 066724 010300      MOV      R3,R0                  ;SET PATTERN IN CORRECT REGISTER
4949 066726 004737 017502      JSR      PC,FILLMEM          ;FILL MEMORY WITH RECORD SIZE
4950 066732 010337 067516      MOV      R3,T35SZ          ;SET UP RECORD SIZE IN PACKET
4951 066736 010465 000000      MOV      R4,TSDB(R5)          ;ISSUE COMMAND
4952 066742 004737 016330      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
4953 066746 016501 000002      MOV      TSSR(R5),R1        ;GET TSSR CONTENTS
4954 066752 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
4955 066756 020102      CMP      R1,R2                  ;ARE THEY EQUAL
4956 066760 001406      BEQ      80$                      ;BR, IF OK
4957 066762 005237 002214      INC      FATFLG              ;ERROR COUNT
4961 066766          ERRHRD  ERRNO,T35WDC,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA
      066766 104456
      066770 001341
      066772 071200
      066774 012126
4962 066776          80$:  CKLOOP                      ;LOOP IF SELECTED
      066776 104406
4963
4964      ;*****
4965      ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
4966      ;
4967      ;*****
4968
4969
4970 067000 012737 111005 067510      MOV      #111005,T35PK3      ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
4971 067006 010465 000000      MOV      R4,TSDB(R5)          ;ISSUE COMMAND
  
```

4972	067012	004737	016330		JSR	PC, WAITF		;WAIT FOR SSR TO SET
4973	067016	0165C1	000002		MOV	TSSR(R5), R1		;GET TSSR CONTENTS
4974	067022	012702	000200		MOV	#SSR, R2		;SET UP EXPECTED
4975	067026	020102			CMP	R1, R2		;ARE THEY EQUAL
4976	067030	001406			BEQ	90\$;BR, IF OK
4977	067032	005237	002214		INC	FATFLG		;ERROR COUNT
4981	067036				ERRHRD	ERRNO, T35WRF, EXPREC		;TSSR INCORRECT AFTER WRITE DATA RETRY
	067036	104456					TRAP	C\$ERHRD
	067040	001342					.WORD	738
	067042	072245					.WORD	T35WRF
	067044	015554					.WORD	EXPREC
4982	067046			90\$:	CKLOOP			;LOOP IF SELECTED
	067046	104406					TRAP	C\$CLP1
4983	067050	005723			TST	(R3)+		;BUMP RECORD SIZE COUNTER
4984	067052	020327	000052		CMP	R3, #42.		;AT 42 SIZE YET
4985	067056	001315			BNE	65\$;BR, IF MORE RECORDS TO WRITE
4986	067060	004737	011074		JSR	PC, REWIND		;CALL TAPE REWIND COMMAND
4987	067064	103411			BCS	230\$;BR, IF NO PROBLEM
4988	067066	016501	000002		MOV	TSSR(R5), R1		;GET TSSR CONTENTS
4989	067072	010004			MOV	R0, R4		;GET PACKET ADDRESS
4990	067074	005237	002214		INC	FATFLG		;ERROR COUNT
4994	067100				ERRHRD	ERRNO, T35RWN, PKTSSR		;REWIND NOT ACCEPTED
	067100	104456					TRAP	C\$ERHRD
	067102	001343					.WORD	739
	067104	070644					.WORD	T35RWN
	067106	012126					.WORD	PKTSSR
4995	067110			230\$:	CKLOOP			;LOOP IF SELECTED
	067110	104406					TRAP	C\$CLP1
4996	067112	013701	067420		MOV	T35BFR+6, R1		;PICK UP XST0
4997	067116	010102			MOV	R1, R2		;SET UP EXPECTED
4998	067120	052702	000002		BIS	#BIT1, R2		;SET BOT BIT IN EXPECTED
4999	067124	020102			CMP	R1, R2		;DOES EXP = REC'D
5000	067126	001406			BEQ	240\$;BR, IF EQUAL (OK)
5001	067130	005237	002214		INC	FATFLG		;ERROR COUNT
5005	067134				ERRHRD	ERRNO, T35BOT, EXPREC		;TAPE NOT AT BOT AFTER REWIND
	067134	104456					TRAP	C\$ERHRD
	067136	001344					.WORD	740
	067140	070340					.WORD	T35BOT
	067142	015554					.WORD	EXPREC
5006	067144			240\$:	CKLOOP			;LOOP IF SELECTED
	067144	104406					TRAP	C\$CLP1
5007	067146	012703	000024		MOV	#20., R3		;STARTING RECORD SIZE
5008	067152	013737	003116	067512	MOV	FREE, T35RB		;STARTING READ BUFFER ADDRESS
5009								
5010								
5011								
5012								
5013								
5014								
5015								
5016	067160	012737	100001	067510	265\$:	MOV	#100001, T35PK3	;READ DATA, ACK COMMAND
5017	067166	012704	067510		MOV	#T35PK3, R4		;SET UP R4 WITH PACKET ADDRESS
5018	067172	010337	067516		MOV	R3, T35S2		;SET UP RECORD SIZE IN PACKET
5019	067176	010465	000000		MOV	R4, TSDB(R5)		;ISSUE COMMAND
5020	067202	004737	016330		JSR	PC, WAITF		;WAIT FOR SSR TO SET
5021	067206	016501	000002		MOV	TSSR(R5), R1		;GET TSSR CONTENTS
5022	067212	012702	000200		MOV	#SSR, R2		;SET UP EXPECTED

5023	067216	020102		CMP	R1,R2		;ARE THEY EQUAL	
5024	067220	001406		BEQ	280\$;BR, IF OK	
5025	067222	005237	002214	INC	FATFLG		;ERROR COUNT	
5029	067226			ERRHRD	ERRNO,T3SRDF,PKTSSR		;TSSR INCORRECT AFTER READ DATA	
	067226	104456					TRAP	C\$ERHRD
	067230	001345					.WORD	741
	067232	067632					.WORD	T3SRDF
	067234	012126					.WORD	PKTSSR
5030	067236			280\$:	CKLOOP		;LOOP IF SELECTED	
	067236	104406					TRAP	C\$CLP1
5031	067240	013702	003116	MOV	FREE,R2		;GET BUFFER ADDRESS	
5032	067244	010304		MOV	R3,R4		;GET RECORD SIZE	
5033	067246	162704	000024	SUB	#20.,R4		;POINT BACK TO 1ST RECORD	
5034	067252	060204		285\$:	ADD	R2,R4	;POINT TO 1ST LOC IN BUFFER	
5035	067254	000303		SWAB	R3		;SWAP BYTES SWB=1 ETC.	
5036	067256	021403		CMP	(R4),R3		;DATA WRITTEN = READ	
5037	067260	001410		BEQ	290\$;BR, IF DATA OK (GOOD)	
5038	067262	011401		MOV	(R4),R1		;PICK UP BAD DATA	
5039	067264	010302		MOV	R3,R2		;SET UP EXPECTED	
5040	067266	005237	002214	INC	FATFLG		;ERROR COUNT	
5044	067272			ERRHRD	ERRNO,T3SDTA,EXPREC		;DATA IN BUFFER NOT CORRECT	
	067272	104456					TRAP	C\$ERHRD
	067274	001346					.WORD	742
	067276	072325					.WORD	T3SDTA
	067300	015554					.WORD	EXPREC
5045	067302			290\$:	CKLOOP		;LOOP IF SELECTED	
	067302	104406					TRAP	C\$CLP1
5046	067304	005724		TST	(R4)+		;BUMP TO NEXT ADDRESS	
5047	067306	160204		SUB	R2,R4		;BACK TO RECORD SIZE	
5048	067310	000303		SWAB	R3		;PUT R3 BACK INTO SHAPE	
5049	067312	020403		CMP	R4,R3		;AT END OF RECORD YET	
5050	067314	001356		BNE	285\$;BR, IF MORE DATA TO CHECK	
5051	067316	005723		TST	(R3)+		;BUMP RECORD SIZE	
5052	067320	020327	000050	CMP	R3,#40.		;DONE YET	
5053	067324	001315		BNE	265\$;BR, IF NOT DONE YET (MORE READS)	
5054	067326			300\$:	CKLOOP		;LOOP IF SELECTED	
	067326	104406					TRAP	C\$CLP1
5055	067330			ENDSUB			;>>>>>>>>> END SUBTEST >>>>>>>>>	
	067330						L10067:	
	067330	104403					TRAP	C\$ESUB
5056	067332	023727	002214 000017	CMP	FATFLG,#15.		;IS ERROR COUNT AT 25	
5057	067340	103402		BLO	999\$;BR, IF LESS THAN 25	
5058	067342	004737	017262	JSR	PC,CKDROP		;TRY TO DROP THE UNIT	
5059	067346			999\$:				
5060				:				
5061				:				
5062				:				
5063	067346	004737	016536	JSR	PC,TSTLOOP		;DO WE NEED TO ITERATE TEST	
5064	067352	103002		BCC	163\$;BR, IF NO LOOP REQUIRED	
5065	067354	000137	063444	JMP	T35LOOP		;EXECUTE AGAIN	
5066	067360			163\$:	EXIT		;ALL DONE THIS TEST	
	067360	104432					TRAP	C\$EXIT
	067362	003760					.WORD	L10063-

```

5068
5069
5070
5072      067370
5074 067370
5075 067370 100004
5076 067372 067400
5077 067374 000000
5078 067376 000012
5079 067400
5080 067400 067412
5081 067402 000000
5082 067404 000024
5083 067406 000000
5084 067410 000000
5085 067412
5086
5087
5088
5090      067500
5092 067500
5093 067500 100006
5094 067502 067520
5095 067504 000000
5096 067506 000006
5097
5101 067510
5102 067510 100005
5103 067512
5104 067512 003116
5105 067514 000000
5106 067516 000000
5107
5108
5109
5110
5111 067520
5112 067520      010
5113 067521      200
5114 067522 000000
5115 067524 000000
5116
5117
5118
5119
5120
5121 067526 100205
5122 067530 100605
5123 067532 102205
5124 067534 177777
5125
5126
5127 067536 000000
5128 067540 000000
5129 067542 000000
5130

;+
;LOCAL STORAGE FOR THIS TEST
;-
;=<.+10>8177770
T35PACKET:
      .WORD 100004
      .WORD T35DATA
      .WORD 0
      .WORD 10.
T35DATA:
      .WORD T35BFR
      .WORD 0
      .WORD 20.
      .WORD 0
T35DSW: .WORD 0
T35BFR: .BLKW 25.

;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 SUBSYSTEM MEMORY COMMAND PACKET
;=<.+10>8177770
T35PK2:
      .WORD 100006
      .WORD T35BFR2
      .WORD 0
      .WORD 6.
T35PK3:
      .WORD 100005
T35RB:
T35WB: .WORD FREE
      .WORD 0
T35SZ: .WORD 0
      .EVEN

;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)

T35BFR2:
T35BS0: .BYTE 10
T35BS1: .BYTE 200
T35S2: .WORD 0
T35S3: .WORD 0

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

;EVEN
;TAPE MOTION PACKET COMMAND VALUES
T35RN: .WORD 100205
T35WDR: .WORD 100605
T35CON: .WORD 102205
      .WORD 177777

