

TSV05

TSV05 CTRL LT4
CVTSDBO

COPYRIGHT (c) 1982-83
RH-T100B-MC
FICHE 01 OF 02

JUL 1984
digital
Made In USA

The main body of the document is a microfiche card containing a grid of 16 columns and 16 rows of data. Each cell in the grid contains a small, high-contrast image of a document page, likely a technical drawing or schematic. The images are arranged in a regular grid pattern across the entire page.

TSV05

TSV05 CTRL LT4
CVTSDBO

COPYRIGHT (c) 1982-83
AH-T100B-MC
FICHE 02 OF 02

JUL 1984
digital
Made In USA

TSV05
CVTSDBO
CTRL LT4
FICHE 02 OF 02

.REM
IDENTIFICATION

PRODUCT CODE: AC T0998 MC
PRODUCT TITLE: CVTSDBO TSV05 CONTROLLER LOGIC TEST 4
AUTHOR: DICK MITCHELL
DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PPG
DATE: APRIL 26, 1983

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) 1982,1983 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL PDP INIBUS 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 (MSAAA.SYS VERSION 34 OR LATER)
PDP 11 DIAGNOSTIC LOADER/MONITOR (XXDP.)

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

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

1.5 ASSUMPTIONS

ALL HARDWARE EXCEPT THE HARDWARE UNDER TEST IS ASSUMED TO WORK PROPERLY OR FALSE ERRORS CAN BE REPORTED.
THE TAPE BEING USED ON THE TSV05 TRANSPORT IS A KNOWN GOOD REEL OF TAPE.
CVTSAA, 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, USER'S MANUAL, DOCUMENT NUMBER AC F348E MC. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC XXDP MEDIA

```
.R VTS0??
DIAG. RUN-TIME SERVICES REV D. APR 79
CVTSD B-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
MOE	HALT ON ERROR CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

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

2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING 'CHANGE HW (L) ?' YOU MUST ANSWER 'Y' AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 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 DEVIC 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 (0) ? 8<CR>

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

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

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

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

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

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

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

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

NOTICE THAT THE DEFAULT VALUE FOR THE Q FACTOR CHANGES WHEN A NON DEFAULT RESPONSI. 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 (O) ? 160000<CR>
SUB-DEVICE # (O) ? 0,1<CR>
Q-FACTOR (O) 0 ? 1,0<CR>

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

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

CVISD WRD ERR 00'21 ON UNIT 00 TST 001 SUB 002 PC: 023306
MOT BIT (XSTO) NOT SET DURING REWIND (EXTENDED FEATURES MODE)
EXPD: 000312 RECV: 000112 XOR: 000200

4.0 PERFORMANCE AND PROGRESS REPORTS

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

SUCCESSFUL RUN EXAMPLE (PDP 11/23)

DR>STA/FLA:PNT:MOE

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	
4	1	1	-
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 START) COMMAND, THE SUPERVISOR PROMPTS THE FOLLOWING P-TABLES PARAMETER CHANGES:

CHANGE HW (L) ?

UNITS (0) ? <ENTER THE NUMBER OF M7196 CONTROLLERS PRESENT TO BE TESTED>

UNIT 0

DEVICE ADDRESS (0) 172520 ? <ENTER THE ADDRESS OF THE TSBA/TSDB REGISTER>

VECTOR (0) 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 (E'S) 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 PARTY 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

REVISION B - APRIL 1983

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

```

2          .TITLE  TSV2  PROGRAM HEADER
3          .SBTTL  PROGRAM HEADER
4
10         .MCALL  SVC
11 000000  SVC          ; INITIALIZE SUPERVISOR MACROS
12         .ENABLE LC
13         .MLIST  BEX,CND
19 000000  .ENABL  ABS,AMA
20 002000  .=2000
21 002000  BGNMOD  TSV2
22
23         ;
24         ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
25         ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
26         ;
27
28 002000  POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
29 002000  HEADER  CVTSD,B,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      102          .ASCII  /B/
002011  L$DEPO::        ;0
002011      060          .ASCII  /O/
002012  L$UNIT::        ;NUMBER OF UNITS
002012  000000          .WORD   0
002014  L$TIML::        ;LONGEST TEST TIME
002014  001217          .WORD   655.
002016  L$MPCP::        ;POINTER TO H.W. QUES.
002016  105546          .WORD   L$HARD
002020  L$SPCP::        ;POINTER TO S.W. QUES.
002020  105700          .WORD   L$SOFT
002022  L$MPTP::        ;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$DISPATCH
002042  L$PRIO::       ;DIAGNOSTIC RUN PRIORITY

```

TSV2 PROGRAM HEADER MACRO M1113 06 FEB 84 18:04
PROGRAM HEADER

SEQ 017

002042	000000		.WORD	0	
002044		L\$ENVI::			;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000		.WORD	0	
002046		L\$EXP1::			;EXPANSION WORD
002046	000000		.WORD	0	
002050		L\$MREV::			;SVC REV AND EDIT #
002050	003		.BYTE		C\$REVISION
002051	003		.BYTE		C\$EDIT
002052		L\$EF::			;DIAG. EVENT FLAGS
002052	000000		.WORD	0	
002054	000000		.WORD	0	
002056		L\$SPC::			
002056	000000		.WORD	0	
002060		L\$DEVP::			; POINTER TO DEVICE TYPE LIST
002060	003374		.WORD	L\$DVTYP	
002062		L\$REPP::			;PTR. TO REPORT CODE
002062	022744		.WORD	L\$RPT	
002064		L\$EXP4::			
002064	000000		.WORD	0	
002066		L\$EXP5::			
002066	000000		.WORD	0	
002070		L\$AUT::			;PTR. TO ADD UNIT CODE
002070	022432		.WORD	L\$AU	
002072		L\$DUT::			;PTR. TO DROP UNIT CODE
002072	022530		.WORD	L\$DU	
002074		L\$IUN::			;LUN FOR EXERCISERS TO FILL
002074	000000		.WORD	0	
002076		L\$DESP::			;POINTER TO DIAG. DESCRIPTION
002076	003402		.WORD	L\$DESC	
002100		L\$LOAD::			;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		EMT	E\$LOAD	
002102		L\$ETP::			;POINTER TO ERR TBL
002102	000000		.WORD	0	
002104		L\$ICP::			;PTR. TO INIT CODE
002104	021636		.WORD	L\$INIT	
002106		L\$CCP::			;PTR. TO CLEAN UP CODE
002106	022716		.WORD	L\$CLEAN	
002110		L\$ACP::			;PTR. TO AUTO CODE
002110	022636		.WORD	L\$AUTO	
002112		L\$PRT::			;PTR. TO PROTECT TABLE
002112	021626		.WORD	L\$PROT	
002114		L\$TEST::			;TEST NUMBER
002114	000000		.WORD	0	
002116		L\$DLY::			;DELAY COUNT
002116	000000		.WORD	0	
002120		L\$HIME::			;PTR. TO HIGH MEM
002120	000000		.WORD	0	

TSV2 PROGRAM HEADER
DISPATCH TABLE

MACRO M1113 06 FEB 84 18:04

SEQ 018

31
32
33
34
35
36
37
38
39

.SBTTL DISPATCH TABLE

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

002122
002122 000011
002124
002124 023526
002126 032334
002130 041432
002132 046770
002134 053046
002136 056042
002140 063414
002142 073344
002144 101120

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

```

41          .SBTTL  DEFAULT HARDWARE P TABLE
42
43          ;**
44          ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
45          ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
46          ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P TABLE.
47          ;
48 002146          BGNHW      DFPTBL      ;DEFAULT HARD P TABLE
          002146      000003      .WORD      L10000 L$HW/2
          002150
          002150      L$HW::
          DFPTBL::
49
50 002150      172520      .WORD      172520      ; 1ST (OF 2) REGISTERS.
51 002152      000224      .WORD      224        ; INTERRUPT VECTOR
52 002154      000200      .WORD      PRI04      ; INTERRUPT PRIORITY.
53 002156          ENDHW
          002156      L10000:

```

```

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

```

TSV3 - GLOBAL AREAS
SOFTWARE P TABLE

MACRO M1113 06 FEB 84 18:04

SEQ 021

```

7          .TITLE  TSV3 - GLOBAL AREAS
8          .SBTTL  GLOBAL EQUATES SECTION
13
19
20 002170  BGNMOD  TSV3
    002170  TSV3::
21
22          .SBTTL  GLOBAL EQUATES SECTION
23
24          ;**
25          ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
26          ; ARE USED IN MORE THAN ONE TEST.
27          ;--
32 002170  EQUALS          ; GET STANDARD EQUATES.
          ;
          ; BIT DIFINITIONS
          ;
          100000  BIT15** 100000
          040000  BIT14** 40000
          020000  BIT13** 20000
          010000  BIT12** 10000
          004000  BIT11** 4000
          002000  BIT10** 2000
          001000  BIT09** 1000
          000400  BIT08** 400
          000200  BIT07** 200
          000100  BIT06** 100
          000040  BIT05** 40
          000020  BIT04** 20
          000010  BIT03** 10
          000004  BIT02** 4
          000002  BIT01** 2
          000001  BIT00** 1
          ;
          001000  BIT9**  BIT09
          000400  BIT8**  BIT08
          000200  BIT7**  BIT07
          000100  BIT6**  BIT06
          000040  BIT5**  BIT05
          000020  BIT4**  BIT04
          000010  BIT3**  BIT03
          000004  BIT2**  BIT02
          000002  BIT1**  BIT01
          000001  BIT0**  BIT00
          ;
          ; EVENT FLAG DEFINITIONS
          ; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
          ;
          000040  EF.START** 32.          ; START COMMAND WAS ISSUED
          000037  EF.RESTART** 31.        ; RESTART COMMAND WAS ISSUED
          000036  EF.CONTINUE** 30.       ; CONTINUE COMMAND WAS ISSUED
          000035  EF.NEW** 29.           ; A NEW PASS HAS BEEN STARTED
          000034  EF.PWR** 28.           ; A POWER FAIL/POWER UP OCCURRED
          ;
          ; PRIORITY LEVEL DEFINITIONS

```

000340
000300
000240
000200
000140
000100
000040
000000

;
PRI07== 340
PRI06== 300
PRI05== 240
PRI04== 200
PRI03== 140
PRI02== 100
PRI01== 40
PRI00== 0

;
;OPERATOR FLAG BITS

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

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

33
34 002170

000250
177572
177574
177576
172516

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

;DEFINE MEMORY MANAGEMENT REGISTERS


```
;*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
```

```

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

```

```

96      040000      X1.SPARE= BIT14      ;NOT USED
97      020000      X1.COR  = BIT13      ;CORRECTABLE DATA ERROR
98      017375      X1.MBZ  = BIT  -BIT11-BIT10-BIT9-BIT7-BIT6-BIT5-BIT4-BIT3-BIT2-BIT0 ;ALWAYS 0
99      000000      X1.RBP  = BIT      ;READ BUS PARITY ERROR
100     000002      X1.UNC  = BIT1      ;UNCORRECTABLE DATA OR HARD ERROR
101
102     ;
103     ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
104     ;(XST2)
105     ;
106     100000      X2.OPM  = BIT15      ;OPERATION IN PROGRESS (TAPE MOVING)
107     040000      X2.RCE  = BIT14      ;RAM CHECKSUM ERROR
108     035400      X2.SPARE= BIT13-BIT12-BIT11-BIT9-BIT8      ;NOT USED BY TSV05 (ALWAYS=0)
109     002000      X2.WCF  = BIT10      ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
110     000200      X2.EXTF = BIT7       ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
111     000100      X2.BUFE = BIT6       ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
112     000077      X2.PRV  = 000077    ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
113     000007      X2.UNIT = BIT2-BIT1-BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
114
115     ;
116     ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
117     ;(XST3)
118     ;
119     177400      X3.MDE  = 177400    ;MICRO-DIAGNOSTIC ERROR CODE
120     000200      X3.SPARE= BIT7       ;NOT USED BY TSV05
121     000100      X3.OPI  = BIT6       ;OPERATION INCOMPLETE
122     000040      X3.REV  = BIT5       ;REVERSE
123     000020      X3.TRF  = BIT4       ;TRANSPORT RESPONSE FAILURE
124     000010      X3.DCK  = BIT3       ;DENSITY CHECK
125     000006      X3.MBZ  =BIT2-BIT1    ;NOT USED ALWAYS 0
126     000001      X3.RIB  = BIT0       ;REVERSE INTO BOT
127
128     ;
129     ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
130     ;(XST4)
131     ;
132     100000      X4.HSP  = BIT15      ;HIGH SPEED
133     040000      X4.RCE  = BIT14      ;RETRY COUNT EXCEEDED
134     020000      X4.TSM  = BIT13      ;TRANSPORT SPECIAL MODE
135     017400      X4.MBZ  = BIT12-BIT11-BIT10-BIT9-BIT8      ;NOT USED ALWAYS 0
136     000377      X4.WRC  = 000377    ;WRITE RETRY COUNT FIELD
137
138     ;
139     ;
140     ;TSR TERMINATION CODES (BIT 0 2)
141     ;
142     ;
143     ;
144     000006      TSREJ= 3*2          ;COMMAND REJECTED
145     000006      UNREC= 6           ;UNRECOVERABLE ERROR
146
147     ;
148     ;
149     ;DEVICE REGISTER OFFSETS
150     ;
151     ;
152

```

```

153      000000      TSBA..  0
154      000000      TSDB..  0      ;TSDB/TSBA REGISTER
155      000001      TSBAH.. 1
156      000001      TSDBH.. 1      ;TSDB/TSBA REGISTER HIGH BYTE
157      000002      TSSR..  2      ;TSSR REGISTER
158      000003      TSSRH.. 3      ;TSSR REGISTER HIGH BYTE
159
160      ;
161      ; TSDB ADDRESS BIT DEFINITIONS
162      ;
163      000003      A1716  = BIT1:BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
164
165      ;
166      ; COMMAND DEFINITIONS
167      ;
168      000017      P.GETSTAT      = 17      ;GET STATUS
169      000013      P.INIT          = 13      ;INITIALIZE
170      000012      P.CONTROL      = 12      ;CONTROL COMMANDS
171      000011      P.FORMAT       = 11      ;FORMAT
172      000010      P.POSITION     = 10      ;POSITION
173      000006      P.WRTSUB       = 6       ;SUBSYSTEM WRITE
174      000005      P.WRITE        = 5       ;WRITE
175      000004      P.WRTCHAR      = 4       ;WRITE CHARACTERISTICS
176      000001      P.READ         = 1       ;READ
177
178      ;
179      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
180      ;
181      100000      P.ACK          = BIT15      ;BUFFER AVAIL FOR CONTROLLER
182      040000      P.CVC          = BIT14      ;CLEAR VOLUME CHECK
183      020000      P.OPP          = BIT13      ;REVERSE SEQUENCE OF DATA BITS
184      010000      P.SWB         = BIT12      ;SWAP BYTES IN MEMORY
185      007400      P.MODE         = BIT11:BIT10:BIT9:BIT8 ;EXTENDED COMMAND MODE FIELD
186      000200      P.IE          = BIT7       ;INTERRUPT ENABLE
187      000140      P.FMT         = BIT6:BITS5 ;PACKET HEADER TYPE (ALWAYS=0)
188      000037      P.CMD         = 37        ;MAJOR COMMAND FIELD
189
190      ;
191      ; CONTROL COMMAND MODE CODES
192      ;
193      000000      PC.RELEASE     = 0*256.    ;RELEASE BUFFER
194      000400      PC.REWIND     = 1*256.    ;REWIND
195      001000      PC.NOOP       = 2*256.    ;NO OP
196      002000      PC.IEREW     = 4*256.    ;REWIND IMMEDIATE INTERRUPT
197      002400      PC.ERASE      = 5*256.    ;SECURITY ERASE
198
199      ;
200      ; CONTROLLER RAM DEFINITIONS
201      ;
202      000167      RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
203      000200      RMCHEND = 200     ;CHARACTERISTICS IO DATA END RAM ADDRESS
204      000201      RMPKTBEGBEG = 201 ;COMMAND PACKET BEGIN RAM ADDRESS
205      000210      RMPKTBEGETEND = 210 ;COMMAND PACKET END RAM ADDRESS
206      000215      RMMSGBEG = 215    ;MESSAGE BUFFER BEGIN RAM ADDRESS
207      000234      RMMSGEND = 234    ;MESSAGE BUFFER END RAM ADDRESS
208      ;
209      ; REGISTER DEFINITIONS IN THE MESSAGE BUFFER

```

```

210      ;
211      ;
212      ;
213      000006      XST0** 6          ;EXTENDED STATUS REGISTER 0 (WORD 4)
214      000010      XST1** 8.         ;EXTENDED STATUS REGISTER 1 (WORD 5)
215      000012      XST2** 10.        ;EXTENDED STATUS REGISTER 2 (WORD 6)
216      000014      XST3** 12.        ;EXTENDED STATUS REGISTER 3 (WORD 7)
217      000016      XST4** 14.        ;EXTENDED STATUS REGISTER 4 (WORD 8)
218      ;
219      ;*
220      ;
221      ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
222      ;
223      ;-
224      ;
225      000002      PKLOW   = 2          ;LOW ORDER CHARACTERISTIC DATA POINTER
226      000004      PKHI    = 4          ;HIGH ORDER CHARACTERISTIC DATA POINTER
227      000006      PKBCNT  = 6          ;NUMBER OF BYTES IN DATA PACKET
228      ;
229      000010      EXBCNT =10          ;NUMBER OF BYTES IN EXTENDED DATA PACKET
230      ;
231      ;*
232      ;DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
233      ;
234      000000      BSELO   = 0          ;BYTE 0
235      000001      BSEL1  = 1          ;BYTE 1
236      000002      SEL2   = 2          ;WORD 2
237      000004      SELDATA = 4          ;WORD 3
238      ;
239      ;*
240      ;BSELO SELECT CODES FOR WRITE SUBSYSTEM COMMAND
241      ;-
242      000000      PW.NOP   = 0          ;NO-OP
243      000001      PW.RDRAM = 1          ;READ RAM
244      000002      PW.WTRAM = 2          ;WRITE RAM
245      000003      PW.RFIFO = 3          ;READ FIFO
246      000004      PW.WFIFO = 4          ;WRITE FIFO
247      000005      PW.RDSTAT = 5         ;READ STATUS
248      000006      PW.WCTL  = 6          ;WRITE TAPE CONTROL
249      000007      PW.WFMT  = 7          ;WRITE TAPE FORMAT
250      000010      PW.WMISC = 10         ;WRITE MISCELLANEOUS
251      000011      PW.WNPR  = 11         ;WRITE NPR CONTROL
252      000020      PW.D22   = 20         ;DO MICROTEST 22
253      000021      PW.D11   = 21         ;DO MICROTEST 11
254      000022      PW.D13   = 22         ;DO MICROTEST 13
255      000023      PW.NO1311 = 23        ;DISABLE MICROTEST 11 AND 13
256      000024      PW.RDXT  = 24         ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
257      ;
258      ;*
259      ;BSEL1 CODES FOR WRITE TAPE CONTROL
260      ;-
261      000200      WC.IFAD   = BIT7      ;IFAD FORMATTER ADDRESS
262      000100      WC.IOTAD  = BIT6      ;ITADO - TRANSPORT ADDRESS BIT 0
263      000040      WC.I1TAD  = BIT5      ;ITAD1 - TRANSPORT ADDRESS BIT 1
264      000020      WC.ISRESV = BIT4      ;IRESV5 - RESERVED #5
265      000010      WC.IREW   = BIT3      ;IREW REWIND
266      000004      WC.IRWU   = BIT2      ;IRWU REWIND AND UNLOAD

```

```

267      000002      WC.IFEN      - BIT1      ;IFEN      FORMATTER ENABLE
268      000001      WC.IGO       - BIT0      ;IGO
269
270      ;
271      ;BSEL1 CODES FOR WRITE FORMAT
272      ;
273      000200      WF.IHISP     - BIT7      ;IHISP     - HIGH SPEED
274      000100      WF.IWRT     - BIT6      ;IWRT     - WRITE
275      000040      WF.IREV     - BIT5      ;IREV     - REVERSE
276      000020      WF.IWFM     - BIT4      ;IWFM     - WRITE FILE MARK
277      000010      WF.IEDIT     - BIT3      ;IEDIT     - EDIT
278      000004      WF.IERASE    - BIT2      ;IERASE    - ERASE
279      000002      WF.I3RESV    - BIT1      ;I3RESV3   - RESERVED #3
280      000001      WF.I4RESV    - BIT0      ;I4RESV4   - RESERVED #4
281
282      ;
283      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
284      ;
285      000200      MS.EXT      - BIT7      ;INVERT SENSE OF EXTENDED FEATURES SWITCH
286      000020      MS.RSFIFO    - BIT4      ;RESET FIFO AND INPUT PARITY ERROR
287      000010      MS.RSTAPE    - BIT3      ;RESET TAPE STATUS IN 2 FLIP-FLOPS
288      000006      MS.ATTN     - BIT2:BIT1 ;ATTENTION TRIGGER FIELD
289      000001      MS.RSD      - BIT0      ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
290
291      ;
292      ; MS.ATTN SUBCODES
293      ;
294      000000      MSA.NOP     - 0*2      ;NO-OP (NOTHING TRIGGERED)
295      000002      MSA.VOL     - 1*2      ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
296      000004      MSA.NRAM    - 2*2      ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
297      000006      MSA.FRAME   - 3*2      ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
298
299      ;
300      ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
301      ;
302      000200      NP.IR       - BIT7      ;INTERRUPT REQUEST (0 1 TRANSITION)
303      000100      NP.OUT      - BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
304      000040      NP.LOOP     - BIT5      ;ENABLE TRANSPORT LOOPBACK
305      000020      NP.WRP      - BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
306
307      ;
308      ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
309      ;
310      000200      S2.DIM      - BIT7      ;WORD #9 BYTE 2 DATA IN MISS
311      000100      S2.ILW      - BIT6      ; ILW M
312      000040      S2.OUTRDY    - BIT5      ; OUT RDY M
313      000020      S2.INRDY    - BIT4      ; IN RDY M
314      000010      S2.ATIMR    - BIT3      ; TIMER A FLAG M
315      000004      S2.BTIMR    - BIT2      ; TIMER B FLAG M
316      000003      S2.UNDEF    - BIT1:BIT0 ;(UNDEFINED)
317      100000      S1.PARIN     - BIT15     ;WORD #8 BYTE 1 PARIN M
318      040000      S1.I2RESV    - BIT14     ; IRESV2
319      020000      S1.I1RESV    - BIT13     ; IRESV1
320      010000      S1.IEOT      - BIT12     ; IEOT L
321      004000      S1.IIDENT    - BIT11     ; IIDENT M
322      002000      S1.ICER      - BIT10     ; ICER M
323      001000      S1.IFMK      - BIT9      ; IFMK M
324      000400      S1.IHER      - BIT8      ; IHER M
325      000200      S0.ISPEED    - BIT7      ;WORD #8 BYTE 0 ISPEED M
    
```

```

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

```


TSV3 GLOBAL AREAS
GLOBAL DATA SECTION

MACRO M1113 06 FEB 84 18:04

SEQ 032

```

438 002276 000000 RCVHIADD:: .WORD 0 ;RECEIVED BUFFER HIGH ADDRESS
439 002300 000000 RCVLOADD:: .WORD 0 ;RECEIVED BUFFER LOW ADDRESS
440 002302 000000 COUNT:: .WORD 0 ;TEST COUNT PATTERN
441 002304 000000 DATA:: .WORD 0 ;TEST DATA
442 002306 000000 TSTFLAG:: .WORD 0 ;TEST FLAG WORD
443 002310 000000 TSTPTR:: .WORD 0 ;TSTBLK POINTER
444 002312 000000 PRMNO:: .WORD 0 ;PRINT ROUTINE TEMP
445 002314 EXPMSG:: .BLKB 100. ;EXPECTED MESSAGE BUFFER DATA
446 002460 RECMSG:: .BLKB 100. ;RECEIVED MESSAGE BUFFER DATA
447 002624 TMPBFR:: .BLKB 80. ;TEMPORARY STORAGE FOR PRINT
448 .SBTTL TSTBLK - TEST DATA TABLE
449
450 ;*
451 ;
452 ; THIS TABLE CONTAINS TEST DATA USED IN SEVERAL TESTS
453 ;
454 ; IN SEQUENCE THE DATA IS:
455 ;
456 ; ALL ZEROS
457 ; ALL ONES
458 ; WALKING ONES
459 ; WALKING ZEROS
460 ; ALTERNATING ONES AND ZEROS
461 ;
462 ; -
463 ;
464 002744 TSTBLK::
465 002744 000000 .WORD 0 ;ALL ZEROS
466 002746 177777 .WORD 177777 ;ALL ONES
467 002750 000001 .WORD BIT0 ;DATA FOR WALKING ONES
468 002752 000002 .WORD BIT1
469 002754 000004 .WORD BIT2
470 002756 000010 .WORD BIT3
471 002760 000020 .WORD BIT4
472 002762 000040 .WORD BIT5
473 002764 000100 .WORD BIT6
474 002766 000200 .WORD BIT7
475 002770 000400 .WORD BIT8
476 002772 001000 .WORD BIT9
477 002774 002000 .WORD BIT10
478 002776 004000 .WORD BIT11
479 003000 010000 .WORD BIT12
480 003002 020000 .WORD BIT13
481 003004 040000 .WORD BIT14
482 003006 100000 .WORD BIT15
483 003010 177776 .WORD †CBIT0 ;DATA FOR WALKING ZEROS
484 003012 177775 .WORD †CBIT1
485 003014 177773 .WORD †CBIT2
486 003016 177767 .WORD †CBIT3
487 003020 177757 .WORD †CBIT4
488 003022 177737 .WORD †CBIT5
489 003024 177677 .WORD †CBIT6
490 003026 177577 .WORD †CBIT7
491 003030 177377 .WORD †CBIT8
492 003032 176777 .WORD †CBIT9
493 003034 175777 .WORD †CBIT10
494 003036 173777 .WORD †CBIT11

```

```

495 003040 167777 .WORD ↑CBIT12
496 003042 157777 .WORD ↑CBIT13
497 003044 137777 .WORD ↑CBIT14
498 003046 077777 .WORD ↑CBIT15
499 003050 125252 .WORD 125252 ;ALTERNATING ONES, ZEROS
500 003052 052525 .WORD 052525 ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
501 003054 003054
502
503
504
505
506 003054 000000 100000 000000 DUMMY: 0,100000,0,0 ;DUMMY DEVICE REGISTERS...
507 003064 000000 000000 000000 0,0,0,0,0,0,0,0
508 ;...FOR MULTI-UNIT CHECKOUT.
509
510 003104 000000 DUFLG:: .WORD 0 ;"DROPPED UNIT" FLAG.
511 ;INHIBITS CODE IN "CLEAN UP".
512 003106 000000 NODEV:: .WORD 0 ;FLAG TO SAY NO DEVICE.
513
514 003110 000000 TEMP1:: .WORD 0 ;SOME TEMP LOCATIONS.
515 003112 000000 TEMP2:: .WORD 0
516 003114 000000 XXCOMM:: .WORD 0 ;XXDP, COMM BLOCK POINTER.
517 003116 000000 FREE:: .WORD 0 ;1ST FREE MEMORY ADDRESS...
518 003120 000000 FRESIZ:: .WORD 0 ;...AND SIZE (IN WORDS).
519 003122 000000 FREEHI: .WORD 0 ;LAST WORD IN FREE SPACE
520 003124 000000 KTFLG:: .WORD 0 ;KT11, MEM AVAIL FLAG
521 ;- .WORD 0 = <24K OR NO KT
522 ;- NZ = >24K AND KT.
523 003126 000000 KTENABLE:: .WORD 0 ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
524 003130 000000 NXMFLG:: .WORD 0 ;SET IF WE CAN TEST CLEARED OTHERWISE
525 003132 000000 NXMLO:: .WORD 0 ;NXM LO ADDRESS BITS
526 003134 000000 NXMHI:: .WORD 0 ;NXM HI ADDRESS BITS FOR DAL 5 16 21
527 003136 000000 T23A:: .WORD 0 ;11/23A FLAG
528 003140 000000 T23B:: .WORD 0 ;11/23B FLAG
529 003142 000000 T3BFLG:: .WORD 0 ;TEST 3B FLAG ↑0
530 003144 002000 PST32W:: .WORD 2000 ;32W BLOCK ADDRESS FOR 32K START
531 003146 000000 SIFLAG:: .WORD 0
532 003150 000000 BADDAT:: .WORD 0 ;ACTUAL DATA
533 003152 000000 GDDAT:: .WORD 0 ;EXPECTED DATA
534 003154 000000 LOOPFL:: .WORD 0
535 003156 CTAB:: ;CONFIGURATION TABLES.
536 003156 000000 CTABM:: .WORD 0 ;CONFIG WORK.
537 003160 .WORD 0
538 003162 .WORD 0
539 003164 .WORD 0
540 003166 177777 .WORD 1 ;END OF MEM TABLE.
541 003170
542
543
544
545
546
547
548
549
550
551 003170

```

CTABE::
;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
;
; 0 = UNIT NOT TESTED
; 100000 = UNIT ONLINE, NO ERRORS
; 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
; 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
; 160001 = UNIT DROPPED, NOT IDLE AT START
; 14XXXX = UNIT DROPPED, ENCOUNTERED XXXX ERRORS
;
ERTABL: .BI KW 64.

TSV3 GLOBAL AREAS MACRO M1113 06 FEB 84 18:04
GLOBAL ENVIRONMENT STORAGE

SEQ 034

552 003370 000000
553
554 003372 000000

ERTABE: .WORD 0
SKIPT: .WORD 0

;1-SKIP SUBTEST 0-NO SKIP OF SUBTEST

```

556 .SBTTL GLOBAL TEXT MESSAGES
557
558 ;*
559 ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
560 ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
561 ; MORE THAN ONE TEST.
562 ;
563 ;*
564 ; NAMES OF DEVICES SUPPORTED
565 ;-
566
567 003374          DEVTYP <TSV05>
   003374          L$DVTYP:
   003374          124      123      126      .ASCIZ /TSV05/
   .EVEN
568
589 ;*
590 ; TEST DESCRIPTION
591 ;-
592 003402          DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC  CHECK TRANSPORT IF ERROR ****>
   003402          L$DESC:
   003402          052      052      052      .ASCIZ /**** TSV05 LOGIC DIAGNOSTIC - CHECK TRANSPORT IF ERROR ****/
   .EVEN
594
595 ;*
596 ; BIT TO ASCII CONVERSION FOR TSSR REGISTER
597 ;-
598
599 003476 003536 003541 003545 TSSRBIT: .WORD 1$,2$,3$,4$,5$,6$,7$,8$
600 003516 003577 003603 003607 .WORD 9$,10$,11$,12$,13$,14$,15$,16$
601 003536      123      103      000 1$: .ASCIZ 'SC'
602 003541      102      111      105 2$: .ASCIZ 'BIE'
603 003545      123      103      105 3$: .ASCIZ 'SCE'
604 003551      122      115      122 4$: .ASCIZ 'RMR'
605 003555      116      130      115 5$: .ASCIZ 'NXM'
606 003561      116      102      101 6$: .ASCIZ 'NBA'
607 003565      102      111      124 7$: .ASCIZ 'BIT9'
608 003572      102      111      124 8$: .ASCIZ 'BIT8'
609 003577      123      123      122 9$: .ASCIZ 'SSR'
610 003603      117      106      114 10$: .ASCIZ 'OFL'
611 003607      102      111      124 11$: .ASCIZ 'BIT5'
612 003614      102      111      124 12$: .ASCIZ 'BIT4'
613 003621      102      111      124 13$: .ASCIZ 'BIT3'
614 003626      102      111      124 14$: .ASCIZ 'BIT2'
615 003633      102      111      124 15$: .ASCIZ 'BIT1'
616 003640      102      111      124 16$: .ASCIZ 'BIT0'
617 .EVEN
618 003646      124      123      123 SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
619 003701      124      123      123 SFHERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
620 003734      040      040      116 NXR: .ASCIZ /NON-EXISTANT DEVICE REGISTER/
621 003773      045      101      040 NXRX: .ASCIZ /#A ADDRESS: #06/
622 004014      045      101      040 TSSX: .ASCIZ /#A TSBA,TSSR EXP D: #06#A,#06#N/
623 004054      045      101      040 .ASCIZ /#A TSBA,TSSR REC D: #06#A,#06#N/
624 004113      045      116      045 FUSI: .ASCIZ /#N#A/
625 004117      040      040      125 USI: .ASCIZ /UNEXPECTED INTERRUPT/
626 004146      040      040      111 NSI: .ASCIZ /INTERRUPT EXPECTED, NOT RECEIVED/
627 004211      045      116      045 FNOINTR: .ASCII /#N#A/

```