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

T35CNT: .WORD 0
T35CNU: .WORD 0
T35DLY: .WORD 0

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

```



```

5132
5133
5134
5135
5136
5137
5138 067544      124      141      160 T35WNG: .ASCIIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
5139 067632      124      123      123 T35RDF: .ASCIIZ 'TSSR Incorrect After READ DATA Command'
5140 067701      122      105      122 T35RRF: .ASCIIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
5141 067776      120      117      123 T35SC: .ASCIIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
5142 070060      122      111      102 T35LOR: .ASCIIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
5143 070130      124      123      123 T35WDF: .ASCIIZ 'TSSR Not Correct After Illegal Mode Bits Set'
5144 070205      111      154      154 T35LOQ: .ASCIIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
5145 070266      124      123      123 T35WDE: .ASCIIZ 'TSSR Not Correct After WRITE DATA Command'
5146 070340      124      141      160 T35BOT: .ASCIIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
5147 070433      127      122      111 T35TIM: .ASCIIZ 'WRITE DATA RETRY'S Erase Tape Not Long Enough'
5148 070510      122      105      122 T35EOT: .ASCIIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
5149 070567      124      123      123 T35TM: .ASCIIZ 'TSSR Not Correct After REREAD COMMAND Reject'
5150 070644      122      145      167 T35RWN: .ASCIIZ 'Rewind (POSITION) Command Not Accepted'
5151 070713      122      101      115 T35RNC: .ASCIIZ 'RAM Error, Correct Data Pattern Not In Ram'
5152 070766      124      123      123 T35AM3: .ASCIIZ 'TSSR Init. Failed After REREAD COMMAND'
5153 071035      104      162      151 T35OFL: .ASCIIZ 'Drive 7 Select Failed To Set 'OFL' In TSSR'
5154 071110      124      123      123 T35WDD: .ASCIIZ 'TSSR Not Correct After RE .EAD DATA Command, SWB Bit Set'
5155 071200      124      123      123 T35WDC: .ASCIIZ 'TSSR Not Correct After REREAD DATA Command'
5156 071253      103      126      103 T35VCK: .ASCIIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
5157 071326      124      123      102 T35BA: .ASCIIZ 'TSBA Not Correct After REREAD DATA Command'
5158 071401      127      122      111 T35WSS: .ASCIIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
5159 071470      122      145      141 T35LON: .ASCIIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
5160 071552      122      145      141 T35LOP: .ASCIIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
5161 071634      122      145      163 T35PBP: .ASCIIZ 'Residual Byte Count Incorrect After Short Record Read'
5162 071722      122      145      141 T35TRL: .ASCIIZ 'Reading Long Record Failed To Give Tape Status Alert'
5163 072010      127      122      111 T35NEF: .ASCIIZ 'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
5164 072106      124      123      123 T35SCF: .ASCIIZ 'TSSR Not Correct After SPACE RECORDS Command'
5165 072163      124      123      123 T35TSA: .ASCIIZ 'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
5166 072245      124      123      123 T35WRF: .ASCIIZ 'TSSR Not Correct After WRITE DATA RETRY Command'
5167 072325      104      141      164 T35DTA: .ASCIIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
5168 072422      124      123      123 T35SSR: .ASCIIZ 'TSSR Incorrect After WRITE MISCELLANEOUS Command'
5169 072503      115      117      124 T35MOT: .ASCIIZ 'MOT Bit (XST0) Not Set During Rewind (Extended Features Mode)'
5170 072601      111      156      164 T35INT: .ASCIIZ 'Interrupt Received After REWIND Command (IE Bit Not Set)'
5171 072672      117      120      115 T35OPM: .ASCIIZ 'OPM Bit (XST2) Not Set During Rewind (Extended Features Mode)'
5172 072770      124      123      123 T35RWE: .ASCIIZ 'TSSR Incorrect After Extended Features REWIND Command'
5173 073056      116      157      040 T35NIN: .ASCIIZ 'No Interrupt Detected After REWIND IMMEDIATE'
5174 073133      105      170      164 T35ID: .ASCIIZ 'Extended Mode Functions'
5175
5176
5177
5178
5179
5180
5181
5182
5183 073164
5184 073164
5185 073170      012701      067370
5186 073174      012721      100004
5187 073200      012721      067400
5188 073204      005021

```

```

: +
: LOCAL TEXT MESSAGES FOR TEST
: -

```

```

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

```

```

T35REST:
      SAVREG
      MOV      #T35PACKET,R1
      MOV      #100004,(R1)+
      MOV      #T35DATA,(R1)+
      CLR      (R1)+
      :SAVE THE REGISTERS
      :START OF THE PACKET
      :WRITE SUBSYSTEM MEM. WITH ACK,
      :ADDRESS OF CHARAISTICS DATA BLOCK
      :EXTENDED ADDRESS

```

5189	073206	012721	000012		MOV	#10.,(R1)+	:SIZE OF DATA BLOCK IN BYTES
5190	073212	012721	067412		MOV	#T35BFR,(R1)+	:ADDRESS OF MESSAGE BUFFER
5191	073216	005021			CLR	(R1)+	
5192	073220	012721	000024		MOV	#20.,(R1)+	:LENGTH OF MESSAGE BUFFER
5193	073224	005021			CLR	(R1)+	
5194	073226	012711	000000		MOV	#0,(R1)	:SELECT DRIVE ZERO
5195	073232	012702	000030		MOV	#24.,R2	:NUMBER OF LOCATIONS TO BE CLEARED
5196	073236	012762	177777	067412 64\$:	MOV	#177777,T35BFR(R2)	:ALL ONES TO MESSAGE BUFFER
5197	073244	005742			TST	-(R2)	:NEXT LOCATION
5198	073246	022702	000000		CMP	#0,R2	:AT END OF LOOP YET
5199	073252	001371			BNE	64\$:KEEP GOING UNTIL DONE
5200	073254	000207			RTS	PC	:RETURN
5201							
5202							
5203	073256				T35RT2:		
5204	073256				SAVREG		:SAVE THE REGISTERS
5205	073262	012701	067500		MOV	#T35PK2,R1	:START OF THE PACKET
5206	073266	012721	100006		MOV	#100006,(R1)+	:WRITE SUBSYSTEM MEM. WITH ACK.
5207	073272	012721	067520		MOV	#T35BF2,(R1)+	:ADDRESS OF DATA BLOCK
5208	073276	005021			CLR	(R1)+	:EXTENDED ADDRESS
5209	073300	012721	000006		MOV	#6.,(R1)+	:SIZE OF DATA BLOCK IN BYTES
5210	073304	005021			CLR	(R1)+	
5211	073306	012701	067520		MOV	#T35BF2,R1	:POINT TO DATA SEL AREA
5212	073312	005021			CLR	(R1)+	
5213	073314	005011			CLR	(R1)	
5214	073316	000207			RTS	PC	:RETURN
5215	073320				T35RT3:		
5216	073320				SAVREG		:SAVE REGISTERS
5217	073324	012701	067510		MOV	#T35PK3,R1	:SET UP POINTER ADDRESS
5218	073330	005021			CLR	(R1)+	:COMMAND SPACE
5219	073332	005021			CLR	(R1)+	:ADDRESS OF DATA BLOCK
5220	073334	005021			CLR	(R1)+	:EXTENDED ADDRESS
5221	073336	005011			CLR	(R1)	:SIZE OF DATA TRANSFER BLOCK
5222	073340	000207			RTS	PC	:RETURN
5223	073342				ENDTST		
	073342						
	073342	104401					

L10063: TRAP C\$ETST

.SBTTL 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. THE FOLLOWING SUBTESTS ARE PERFORMED:

VERIFIES THAT NORMAL BUFFERING ON WRITE DATA COMMANDS OPERATES PROPERLY AT LOW TAPE SPEED. THE FOLLOWING SEQUENCE IS PERFORMED:

1. THE TAPE IS REWOUND.
2. BUFFERING IS DISABLED AND LOW TAPE SPEED IS SELECTED (VIA WRITE CHARACTERISTICS COMMAND).
3. AN INITIAL RECORD IS WRITTEN ONTO THE TAPE IN ORDER TO MOVE THE TAPE OFF BOT.
4. THE PROGRAM DELAYS FOR A TIME SUFFICIENT TO ALLOW THE TAPE TO REPOSITION AND COME TO REST.
5. A WRITE DATA COMMAND, WITH A BYTE COUNT LESS THAN 3.5K, IS ISSUED, AND THE PROGRAM COUNTS, IN A WAIT LOOP, THE TIME IT TAKES TO RECEIVE COMMAND TERMINATION. THIS SHOULD BE A RELATIVELY LONG TIME SINCE BUFFERING IS DISABLED.
6. BUFFERING IS ENABLED.
7. THE WRITE DATA COMMAND IS AGAIN ISSUED, WITH THE SAME BYTE COUNT AS THAT USED PREVIOUSLY. THE TIME TO COMPLETION IS AGAIN MEASURED.
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.

THE TEST CONSISTS OF THE FOLLOWING 2 SUBTESTS

Line	Address	Offset	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	
------	---------	--------	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

Line	Address	Offset	Label	Operation	Comments	Trap	Trap Data
	073540	104456				TRAP	C\$ERHRD
	073542	001442				.WORD	802
	073544	005052				.WORD	WRTMSG
	073546	012114				.WORD	SFIMSG
5335	073550		25\$:	CKLOOP	;LOOP IF SELECTED		
	073550	104406				TRAP	C\$CLP1
5336	073552	004737	011074	JSR	PC,REWIND		
5337	073556	103407		BCS	30\$		
5338	073560	010004		MOV	R0,R4		
5339	073562	005237	002214	INC	FATFLG		
5343	073566			ERRHRD	ERRNO,T36RWN,PKTSSR		
	073566	104456				TRAP	C\$ERHRD
	073570	001443				.WORD	803
	073572	077153				.WORD	T36RWN
	073574	012126				.WORD	PKTSSR
5344	073576		30\$:	CKLOOP	;LOOP IF SELECTED		
	073576	104406				TRAP	C\$CLP1
5345	073600	013701	075610	MOV	T36BFR+6,R1		
5346	073604	010102		MOV	R1,R2		
5347	073606	052702	000002	BIS	#BIT1,R2		
5348	073612	020102		CMP	R1,R2		
5349	073614	001406		BEQ	40\$		
5350	073616	005237	002214	INC	FATFLG		
5354	073622			ERRHRD	ERRNO,T36BOT,EXPREC		
	073622	104456				TRAP	C\$ERHRD
	073624	001444				.WORD	804
	073626	076647				.WORD	T36BOT
	073630	015554				.WORD	EXPREC
5355	073632		40\$:	CKLOOP	;LOOP IF SELECTED		
	073632	104406				TRAP	C\$CLP1
5356	073634	013737	002174	MOV	UNITN,T36DSW		
5357	073642	052737	000010	BIS	#BIT3,T36DSW		
5358	073650	012704	075560	MOV	#T36PACKET,R4		
5359	073654	004737	010742	JSR	PC,WRTCHR		
5360	073660	103407		BCS	50\$		
5361	073662	005237	002214	INC	FATFLG		
5365	073666	010001		MOV	R0,R1		
5366	073670			ERRHRD	ERRNO,WRTMSG,SFIMSG		
	073670	104456				TRAP	C\$ERHRD
	073672	001445				.WORD	805
	073674	005052				.WORD	WRTMSG
	073676	012114				.WORD	SFIMSG
5367	073700		50\$:	CKLOOP	;LOOP IF SELECTED		
	073700	104406				TRAP	C\$CLP1
5368	073702	012737	003720	MOV	#2000.,T36SZ		
5369	073710	013737	003116	MOV	FREE,T36WB		
5370	073716	012737	140005	MOV	#140005,T36PK3		
5371	073724	012704	075700	MOV	#T36PK3,R4		
5372	073730	010465	000000	MOV	R4,TSDB(R5)		
5373	073734	004737	016330	JSR	PC,WAITF		
5374	073740	016501	000002	MOV	TSSR(R5),R1		
5375	073744	012702	000200	MOV	#SSR,R2		
5376	073750	020102		CMP	R1,R2		
5377	073752	001406		BEQ	60\$		
5378	073754	005237	002214	INC	FATFLG		
5382	073760			ERRHRD	ERRNO,WRTERR,PKTSSR		
	073760	104456				TRAP	C\$ERHRD

Address	Hex	Dec	Label	Instruction	Comment	Register/Value	Operation	Value
073762	001446							
073764	005107							
073766	012126							
5383 073770	104406		60\$:	CKLOOP	:LOOP IF SELECTED			
073770	104406							
5384 073772	012737	000012	075732	MOV	#10.,T36DLY			
5385 074000	012727	000250		70\$:	DELAY	250		
074000	000000							
074004	013727	002116						
074012	000000							
074014	005367	177772						
074020	001375							
074022	005367	177756						
074026	001367							
5386 074030	005337	075732		DEC	T36DLY			
5387 074034	001361			BNE	70\$			
5388 074036	012737	006642	075706	MOV	#3490.,T36SZ			
5389 074044	012737	140005	075700	MOV	#140005,T36PK3			
5390 074052	012704	075700		MOV	#T36PK3,R4			
5391 074056	005037	075726		CLR	T36CNT			
5392 074062	012737	001750	075732	MOV	#1000.,T36DLY			
5393 074070	010465	000000		MOV	R4,T36DLY			
5394 074074	016501	000002		MOV	TSSR(R5),R1			
5395 074100	032701	000200		BIT	#SSR,R1			
5396 074104	001021			BNE	90\$			
5397 074106	005237	075726		INC	T36CNT			
5398 074112				DELAY	1			
074112	012727	000001						
074116	000000							
074120	013727	002116						
074124	000000							
074126	005367	177772						
074132	001375							
074134	005367	177756						
074140	001367							
5399 074142	005337	075732		DEC	T36DLY			
5400 074146	001352			BNE	80\$			
5401 074150	012702	000200		MOV	#SSR,R2			
5402 074154	020102			CMP	R1,R2			
5403 074156	001406			BEQ	'00\$			
5404 074160	005237	002214		INC	FATFLG			
5408 074164				ERRHRD	ERRNO,T36WDE,PKTSSR			
074164	104456							
074166	001447							
074170	076575							
074172	012126							
5409 074174	104406		100\$:	CKLOOP	:LOOP IF SELECTED			
074174	104406							
5410 074176	013737	002174	075600	MOV	UNITN,T36DSW			
5411 074204	052737	000030	075600	BIS	#BIT3!BIT4,T36DSW			
5412 074212	012704	075560		MOV	#T36PACKET,R4			
5413 074216	004737	010742		JSR				

Address	Offset	Hex	Dec	Label	Instruction	Comment	Trap	Value
074232	104456						TRAP	C\$ERHRD
074234	001450						.WORD	808
074236	005052						.WORD	WRTMSG
074240	012114						.WORD	SFIMSG
5421	074242			110\$:	CKLOOP	:LOOP IF SELECTED	TRAP	C\$CLP1
5422	074242	104406						
5423	074244	012737	006642	075706	MOV #3490,T36SZ	:SET SIZE OF TRANSFER		
5424	074252	012737	140005	075700	MOV #140005,T36PK3	:WRITE DATA,ACK,CVC=1 COMMAND		
5425	074260	012704	075700		MOV #T36PK3,R4	:SET UP R4 WITH PACKET ADDRESS		
5426	074264	005037	075730		CLR T36CNU	:CLEAR COUNTER		
5427	074270	012737	001750	075732	MOV #1000,T36DLY	:SET DROP DEAD COUNTER VALUE		
5428	074276	010465	000000		MOV R4,TSDB(R5)	:ISSUE COMMAND		
5429	074302	016501	000002	120\$:	MOV TSSR(R5),R1	:GET TSSR CONTENTS		
5430	074306	032701	000200		BIT #SSR,R1	:CHECK FOR SSR SET		
5431	074312	001021			BNE 130\$:BR, IF SSR IS SET		
5432	074314	005237	075730		INC T36CNU	:BUMP CYCLE COUNTER		
5433	074320				DELAY 1	:CUT NUMBER OF LOOPS DOWN		
5434	074320	012727	000001				MOV	#1,(PC)+
5435	074324	000000					.WORD	0
5436	074326	013727	002116				MOV	L\$DLY,(PC)+
5437	074332	000000					.WORD	0
5438	074334	005367	177772				DEC	-6(PC)
5439	074340	001375					BNE	.-4
5440	074342	005367	177756				DEC	-22(PC)
5441	074346	001367					BNE	.-20
5442	074350	005337	075732		DEC T36DLY	:BUMP DROP DEAD COUNTER		
5443	074354	001352			BNE 120\$:BR, IF THERE IS STILL TIME		
5444	074356	012702	000200	130\$:	MOV #SSR,R2	:SET UP EXPECTED		
5445	074362	020102			CMP R1,R2	:ARE THEY EQUAL		
5446	074364	001406			BEQ 140\$:BR, IF OK		
5447	074366	005237	002214		INC FATFLG	:ERROR COUNT		
5448	074372				ERRHRD ERRNO,WRTERR,PKTSSR	:TSSR INCORRECT AFTER WRITE DATA		
5449	074372	104456					TRAP	C\$ERHRD
5450	074374	001451					.WORD	809
5451	074376	005107					.WORD	WRTERR
5452	074400	012126					.WORD	PKTSSR
5453	074402			140\$:	CKLOOP	:LOOP IF SELECTED	TRAP	C\$CLP1
5454	074402	104406						
5455	074404	013701	075726		MOV T36CNT,R1	:GET FIRST COUNTER		
5456	074410	013702	075730		MOV T36CNU,R2	:GET SECOND COUNTER		
5457	074414	020201			CMP R2,R1	:COMPARE EM		
5458	074416	003406			BLE 300\$:BR, IF VALUES ARE CORRECT (OK)		
5459	074420	005237	002214		INC FATFLG	:ERROR COUNT		
5460	074424				ERRHRD ERRNO,T36NAS,EXPREC	:TAPE NOT AT CORRECT SPEED		
5461	074424	104456					TRAP	C\$ERHRD
5462	074426	001452					.WORD	810
5463	074430	075734					.WORD	T36NAS
5464	074432	015554				</		

074454
074454

•

.....

- _____

—

•

Address	Offset	PC	Instruction	Comment	Label	Trap	Trap Value
5516	074454	104402				TRAP	C\$BSUB
5517	074456	004737	JSR	PC,T36REST			
5518	074462	004737	JSR	PC,T36RT2			
5519	074466	004737	JSR	PC,T36RT3			
5520	074472	012737	MOV	#65000.,T36DLY			
5521	074500	005037	CLR	T36CNT			
5522	074504	004737	JSR	PC,SOFINIT			
5523	074510	103426	BCS	20\$			
	074512		DELAY	250			
	074512	012727				MOV	#250,(PC)+
	074516	000000				.WORD	0
	074520	013727				MOV	L\$DLY,(PC)+
	074524	000000				.WORD	0
	074526	005367				DEC	-6(PC)
	074532	001375				BNE	.-4
	074534	005367				DEC	-22(PC)
	074540	001367				BNE	.-20
5524	074542	005337	DEC	T36DLY			
5525	074546	001356	BNE	10\$			
5526	074550	005237	INC	FATFLG			
5530	074554	010001	MOV	R0,R1			
5531	074556		ERRDF	ERRNO,SFIERR,SFIMSG			
	074556	104455				TRAP	C\$ERDF
	074560	001453				.WORD	811
	074562	003646				.WORD	SFIERR
	074564	012114				.WORD	SFIMSG
5532	074566	013737	MOV	UNITN,T36DSW			
5533	074574	052737	BIS	#BIT5,T36DSW			
5534	074602	012704	MOV	#T36PACKET,R4			
5535	074606	004737	JSR	PC,WRTCHR			
5536	074612	103407	BCS	25\$			
5537	074614	005237	INC	FATFLG			
5541	074620	010001	MOV	R0,R1			
5542	074622		ERRHRD	ERRNO,WRTMSG,SFIMSG			
	074622	104456				TRAP	C\$ERHRD
	074624	001454				.WORD	812
	074626	005052				.WORD	WRTMSG
	074630	012114				.WORD	SFIMSG
5543	074632		CKLOOP				
	074632	104406				TRAP	C\$CLP1
5544	074634	004737	JSR	PC,REWIND			
5545	074640	103407	BCS	30\$			
5546	074642	010004	MOV	R0,R4			
5547	074644	005237	INC	FATFLG			
5551	074650		ERRHRD	ERRNO,T36RWN,PKTSSR			
	074650	104456				TRAP	C\$ERHRD
	074652	001455				.WORD	813
	074654	077153				.WORD	T36RWN
	074656	012126				.WORD	PKTSSR
5552	074660		CKLOOP				
	074660	104406				TRAP	C\$CLP1
5553	074662	013701	MOV	T36BFR+6,R1			
5554	074666	010102	MOV	R1,R2			
5555	074670	052702	BIS	#BIT1,R2			
5556	074674	020102	CMP	R1,R2			
5557	074676	001406	BEQ	40\$			
5558	074700	005237	INC	FATFLG			

5562	074704				ERRHRD	ERRNO,T36BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND	
	074704	104456						TRAP	C\$ERHRD
	074706	001456						.WORD	814
	074710	076647						.WORD	T36BOT
	074712	015554						.WORD	EXPREC
5563	074714			40\$:	CKLOOP			;LOOP IF SELECTED	
	074714	104406						TRAP	C\$CLP1
5564	074716	013737	002174	075600	MOV	UNITN,T36DSW		;SET UP DRIVE NUMBER	
5565	074724	052737	000010	075600	BIS	#BIT3,T36DSW		;TURN OFF BUFFERING CAPABILITY	
5566	074732	012704	075560		MOV	#T36PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS	
5567	074736	004737	010742		JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS	
5568	074742	103407			BCS	50\$;BR, IF COMMAND ISSUED OK	
5569	074744	005237	002214		INC	FATFLG		;ERROR COUNT	
5573	074750	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
5574	074752				ERRHRD	ERRNO,WRTMSG,SFMSG		;WRITE CHARACTERISTICS FAILED	
	074752	104456						TRAP	C\$ERHRD
	074754	001457						.WORD	815
	074756	005052						.WORD	WRTMSG
	074760	012114						.WORD	SFMSG
5575	074762			50\$:	CKLOOP			;LOOP IF SELECTED	
	074762	104406						TRAP	C\$CLP1
5576	074764	012737	003720	075706	MOV	#2000.,T36SZ		;SET UP RECORD SIZE	
5577	074772	013737	003116	075702	MOV	FREE,T36WB		;ADDRESS OF WRITE BUFFER	
5578	075000	012737	140005	075700	MOV	#140005,T36PK3		;WRITE DATA,ACK,CVC=1 COMMAND	
5579	075006	012704	075700		MOV	#T36PK3,R4		;SET UP R4 WITH PACKET ADDRESS	
5580	075012	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND	
5581	075016	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET	
5582	075022	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS	
5583	075026	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED	
5584	075032	020102			CMP	R1,R2		;ARE THEY EQUAL	
5585	075034	001406			BEQ	60\$;BR, IF OK	
5586	075036	005237	002214		INC	FATFLG		;ERROR COUNT	
5590	075042				ERRHRD	ERRNO,WRTERR,PKTSSR		;TSSR INCORRECT AFTER READ DATA	
	075042	104456						TRAP	C\$ERHRD
	075044	001460						.WORD	816
	075046	005107						.WORD	WRTERR
	075050	012126						.WORD	PKTSSR
5591	075052			60\$:	CKLOOP			;LOOP IF SELECTED	
	075052	104406						TRAP	C\$CLP1
5592	075054	012737	000012	075732	MOV	#10.,T36DLY		;DELAY FOR TAPE TO STOP	
5593	075062			70\$:	DELAY	250		;DELAY ROUTINE CALL	
	075062	012727	000250					MOV	#250,(PC)+
	075066	000000						.WORD	0
	075070	013727	002116					MOV	L\$DLY,(PC)+
	075074	000000						.WORD	0
	075076	005367	177772					DEC	-6(PC)
	075102	001375						BNE	.-4
	075104	005367	177756					DEC	-22(PC)
	075110	001367						BNE	.-20
5594	075112	005337	075732		DEC	T36DLY		;BUMP COUNTER DOWN	
5595	075116	001361			BNE	70\$;BR, IF MORE DELAY TO GO	
5596	075120	012737	006642	075706	MOV	#3490.,T36SZ		;SET SIZE OF TRANSFER	
5597	075126	012737	140005	075700	MOV	#140005,T36PK3		;WRITE DATA,ACK,CVC=1 COMMAND	
5598	075134	012704	075700		MOV	#T36PK3,R4		;SET UP R4 WITH PACKET ADDRESS	
5599	075140	005037	075726		CLR	T36CNT		;CLEAR COUNTER	
5600	075144	012737	001750	075732	MOV	#1000.,T36DLY		;SET DROP DEAD COUNTER VALUE	
5601	075152	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND	

5602	075156	016501	000002
5603	075162	032701	000200
5604	075166	001021	
5605	075170	005237	075726
5606	075174		
	075174	012727	000001
	075200	000000	
	075202	013727	002116
	075206	000000	
	075210	005367	177772
	075214	001375	
	075216	005367	177756
	075222	001367	
5607	075224	005337	075732
5608	075230	001352	
5609	075232	012702	000200
5610	075236	020102	
5611	075240	001406	
5612	075242	005237	002214
5616	075246		
	075246	104456	
	075250	001461	
	075252	076575	
	075254	012126	
5617	075256		
	075256	104406	
5618	075260	013737	002174
5619	075266	052737	000030
5620	075274	012704	075560
5621	075300	004737	010742
5622	075304	103407	
5623	075306	005237	002214
5627	075312	010001	
5628	075314		
	075314	104456	
	075316	001462	
	075320	005052	
	075322	012114	
5629	075324		
	075324	104406	
5630	075326	012737	006642
5631	075334	012737	140005
5632	075342	012704	075700
5633	075346	005037	075730
5634	075352	012737	001750
5635	075360	010465	000000
5636	075364	016501	000002
5637	075370	032701	000200
5638	075374	001021	
5639	075376	005237	075730
5640	075402		
	075402	012727	000001
	075406	000000	
	075410	013727	002116
	075414	000000	
	075416	005367	177772
	075422	001375	

```

80$:      MOV      TSSR(R5),R1
          BIT      #SSR,R1
          BNE      90$
          INC      T36CNT
          DELAY    1

          DEC      T36DLY
          BNE      80$
90$:      MOV      #SSR,R2
          CMP      R1,R2
          BEQ      100$
          INC      FATFLG
          ERRHRD   ERRNO,T36WDE,PKTSSR

100$:     CKLOOP

          MOV      UNITN,T36DSW
          BIS      #BIT3:BIT4,T36DSW
          MOV      #T36PACKET,R4
          JSR      PC,WRTCHR
          BCS      110$
          INC      FATFLG
          MOV      R0,R1
          ERRHRD   ERRNO,WRTMSG,SFMSG

110$:     CKLOOP

          MOV      #3490.,T36SZ
          MOV      #14000$,T36PK3
          MOV      #T36PK3,R4
          CLR      T36CNU
          MOV      #1000.,T36DLY
          MOV      R4,TSDB(R5)
120$:     MOV      TSSR(R5),R1
          BIT      #SSR,R1
          BNE      130$
          INC      T36CNU
          DELAY    1

```