```

628 004215      040      040      116 NOINTR: .ASCIZ / NO INTERRUPT WAS GENERATED/
629 004252      040      040      111 IFAULT: .ASCIZ / INTERRUPT FAULT/
630 004274      045      101      040 INTX: .ASCIZ /#A CPU PC: #06#A TSBA: #06/
631 004331      C40      040      042 NOINIT: .ASCIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
632 004403      040      040      042 NSINIT: .ASCIZ / "SOFT INIT" DIDN'T INITIALIZE THE DPU/
633 004453      040      040      042 BRINIT: .ASCIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
634
635 004523      000
636 004524      045      116      000 NULCR: .ASCIZ /#N/
637 004527      045      101      040 EXPGOT: .ASCIZ /#A EXP'D: #06#A, REC'D: #06/
638 004563      045      116      045 EXPGT2: .ASCIZ /#N#A EXP'D: #06#A, #06#N#A REC'D: #0#A, #06/
639 004637      045      101      040 DUAD12: .ASCIZ /#A REG(W) WRITTEN TO: #06#A REG(R) READ; EXP'D: #06#A, REC D: #06/
640 004741      122      101      115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
641 005007      040      040      103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
642 005052      127      122      111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
643 005107      124      123      123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
644 005202      124      123      123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
645 005274      106      101      124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
646 005366      105      122      122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
647 005454      045      116      045 NOMEM: .ASCIZ '#N#A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****N'
648 005550      045      116      045 M8186: .ASCIZ '#N#A ***** 11/23A SYSTEM *****N'
649 005641      045      116      045 M8189: .ASCIZ '#N#A ***** 11/23B SYSTEM *****N'
650
651
652
653
654
655
656
657
658

```

.SBTTL GLOBAL ERROR REPORT SECTION

```

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

```

```

659 005732      BGNMSG  NXRERR      ;NON EXISTANT DEVICE REGISTER.
005732
660 005732      PRINTX  #NXRX,NODEV ;NODEV = NEXM ADDRESS.
005732 013746 003106      MOV      NODEV, (SP)
005736 012746 003773      MOV      #NXRX, (SP)
005742 012746 000002      MOV      #2, (SP)
005746 010600      MOV      SP,RO
005750 104415      TRAP     C$PNTX
005752 062706 000006      ADD      #6,SP
661 005756 004737 005764      JSR      PC,EXTEND      ; PRINT EXTENSION IF REQUIRED.
662 005762      ENDMMSG
005762
005762 104423      L10002: TRAP     C$MSG
663
664
665
666
667
668 005764 005727      ;
669 005766 000000      ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
670 005770 001402      ; TO ANY OF THE ABOVE ERROR SIGNATURES.
671 005772 004777 177770      ;
672 005776      EXTEND: TST      (PC)+
005776 012746 004524      EXTA: 0          ; 0 NO EXTENSION.
006002 012746 000001      BEQ      1$
006006 010600      JSR      PC,@EXTA      ; APPEND EXTENSION TEXT.
1$: PRINTX #NULCR      ; PRINT A BLANK LINE
      MOV      #NULCR, (SP)
      MOV      #1, (SP)
      MOV      SP,RO

```

L3

TSV3 GLOBAL AREAS MACRO M1113 06 FEB 84 18:04
GLOBAL ERROR REPORT SECTION

SEQ 037

006010	104415		TRAP	C:PNTX
006012	062706	000004	ADD	44,SP
63 006016	000207		RTS	PC

```

675          .SBTTL PRITSSR PRINT TSSR CONTENTS
676
677          ;*
678          ;
679          ;ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
680          ;THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
681          ;BY A MESSAGE PRINTING ROUTINE
682          ;
683          ;INPUTS:
684          ;
685          ;       R1      CONTENTS OF TSSR
686          ;
687          ;SUBORDINATE ROUTINES:
688          ;
689          ;       CHKAMB  CHECK FOR AMBIGUOUS CONTENTS
690          ;
691          ;
692          ;
693          PRITSSR:
694          SAVREG          ;SAVE GENERAL REGISTERS
695          MOV R1,R4       ;SAVE THE TSSR CONTENTS
696          PRINTB @TSSRFOR,R4 ;PRINT THE CONTENTS OF TSSR
697          MOV R4,-(SP)
698          MOV @TSSRFOR,(SP)
699          MOV @2,-(SP)
700          MOV SP,R0
701          TRAP C$PNTB
702          ADD @6,SP
703          MOV R4,R0       ;GET TSSR BACK FOR CHKAMB
704          JSR PC,CHKAMB  ;ARE CONTENTS AMBIGUOUS ?
705          BCS 5$         ;BRANCH IF NOT
706          PRINTX @AMBTSSR ;SHOW CONTENTS ARE AMBIGUOUS
707          MOV @AMBTSSR,-(SP)
708          MOV @1,-(SP)
709          MOV SP,R0
710          TRAP C$PNTX
711          ADD @4,SP
712          MOV R4,R3       ;CONTENTS OF TSSR
713          BIC @HIADDR!FATERR!TERCLS,R3 ;CLEAR ALL MULTIPLE BIT FIELDS
714          BEQ 20$        ;NO BITS ARE SET
715          MOV @TMPBFR,R2 ;TEMPORARY ASCII BUFFER
716          MOV @TSSRBIT,R1 ;ASCII EQUIVALENT OF BITS
717          TST R3         ;REMAINING BITS TO CONVERT
718          BEQ 15$        ;BRANCH WHEN ALL ARE DONE
719          CLC           ;CLEAR CARRY FOR SHIFT
720          ROL R3         ;SHIFT NEXT BIT TO CARRY
721          BCC 13$        ;BRANCH IF BIT NOT SET
722          MOV (R1),R0    ;POINTER TO BIT DEFINITION
723          MOV (R0)+,(R2)+ ;MOVE ASCII TO BUFFER
724          BNE 11$        ;MOVE ALL BITS
725          MOV @'-1(R2) ;INSERT A COMMA TO TERMINATE
726          TST (R1)+     ;POINT TO NEXT DESCRIPTION
727          BR 10$         ;GET THE REMAINING BITS
728          CLRB (R2)     ;TERMINATE THE LINE
729          PRINTX @TSSDEF,@TMPBFR ;PRINT THE BIT DEFINITIONS
730          MOV @TMPBFR,-(SP)
731          MOV @TSSDEF,(SP)

```



```

006104 012746 000002      MOV    #2,-(SP)
006170 010600      MOV    SP,R0
006172 104415      TRAP  C$PNTX
006174 062706 000006      ADD    #6,SP
719
720 006200 010403      20$:  MOV    R4,R3          ;GET THE TSSR CONTENTS
721 006202 042703 177761      BIC    #1CTERCLS,R3   ;CLEAR ALL BUT TERMINATION
722 006206 016303 006754      MOV    TCOCOD(R3),R3  ;GET THE TERMINATION CODE MEANING
723 006212      PRINTX #TCOASC,R3    ;PRINT THE TERMINATION CODE
      006212 010346      MOV    R3,-(SP)
      006214 012745 006554      MOV    #TCOASC,-(SP)
      006220 012746 000002      MOV    #2,-(SP)
      006224 010600      MOV    SP,R0
      006226 104415      TRAP  C$PNTX
      006230 062706 000006      ADD    #6,SP
724 006234 010403      MOV    R4,R3          ;TSSR CONTENTS AGAIN
725 006236 042703 177717      BIC    #1CFATERR,R3  ;CLEAR ALL BUT FATAL TERMINATION
726 006242 001416      BEQ    25$           ;DON'T PRINT IF ZERO
727 006244 006203      ASR    R3
728 006246 006203      ASR    R3
729 006250 006203      ASR    R3
730 006252 016303 007314      MOV    TSFCOD(R3),R3 ;ALINE TERMINATION CODE FOR INDEX
731 006256      PRINTX #TFCASC,R3   ;GET THE FATAL TERMINATION CODE
      006256 010346      MOV    R3,-(SP)
      006260 012746 006615      MOV    #TFCASC,-(SP)
      006264 012746 000002      MOV    #2,-(SP)
      006270 010600      MOV    SP,R0
      006272 104415      TRAP  C$PNTX
      006274 062706 000006      ADD    #6,SP
732 006300 042704 176377      25$:  BIC    #1CHIADDR,R4  ;CLEAR ALL BUT EXTENDED ADDRESS
733 006304 001411      BEQ    30$           ;DON'T PRINT IF ZERO
734 006306      PRINTX #TEXASC,R4   ;PRINT THE EXTENDED ADDRESS BITS
      006306 010446      MOV    R4,-(SP)
      006310 012746 006513      MOV    #TEXASC,-(SP)
      006314 012746 000002      MOV    #2,-(SP)
      006320 010600      MOV    SP,R0
      006322 104415      TRAP  C$PNTX
      006324 062706 000006      ADD    #6,SP
735 006330 013703 002172      30$:  MOV    EPRTSW,R3     ;PRINT MESSAGE BUFFER ADDRESS
736 006334      PRINTX R3           ;PRINT PROPER MESSAGE
      006334 010346      MOV    R3,-(SP)
      006336 012746 000001      MOV    #1,-(SP)
      006342 010600      MOV    SP,R0
      006344 104415      TRAP  C$PNTX
      006346 062706 000004      ADD    #4,SP
737 006352 000207      RTS    PC            ;RETURN TO CALLER
738
753 006354      045      116      045  EPRT1:  .ASCIZ  'NSA *****CHECK TRANSPORT*****'
754 006413      045      116      045  EPRT2:  .ASCIZ  'NSA *****CHECK PARITY SWITCH IN TRANSPORT*****'
756 006473      045      116      045  TSSRFOR: .ASCIZ  'NSA TSSR = #06'
757 006513      045      116      045  TEXASC:  .ASCIZ  'NSA Extended Address Bits = #06'
758 006554      045      116      045  TCOASC:  .ASCIZ  'NSA Termination Class Code = #1'
759 006615      045      116      045  TFCASC:  .ASCIZ  'NSA Fatal Termination Class Code = #1'
760 006664      045      116      045  TSSDEF:  .ASCIZ  'NSA TSSR Bits Set: #1'
761 006713      045      116      045  AMBTSSR: .ASCIZ  'NSA TSSR Contents Are Ambiguous'
762
763 006754 006774 007017 007045 TCOCOD: .EVEN
      .WORD  1$,2$,3$,4$,5$,6$,7$,8$
  
```

```

764 006774      116      157      162 18: .ASCIZ 'Normal Termination
765 007017      124      145      162 21: .ASCIZ 'Termination Condition'
766 007045      124      141      160 31: .ASCIZ 'Tape Status Alert'
767 007067      106      165      156 41: .ASCIZ 'Function Reject'
768 007107      122      145      143 51: .ASCIZ 'Recoverable Error - Tape Position One Record Down
769 007171      122      145      143 61: .ASCIZ 'Recoverable Error - Tape Was Not Moved
770 007240      125      156      162 71: .ASCIZ 'Unrecoverable Error'
771 007264      106      141      164 81: .ASCIZ 'Fatal Controller Error'
772          .EVEN
773
774 007314 007324 007360 007371 TSFCOD: .WORD 18,21,31,41
775 007324      111      156      164 11: .ASCIZ 'Internal Diagnostic Failure'
776 007360      122      145      163 21: .ASCIZ 'Reserved'
777 007371      102      165      163 31: .ASCIZ 'Bus Interface or Sanity Check Error'
778 007435      122      145      163 41: .ASCIZ 'Reserved'
779          .EVEN
780          .SBTTL PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
781
782
783          ;
784          ; THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
785          ; THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
786          ;
787          ; INPUT:
788          ;
789          ; R0      NUMBER OF WORDS IN PACKET
790          ; R3      HIGH ORDER COMMAND PACKET ADDRESS
791          ; R4      ADDRESS OF COMMAND PACKET
792          ;
793          ; NOTE: R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
794          ;
795          PRIPKT:
796          SAVREG          ;SAVE THE REGISTERS
797          MOV R0,R5        ;SAVE NO. OF WORDS IN PACKET
798          TST KTENABLE    ;ABOVE 28K UNDER TEST?
799          BNE 101         ;BR IF YES
800          CLR R3          ;SET HIGH ORDER ADDRESS TO 0
801          MOV R3,R1       ;COPY HIGH ORDER ADDRESS
802          MOV R4,R0       ;GET LOWER ADDRESS
803          ROL R0          ;SHIFT BIT 15 INTO C BIT
804          ROL R1          ;AND INTO HIGH ORDER.
805          PRINTB @PKTADD,R1,R4 ;PRINT PACKET ADDRESS
806          MOV R4,(SP)
807          MOV R1,(SP)
808          MOV @PKTADD,(SP)
809          MOV @3,(SP)
810          MOV SP,R0
811          TRAP C1PNTB
812          ADD @10,SP
813          MOV R3,R0      ;GET HIGH ORDER ADDRESS
814          BEQ 201       ;BR IF NOT ABOVE 28K.
815          MOV R4,R1     ;GET LOW ORDER ADDRESS
816          JSR PC,SETMAP ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
817          MOV R0,R4     ;GET RETURNED PAR6 ADDRESS BIAS
818          CLR R1        ;SAVE WORD NUMBER
819          MOV (R4),R2    ;GET PACKET CONTENTS
820          PRINTB @PKTFRM,R1,R2 ;PRINT THE DATA

```

TSV3 GLOBAL AREAS MACRO M1113 06 FEB 84 18:04
 PRIPKT PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

SEQ 041

```

007540 010246      MOV      R2,(SP)
007542 010146      MOV      R1,-(SP)
007544 012746 007574  MOV      @PKTFRM,-(SP)
007550 012746 000003  MOV      @3,-(SP)
007554 010600      MOV      SP,R0
007556 104414      TRAP    C:PNTB
007560 062706 000010  ADD      @10,SP
814 007564 005201      INC      R1          ;NEXT WORD NUMBER
815 007566 020105      CMP      R1,R5      ;DONE ALL PACKET WORDS?
816 007570 002762      BLT     25         ;LOOP TILL ALL DONE
817 007572 000207      RTS     PC          ;RETURN
818
819 007574      045      116      045  PKTFRM: .ASCIZ  'NNA Packet Word #D1NA = #06'
820 007632      045      116      045  PKTADD: .ASCIZ  'NNA Packet Address = #01#05'
821
822                          .EVEN
823                          .SBTTL  PRIBXOR  PRINT EXPD, RECV AND XOR BYTE
824
825                          ;*
826                          ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
827                          ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
828
829                          ;INPUTS:
830
831                          ;      R1      RECEIVED DATA
832                          ;      R2      EXPECTED DATA
833
834                          ;OUTPUT:
835
836                          ;      R0      XOR OF EXPECTED/RECEIVED DATA
837
838                          ;
839
840 007670      PRIBXOR::
841 007670      SAVREG
842 007674 010203      MOV      R2,R3          ;SAVE THE REGISTERS
843 007676      XOR      R1,R3          ;EXPECTED DATA
844 007706 012700 177400  MOV      @C<377>,R0     ;FORM THE EXCLUSIVE OR
845 007712 040001      BIC     R0,R1          ;BYTE MASK
846 007714 040002      BIC     R0,R2          ;SAVE LOW BYTE RECV
847 007716 040003      BIC     R0,R3          ;SAVE LOW BYTE EXPD
848 007720      PRINTB @XORBFOR,R2,R1,R3 ;SAVE LOW BYTE XOR
849                          ;PRINT THE MESSAGE
850 007720 010346      MOV      R3,(SP)
851 007722 010146      MOV      R1,(SP)
852 007724 010246      MOV      R2,(SP)
853 007726 012746 007752  MOV      @XORBFOR,(SP)
854 007732 012746 000004  MOV      @4,(SP)
855 007736 010600      MOV      SP,R0
856 007740 104414      TRAP    C:PNTB
857 007742 062706 000012  ADD      @12,SP
858 007746 010300      MOV      R3,R0          ;R0 HAS XOR ON RETURN
859 007750 000207      RTS     PC          ;RETURN TO CALLER
860
861 007752      045      116      045  XORBFOR: .ASCIZ  'NNA EXPD: #03NA RECV: #03NA XOR: #03'
862                          .EVEN
863                          .SBTTL  PRIBXOR  PRINT EXPD, RECV AND XOR
864
865

```

```

856 ;*
857 ;
858 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
859 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
860 ;
861 ;INPUTS:
862 ;
863 ;      R1      RECEIVED DATA
864 ;      R2      EXPECTED DATA
865 ;
866 ;OUTPUT:
867 ;
868 ;      R0      XOR OF EXPECTED/RECEIVED DATA
869 ;
870 ;
871 ;
872 010020      PRI XOR::
873 010020      SAVREG                      ;SAVE THE REGISTERS
874 010024      010203      MOV      R2,R3      ;EXPECTED DATA
875 010026      XOR      R1,R3      ;FORM THE EXCLUSIVE OR
876 010036      PRINTB     #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
      010036      MOV      R3,-(SP)
      010040      010146      MOV      R1,-(SP)
      010042      010246      MOV      R2,-(SP)
      010044      012746      010070      MOV      #XORFOR,-(SP)
      010050      012746      000004      MOV      #4,-(SP)
      010054      010600      MOV      SP,R0
      010056      104414      TRAP     C:PNTB
      010060      062706      000012      ADD      #12,SP
877 010064      010300      MOV      R3,R0      ;R0 HAS XOR ON RETURN
878 010066      000207      RTS      PC      ;RETURN TO CALLER
879
880 010070      045      !16      045      XORFOR: .ASCIZ 'N#A EXPD: #06#A RECV: #06#A XOR: #06'
881 .EVEN
882 .SBTTL PRIEQU      PRINT BIT NUMBERS AS ASCII EQUIVALENT
883
884 ;*
885 ;
886 ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
887 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
888 ;
889 ;INPUTS:
890 ;
891 ;      R0      OCTAL VALUE TO CONVERT
892 ;      R1      TABLE OF POINTERS TO ASCII EQUIVALENT
893 ;
894 ;
895 ;
896 010136      PRIEQU:
897 010136      SAVREG                      ;SAVE THE REGISTERS
898 010142      000207      RTS      PC      ;RETURN TO CALLER
899
900 .SBTTL PRI RAM      PRINT RAM ADDRESS
901 ;*
902 ;
903 ;PRINT CONTROLLER RAM ADDRESS.
904 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
    
```

```

905
906 ; INPUTS:
907 ;
908 ; R4 RAM ADDRESS
909 ;
910 ;
911 PRIRAM:
912 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
913 PRINTB #RAMFOR,R4 ;PRINT RAM ADDRESS IN ERROR
    010150 010446 MOV R4,-(SP)
    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
914 010172 000207 RTS PC ;RETURN
915
916 010174 045 116 045 RAMFOR: .ASCIZ 'NWA CONTROLLER RAM ADDRESS = #06'
917 .EVEN
918
919 .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
920 ;
921 ;
922 ;PRINT MEMORY ADDRESS
923 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
924 ;
925 ; IMPLICIT INPUTS
926 ;
927 ; ERRHI - HIGH ORDER ADDRESS
928 ; ERRLO - LOW ORDER ADDRESS
929 ;
930 ;
931 PRIADD:
932 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
933 MOV ERRHI,R0 ;GET HIGH ADDRESS
934 MOV ERRLO,R1 ;GET LOW ADDRESS
935 MOV R1,R2 ;COPY LOW ADDRESS
936 ROL R1 ;SHIFT BIT 15 TO C BIT
937 ROL R0 ;SHIFT INTO HIGH ORDER
938 PRINTB #PRIA0,R0,R2 ;PRINT MEMORY ADDRESS IN ERROR
    010260 010246 MOV R2,(SP)
    010262 010046 MOV R0,-(SP)
    010264 012746 010306 MOV #PRIA0,(SP)
    010270 012746 000003 MOV #3,-(SP)
    010274 010600 MOV SP,R0
    010276 104414 TRAP C:PNTB
    010300 062706 000010 ADD #10,SP
939 010304 000207 RTS PC ;RETURN
940
941 010306 045 116 045 PRIA0: .ASCIZ 'NWA MEMORY ERROR ADDRESS = #01#05'
942 .EVEN
943
944 .SBTTL PRITADD PRINT MEMORY TEST ADDRESS
945 ;
946 ;
947 ;PRINT MEMORY ADDRESS
948 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.

```

```

949
950 ; IMPLICIT INPUTS
951 ;
952 ; ERRHI HIGH ORDER ADDRESS
953 ; ERRLO - LOW ORDER ADDRESS
954 ;
955 ;
956 010352 PRITADD: SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
957 010352 MOV ERRHI,R2 ;GET HIGH ADDRESS
958 010356 013702 002230 MOV ERRLO,R1 ;GET LOW ADDRESS
959 010362 013701 002232 ;MOV R1,R2 ;COPY LOW ADDRESS
960 ;ROL R1 ;SHIFT BIT 15 TO C BIT
961 ;ROL R0 ;SHIFT INTO HIGH ORDER
962 ;PRINTB @PRIT0,R1 ;PRINT MEMORY ADDRESS LOW IN ERROR
963 010366 MOV R1,-(SP)
010370 012746 010434 MOV @PRIT0,-(SP)
010374 012746 000002 MOV @2,-(SP)
010400 010600 MOV SP,R0
010402 104414 TRAP C:PNTB
010404 062706 000006 ADD #6,SP
964 010410 PRINTB @PRIT1,R2 ;PRINT MEMORY ADDRESS HIGH IN ERROR
010410 010246 MOV R2,-(SP)
010412 012746 010477 MOV @PRIT1,-(SP)
010416 012746 000002 MOV @2,-(SP)
010422 010600 MOV SP,R0
010424 104414 TRAP C:PNTB
010426 062706 000006 ADD #6,SP
965 010432 000207 RTS PC ;RETURN
966
967 010434 045 116 045 PRIT0: .ASCIZ 'NWA MEMORY TEST ADDRESS LOW = #06'
968 010477 045 116 045 PRIT1: .ASCIZ 'NWA MEMORY TEST ADDRESS HIGH = #06'
969 .EVEN
970 .SBTTL SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND
971
972 ;*
973 ;
974 ;ROUTINE TO ISSUE A SPACE RECORDS
975 ;COMMAND (FORWARD OR REVERSE)
976 ;
977 ;INPUT:
978 ;
979 ; R3 NUMBER OF RECORDS TO BE SPACED OVER
980 ; BIT15 CONTROLS DIRECTION
981 ; BIT15 = 0 IS FORWARD
982 ; BIT15 = 1 IS REVERSE
983 ; R5 FIRST DEVICE UNIBUS ADDRESS
984 ;
985 ; REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
986 ;
987 ;OUTPUT:
988 ;
989 ; CARRY SET - SPACE RECORDS COMMAND OK
990 ; CLR - SPACE RECORDS FAILED
991 ;
992 ;
993 ; R0 THE CONTENTS OF R4 IS MOVED TO R0
    
```

TSV3 GLOBAL AREAS MACRO M1113 06 FEB 84 18:04
 SPACE SPACE RECORDS (FORWARD AND REVERSE) COMMAND

SEQ 045

```

994
995
996      ;IMPLICIT OUTPUT:
997
998      ;      TAPE HAS BEEN MOVED
999
1000     ;SIDE EFFECTS:
1001
1002
1003
1004
1005 010544      SPACE::
1006 010544      SAVREG
1007 010550 012737 000764 010740      MOV      #500.,SDELAY      ;SAVE THE GENERAL REGISTERS
1008 010556 012737 140010 010730      MOV      #140010.80$      ;SET UP DELAY
1009 010564 005703      TST      R3      ;SET UP COMMAND, SPACE FORWARD
1010 010566 100403      BMI      5$      ;CHECK FOR DIRECTION
1011 010570 010337 010732      MOV      R3,90$      ;BR, IF REVERSE INDICATED
1012 010574 000407      BR      10$      ;LOAD UP NUMBER OF RECORDS TO SPACE
1013 010576 042703 100000      5$:      BIC      #BIT15,R3      ;GO DO COMMAND
1014 010602 010337 010732      MOV      R3,90$      ;CLEAR DIRECTION BIT
1015 010606 052737 000400 010730      BIS      #BIT8.80$      ;LOAD UP NUMBER OF RECORDS TO SPACE
1016 010614 012704 010730      10$:      MOV      #80$,R4      ;SET REVERSE BIT IN COMMAND PACKET
1017 010620 010465 000000      MOV      R4,TSSB(R5)      ;SET UP R4 WITH PACKET ADDRESS
1018 010624 004737 016330      15$:      JSR      PC,WAITF      ;SEND OUT COMMAND
1019 010630 103420      BCS      20$      ;WAIT FOR SSR
1020 010632      DELAY      250      ;BR, IF SSR IS SET AND OK
      010632 012727 000250      MOV      #250,(PC).      ;DELAY ABOUT .25 SECONDS
      010636 000000      .WORD      0
      010640 013727 002116      MOV      L$DLY,(PC).
      010644 000000      .WORD      0
      010646 005367 177772      DEC      -6(PC)
      010652 001375      BNE      .-4
      010654 005367 177756      DEC      -22(PC)
      010660 001367      BNE      .-20
1021 010662 005337 010740      DEC      SDELAY      ;BUMP DELAY COUNTER DOWN
1022 010666 001356      BNE      15$      ;BR, IF MORE DELAY
1023 010670 000411      BR      60$      ;BR IF TROUBLE CARRY = CLEAR
1024 010672 016501 000002      20$:      MOV      TSSR(R5),R1      ;READ TSSR
1025 010676 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
1026 010702 020201      25$:      CMP      R2,R1      ;ARE THEY OK
1027 010704 001401      BEQ      40$      ;BR, IF EQUAL = OK
1028 010706 000402      BR      60$      ;TROUBLE EXIT
1029 010710 000261      40$:      SEC      ;SET CARRY NO TROUBLE
1030 010712 000401      BR      70$      ;EXIT
1031 010714 000241      60$:      CLC      ;CARRY CLEAR = ERROR
1032 010716      70$:
1033 010716 010400      MOV      R4,R0      ;PASS PACKET ADDRESS
1034 010720 000207      RTS      PC      ;RETURN

```

```

1036      ;
1037      ;
1038      ;
1039      ; PACKET FOR SPACE COMMAND
1040      ;
1041      ;
1042      ;           .=<..10>&177770
1043      ;
1044      ;
1045      ; COMMAND WORD
1046      010730 000000 80$: .WORD
1047      ; NUMBER OF RECORDS TO BE SPACED OVER WORD
1048      010732 000000 90$: .WORD
1049      010734 000000 .WORD
1050      010736 000000 .WORD
1051      010740 000000 SDELAY: .WORD 0 ; DELAY COUNTER
1052      ;
1053      ;           .EVEN
1054      ;           .SBTTL WRTCHR WRITE CHARACTERISTICS COMMAND
1055      ;*
1056      ;
1057      ; ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1058      ; COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1059      ;
1060      ; INPUT:
1061      ;
1062      ;           R4 ADDRESS OF PACKET FROM TEST
1063      ;           R5 FIRST DEVICE UNIBUS ADDRESS
1064      ;           REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1065      ;
1066      ; OUTPUT:
1067      ;
1068      ;           R0 TSSR CONTENTS
1069      ;           CARRY SET - WRITE CHARACTERISTICS COMMAND OK
1070      ;           CLR - WRITE CHARACTERISTICS FAILED
1071      ;
1072      ; IMPLICIT OUTPUT:
1073      ;
1074      ;           MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
1075      ;           SOFTWARE SWITCHES SET AS FOLLOWS:
1076      ;           EXTFEA = EXTENDED FEATURES PRESENT
1077      ;           BENBSW = BUFFER ENABLE SWITCH ON OR OFF
1078      ;
1079      ;
1080      ; SIDE EFFECTS:
1081      ;
1082      ;
1083      ; -
1084      ;
1085      010742 WRTCHR: :
1086      010742 SAVREG ; SAVE THE GENERAL REGISTERS
1087      010746 005037 002222 CLR BENBSW ; CLEAR BUFFER ENABLE SWITCH
1088      010752 005037 002220 CLR EXTFEA ; CLEAR EXTENDED FEATURES SW SWITCH
1089      010756 010465 000000 10$: MOV R4,TSDB(R5) ; SEND OUT COMMAND
1090      010762 004737 016416 JSR PC,CHKTSSR ; WAIT FOR SSR
1091      010766 103401 BCS 20$ ; BR, IF SSR IS SET AND OK
1092      010770 000435 BR 60$ ; BR IF TROUBLE CARRY = CLEAR
1093      010772 016501 000002 20$: MOV TSSR(R5),R1 ; READ TSSR
1094      010776 012702 000200 MOV @SSR,R2 ; SET UP EXPECTED

```



```

011122 012727 000372      MOV      #250.,(PC).
011126 000000      .WORD   0
011130 013727 002116      MOV      L$DLY,(PC).
011134 000000      .WORD   0
011136 005367 177772      DEC      6(PC)
011142 001375      BNE      . 4
011144 005367 177756      DEC      -22(PC)
011150 001367      BNE      . 20
1152 011152 005303      DEC      R3          ;BUMP COUNTER DOWN
1153 011154 001357      BNE      10$        ;KEEP GOING
1154 011156 000241      CLC          ;CLEAR CARRY TO SET ERROR
1155 011160 010400 20$:  MOV      R4,R0      ;PASS THE PACKET ADDRESS
1156 011162 000207      RTS      PC         ;RETURN
1157
1159          011170      RWPACK: .-<..*10>E177770
1161 011170      .WORD   102010      ;POSTION COMMAND (REWIND)
1162 011170 102010      .WORD   0           ;NOT USED
1163 011172 000000      .SBTTL  CKRAM      - COMPARE RAM TO I/O PACKET
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193 011174
1194 011174
1195 011200 012701 002234
1196 011204 012702 000201
1197 011210 005003
1198 011212 004737 016416
1199 011216 112765 000000 000000
1200 011224 004737 016416 10$:
1201 011230 010265 000000
1202 011234 004737 016416

;
;
;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
;
;INPUT:
;
;      R4      ADDRESS OF THE COMMAND PACKET
;      R5      FIRST DEVICE UNIBUS ADDRESS
;
;OUTPUT:
;
;      CARRY   SET - RAM MATCHES PACKET
;             CLR - RAM DOES NOT MATCH PACKET
;
;IMPLICIT OUTPUT:
;
;      THE TABLE RAMDATA IS FILLED WITH THE
;      DATA HELD IN RAM.
;      RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
;
;SIDE EFFECTS:
;
;      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
;
;
;
CKRAM::
SAVREG          ;SAVE THE GENERAL REGISTERS
MOV      #RAMDATA,R1      ;ADDRESS TO SAVE THE RAM DATA
MOV      #RMPKTBEG,R2     ;BYTE ADDRESS OF FIRST RAM DATA
CLR      R3              ;CLEAR THE ERROR FLAG
JSR      PC,CHKTSSR       ;WAIT FOR SSR
MOV      #0,TSDB(R5)      ;SET MAINTENANCE MODE
JSR      PC,CHKTSSR       ;WAIT FOR SSR TO SET
MOV      R2,TSDB(R5)      ;SELECT NEXT RAM ADDRESS
JSR      PC,CHKTSSR       ;WAIT FOR SSR TO SET
    
```

```

1203 011240 116511 0G0000      MOVB   TSBA(R5),(R1)      ;READ THE RAM DATA
1204 011244 122124      CMPB   (R1)*,(R4)*      ;COMPARE TO EXPECTED
1205 011246 001401      BEQ    20$              ;BRANCH IF OK
1206 011250 005203      INC    R3               ;SET ERROR FLAG
1207 011252 005202      20$:  INC    R2          ;ADDRESS OF NEXT RAM LOCATION
1208 011254 020227 000210      CMP    R2,#RMPKTEND    ;REACHED END YET ?
1209 011260 003761      BLE   10$              ;BRANCH TILL ALL READ
1210 011262 005703      TST   R3               ;WAS AN ERROR FOUND ?
1211 011264 001402      BEQ   30$              ;BRANCH IF NOT
1212 011266 000241      CLC                      ;CLEAR CARRY TO SHOW ERROR
1213 011270 000401      BR    50$              ;AND EXIT
1214 011272 000261      30$:  SEC                      ;SHOW GOOD COMPARE
1215 011274 012737 000010 002274 50$:  MOV    #8.,RAMSIZ      ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
1216 011302 000207      RTS    PC               ;RETURN
1217                                     .SBTTL CKRAM2 COMPARE RAM TO I/O CHARACTERISTICS DATA
1218                                     ;*
1219                                     ;
1220                                     ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
1221                                     ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
1222                                     ;
1223                                     ;INPUT:
1224                                     ;
1225                                     ;      R4 ADDRESS OF THE CHARACTERISTICS DATA
1226                                     ;      R5 FIRST DEVICE UNIBUS ADDRESS
1227                                     ;
1228                                     ;OUTPUT:
1229                                     ;
1230                                     ;      CARRY SET - RAM MATCHES PACKET
1231                                     ;      CLR - RAM DOES NOT MATCH PACKET
1232                                     ;
1233                                     ;IMPLICIT OUTPUT:
1234                                     ;
1235                                     ;      THE TABLE RAMDATA IS FILLED WITH THE
1236                                     ;      DATA HELD IN RAM.
1237                                     ;      RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
1238                                     ;
1239                                     ;SIDE EFFECTS:
1240                                     ;
1241                                     ;      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1242                                     ;
1243                                     ;-
1244
1245 011304      CKRAM2::
1246 011304      SAVREG                      ;SAVE THE GENERAL REGISTERS
1247 011310 012701 002234      MOV    #RAMDATA,R1      ;ADDRESS TO SAVE THE RAM DATA
1248 011314 012702 000167      MOV    #RMCHBEG,R2     ;BYTE ADDRESS OF FIRST RAM DATA
1249 011320 005003      CLR    R3               ;CLEAR THE ERROR FLAG
1250 011322 004737 016416      JSR    PC,CHKTSSR      ;WAIT FOR SSR
1251 011326 112765 000000 000000      MOVB   #0,TSDB(R5)     ;SET MAINTENANCE MODE
1252 011334 004737 016416      10$:  JSR    PC,CHKTSSR      ;WAIT FOR SSR TO SET
1253 011340 010265 000000      MOV    R2,TSDB(R5)     ;SELECT NEXT RAM ADDRESS
1254 011344 004737 016416      JSR    PC,CHKTSSR      ;WAIT FOR SSR TO SET
1255 011350 116511 000000      MOVB   TSBA(R5),(R1)   ;READ THE RAM DATA
1256 011354 122124      CMPB   (R1)*,(R4)*     ;COMPARE TO EXPECTED
1257 011356 001401      BEQ   20$              ;BRANCH IF OK
1258 011360 005203      INC    R3               ;SET ERROR FLAG
1259 011362 005202      20$:  INC    R2          ;ADDRESS OF NEXT RAM LOCATION

```


TSV3 GLOBAL AREAS MACRO M1113 06 FEB 84 18:04
 CKMSG COMPARE WRITE CHAR. MESSAGE BUFFERS

SEQ 051

```

1317 011520 020427 000014      CMP      R4,#014      ;DONE FIRST 7 WORDS?
1318 011524 003764           BLE      15#         ;BR IF NO
1319 011526 032765 000200 000012  BIT      #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
1320 011534 001403           BEQ      50#         ;BR IF NO
1321 011536 020427 000016      CMP      R4,#016      ;DONE EXTENDED FEATURES WORD?
1322 011542 003755           BLE      15#         ;BR IF NO
1323 011544 005703           50$:    TST      R3         ;ANY ERRORS SEEN?
1324 011546 001402           BEQ      55#         ;BR IF NO
1325 011550 000241           CLC                     ;SET FAILURE
1326 011552 000401           BR      60#         ;
1327 011554 000261           55$:    SEC                     ;SET SUCCESS
1328 011556 000207           60$:    RTS      PC         ;RETURN
1329                                     .SBTTL  CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
1330                                     ;*
1331                                     ;
1332                                     ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
1333                                     ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1334                                     ;ERROR PRINT ROUTINES.
1335                                     ;
1336                                     ;INPUT:
1337                                     ;
1338                                     ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1339                                     ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1340                                     ;      R2      EXPD MESSAGE BUFFER ADDRESS
1341                                     ;      R3      NUMBER OF BYTES TO COMPARE
1342                                     ;
1343                                     ;OUTPUT:
1344                                     ;
1345                                     ;      CARRY   SET - MESSAGE BUFFERS MATCH
1346                                     ;           CLR - MESSAGE BUFFERS DON'T MATCH
1347                                     ;
1348                                     ;IMPLICIT OUTPUT:
1349                                     ;
1350                                     ;      EXPMSG   BUFFER IS SET TO EXPD DATA
1351                                     ;      RECVMSG  BUFFER IS SET TO RECV DATA
1352                                     ;      RCVHIADD  SET TO HIGH ORDER ADDRESS OF RECV
1353                                     ;      RCVLOADD  SET TO LOW ORDER ADDRESS OF RECV
1354                                     ;
1355                                     ;-
1356 011560  CKMSG2::
1357 011560  SAVREG                    ;SAVE R1-R5 UNTIL NEXT RETURN
1358 011564 020327 000144      CMP      R3,#RECVMSG-EXPMSG;#00 IS COUNT ABOVE MAX ALLOWED?
1359 011570 003412           BLE      5#         ;#00 BR IF NO
1360 011572 012703 000144      MOV      #RECVMSG-EXPMSG,R3;#00
1361 011576  PRINTF #DEFJGMSG ;#00
1362 011576 012746 011712      MOV      #DEBUGMSG,-(SP)
1363 011602 012746 000001      MOV      #,-(SP)
1364 011606 010600           MOV      SP,R0
1365 011610 104417           TRAP    C#PNTF
1366 011612 062706 000004      ADD      #4,SP
1367 011616 010037 002276  5$:    MOV      R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
1368 011622 010137 002300      MOV      R1,RCVLOAD  ;SAVE RECV LOW ADDRESS
1369 011626 005737 003126      TST      #TENABLE   ;TESTING ABOVE 28K?
1370 011632 001403           BEQ      10#        ;BR IF NO
1371 011634 004737 017376      JSR      PC,SETMAP  ;RETURN ADDRESS BIASED TO PAR6 IN R0
1372 011640 010001           MOV      R0,R1      ;GET RETURNED ADDRESS BIASED TO PAR6
1373 011642 005004           10$:    CLR      R4         ;WORD IN BUFFER

```

```

1369 011644 005005          CLR      R5          ;CLEAR ERROR SEEN FLAG
1370 011646 111264 002314 15$:  MOVR    (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1371 011652 111164 002460      MOVB   (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
1372 011656 122221          CMPB   (R2)*,(R1)*    ;EXPD EQUAL RECV?
1373 011660 001401          BEQ    25$          ;BR IF YES
1374 011662 005205          INC    R5          ;SET ERROR SEEN FLAG
1375 011664 062704 000001 25$:  ADD    @1,R4       ;POINT TO NEXT BYTE
1376 011670 020403          CMP    R4,R3       ;DONE ALL BYTES?
1377 011672 002001          BGE   50$          ;BR IF YES
1378 011674 000764          BR    15$          ;DO NEXT BYTE
1379 011676 005705          50$:  TST    R5          ;ANY ERRORS SEEN?
1380 011700 001402          BEQ   55$          ;BR IF NO
1381 011702 000241          CLC                   ;SET FAILURE
1382 011704 000401          BR    60$          ;
1383 011706 000261          55$:  SEC                   ;SET SUCCESS
1384 011710 000207          60$:  RTS    PC          ;RETURN
1385
1386 011712          120      122      117  DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR C MSG2 MESSAGE BUFFER EXCEEDED ;@@D
1387 012002          045      116      045  FERCM: .ASCII /WNA ***/
1388 012013          040      040      124  ERCM: .ASCIZ / TSSR ERROR CODE REC D * /
1389 012046          056      056      056  SIMSG: .ASCIZ /... AFTER DOING SOFT INIT/
1390 012101          124      105      123  TINERR: .ASCIZ /TEST: .../
1391
1392
1393          ;*
1394          ;
1395          ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
1396          ;
1397          ;INPUT:
1398          ;
1399          ;      R1      CONTENTS OF TSSR AT ERROR
1400          ;
1401          ;SIDE EFFECTS:
1402          ;
1403          ;      EXECUTES DROP UNIT TO CEASE TESTING
1404          ;
1405          ;
1406          ;
1407 012114          BGNMSG  SFIMSG
1408 012114 004737 006020  SFIMSG: J3H    PC,PRIT,JSR    ;PRINT CONTENTS OF TSSR REGISTER
1409 012120 004737 017262  JSR    PC,CKDROP    ;DROP UNIT, IF ALLOWED
1410 012124          ENDMMSG
1411          L10003:
1412          TRAP   C:MSG
1413
1414          ;*
1415          ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1416          ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
1417          ;
1418          ;INPUTS:
1419          ;
1420          ;      R1      TSSR CONTENTS
1421          ;      R4      ADDRESS OF COMMAND PACKET
1422          ;

```

```

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

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

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

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

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

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

```

C5

1471
1472 012170
1473 012170
1474 012170 004737 006070
1475 012174 010200
1476 012176 010301
1477 012200 004737 014322
1478 012204
1479 012204
1480 012204 104423
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490 012206
1491 012206 004737 010352
1492 012212 016501 000002
1493 012216 004737 006020
1494 012222
1495 012222
1496 012222 104423
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508 012224
1509 012224 012700 000007
1510 012230 005737 002220
1511 012234 001402
1512 012236 012700 000010
1513 012242 004737 014632
1514 012246
1515 012246 104423
1516
1517
1518

```

;
; BGNMSG PKTMS
PKTMS:
; JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR
; MOV R2,R0 ;LOW ORDER ADDRESS
; MOV R3,R1 ;HIGH ORDER ADDRESS
; JSR PC,PRMESS ;PRINT THE MESSAGE BUFFER
; ENDMSG
L10007:
; TRAP CMSG ADDSSR PRINT TEST ADDRESS AND TSSR
; .SBTTL
; *
; PRINT ROUTINE TO PRINT THE CONTENTS OF
; TSSR AND A MEMORY TEST ADDRESS
;
; INPUTS:
;
; R5 FIRST DEVICE UNIBUS ADDRESS
; ERRHI HIGH ORDER MEMORY TEST ADDRESS
; ERRLO LOW ORDER MEMORY TEST ADDRESS
;
;
; BGNMSG ADDSSR
ADDSSR:
; JSR PC,PRITADD ;PRINT MEMORY TEST ADDRESS
; MOV TSSR(R5),R1 ;GET CURRENT TSSR
; JSR PC,PRITSSR ;PRINT THE CONTENTS OF TSSR REGISTER
; ENDMSG
L10010:
; TRAP CMSG
; .SBTTL MSGEXP - PRINT WRITE CHAR. EXPD RECV MESSAGE BUFFERS
; *
; PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
;
; IMPLICIT INPUTS:
;
; EXPMSG - EXPECTED MESSAGE BUFFER
; RECHMSG - RECEIVED MESSAGE BUFFER
; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
;
;
; BGNMSG MSGEXP
MSGEXP:
; MOV #7,R0 ;ASSUME NO EXT FEATURES
; TST EXTFEA ;EXT FEATURES SET?
; BEQ 5$ ;BR IF NO
; MOV #8.,R0 ;EXT FEATURE BUFFER IS 8 WORDS
; JSR PC,PRMSGEXP ;PRINT EXPD/RECV MESSAGE BUFFERS
; ENDMSG
5$:
L10011:
; TRAP CMSG FIFEXP PRINT FIFO EXP/RECV DATA
; .SBTTL
; *
; PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA

```



```

1519
1520          ;          R1          BYTE COUNT
1521          ;
1522          ;IMPLICIT INPUTS:
1523          ;
1524          ;          EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY
1525          ;          RECMMSG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
1526          ;
1527          ;          BGNMSG FIFEXP
1528          FIFEXP:
1528          012250 010146 012322 PRINTX #FIF1MSG,R1          ;PRINT BYTES TRANSFERRED
1528          012250 012746 000002 MOV R1,-(SP)
1528          012252 012746 000002 MOV #FIF1MSG,-(SP)
1528          012256 010600 000006 MOV #2,-(SP)
1528          012262 104415 000006 MOV SP,R0
1528          012264 062706 000006 TRAP C#PNTX
1528          012266 010600 000006 ADD #6,SP
1529          012272 012746 012371 PRINTX #FIF2MSG          ;PRINT HEADER MSG
1529          012272 012746 000001 MOV #FIF2MSG,-(SP)
1529          012276 010600 000004 MOV #1,-(SP)
1529          012302 104415 000004 MOV SP,R0
1529          012304 062706 000004 TRAP C#PNTX
1529          012306 010100 015202 ADD #4,SP
1529          012312 004737 015202 MOV R1,R0          ;GET BYTE COUNT
1531          012314 004737 015202 JSR PC,PRBYTEXP          ;PRINT FIFO BYTES IN ERROR
1532          012320
1532          012320
1532          012320 104423
1533          012322 045 116 045 FIF1MSG: .ASCIZ '##A NUMBER OF BYTES TRANSFERRED - #D2
1534          012371 045 116 045 FIF2MSG: .ASCIZ '##A FIFO DATA BYTES IN ERROR:'
1535          .EVEN
1536          .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
1537          ;*
1538          ;
1539          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
1540          ;
1541          ;
1542          ;IMPLICIT INPUTS:
1543          ;
1544          ;          EXPMSG - EXPECTED MESSAGE BUFFER
1545          ;          RECMMSG - RECEIVED MESSAGE BUFFER
1546          ;          RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1547          ;          RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1548          ;
1549          ;          BGNMSG MSGSTAT
1550          MSGSTAT:
1550          012430 012701 012472 MOV #STATCOD,R1          ;ASCII ADDRESS TABLE
1551          012434 012100 001410 10$: MOV (R1),R0          ;DONE ALL MSG LINES?
1552          012436 001410 000001 BEQ 20$          ;BR IF YES
1553          012440 010046 000001 PRINTX R0          ;PRINT STATUS BIT NAMES
1553          012440 012746 000001 MOV R0,-(SP)
1553          012442 010600 000001 MOV #1,(SP)
1553          012446 104415 000004 MOV SP,R0
1553          012450 062706 000004 TRAP C#PNTX
1553          012452 000766 000004 ADD #4,SP
1554          012456 000766 000012 BR 10$          ;DO ANOTHER MSG LINE
1555          012460 012700 000012 20$: MOV #10,R0          ;NUMBER OF WORDS IN A READ STATUS BUFFER
  
```

TSV3 GLOBAL AREAS MACRO M1113 06 FEB 84 18:04
MSGSTAT PRINT STATUS HEADER AND MESSAGE BUFFERS

SEQ 056

```

1556 012464 004737 014632          JSR    PC,PRMSGEXP      ;PRINT EXPD/RCV MESSAGE BUFFERS
1557 012470          ENDMSG
      012470          L10013:
      012470 104423          TRAP    C#MSG
1558
1559 012472 012510 012552 012643  STATCOD:      .WORD    1#,2#,3#,4#,5#,6#,0
1560 012510      045      116      045 1#:.ASCIZ  '#NSA Tape Bus Signals in Word #8:'
1561 012552      045      116      045 2#:.ASCIZ  '#NSA PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
1562 012643      045      116      045 3#:.ASCIZ  '#NSA IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
1563 012734      045      116      045 4#:.ASCIZ  '#NSA IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
1564 013025      045      116      045 5#:.ASCIZ  '#NSA Tape Bus Signals in Word #9:'
1565 013067      045      116      045 6#:.ASCIZ  '#NSA DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
1566          .EVEN
1567
1568          .SBTTL MSGLOOP PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
1569
1570          ;*
1571          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1572          ;
1573          ;IMPLICIT INPUTS:
1574          ;
1575          ; EXPMSG - EXPECTED MESSAGE BUFFER
1576          ; RECMMSG - RECEIVED MESSAGE BUFFER
1577          ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1578          ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1579          ;
1580          BGNMSG MSGLOOP
1581 013144 013144          MSGLOOP:
1582 013150 012701 013206          MOV     @LOOPCOD,R1      ;ASCII ADDRESS TABLE
1583 013152 012100          10#:  MOV     (R1),R0      ;DONE ALL MSG LINES?
1584 013154 001410          BEQ     20#           ;BR IF YES
      013154 010046          PRINTX  R0             ;PRINT STATUS BIT NAMES
      013154 012746          MOV     RO, (SP)
      013156 010600          MOV     #1,-(SP)
      013162 010600          MOV     SP,R0
      013164 104415          TRAP    C#PNTX
      013166 062706 000004          ADD     #4,SP
1585 013172 000766          BR      10#           ;DO ANOTHER MSG LINE
1586 013174 012700 000012          20#:  MOV     #10,R0     ;NUMBER OF WORDS IN A READ STATUS BUFFER
1587 013200 004737 014632          JSR     PC,PRMSGEXP    ;PRINT EXPD/RCV MESSAGE BUFFERS
1588 013204          ENDMSG
      013204          L10014:
      013204 104423          TRAP    C#MSG
1589
1590 013206 013226 013301 013400  LOOPCOD:      .WORD    1#,2#,3#,4#,5#,6#,7#,0
1591 013226      045      116      045 1#:.ASCIZ  '#NSA Tape Bus Loopback Signals in Word #8:'
1592 013301      045      116      045 2#:.ASCIZ  '#NSA PARERR<15> IRESV2<14> IRESV1<13>'
1593 013400      045      116      045 3#:.ASCIZ  '#NSA IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
1594 013477      045      116      045 4#:.ASCIZ  '#NSA IWM =>IFMK<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
1595 013576      045      116      045 5#:.ASCIZ  '#NSA ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDP <04>'
1596 013675      045      116      045 6#:.ASCIZ  '#NSA IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
1597 013774      045      116      045 7#:.ASCIZ  '#NSA IGO =>IFPT<00>'
1598          .EVEN
1599          .SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
1600
1601          ;*
1602          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV

```

```

1603      ;
1604      ;
1605      ;IMPLICIT INPUTS:
1606      ;
1607      ;     EXPMSG - EXPECTED MESSAGE BUFFER
1608      ;     RECMSG - RECEIVED MESSAGE BUFFER
1609      ;     RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1610      ;     RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1611      ;
1612 014022      BGNMSG MSGSUB
1613 014022      MSGSUB:  MOV     #10.,R0      ;SIZE OF WRITE SUBSYSTEM BUFFER
1614 014026      012700 000012      JSR     PC,PRMSGEXP      ;PRINT EXPD/RCV MESSAGE BUFFERS
1615 014032      ENDMSG
1616      ;
1617      ;
1618      ;
1619      ;
1620      ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
1621      ;
1622      ;IMPLICIT INPUTS:
1623      ;
1624      ;     ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
1625      ;     ERRLO - MEMORY ERROR LOW ORDER ADDRESS
1626      ;     EXP   - EXPECTED DATA
1627      ;     RECV  - RECEIVED DATA
1628      ;
1629 014034      BGNMSG MEMADD
1630 014034      MEMADD:  JSR     PC,PRIADD      ;PRINT MEMORY ADDRESS IN ERROR
1631 014040      004737 010236      MOV     EXPD,R1          ;GET EXPD DATA
1632 014044      013701 002224      MOV     RECV,R2         ;GET RECEIVED DATA
1633 014050      004737 010020      JSR     PC,PRIXOR       ;PRINT EXPD/RCV
1634 014054      ENDMSG
1635      ;
1636      ;
1637      ;
1638      ;
1639      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1640      ;WHEN THE RAM DATA DOES NOT MATCH.
1641      ;
1642      ;INPUTS:
1643      ;     R4     POINTER TO COMMAND PACKET
1644      ;
1645      ;IMPLICIT INPUTS:
1646      ;
1647      ;     RAMDATA  DATA AS READ FROM THE RAM
1648      ;     RAMSIZ  NUMBER OF BYTES IN PACKET
1649      ;             IF RAMSIZ=0 THEN DEFAULT TO 8.
1650      ;
1651      ;IMPLICIT OUTPUTS:
1652      ;
1653      ;     RAMSIZ  SET TO 0

```

```

1654
1655
1656 014056
1657 014056
1658 014062 012701 002234
1659 014066 005002
1660 014070 122124
1661 014072 001005
1662 014074
1663 014104 000436
1664 014106 116105 177777
1665 014112 116403 177777
1666 014116
1667 014126 042703 177400
1668 014132 116137 177777 002226
1669 014140 116437 177777 002224
1670 014146
    014146 010346
    014150 013746 002224
    014154 013746 002226
    014160 010246
    014162 012746 014236
    014166 012746 000005
    014172 010600
    014174 104414
    014176 062706 000014
1671 014202 005202
1672 014204 005737 002274
1673 014210 001404
1674 014212 020237 002274
1675 014216 003724
1676 014220 000403
1677 014222 020227 000010
1678 014226 002720
1679 014230 005037 002274
1680 014234 000207
1681
1682 014236 045 116 045 RAMASC: .ASCIZ 'NNA BYTE: D2A RAM: 03A Packet: 03A XOR:03
1683 .EVEN
1684 .SBTTL PRMESS PRINT CONTENTS OF MESSAGE BUFFER
1685
1686
1687 ;
1688 ;THIS ROUTINE PRINTS THE CONTENTS OF
1689 ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE
1690 ;TSV-05.
1691 ;
1692 ;INPUT:
1693 ;
1694 ; RO LOW ORDER ADDRESS OF MESSAGE BUFFER
1695 ; R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER
1696 ; NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
1697 ;
1698 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
1699 ;
1700 ;
1701 PRMESS:

```

```

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

```

1740      ; EXPMSG - EXPECTED MESSAGE BUFFER
1741      ; RECMMSG - RECEIVED MESSAGE BUFFER
1742      ; RCVHIADD - RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1743      ; RCVLOADD - RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1744      ;
1745 014632 PRMSGEXP::
1746 014632 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1747 014636 010005 MOV R0,R5 ;SAVE NUMBER OF WORDS
1748 014640 013700 002300 MOV RCVLOADD,R0 ;GET RCV LOW ADDRESS
1749 014644 010004 MOV R0,R4 ;COPY LOW ADDRESS
1750 014646 013701 002276 MOV RCVHIADD,R1 ;GET RCV HIGH ADDRESS
1751 014652 006100 ROL R0 ;SHIFT BIT15 TO C BIT
1752 014654 006101 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
1753 014656 PRINTX #PRMSG0,R1,R4 ;PRINT MESSAGE BUFFER ADDRESS
      014656 010446 MOV R4,-(SP)
      014660 010146 MOV R1,(SP)
      014662 012746 015012 MOV #PRMSG0,-(SP)
      014666 012746 000003 MOV #3,-(SP)
      014672 010600 MOV SP,R0
      014674 104415 TRAP C:PNTX
      014676 062706 000010 ADD #10,SP
1754 014702 PRINTX #PRMSG1 ;PRINT HEADER FOR CONTENTS
      014702 012746 015057 MOV #PRMSG1,-(SP)
      014706 012746 000001 MOV #1,-(SP)
      014712 010600 MOV SP,R0
      014714 104415 TRAP C:PNTX
      014716 062706 000004 ADD #4,SP
1755 014722 005004 CLR R4 ;NUMBER OF THE CURRENT WORD
1756 014724 012701 002314 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
1757 014730 012702 002460 MOV #RECMMSG,R2 ;GET RCV BUFFER ADDRESS
20$: MOV (R1),R0 ;GET EXPD
      MOV (R2),R3 ;GET RCV
      XOR R0,R3 ;XOR EXPD/RCV
1761 014750 PRINTX #PRMSG2,R4,(R1),,(R2),,R3
      014750 010346 MOV R3,-(SP)
      014752 012246 MOV (R2),-(SP)
      014754 012146 MOV (R1),-(SP)
      014756 010446 MOV R4,-(SP)
      014760 012746 015115 MOV #PRMSG2,-(SP)
      014764 012746 000005 MOV #5,(SP)
      014770 010600 MOV SP,R0
      014772 104415 TRAP C:PNTX
      014774 062706 000014 ADD #14,SP
1762 015000 005204 INC R4 ;NUMBER OF THE NEXT
1763 015002 020405 CMP R4,R5 ;DONE ALL YET?
1764 015004 002001 BGE 50$ ;BR IF YES
1765 015006 000752 BR 20$ ;DO ANOTHER
1766 015010 000207 50$: RTS PC ;RETURN
1767
1768 015012 045 116 045 PRMSG0: .ASCIZ '##N##A Message Buffer Address = #01#05'
1769 015057 045 116 045 PRMSG1: .ASCIZ '##N##A Message Buffer Contents:
1770 015115 045 116 045 PRMSG2: .ASCIZ '##N##A WORD #D2##A EXPD: #06##A RCV: #06##A XOR: #06
1771 .EVEN
1772 .SBTTL PRBYTEXP PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
1773 ;*
1774 ;
1775 ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
    
```

```

1776 ; ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
1777 ;
1778 ; RO NUMBER OF BYTES IN BUFFER
1779 ;
1780 ;IMPLICIT INPUTS:
1781 ;
1782 ; EXPMSG - EXPECTED MESSAGE BUFFER
1783 ; RECMSG - RECEIVED MESSAGE BUFFER
1784 ;
1785 015202 PRBYTEXP::
1786 015202 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1787 015206 010005 MOV R0,R5 ;SAVE NUMBER OF BYTES
1788 015210 005037 002312 CLR PRMNO ;INIT ERROR COUNT
1789 015214 005004 CLR R4 ;NUMBER OF THE CURRENT BYTE
1790 015216 012701 002314 MOV @EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
1791 015222 012702 002460 MOV @RECMSG,R2 ;GET RECV BUFFER ADDRESS
1792 015226 111100 20$: MOVB (R1),R0 ;GET EXPD BYTE
1793 015230 042700 177400 BIC @C<377>,R0 ;CLEAR UPPER BYTE
1794 015234 110037 015550 MOVB R0,PRBEXP ;SAVE FOR ERROR REPORT
1795 015240 111203 MOVB (R2),R3 ;GET RECV BYTE
1796 015242 042703 177400 BIC @C<377>,R3 ;CLEAR UPPER BYTE
1797 015246 110337 015552 MOVB R3,PRBREC ;FOR ERROR REPORT
1798 015252 XOR R0,R3 ;XOR EXPD/RECV
1799 015262 122122 CMPB (R1)*,(R2)* ;EXPD = RECV?
1800 015264 001431 BEQ 30$ ;BR IF YES
1801 015266 005237 002312 INC PRMNO ;UPDATE ERROR COUNT
1802 015272 023727 002312 000010 CMP PRMNO,@8. ;PRINTED 8?
1803 015300 101023 BHI 30$ ;BR IF YES
1804 015302 27$: PRINTX @PRBMSG,R4,PRBEXP,PRBREC,R3
015302 010346 MOV R3,-(SP)
015304 013746 015552 MOV PRBREC,-(SP)
015310 013746 015550 MOV PRBEXP,-(SP)
015314 010446 MOV R4,-(SP)
015316 012746 015416 MOV @PRBMSG,-(SP)
015322 012746 000005 MOV @5,-(SP)
015326 010600 MOV SP,R0
015330 104415 TRAP C$PNTX
015332 062706 000014 ADD @14,SP
1805 015336 FORCEXIT 50$ ;@@D
1806 015346 000404 BR 35$ ;@D
1807 015350 30$:
1808 015350 FORCERROR 27$,NOTSSR ;@D
1809 015360 35$:
1810 015360 005204 INC R4 ;NUMBER OF THE NEXT
1811 015362 020405 CMP R4,R5 ;DONE ALL YET?
1812 015364 002001 BGE 50$ ;BR IF YES
1813 015366 000717 BR 20$ ;DO ANOTHER
1814 015370 50$: PRINTX @PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
015370 013746 002312 MOV PRMNO,(SP)
015374 012746 015503 MOV @PRBTOT,(SP)
015400 012746 000002 MOV @2,-(SP)
015404 010600 MOV SP,R0
015406 104415 TRAP C$PNTX
015410 062706 000006 ADD @6,SP
1815 015414 000207 RTS ;RETURN
1816
1817 015416 045 116 045 PRBMSG: .ASCIZ 'NNA BYTE 02NA EXPD: 03NA RECV: 03NA XOR: C3

```

TSV3 GLOBAL AREAS MACRO M1113 06 FEB 84 18:04
 PRBTEXP PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

SEQ 062

```

1818 015503    045    116    045 PRBTOT: .ASCIZ 'NUMBER OF BYTES IN ERROR = #D2'
1819                                     .EVEN
1820 015550 000000 PRBEXP: .WORD 0 ;EXPD
1821 015552 000000 PRBREC: .WORD 0 ;RECV
1822                                     .SBTTL EXPREC - PRINT EXPD/RECV WORD DATA
1823 ;*
1824 ;
1825 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1826 ;
1827 ;INPUTS:
1828 ;
1829 ;      R1      RECEIVED DATA
1830 ;      R2      EXPECTED DATA
1831 ;
1832 ;
1833 ;
1834 015554      BGNMSG EXPREC
1835 015554 004737 010020 EXPREC:: JSR PC,PRIXOR ;PRINT THE DATA
1836 015560      ENDMMSG
1837 015560 104423 L10017: TRAP C$MSG
1838                                     .SBTTL EXPBREC - PRINT EXPD/RECV BYTE DATA
1839 ;*
1840 ;
1841 ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
1842 ;
1843 ;INPUTS:
1844 ;
1845 ;      R1      RECEIVED DATA BYTE
1846 ;      R2      EXPECTED DATA BYTE
1847 ;
1848 ;
1849 ;
1850 015562      BGNMSG EXPBREC
1851 015562 004737 007670 EXPBREC:: JSR PC,PRIBXOR ;PRINT THE DATA
1852 015566      ENDMMSG
1853 015566 104423 L10020: TRAP C$MSG
1854 ;
1855 ;      .SBTTL RAMERR PRINT RAM AND PACKET DATA
1856 ;*
1857 ;
1858 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1859 ;
1860 ;INPUTS:
1861 ;
1862 ;      R4      POINTER TO COMMAND PACKET
1863 ;
1864 ;IMPLICIT INPUTS
1865 ;
1866 ;      RAMDATA DATA AS READ FROM THE RAM
1867 ;      RAMSIZ  NUMBER OF BYTES IN PACKET
1868 ;                  IF RAMSIZ=0 THEN DEFAULT TO 8.