```

;GET TSSR CONTENTS
;CHECK FOR SSR SET
;BR, IF SSR IS SET
;BUMP CYCLE COUNTER
;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

;BUMP DROP DEAD COUNTER
;BR, IF THERE IS STILL TIME
;SET UP EXPECTED
;ARE THEY EQUAL
;BR, IF OK
                                ;ERROR COUNT
;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD     817
                                .WORD     T36WDE
                                .WORD     PKTSSR

;LOOP IF SELECTED
                                TRAP      C$CLP1

;SET UP DRIVE NUMBER
;TURN ON THE BUFFERING
;SUBROUTINE NEEDS PACKET ADDRESS
;ISSUE WRITE CHARACTERISTICS
;BR, IF COMMAND ISSUED OK
                                ;ERROR COUNT
;SAVE CONTENTS OF TSSR
;WRITE CHARACTERISTICS FAILED
                                TRAP      C$ERHRD
                                .WORD     818
                                .WORD     WRTMSG
                                .WORD     SFIMSG

;LOOP IF SELECTED
                                TRAP      C$CLP1

;SET SIZE OF TRANSFER
;WRITE DATA,ACK,CVC=1 COMMAND
;SET UP R4 WITH PACKET ADDRESS
;CLEAR COUNTER
;SET DROP DEAD COUNTER VALUE
;ISSUE COMMAND
;GET TSSR CONTENTS
;CHECK FOR SSR SET
;BR, IF SSR IS SET
;BUMP CYCLE COUNTER
;CUT NUMBER OF LOOPS DOWN
                                MOV      #1,(PC)+
                                .WORD    0
                                MOV      L$DLY,(PC)+
                                .WORD    0
                                DEC      -6(PC)
                                BNE      .-4

```

Line	Address	Offset	Hex	Label	Instruction	Comment	Trap	Value
	075424	005367	177756				DEC	-22(PC)
	075430	001367					BNE	.-20
5641	075432	005337	075732		DEC T36DLY	;BUMP DROP DEAD COUNTER		
5642	075436	001352			BNE 120\$;BR, IF THERE IS STILL TIME		
5643	075440	012702	000200	130\$:	MOV #SSR,R2	;SET UP EXPECTED		
5644	075444	020102			CMP R1,R2	;ARE THEY EQUAL		
5645	075446	001406			BEQ 140\$;BR, IF OK		
5646	075450	005237	002214		INC FATFLG	;ERROR COUNT		
5650	075454				ERRHRD ERRNO,WRTErr,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA		
	075454	104456					TRAP	C\$ERHRD
	075456	001463					.WORD	819
	075460	005107					.WORD	WRTErr
	075462	012126					.WORD	PKTSSR
5651	075464			140\$:	CKLOOP	;LOOP IF SELECTED		
	075464	104406					TRAP	C\$CLP1
5652	075466	013701	075726		MOV T36CNT,R1	;GET FIRST COUNTER		
5653	075472	013702	075730		MOV T36CNU,R2	;GET SECOND COUNTER		
5654	075476	020201			CMP R2,R1	;COMPARE EM		
5655	075500	003406			BLE 300\$;BR, IF VALUES ARE CORRECT (OK)		
5656	075502	005237	002214		INC FATFLG	;ERROR COUNT		
5660	075506				ERRHRD ERRNO,T36NAS,EXPREC	;TAPE NOT AT CORRECT SPEED		
	075506	104456					TRAP	C\$ERHRD
	075510	001464					.WORD	820
	075512	075734					.WORD	T36NAS
	075514	015554					.WORD	EXPREC
5661	075516			300\$:	CKLOOP	;LOOP IF SELECTED		
	075516	104406					TRAP	C\$CLP1
5662	075520				ENDSUB			
	075520							
	075520	104403						
5663	075522	023727	002214 000017		CMP FATFLG,#15.	;IS ERROR COUNT AT 25	TRAP	C\$ESUB
5664	075530	103402			BLO 999\$;BR, IF LESS THAN 25		
5665	075532	004737	017262		JSR PC,CKDROP	;TRY TO DROP THE UNIT		
5666	075536			999\$:				

5668
5669
5670

5671 075536 004737 016536
5672 075542 103002
5673 075544 000137 073400
5674 075550
5675 075550
075550 104432
075552 003356

⋮

163\$:

JSR PC,TSTLOOP
BCC 163\$
JMP T36LOOP
EXIT TST

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

TRAP C\$EXIT
.WORD L10070-

5677		;		
5678		;	LOCAL STORAGE FOR THIS TEST	
5679		;		
5681	075560		;	
5683	075560		T36PACKET: .=<.+10>8177770	
5684	075560	100004	.WORD 100004	;COMMAND PACKET FOR TEST
5685	075562	075570	.WORD T36DATA	;WRITE CHARACTERISTICS COMMAND, WITH , ACK
5686	075564	000000	.WORD 0	;ADDRESS OF CHARACTERISTICS BLOCK
5687	075566	000012	.WORD 10.	
5688	075570		T36DATA: .WORD T36BFR	;STARTING VALUE OF BLOCK SIZE
5689	075570	075602	.WORD 0	;CHARACTERISTICS DATA BLOCK
5690	075572	000000	.WORD 20.	;ADDRESS OF MESSAGE BUFFER
5691	075574	000024	.WORD 0	
5692	075576	000000	T36DSW: .WORD 0	;LENGTH OF MESSAGE BUFFER
5693	075600	000000	T36BFR: .BLKW 25.	;SELECT DRIVE 0
5694	075602			;MESSAGE BUFFER
5695				
5696			;	
5697			;	
5699	075670		;	
5701	075670		T36PK2: .=<.+10>8177770	
5702	075670	100006	.WORD 100006	;WRITE SUB SYS MEM COMMAND, AND ACK
5703	075672	075710	.WORD T36BF2	;ADDRESS OF SELECT BLOCK DATA
5704	075674	000000	.WORD 0	
5705	075676	000006	.WORD 6.	;SIZE OF DATA PACKET
5706				
5710	075700		T36PK3: .WORD 100005	;REREAD COMMAND, AND ACK
5711	075700	100005	T36RB: .WORD FREE	;ADDRESS OF WRITE BUFFER
5712	075702		T36WB: .WORD 0	
5713	075702	003116	T36SZ: .WORD 0	;SIZE OF BUFFER (EXTENT)
5714	075704	000000	.EVEN	
5715	075706	000000		
5716				
5717				
5718				
5719				
5720	075710		T36BF2: .BYTE 10	;BSELO AREA
5721	075710	010	T36BS0: .BYTE 200	;BSEL1 AREA
5722	075711	200	T36S2: .WORD 0	;SEL 2 AREA
5723	075712	000000	T36S3: .WORD 0	;DATA AREA
5724	075714	000000		
5725				
5726				
5727			.EVEN	
5728			;	
5729			;	
5730	075716	100205	T36RN: .WORD 100205	;REREAD DATA (NEXT)
5731	075720	100605	T36WDR: .WORD 100605	;REREAD DATA RETRY
5732	075722	102205	T36CON: .WORD 102205	;WRITE CONTINUOUS
5733	075724	177777	.WORD 177777	;END OF DATA
5734				
5735				
5736	075726	000000	T36CNT: .WORD 0	;TAPE TIMER COUNTER STORAGE AREA
5737	075730	000000	T36CNU: .WORD 0	;TAPE TIMER COUNTER STORAGE AREA
5738	075732	000000	T36DLY: .WORD 0	;DELAY COUNTER
5739				

```

: +
: LOCAL TEXT MESSAGES FOR TEST
: -

```

```

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

```

```

T36REST:      SAVREG
               MOV      #T36PACKET,R1
               MOV      #100004,(R1)+
               MOV      #T36DATA,(R1)+
               CLR      (R1)+
               MOV      #10.,(R1)+
               MOV      #T36BFR,(R1)+
               CLR      (R1)+
               MOV      #20.,(R1)+
               ;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
               ;LENGTH OF MESSAGE BUFFER

```

5798	101012	005021		CLR	(R1)+	
5799	101014	012711	000000	MOV	#0,(R1)	;SELECT DRIVE ZERO
5800	101020	012702	000030	MOV	#24,R2	;NUMBER OF LOCATIONS TO BE CLEARED
5801	101024	012762	177777 075602 64\$:	MOV	#177777,T36BFR(R2)	;ALL ONES TO MESSAGE BUFFER
5802	101032	005742		TST	-(R2)	;NEXT LOCATION
5803	101034	022702	000000	CMP	#0,R2	;AT END OF LOOP YET
5804	101040	001371		BNE	64\$;KEEP GOING UNTIL DONE
5805	101042	000207		RTS	PC	;RETURN
5806						
5807						
5808	101044			T36RT2:		
5809	101044			SAVREG		;SAVE THE REGISTERS
5810	101050	012701	075670	MOV	#T36PK2,R1	;START OF THE PACKET
5811	101054	012721	100006	MOV	#100006,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK,
5812	101060	012721	075710	MOV	#T36BF2,(R1)+	;ADDRESS OF DATA BLOCK
5813	101064	005021		CLR	(R1)+	;EXTENDED ADDRESS
5814	101066	012721	000006	MOV	#6,(R1)+	;SIZE OF DATA BLOCK IN BYTES
5815	101072	005021		CLR	(R1)+	
5816	101074	012701	075710	MOV	#T36BF2,R1	;POINT TO DATA SEL AREA
5817	101100	005021		CLR	(R1)+	
5818	101102	005011		CLR	(R1)	
5819	101104	000207		RTS	PC	;RETURN
5820	101106			T36RT3:		
5821	101106			SAVREG		;SAVE REGISTERS
5822	101112	012701	075700	MOV	#T36PK3,R1	;SET UP POINTER ADDRESS
5823	101116	005021		CLR	(R1)+	;COMMAND SPACE
5824	101120	005021		CLR	(R1)+	;ADDRESS OF DATA BLOCK
5825	101122	005021		CLR	(R1)+	;EXTENDED ADDRESS
5826	101124	005011		CLR	(R1)	;SIZE OF DATA TRANSFER BLOCK
5827	101126	000207		RTS	PC	;RETURN
5828	101130			ENDTST		
	101130					
	101130	104401				

L10070: TRAP C\$ETST


```

5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846 101132
101132
5847 101132 012737 006354 002172
5848 101140 004737 017354
5853 101144 012700 105353
5854 101150 004737 016570
5855 101154 012737 000005 002210
5856 101162 005037 102416
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866 101166