```



```

1869
1870 ;IMPLICIT OUTPUTS:
1871 ;
1872 ; RAMSIZ SET TO 0
1873 ;
1874
1875 015570 BGNMSG RAMERR
015570 RAMERR::
1876 015570 004737 014056 JSR PC,PRAMPKT ;PRINT RAM/PACKET DATA
1877 015574 ENDMSG
015574 L10021:
015574 104423 TRAP C$MSG

1878 .SBTTL RAMTADD PRINT TEST ADDRESS, RAM AND PACKET DATA
1879 ;*
1880 ;
1881 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1882 ;
1883 ;INPUTS:
1884 ;
1885 ; R4 POINTER TO COMMAND PACKET
1886 ;
1887 ;IMPLICIT INPUTS:
1888 ;
1889 ; RAMDATA DATA AS READ FROM THE RAM
1890 ; RAMSIZ NUMBER OF BYTES IN PACKET
1891 ; IF RAMSIZ=0 THEN DEFAULT TO 8.
1892 ; ERRHI HIGH ORDER TEST ADDRESS
1893 ; ERRLO LOW ORDER TEST ADDRESS
1894 ;
1895 ;IMPLICIT OUTPUTS:
1896 ;
1897 ; RAMSIZ SET TO J
1898 ;
1899 ;
1900
1901 015576 BGNMSG RAMTADD
015576 RAMTADD::
1902 015576 004737 010352 JSR PC,PRITADD ;PRINT TEST ADDRESS
1903 015607 004737 014056 JSR PC,PRAMPKT ;PRINT RAM/PACKET DATA
1904 015606 ENDMSG
015606 L10022:
015606 104423 TRAP C$MSG

1905 .SBTTL RAMEXP PRINT RAM EXPD/RECV DATA
1906 ;*
1907 ;
1908 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1909 ;
1910 ;INPUTS:
1911 ;
1912 ; R1 RECEIVED DATA
1913 ; R2 EXPECTED DATA
1914 ; R4 CONTROLLER RAM ADDRESS
1915 ;
1916 ;
1917 ;
1918 015610 BGNMSG RAMEXP
015610 RAMEXP::

```

```

1919 015610 042701 177400      BIC    #C<377>,R1      ;SAVE EXPD RAM DATA BYTE
1920 015614 042702 177400      BIC    #C<377>,R2      ;SAVE EXPD RAM DATA BYTE
1921 015620 004737 010144      JSR    PC,PRIRAM      ;PRINT THE RAM ADDRESS
1922 015624 004737 010020      JSR    PC,PRIXOR      ;PRINT THE DATA
1923 015630      ENDMMSG
      015630      L10023:
      015630 104423      TRAP   C#MSG
1924
1925      .SBTTL  TIMEXP - PRINT TIMER A,B AND EXP/REC
1926
1927      ;*
1928      ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1929      ;AND TIMER A,B HEADER MESSAGE
1930
1931      ;INPUTS:
1932      ;
1933      ;      R1      RECEIVED DATA
1934      ;      R2      EXPECTED DATA
1935      ;
1936
1937 015632      BGNMSG  TIMEXP
      015632      TIMEXP::
1938 015632      PRINTX  #TIMSGO      ;PRINT HEADER
      015632 012746 015660      MOV    #TIMSGO,-(SP)
      015636 012746 000001      MOV    #1,(SP)
      015642 010600      MOV    SP,R0
      015644 104415      TRAP   C#PNTX
      015646 062706 000004      ADD    #4,SP
1939 015652 004737 010020      JSR    PC,PRIXOR      ;PRINT THE DATA
1940 015656      ENDMMSG
      015656      L10024:
      015656 104423      TRAP   C#MSG
1941
1942 015660      045      116      045  TIMSGO: .ASCIZ  'N/A TIMER A STATUS IS IN BIT 3N/A TIMER B STATUS IS IN BIT 2
1943      .EVEN
1944      .SBTTL  BADSSR PRINT TSSR ERRORS ON DATA TRANSFERS
1945
1946      ;*
1947      ;
1948      ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
1949      ;
1950      ;INPUTS:
1951      ;
1952      ;      R1      CONTENTS OF TSSR
1953      ;      R2      DATA WRITTEN (8 BITS)
1954      ;
1955      ;
1956
1957 015760      BGNMSG  BADSSR
      015760      BADSSR::
1958 015760 010246      MOV    R2,-(SP)      ;SAVE DATA TRANSFERRED
1959 015762 042702 177400  BIC    #177400,R2      ;GET JUST ONE BYTE
1960 015766      PRINTB  #XFERASC,R2
      015766 010246      MOV    R2,-(SP)
      015770 012746 016020      MOV    #XFERASC,-(SP)
      015774 012746 000002      MOV    #2,-(SP)
      016000 010600      MOV    SP,R0

```

1961	016010	012602				TRAP	C#PNTB		
1962	016012	004737	06020			ADD	#6 SP		
1963	016016					MOV	(SP),R2		;RESTORE R2
	016016					JSR	PC,PRITSSR		;DECODE TSSR CONTENTS
	016016					ENDMSG			
1964	016020	104423	045	116	045	TRM	C#MSG		
1965						.ASCIZ	'#N#A Data Transferred - #03'		
1966						.SBTTL	GLOBAL SUBROUTINE SECTION		
1967									
1968									
1969									
1970									
1971									
1972									
1973									
1974									
1975									
1976									
1977									
1978									
1979									
1980									
1981									
1982									
1983									
1984									
1985									
1986									
1987									
1988									
1989									
1990									
1991									
1992									
1993									
1994									
1995									
1996									
1997									
1998									
1999	016054								
2000	016054								
2001	016060	012765	000000	000002					
2002	016066	004737	016330						
2003	016072	016500	000002						
2004	016076	010004							
2005	016100	042704	76277						
2006	016104	052704	002200						
2007	016110	020400							
2008	016112	001402							
2009	016114	000241							
2010	016116	000401							
2011	016120	000261							
2012	016122	000207							
2013									

```

L10025:
;***
; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
; THAT ARE USED IN MORE THAN ONE TEST.
;---
.SBTTL SOFINIT - SO T INITIALIZE OF CONTROLLER
; *
;
; ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
; BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
; THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
; DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
;
; INPUTS:
;
; R5 ADDRESS OF FIRST REGISTER
;
; OUTPUTS
;
; R0 CONTENTS OF TSSR, IF ERROR
; CARRY SET IF INIT WAS OKAY
; CLEAR IF FATAL ERROR
;
; CALLING SEQUENCE:
;
; MOV #ADDRESS,R5
; JSR PC,SOFINIT
; BCS CONTINUE
; ERROF ;REPORT FATAL ERROR
;
;
SOFINIT::
SAVREG ; SAVE THE REGISTERS
MOV #0,TSSR(R5) ; DO THE INIT.
JSR PC,WAITF ; WAIT FOR SSR
MOV TSSR(R5),R0 ;GET THE TSSR REGISTER
MOV R0,R4 ;TSSR CONTENTS
BIC #C<MIADDR!OFL>,R4
BIS #SSR!NBA,R4 ;R4 HAS EXPECTED CONTENTS
CMP R4,R0 ;ONLY EXPECTED BITS SET ?
BEQ 5$ ;BRANCH IF OKAY
CLC ;CLEAR THE CARRY FOR ERROR
BR 10$ ;GO TO EXIT
5$: SEC ;SET THE CARRY BIT
10$: RTS PC ;RETURN TO CALLER
.SBTTL CHKAMB CHECK TSSR FOR AMBIGUITY

```

```

2014
2015
2016
2017 ; THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
2018 ; FOR AMBIGUITY
2019
2020 ; INPUT:
2021
2022 ;         RO         CONTENTS OF TSSR
2023
2024 ; OUTPUT:
2025
2026 ;         RO         CONTENTS OF TSSR
2027
2028 ;         CARRY      SET - NO AMBIGUITY
2029 ;                   CLR - AMBIGUOUS CONTENTS
2030
2031
2032
2033 CHKAMB:
2034         SAVREG                ; SAVE THE GENERAL REGISTERS
2035         MOV      RO,R4         ; CONTENTS OF TSSR
2036         BIT     @SC,RO        ; IS BIT 15 SET ?
2037         BNE     50           ; BRANCH IF YES
2038         BIT     @C<NBA!OFL!SSR!MIADDR>,RO ; ANY OTHER BITS SET ?
2039         BNE     40           ; MUST BE AN ERROR
2040         BR      45           ; RETURN WITH SUCCESS
2041         BIT     @SSR,RO       ; IS READY BIT SET ?
2042         BNE     10           ; BRANCH IF READY BIT IS SET.
2043         BIT     @BIT5,RO     ; IS FATAL ERROR BIT SET ?
2044         BEQ     40           ; ERROR IF NOT
2045         BIC     @CTERCLS,R4   ; CLEAR ALL BUT TERMINATION CODE
2046         CMP     R4,#16       ; ALL THREE BITS MUST BE SET
2047         BNE     40           ; ERROR IF NOT SET
2048         BR      45           ; OK I. ALL ARE SET
2049         BIT     @BIT5,RO     ; IS FATAL ERROR BIT SET ?
2050         BEQ     45           ; ERROR IF BIT IS SET WITH SSR
2051         BIT     @BIT2!BIT1,RO ; IS THIS A FUNCTION REJECT
2052         BNE     45           ; BR. IF TSSR IS OK
2053         CLC                    ; AMBIGUOUS CONTENTS
2054         BR      50
2055         SEC                    ; SHOW SUCCESS - NO AMBIGUITY
2056         RTS     PC           ; RETURN TO CALLER
2057         .SBTTL ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
2058
2059 ; DEFAULT DISPLAY INTERRUPT HANDLERS.
2060 ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2061 ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
2062
2063
2064 ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2065
2066 ;         IOKCKIN=BIT7 ; DON'T CHECK FOR BAD INTERRUPTS - TEST WILL.
2067 ;         IOKSTP=BIT0 ; EXPECT "STOP" INTERRUPT.
2068
2069 ; INTERRUPT MASK - SAYS EXPECTING INTERRUPTS
2070 INTMASK: .BYTE 0

```

```

2071 ; INTERRUPT FLAG - SAYS WE GOT ONE (IF POSITIVE)
2072 016225 000 INTFLAG: .BYTE 0
2073
2074 ; SAVED INTEPRUPT VECTOR:
2075 016226 000000 INTVEC: .WORD 0
2076 ; SAVE CPU PC
2077 016230 000000 INTCPC: .WORD 0
2078
2079 ; SUBROUTINE TO ENABLE INTERRUPTS:
2080 016232 010040 ENAINT: MOV RO,-(SP) ;SAVE RO
2081 016234 0137C0 002202 MOV IVEC,RO ;GET POINTER TO VECTORS
2082 016240 012720 016276 MOV @INTR,(RO). ;SET UP INTERRUPT VECTOR
2083 016244 012720 000340 MOV @PRI07,(RO).
2084 016250 012600 MOV (SP),RO ;RESTORE RO
2085 016252 011646 MOV (SP),-(SP)
2086 016254 012766 000000 000002 MOV @0,2(SP) ;SET CPU TO LEVEL 0
2087 016262 000002 RTI
2088
2089 ; SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2090 016264 011646 DSBINT: MOV (SP),-(SP)
2091 016266 012766 000340 000002 MOV @PRI07,2(SP)
2092 016274 000002 RTI
2093 .SBTTL INTR INTERRUPT HANDLERS
2094
2095 016276 BGNSRV INTR ;DEFINE INTERRUPT ENTRY
016276 INTR::
2096 016276 012737 000001 002216 MOV @1,INTRECV ;SET FLAG TO SHOW INTERRUPT RECEIVED
2097 016304 105037 016225 CLRB INTFLAG ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2098 016310 132737 000001 016224 BITB @IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2099 016316 001003 BNE 1$ ;BR IF YES
2100 016320 152737 000001 016225 BISB @IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2101
2102 ;SAVE REGISTERS, MSG BUFFER, ETC.
2103 016326 1$: ENDSRV
2104 016326 L10026:
016326 RTI
016326 .SBTTL WAITF WAIT FOR SUBSYSTEM READY
2105
2106 ;
2107 ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2108 ;
2109 ; INPUTS:
2110 ;
2111 ; R5 ADDRESS OF FIRST DEVICE REGISTER
2112 ;
2113 ; OUTPUTS:
2114 ;
2115 ; RO CONTENTS OF LAST TSSR READ
2116 ; CARRY SET READY BIT SET
2117 ; CLR TIMEOUT WAITING FOR READY
2118 ;
2119 016330 000401 WAITF:: BR 1$ ;NOP WHEN SUPER FIXED
2120 016332 BREAK ; DO A SUPVSR BREAK FIRST.
016332 104422 TRAP C#BRK
2121 016334 012746 011000 1$: MOV @11000,(SP) ;25 APRIL-83 REV B - 1100 MSEC TIMER
2122 016340 016500 000002 2$: MOV TSSR(R5),RO ;READ THE TSSR REGISTER
2123 016344 105700 TSTB RO ;TEST FOR READY BIT SET

```

```

2124
2125 016346 100420          BMI    3$          ; EXIT ON STOP FLAG.
2126 016350          DELAY  1          ; WAIT 100 USEC
      016350 012727 000001    MOV    #1,(PC).
      016354 000000          .WORD  0
      016356 013727 002116    MOV    L#DLY,(PC).
      016362 000000          .WORD  0
      016364 005367 177772    DEC    -6(PC)
      016370 001375          BNE    -4
      016372 005367 177756    DEC    22(PC)
      016376 001367          BNE    -20
2127 016400 005316          DEC    (SP)          ;REDUCE DELAY COUNT
2128 016402 001356          BNE    2$          ;RETRY UNTIL TIMER EXPIRES
2129 016404 000241          CLC
2130 016406 000401          BR     4$          ; C = 0, CONTROLLER STILL RUNNING...
2131 016410 000261          3$: SEC          ;...OR HUNG-UP AFTER 300 MSEC.
2132 016412 005326          4$: DEC    (SP).    ; C = 1, CONTROLLER IS STOPPED.
2133 016414 000207          RTS    PC          ;RESTORE STACK WITHOUT CHANGING CARRY BIT
2134          .SBTTL  CHKTISSR  CHECK TSSR FOR READY
2135
2136          ;*
2137          ;
2138          ;THIS ROUTINE WAITS FOR READY IN THE TSSR
2139          ;AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
2140          ;
2141          ;INPUT:
2142          ;
2143          ;    R5    ADDRESS OF CSR REGISTERS
2144          ;
2145          ;OUTPUT:
2146          ;
2147          ;    R0    CONTENTS OF TSSR
2148          ;    CARRY SET - OKAY
2149          ;          CLR - NOT READY AMBIGUOUS, OR SC SET
2150          ;
2151          ;-
2152
2153 016416          CHKTISSR:
2154 016416 004737 016330    JSR    PC,WAITF    ;WAIT FOR READY
2155 016422 103014          BCC    20$          ;BRANCH IF TIME OUT
2156 016424 004737 016124    JSR    PC,CHKAMB   ;TSSR AMBIGUOUS?
2157 016430 103006          BCC    10$          ;BR IF YES
2158 016432 032700 100000    BIT    #SC,R0      ;SPECIAL CONDITION SET?
2159 016436 001405          BEQ    15$          ;BR IF NO
2160 016440 032700 074000    BIT    #<SCE!BIE!RMR!NXM>,R0 ;ANY ERROR BITS SET?
2161 016444 001402          BEQ    15$          ;BR IF NO
2162 016446 000241          10$: CLC          ;SET FAILURE
2163 016450 000401          BR     20$          ;
2164 016452 000261          15$: SEC          ;SET SUCCESS
2165 016454 000207          20$: RTS    PC     ;RETURN TO CALLER
2166          .SBTTL  XNXM    CHECK FOR NONEXISTENT MEMORY
2167
2168          ;*
2169          ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
2170          ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
2171          ;          "C" = 0, ALL ADDRESSES OK.
2172          ;
          ;CALL:  MOV  ADR1,R1

```

```

2173      ;      MOV ADR2,R2
2174      ;      JSR PC,NXM
2175      ;      RETURN          ;TEST "C" AND PROCEED.
2176      ;
2177 016456 012737 016510 000004 XNXM: MOV    021,004      ; SET BUSERR VECTOR.
2178 016464 012737 000200 000006 MOV    0PRI04,006
2179 016472 005003          CLR    R3          ;FLAG.
2180 016474 005711 1$:   TST    (R1)      ;TEST THE ADDRESS(ES).
2181          ;      ;IF ANY TRAP, CONTINUE AT 2$.
2182 016476 020102          CMP    R1,R2      ;OTHERWISE, CONTINUE HERE.
2183 016500 001407          BEQ    3$          ;BR IF FINISHED (NO NEXM'S).
2184 016502 062701 000002 ADD    02,R1      ;SET NEXT ADDRESS...
2185 016506 000772          BR    1$          ;...AND CONTINUE.
2186      ;
2187 016510 005103 2$:   COM    R3          ;GOT ONE, SET FLAG...
2188 016512 012716 016520 MOV    031,(SP)
2189 016516 000002          RTI          ;...AND DISMISS INTERRUPT...
2190 016520 3$:   CLRVEC  04          ;...AND GIVE BACK THE VECTOR.
      016520 012700 000004 MOV    04,R0
      016524 104436 TRAP   C1CVEC
2191 016526 005703          TST    R3          ;DID WE CATCH ONE ??
2192 016530 001401          BEQ    .+4         ;NO, "C" = 0, SKIP NEXT.
2193 016532 000261          SEC          ;YES, "C" = 1, (R1) = NEXM ADDR.
2194 016534 000207          RTS    PC
2195      ;
2196      ;
2197      ;      .SBTTL TSTLOOP - CHECK ITERATION COUNT
2198      ;*
2199      ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
2200      ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON ZERO.
2201      ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
2202      ;
2203      ; CALL: LOOPTO ARG
2204      ;
2205 016536      ;TSTLOOP:;
2206 016536 005737 002162 TST    NOITS      ; ITERATIONS INHIBITED?
2207 016542 001006          BNE    1$          ; YES.
2208 016544 005737 002176 TST    QVP        ; NO.
2209 016550 100403          BMI    1$          ;LOOPS DISALLOWED IN QUICK PASS.
2210 016552 005337 002210 DEC    LOOPCNT    ; BUMP LOOP COUNTER.
2211 016556 001002          BNE    2$          ;
2212 016560 000241 1$:   CLC          ;LOOP DISALLOWED, OR DONE.
2213 016562 000401          BR    3$          ;
2214 016564 000261 2$:   SEC          ;LOOP ENABLED.
2215 016566 000207 3$:   RTS    PC
2216      ;
2217      ;      .SBTTL TSTSETUP PRINT TEST NAME AND INIT ERROR COUNTS
2218      ;*
2219      ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
2220      ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
2221      ; IN THE CURRENT RUN SEQUENCE.
2222      ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
2223      ;
2224      ;INPUT:
2225      ;
2226      ;      R0      POINTER TO TEST ID ASCIZ STRING
2227      ;

```

```

2228 ;OUTPUT:
2229 ;
2230 ; R5 ADDRESS OF FIRST DEVICE REGISTER
2231 ;
2232 ;IMPLICIT OUTPUTS:
2233 ;
2234 ; TSTCNT UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
2235 ;
2236 ;SIDE EFFECTS:
2237 ;
2238 ; INTERRUPT LEVEL IS RASIED TO LEVEL OF
2239 ; THE DEVICE UNDER TEST
2240 ;
2241 ;
2242 ;
2243 TSTSETUP::
2244 016570 010046 MOV RO, -(SP) ;SAVE THE TEST ID MESSAGE
2245 016572 005037 003146 CLR SIFLAG ; CLEAR "SOFT INIT" FLAG
2246 016576 005037 017036 CLR ERRK ; CLEAR LOCAL ERROR COUNTER.
2247 016602 005037 005766 CLR EXTA ; CLEAR ERROR EXTENSION FLAG.
2248 016606 105037 016224 CLRB INTMASK ; CLEAR INTERRUPT MASK (CHECK ERROR)
2249 016612 013700 002174 MOV UNITN,RO ; GET THE UNIT NUMBER,
2250 016616 006300 ASL RO ; ... AND MAKE IT A WORD OFFSET.
2251 016620 005737 003106 TST NODEV ; DID STARTUP FIND THE DEVICE?
2252 016624 001430 BEQ 4$ ; BR IF YES
2253 016626 100010 BPL 3$ ; BR IF NOT IDLE
2254 016630 052760 160000 003170 BIS #160000,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
2255 016636 ERRDF 1,NXR,NXRERR ; NO DEVICE HERE - PRINT IT
016636 104455 TRAP C$ERRDF
016640 000001 .WORD 1
016642 003734 .WORD NXR
016644 005732 .WORD NXRERR
2256 016646 000407 BR 2$
2257 016650 052760 160001 003170 3$: BIS #160001,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
2258 016656 ERRDF 2,NOINIT ; DEVICE NOT IDLE
016656 104455 TRAP C$ERRDF
016660 000002 .WORD 2
016662 004331 .WORD NOINIT
016664 000000 .WORD 0
2259 016666 012737 177777 003104 2$: MOV #-1,DUFLG ; DROP THE UNIT
2260 016674 DODU UNITN
016674 013700 002174 MOV UNITN,RO
016700 104451 TRAP C$DODU
2261 016702 DOCLN ; ABORT THE PASS
016702 104444 TRAP C$DOCLN
2262 016704 000423 BR 5$
2263
2264 016706 4$: RFLAGS RO ; GET THE OPERATOR FLAGS.
016706 104421 TRAP C$RFLA
2265 016710 032700 001000 BIT #PNT,RO ; PRINT THE TEST NUMBERS?
2266 016714 001412 BEQ 1$ ; BR IF NO
2267 016716 011600 MOV (SP),RO ;GET THE ID MESSAGE
2268 016720 PRINTF #TNAM,RO ;DISPLAY THE TEST ID
016720 010046 MOV RO, -(SP)
016722 012746 016764 MOV #TNAM, -(SP)
016726 012746 000002 MOV #2, -(SP)
016732 010600 MOV SP,RO

```



```

016734 104417          TRAP  C:PNTF
016736 062706 000006   ADD   #6,SP
2269 016742 005237 002206   1$:  INC   TSTCNT          ; BUMP TEST COUNTER.
2270 016746          SETPRI IPRI             ; PRIORITY THAT OF DEVICE
      016746 013700 002204   MOV   IPRI,RO
      016752 104441          TRAP  C:SPRI
2271 016754 005726          TST   (SP),          ; FIX UP THE STACK
2272 016756 013705 002200   MOV   CSRADDR,R5     ; ADDRESS OF TSV REGISTERS ON UNIBUS
2273 016762 000207          RTS   PC
2274 016764 045 123 045  TNAM:  .ASCIZ  'SATA Test'
2275          .EVEN
2276          .SBTTL  TSTEND - PRINT ERRORS RECEIVED
2277
2278          ;
2279          ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2280          ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2281          ;
2281 017000          TSTEND: RFLAGS RO
      017000 104421          TRAP  C:RFLA
2282 017002 030027 020000   BIT   RO,#IER
2283 017006 001412          BEQ   1$             ; BR IF "IER" NOT SET.
2284 017010          PRINTF #ESUM,ERRK          ; PRINT ERROR COUNT.
      017010 013746 017036   MOV   ERRK,(SP)
      017014 012746 017040   MOV   #ESUM,-(SP)
      017020 012746 000002   MOV   #2,-(SP)
      017024 010600          MOV   SP,RO
      017026 104417          TRAP  C:PNTF
      017030 062706 000006   ADD   #6,SP
2285 017034 000207          1$:  RTS   PC
2286
2287 017036 000000          ERRK:  0             ; LOCAL ERROR COUNT.
2288 017040 045 101 040  ESUM:  .ASCIZ  /#A #D#A ERRORS/
2289 017057 105 122 122  EMAXDU: .ASCIZ  /ERROR LIMIT REACHED DROPPING UNIT/
2290          .EVEN
2291
2292          .SBTTL  INCERK  INCREMENT LOCAL ERROR COUNT
2293
2294          ;*
2295          ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
2296          ;
2296 017124 005237 017036   INCERK: INC  ERRK          ; INCREMENT LOCAL ERROR COUNT
2297 017130 010046          MOV   RO,-(SP)       ; SAVE RO
2298 017132 013700 002174   MOV   UNITN,RO      ; GET UNIT NUMBER.
2299 017136 006300          ASL   RO             ; ... AND MAKE IT A WORD OFFSET.
2300 017140 062700 003170   ADD   #ERTABL,RO    ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
2301 017144 005210          INC   (RO)           ; INCREMENT THE DEVICE ERROR COUNT
2302 017146 032710 007777   BIT   #7777,(RO)    ; DID WE OVERFLOW THE FIELD?
2303 017152 001001          BNE   1$             ; BR IF NO.
2304 017154 005310          DEC   (RO)           ; YES -- BACK IT UP TO 7777.
2305 017156 012600          1$:  MOV   (SP),RO      ; RESTORE RO
2306 017160 000207          RTS   PC             ; RETURN TO CALLER.
2307
2308 017162 010046          CKEMAX: MOV  RO,(SP)     ; SAVE RO
2309 017164 013700 002174   MOV   UNITN,RO      ; GET UNIT NUMBER
2310 017170 006300          ASL   RO             ; ... AND MAKE IT A WORD OFFSET
2311 017172 016000 003170   MOV   ERTABL(RO),RO ; GET ERROR TABLE ENTRY
2312 017176 042700 170000   BIC   #170000,RO    ; EXTRACT ERROR COUNT FIELD
2313 017202 020037 002166   CMP   RO,GERRMAX    ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
2314 017206 103004          BHIS  1$             ; BR IF YES

```

```

2315 017210 023737 017036 002164      CMP      ERRK,LERRMAX      ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2316 017216 103417                    BLN      2$                ; BR IF NO
2317 017220                    1$:  RFLAGS  RO                ; GET OPERATOR FLAGS
      017220 104421                    TRAP    C$RFLA
2318 017222 032700 000040              BIT      @IDU,RO          ; IS DROPPING INHIBITED?
2319 017226 001013                    BNE     1$                ; BR IF YES.
2320 017230 012737 177777 003104      MOV     @-1,DUFLG        ; NO -- DROP THE UNIT
2321 017236                    ERRDF   4,EMAXDU
      017236 104455                    TRAP    C$ERDF
      017240 000004                    .WORD  4
      017242 017057                    .WORD  EMAXDU
      017244 000000                    .WORD  0
2322 017246                    DODU    UNITN
      017246 013700 002174              MOV     UNITN,RO
      017252 104451                    TRAP    C$DODU
2323 017254                    DOCLN
      017254 104444                    TRAP    C$DCLN
2324 017256 012600                    2$:  MOV     (SP)+,RO      ; RESTORE RO
2325 017260 000207                    RTS     PC                ; RETURN TO CALLER
2326                    .SBTTL  CKDROP - CHECK IF UNIT SHOULD BE DROPPED
2327                    ;*
2328                    ; CHECK IF UNIT SHOULD BE DROPPED
2329                    ;-
2330 017262 010046                    CKDROP: MOV    RO,-(SP)
2331 017264                    FORCERROR 1$,NOTSSR
2332 017274                    RFLAGS  RO
      017274 104421                    TRAP    C$RFLA
2333 017276 032700 000040              BIT      @IDU,RO
2334 017302 001010                    BNE     1$
2335 017304 011600                    MOV     (SP),RO
2336 017306 012737 177777 003104      MOV     @-1,DUFLG
2337 017314                    DODU    UNITN
      017314 013700 002174              MOV     UNITN,RO
      017320 104451                    TRAP    C$DODU
2338 017322                    DOCLN
      017322 104444                    TRAP    C$DCLN                ;ABORT THE PASS
2339 017324 012600                    1$:  MOV     (SP)+,RO
2340 017326 000207                    RTS     PC
2341
2342
2343                    .SBTTL  CONFIG  DETERMINE CONFIGURATION OF SYSTEM
2344                    ;
2345                    ; SUBROUTINE  DETERMINE CONFIGURATION OF TSV05 SYSTEM.
2346                    ;
2347 017330                    CONFIG:
2348 017330 004737 016054              JSR     PC,SOFINIT
2349 017334 000207                    RTS     PC
2350                    .SBTTL  KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
2351                    ;
2352                    ; SUBROUTINE  ENABLE MEM MGT.
2353                    ;
2354 017336 005737 003124              KTON:  TST     KFLG        ; GOT KT?
2355 017342 001403                    BEQ     1$                ; NO.
2356 017344 012737 000001 177572      MOV     @1,SRO          ; YES. ENABLE KT11.
2357 017352 000207                    1$:  RTS     PC
2358
2359                    ;

```

```

2360 ; SUBROUTINE DISABLE MEM MGT.
2361 ;
2362 017354 005737 003124 ; KTOFF: TST KTF LG ; GOT KT11?
2363 017360 001405 ; BEQ 18 ; NO.
2364 017362 00024C ; NOP
2365 017364 000240 ; NOP
2366 017366 012737 000000 177572 ; MOV #0,SRO ; DISABLE KT.
2367 017374 000207 18: ; RTS PC
2368 ; .SBTTL SETMAP SETUP PAR6 MAPPING
2369 ;
2370 ;
2371 ;
2372 ; THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
2373 ; AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
2374 ; IS RETURNED BIASED TO PAR6.
2375 ;
2376 ; INPUTS:
2377 ;
2378 ; R0 HIGH ORDER ADDRESS BITS
2379 ; R1 LOW ORDER ADDRESS BITS
2380 ;
2381 ; OUTPUTS:
2382 ;
2383 ; R0 OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
2384 ; CARRY SET IF SUCCESS
2385 ; CLR IF ERROR
2386 ;
2387 ; SETMAP:
2388 017376 ; SAVREG ; SAVE R1 R4 UNTIL NEXT RETURN
2389 017402 005737 003124 ; TST KTF LG ; SYSTEM HAVE ABOVE 28K?
2390 017406 001433 ; BEQ 108 ; BR IF NO
2391 017410 010102 ; MOV R1,R2 ; SAVE LOW ORDER BITS
2392 ; .REPT 6
2393 ; ASR R0 ; CONVERT WORD ADDRESS TO 32W BLOCKS
2394 ; ROR R1 ; MAKE IT DOUBLE PRECISION
2395 ; .ENDR
2396 017442 042701 000177 ; BIC #177,R1 ; ALINE FOR LOWER 4K BOUNDARY
2397 017446 020137 003124 ; CMP R1,KTF LG ; HIGHER THAN EXISTING MEMORY?
2398 017452 103011 ; BMIS 108 ; BR IF YES
2399 017454 010137 172354 ; MOV R1,#*IPAR6 ; SETUP MAPPING REGISTER PAR6
2400 017460 042702 160000 ; BIC #160000,R2 ; SETUP DISPLACEMENT IN PAGE
2401 017464 062702 140000 ; ADD #140000,R2 ; ADD IN PAR6 BIAS
2402 017470 010200 ; MOV R2,R0 ; RETURN IN R0
2403 017472 000261 ; SEC ; SET SUCCESS
2404 017474 000401 ; BR 158 ;
2405 017476 000241 108: ; CLC ; SET FAILURE
2406 017500 000207 158: ; RTS PC ; RETURN
2407 ; .SBTTL FILLMEM FILL MEMORY WITH BACKGROUND PATTERN
2408 ;
2409 ; FILL MEMORY WITH A BACKGROUND PATTERN
2410 ;
2411 ; INPUTS:
2412 ;
2413 ; R0 = BACKGROUND PATTERN
2414 ; FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2415 ; KTF LG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2416 ;

```

```

2417      ; OUTPUTS:
2418      ;
2419      ;     NONE
2420      ;
2421      ;
2422      ; FILLMEM:
2423      ; SAVREG      ;SAVE R1 R5 UNTIL NEXT RETURN
2424      ; JSR PC,KTOFF ;DISABLE KT.
2425      ; MOV R0,R3    ;COPY TEST PATTERN
2426      ; MOV FREE,R1  ;GET FIRST FREE LOCAT.
2427      ; MOV FRESIZ,R2 ;SIZE OF FREE SPACE BELOW 28K.
2428      ; MOV R3,(R1).  ;STORE A BACKGROUND WORD
2429      ; DEC R2        ;DONE ALL MEMORY IN FREE SPACE?
2430      ; BGT 10$     ;BR IF NO
2431      ; TST KIFLG    ; GOT KT?
2432      ; BEQ 55$     ; NO. GET OUT.
2433      ; JSR PC,KTON  ; YES. ENABLE KT.
2434      ; CLR R0      ;HIGH ORDER ADDRESS START
2435      ; MOV PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
2436      ; .REPT 6
2437      ; CLC          ;CLEAR C BIT
2438      ; ROL R1       ;CONVERT BLOCKS TO WORDS
2439      ; ROL R0       ;MAKE IT DOUBLE PRECISION
2440      ; .ENDR
2441      ; JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
2442      ; MCV R3,(R0).  ;STORE TEST PATTERN IN >28K ADDRESS
2443      ; CMP R0,#160000 ;END OF PAR6 MAPPING AREA?
2444      ; BLO 30$     ;BR IF NO
2445      ; SUB #20000,R0 ;BACKUP INTO PAR6 MAPPING BEGIN
2446      ; ADD #200,#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2447      ; CMP #KIPAR6,KIFLG ;END OF MEMORY?
2448      ; BEQ 50$     ;BR IF YES
2449      ; TST T23A    ;11/23A?
2450      ; BEQ 35$     ;NO KEEP GOING
2451      ; MOV SRO,R4   ;GET SRO CONTENTS
2452      ; BIC #177761,R4 ;CLEAR ALL BUT PAGE NUMBER
2453      ; CMP #16,R4  ;SEE IF PAGE 7
2454      ; BEQ 50$     ;EXIT IF THERE
2455      ; TST T23B    ;11/23B?
2456      ; BEQ 45$     ;NO KEEP GOING
2457      ; CMP #KIPAR6,#7600 ;REACHED 18 BITS?
2458      ; BHIS 40$    ;YES
2459      ; BR 45$      ;NO KEEP GOING
2460      ; MOV #20,SRO  ;SET 22 BIT RELOCATION
2461      ; JMP 30$     ;KEEP GOING ON ETC.
2462      ; JSR PC,KTOFF ; DISABLE KT.
2463      ; RTS PC
2464      ; .SBTTL CMPMEM COMPARE MEMORY TO BACKGROUND PATTERN
2465      ;
2466      ; COMPARE MEMORY WITH A BACKGROUND PATTERN
2467      ;
2468      ; INPUTS:
2469      ;
2470      ; RO = BACKGROUND PATTERN
2471      ; FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2472      ; KIFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2473      ;
  