          .SBTTL 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
          :SPACE RECORDS COMMAND WITH A RECORD COUNT OF 80 OR MORE, AND A
          :SKIP TAPE MARKS COMMAND WITH A COUNT OF 2 OF 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.
          :-
          BGNTST
          MOV      #EPRT1,EPRTSW      ;PRIMARY ERROR MESSAGE
          JSR      PC,KTOFF           ;TURN KT OFF
          MOV      #TST37ID,R0        ;ASCII MESSAGE TO IDENTIFY TEST
          JSR      PC,TSTSETUP         ;DO INITIAL TEST SETUP
          MOV      #5,LOOPCNT         ;PERFORM 5 ITERATIONS
          CLR      T37CNT             ;CLEAR TAPE RECORD COUNTER
          :+
          :TEST 9, SUBTEST 1
          :+
          :-
          T37LOOP:

```

[illegible]

5910	101406	010102		MOV	R1,R2		:SET UP EXPECTED
5911	101410	052702	000002	BIS	#BIT1,R2		:SET BOT BIT IN EXPECTED
5912	101414	020102		CMP	R1,R2		:DOES EXP = REC'D
5913	101416	001406		BEQ	40\$:BR, IF EQUAL (OK)
5914	101420	005237	002214	INC	FATFLG		:ERROR COUNT
5918	101424			ERRHRD	ERRNO,T37BOT,EXPREC		:TAPE NOT AT BOT AFTER REWIND
	101424	104456				TRAP	C\$ERHRD
	101426	001610				.WORD	904
	101430	103271				.WORD	T37BOT
	101432	015554				.WORD	EXPREC
5919	101434		40\$:	CKLOOP			:LOOP IF SELECTED
	101434	104406				TRAP	C\$CLP1
5920	101436	012703	000144	MOV	#100.,R3		:NUMBER OF RECORDS TO BE WRITTEN
5921	101442	013737	003116	MOV	FREE,T37WB		:STARTING WRITE BUFFER ADDRESS
5922	101450	012737	140005	MOV	#140005,T37PK3		:WRITE DATA,ACK,CVC=1 COMMAND
5923	101456	012704	102370	MOV	#T37PK3,R4		:SET UP R4 WITH PACKET ADDRESS
5924	101462	012737	001130	MOV	#600.,T37SZ		:SET UP RECORD SIZE IN PACKET
5925	101470	010465	000000	MOV	R4,TSDB(R5)		:ISSUE COMMAND
5926	101474	004737	016330	JSR	PC,WAITF		:WAIT FOR SSR TO SET
5927	101500	016501	000002	MOV	TSSR(R5),R1		:GET TSSR CONTENTS
5928	101504	012702	000200	MOV	#SSR,R2		:SET UP EXPECTED
5929	101510	020102		CMP	R1,R2		:ARE THEY EQUAL
5930	101512	001406		BEQ	70\$:BR, IF OK
5931	101514	005237	002214	INC	FATFLG		:ERROR COUNT
5935	101520			ERRHRD	ERRNO,T37WDC,PKTSSR		:TSSR INCORRECT AFTER WRITE DATA
	101520	104456				TRAP	C\$ERHRD
	101522	001611				.WORD	905
	101524	104131				.WORD	T37WDC
	101526	012126				.WORD	PKTSSR
5936	101530		70\$:	CKLOOP			:LOOP IF SELECTED
	101530	104406				TRAP	C\$CLP1
5937	101532	005303		DEC	R3		:DEC RECORD COUNTER
5938	101534	001345		BNE	65\$:BR, IF MORE RECORDS TO WRITE
5939	101536	004737	011074	JSR	PC,REWIND		:CALL TAPE REWIND COMMAND
5940	101542	103411		BCS	130\$:BR, IF NO PROBLEM
5941	101544	016501	000002	MOV	TSSR(R5),R1		:GET TSSR CONTENTS
5942	101550	010004		MOV	R0,R4		:GET PACKET ADDRESS
5943	101552	005237	002214	INC	FATFLG		:ERROR COUNT
5947	101556			ERRHRD	ERRNO,T37RWN,PKTSSR		:REWIND NOT ACCEPTED
	101556	104456				TRAP	C\$ERHRD
	101560	001612				.WORD	906
	101562	103575				.WORD	T37RWN
	101564	012126				.WORD	PKTSSR
5948	101566		130\$:	CKLOOP			:LOOP IF SELECTED
	101566	104406				TRAP	C\$CLP1
5949	101570	013701	102300	MOV	T37BFR+6,R1		:PICK UP XST0
5950	101574	010102		MOV	R1,R2		:SET UP EXPECTED
5951	101576	052702	000002	BIS	#BIT1,R2		:SET BOT BIT IN EXPECTED
5952	101602	020102		CMP	R1,R2		:DOES EXP = REC'D
5953	101604	001406		BEQ	140\$:BR, IF EQUAL (OK)
5954	101606	005237	002214	INC	FATFLG		:ERROR COUNT
5958	101612			ERRHRD	ERRNO,T37BOT,EXPREC		:TAPE NOT AT BOT AFTER REWIND
	101612	104456				TRAP	C\$ERHRD
	101614	001613				.WORD	907
	101616	103271				.WORD	T37BOT
	101620	015554				.WORD	EXPREC
5959	101622		140\$:	CKLOOP			:LOOP IF SELECTED

Address	Offset	Hex	Dec	Label	Instruction	Comment	Trap	Trap Comment
5960	101622	104406					TRAP	C\$CLP1
5961	101624	012704	102370		MOV #T37PK3,R4	;SET UP PACKET ADDRESS		
5962	101630	012737	000037	102372	MOV #31,T37RB	;SET UP RECORDS TO SPACE OVER		
5963	101636	012737	140010	102370	MOV #140010,T37PK3	;ACK,CVC=1,SPACE FORWARD COMMAND		
5964	101644	010465	000000		MOV R4,TSDDB(R5)	;ISSUE COMMAND		
5965	101650	005237	102416	150\$:	INC T37CNT	;BUMP TIMER		
	101654			152\$:	DELAY 1	;DELAY ABOUT 100US		
	101654	012727	000001				MOV #1,(PC)+	
	101660	000000					.WORD 0	
	101662	013727	002116				MOV L\$DLY,(PC)+	
	101666	000000					.WORD 0	
	101670	005367	177772				DEC -6(PC)	
	101674	001375					BNE -4	
	101676	005367	177756				DEC -22(PC)	
	101702	001367					BNE -20	
5966	101704	016501	000002		MOV TSSR(R5),R1	;GET TSSR		
5967	101710	032701	000200		BIT #SSR,R1	;CHECK FOR TSSR'S SSR SET		
5968	101714	001755			BEQ 152\$;KEEP COUNTING UNTIL SET		
5969	101716	012702	000200		MOV #SSR,R2	;SET UP EXPECTED		
5970	101722	020201			CMP R2,R1	;WAS EVERYTHING OK		
5971	101724	001406			BEQ 160\$;BR, IF ALL IS WELL		
5972	101726	005237	002214		INC FATFLG	;ERROR COUNT		
5976	101732				ERRHRD	ERRNO,T37SCF,PKTSSR	;SPACE FORWARD DIDN'T WORK OUT	
	101732	104456					TRAP	C\$ERHRD
	101734	001614					.WORD 908	
	101736	105037					.WORD T37SCF	
	101740	012126					.WORD PKTSSR	
5977	101742			160\$:	CKLOOP	;LOOP IF SELECTED		
	101742	104406					TRAP	C\$CLP1
5978	101744	004737	011074		JSR PC,REWIND	;CALL TAPE REWIND COMMAND		
5979	101750	103411			BCS 170\$;BR, IF NO PROBLEM		
5980	101752	010004			MOV R0,R4	;GET PACKET ADDRESS		
5981	101754	016501	000002		MOV TSSR(R5),R1	;GET STATUS FROM TSSR		
5982	101760	005237	002214		INC FATFLG	;ERROR COUNT		
5986	101764				ERRHRD	ERRNO,T37RWN,PKTSSR	;REWIND NOT ACCEPTED	
	101764	104456					TRAP	C\$ERHRD
	101766	001615					.WORD 909	
	101770	103575					.WORD T37RWN	
	101772	012126					.WORD PKTSSR	
5987	101774			170\$:	CKLOOP	;LOOP IF SELECTED		
	101774	104406					TRAP	C\$CLP1
5988	101776	013701	102300		MOV T37BFR+6,R1	;PICK UP XST0		
5989	102002	010102			MOV R1,R2	;SET UP EXPECTED		
5990	102004	052702	000002		BIS #BIT1,R2	;SET BOT BIT IN EXPECTED		
5991	102010	020102			CMP R1,R2	;DOES EXP = REC'D		
5992	102012	001406			BEQ 175\$;BR, IF EQUAL (OK)		
5993	102014	005237	002214		INC FATFLG	;ERROR COUNT		
5997	102020				ERRHRD	ERRNO,T37BOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND	
	102020	104456						

Address	Offset	Hex	Dec	Label	Instruction	Comment	Register/Value
6002	102052	010465	000000		MOV	R4,TSDB(R5)	
6003	102056	005237	102420	250\$:	INC	T37CNU	
6004	102062			252\$:	DELAY	1	
	102062	012727	000001				
	102066	000000					
	102070	013727	002116				
	102074	000000					
	102076	005367	177772				
	102102	001375					
	102104	005367	177756				
	102110	001367					
6005	102112	016501	000002		MOV	TSSR(R5),R1	
6006	102116	032701	000200		BIT	#SSR,R1	
6007	102122	001755			BEQ	252\$	
6008	102124	012702	000200		MOV	#SSR,R2	
6009	102130	020201			CMP	R2,R1	
6010	102132	001406			BEQ	260\$	
6011	102134	005237	002214		INC	FATFLG	
6015	102140				ERRHRD	ERRNO,T37SCF,PKTSSR	
	102140	104456					
	102142	001617					
	102144	105037					
	102146	012126					
6016	102150			260\$:	CKLOOP		
	102150	104406					
6017	102152	013701	102416		MOV	T37CNT,R1	
6018	102156	013702	102420		MOV	T37CNU,R2	
6019	102162	042701	000077		BIC	#000077,R1	
6020	102166	042702	000077		BIC	#000077,R2	
6021	102172	020102			CMP	R1,R2	
6022	102174	003406			BLE	300\$	
6023	102176	005237	002214		INC	FATFLG	
6027	102202				ERRHRD	ERRNO,T37TIM,EXPREC	
	102202	104456					
	102204	001620					
	102206	103364					
	102210	015554					
6028	102212			300\$:	CKLOOP		
	102212	104406					
6029	102214				ENDSUB		
	102214	104403					
6030	102216	023727	002214	000017	CMP	FATFLG,#15.	
6031	102224	103402			BLO	999\$	
6032	102226	004737	017262		JSR	PC,CKDROP	
6033	102232			999\$:			
6034				:			
6035				:			
6036				:			
6037	102232	004737	016536		JSR	PC,TSTLOOP	
6038	102236	103002			BCC	163\$	
6039	102240	000137	101166		JMP	T37LOOP	
6040	102244			163\$:			
6041	102244				EXIT	TST	
	102244	104432					
	102246	003304					

6043			;		
6044			;	LOCAL STORAGE FOR THIS TEST	
6045			;		
6049	102250		T37PACKET:		;COMMAND PACKET FOR TEST
6050	102250	100004	.WORD	100004	;WRITE CHARACTERISTICS COMMAND, WITH , ACK
6051	102252	102260	.WORD	T37DATA	;ADDRESS OF CHARACTERISTICS BLOCK
6052	102254	000000	.WORD	0	
6053	102256	000012	.WORD	10.	;STARTING VALUE OF BLOCK SIZE
6054	102260		T37DATA:		;CHARACTERISTICS DATA BLOCK
6055	102260	102272	.WORD	T37BFR	;ADDRESS OF MESSAGE BUFFER
6056	102262	000000	.WORD	0	
6057	102264	000024	.WORD	20.	;LENGTH OF MESSAGE BUFFER
6058	102266	000000	.WORD	0	
6059	102270	000000	T37DSW:	.WORD 0	;SELECT DRIVE 0
6060	102272		T37BFR:	.BLKW 25.	;MESSAGE BUFFER
6061			;		
6062			;	WRITE SUBSYSTEM MEMORY COMMAND PACKET	
6063			;		
6065		102360		.=<. +10>B177770	
6067	102360		T37PK2:		
6068	102360	100006	.WORD	100006	;WRITE SUB SYS MEM COMMAND, AND ACK
6069	102362	102400	.WORD	T37BF2	;ADDRESS OF SELECT BLOCK DATA
6070	102364	000000	.WORD	0	
6071	102366	000006	.WORD	6.	;SIZE OF DATA PACKET
6072					
6076	102370		T37PK3:		
6077	102370	100005	.WORD	100005	;REREAD COMMAND, AND ACK
6078	102372		T37RB:		
6079	102372	003116	T37WB:	.WORD FREE	;ADDRESS OF WRITE BUFFER
6080	102374	000000	.WORD	0	
6081	102376	000000	T37SZ:	.WORD 0	;SIZE OF BUFFER (EXTENT)
6082			.EVEN		
6083			;		
6084			;		
6085			;		
6086	102400		T37BF2:		
6087	102400	010	T37BS0:	.BYTE 10	;BSELO AREA
6088	102401	200	T37BS1:	.BYTE 200	;BSEL1 AREA
6089	102402	000000	T37S2:	.WORD 0	;SEL 2 AREA
6090	102404	000000	T37S3:	.WORD 0	;DATA AREA
6091			;		
6092			;		
6093			.EVEN		
6094			;	TAPE MOTION PACKET COMMAND VALUES	
6095					
6096	102406	100205	T37RN:	.WORD 100205	;REREAD DATA (NEXT)
6097	102410	100605	T37WDR:	.WORD 100605	;REREAD DATA RETRY
6098	102412	102205	T37CON:	.WORD 102205	;WRITE CONTINUOUS
6099	102414	177777	.WORD	177777	;END OF DATA
6100					
6101			;		
6102	102416	000000	T37CNT:	.WORD 0	;TAPE TIMER COUNTER STORAGE AREA
6103	102420	000000	T37CNU:	.WORD 0	;TAPE TIMER COUNTER STORAGE AREA
6104	102422	000000	T37DLY:	.WORD 0	;DELAY COUNTER
6105					