```

```

2474      ; OUTPUTS:
2475      ;
2476      ;       CARRY  - SET IF NO ERROR
2477      ;       CARRY  - CLR IF ERROR
2478      ;
2479      ; IMPLICIT OUTPUTS:
2480      ;
2481      ;       ERRHI   ERROR HIGH ADDRESS
2482      ;       ERRLO  - ERROR LOW ADDRESS
2483      ;       EXPD   - EXPECTED DATA
2484      ;       RECV   RECEIVED DATA
2485      ;
2486 017740  ; CMPMEM:
2487 017740      SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
2488 017744 010003  MOV      R0,R3      ;COPY TEST PATTERN
2489 017746 004737 017354  JSR      PC,KTOFF      ;DISABLE KT.
2490 017752 013701 003116  MOV      FREE,R1      ;GET FIRST FREE LOCATION
2491 017756 013702 003120  MOV      FRESIZ,R2    ;SIZE OF FREE SPACE BELOW 28K.
2492 017762 020311      10$:  CMP      R3,(R1)    ;FREE SPACE LOCATION EQUAL TO EXPD?
2493 017764 001411      BEQ      15$      ;BR IF YES
2494 017766 010137 002232  MOV      R1,ERRLO    ;SAVE ADDRESS IN ERROR
2495 017772 005037 002230  CLR      ERRHI      ;NO HIGH ADDRESS
2496 017776 010337 002224  MOV      R3,EXPD    ;SAVE EXPD FOR ERROR REPORT
2497 020002 011137 002226  MOV      (R1),RECV  ;SAVE RECV FOR ERROR REPORT
2498 020006 000474      BR      50$      ;
2499 020010 005721      15$:  TST      (R1)    ;POINT TO NEXT ADDRESS
2500 020012 005302      DEC      R2      ;DONE ALL MEMORY IN FREE SPACE?
2501 020014 003362      BGT      10$      ;BR IF NO
2502 020016 005737 003124  TST      KTFLG      ; GOT KT?
2503 020022 001472      BEQ      55$      ; NO. GET OUT.
2504 020024 004737 017336  JSR      PC,KTON    ; YES. ENABLE KT.
2505 020030 005000      CLR      R0      ;HIGH ORDER ADDRESS START
2506 020032 013701 003144  MOV      PST32W,R1  ;GET >28K START ADDRESS (IN 32W BLOCKS)
2507      000006      .REPT      6
2508      ROL      R1      ;CONVERT BLOCKS TO WORDS
2509      ROL      R0      ;MAKE IT DOUBLE PRECISION
2510      .ENDR
2511 020066 042701 000177  BIC      @177,R1    ;ALINE 4K BOUNDARY
2512 020072 010046      MOV      R0,(SP)    ;SAVE HIGH ORDER
2513 020074 010146      MOV      R1,(SP)    ;SAVE LOW ORDER
2514 020076 004737 017376  JSR      PC,SETMAP  ;SETUP PAR6 MAPPING REGISTER
2515 020102 010004      MOV      R0,R4      ;COPY ADDRESS BIASED TO PAR6
2516 020104 012601      MOV      (SP),R1    ;RESTORE LOW ORDER IN NON PAR6 FORMAT
2517 020106 012600      MOV      (SP),R0    ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
2518 020110 020314      30$:  CMP      R3,(R4)    ;ABOVE 28K LOCATION EQUAL EXPD?
2519 020112 001411      BEQ      32$      ;BR IF YES
2520 020114 010037 002230  MOV      R0,ERRHI  ;SAVE HIGH ORDER IN ERROR
2521 020120 010137 002232  MOV      R1,ERRLO  ;SAVE LOW ORDER IN ERROR
2522 020124 010337 002224  MOV      R3,EXPD   ;SAVE EXPD FOR ERROR REPORT
2523 020130 011437 002226  MOV      (R4),RECV ;SAVE RECV FOR ERROR REPORT
2524 020134 000421      BR      50$      ;
2525 020136 062701 000002  32$:  ADD      @2,R1      ;UPDATE NON PAR6 ADDRESS
2526 020142 005500      ADC      R0      ;MAKE IT DOUBLE PRECISION ADD
2527 020144 062704 000002  ADD      @2,R4      ;UPDATE PAR6 MAPPING ADDRESS
2528 020150 020427 160000  CMP      R4,@160000 ;END OF PAR6 MAPPING AREA?
2529 020154 103755      BLO      30$      ;BR IF NO
2530 020156 162704 020000  SUB      @20000,R4  ;BACKUP INTO PAR6 MAPPING BEGIN

```

```

2531 020162 062737 000200 172354      ADD    #200,0<IPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2532 020170 023737 172354 003124      CMP    @KIPAR6,KIFLG ;END OF MEMORY?
2533 020176 101744                      BLUS   30$           ;BR IF NO
2534 020200 004737 017354      50$:   JSR    PC,KTOFF ;TURN OFF MEMORY MAPPING
2535 020204 000241                      CLC                    ;SET FAILURE
2536 020206 000403                      BR     60$           ;
2537 020210 004737 017354      55$:   JSR    PC,KTOFF ;TURN OFF MEMORY MAPPING
2538 020214 000261                      SEC                    ;SET SUCCESS
2539 020216 000207      60$:   RTS    PC
2540                      .SBTTL  REGSAV   SAVE R1 R5 ON STACK
2541                      ;
2542                      ;
2543                      ;ROUTINE TO
2544                      ;SAVE R1 THROUGH R5 ON THE STACK
2545                      ;
2546                      ;CALLING SEQUENCE:
2547                      ;
2548                      ;       JSR    R5,REGSAV
2549                      ;
2550                      ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
2551                      ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
2552                      ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
2553                      ;REGISTERS.
2554                      ;
2555                      ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
2556                      ;CALLED VIA A JSR PC INSTRUCTION
2557                      ;
2558                      ;
2559                      ;
2560 020220      REGSAV:
2561 020220 010446      MOV    R4,(SP)
2562 020222 010346      MOV    R3,-(SP)
2563 020224 010246      MOV    R2,(SP)
2564 020226 010146      MOV    R1,-(SP)
2565 020230 010546      MOV    R5,-(SP)
2566 020232 016605 000012      MOV    10.(SP),R5
2567 020236 004736      JSR    PC,@(SP)+
2568 020240 012601      MOV    (SP)+,R1
2569 020242 012602      MOV    (SP)+,R2
2570 020244 012603      MOV    (SP)+,R3
2571 020246 012604      MOV    (SP)+,R4
2572 020250 012605      MOV    (SP)+,R5
2573 020252 000207      RTS    PC
2574                      .SBTTL  GETPAT   GET 8 BIT PATTERN FROM OPERATOR
2575                      ;
2576                      ;
2577                      ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
2578                      ;
2579                      ;INPUTS:
2580                      ;
2581                      ;       NONE.
2582                      ;
2583                      ;OUTPUTS:
2584                      ;
2585                      ;       R0      OCTAL NUMBER FROM THE OPERATOR
2586                      ;
2587                      ;CALLING SEQUENCE:

```

```

2588      ;
2589      ;      JSR      PC,GETPAT
2590      ;
2591      ;
2592
2593 020254  GETPAT::
2594 020254      SAVREG          ;SAVE THE GENERAL REGISTERS
2595 020260  1$:      GMANID    DATASC,PATDAT,0,377,0,377,NO
                TRAP      C$GMAN
                BR        10000$
                .WORD    PATDAT
                .WORD    T$CODE
                .WORD    DATASC
                .WORD    377
                .WORD    T$LOLIM
                .WORD    T$HILIM
                10000$:
2596 020300      BNCOMPLETE    1$      ;RETRY IF ERROR
                BCC      1$
2597 020302  013700  020310      MOV      PATDAT,R0      ;DATA PATTERN FROM OPERATOR
2598 020306  000207      RTS      PC      ;RETURN TO CALLER
2599
2600      ;*
2601      ;LOCAL DATA AREA
2602      ;-
2603
2604 020310  000000      PATDAT: .WORD    0      ;TEMPORARY STORAGE FOR DATA
2605 020312  105      116      124  DATASC: .ASCIZ  'ENTER DATA PATTERN'
2606      .EVEN
2607      .SBTTL  GETSEL      ISSUE MENU AND GET OPEKATOR RESPONSE
2608
2609      ;*
2610      ;ROUTINE TO ISSUE A MENU AND GET
2611      ;THE OPERATOR'S RESPONSE.
2612      ;
2613      ;INPUTS:
2614      ;
2615      ;      R0      ADDRESS OF ASCIZ STRING OF MENU
2616      ;      R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
2617      ;
2618      ;OUTPUTS:
2619      ;
2620      ;      R0      NUMBER OF THE OPERATOR S SELECTION
2621      ;
2622      ;-
2623
2624 020336  GETSEL::
2625 020336      SAVREG          ;SAVE GENERAL REGISTERS
2626 020342  010002      MOV      R0,R2      ;SAVE THE MENU ADDRESS
2627 020344  010203  1$:      MOV      R2,R3      ;START OF MENU STRING
2628 020346  005713  2$:      TST      (R3)      ;END OF ASCII ?
2629 020350  001412      BEQ      3$      ;BRANCH IF ALL LINES DISPLAYED
2630 020352      PRINTF      @SELASC,(R3)+      ;DISPLAY THE MENU
                MOV      (R3)+,-(SP)
                MOV      @SELASC,(SP)
                MOV      @2,-(SP)
                MOV      SP,R0

```

```

020366 104417          TRAP  C$PNTF
020370 062706 000006  ADD   #6,SP
2631 020374 000764          BR    2$
2632 020376          3$:  GMANID MENASC,MENRES,D. 1.0. 1.NO
020376 104443          TRAP  C$GMAN
020400 000406          BR    10001$
020402 020556          .WORD MENRES
020404 000042          .WORD T$CODE
020406 020527          .WORD MENASC
020410 177777          .WORD -1
020412 000000          .WORD T$LOLIM
020414 177777          .WORD T$HILIM
020416          10001$:
2633 020416          BNCOMPLETE 1$ ;RETRY IF ERROR
020416 103352          BCC   1$
2634 020420 013700 020556  MOV   MENRES,RO ;GET THE OPERATOR'S REPLY
2635 020424 020001          CMP   RO,R1 ;COMPARE TO MAXIMUM ALLOWED
2636 020426 101411          BLOS  5$ ;BRANCH IF OK
2637 020430          PRINTF #MENERR ;DISPLAY ERROR MESSAGE
020430 012746 020454  MOV   #MENERR,-(SP)
020434 012746 000001  MOV   #1,-(SP)
020440 010600          MOV   SP,RO
020442 104417          TRAP  C$PNTF
020444 062706 000004  ADD   #4,SP
2638 020450 000735          BR    1$ ;RETRY
2639 020452 000207          5$:  RTS   PC ;RETURN TO CALLER
2640 020454 045 116 045  MENERR: .ASCIZ 'MNA *** Menu Selection Too Large ***'
2641 020522 045 116 045  SELASC: .ASCIZ 'MNT'
2642 020527 105 156 164  MENASC: .ASCIZ 'Enter Menu Selection: '
2643          .EVEN
2644 020556 000000  MENRES: .WORD 0
2645          .SBTTL CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
2646          ;
2647          ;
2648          ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2649          ;
2650          ;INPUT:
2651          ;
2652          ; NONE.
2653          ;
2654          ;OUTPUT:
2655          ;
2656          ; CARRY 0 MANUAL INTERVENTION NOT ALLOWED
2657          ; CARRY 1 MANUAL INTERVENTION IS OK
2658          ;
2659          ;SIDE EFFECTS:
2660          ;
2661          ; A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2662          ; NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2663          ; ALLOWED.
2664          ;
2665          ;
2666          ;
2667 020560          CHKMAN: :
2668 020560          SAVREG ;SAVE THE REGISTERS
2669 020564          MANUAL ;SEE IF MANUAL INTERVENTION OK
020564 104450          TRAP  C$MANI
  
```



```

2670 020566          BCOMPLETE 18          ;BRANCH IF ALLOWED
      020566 103411  BCS 18
2671 020570          PRINTF #NOMAN          ;PRINT THE WARNING MESSAGE
      020570 012746 020614  MOV #NOMAN, (SP)
      020574 012746 000001  MOV #1, -(SP)
      020600 010600  MOV SP, R0
      020602 104417  TRAP C:PNTF
      020604 062706 000004  ADD #4, SP
2672 020610 000241  CLC          ;CLEAR CARRY FOR ERROR
2673 020612 000207  18:  RTS PC          ;RETURN
2674
2675 020614 045 116 045 NOMAN: .ASCIZ 'ANSA *** Manual Intervention not Allowed Test Aborted ***'
2676 .even
2677 .SBTTL ENVIRN - SETUP FREE DIAGNOSTIC SPACE
2678 ;
2679 ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
2680 ;
2681 020710          ;ENVIRN: MEMORY R0
      020710 104431  TRAP C:MEM
2682 020712 010037 003116  MOV R0, FREE          ; GET 1ST FREE ADDRESS...
2683 020716 062737 000002 003116  ADD #2, FREE
2684 020724 011037 003120          MOV (R0), FRESIZ      ; ...AND WORD COUNT.
2685 020730 162737 000004 003120  SUB #4, FRESIZ
2686 020736 013702 002012          MOV L:UNIT, R2        ; GET NUMBER OF UNITS
2687 020742 162737 000007 003120 10:  SUB #7, FRESIZ        ; TAKE AWAY 7 WORDS PER UNIT
2688 020750 005302  DEC R2
2689 020752 001373  BNE 10:
2690 020754 013700 003116  MOV FREE, R0          ;GET FIRST FREE ADDRESS
2691 020760 063700 003120  ADD FRESIZ, R0        ;POINT TO LAST FREE ADDRESS
2692 020764 162700 000002  SUB #2, R0            ;BACKUP 1 WORD
2693 020770 010037 003122  MOV R0, FREEHI        ;STORE LAST FREE ADDRESS
2694 020774 000240  NOP
2695 020776 012701 177520  MOV #BDVPCR, R1      ;GET BDV11 PCR ADDRESS
2696 021002 010102  MOV R1, R2            ;COPY TO R2
2697 021004 062702 000002  ADD #2, R2            ;SET THE RANGE
2698 021010 004737 016456  JSR PC, XNXM          ;SEE IF WE HAVE ONE
2699 021014 103001  BCC 15:              ;OK TO SET FLAGS
2700 021016 000445  BR 40:                ;RETURN WITH FLAGS CLEAR
2701 021020 013701 177520 15:  MOV BDVPCR, R1        ;SAVE PCR CONTENTS
2702 021024 062701 000001  ADD #1, R1            ;ADD ONE TO IT
2703 021030 012702 177520  MOV #BDVPCR, R2        ;GET BDV11 PCR ADDRESS
2704 021034 005212  INC (R2)              ;TRY TO WRITE TO IT
2705 021036 013703 177520  MOV BDVPCR, R3        ;GET RESULTS
2706 021042 020103  CMP R1, R3            ;DID IT CHANGE?
2707 021044 001017  BNE 20:                ;NO, MUST BE 11/23B
2708 021046 005237 003136  INC T23A              ;SET THE FLAG
2709 021052 042737 170000 002120  BIC #170000, L:TIME ;SUPERVISOR COULD BE WRONG
2710 021060 000240  NOP
2711 021062          PRINTF #M8186          ;TELL THE SYSTEM TYPE
      021062 012746 005550  MOV #M8186, -(SP)
      021066 012746 000001  MOV #1, -(SP)
      021072 010600  MOV SP, R0
      021074 104417  TRAP C:PNTF
      021076 062706 000004  ADD #4, SP
2712 021102 000413  BR 40:                ;RETURN
2713 021104 005237 003140 20:  INC T23B              ;SET THE FLAG
2714 021110 000240  NOP
      ;BR 40: FOR RELEASE
  
```

```

2715 021112          PRINTF  #M8189      ; TELL THE SYSTEM TYPE
      021112 012746 005641      MOV      #M8189, (SP)
      021116 012746 000001      MOV      #1, (SP)
      021122 010600          MOV      SP, R0
      021124 104417          TRAP     C:PNTR
      021126 062706 000004      ADD      #4, SP
2716 021132 000207          40$:   RTS      PC      ; RETURN
2717          .SB11L  KTINIT  SETUP  KT11 MEMORY MANAGEMENT REGISTERS
2718          ;
2719          ;
2720          ; ROUTINE TO INIT KT 11
2721          ;
2722          ;
2723          ;
2724 021134          KTINIT:
2725 021134 005037 003124      CLR      KTFLG      ; INIT >28K MEMORY FLAG
2726 021140 005037 003126      CLR      KTENABLE   ; INIT TEST >28K FLAG
2727 021144 023727 002120 001577  CMP     L#HIME, #1577 ; GOT ENOUGH MEMORY (>28K)?
2728 021152 101444          BLOS    9$          ; NO.
2729 021154 013700 000004      MOV     @#ERRVEC, R0 ; SAVE OLD ERR VEC PTR.
2730 021160 012737 021252 000004  MOV     #2$, @#ERRVEC ; SET ERR VEC PTR.
2731 021166 005737 177572      TST     @#SRO      ; GOT KT11?
2732 021172 000240          NOP          ; (TRAP IF NO).
2733 021174 013737 002120 003124  MOV     L#HIME, KTFLG ; YES. SET KT FLAG.
2734 021202 042737 000177 003124  BIC     #177, KTFLG
2735 021210 010037 000004      MOV     R0, @#ERRVEC ; RESTORE OLD ERR VEC PTR.
2736 021214 005000          CLR      R0      ; R0 = AR DATA.
2737 021216 012701 172340      MOV     #KIPAR0, R1 ; R1 = KI REGS PTR.
2738 021222 012761 077406 177740 1$:   MOV     #77406, -40(R1) ; SET DESCRIPTOR REG.
2739 021230 010021          MOV     R0, (R1)    ; SET KIPAR REG.
2740 021232 062700 000200      ADD     #200, R0   ; BUMP AR DATA BY "4K".
2741 021236 020027 002000      CMP     R0, #2000 ; AT "I/O"?
2742 021242 001367          BNE     1$          ; NO.
2743 021244 012741 177600      MOV     #177600, -(R1) ; YES. SET KTPAR7 FOR I/O.
2744 021250 000405          BR      9$
2745          ;
2746 021252 012716 021260      2$:   MOV     #6$, (SP)   ; SET UP RETURN
2747 021256 000002          RTI          ; RTI TO NEXT LOCATION
2748          ;
2749 021260 010037 000004      6$:   MOV     R0, @#ERRVEC ; RESTORE OLD ERR VEC PTR.
2750          ;
2751 021264 000207          9$:   RTS      PC
2752          ;
2753          ; SUBROUTINE TO SET EXTENDED FEATURES SWITCH
2754          ;
2755          ; Requires that SOFINIT and WATCHR have been done previous to call.
2756          ;
2757          ;
2758          ; INPUTS:
2759          ; R5      CURRENT UNIT NUMBER
2760          ; OUTPUTS:
2761          ; The Extended Features Switch is set.
2762          ;
2763          ;
2764          ;
2765 021266          INVERT:
2766          ;

```

```

2767 021266 005737 002220          TST     EXTFEA          ; IS SWITCH SET?
2768 021272 001020          BNE     1$             ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
2769 021274 012737 100206 021340    MOV     @100206,CMDPKT ; WRT SUB SYS MEM CMD
2770 021302 012737 021350 021342    MOV     @WSMBK,CMDPKT.2 ; MSG BUF ADDR
2771 021310 012737 000006 021346    MOV     @6,CMDPKT.6     ; BYTE COUNT
2772 021316 012737 100010 021350    MOV     @100010,WSMBK   ; INVERT THE SWITCH
2773 021324 012704 021340          MOV     @CMDPKT,R4      ; SET CMDPKT INTO R4
2774 021330 004737 010742          JSR     PC,WRTCHR       ; DO IT
2775 021334 000207          1$:   RTS     PC        ; RETURN
2776
2777          ;   COMMAND PACKET.
2778
2779          021340          .   "   <..3>&177774 ;MUST BE ON MOD 4 BOUNDARY.
2780
2781 021340 000000          CMDPKT:: 0             ;1ST WORD IS TS05 COMMAND.
2782 021342 000000          0             ;2ND WORD IS THE BUFFER LOW ADDRESS.
2783 021344 000000          0             ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2784 021346 000000          0             ;4TH WORD IS THE BYTE/RECCRD/FILE COUNT.
2785
2786          ;   WRITE SUB SYSTEM MEMORY CHARACTERISTIC BLOCK.
2787
2788 021350 000000          WSMBK:: 0             ;1ST WORD:: SEL 0
2789 021352 000000          0             ;2ND WORD:: SEL 2
2790 021354 000000          0             ;3RD WORD:: SEL 4
2791          .EVEN
2792
2793          ;*   SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
2794          ;
2795          ;
2796          ;INPUTS:
2797          ;OUTPUTS:
2798          ;   The NXMFLG is set if we can test.
2799          ;   The NXML0 and NXMMI addresses are setup.
2800          ;
2801          ;
2802 021356          MEMCK::
2803
2804 021356          SAVREG          ;SAVE THE REGISTERS
2805 021362 005037 003130          CLR     NXMFLG         ;CLEAR THE FLAG
2806 021366 005037 003132          CLR     NXML0         ;CLEAR THE TEST ADDRESS LO
2807 021372 005037 003134          CLR     NXMMI         ;CLEAR THE TEST ADDRESS HI
2808 021376 005737 003140          TST     T23B          ;IS IT A 11/23B?
2809 021402 001407          BEQ     1$             ;NO
2810 021404 023727 002120 007777    CMP     L$HIME,@7777   ; GREATER THAN 128K
2811 021412 103406          BLO     2$             ; NO
2812 021414 004737 021532          JSR     PC,NXMTST     ;SETUP THE ADDRESS
2813 021420 000427          BR     13$            ;SET THE FLAG AND EXIT
2814 021422 005737 003136          1$:   TST     T23A          ;IS IT A 11/23A?
2815 021426 001413          BEQ     4$             ;NO
2816 021430 023727 002120 005777    2$:   CMP     L$HIME,@5777 ;GREATER THAN 96K
2817 021436 101023          BHI     14$            ;YES,23A/23B WITH 128K MEMORY
2818 021440 023727 002120 003777    CMP     L$HIME,@3777   ;GREATER THAN 64K BUT LESS THAN 92K?
2819 021446 103403          BLO     4$             ;NO, CHECK 24K
2820 021450 004737 021532          JSR     PC,NXMTST     ;SETUP THE ADDRESS
2821 021454 000411          BR     13$            ;SET THE FLAG AND EXIT
2822 021456 023727 002120 001577    4$:   CMP     L$HIME,@1577 ;GREATER THAN 24K BUT LESS THAN 64K?
2823 021464 103410          BLO     14$            ;NO, TELL THEM AND EXIT WITH FLAG CLEAR

```

TSV3 GLOBAL AREAS MACRO M1113 06 FEB 84 18:04
 KTINIT SETUP KT11 MEMORY MANAGEMENT REGISTERS

SEQ 082

```

2824 021466 004737 021532      JSR    PC,NXMTST      ;SETUP THE ADDRESS
2825 021472 062737 000077 003134  ADD    #77,NXMHI     ;FOOL THE 11/02 & 11/03
2826 021500 005237 003130      INC    NXMFLG        ;SET THE FLAG
2827 021504 000411              BR     15$           ;EXIT
2828 021506 000410      14$:  BR     15$           ;NOP FOR PRINTOUT
2829 021510              PRINTF #NOMEM        ;TELL THEM & EXIT ***NO PRINT*****
      021510 012746 005454      MOV    #NOMEM, (SP)
      021514 012746 000001      MOV    #1,-(SP)
      021520 010600              MOV    SP,R0
      021522 104417              TRAP  C#PNTF
      021524 062706 000004      ADD    #4,SP
2830 021530 000207      15$:  RTS    PC              ;RETURN
2831
2832      ;*
2833      ;      SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
2834      ;
2835      ;OUTPUTS: NXMLO,NXMHI      ;SETUP WITH NXM ADDRESS
2836      ;
2837      ;-
2838
2839 021532 013701 002120  NXMTST: MOV    L#HIME,R1      ;GET TOP OF MEMORY
2840 021536 062701 000200      ADD    #200,R1         ;MAKE IT I/O BLOCK OR OTHER NXM
2841 021542 042701 000177      BIC    #177,R1
2842 021546 C10102              MOV    R1,R2           ;RESAVE RESULTS
2843              .REPT    6
2844              ASL    R1           ;PUT IN PLACE FOR XFER
2845              .ENDR
2846 021564 010137 003132      MOV    R1,NXMLO        ;SAVE TEST ADDRESS LOW
2847              .REPT    10.
2848              ASR    R2           ;PUT IN PLACE FOR XFER
2849              .ENDR
2850 021614 042702 177700      BIC    #177700,R2     ;DON'T WANT ILA!
2851 021620 010237 003134      MOV    R2,NXMHI        ;SAVE TEST ADDRESS HIGH
2852 021624 000207      RTS    PC              ;RETURN
2853
2854
2855
2856 021626      ENDMOD

```

TSV4 MISCELLANEOUS SECTIONS MACRO M1113 06 FEB 84 18:04
KTINIT SETUP KT11 MEMORY MANAGEMENT REGISTERS

SEQ 083

7 .TITLE TSV4 MISCELLANEOUS SECTIONS
8
9 021626 BGNMOD TSV4
021626 TSV4::
10

G7

TSV4 - MISCELLANEOUS SECTIONS MACRO M1113 06-FEB 84 18:04
PROTECTION TABLE

SEQ 084

```
17  
18 021626 .SBTTL PROTECTION TABLE  
    021626 BGNPROT  
19 021626 177777 177777 177777 L$PROT::  
20 021636 .WORD -1. 1. 1. 1  
    ENDPROT
```

;NO DEVICE PROTECTION REQUIRED.

```

22          .SBTTL  INITIALIZE SECTION
23
24          ;**
25          ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
26          ;AT THE BEGINNING OF EACH PASS.
27
28          ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
29          ;IF "CONTINUE", NOTHING IS REQUIRED.
30
31          ;
32          ;
33          ;*
34          ;INSERT TEMPORARY JUMP TO ODT
35          ;
36          BGNINIT
37          L$INIT::
38          40$:  CLR      EXTFEA
39              CLR      NXMFLG
40              MOV      @EPR1,EPR1SW      ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
41              CLR      SIFLAG            ;CLEAR "SOFT INIT" FLAG
42              CLR      KTENABLE         ;CLEAR TEST ABOVE 28K FLAG
43              CLR      RAMSIZ           ;CLEAR RAM SIZE FOR RAMERR ROUTINE
44              READEF   @EF.CONTINUE
45              MOV      @EF.CONTINUE,RO
46              TRAP     C$REFG
47          BNCOMPLETE 1$
48          BCC      1$
49          CMP      UNITN,L$UNIT          ;UNIT IN RANGE?
50          BHIS     4$                    ;BR IF NO.
51          TST     DUFLG                  ;DROPPED UNIT?
52          BMI     NXTU                   ;BR IF YES
53          MOV     UNITN,R1
54          ASL     R1
55          TST     ERTABL(R1)
56          BEQ     SETU
57          BIT     @BIT14,ERTABL(R1)      ;DROPPED?
58          BNE     NXTU
59          EXIT     INIT                  ;DO NOTHING IF 'CONTINUE'.
60          TRAP     C$EXIT
61          .WORD   L10030-.
62          1$:  READEF   @EF.NEW
63              MOV     @EF.NEW,RO
64              TRAP     C$REFG
65          BNCOMPLETE NXTU                ;TAKE NEXT UNIT IF NOT NEW PASS.
66          BCC     NXTU
67          READEF   @EF.START
68              MOV     @EF.START,RO
69              TRAP     C$REFG
70          BCOMPLETE 2$
71          BCS     2$
72          READEF   @EF.RESTART
73              MOV     @EF.RESTART,RO
74              TRAP     C$REFG
75          BNCOMPLETE 31$
76          BCC     31$
77          2$:  BRESET
78              TRAP     C$RESET
79              ;1ST PASS, BUS INIT...
80              ;BUS RESET.