```

6107
6108
6109
6110
6111
6112
6113 102424 124 141 160 T37WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
6114 102512 124 123 123 T37RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
6115 102561 122 105 122 T37RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
6116 102656 120 117 123 T37SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
6117 102740 122 111 102 T37LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
6118 103010 124 123 123 T37WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
6119 103065 111 154 154 T37LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
6120 103146 122 105 122 T37SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
6121 103202 124 123 123 T37WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command, At BOT'
6122 103271 124 141 160 T37BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
6123 103364 127 122 111 T37TIM: .ASCIZ 'WRITE DATA RETRY'S Erase Tape Not Long Enough'
6124 103441 122 105 122 T37EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
6125 103520 124 123 123 T37TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
6126 103575 122 145 167 T37RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
6127 103644 122 101 115 T37RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
6128 103717 124 123 123 T37AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
6129 103766 104 162 151 T37OFL: .ASCIZ 'Drive 7 Select Failed To Set 'OFL' In TSSR'
6130 104041 124 123 123 T37WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
6131 104131 124 123 123 T37WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
6132 104204 103 126 103 T37VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
6133 104257 124 123 102 T37BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
6134 104332 127 122 111 T37WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
6135 104421 122 145 141 T37LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
6136 104503 122 145 141 T37LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
6137 104565 122 145 163 T37PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
6138 104653 122 145 141 T37TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
6139 104741 127 122 111 T37NEF: .ASCIZ 'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
6140 105037 124 123 123 T37SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
6141 105114 124 123 123 T37TSA: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
6142 105176 124 123 123 T37WRF: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command'
6143 105256 104 141 164 T37DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
6144 105353 106 165 156 T37ID: .ASCIZ 'Function Timing'
6145
6146
6147
6148
6149
6150
6151
6152
6153 105374
6154 105374
6155 105400 012701 102250
6156 105404 012721 100004
6157 105410 012721 102260
6158 105414 005021
6159 105416 012721 000012
6160 105422 012721 102272
6161 105426 005021
6162 105430 012721 000024
6163 105434 005021
  
```

```

;+
;LOCAL TEXT MESSAGES FOR TEST
;-
  
```

.EVEN

```

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

T37REST:

```

      SAVREG
      MOV      #T37PACKET,R1
      MOV      #100004,(R1)+
      MOV      #T37DATA,(R1)+
      CLR      (R1)+
      MOV      #10,(R1)+
      MOV      #T37BFR,(R1)+
      CLR      (R1)+
      MOV      #20,(R1)+
      CLR      (R1)+

      ;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
      ;LENGTH OF MESSAGE BUFFER
  
```

6164	105436	012711	000000	MOV	#0,(R1)	;SELECT DRIVE ZERO
6165	105442	012702	000030	MOV	#24,R2	;NUMBER OF LOCATIONS TO BE CLEARED
6166	105446	012762	177777	MOV	#177777,T3BFR(R2)	;ALL ONES TO MESSAGE BUFFER
6167	105454	005742		TST	-(R2)	;NEXT LOCATION
6168	105456	022702	000000	CMP	#0,R2	;AT END OF LOOP YET
6169	105462	001371		BNE	64\$;KEEP GOING UNTIL DONE
6170	105464	000207		RTS	PC	;RETURN
6171						
6172						
6173	105466			T37RT2:	SAVREG	;SAVE THE REGISTERS
6174	105466			MOV	#T37PK2,R1	;START OF THE PACKET
6175	105472	012701	102360	MOV	#100006,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK,
6176	105476	012721	100006	MOV	#T37BF2,(R1)+	;ADDRESS OF DATA BLOCK
6177	105502	012721	102400	CLR	(R1)+	;EXTENDED ADDRESS
6178	105506	005021		MOV	#6,(R1)+	;SIZE OF DATA BLOCK IN BYTES
6179	105510	012721	000006	CLR	(R1)+	
6180	105514	005021		MOV	#T37BF2,R1	;POINT TO DATA SEL AREA
6181	105516	012701	102400	CLR	(R1)+	
6182	105522	005021		CLR	(R1)	
6183	105524	005011		CLR	(R1)	
6184	105526	000207		RTS	PC	;RETURN
6185	105530			T37RT3:	SAVREG	;SAVE REGISTERS
6186	105530			MOV	#T37PK3,R1	;SET UP POINTER ADDRESS
6187	105534	012701	102370	CLR	(R1)+	;COMMAND SPACE
6188	105540	005021		CLR	(R1)+	;ADDRESS OF DATA BLOCK
6189	105542	005021		CLR	(R1)+	;EXTENDED ADDRESS
6190	105544	005021		CLR	(R1)	;SIZE OF DATA TRANSFER BLOCK
6191	105546	005011		RTS	PC	;RETURN
6192	105550	000207		ENDTST		
6193	105552					
	105552					
	105552	104401				
6194	105554			ENDMOD		

L10073: TRAP C\$ETST


```

1          .TITLE   TSV6 - PARAMETER CODING
7
12
18
19 105554    BGNMOD   TSV6
105554      TSV6::

20
21
22          .SBTTL   HARDWARE PARAMETER CODING SECTION
23
24          ;++
25          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
26          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
27          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
28          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
29          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
30          ; WITH THE OPERATOR.
31          ;--
32 105554    BGNHRD
105554      .WORD L10075-L$HARD/2
105556      L$HARD::

33
34 105556    GPRMA    HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
105556      .WORD    T$CODE
105560      .WORD    HPM1
105562      .WORD    T$LOLIM
105564      .WORD    T$HILIM
35 105566    GPRMA    HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
105566      .WORD    T$CODE
105570      .WORD    HPM2
105572      .WORD    T$LOLIM
105574      .WORD    T$HILIM
36          ;GPRMD   HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
37 105576    ENDHRD
          .EVEN

          L10075:
38 105576    104      105      126  HPM1:  .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
39 105632    111      116      124  HPM2:  .ASCIZ  'INTERRUPT VECTOR '
40 105656    111      116      124  HPM3:  .ASCIZ  'INTERRUPT PRIORITY '
41          .EVEN
42

```

```

44      .SBTTL  SOFTWARE PARAMETER CODING SECTION
45
46      ;++
47      ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
48      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
49      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
50      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
51      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
52      ; WITH THE OPERATOR.
53      ;--
54      105706      BGNSFT
55      105706      .WORD L10076-L$SOFT/2
56      105710      L$SOFT::
57      105710      GPRML  SPM1,0,-1,YES      ; GET TRANSPORT TEST FLAG.
58      105710      GPRML  SPM4,2,-1,YES      ; GET ITERATION CONTROL.
59      105710      .WORD  T$CODE
60      105710      .WORD  SPM4
61      105710      .WORD  -1
62      105710      GPRMD  SPM6,4,D,7777,0,7777,YES      ; GET LOCAL ERROR LIMIT
63      105710      GPRMD  SPM7,6,D,7777,0,7777,YES      ; GET GLOBAL ERROR LIMIT
64      105710      ENDSFT
65      105710      .EVEN
66      105710      L10076:
67
68      105716      105      116      101  SPM1:  .ASCIZ  'ENABLE TRANSPORT TESTS '
69      105746      111      116      110  SPM4:  .ASCIZ  'INHIBIT ITERATIONS '
70      105776      120      105      122  SPM6:  .ASCIZ  'PER TEST ERROR LIMIT '
71      106026      120      105      122  SPM7:  .ASCIZ  'PER UNIT ERROR LIMIT '
72      .SBTTL  PATCH AREA
73
74      ;
75      ; FINALLY A GENEROUS PATCH AREA.
76      ;
77      ; AND AN ADJUSTMENT TO ACCOUNT FOR THE 'LASTAD BIT7' HACK
78      ; DESCRIBED IN 'SUPPRG.MEM' (FOR REV C).
79      ;
80      PATCH::
81
82      .BLKW  32.
83
84      .=.!377+1
85      LASTAD      ;SET LAST USED ADDRESS.
86      .EVEN
87      .WORD  0
88      .WORD  0
89
90      I$LAST::
91      ENDMOD
92      .END

```

ADDSR	012206	G	C\$AU	=	000052	DEVDR0	023456	FRESIZ	003120	G	INTFLA	016225				
ADR	=	000020	G	C\$AUTO	=	000061	DEVNRD	023375	FUSI	004113	INTMAS	016224				
AMBTSS	006713		C\$BRK	=	000022	DEVNXR	023313	F\$AU	=	000015	INTR	016276				
ASSEMB	=	000010	C\$BSEG	=	000004	DEVONL	023243	F\$AUTO	=	000020	INTREC	002216				
A1716	=	000003	C\$BSUB	=	000002	DEVSUM	023206	F\$BGN	=	000040	INTVEC	016226				
BADDAT	003150	G	C\$CEFG	=	000045	DFPTBL	002150	F\$CLEA	=	000007	INTX	004274				
BADSSR	015760	G	C\$CLCK	=	000062	DIAGMC	=	000000	F\$DU	=	000016	INVERT	021266			
BDVPCR	=	177520	G	C\$CLEA	=	000012	DICED	=	000001	F\$END	=	000041	IOKCKI	=	000200	
BENBSW	002222	G	C\$CLOS	=	000035	DSBINT	016264	F\$HARD	=	000004	IOKSTP	=	000001			
BIE	=	040000	C\$CLP1	=	000006	DUAD12	004637	F\$HW	=	000013	IPRI	002204	G			
BIT0	=	000001	G	C\$CVEC	=	000036	DUFLG	003104	F\$INIT	=	000006	ISR	=	000100	G	
BIT00	=	000001	G	C\$DCLN	=	000044	DUMMY	003054	F\$JMP	=	000050	IVEC	002202	G		
BIT01	=	000002	G	C\$DODU	=	000051	EF.CON	=	000036	F\$MOD	=	000000	IXE	=	004000	G
BIT02	=	000004	G	C\$DRPT	=	000024	EF.NEW	=	000035	F\$MSG	=	000011	ISAU	=	000041	
BIT03	=	000010	G	C\$DU	=	000053	EF.PWR	=	000034	F\$PROT	=	000021	ISAUTO	=	000041	
BIT04	=	000020	G	C\$EDIT	=	000003	EF.RES	=	000037	F\$PWR	=	000017	ISCLN	=	000041	
BIT05	=	000040	G	C\$ERDF	=	000055	EF.STA	=	000040	F\$RPT	=	000012	ISDU	=	000041	
BIT06	=	000100	G	C\$ERHR	=	000056	EMAXDU	017057	F\$SEG	=	000003	ISHRD	=	000041		
BIT07	=	000200	G	C\$ERRO	=	000060	EN	=	000000	F\$SOFT	=	000005	ISINIT	=	000041	
BIT08	=	000400	G	C\$ERSF	=	000054	ENAIN	016232	F\$SRV	=	000010	ISMOD	=	000041		
BIT09	=	001000	G	C\$ERSO	=	000057	ENVIRN	020710	F\$SUB	=	000002	ISMSG	=	000041		
BIT1	=	000002	G	C\$ESCA	=	000010	EPRTSW	002172	F\$SW	=	000014	ISPROT	=	000040		
BIT10	=	002000	G	C\$ESG	=	000005	EPRT1	006354	F\$TEST	=	000001	ISPTAB	=	000041		
BIT11	=	004000	G	C\$ESUB	=	000003	EPRT2	006413	GDDAT	003152	G	ISPWR	=	000041		
BIT12	=	010000	G	C\$ETST	=	000001	ERCM	012013	GERRMA	002166	G	ISRPT	=	000041		
BIT13	=	020000	G	C\$EXIT	=	000032	ERRHI	002230	GGETPAT	020254	G	ISSEG	=	000041		
BIT14	=	040000	G	C\$GETB	=	000026	ERRK	017036	GGETSEL	020336	G	ISSETU	=	000041		
BIT15	=	100000	G	C\$GETW	=	000027	ERRLO	002232	GSCNT0	=	000200	ISSFT	=	000041		
BIT2	=	000004	G	C\$GMAN	=	000043	ERRNO	=	001620	G\$DELM	=	000372	ISSRV	=	000041	
BIT3	=	000010	G	C\$GPHR	=	000042	ERRVEC	=	000004	G\$DISP	=	000003	ISSUB	=	000041	
BIT4	=	000020	G	C\$GPLO	=	000030	ERTABE	003370	G\$EXCP	=	000400	ISTST	=	000041		
BIT5	=	000040	G	C\$GPRI	=	000040	ERTABL	003170	G\$HILI	=	000002	JSJMP	=	000167		
BIT6	=	000100	G	C\$INIT	=	000011	ESUM	017040	G\$LOLI	=	000001	KIPAR0	=	172340		
BIT7	=	000200	G	C\$INLP	=	000020	EVL	=	000004	G\$NO	=	000000	KIPAR1	=	172342	
BIT8	=	000400	G	C\$MANI	=	000050	EXBCNT	=	000010	G\$OFFS	=	000400	KIPAR2	=	172344	
BIT9	=	001000	G	C\$MEM	=	000031	EXPBRE	015562	G\$OFFSI	=	000376	KIPAR3	=	172346		
BOE	=	000400	G	C\$MSG	=	000023	EXPD	002224	G\$PRMA	=	000001	KIPAR4	=	172350		
BRINIT	004453		C\$OPEN	=	000034	EXPGOT	004527	G\$PRMD	=	000002	KIPAR5	=	172352			
BSELO	=	000000	C\$PNTB	=	000014	EXPGT2	004563	G\$PRML	=	000000	KIPAR6	=	172354			
BSEL1	=	000001	C\$PNTF	=	000017	EXPMSG	002314	G\$RADA	=	000140	KIPAR7	=	172356			
CHKAMB	016124		C\$PNTS	=	000016	EXPREC	015554	G\$RADB	=	000000	KIPDR0	=	172300			
CHKMAN	020560	G	C\$PNTX	=	000015	EXTA	005766	G\$RADL	=	000040	KIPDR1	=	172302			
CHKTSS	016416		C\$QIO	=	000377	EXTEND	005764	G\$RADL	=	000120	KIPDR2	=	172304			
CKDROP	017262		C\$RDBU	=	000007	EXTFEA	002220	G\$RADO	=	000020	KIPDR3	=	172306			
CKEMAX	017162		C\$REFG	=	000047	E\$END	=	002100	G\$XFER	=	000004	KIPDR4	=	172310		
CKMSG	011440	G	C\$RESE	=	000033	E\$LOAD	=	000035	G\$YES	=	000010	KIPDR5	=	172312		
CKMSG2	011560	G	C\$REVI	=	000003	FATERR	=	000060	HIADDR	=	001400	KIPDR6	=	172314		
CKRAM	011174	G	C\$RFLA	=	000021	FATFLG	002214	G	HOE	=	100000	KIPDR7	=	172316		
CKRAM2	011304	G	C\$RPT	=	000025	FERCM	012002	HPM1	105576			KTENAB	003126	G		
CMDPKT	021340	G	C\$SEFG	=	000046	FIFEXP	012250	G	HPM2	105632		KTFLG	003124	G		
CMPMEM	017740		C\$SPRI	=	000041	FIF1MS	012322	HPM3	105656			KTINIT	021134			
CONFIG	017330		C\$SVEC	=	000037	FIF2MS	012371	IBE	=	010000	G	KTOFF	017354			
COUNT	002302	G	C\$TPRI	=	000013	FILLME	017502	IDU	=	000040	G	KTON	017336			
CSRADD	002200	G	DATA	002304	G	FNOINT	004211	IER	=	020000	G	LERRMA	002164	G		
CTAB	003156	G	DATASC	020312		FORCER	002170	IFAU	004252			LISTAL	=	000001		
CTABE	003170	G	DEBUGM	011712		FREE	003116	INCERK	017124			LOE	=	040000	G	
CTABM	003156	G	DEVCNT	002212	G	FREEHI	003122	INTCPC	016230			LOOPCN	002210	G		

LODPCO 013206	L10001 002170	L10073 105552	OSERRT= 000000	PST32W 003144 G
LOOPFL 003154 G	L10002 003762	L10074 102214	OSGNSW= 000001	PUNIT 022364
LOT = 000010 G	L10003 012124	L10075 105576	OSPOIN= 000001	PW.D11= 000021
LSACP 002110 G	L10004 012142	L10076 105716	OSSETU= 000000	PW.D13= 000022
LSAPT 002036 G	L10005 012160	MEMADD 014034 G	PASRPT 022134	PW.D22= 000020
LSAU 022432 G	L10006 012166	MEMCK 021356 G	PATCH 106056 G	PW.NOP= 000000
LSAUT 002070 G	L10007 012204	MENASC 020527	PATDAT 020310	PW.NO1= 000023
LSAUTO 022636 G	L10010 012222	MENERR 020454	PC.ERA= 002400	PW.RDE= 000024
LSCCP 002106 G	L10011 012246	MENRES 020556	PC.IER= 002000	PW.RDR= 000001
LSCLEA 022716 G	L10012 012320	MMVEC = 000250	PC.NOO= 001000	PW.RDS= 000005
LSCO 002032 G	L10013 012470	MSA.FR= 000006	PC.REL= 000000	PW.RFI= 000003
LSDEPO 002011 G	L10014 013204	MSA.NO= 000000	PC.REW= 000400	PW.WCT= 000006
LSDESC 003402 G	L10015 014032	MSA.NR= 000004	PKBCNT= 000006	PW.WFI= 000004
LSDESP 002076 G	L10016 014054	MSA.VO= 000002	PKHI = 000004	PW.WFM= 000007
LSDEVP 002060 G	L10017 015560	MSGEXP 012224 G	PKLOW = 000002	PW.WMI= 000010
LSDISP 002124 G	L10020 015566	MSGLOO 013144 G	PKTADD 007632	PW.WNP= 000011
LSDLY 002116 G	L10021 015574	MSGSTA 012430 G	PKTFRM 007574	PW.WTR= 000002
LSDTP 002040 G	L10022 015606	MSGSUB 014022 G	PKTGET 012144 G	P.ACK = 100000
LSDTYP 002034 G	L10023 015630	MS.ATT= 000006	PKTMES 012170 G	P.CMD = 000037
LSDU 022530 G	L10024 015656	MS.EXT= 000200	PKTRAM 004741 G	P.CONT= 000012
LSDUT 002072 G	L10025 016016	MS.RSD= 000001	PKTSSR 012126 G	P.CVC = 040000
LSDVTY 003374 G	L10026 016326	MS.RSF= 000020	PNT = 001000 G	P.FMT = 000140
LSEF 002052 G	L10030 022362	MS.RST= 000010	PRAMPK 014056	P.FORM= 000011
LSENV1 002044 G	L10031 022526	M8186 005550	PRASC 014603	P.GETS= 000017
LSETP 002102 G	L10032 022634	M8189 005641	PRBEXP 015550	P.IE = 000200
LSEXP1 002046 G	L10033 022714	NBA = 002000	FRBMSG 015416	P.INIT= 000013
LSEXP4 002064 G	L10034 022742	NEWPAS 022070	PRBREC 015552	P.MODE= 007400
LSEXP5 002066 G	L10035 023204	NODEV 003106 G	PRBTOT 015503	P.OPP = 020000
LSHARD 105556 G	L10036 032332	NOINIT 004331	PRBYTE 015202 G	P.POSI= 000010
LSHIME 002120 G	L10037 024170	NOINTR 004215	PRI = 002000 G	P.READ= 000001
LSHPCP 002016 G	L10040 024712	NOITS 002162 G	PRIADD 010236	P.SWB = 010000
LSHPTP 002022 G	L10041 025436	NOMAN 020614	PRIAO 010306	P.WRIT= 000005
LSHW 002150 G	L10042 026260	NOMEM 005454	PRIBXO 007670 G	P.WRTC= 000004
LSICP 002104 G	L10043 041430	NP.IR = 000200	PRIEQU 010136	P.WRTS= 000006
LSINIT 021636 G	L10044 033734	NP.LOO= 000040	PRIPKT 007446 G	QVP 002176 G
LSLADP 002026 G	L10045 035360	NP.OUT= 000100	PRIRAM 010144	RAMASC 014236
LSLAST 106404 G	L10046 035754	NP.WRP= 000020	PRITAD 010352	RAMDAT 002234 G
LSLOAD 002100 G	L10047 036440	NSI 004146	PRITSS 006020	RAMERR 015570 G
LSLUN 002074 G	L10050 046766	NSINIT 004403	PRITO 010434	RAMEXP 015610 G
LSMREV 002050 G	L10051 042322	NUL 004523	PRITI 010477	RAMFOR 010174
LSNAME 002000 G	L10052 043134	NULCR 004524	PRIXOR 010020 G	RAMSJZ 002274 G
LS'PRIO 002042 G	L10053 053044	NXM = 004000	PRI00 = 000000 G	RAMTAD 015576 G
LSPROT 021626 G	L10054 047642	NXMFLG 003130 G	PRI01 = 000040 G	RCVH1A 002276 G
LSPRT 002112 G	L10055 050452	NXMHI 003134 G	PRI02 = 000100 G	RCVLOA 002300 G
LSREPP 002062 G	L10056 051266	NXMLO 003132 G	PRI03 = 000140 G	RDERR 005202
LSREV 002010 G	L10057 056040	NXMTST 021532	PRI04 = 000200 G	RECMSG 002460 G
LSRPT 022744 G	L10060 054506	NXR 003734	PRI05 = 000240 G	RECV 002226 G
LSOFT 105710 G	L10061 063412	NXRERR 005732 G	PRI06 = 000300 G	REGSAV 020220
LSGPC 002056 G	L10062 060476	NXR 003773	PRI07 = 000340 G	RETER 005366
LSGPCP 002020 G	L10063 073342	NXTU 022102	PRMESS 014322	REWIND 011074 G
LSPTP 002024 G	L10064 064504	OFL = 000100	PRMNO 002312 G	RMCHBE= 000167
LSSTA 002030 G	L10065 065564	ONEFIL= 000000	PRMSG 014632 G	RMCHEN= 000200
LSW 002160 G	L10066 066426	OSAPTS= 000000	PRMSG0 015012	RMMSGB= 000215
LSTEST 002114 G	L10067 067330	OSAU = 000001	PRMSG1 015057	RMMSGC= 000234
LSIML 002014 G	L10070 101130	OSBGJR= 000001	PRMSG2 015115	RMPKTB= 000201
LSUNIT 002012 G	L10071 074436	OSBGNS= 000001	PROASC 014500	RMPKTE= 000210
L10000 002156	L10072 075520	OSDU = 000001	PR1ASC 014545	RMR = 010000

RWPACK 011170	S2.INR= 000020	TSEXCP= 000000	T29CON 026462	T30BOT 040041
SC = 100000	S2.OUT= 000040	T\$FLAG= 000040	T29DAT 026330	T30BS0 036630
SCE = 020000	S2.UND= 000003	T\$GMAN= 000000	T29DLY 026500	T30BS1 036631
SCHERR 005274	TBLEND= 003054 G	T\$HILI= 000776	T29DSW 026340	T30CNT 036650
SCME 005007	TCOASC 006554	T\$LAST= 000001	T29DTA 030043	T30CNU 036652
SDELAY 010740	TCOCOD 006754	T\$LOLI= 000000	T29EOT 030131	T30DAT 036510
SELASC 020522	TEMP1 003110 G	T\$LSYM= 010000	T29LON 031225	T30DLY 036656
SELDAT= 000004	TEMP2 003112 G	T\$LTNO= 000011	T29LOO 023556	T30DSW 036520
SEL2 = 000002	TERCLS= 000016	T\$NEST= 177777	T29LOP 031307	T30DTA 041134
SETMAP 017376	TESTNO= 000011	T\$NS0 = 000000	T29LOQ 027426	T30DTR 041070
SETU 022166	TEXASC 006513	T\$NS1 = 000005	T29LOR 027301	T30ETM 036516
SFFMSG 012162 G	TFCASC 006615	T\$NS2 = 000002	T29NEF 026630	T30FCN 036654
SFHERR 003701	TIMEXP 015632 G	T\$PTNU= 000000	T29NEQ 031545	T30IBT 037031
SFIERR 003646	TIMSGO 015660	T\$SAVL= 177777	T29OFL 026502	T30IBU 036660
SFIMSG 012114 G	TINERR 012101	T\$SEGL= 177777	T29OF7 030515	T30IMV 036636
SFPTBL 002160 G	TMPBFR 002624 G	T\$SUBN= 000001	T29PAC 026320	T30LOO 032360
SIFLAG 003146 G	TNAM 016764	T\$TAGL= 177777	T29PBF 031371	T30LOQ 037630
SIMSG 012046	TRANST 002160 G	T\$TAGN= 010077	T29PK2 026430	T30NEF 040576
SKIPT 003372	TSBA = 000000 G	T\$TEMP= 000000	T29PK3 026440	T30OFL 040307
SOFINI 016054 G	TSBAH = 000001 G	T\$TEST= 000011	T29RB 026442	T30PAC 036500
SPACE 010544 G	TSDB = 000000 G	T\$TSTM= 177777	T29RDF 026720	T30PK2 036610
SPM1 105716	TSDBH = 000001 G	T\$TSTS= 000001	T29RDG 031643	T30PK3 036620
SPM4 105746	TSFCOD 007314	T\$SAU = 010031	T29RES 032146	T30PTB 037242
SPM6 105776	TSREJ = 000006	T\$SAUT= 010033	T29RIB 031724	T30RB 036622
SPM7 106026	TSSDEF 006664	T\$SCLE= 010034	T29RN 026456	T30RDF 037413
SR0 = 177572	TSSR = 000002 G	T\$SDU = 010032	T29RNC 030354	T30RDG 037471
SR1 = 177574	TSSRBI 003476 G	T\$SHAR= 010075	T29RRF 026767	T30RES 041252
SR2 = 177576	TSSRFO 006473	T\$SHW = 010000	T29RRG 027103	T30RIB 036745
SR3 = 172516	TSSRH = 000003 G	T\$SINI= 010030	T29RRN 032024	T30RN 036636
SSR = 000200	TSSX 004014	T\$MSG= 010025	T29RSZ 026476	T30RRM 040655
STATCO 012472	TSTBLK 002744 G	T\$SPRO= 010027	T29RT2 032240	T30RRN 040733
SVCGBL= 000000	TSTCNT 002206 G	T\$SRPT= 010035	T29RT3 032302	T30RRP 041012
SVCINS= 000000	TSTEND 017000	T\$SOF= 010076	T29RWN 030305	T30RT2 041344
SVCSUB= 000001	TSTFLA 002306 G	T\$SRV= 010026	T29SC 027217	T30RT3 041406
SVCTAG= 000000	TSTLOO 016536 G	T\$SJB= 010074	T29SSR 027507	T30RWN 040240
SVCTST= 000001	TSTPTR 002310 G	T\$SW = 010001	T29SZ 026446	T3USKM 037114
S\$LSYM= 010000	TSTSET 016570 G	T\$TES= 010073	T29S2 026452	T3OSSR 037711
SO.IDB= 000010	TST29I 032117	T1 023526 G	T29S3 026454	T3OSZ 036626
SO.IFB= 000002	TST30I 041231	T1.1 023556	T29TM 030227	T3OS2 036632
SO.IFP= 000001	TST31I 046543	T1.2 024206	T29TRL 031457	T3OS3 036634
SO.ILD= 000020	TST32I 052640	T1.3 024730	T29VCK 030771	T30TM 040106
SO.ION= 000040	TST33I 055645	T1.4 025454	T29WB 026442	T30TMK 040514
SO.IRD= 000100	TST34I 063207	T2 032334 G	T29WDC 030677	T30TM2 040163
SO.IRW= 000004	TST35I 073133	T2.1 032360	T29WDD 030570	T30TPB 037333
SO.ISP= 000200	TST36I 100731	T2.2 033752	T29WDE 027562	T30VCK 040441
S1.ICE= 002000	TST37I 105353	T2.3 035376	T29WDF 027351	T30WB 036622
S1.IEO= 010000	TSV2 002000 G	T2.4 035772	T29WDR 026460	T30WDC 040362
S1.IFM= 001000	TSV3 002170 G	T23A 003136 G	T29WLK 027644	T30WDD 037170
S1.IHE= 000400	TSV4 021626 G	T23B 003140 G	T29WNG 026523	T30WDE 037762
S1.IID= 004000	TSV6 105554 G	T29AM3 030427	T29WRT 027731	T30WDF 037553
S1.IIR= 020000	TSV7B 023526 G	T29BA 031044	T29WSS 031136	T31AM3 045016
S1.I2R= 040000	TTIBFR= 177562 G	T29BF 026342	T3 041432 G	T31BA 045356
S1.PAR= 100000	TTICSR= 177560 G	T29BF2 026450	T3.1 041462	T31BFR 043212
S2.ATI= 000010	TTIVEC= 000060 G	T29BOT 027776	T3.2 042340	T31BF2 043320
S2.BTI= 000004	T\$ARGC= 000003	T29BS0 026450	T30BFR 036522	T31BOT 044345
S2.DIM= 000200	T\$CODE= 001130	T29BS1 026451	T30BF2 036630	T31BS0 043320
S2.ILW= 000100	T\$ERRN= 001620	T29CNT 026474		T31BS1 043321