```

63	022000	005037	002206		CLR	TSTCNT		;NUMBER OF TESTS RUN IN PASS
64	022004	005037	002214		CLR	FATFLG		;CLEAR FATAL ERROR COUNT
65	022010	005037	003136		CLR	T23A		;CLEAR 11/23A FLAG
66	022014	005037	003140		CLR	T23B		;CLEAR 11/23B FLAG
67					MOV	#340, (SP)		
68					MOV	#208, (SP)		;RETURN TO DEBUGGER
69					JMP	0.00T		;ENTER THE DEBUGGER
70	022020	005037	003372		CLR	SKIPT		;CLEAR THE SUBTEST "SKIPPER"
71	022024			208:				
72	022024	012737	177777	002176	MOV	#-1 QVP		;...QUICK VERIFY...
73	022032	004737	020710		JSR	PC,ENVIRN		;SET ENVIRONMENT.
74	022036	004737	021134		JSR	PC,KTINIT		;INITIALIZE KT MEMORY MANAGEMENT
75	022042	012700	003170		MOV	#ERTABL,RO		
76	022046	005020		308:	CLR	(RO).		;CLEAR THE ERROR TABLE
77	022050	020027	003370		CMP	RO,#ERTABE		
78	022054	103774			BLO	308		
79	022056	000404			BR	48		
80	022060	005037	002176	318:	CLR	QVP		
81	022064	000137	022134		JMP	PASRPT		;GO REPORT THE STATUS
82								
83	022070			48:				
84	022070	012737	177777	002174	NEWPAS:	MOV	#-1,UNITN.	;INIT UNIT NUMBER...
85	022076	005037	002212		CLR	DEV CNT		;CLEAR COUNT OF DEVICES RUNNING.
86	022102			NXTU:	BREAK			
	022102	104422			TRAP	C#BRK		
87	022104	005237	002174		INC	UNITN		;...AND SET NEXT UNIT NUMBER.
88	022110	023737	002174	002012	CMP	UNITN,L#UNIT		
89	022116	103423			BLO	SETU		
90	022120	012737	177777	003104	MOV	#1,DUFLG		
91	022126	000401			BR	118		
92	022130				DOCLN			;ABORT, NO MORE UNITS.
	022130	104444			TRAP	C#DCLN		
93	022132	000240		118:	NOP			
94	022134			PASRPT:				
95	022134	023727	002012	000001	CMP	L#UNIT,#1		;HOW MANY UNITS SELECTED?
96	022142	101752			BLOS	NEWPAS		;BR IF ONLY 1
97	022144	005737	002212		TST	DEV CNT		;ARE ANY STILL RUNNING?
98	022150	001747			BEQ	NEWPAS		;BR IF NO
99	022152				RFLAGS	RO		
	022152	104421			TRAP	C#RFLA		
100	022154	032700	000100		BIT	#ISR,RO		;SHOULD WE PRINT STATISTICS
101	022160	001343			BNE	NEWPAS		;BR IF NO
102								
103	022162				DORPT			
	022162	104424			TRAP	C#DRPT		
104	022164	000741			BR	NEWPAS		
105	022166			108:				
106								
107	022166			SETU:	GPHARD	UNITN,RO		;GET UNIT N P TABLE POINTER.
	022166	013700	002174		MOV	UNITN,RO		
	022172	104442			TRAP	C#GPHRD		
108	022174				BNCOMPLETE	NXTU		;BR IF UNIT NOT AVAILABLE.
	022174	103342			BCC	NXTU		
109	022176	005037	003104		CLR	DUFLG		;CLEAR "DROPPED" FLAG.
110	022202	005237	002212		INC	DEV CNT		
111	022206	012001			MOV	(RO),R1		;GET 1ST REGISTER ADDRESS.
112	022210	010137	002206		MOV	R1,CSRADDR		;ADDRESS OF REGISTERS OF UNIT UNDER TEST

J7

```

113
114 022214 012001      MOV      (R0),R1      ;GET VECTOR ADDRESS.
115                   ;MOV      (R0),R2      ;GET INTERRUPT PRIORITY
116                   ;MOV      R2,IPRT    ;SET INTERRUPT PRIORITY.
117 022216 010137 002202 MOV      R1,IVCC      ;SET INTERRUPT VECTOR POINTER...
118 022222 012721 016276 MOV      @INTR,(R1).  ;...VECTOR...
119 022226 013721 002204 MOV      IPRI,(R1).  ;...AND PRIORITY.
120
121 022232             1$:
122                   ; TST      QVP          ;1ST PASS ??
123                   ; BEQ      5$          ;NO, SKIP THE PASS 1 STUFF.
124
125
126                   ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
127                   ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
128
129 022232 013701 002174      MOV      UNITN,R1
130 022236 006301             ASL      R1
131 022240 052761 100000 003170 BIS      @BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
132 022246 005037 005766      CLR      EXTA        ;CLEAR ERROR EXTENSION FLAG.
133 022252 023727 002012 000001 CMP      L$UNIT,#1    ;ARE WE TESTING MULTIPLE UNITS?
134 022260 101416             BLOS    10$          ;BR IF NO.
135 022262             RFLAGS  R0        ;YES - GET OPERATOR FLAGS.
136 022264 104421             TRAP   C$RFLA
137 022270 001412             BIT     @PNT,R0        ;SHOULD WE PRINT UNIT #?
138 022272             BEQ     10$          ;BR IF NOT.
139 022272             PRINTF @PUNIT,UNITN ;PRINT THE UNIT #
140 022276 013746 002174      MOV      UNITN,(SP)
141 022276 012746 022364      MOV      @PUNIT,(SP)
142 022302 012746 000002      MOV      #2,(SP)
143 022306 010600             MOV     SP,R0
144 022310 104417             TRAP   C$PNTF
145 022312 062706 000006      ADD     #6,SP
146 022316             10$:
147 022316 005037 003106      CLR     NODEV
148 022322 013701 002200      MOV     CSRADDR,R1  ;ADDRESS OF FIRST REGISTER
149 022326 010102             MOV     R1,R2        ;START OF REGISTERS
150 022330 062702 000002      ADD     @TSSR,R2    ;ADDRESS OF TSSR REGISTER
151 022334 004737 016456      JSR     PC,XNXM     ;TEST BOTH CONTROLLER REGISTERS...
152 022340 103005             BCC    2$          ;...AND BR IF ALL OK.
153 022342 010137 003106      MOV     R1,NODEV   ;FLAG DEVICE AS NON EXISTENT
154 022346 012737 177777 003104 MOV     #!,DUFLG    ;DROP THIS UNIT.
155 022354             2$:
156                   ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
157
158                   5$:
159 022354             SETPRI @PRI00          ;ENABLE INTERRUPTS.
160 022354 012700 000000      MOV     @PRIOC,R0
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 /#N#N#A***** TESTING UNIT #02#A *****/
167 .EVEN

```

```

158                                     .SBTTL  ADD AND DROP UNITS SECTIONS
159
160                                     ;**
161                                     ; THE ADD UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
162                                     ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
163                                     ; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
164                                     ;
165 022432                               BGNAU
166 022432                               L$AU::
167 022432 010001                         MOV     R0,R1                ; GET UNIT TO BE ADDED (R0)
168 022434 006301                         ASL     R1                  ; MAKE IT A WORD INDEX
169 022436 052761 100000 003170           BIS     #100000,ERTABL(R1) ; SET THE 'ACTIVE' BIT
170 022444 042761 040000 003170           BIC     #40000,ERTABL(R1) ; CLEAR THE 'DROPPED' BIT
171 022452                               PRINTF  #1$,R0
172 022452 010046                         MOV     R0,-(SP)
173 022454 012746 022500                   MOV     #1$,-(SP)
174 022460 012746 000002                   MOV     #2,-(SP)
175 022464 010600                         MOV     SP,R0
176 022466 104417                         TRAP   C$PNTF
177 022470 062706 000006                   ADD     #6,SP
178 022474                               EXIT   AU
179 022474 000167                         .WORD  J$JMP
180 022476 000026                         .WORD  L10031-2-
181 022500 045 116 045 1$:                .ASCIZ  /%VA UNIT %D%A ADDED/
182                                     .EVEN
183
184                                     ENDAU                ; UNUSED.
185 022526                               L10031-
186 022526 104452                         TRAP   C$AU
187
188                                     ;**
189                                     ; THE DROP UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
190                                     ; TO BE REMOVED FROM THE TEST LIST.
191                                     ;
192                                     ; SUPVSR DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN.
193                                     ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
194                                     ; COMMAND, OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
195                                     ; WILL PRINT ALL DROPPED UNITS AND THE P-TABLES OF THOSE
196                                     ; WHICH ARE STILL ACTIVE.
197                                     ; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.
198
199 022530                               BGNDU
200 022530                               L$DU::
201 022530 012737 177777 003104           MOV     #-1,DUFLG
202 022536 010001                         MOV     R0,R1
203 022540 006301                         ASL     R1
204 022542 052761 140000 003170           BIS     #140000,ERTABL(R1) ; SAY DROPPED
205 022550 000240 000240 000240           240,240,240 ; ??????????
206 022556                               PRINTF  #1$,R0
207 022556 010046                         MOV     R0,(SP)
208 022560 012746 022604                   MOV     #1$,-(SP)
209 022564 012746 000002                   MOV     #2,-(SP)
210 022570 010600                         MOV     SP,R0
211 022572 104417                         TRAP   C$PNTF
212 022574 062706 000006                   ADD     #6,SP
213 022600                               EXIT   DU
214 022600 000167                         .WORD  J$JMP
215 022602 000030                         .WORD  L10032 2 .

```

```

195 022604      045      116      045 1$: .ASCIZ /#N#A UNIT #D#A DROPPED/
196                                     .EVEN
197 022634                                     ENDCJ
    022634                                     L10032:
    022634 104453                                     TRAP C$DU
198                                     ;**
199                                     ; AUTO-DROP CODE SECTION.
200                                     ;
201 022636                                     BGNALTO
    022636                                     L$ALTO:
202 022636 013705 002200                                     MOV CSRADDR,R5 ;POINT TO DEVICE REGISTER
203 022642 012703 000550                                     MOV #360.,R3 ;ENOUGH TIME FOR 2400 REEL TO REWIND
204 022646 004737 016330 10$: JSR PC,WAITF ;WAIT FOR SSR TO SET
205 022652 103420                                     BCS 20$ ;LEAVE WHEN SSR IS SET
206 022654                                     DELAY 250. ;WAIT FOR .25 SECONDS
    022654 012727 000372                                     MOV #250.,(PC)+
    022660 000000                                     .WORD 0
    022662 013727 002116                                     MOV L$DLY,(PC)+
    022666 000000                                     .WORD 0
    022670 005367 177772                                     DEC -6(PC)
    022674 001375                                     BNE -.4
    022676 005367 177756                                     DEC -22(PC)
    022702 001367                                     BNE .-20
207 022704 005303                                     DEC R3 ;BUMP COUNTER DOWN
208 022706 001357                                     BNE 10$ ;KEEP GOING
209 022710 004737 017262                                     JSR PC,CKDROP ;TRY AND DROP UNIT
210 022714
211 022714
    022714                                     20$: ENDAUTO ; UNUSED.
    022714 104461                                     L10033:
    022714                                     TRAP C$AUTO

```

```

213                                     .SBTTL CLEAN UP AND REPORT CODING SECTIONS
214
215
216     ;**
217     ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
218     ; EXECUTED AT THE END OF EACH PASS (OR SUB PASS).
219     ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
220
220 022716                                BGNCLN
220 022716                                L$CLEAN::
221 022716 013705 002200                    MOV     CSRADDR,R5                ;POINT TO DEVICE REGISTER
222 022722 005737 003104                    TST     DUFLG                    ; DROPPED" FLAG IS SET ON...
223 022726 100405                            BMI     1$                       ;...AND GROSS CONTROLLER FAULT...
224
224
224
225
225 022730 012765 000000 000002            MOV     #0,TSSR(R5)             ;DO SOFT INIT
227 022736 004737 016330                    JSR     PC,WAITF
228 022742
229 022742
229 022742
229 022742 104412                            1$:
229 022742                                2$:
229 022742                                L10034:
229 022742                                TRAP   C$CLEAN
230
230     ;**
231     ; THE REPORT CODING SECTION CONTAINS THE
232     ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
233     ;
234
234 022744                                BGNRPT
234 022744                                L$RPT::
235 022744                                PRINTS #DEVSUM
235 022744 012746 023206                    MOV     #DEVSUM,-(SP)
235 022750 012746 000001                    MOV     #1,-(SP)
235 022754 010600                            MOV     SP,R0
235 022756 104416                            TRAP   C$PNTS
235 022760 062706 000004                    ADD     #4,SP
236 022764 010246                            MOV     R2,-(SP)
237 022766 010346                            MOV     R3,-(SP)
238 022770 010446                            MOV     R4,-(SP)
239 022772 012704 003170                    MOV     #ERTABL,R4             ; GET START OF ERROR TABLE.
240 022776 005003                            CLR     R3                     ; CLEAR UNIT NUMBER
241 023000 011402                            1$: MOV     (R4),R2             ; GET ERROR TABLE ENTRY & TEST IT.
242 023002 001467                            BEQ     4$                     ; ZERO IF UNIT NOT RUN
243 023004 100066                            BPL     4$
244 023006 032702 040000                    BIT     #BIT14,R2             ; WAS UNIT DROPPED?
245 023012 001015                            BNE     2$                     ; BR IF YES
246 023014 042702 170000                    BIC     #C7777,R2             ; GET ERROR COUNT FIELD
247 023020                                PRINTS #DEVONL,R3,R2          ; PRINT
247 023020 010246                            MOV     R2,-(SP)
247 023022 010346                            MOV     R3,-(SP)
247 023024 012746 023243                    MOV     #DEVONL,-(SP)
247 023030 012746 000003                    MOV     #3,-(SP)
247 023034 010600                            MOV     SP,R0
247 023036 104416                            TRAP   C$PNTS
247 023040 062706 000010                    ADD     #10,SP
248 023044 000446                            BR     4$
249 023046 020227 160000                    2$: CMP     R2,#160000          ; WAS UNIT NON-EXISTENT?
250 023052 001012                            BNE     3$                     ; BR IF NO
251 023054                                PRINTS #DEVNXR,R3
251 023054 010346                            MOV     R3,-(SP)
251 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
252 023076 000431      BR       4$
253 023100 020227 160001      3$:    CMP      R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
254 023104 001012      BNE     30$      ; BR IF NO.
255 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
256 023130 000414      BR       4$
257 023132 042702 170000      30$:   BIC     #C7777,R2
258 023136      PRINTS  #DEVDR0,R3,R2
      023136 010246      MOV      R2,-(SP)
      023140 010346      MOV      R3,-(SP)
      023142 012746 023456      MOV      #DEVDR0,-(SP)
      023146 012746 000003      MOV      #3,-(SP)
      023152 010600      MOV      SP,R0
      023154 104416      TRAP    C:PNTS
      023156 062706 000010      ADD      #10,SP
259 023162 062704 000002      4$:    ADD      #2,R4
260 023166 005203      INC     R3
261 023170 020427 003370      CMP     R4,#ERTABE
262 023174 103701      BLO    1$
263 023176 012604      MOV     (SP),R4
264 023200 012603      MOV     (SP),R3
265 023202 012602      MOV     (SP),R2
266 023204      ENDRPT      ; UNUSED.
      023204      L10035:
      023204 104425      TRAP    C:RPT
267
268 023206      045      116      045  DEVSUM: .ASCIZ /#N#ADEVICE STATUS SUMMARY:#N/
269 023243      045      101      040  DEVONL: .ASCIZ /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
270 023313      045      101      040  DEVNXR: .ASCIZ /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
271 023375      045      101      040  DEVNRD: .ASCIZ /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
272 023456      045      101      040  DEVDR0: .ASCIZ /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
273      .EVEN
274
275 023526      ENDMOD
276

```

TSV4 MISCELLANEOUS SECTIONS MACRO M1113 06 FEB 84 18:04
CLEAN UP AND REPORT CODING SECTIONS

SEQ 092

1
2
3
10
11
12

023526
023526

.TITLE TEST 1 HARDWARE TEST 1 8 TESTS
BGNMOD TSV7B
TSV7B::

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 1: WRITE TAPE MARK RETRY

SEQ 096

	024242	000154								.WORD	108
	024244	003646								.WORD	SFIERR
	024246	012114								.WORD	SFIMSG
177	024250	013737	002174	026340	20:	MOV	UNITN,T29DSW		;SET UP UNIT NUMBER		
178											
179	024256	012704	026320			MOV	#T29PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS		
180	024262	004737	010742			JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS		
181	024266	103407				BCS	25:		;BR, IF COMMAND ISSUED OK		
182	024270	005237	002214			INC	FATFLG		;ERROR COUNT		
186	024274	010001				MOV	RO,R1		;SAVE CONTENTS OF TSSR		
187	024276					ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTIC FAILED		
	024276	104456								TRAP	C\$ERHRD
	024300	000155								.WORD	109
	024302	005052								.WORD	WRTMSG
	024304	012114								.WORD	SFIMSG
188	024306				25:	CKLOOP			;LOOP IF SELECTED		
	024306	104406								TRAP	C\$CLP1
189	024310	004737	011074		26:	JSR	PC,REWIND		;CALL TAPE REWIND COMMAND		
190	024314	016501	000002			MOV	TSSR(R5),R1		;GET TSSR		
191	024320	012702	000200			MOV	#SSR,R2		;SET UP EXPECTED TSSR		
192	024324	103407				BCS	30:		;BR, IF NO PROBLEM		
193	024326	010004				MOV	RO,R4		;PACKET ADDRESS SET UP		
194	024330	005237	002214			INC	FATFLG		;ERROR COUNT		
198	024334					ERRHRD	ERRNO,T29RWN,PKTSSR		;REWIND NOT ACCEPTED		
	024334	104456								TRAP	C\$ERHRD
	024336	000156								.WORD	110
	024340	030305								.WORD	T29RWN
	024342	012126								.WORD	PKTSSR
199	024344				30:	CKLOOP			;LOOP IF SELECTED		
	024344	104406								TRAP	C\$CLP1
200	024346	013701	026350			MOV	T29BFR+6,R1		;PICK UP XSTO		
201	024352	010102				MOV	R1,R2		;SET UP EXPECTED		
202	024354	052702	000002			BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED		
203	024360	020102				CMP	R1,R2		;DOES EXP = REC'D		
204	024362	001406				BEQ	40:		;BR, IF EQUAL (OK)		
205	024364	005237	002214			INC	FATFLG		;ERROR COUNT		
209	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
210	024400	012737	000001	026442	40:	MOV	#1,T29RB		;NUMBER OF RECORDS TO SPACE OVER		
211	024406	012737	000400	026446		MOV	#256.,T29SZ		;SET UP RECORD SIZE		
212	024414	012737	140005	026440		MOV	#140005,T29PK3		;WRITE FORWARD,CVC=1,ACK COMMAND		
213	024422	012704	026440			MOV	#T29PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
214	024426	010465	000000			MOV	R4,TSD8(R5)		;ISSUE COMMAND		
215	024432	004737	016330			JSR	PC,WAITF		;WAIT FOR SSR TO SET		
216	024436	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
217	024442	012702	000200			MOV	#SSR,R2		;SET UP EXPECTED		
218	024446	020102				CMP	R1,R2		;ARE THEY EQUAL		
219	024450	001420				BEQ	75:		;BR, IF OK		
220	024452	013703	026350			MOV	T29BFR+6,R3		;PICK UP XTSD		
221	024456	032703	000004			BIT	#4,R3		;IS UNIT WRITE-LOCKED?		
222	024462	001405				BEQ	41:		;NO,PROCEED WITH NORMAL ERROR		
223	024464					ERRDF	ERRNO,T29WLK,SFIMSG		;TAPE IS WRITE LOCKED		
	024464	104455								TRAP	C\$ERDF
	024466	000157								.WORD	111

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 1: WRITE TAPE MARK RETRY

SEQ 099

313	025076	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND		
314	025102	103411			BCS	308		;BR, IF NO PROBLEM		
315	025104	016501	000002		MOV	TSSR(R5),R1		;GET TSSR		
316	025110	010004			MOV	R0,R4		;SAVE PACKET ADDRESS		
317	025112	005237	002214		INC	FATFLG		;ERROR COUNT		
321	025116				ERRHRD	ERRNO,T29RWN,PKTSSR		;REWIND NOT ACCEPTED		
	025116	104456							TRAP	C8ERHRD
	025120	000166							.WORD	118
	025122	030305							.WORD	T29RWN
	025124	012126							.WORD	PKTSSR
322	025126			308:	CKLOOP			;LOOP IF SELECTED		
	025126	104406							TRAP	C8CLP1
323	025130	013701	026350		MOV	T298FR+6,R1		;PICK UP XSTO		
324	025134	010102			MOV	R1,R2		;SET UP EXPECTED		
325	025136	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED		
326	025142	020102			CMP	R1,R2		;DOES EXP = REC D		
327	025144	001406			BEQ	408		;BR, IF EQUAL (OK)		
328	025146	005237	002214		INC	FATFLG		;ERROR COUNT		
332	025152				ERRHRD	ERRNO,T298OT,EXPREC		;TAPE NOT AT BOT AFTER REWIND		
	025152	104456							TRAP	C8ERHRD
	025154	000167							.WORD	119
	025156	027776							.WORD	T298OT
	025160	015554							.WORD	EXPREC
333	025162			408:	CKLOOP			;LOOP IF SELECTED		
	025162	104406							TRAP	C8CLP1
334	025164	012737	140011	026440	MOV	#140011,T29PK3		;WRITE TAPE MARK,ACK,CVC=1 COMMAND		
335	025172	012704	026440		MOV	#T29PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
336	025176	010465	000000		MOV	R4,TSD8(R5)		;ISSUE COMMAND		
337	025202	004737	016330		JSR	PC,WAIF		;WAIT FOR SSR TO SET		
338	025206	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
339	025212	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED		
340	025216	020102			CMP	R1,R2		;ARE THEY EQUAL		
341	025220	001406			BEQ	708		;BR, IF OK		
342	025222	005237	002214		INC	FATFLG		;ERROR COUNT		
346	025226				ERRHRD	ERRNO,T29WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE TAPE MARK		
	025226	104456							TRAP	C8ERHRD
	025230	000170							.WORD	120
	025232	030677							.WORD	T29WDC
	025234	012126							.WORD	PKTSSR
347	025236			708:	CKLOOP			;LOOP IF SELECTED		
	025236	104406							TRAP	C8CLP1
348	025240	012703	000001		MOV	#1.,R3		;NUMBER OF RECORDS TO WRITE TM		
349	025244	012737	141011	026440	MOV	#141011,T29PK3		;WRITE TAPE MARK RETRY,ACK,CVC=1 COMMAND		
350	025252	012704	026440		MOV	#T29PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
351	025256	010465	000000		MOV	R4,TSD8(R5)		;ISSUE COMMAND		
352	025262	004737	016330		JSR	PC,WAIF		;WAIT FOR SSR TO SET		
353	025266	016501	000002		MOV	TSSR(R5),R1		;PICK UP TSSR		
354	025272	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED (SSR ONLY)		
355	025276	020102			CMP	R1,R2		;WAS STATUS GOOD		
356	025300	001406			BEQ	1658		;BR, IF TERMINATION WAS GOOD		
357	025302	005237	002214		INC	FATFLG		;ERROR COUNT		
361	025306				ERRHRD	ERRNO,T29WDC,PKTSSR		;TSSR NOT CORRECT AFTER WRT TAPE M.		
	025306	104456							TRAP	C8ERHRD
	025310	000171							.WORD	121
	025312	030677							.WORD	T29WDC
	025314	012126							.WORD	PKTSSR
362	025316			1658:	CKLOOP			;LOOP IF SELECTED		

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 1: WRITE TAPE MARK RETRY

SEQ 101

```

411 025504 103426          BCS 20$          ;BR IF INIT WAS OK
412 025506          DELAY 250        ;DELAY ABOUT .25 SECONDS
      025506 012727 000250          MOV #250,(PC)+
      025512 000000          .WORD 0
      025514 013727 002116          MOV L$DLY,(PC)+
      025520 000000          .WORD 0
      025522 005367 177772          DEC -6(PC)
      025526 001375          BNE .4
      025530 005367 177756          DEC 22(PC)
      025534 001367          BNE .20
413 025536 005337 026500      DEC T29DLY        ;BUMP DELAY ROUTINE DOWN
414 025542 001356          BNE 10$          ;BR, IF MORE DELAY TIME LEFT
415 025544 005237 002214      INC FATFLG        ;ERROR COUNT
419 025550 010001          MOV R0,R1         ;CONTENTS OF TSSR REGISTER
420 025552          ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      025552 104455          TRAP C$ERDF
      025554 000174          .WORD 124
      025556 003646          .WORD SFIERR
      025560 012114          .WORD SFIMSG
421 025562 013737 002174 026340 20$: MOV UNITN,T29DSW ;SET UP DRIVE NUMBER
422 025570 012704 026320      MOV #T29PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
423 025574 004737 010742      JSR PC,WRTCHR    ;ISSUE WRITE CHARACTERISTICS
424 025600 103407          BCS 23$          ;BR, IF COMMAND ISSUED OK
425 025602 005237 002214      INC FATFLG        ;ERROR COUNT
429 025606 010001          MOV R0,R1         ;SAVE CONTENTS OF TSSR
430 025610          ERRMRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      025610 104456          TRAP C$ERMRD
      025612 000175          .WORD 125
      025614 005052          .WORD WRTMSG
      025616 012114          .WORD SFIMSG
431 025620          23$: CKLOOP        ;LOOP IF SELECTED
      025620 104406          TRAP C$CLP1
432 025622 004737 011074      JSR PC,REWIND   ;CALL TAPE REWIND COMMAND
433 025626 103411          BCS 30$          ;BR, IF NO PROBLEM
434 025630 016501 000002      MOV TSSR(R5),R1 ;GET TSSR
435 025634 010004          MOV R0,R4         ;SAVE PACKET ADDRESS
436 025636 005237 002214      INC FATFLG        ;ERROR COUNT
440 025642          ERRMRD ERRNO,T29RWN,PKTSSR ;REWIND NOT ACCEPTED
      025642 104456          TRAP C$ERMRD
      025644 000176          .WORD 126
      025646 030305          .WORD T29RWN
      025650 012126          .WORD PKTSSR
441 025652          30$: CKLOOP        ;LOOP IF SELECTED
      025652 104406          TRAP C$CLP1
442 025654 013701 026350      MOV T298FR+6,R1 ;PICK UP XST0
443 025660 010102          MOV R1,R2         ;SET UP EXPECTED
444 025662 052702 000002      BIS #BIT1,R2    ;SET BOT BIT IN EXPECTED
445 025666 020102          CMP R1,R2        ;DOES EXP = REC'D
446 025670 001406          BEQ 40$          ;BR, IF EQUAL (OK)
447 025672 005237 002214      INC FATFLG        ;ERROR COUNT
451 025676          ERRMRD ERRNO,T29BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      025676 104456          TRAP C$ERMRD
      025700 000177          .WORD 127
      025702 027776          .WORD T29BOT
      025704 015554          .WORD EXPREC
452 025706          40$: CKLOOP        ;LOOP IF SELECTED
      025706 104406          TRAP C$CLP1

```

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 1: WRITE TAPE MARK RETRY

SEQ 102

453	025710	012737	140011	026440	MOV	#140011,T29PK3	;WRITE TAPE MARK,ACK,CVC=1 COMMAND
454	025716	012704	026440		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
455	025722	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND
456	025726	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
457	025732	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS
458	025736	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED
459	025742	020102			CMP	R1,R2	;ARE THEY EQUAL
460	025744	001406			BEQ	70\$;BR, IF OK
461	025746	005237	002214		INC	FATFLG	;ERROR COUNT
465	025752				ERRHRD	ERRNO,T29WDC,PKTSSR	;TSSR INCORRECT AFTER WRITE TAPE MARK
	025752	104456					TRAP C\$ERHRD
	025754	000200					.WORD 128
	025756	030677					.WORD T29WDC
	025760	012126					.WORD PKTSSR
466	025762			70\$:	CKLOOP		;LOOP IF SELECTED
	025762	104406					TRAP C\$CLP1
467	025764	012703	000012		150\$:	MOV #10.,R3	;NUMBER OF RECORDS TO WRITE TM
468	025770	012737	000001	026442	MOV	#1,T29RB	;SET UP PACKET
469	025776	012737	141011	026440	MOV	#141011,T29PK3	;WRITE TAPE MARK RETRY,ACK,CVC=1 COMMAND
470	026004	012704	026440		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
471	026010	010465	000000		155\$:	MOV R4,TSDB(R5)	;ISSUE COMMAND
472	026014	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
473	026020	016501	000002		MOV	TSSR(R5),R1	;PICK UP TSSR
474	026024	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED (SSR ONLY)
475	026030	020102			CMP	R1,R2	;WAS STATUS GOOD
476	026032	001406			BEQ	165\$;BR, IF TERMINATION WAS GOOD
477	026034	005237	002214		INC	FATFLG	;ERROR COUNT
481	026040				ERRHRD	ERRNO,T29WDC,PKTSSR	;TSSR NOT CORRECT AFTER WRT TAPE M.
	026040	104456					TRAP C\$ERHRD
	026042	000201					.WORD 129
	026044	030677					.WORD T29WDC
	026046	012126					.WORD PKTSSR
482	026050			165\$:	CKLOOP		;LOOP IF SELECTED
	026050	104406					TRAP C\$CLP1
483	026052	005303			DEC	R3	;BUMP COUNTER DOWN
484	026054	001355			BNE	155\$;BR, IF LESS THAN 10 TAPE MARKS
485	026056	012737	140410	026440	MOV	#140410,T29PK3	;SPACE REVERSE,ACK,CVC=1. COMMAND
486	026064	012737	000001	026442	MOV	#1,T29RB	;NUMBER OF RECORDS TO SPACE BACK
487	026072	012704	026440		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
488	026076	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND
489	026102	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
490	026106	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS
491	026112	012702	100204		MOV	#SSR!SC!BIT2,R2	;SET UP EXPECTED
492	026116	020102			CMP	R1,R2	;ARE THEY EQUAL
493	026120	001406			BEQ	222\$;BR, IF OK
494	026122	005237	002214		INC	FATFLG	;ERROR COUNT
498	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
499	026136			222\$:	CKLOOP		;LOOP IF SELECTED
	026136	104406					TRAP C\$CLP1
500	026140	012737	100410	026440	MOV	#100410,T29PK3	;SPACE REVERSE,ACK. COMMAND
501	026146	012737	000005	026442	MOV	#5,T29RB	;NUMBER OF RECORDS TO SPACE BACK
502	026154	012704	026440		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
503	026160	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 1: WRITE TAPE MARK RETRY

SEQ 104

```

554 026336 000000          .WORD 0
555 026340 000000 T29DSW: .WORD 0          ;SELECT DRIVE 0
556 026342          T29BFR: .BLKW 25.      ;MESSAGE BUFFER
557
558          ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
559          ;
561          026430          .=<..+10>E177770
563 026430          T29PK2:
564 026430 100006          .WORD 100006          ;WRITE SUB SYS MEM COMMAND, AND ACK
565 026432 026450          .WORD T29BF2          ;ADDRESS OF SELECT BLOCK DATA
566 026434 000000          .WORD 0
567 026436 000006          .WORD 6.          ;SIZE OF DATA PACKET
568
572 026440          T29PK3:
573 026440 140005          .WORD 140005          ;WRITE TAPE MARK RETRY COMMAND, CVC=1 AND ACK
574 026442          T29RB:
575 026442 003116          T29WB: .WORD FREE          ;ADDRESS OF WRITE BUFFER
576 026444 000000          .WORD 0
577 026446 000000          T29SZ: .WORD 0          ;SIZE OF BUFFER (EXTENT)
578          .EVEN
579          ;
580          ;
581          ;
582 026450          T29BF2:
583 026450          010          T29BS0: .BYTE 10          ;BSELO AREA
584 026451          200          T29BS1: .BYTE 200          ;BSEL1 AREA
585 026452 000000          T29S2: .WORD 0          ;SEL 2 AREA
586 026454 000000          T29S3: .WORD 0          ;DATA AREA
587          ;
588          ;
589          .EVEN
590          ;TAPE MOTICN PACKET COMMAND VALUES
591
592 026456 140001          T29RN: .WORD 140001          ;READ DATA
593 026460 140401          T29WC?: .WORD 140401          ;READ DATA REVERSE
594 026462 141001          T29CON: .WORD 141001          ;READ PREVIOUS OPP=0
595 026464 161001          .WORD 161001          ;READ PREVIOUS OPP=1
596 026466 141401          .WORD 141401          ;WRITE TAPE MARK RETRY NEXT OPP=0
597 026470 161401          .WORD 161401          ;WRITE TAPE MARK RETRY NEXT OPP=1
598 026472 177777          .WORD 177777          ;END OF DATA
599
600          ;
601 026474 000000          T29CNT: .WORD 0          ;TAPE RECORD COUNTER STORAGE AREA
602
603 026476 000000          T29RSZ: .WORD 0          ;RECORD STORAGE SIZE AREA
604 026500 000000          T29DLY: .WORD          ;DELAY COUNTER STORAGE AREA
605          ;
606          ;LOCAL TEXT MESSAGES FOR TEST
607          ;-
608
609 026502          104          162          151          T29OFL: .ASCIZ 'Drive is OFFLINE
610 026523          124          141          160          T29WNG: .ASCIZ 'Tape Position Incorrect After WRITE TAPE MARK RETRY Previous (OPP=1)
611 026630          127          122          111          T29NEF: .ASCIZ 'WRITE TAPE MARK RETRY, At BOT, Failed To Set NEF (XSTO)
612 026720          124          123          123          T29RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
613 026767          127          122          111          T29RRF: .ASCIZ 'WRITE TAPE MARK RETRY Previous (Space Reverse, Read Forward) Command Failed
614 027103          127          122          111          T29RRG: .ASCIZ 'WRITE TAPE MARK RETRY Previous (Read Forward, Space Reverse) Command Failed
615 027217          120          117          123          T29SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct

```

616	027301	122	111	102	T29LOR:	.ASCIZ	'RIB NOT SET AFTER READ REVERSE INTO BOT'
617	027351	124	123	123	T29WDF:	.ASCIZ	'TSSR Not Correct After Illegal Mode Bits Set'
618	027426	111	154	154	T29LOQ:	.ASCIZ	'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
619	027507	127	122	111	T29SSR:	.ASCIZ	'WRITE TAPE MARK RETRY COMMAND Not Accepted'
620	027562	124	123	123	T29WDE:	.ASCIZ	'TSSR Not Correct After SPACE REVERSE DATA Command'
621	027644	052	052	052	T29WLK:	.ASCIZ	'*****TAPE IS WRITE-LOCKED AND WILL CAUSE ERRORS*****'
622	027731	124	123	123	T29WRT:	.ASCIZ	'TSSR Not Correct After WRITE Command'
623	027776	124	141	160	T29BOT:	.ASCIZ	'Tape Not At BOT After REWIND Command'
624	030043	104	141	164	T29DTA:	.ASCIZ	'Data Written To Tape Not Equal To Data Read From Tape'
625	030131	127	122	111	T29EOT:	.ASCIZ	'WRITE TAPE MARK RETRY DATA OVER EOT GAVE NO TAPE STATUS ALERT'
626	030227	124	123	123	T29TM:	.ASCIZ	'TSSR Not Correct After SPACE REVERSE Into BOT'
627	030305	122	145	167	T29RWI:	.ASCIZ	'Rewind (POSITION) Command Not Accepted'
628	030354	122	101	115	T29RNC:	.ASCIZ	'RAM Error, Correct Data Pattern Not In Ram'
629	030427	124	123	123	T29AMS:	.ASCIZ	'TSSR Init. Failed After WRITE TAPE MARK RETRY COMMAND'
630	030515	104	162	151	T29OF7:	.ASCIZ	'Drive 7 Select Failed To Set "OFL" In TSSR'
631	030570	124	123	123	T29WDD:	.ASCIZ	'TSSR Not Correct After WRITE TAPE MARK RETRY DATA Command, SWB Bit Set'
632	030677	124	123	123	T29WDC:	.ASCIZ	'TSSR Not Correct After WRITE TAPE MARK RETRY DATA Command'
633	030771	103	126	103	T29VCK:	.ASCIZ	'CVC Set, Didn't Reset VCK In Message Buffer'
634	031044	124	123	102	T29BA:	.ASCIZ	'TSBA Not Correct After WRITE TAPE MARK RETRY DATA Command'
635	031135	127	122	111	T29WSS:	.ASCIZ	'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
636	031225	122	145	141	T29LON:	.ASCIZ	'Reading Long Record Failed To Set RLL Bit In XSTO'
637	031307	122	145	141	T29LOP:	.ASCIZ	'Reading Long Record Failed To Set RLS Bit In XSTO'
638	031371	122	145	163	T29PBP:	.ASCIZ	'Residual Byte Count Incorrect After Short Record Read'
639	031457	122	145	141	T29TRL:	.ASCIZ	'Reading Long Record Failed To Give Tape Status Alert'
640	031545	104	141	164	T29NEQ:	.ASCIZ	'Data WRITE TAPE MARK RETRY From Tape Not Correct, After SWB=1'
641	031643	124	123	123	T29RDG:	.ASCIZ	'TSSR Incorrect After READ REVERSE Into Tape Mark'
642	031724	127	122	111	T29RIB:	.ASCIZ	'WRITE TAPE MARK RETRY At First Record, Failed To Set RIB (XST3)'
643	032024	124	115	113	T29RRN:	.ASCIZ	'TMK (XSTO) Failed To Set After READ REVERSE Into Tape Mark'
644	032117	127	162	151	TST29ID:	.ASCIZ	'Write Tape Mark Retry'

```

645 .EVEN
646 ;
647 ;
648 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
649 ;WRITE SUBSYSTEM MEMORY COMMAND
650 ;
651 ;
652 ;
    
```

```

653 032146 T29REST:
654 032146 SAVREG
655 032152 012701 026320 MOV #T29PACKET,R1 ;SAVE THE REGISTERS
656 032156 012721 140004 MOV #140004,(R1). ;START OF THE PACKET
657 032162 012721 026330 MOV #T29DATA,(R1). ;WRITE SUBSYSTEM MEM. WITH ACK, CVC=1
658 032166 005021 CLR (R1). ;ADDRESS OF CHARAISTICS DATA BLOCK
659 032170 012721 000012 MOV #10,(R1). ;EXTENDED ADDRESS
660 032174 012721 026342 MOV #T29BFR,(R1). ;SIZE OF DATA BLOCK IN BYTES
661 032200 005021 CLR (R1). ;ADDRESS OF MESSAGE BUFFER
662 032202 012721 000024 MOV #20,(R1). ;LENGTH OF MESSAGE BUFFER
663 032206 005021 CLR (R1).
664 032210 012711 000000 MOV #0,(R1) ;SELECT DRIVE ZERO (0)
665 032214 012702 000030 MOV #24,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
666 032220 012762 177777 026342 648: MOV #177777,T29BFR(R2) ;ALL ONES TO MESSAGE BUFFER
667 032226 005742 TST (R2) ;NEXT LOCATION
668 032230 020227 000000 CMP R2,#0 ;CHECK FOR END OF LOOP
669 032234 001371 BNE 648 ;KEEP GOING UNTIL DONE
670 032236 000207 RTS PC ;RETURN
671
672 032240 T29RT2:
    
```

```

673 032240          SAVREG          ;SAVE THE REGISTERS
674 032244 012701 026430          MOV      #T29PK2,R1          ;START OF THE PACKET
675 032250 012721 140006          MOV      #140006,(R1).        ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1.
676 032254 012721 026450          MOV      #T29BF2,(R1).        ;ADDRESS OF DATA BLOCK
677 032260 005021          CLR      (R1).          ;EXTENDED ADDRESS
678 032262 012721 000006          MOV      #6,(R1).          ;SIZE OF DATA BLOCK IN BYTES
679 032266 005021          CLR      (R1).
680 032270 012701 026450          MOV      #T29BF2,R1          ;POINT TO DATA SEL AREA
681 032274 005021          CLR      (R1).
682 032276 005011          CLR      (R1)
683 032300 000207          RTS      PC          ;RETURN
684 032302          T29RT3:
685 032302          SAVREG          ;SAVE THE REGISTERS
686 032306 012701 026440          MOV      #T29PK3,R1          ;START OF THE PACKET
687 032312 012721 000000          MOV      #0,(R1).        ;WRITE SUBSYSTEM MEM. WITH ACK.
688 032316 012721 000000          MOV      #0,(R1).        ;ADDRESS OF DATA BLOCK
689 032322 005021          CLR      (R1).          ;EXTENDED ADDRESS
690 032324 012711 000000          MOV      #0,(R1)          ;SIZE OF DATA BLOCK IN BYTES
691 032330 000207          RTS      PC          ;RETURN
692 032332          ENDTST
        032332          L10036:          TRAP      C#ETST
        032332          104401

```

.SBTTL TEST 2: SKIP TAPE MARKS

```

693          ;
694          ;
695          ;
696          ;THIS TEST VERIFIES PROPER OPERATION OF THE SKIP TAPE MARKS
697          ;FORWARD AND SKIP TAPE MARKS REVERSE COMMANDS. PROPER OPERATION
698          ;UNDER CONTROL OF ALL COMBINATIONS OF THE ENABLE SKIP TAPE MARKS
699          ;STOP (ESS) AND ENABLE TAPE MARKS STOP OFF BOT (ENB) BITS SPECIFIED
700          ;BY THE WRITE CHARACTERISTICS COMMAND. THE TEST CONSISTS OF THE
701          ;FOLLOWING SUBTESTS (FOR EACH SUBTEST, THE TAPE IS FIRST WRITTEN
702          ;WITH AN APPROPRIATE SERIES OF DATA RECORDS AND/OR TAPE MARKS
703          ;AND/OR DOUBLE TAPE MARKS.
704          ;
705          ;
706          ;THE TEST CONSISTS OF THE FOLLOWING 11 SUBTESTS
707          ;
708          ;
709          ;
710          ;
711          ;
712          ;          BGN1ST
713          ;
714          ;          T2::
715          ;          MOV      #EPT1,EPTSW          ;PRIMARY ERROR MESSAGE
716          ;          MOV      #TST30ID,R0          ;ASCII MESSAGE TO IDENTIFY TEST
717          ;          JSR      PC,TSTSETUP          ;DO INITIAL TEST SETUP
718          ;          MOV      #5,LOOPCNT          ;PERFORM 5 ITERATIONS
719          ;
720          ;
721          ;
722          ;TEST 2, SUBTEST 1
723          ;
724          ;
725          ;VERIFIES THAT A SKIP TAPE MARKS FORWARD COMMAND WITH
726          ;A TAPE MARK COUNT OF 1 OPERATES OPERATES PROPERLY. THE TAPE
727          ;IS FIRST REWOUND, THEN WRITTEN WITH SEVERAL "FILES";
728          ;EACH FILE CONSISTS OF A NUMBER OF DATA RECORDS
729          ;FOLLOWED BY A TAPE MARK. THE FINAL FILE IS
730          ;TERMINATED BY A DOUBLE TAPE MARK. EACH DATA RECORD

```



```

777 032512 005237 002214          INC    FATFLG          ;ERROR COUNT
781 032516 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
782 032520          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      032520 104456          TRAP   C$ERHRD
      032522 000312          .WORD 202
      032524 005052          .WORD WRTMSG
      032526 012114          .WORD SFIMSG
783 032530          23$:   CKLOOP          ;LOOP IF SELECTED
      032530 104406          TRAP   C$CLP1
784
785          ;*****
786          ;
787          ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
788          ;
789          ;*****
790
791 032532 004737 011074          JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
792 032536 103411          BCS   30$            ;BR. IF NO PROBLEM
793 032540 010004          MOV    R0,R4          ;GET PACKET ADDRESS
794 032542 016501 000002          MOV    TSSR(R5),R1    ;GET STATUS REGISTER
795 032546 005237 002214          INC    FATFLG          ;ERROR COUNT
799 032552          ERRHRD  ERRNO,T3ORWN,PKTSSR ;REWIND NOT ACCEPTED
      032552 104456          TRAP   C$ERHRD
      032554 000313          .WORD 203
      032556 040240          .WORD T3ORWN
      032560 012126          .WORD PKTSSR
800 032562          30$:   CKLOOP          ;LOOP IF SELECTED
      032562 104406          TRAP   C$CLP1
801
802          ;*****
803          ;
804          ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
805          ;
806          ;*****
807
808 032564 013701 036530          MOV    T30BFR+6,R1    ;PICK UP XSTO
809 032570 010102          MOV    R1,R2          ;SET UP EXPECTED
810 032572 052702 000002          BIS   @BIT1,R2        ;SET BOT BIT IN EXPECTED
811 032576 020102          CMP   R1,R2          ;DOES EXP = REC'D
812 032600 001406          BEQ   40$            ;BR. IF EQUAL (OK)
813 032602 005237 002214          INC    FATFLG          ;ERROR COUNT
817 032606          ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      032606 104456          TRAP   C$ERHRD
      032610 000314          .WORD 204
      032612 040041          .WORD T30BOT
      032614 015554          .WORD EXPREC
818 032616          40$:   CKLOOP          ;LOOP IF SELECTED
      032616 104406          TRAP   C$CLP1
819 032620 012737 000001 036654          MOV    @1.,T30FCN     ;SET "FILE" COUNTER AT 1 DECIMAL
820 032626 012703 000001          64$:  MOV    @1,R3      ;ONE RECORD PER "FILE"
821 032632 013737 003116 036622          65$:  MOV    FREE,T30WB ;SET UP PACKETS S WRITE B'FFER
822 032640 012737 003720 036626          MOV    @2000.,T30SZ  ;SET RECORD SIZE AT 2000 BYTES
823
824          ;*****
825          ;
826          ;WRITE DATA,ACK,CVC=1 COMMAND
827          ;

```

```

828 ;*****
829
830 032646 012737 140005 036620      MOV      #140005,T30PK3      ;WRITE DATA,ACK,CVC=1 COMMAND
831 032654 012704 036620              MOV      #T30PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
832 032660 013702 036654              MOV      T30FCN,R2        ;GET FILE COUNTER
833 032664 000302                      SWAB     R2                ;MOVE TO UPPER BYTE
834 032666 010301                      MOV      R3,R1            ;GET RECORD COUNTER
835 032670 060201                      ADD      R2,R1            ;FILE COUNTER IN UPPER, RECORD # LOW
836 032672 010177 150220              MOV      R1,#FREE         ;MOV TO OUT PUT BUFFER
837 032676 010465 000000              MOV      R4,TSDB(R5)     ;ISSUE COMMAND
838 032702 004737 016330              JSR      PC,WAITF        ;WAIT FOR SSR TO SET
839 032706 016501 000002              MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
840 032712 012702 000200              MOV      #SSR,R2        ;SET UP EXPECTED
841 032716 020102                      CMP      R1,R2           ;ARE THEY EQUAL
842 032720 001406                      BEQ      70$             ;BR, IF OK
843 032722 005237 002214              INC      FATFLG          ;ERROR COUNT
847 032726                      ERRHRD  ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    205
                                .WORD    T30WDD
                                .WORD    PKTSSR
848 032736                      70$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
849 032740 005203                      INC      R3              ;COUNT THE RECORD COUNTER DOWN
850 032742 020327 000021              CMP      R3,#21         ;AT 20 YET
851 032746 001331                      BNE     65$             ;BR, IF NOT AT 20 RECORDS WRITTEN
852
853 ;*****
854 ;
855 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
856 ;
857 ;*****
858
859 032750 012737 141011 036620      MOV      #141011,T30PK3  ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
860 032756 012704 036620              MOV      #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
861 032762 010465 000000              MOV      R4,TSDB(R5)    ;ISSUE COMMAND
862 032766 004737 016330              JSR      PC,WAITF       ;WAIT FOR SSR TO SET
863 032772 016501 000002              MOV      TSSR(R5),R1    ;PICK UP TSSR
864 032776 012702 000200              MOV      #SSR,R2        ;SET UP EXPECTED (SSR ONLY)
865 033002 020102                      CMP      R1,R2          ;WAS STATUS GOOD
866 033004 001406                      BEQ      160$           ;BR, IF TERMINATION WAS GOOD
867 033006 005237 002214              INC      FATFLG          ;ERROR COUNT
871 033012                      ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
                                TRAP      C$ERHRD
                                .WORD    206
                                .WORD    T30WDC
                                .WORD    PKTSSR
872 033022                      160$:  ;LOOP IF SELECTED
                                TRAP      C$CLP1
873 033024 005237 036654              INC      T30FCN          ;COUNT THE "FILE" COUNTER DOWN
874 033030 023727 036654 000006      CMP      T30FCN,#6      ;WRITE 5 FILE TO TAPE
875 033036 001273                      BNE     64$             ;BR, IF NOT AT 5 FILES WRITTEN
876
877 ;*****
878 ;
879 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
880 ;
    
```

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 06-FEB 84 18:04
 TEST 2: SKIP TAPE MARKS

SEQ 110

```

881 ;*****
882
883 033040 012737 141011 036620      MOV      #141011,T30PK3      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
884 033046 012704 036620              MOV      #T30PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
885 033052 010465 000000              MOV      R4,T30DB(R5)      ;ISSUE COMMAND
886 033056 004737 016330              JSR      PC,WAITF         ;WAIT FOR SSR TO SET
887 033062 016501 000002              MOV      T30SR(R5),R1     ;PICK UP T30SR
888 033066 012702 000200              MOV      #SSR,R2         ;SET UP EXPECTED (SSR ONLY)
889 033072 020102                      CMP      R1,R2           ;WAS STATUS GOOD
890 033074 001406                      BEQ      165$            ;BR, IF TERMINATION WAS GOOD
891 033076 005237 002214              INC      FATFLG          ;ERROR COUNT
895 033102                      ERRHRD  ERRNO,T30WDC,PKTSSR ;T30SR NOT CORRECT AFTER WRT TAPE M.
                                TRAP      C$ERRHRD
                                .WORD    207
                                .WORD    T30WDC
                                .WORD    PKTSSR
896 033112                      165$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406
897
898 ;*****
899 ;
900 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
901 ;
902 ;*****
903
904 033114 004737 011074              JSR      PC,REWIND        ;CALL TAPE REWIND COMMAND
905 033120 103411                      BCS     170$            ;BR, IF NO PROBLEM
906 033122 010004                      MOV      R0,R4           ;GET PACKET ADDRESS
907 033124 016501 000002              MOV      T30SR(R5),R1     ;GET STATUS REGISTER
908 033130 005237 002214              INC      FATFLG          ;ERROR COUNT
912 033134                      ERRHRD  ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERRHRD
                                .WORD    208
                                .WORD    T30RWN
                                .WORD    PKTSSR
913 033144                      170$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406
914
915 ;*****
916 ;
917 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
918 ;
919 ;*****
920
921 033146 013701 036530              MOV      T30BFR+6,R1     ;PICK UP XSTO
922 033152 010102                      MOV      R1,R2           ;SET UP EXPECTED
923 033154 052702 000002              BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
924 033160 020102                      CMP      R1,R2           ;DOES EXP = REC'D
925 033162 001406                      BEQ      180$            ;BR, IF EQUAL (OK)
926 033164 005237 002214              INC      FATFLG          ;ERROR COUNT
930 033170                      ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERRHRD
                                .WORD    209
                                .WORD    T30BOT
                                .WORD    EXPREC
931 033200                      180$:  CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406

```



```

932 033202 012703 036636          MOV      #T30IMV,R3          ;SET UP POINTER TO COMMAND TABLE
933 033206 013737 002174 036520  MOV      UNITN,T30DSW      ;SET UP UNIT NUMBER
934 033214 011337 036516          MOV      (R3),T30ETM      ;GET NEXT COMMAND
935 033220 012704 036500          MOV      #T30PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
936
937
938
939
940
941
942
943 033224 004737 010742          JSR      PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
944 033230 103407          BCS      188$            ;BR, IF COMMAND ISSUED OK
945 033232 005237 002214          INC      FATFLG          ;ERROR COUNT
949 033236 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
950 033240          ERRMRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
          TRAP      C$ERRMRD
          .WORD    210
          .WORD    WRTMSG
          .WORD    SFIMSG
951 033250          188$:  CKLOOP          ;LOOP IF SELECTED
          TRAP      C$CLP1
952
953
954
955
956
957
958
959 033252 012737 141010 036620  MOV      #141010,T30PK3   ;SKIP TAPE MARK,ACK,CVC-1 COMMAND
960 033260 012737 000001 036622  MOV      #1,T30RB        ;SET UP NUMBER TO SKIP
961 033266 012704 036620          MOV      #T30PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
962 033272 010465 000000          189$:  MOV      R4,TSD8(R5) ;ISSUE COMMAND
963 033276 012737 176750 036656  MOV      #65000.,T30DLY   ;SET UP DELAY COUNTER
964 033304 004737 016330          190$:  JSR      PC,WAITF    ;WAIT FOR SSR TO SET
965 033310 016501 000002          MOV      TSSR(R5),R1     ;PICK UP TSSR
966 033314 032701 0C0200          BIT      #SSR,R1        ;IS SSR SET YET
967 033320 001017          BNE      191$            ;BR, IF SSR IS SET
968 033322          DELAY  250            ;CALL DELAY ROUTINE
          MOV      #250,(PC)
          .WORD    0
          MOV      L$DLY,(PC)
          .WORD    0
          DEC      6(PC)
          BNE      . 4
          DEC      22(PC)
          BNE      . 20
969 033352 005337 036656          DEC      T30DLY          ;BUMP DELAY ROUTINE
970 033356 001352          BNE      190$            ;BR, IF MORE DELAY TO GO
971 033360 012702 000200          191$:  MOV      #SSR,R2    ;SET UP EXPECTED (SSR ONLY)
972 033364 020102          CMP      R1,R2          ;WAS STATUS GOOD
973 033366 001406          BEQ      192$            ;BR, IF TERMINATION WAS GOOD
974 033370 005237 002214          INC      FATFLG          ;ERROR COUNT
978 033374          ERRMRD  ERRNO,T30SKM,PKTSSR ;TSSR NOT CORRECT AFTER SKIP TAPE M.
          TRAP      C$ERRMRD
          .WORD    211
          .WORD    T30SKM
033374 104456
033376 000323
033400 037114
    
```

```

979 033402 012126          1928:  CKLOOP          ;LOOP IF SELECTED          .WORD  PKTSSR
      033404          ;
      033404 104406          ;
980
981          ;.....
982          ;
983          ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
984          ;
985          ;.....
986
987 033406 013701 036530      MOV      T308FR+6,R1      ;PICK UP XSTO
988 033412 010102          MOV      R1,R2           ;SET UP EXPECTED
989 033414 052702 100000      BIS      @BIT15,R2       ;SET TMK BIT IN EXPECTED
990 033420 020102          CMP      R1,R2           ;DOES EXP = REC'D
991 033422 001406          BEQ      1958           ;BR, IF EQUAL (OK)
992 033424 005237 002214      INC      FATFLG          ;ERROR COUNT
996 033430          ERRHRD  ERRNO,T30TMK,EXPREC ;TMK NOT SET AFTER WRT TAPE MARK
      033430 104456          TRAP      C8ERHRD
      033432 000324          .WORD    212
      033434 040514          .WORD    T30TMK
      033436 015554          .WORD    EXPREC
997 033440          1958:  CKLOOP          ;LOOP IF SELECTED          TRAP    C8CLP1
      033440 104406          ;
998 033442 012700 177777      MOV      @177777,R0      ;VALUE TO WRITTEN TO MEMORY
999 033446 004737 017502      JSR      PC,FILLMEM      ;FILL MEM WITH ALL ONES
1000 033452 013737 003116 036622 MOV      FREE,T30RB      ;STARTING READ BUFFER ADDRESS
1001
1002          ;.....
1003          ;
1004          ;READ FORWARD,ACK,CVC-1 COMMAND
1005          ;
1006          ;.....
1007
1008 033460 012737 140001 036620 MOV      @140001,T30PK3  ;READ FORWARD,ACK,CVC-1 COMMAND
1009 033466 012704 036620      MOV      @T30PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
1010 033472 012737 003720 036626 MOV      @2000,T30SZ      ;SET UP RECORD SIZE IN PACKET
1011 033500 010465 000000      MOV      R4,T30B(R5)     ;ISSUE COMMAND
1012 033504 004737 016330      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
1013 033510 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
1014 033514 012702 000200      MOV      @SSR,R2        ;SET UP EXPECTED
1015 033520 020102          CMP      R1,R2          ;ARE THEY EQUAL
1016 033522 001406          BEQ      2008           ;BR, IF OK
1017 033524 005237 002214      INC      FATFLG          ;ERROR COUNT
1021 033530          ERRHRD  ERRNO,T30RDF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      033530 104456          TRAP      C8ERHRD
      033532 000325          .WORD    213
      033534 037413          .WORD    T30RDF
      033536 012126          .WORD    PKTSSR
1022 033540          2008:  CKLOOP          ;LOOP IF SELECTED          TRAP    C8CLP1
      033540 104406          ;
1023 033542 017701 147350      MOV      @FREE,R1        ;FIRST LOC IN READ BUFFER
1024 033546 012702 177777      MOV      @177777,R2     ;EXPECTED IF NO DATA TRANS.
1025 033552 020102          CMP      R1,R2          ;DID ANY DATA GET TRANSFERRED
1026 033554 001006          BNE      2208           ;BR, IF NO DATA TRANS (GOOD)
1027 033556 005237 002214      INC      FATFLG          ;ERROR COUNT
1031 033562          ERRHRD  ERRNO,T30DTR,EXPREC ;DATA TRANSFERRED ON READ TAPE MARK
      033562 104456          TRAP      C8ERHRD
    
```

	033564	000326					.WORD	214
	033566	041070					.WORD	T30DTR
	033570	015554					.WORD	EXPREC
1032	033572		220\$:	CKLOOP		;LOOP IF SELECTED		
	033572	104406					TRAP	C\$CLP1
1033	033574	012702	001001	MOV	#1001,R2	;SET UP RECORD NUMBER EXPECTED (FILE 2)		
1034	033600	017701	147312	MOV	BFREE,R1	;GET INFO FROM BUFFER		
1035	033604	020201		CMP	R2,R1	;ARE THEY EQUAL		
1036	033606	001406		BEQ	228\$;BR, IF EQUAL (OK)		
1037	033610	005237	002214	INC	FATFLG	;ERROR COUNT		
1041	033614			ERRHRD	ERRNO,T30PTB,EXPREC	;RECORD POSITION WAS NOT CORRECT		
	033614	10445					TRAP	C\$ERHRD
	033616	000327					.WORD	215
	033620	037242					.WORD	T30PTB
	033622	015554					.WORD	EXPREC
1042	033624		228\$:	CKLOOP		;LOOP IF SELECTED		
	033624	104406					TRAP	C\$CLP1
1043								
1044								
1045								
1046								
1047								
1048								
1049								
1050	033626	004737	011074	JSR	PC,REWIND	;CALL TAPE REWIND COMMAND		
1051	033632	103411		BCS	230\$;BR, IF NO PROBLEM		
1052	033634	010004		MOV	R0,R4	;SAVE PACKET ADDRESS		
1053	033636	016501	000002	MOV	TSSR(R5),R1	;GET TSSR STATUS		
1054	033642	005237	002214	INC	FATFLG	;ERROR COUNT		
1058	033646			ERRHRD	ERRNO,T30RWN,PKTSSR	;REWIND NOT ACCEPTED		
	033646	104456					TRAP	C\$ERHRD
	033650	000330					.WORD	216
	033652	040240					.WORD	T30RWN
	033654	012126					.WORD	PKTSSR
1059	033656		230\$:	CKLOOP		;LOOP IF SELECTED		
	033656	104406					TRAP	C\$CLP1
1060								
1061								
1062								
1063								
1064								
1065								
1066								
1067	033660	013701	036530	MOV	T30BFR+6,R1	;PICK UP XSTO		
1068	033664	010102		MOV	R1,R2	;SET UP EXPECTED		
1069	033666	052702	000002	BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED		
1070	033672	020102		CMP	R1,R2	;DOES EXP = REC'D		
1071	033674	001406		BEQ	240\$;BR, IF EQUAL (OK)		
1072	033676	005237	002214	INC	FATFLG	;ERROR COUNT		
1076	033702			ERRHRD	ERRNO,T30BOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	033702	104456					TRAP	C\$ERHRD
	033704	000331					.WORD	217
	033706	040041					.WORD	T30BOT
	033710	015554					.WORD	EXPREC
1077	033712		240\$:	CKLOOP		;LOOP IF SELECTED		
	033712	104406					TRAP	C\$CLP1
1078	033714	005723		TST	(R3),	;POINT TO NEXT POSITION		


```

1122
1123 ;*****
1124 ;
1125 ;ISSUE WRITE CHARACTERISTICS COMMAND
1126 ;
1127 ;*****
1128
1129 034076 004737 010742          JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
1130 034102 103407                BCS      23$                ;BR, IF COMMAND ISSUED OK
1131 034104 005237 002214          INC      FATFLG            ;ERROR COUNT
1135 034110 010001                MOV      R0,R1             ;SAVE CONTENTS OF TSSR
1136 034112                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP      C$ERHRD
                                .WORD    219
                                .WORD    WRTMSG
                                .WORD    SFIMSG
1137 034122 104406          23$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
1138
1139 ;*****
1140 ;
1141 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1142 ;
1143 ;*****
1144
1145 034124 004737 011074          JSR      PC,REWIND         ;CALL TAPE REWIND COMMAND
1146 034130 103411                BCS      30$                ;BR, IF NO PROBLEM
1147 034132 010004                MOV      R0,R4             ;GET PACKET ADDRESS
1148 034134 016501 000002          MOV      TSSR(R5),R1      ;GET STATUS REGISTER
1149 034140 005237 002214          INC      FATFLG            ;ERROR COUNT
1153 034144                ERRHRD  ERRNO,T3ORWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    220
                                .WORD    T3ORWN
                                .WORD    PKTSSR
1154 034154 104406          30$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
1155
1156 ;*****
1157 ;
1158 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1159 ;
1160 ;*****
1161
1162 034156 013701 036530          MOV      T30BFR+6,R1      ;PICK UP XSTO
1163 034162 010102                MOV      R1,R2             ;SET UP EXPECTED
1164 034164 052702 000002          BIS      @BIT1,R2         ;SET BOT BIT IN EXPECTED
1165 034170 020102                CMP      R1,R2             ;DOES EXP = RECD
1166 034172 001406                BEQ      40$                ;BR, IF EQUAL (OK)
1167 034174 005237 002214          INC      FATFLG            ;ERROR COUNT
1171 034200                ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    221
                                .WORD    T30BOT
                                .WORD    EXPREC
1172 034210 104406          40$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
    
```

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 06-FEB-84 18:04
 TEST 2: SKIP TAPE MARKS

SEQ 116

```

1173 034212 012737 000001 036654      MOV      #1.,T30FCN      ;SET "FILE" COUNTER AT 1 DECIMAL
1174 034220 012703 000001      64$:    MOV      #1,R3      ;ONE RECORD PER "FILE"
1175 034224 013737 003116 036622      65$:    MOV      FREE,T30WB   ;SET UP PACKETS'S WRITE BUFFER
1176 034232 012737 000024 036626      MOV      #20.,T30SZ     ;SET RECORD SIZE AT 2000 BYTES
1177
1178      ;*****
1179      ;
1180      ;WRITE DATA,ACK,CVC=1 COMMAND
1181      ;
1182      ;*****
1183
1184 034240 012737 140005 036620      MOV      #140005,T30PK3  ;WRITE DATA,ACK,CVC=1 COMMAND
1185 034246 012704 036620      MOV      #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
1186 034252 013702 036654      MOV      T30FCN,R2     ;GET FILE COUNTER
1187 034256 000302      C'WAB  R2              ;MOVE TO UPPER BYTE
1188 034260 010301      MOV      R3,R1         ;GET RECORD COUNTER
1189 034262 060201      ADD      R2,R1         ;FILE COUNTER IN UPPER, RECORD # LOW
1190 034264 010177 146626      MOV      R1,#FREE      ;MOV TO OUT PUT BUFFER
1191 034270 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
1192 034274 004737 016330      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
1193 034300 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
1194 034304 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
1195 034310 020102      CMP      R1,R2         ;ARE THEY EQUAL
1196 034312 001406      BEQ      70$          ;BR, IF OK
1197 034314 005237 002214      INC      FATFLG        ;ERROR COUNT
1201 034320      ERRHRD  ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      034320 104456      TRAP      C$ERHRD
      034322 000336      .WORD    222
      034324 037170      .WORD    T30WDD
      034326 012126      .WORD    PKTSSR
1202 034330      70$:    CKLOOP          ;LOOP IF SELECTED
      034330 104406      TRAP      C$CLP1
1203 034332 005203      INC      R3            ;COUNT THE RECORD COUNTER DOWN
1204 034334 020327 000021      CMP      R3,#21       ;AT 20 YET
1205 034340 001331      BNE      65$          ;BR, IF NOT AT 20 RECORDS WRITTEN
1206
1207      ;*****
1208      ;
1209      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1210      ;
1211      ;*****
1212
1213 034342 012737 141011 036620      MOV      #141011,T30PK3 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1214 034350 012704 036620      MOV      #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
1215 034354 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
1216 034360 004737 016330      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
1217 034364 016501 000002      MOV      TSSR(R5),R1   ;PICK UP TSSR
1218 034370 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED (SSR ONLY)
1219 034374 020102      CMP      R1,R2         ;WAS STATUS GOOD
1220 034376 001406      BEQ      160$         ;BR, IF TERMINATION WAS GOOD
1221 034400 005237 002214      INC      FATFLG        ;ERROR COUNT
1225 034404      ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
      034404 104456      TRAP      C$ERHRD
      034406 000337      .WORD    223
      034410 040362      .WORD    T30WDC
      034412 012126      .WORD    PKTSSR
1226 034414      160$:  CKLOOP          ;LOOP IF SELECTED