T31CNT 043336
T31CNU 043340
T31CON 043332
T31DAT 043200
T31DLY 043342
T31DSW 043210
T31DTA 046446
T31EOT 044540
T31LON 045520
T31LOO 041462
T31LOP 045602
T31LOQ 044116
T31LOR 043771
T31NEF 046040
T31OFL 045065
T31PAC 043170
T31PBP 045664
T31PK2 043300
T31PK3 043310
T31RB 043312
T31RDE 043344
T31RDF 043543
T31RES 046610
T31RN 043326
T31RNC 044743
T31RRF 043612
T31RT2 046702
T31RT3 046744
T31RWN 044674
T31SC 043707
T31SCF 046161
T31SSR 044177
T31SZ 043316
T31S2 043322
T31S3 043324
T31TIM 044440
T31TM 044617
T31TRL 045752
T31TSA 046236
T31VCK 045303
T31WB 043312
T31WDC 045230
T31WDD 045140
T31WDE 044233
T31WDF 044041
T31WDR 043330
T31WNG 043471
T31WNH 043410
T31WRF 046343
T31WSS 045431
T32AM3 051747
T32BA 052063
T32BFR 051352
T32BOE 052366
T32BOT 051516
T32CMD 051460
T32CNT 051510

T32CNU 051512
T32DAT 051340
T32DLY 051514
T32DSW 051350
T32ECF 052455
T32EOT 051611
T32ERA 052016
T32LOO 047020
T32OP1 052603
T32PAC 051330
T32PK2 051440
T32PK3 051450
T32RB 051452
T32RES 052700
T32RIB 052136
T32RT2 052772
T32RT3 053022
T32RWN 051700
T32SCF 052234
T32SZ 051456
T32TSA 052311
T32WB 051452
T32WDC 052536
T33BFR 054572
T33BF2 054700
T33BOT 055325
T33BSO 054700
T33BS1 054701
T33CNT 054716
T33CNU 054720
T33CON 054712
T33DAT 054560
T33DLY 054722
T33DSW 054570
T33DTA 055550
T33LOO 053076
T33PAC 054550
T33PK2 054660
T33PK3 054670
T33RB 054672
T33RBP 054724
T33RES 055662
T33RN 054706
T33RT2 055754
T33RT3 056016
T33RWN 055420
T33SSR 055241
T33SZ 054676
T33S2 054702
T33S3 054704
T33UNC 055062
T33UND 055152
T33WB 054672
T33WDC 055467
T33WDR 054710
T33WPW 055002
T34AM3 062461

T34BA 063046
T34BFR 060562
T34BF2 060676
T34BOT 061234
T34BSO 060676
T34BS1 060677
T34CNT 060672
T34CON 060710
T34DAT 060550
T34DLY 060674
T34DSW 060560
T34EOT 062205
T34ET 062116
T34ETC 061157
T34ETN 061451
T34ETO 061002
T34ETS 061530
T34ETZ 061622
T34ET2 061367
T34LOO 056072
T34OFL 062527
T34PAC 060540
T34PK2 060650
T34PK3 060660
T34POS 060714
T34RB 060662
T34RES 063232
T34RNC 062406
T34RRE 061066
T34RSZ 060670
T34RT2 063324
T34RT3 063366
T34RWN 062337
T34SSR 062063
T34STM 061700
T34SZ 060666
T34S2 060700
T34S3 060702
T34TM 062263
T34TMK 061763
T34VCK 062773
T34WB 060662
T34WD 060704
T34WDC 062671
T34WDD 062602
T34WDR 060706
T34WSS 063120
T34WTM 061300
T35AM3 070766
T35BA 071326
T35BFR 067412
T35BF2 067520
T35BOT 070340
T35BSO 067520
T35BS1 067521
T35CNT 067536
T35CNU 067540

T35CON 067532
T35DAT 067400
T35DLY 067542
T35DSW 067410
T35DTA 072325
T35EOT 070510
T35INT 072601
T35LON 071470
T35LOO 063444
T35LOP 071552
T35LOQ 070205
T35LOR 070060
T35MOT 072503
T35NEF 072010
T35NIN 073056
T35OFL 071035
T35OPM 072672
T35PAC 067370
T35PBP 071634
T35PK2 067500
T35PK3 067510
T35RB 067512
T35RDF 067632
T35RES 073164
T35RN 067526
T35RNC 070713
T35RRF 067701
T35RT2 073256
T35RT3 073320
T35RWE 072770
T35RWN 070644
T35SC 067776
T35SCF 072106
T35SSR 072422
T35SZ 067516
T35S2 067522
T35S3 067524
T35TIM 070433
T35TM 070567
T35TRL 071722
T35TSA 072163
T35VCK 071253
T35WB 067512
T35WDC 071200
T35WDD 071110
T35WDE 070266
T35WDF 070130
T35WDR 067530
T35WNG 067544
T35WRF 072245
T35WSS 071401
T36AM3 077275
T36BA 077635
T36BFR 075602
T36BF2 075710
T36BOT 076647
T36BSO 075710

T36BS1 075711
T36CNT 075726
T36CNU 075730
T36CON 075722
T36DAT 075570
T36DLY 075732
T36DSW 075600
T36DTA 100634
T36EOT 077017
T36LON 077777
T36LOO 073400
T36LOP 100061
T36LOQ 076460
T36LOR 076333
T36NAS 075734
T36NEF 100317
T36OFL 077344
T36PAC 075560
T36PBP 100143
T36PK2 075670
T36PK3 075700
T36RB 075702
T36RDF 076105
T36RES 100752
T36RN 075716
T36RNC 077222
T36RRF 076154
T36RT2 101044
T36RT3 101106
T36RWN 077153
T36SC 076251
T36SCF 100415
T36SSR 076541
T36SZ 075706
T36S2 075712
T36S3 075714
T36TIM 076742
T36TM 077076
T36TRL 100231
T36TSA 100472
T36VCK 077562
T36WB 075702
T36WDC 077507
T36WDD 077417
T36WDE 076575
T36WDF 076403
T36WDR 075720
T36WNG 076017
T36WRF 100554
T36WSS 077710
T37AM3 103717
T37BA 104257
T37BFR 102272
T37BF2 102400
T37BOT 103271
T37BSO 102400
T37BS1 102401

T37CNT 102416	T37SSR 103146	T7.4 066444	WSMBK 021350 G	X\$OFFS= 000400
T37CNU 102420	T37SZ 102376	T8 073344 G	XFERAS 016020	X\$TRUE= 000020
T37CON 102412	T37S2 102402	T8.1 073400	XNXM 016456	X1.CGR= 020000
T37DAT 102260	T37S3 102404	T8.2 074454	XORBFO 007752	X1.DLT= 100000
T37DLY 102422	T37TIM 103364	T9 101132 G	XORFOR 010070	X1.MBZ= 017375
T37DSW 102270	T37TM 103520	T9.1 101166	XST0 = 000006 G	X1.RBP= 000400
T37DTA 105256	T37TRL 104653	UAM = 000200 G	XST1 = 000010 G	X1.SPA= 040000
T37EOT 103441	T37TSA 105114	UNITN = 002174 G	XST2 = 000012 G	X1.UNC= 000002
T37LON 104421	T37VCK 104204	UNREC = 000006	XST3 = 000014 G	X2.BUF= 000100
T37LOO 101166	T37WB 102372	USI 004117	XST4 = 000016 G	X2.EXT= 000200
T37LOP 104503	T37WDC 104131	WAITF 016330 G	XSOBOT= 000002	X2.OPM= 100000
T37LOQ 103065	T37WDD 104041	WC.IFA= 000200	XSOEOT= 000001	X2.RCE= 040000
T37LOR 102740	T37WDE 103202	WC.IFE= 000002	XSOIE = 000040	X2.REV= 000077
T37NEF 104741	T37WDF 103010	WC.IGO= 000001	XSOILA= 000400	X2.SPA= 035400
T37OFL 103766	T37WDR 102410	WC.IRE= 000010	XSOILC= 001000	X2.UNI= 000007
T37PAC 102250	T37WNG 102424	WC.IRW= 000004	XSOLET= 020000	X2.WCF= 002000
T37PBP 104565	T37WRF 105176	WC.IOT= 000100	XSOMOT= 000200	X3.DCK= 000010
T37PK2 102360	T37WSS 104332	WC.IIT= 000040	XSONEF= 002000	X3.MBZ= 000006
T37PK3 102370	T4 046770 G	WC.ISR= 000020	XSOONL= 000100	X3.MDE= 177400
T37RB 102372	T4.1 047020	WF.IED= 000010	XSOPED= 000010	X3.OPI= 000100
T37RDF 102512	T4.2 047660	WF.IER= 000004	XSORLL= 010000	X3.REV= 000040
T37RES 105374	T4.3 050470	WF.IHI= 000200	XSORLS= 040000	X3.RIB= 000001
T37RN 102406	T5 053046 G	WF.IRE= 000040	XSOTMK= 100000	X3.SPA= 000200
T37RNC 103644	T5.1 053076	WF.IWF= 000020	XSOVCK= 000020	X3.TRF= 000020
T37RRF 102561	T6 056042 G	WF.IWR= 000100	XSOWLE= 004000	X4.HSP= 100000
T37RT2 105466	T6.1 056072	WF.I3R= 000002	XSOWLK= 000004	X4.MBZ= 017400
T37RT3 105530	T7 063414 G	WF.I4R= 000001	XXCOMM 003114 G	X4.RCE= 040000
T37RWN 103575	T7.1 063444	WRTCHR 010742 G	X\$ALWA= 000000	X4.TSM= 020000
T37SC 102656	T7.2 064522	WRTERR 005107	X\$FALS= 000040	X4.WRC= 000377
T37SCF 105037	T7.3 065602	WRTMSG 005052		

. ABS. 106404 000
 000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 30328 WORDS (119 PAGES)

DYNAMIC MEMORY: 20346 WORDS (78 PAGES)

ELAPSED TIME: 00:39:26

CVTSDAO, CVTSDAO/-SP=SVC/ML, TSV1D, TSV22D, TSV3B, TSV4, TSV7B, TSV6