```

```

1227 034414 104406                                TRAP      C$CLP1
1228 034416 005237 036654                        INC      T30FCN                ;COUNT THE "FILE" COUNTER DOWN
1229 034422 023727 036654 000031                CMP      T30FCN,#25.          ;WRITE 25 FILES TO TAPE
1230 034430 001273                                BNE      64$                  ;BR. IF NOT AT 25 FILES WRITTEN
1231
1232 ;*****
1233 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1234 ;
1235 ;*****
1236
1237 034432 012737 141011 036620                MOV      #141011,T30PK3      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1238 034440 012704 036620                        MOV      #T30PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
1239 034444 010465 000000                        MOV      R4,TSD8(R5)         ;ISSUE COMMAND
1240 034450 004737 016330                        JSR      PC,WAITF            ;WAIT FOR SSR TO SET
1241 034454 016501 000002                        MOV      TSSR(R5),R1         ;PICK UP TSSR
1242 034460 012702 000200                        MOV      #SSR,R2             ;SET UP EXPECTED (SSR ONLY)
1243 034464 020102                                CMP      R1,R2               ;WAS STATUS GOOD
1244 034466 001406                        BEQ      165$                 ;BR. IF TERMINATION WAS GOOD
1245 034470 005237 002214                        INC      FATFLG              ;ERROR COUNT
1249 034474                                ERRHRD  ERRNO,T30WDC,PKTSSR  ;TSSR NOT CORRECT AFTER WRT TAPE M.
                                TRAP      C$ERRHRD
                                .WORD      224
                                .WORD      T30WDC
                                .WORD      PKTSSR
1250 034504                                165$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
1251
1252 ;*****
1253 ;
1254 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1255 ;
1256 ;*****
1257
1258 034506 004737 011074                        JSR      PC,REWIND           ;CALL TAPE REWIND COMMAND
1259 034512 103411                        BCS      170$                 ;BR. IF NO PROBLEM
1260 034514 010004                        MOV      R0,R4               ;GET PACKET ADDRESS
1261 034516 016501 000002                        MOV      TSSR(R5),R1         ;GET STATUS REGISTER
1262 034522 005237 002214                        INC      FA FLG              ;ERROR COUNT
1266 034526                                ERRHRD  ERRNO,T30RWN,PKTSSR  ;REWIND NOT ACCEPTED
                                TRAP      C$ERRHRD
                                .WORD      225
                                .WORD      T30RWN
                                .WORD      PKTSSR
1267 034536                                170$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
1268
1269 ;*****
1270 ;
1271 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1272 ;
1273 ;*****
1274
1275 034540 013701 036530                        MOV      T30BFR+6,R1         ;PICK UP XSTO
1276 034544 010102                        MOV      R1,R2               ;SET UP EXPECTED
1277 034546 05270 000002                        BIS      #BIT1,R2            ;SET BOT BIT IN EXPECTED
1278 034552 020102                        CMP      R1,R2               ;DOFS EXP = REC'D
    
```



```

1324 034752 005337 036656          DEC      T30DLY          ;BUMP DELAY ROUTINE
1325 034756 001352                   BNE      1901          ;BR, IF MORE DELAY TO GO
1326 034760 2702 000200          1911:  MOV      @SSR,R2          ;SET UP EXPECTED (SSR ONLY)
1327 034764 020102                   CMP      R1,R2          ;WAS STATUS GOOD
1328 034766 001406                   BEQ      1921          ;BR, IF TERMINATION WAS GOOD
1329 034770 005237 002214          INC      FATFLG          ;ERROR COUNT
1333 034774                   ERRHRD  ERRNO,T30SKM,PKTSSR ;TSSR NOT CORRECT AFTER SKIP TAPE M.
                                TRAP      C1ERHRD
                                .WORD    228
                                .WORD    T30SKM
                                .WORD    PKTSSR
    034774 104456
    034776 000344
    035000 037114
    035002 012126
1334 035004          1921:  CKLOOP          ;LOOP IF SELECTED
    035004 104406          TRAP      C1CLP1
1335
1336          ;*****
1337          ;
1338          ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1339          ;
1340          ;*****
1341
1342 035006 013701 036530          MOV      T30BFR+6,R1          ;PICK UP XSTO
1343 035012 010102                   MOV      R1,R2          ;SET UP EXPECTED
1344 035014 052702 100000          BIS      @BIT15,R2          ;SET TMK BIT IN EXPECTED
1345 035020 020102                   CMP      R1,R2          ;DOES EXP = REC'D
1346 035022 001406                   BEQ      1951          ;BR, IF EQUAL (OK)
1347 035024 005237 002214          INC      FATFLG          ;ERROR COUNT
1351 035030          ERRHRD  ERRNO,T30TMK,EXPREC ;TMK NOT SET AFTER WRT TAPE MARK
                                TRAP      C1ERHRD
                                .WORD    229
                                .WORD    T30TMK
                                .WORD    EXPREC
    035030 104456
    035032 000345
    035034 040514
    035036 015554
1352 035040          1951:  CKLOOP          ;LOOP IF SELECTED
    035040 104406          TRAP      C1CLP1
1353 035042 012700 177777          MOV      @177777,R0          ;VALUE TO WRITTEN TO MEMORY
1354 035046 004737 017502          JSR      PC,FILLMEM          ;FILL MEM WITH ALL ONES
1355 035052 013737 003116 036622  MOV      FREE,T30RB          ;STARTING READ BUFFER ADDRESS
1356
1357          ;*****
1358          ;
1359          ;READ FORWARD,ACK,CVC=1 COMMAND
1360          ;
1361          ;*****
1362
1363 035060 012737 140001 036620  MOV      @140001,T30PK3          ;READ FORWARD,ACK,CVC=1 COMMAND
1364 035066 012704 036620          MOV      @T30PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
1365 035072 012737 000024 036626  MOV      @20.,T30SZ          ;SET UP RECORD SIZE IN PACKET
1366 035100 010465 000000          MOV      R4,T30B(R5)          ;ISSUE COMMAND
1367 035104 004737 016330          JSR      PC,WAITF          ;WAIT FOR SSR TO SET
1368 035110 016501 000002          MOV      TSSR(R5),R1          ;GET TSSR CONTENTS
1369 035114 012702 000200          MOV      @SSR,R2          ;SET UP EXPECTED
1370 035120 020102                   CMP      R1,R2          ;ARE THEY EQUAL
1371 035122 001406                   BEQ      2001          ;BR, IF OK
1372 035124 005237 002214          INC      FATFLG          ;ERROR COUNT
1376 035130          ERRHRD  ERRNO,T30RDF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C1ERHRD
                                .WORD    230
                                .WORD    T30RDF
    035130 104456
    035132 000346
    035134 037413

```

```

035136 012126
1377 035140 2004: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
035140 104406 ;FIRST LOC IN READ BUFFER TRAP C$CLP1
1378 035142 017701 145750 MOV $FREE,R1 ;EXPECTED IF NO DATA TRANS.
1379 035146 012702 177777 MOV $177777,R2 ;DID ANY DATA GET TRANSFERRED
1380 035152 020102 CMP R1,R2 ;BR, IF NO DATA TRANS (GOOD)
1381 035154 001006 BNE 2204 ;ERROR COUNT
1382 035156 005237 002214 INC FATFLG ;DATA TRANSFERRED ON READ TAPE MARK
1386 035162 ERRHRD LRRNO,T30DTR,EXPREC ;TRAP C$ERHRD
035162 104456 .WORD 231
035164 000347 .WORD T30DTR
035166 041070 .WORD EXPREC
035170 015554
1387 035172 2204: CKLOOP ;LOOP IF SELECTED .WORD C$CLP1
035172 104406 ;TRAP C$CLP1
1388 035174 013702 036654 MOV T30FCN,R2 ;GET NUMBER OF SKIPS
1389 035200 005202 INC R2 ;SET TO EXPECT FILE VALUE
1390 035202 000302 SWAB R2 ;SWAP BY HALVES
1391 035204 052702 000001 BIS $BIT0,R2 ;SET FOR WORD #1
1392 035210 017701 145702 MOV $FREE,R1 ;GET INFO FROM BUFFER
1393 035214 020201 CMP R2,R1 ;ARE THEY EQUAL
1394 035216 001406 BEQ 2284 ;BR, IF EQUAL (OK)
1395 035220 005237 002214 INC FATFLG ;ERROR COUNT
1399 035224 ERRHRD ERRNO,T30PTB,EXPREC ;RECORD POSITION WAS NOT CORRECT
035224 104456 TRAP C$ERHRD
035226 000350 .WORD 232
035230 037242 .WORD T30PTB
035232 015554 .WORD EXPREC
1400 035234 2284: CKLOOP ;LOOP IF SELECTED .WORD C$CLP1
035234 104406 ;TRAP C$CLP1
1401
1402 ;*****
1403 ;
1404 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1405 ;
1406 ;*****
1407
1408 035236 004737 011074 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
1409 035242 103411 BCS 2304 ;BR, IF NO PROBLEM
1410 035244 010004 MOV R0,R4 ;SAVE PACKET ADDRESS
1411 035246 016501 000002 MOV TSSR(R5),R1 ;GET TSSR STATUS
1412 035252 005237 002214 INC FATFLG ;ERROR COUNT
1416 035256 ERRHRD ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
035256 104456 TRAP C$ERHRD
035260 000351 .WORD 233
035262 040210 .WORD T30RWN
035264 012126 .WORD PKTSSR
1417 035266 2304: CKLOOP ;LOOP IF SELECTED .WORD C$CLP1
035266 104406 ;TRAP C$CLP1
1418
1419 ;*****
1420 ;
1421 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1422 ;
1423 ;*****
1424
1425 035270 013701 036530 MOV T30BFR+6,R1 ;PICK UP XSTO

```



```

035440 000000 .WORD 0
035442 013727 002116 MOV L$DLY,(PC)+
035446 000000 .WORD 0
035450 005367 177772 DEC -6(PC)
035454 001375 BNE .-4
035456 005367 177756 DEC 22(PC)
035462 001367 BNE .-20
1475 035464 005337 036656 DEC T30DLY ;BUMP COUNTER
1476 035470 001356 BNE 10$ ;BR, IF MORE COUNTING TO DO
1477 035472 005237 002214 INC FATFLG ;ERROR COUNT
1481 035476 010001 MOV R0,R1 ;CONTENTS OF TSSR REGISTER
1482 035500 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
035500 104455 TRAP C$ERDF
035502 000353 .WORD 235
035504 003646 .WORD SFIERR
035506 012114 .WORD SFIMSG
1483 035510
1484 035510 013737 002174 036520 20$: MOV UNITN,T30DSW ;SET UP UNIT NUMBER
1485 035516 012704 036500 MOV @T30PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
1486
1487 ;*****
1488 ;
1489 ;ISSUE WRITE CHARACTERISTICS COMMAND
1490 ;
1491 ;*****
1492
1493 035522 004737 010742 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
1494 035526 103407 BCS 23$ ;BR, IF COMMAND ISSUED OK
1495 035530 005237 002214 INC FATFLG ;ERROR COUNT
1499 035534 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
1500 035536 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
035536 104456 TRAP C$ERHRD
035540 000354 .WORD 236
035542 005052 .WORD WRTMSG
035544 012114 .WORD SFIMSG
1501 035546 23$: CKLOOP ;LOOP IF SELECTED
035546 104406 TRAP C$CLP1
1502
1503 ;*****
1504 ;
1505 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1506 ;
1507 ;*****
1508
1509 035550 004737 011074 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
1510 035554 103411 BCS 30$ ;BR, IF NO PROBLEM
1511 035556 010004 MOV R0,R4 ;GET PACKET ADDRESS
1512 035560 016501 000002 MOV TSSR(R5),R1 ;GET STATUS REGISTER
1513 035564 005237 002214 INC FATFLG ;ERROR COUNT
1517 035570 ERRHRD ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
035570 104456 TRAP C$ERHRD
035572 000355 .WORD 237
035574 040240 .WORD T30RWN
035576 012126 .WORD PKTSSR
1518 035600 30$: CKLOOP ;LOOP IF SELECTED
035600 104406 TRAP C$CLP1
1519
    
```

G10

TEST 1: HARDWARE TEST 1 & TEST MACRO M1113 06 FEB 84 18:04
 TEST 2: SKIP TAPE MARKS

SEQ 123

```

1520 ;*****
1521 ;
1522 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1523 ;
1524 ;*****
1525
1526 035602 013701 036530          MOV     T30BFR+6,R1      ;PICK UP XSTO
1527 035606 010102                MOV     R1,R2           ;SET UP EXPECTED
1528 035610 052702 000002        BIS     @BIT1,R2        ;SET BOT BIT IN EXPECTED
1529 035614 020102                CMP     R1,R2          ;DOES EXP = REC'D
1530 035616 001406                BEQ     40$            ;BR, IF EQUAL (OK)
1531 035620 005237 002214        INC     FATFLG         ;ERROR COUNT
1535 035624                ERRMRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP     C$ERRMRD
                                .WORD    238
                                .WORD    T30BOT
                                .WORD    EXPREC
1536 035634                40$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP1
1537 035636 012737 000001 036622  MOV     @1,T30WB       ;SET # OF TM TO SKIP
1538
1539 ;*****
1540 ;
1541 ;SKIP TAPE MARK REVERSE,ACK,CVC=1 COMMAND
1542 ;
1543 ;*****
1544
1545 035644 012737 141410 036620    MOV     @141410,T30PK3 ;SKIP TAPE MARK REVERSE,ACK,CVC=1 CMD
1546 035652 012704 036620          MOV     @T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
1547 035656 010465 000000          MOV     R4,TSD8(R5)   ;ISSUE COMMAND
1548 035662 004737 016330          JSR     PC,WAITF      ;WAIT FOR SSR TO SET
1549 035666 016501 000002          MOV     TSSR(R5),R1   ;GET TSSR CONTENTS
1550 035672 012702 100206          MOV     @SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
1551 035676 020102                CMP     R1,R2          ;ARE THEY EQUAL
1552 035700 001406                BEQ     70$            ;BR, IF OK
1553 035702 005237 002214        INC     FATFLG         ;ERROR COUNT
1557 035706                ERRMRD  ERRNO,T30IBT,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP     C$ERRMRD
                                .WORD    239
                                .WORD    T30IBT
                                .WORD    PKTSSR
1558 035716                70$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP1
1559
1560 ;*****
1561 ;
1562 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1563 ;
1564 ;*****
1565
1566 035720 013701 036530          MOV     T30BFR+6,R1   ;PICK UP XSTO
1567 035724 010102                MOV     R1,R2         ;SET UP EXPECTED
1568 035726 052702 002000        BIS     @BIT10,R2     ;SET NEF BIT IN EXPECTED
1569 035732 020102                CMP     R1,R2         ;DOES EXP = REC'D
1570 035734 001406                BEQ     180$          ;BR, IF EQUAL (OK)
1571 035736 005237 002214        INC     FATFLG         ;ERROR COUNT
1575 035742                ERRMRD  ERRNO,T30NEF,EXPREC ;TAPE NOT AT NEF
    
```



```

1615 036104
1616 036104 013737 002174 036520 208:  MOV    UNITN,T30DSW      ;SET UP UNIT NUMBER
1617 036112 012704 036500      MOV    @T30PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
1618
1619 ;.....
1620 ;
1621 ;ISSUE WRITE CHARACTERISTICS COMMAND
1622 ;
1623 ;.....
1624
1625 036116 004737 010742      JSR    PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
1626 036122 103407          BCS    238              ;BR. IF COMMAND ISSUED OK
1627 036124 005237 002214      INC    FATFLG           ;ERROR COUNT
1631 036130 010001          MOV    R0,R1            ;SAVE CONTENTS OF TSSR
1632 036132          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP    C:ERHRD
                                .WORD   242
                                .WORD   WRTMSG
                                .WORD   SFIMSG
1633 036142 231:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C:CLP1
1634
1635 ;.....
1636 ;
1637 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1638 ;
1639 ;.....
1640
1641 036144 004737 011074      JSR    PC,REWIND       ;CALL TAPE REWIND COMMAND
1642 036150 103411          BCS    308              ;BR. IF NO PROBLEM
1643 036152 010004          MOV    R0,R4            ;GET PACKET ADDRESS
1644 036154 016501 000002      MOV    TSSR(R5),R1     ;GET STATUS REGISTER
1645 036160 005237 002214      INC    FATFLG           ;ERROR COUNT
1649 036164          ERRHRD  ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C:ERHRD
                                .WORD   243
                                .WORD   T30RWN
                                .WORD   PKTSSR
1650 036174 308:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C:CLP1
1651
1652 ;.....
1653 ;
1654 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1655 ;
1656 ;.....
1657
1658 036176 013701 036530      MOV    T30BFR+6,R1     ;PICK UP XSTO
1659 036202 010102          MOV    R1,R2            ;SET UP EXPECTED
1660 036204 052702 000002      BIS    @BIT1,R2        ;SET BOT BIT IN EXPECTED
1661 036210 020102          CMP    R1,R2            ;DOES EXP = REC D
1662 036212 001406          BEQ    408              ;BR. IF EQUAL (OK)
1663 036214 005237 002214      INC    FATFLG           ;ERROR COUNT
1667 036220          ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C:ERHRD
                                .WORD   244
                                .WORD   T30BOT
1668 036220 104456
1669 036222 000364
1670 036224 040041
    
```

```

1668 036226 01554
036230 104406
1669 036232 013737 003116 036622
1670 036240 012737 000400 036626
1671
1672
1673
1674
1675
1676
1677
1678 036246 012737 140005 036620
1679 036254 012704 036620
1680 036260 010465 000000
1681 036264 004737 016330
1682 036270 016501 000002
1683 036274 012702 000200
1684 036300 020102
1685 036302 001406
1686 036304 005237 002214
1690 036310
036310 104456
036312 000365
036314 037170
036316 012126
1691 036320
036320 104406
1692
1693
1694
1695
1696
1697
1698
1699 036322 012737 000001 036622
1700 036330 012737 141410 036620
1701 036336 012704 036620
1702 036342 010465 000000
1703 036346 004737 016330
1704 036352 016501 000002
1705 036356 012702 100204
1706 036362 020102
1707 036364 001406
1708 036366 005237 002214
1712 036372
036372 104456
036374 000366
036376 036660
036400 012126
1713 036402
036402 104406
1714
1715
1716
1717
1718

40$: CKLOOP ;LOOP IF SELECTED .WORD EXPREC
TRAP C$CLP1
MOV FREE,T30WB ;SET UP GOOD WRITE BUFFER
MOV #256.,T30SZ ;SET UP SIZE
;*****
;WRITE DATA,ACK,CVC=1 COMMAND
;*****
MOV #140005,T30PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
MOV #T30PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
MOV R4,TSDB(R5) ;ISSUE COMMAND
JSR PC,WAITF ;WAIT FOR SSR TO SET
MOV TSSR(R5),R1 ;GET TSSR CONTENTS
MOV #SSR,R2 ;SET UP EXPECTED
CMP R1,R2 ;ARE THEY EQUAL
BEQ 70$ ;BR, IF OK
INC FATFLG ;ERROR COUNT
ERRHRD ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
TRAP C$ERHRD
.WORD 245
.WORD T30WDD
.WORD PKTSSR
70$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
;*****
;SKIP TAPE MARK REVERSE,ACK,CVC=1 COMMAND
;*****
MOV #1,T30WB ;# OF TM TO SKIP
MOV #141410,T30PK3 ;SKIP TAPE MARK REVERSE,ACK,CVC=1 CMD
MOV #T30PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
MOV R4,TSDB(R5) ;ISSUE COMMAND
JSR PC,WAITF ;WAIT FOR SSR TO SET
MOV TSSR(R5),R1 ;PICK UP TSSR
MOV #SSR!BIT2!SC,R2 ;SET UP EXPECTED (SSR AND SC ONLY)
CMP R1,R2 ;WAS STATUS GOOD
BEQ 160$ ;BR, IF TERMINATION WAS GOOD
INC FATFLG ;ERROR COUNT
ERRHRD ERRNO,T30IBU,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
TRAP C$ERHRD
.WORD 246
.WORD T30IBU
.WORD PKTSSR
160$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
;*****
;GET EXTENDED STATUS REGISTER ZERO (XST3) FROM MESSAGE BUFFER
;

```


TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB-84 18:04
 TEST 2: SKIP TAPE MARKS

SEQ 128

1777	036620			T30PK3:					
1778	036620	100205			.WORD	100205			;REREAD COMMAND, IE AND ACK
1779	036622			T30RB:					
1780	036622	003116		T30WB:	.WORD	FREE			;ADDRESS OF WRITE BUFFER
1781	036624	000000			.WORD	0			
1782	036626	000000		T30SZ:	.WORD	0			;SIZE OF BUFFER (EXTENT)
1783					.EVEN				
1784				:					
1785				:					
1786				:					
1787	036630			T30BF2:					
1788	036630	010		T30BS0:	.BYTE	10			;BSELO AREA
1789	036631	200		T30BS1:	.BYTE	200			;BSEL1 AREA
1790	036632	000000		T30S2:	.WORD	0			;SEL 2 AREA
1791	036634	000000		T30S3:	.WORD	0			;DATA AREA
1792				:					
1793				:					
1794					.EVEN				
1795					;TAPE MOTION PACKET COMMAND VALUES				
1796									
1797	036636			T30IMV:					
1798	036636			T30RN:					
1799	036636	000000			.WORD	000000			;NEITHER EWB NOR ESS
1800	036640	000100			.WORD	000100			;EWB SET
1801	036642	000200			.WORD	000200			;ESS SET
1802	036644	000300			.WORD	000300			;BOTH EWB AND ESS SE
1803	036646	177777			.WORD	177777			;END OF DATA
1804									
1805				:					
1806	036650	000000		T30CNT:	.WORD	0			;TAPE TIMER COUNTER STORAGE AREA
1807	036652	000000		T30CNU:	.WORD	0			;TAPE TIMER COUNTER STORAGE AREA
1808	036654	000000		T30FCN:	.WORD	0			;FILE NUMBER COUNTER
1809	036656	000000		T30DLY:	.WORD	0			;DELAY COUNTER STORAGE
1810				:					
1811				:					
1812					;LOCAL TEXT MESSAGES FOR TEST				
1813				:					
1814	036660	124	123	123	T30IBU:	.ASCIZ			'TSSR Incorrect After SKIP TAPE MARK REVERSE Into BOT'
1815	036745	122	111	102	T30RIB:	.ASCIZ			'RIB Bit (XST3) Failed To Set After Reverse Into BOT'
1816	037031	124	123	123	T30IBT:	.ASCIZ			'TSSR Incorrect After SKIP TAPE MARK REVERSE At BOT'
1817	037114	124	123	123	T30SKM:	.ASCIZ			'TSSR Incorrect After SKIP TAPE MARK Command'
1818	037170	124	123	123	T30WDD:	.ASCIZ			'TSSR Not Correct After WRITE DATA Command'
1819	037242	124	141	160	T30PTB:	.ASCIZ			'Tape Not Positioned On Correct Record After READ REVERSE'
1820	037333	124	141	160	T30TPB:	.ASCIZ			'Tape Not Positioned On Second File First Record'
1821	037413	124	123	123	T30RDF:	.ASCIZ			'TSSR Incorrect After READ FORWARD Into "File"'
1822	037471	124	123	123	T30RDG:	.ASCIZ			'TSSR Incorrect After SPACE Command Into TAPE MARK'
1823	037553	124	123	123	T30WDF:	.ASCIZ			'TSSR Not Correct After Illegal Mode Bits Set'
1824	037630	111	154	154	T30LOQ:	.ASCIZ			'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
1825	037711	127	102	111	T30SSR:	.ASCIZ			'WRITE MISCELLANEOUS Command Not Accepted'
1826	037762	124	123	123	T30WDE:	.ASCIZ			'TSSR Not Correct After SKIP TAPE MARKS, At BOT'
1827	040041	124	141	160	T30BOT:	.ASCIZ			'Tape Not At BOT After REWIND Command'
1828	040106	124	123	123	T30TM:	.ASCIZ			'TSSR Not Correct After SPACE FORWARD Command'
1829	040163	124	123	123	T30TM2:	.ASCIZ			'TSSR Not Correct After SPACE REVERSE Command'
1830	040240	122	145	167	T30RWN:	.ASCIZ			'Rewind (POSITION) Command Not Accepted'
1831	040307	104	162	151	T30OFL:	.ASCIZ			'Drive 7 Select Failed To Set "OFL" In TSSR'
1832	040362	124	123	123	T30WDC:	.ASCIZ			'TSSR Not Correct After WRITE TAPE MARK Command'
1833	040441	103	126	103	T30VCK:	.ASCIZ			'CVC Set, Didr.'t Reset VCK In Message Buffer'

```

1834 040514      124      115      113  T30TMK: .ASCIZ 'TMK Not Set After WRITE TAPE MARK (RETRY) Command'
1835 040576      123      113      111  T30NEF: .ASCIZ 'SKIP TAPE MARKS, At BCT, Failed To Set NEF Bit'
1836 040655      124      115      113  T30RRM: .ASCIZ 'TMK Not Set After READ REVERSE Into TAPE MARK'
1837 040733      124      115      113  T30RRN: .ASCIZ 'TMK Not Set After SPACE REVERSE Into TAPE MARK'
1838 041012      124      115      113  T30RRP: .ASCIZ 'TMK Not Set After READ FORWARD Into TAPE MARK'
1839 041070      116      117      040  T30DTR: .ASCIZ 'NO Data Transferred On READ FORWARD'
1840 041134      104      141      164  T30DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
1841 041231      123      153      151  TST30ID: .ASCIZ 'Skip Tape Marks'
1842                                     .EVEN
1843                                     ;+
1844                                     ;
1845                                     ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1846                                     ;WRITE SUBSYSTEM MEMORY COMMAND
1847                                     ;
1848                                     ;-
1849
1850 041252      T30REST:
1851 041252      SAVREG                                     ;SAVE THE REGISTERS
1852 041252      MOV                                     ;START OF THE PACKET
1853 041262      MOV #T30PACKET,R1                                     ;WRITE SUBSYSTEM MEM. WITH ACK.
1854 041266      MOV #100004,(R1)+                                     ;ADDRESS OF CHARAISTICS DATA BLOCK
1855 041272      MOV #T30DATA,(R1)+                                     ;EXTENDED ADDRESS
1856 041274      CLR (R1)+                                     ;SIZE OF DATA BLOCK IN BYTES
1857 041300      MOV #10.,(R1)+                                     ;ADDRESS OF MESSAGE BUFFER
1858 041304      CLR (R1)+
1859 041306      MOV #20.,(R1)+                                     ;LENGTH OF MESSAGE BUFFER
1860 041312      CLR (R1)+
1861 041314      MOV #0,(R1)                                     ;SELECT DRIVE ZERO
1862 041320      MOV #24.,R2                                     ;NUMBER OF LOCATIONS TO BE CLEARED
1863 041324      MOV #177777,T30BFR(R2)                             ;ALL ONES TO MESSAGE BUFFER
1864 041332      TST -(R2)                                         ;NEXT LOCATION
1865 041334      CMP #0.,R2                                         ;CHECK R2 FOR DONE
1866 041340      BNE 64$                                           ;KEEP GOING UNTIL DONE
1867 041342      RTS PC                                           ;RETURN
1868
1869 041344      T30RT2:
1870 041344      SAVREG                                     ;SAVE THE REGISTERS
1871 041350      MOV #T30PK2,R1                                     ;START OF THE PACKET
1872 041354      MOV #100006,(R1)+                                     ;WRITE SUBSYSTEM MEM. WITH ACK.
1873 041360      MOV #T30BF2,(R1)+                                     ;ADDRESS OF DATA BLOCK
1874 041364      CLR (R1)+                                     ;EXTENDED ADDRESS
1875 041366      MOV #6.,(R1)+                                     ;SIZE OF DATA BLOCK IN BYTES
1876 041372      CLR (R1)+
1877 041374      MOV #T30BF2,R1                                     ;POINT TO DATA SEL AREA
1878 041400      CLR (R1)+
1879 041402      CLR (R1)
1880 041404      RTS PC                                           ;RETURN
1881 041406      T30RT3:
1882 041406      SAVREG                                     ;SAVE REGISTERS
1883 041412      MOV #T30PK3,R1                                     ;SET UP POINTER ADDRESS
1884 041416      CLR (R1)+                                     ;COMMAND SPACE
1885 041420      CLR (R1)+                                     ;ADDRESS OF DATA BLOCK
1886 041422      CLR (R1)+                                     ;EXTENDED ADDRESS
1887 041424      CLR (R1)                                         ;SIZE OF DATA TRANSFER BLOCK
1888 041426      RTS PC                                           ;RETURN
1889 041430      ENDTST
041430

```

L10043:

Line	Address	Offset	Label	Code	Comment	Trap	Trap Value
1989	041742	012737	000144	043316	MOV #100, T31SZ		
1990	041750	010465	000000		MOV R4, TSDB(R5)		
1991	041754	004737	016330		JSR PC, WAITF		
1992	041760	016501	000002		MOV TSSR(R5), R1		
1993	041764	012702	000200		MOV #SSR, R2		
1994	041770	020102			CMP R1, R2		
1995	041772	001406			BEQ 80:		
1996	041774	005237	002214		INC FATFLG		
2000	042000				ERRHRD ERRNO, T31WDC, PKTSSR		
	042000	104456				TRAP	C:ERHRD
	042002	000461				.WORD	305
	042004	045230				.WORD	T31WDC
	042006	012126				.WORD	PKTSSR
2001	042010			80:	CKLOOP		
	042010	104406					
2002	042012	004737	011074		JSR PC, REWIND		
2003	042016	103407			BCS 230:		
2004	042020	010001			MOV R0, R1		
2005	042022	005237	002214		INC FATFLG		
2009	042026				ERRHRD ERRNO, T31RWN, EXPREC		
	042026	104456				TRAP	C:ERHRD
	042030	000462				.WORD	306
	042032	044674				.WORD	T31RWN
	042034	015554				.WORD	EXPREC
2010	042036			230:	CKLOOP		
	042036	104406					
2011	042040	013701	043220		MOV T31BFR+6, R1		
2012	042044	010102			MOV R1, R2		
2013	042046	052702	000002		BIS #BIT1, R2		
2014	042052	020102			CMP R1, R2		
2015	042054	001406			BEQ 240:		
2016	042056	005237	002214		INC FATFLG		
2020	042062				ERRHRD ERRNO, T31BOT, EXPREC		
	042062	104456				TRAP	C:ERHRD
	042064	000463				.WORD	307
	042066	044345				.WORD	T31BOT
	042070	015554				.WORD	EXPREC
2021	042072			240:	CKLOOP		
	042072	104406					
2022	042074	012737	041012	043310	265:	MOV #041012, T31PK3	
2023	042102	012704	043310		MOV #T31PK3, R4		
2024	042106	010337	043316		MOV R3, T31SZ		
2025	042112	010465	000000		MOV R4, TSDB(R5)		
2026	042116	004737	016330		JSR PC, WAITF		
2027	042122	016501	000002		MOV TSSR(R5), R1		
2028	042126	012702	000200		MOV #SSR, R2		
2029	042132	020102			CMP R1, R2		
2030	042134	001406			BEQ 280:		
2031	042136	005237	002214		INC FATFLG		
2035	042142				ERRHRD ERRNO, T31RDF, PKTSSR		
	042142	104456				TRAP	C:ERHRD
	042144	000464				.WORD	308
	042146	043543				.WORD	T31RDF
	042150	012126				.WORD	PKTSSR
2036	042152			280:	CKLOOP		
	042152	104406					
2037	042154	013701	043220		MOV T31BFR+6, R1		
						TRAP	C:CLP1

2136	042540	012704	043310		MOV	#T31PK3,R4		;SET UP R4 WITH PACKET ADDRESS
2137	042544	012700	000144		MOV	#100.,R0		;SET PATTERN IN CORRECT REGISTER
2138	042550	004737	017502		JSR	PC,FILLMEM		;FILL MEMORY WITH RECORD SIZE
2139	042554	012737	000144	043316	MOV	#100.,T31SZ		;SET UP RECORD SIZE IN PACKET
2140	042562	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND
2141	042566	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET
2142	042572	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
2143	042576	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED
2144	042602	020102			CMP	R1,R2		;ARE THEY EQUAL
2145	042604	001406			BEQ	80\$;BR, IF OK
2146	042606	005237	002214		INC	FATFLG		;ERROR COUNT
2150	042612				ERRHRD	ERRNO,T31WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA
	042612	104456					TRAP	C\$ERHRD
	042614	000474					.WORD	316
	042616	045230					.WORD	T31WDC
	042620	012126					.WORD	PKTSSR
2151	042622			80\$:	CKLOOP			;LOOP IF SELECTED
	042622	104406					TRAP	C\$CLP1
2152	042624	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND
2153	042630	103407			BCS	230\$;BR, IF NO PROBLEM
2154	042632	010001			MOV	R0,R1		;SAVE TSSR
2155	042634	005237	002214		INC	FATFLG		;ERROR COUNT
2159	042640				ERRHRD	ERRNO,T31RWN,EXPREC		;REWIND NOT ACCEPTED
	042640	104456					TRAP	C\$ERHRD
	042642	000475					.WORD	317
	042644	044674					.WORD	T31RWN
	042646	015554					.WORD	EXPREC
2160	042650			230\$:	CKLOOP			;LOOP IF SELECTED
	042650	104406					TRAP	C\$CLP1
2161	042652	013701	043220		MOV	T31BFR*6,R1		;PICK UP XSTO
2162	042656	010102			MOV	R1,R2		;SET UP EXPECTED
2163	042660	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED
2164	042664	020102			CMP	R1,R2		;DOES EXP = REC'D
2165	042666	001406			BEQ	240\$;BR, IF EQUAL (OK)
2166	042670	005237	002214		INC	FATFLG		;ERROR COUNT
2170	042674				ERRHRD	ERRNO,T31BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND
	042674	104456					TRAP	C\$ERHRD
	042676	000476					.WORD	318
	042700	044345					.WORD	T31BOT
	042702	015554					.WORD	EXPREC
2171	042704			240\$:	CKLOOP			;LOOP IF SELECTED
	042704	104406					TRAP	C\$CLP1
2172	042706	012737	041012	043310	265\$:	MOV	#041012,T31PK3	;INITIALIZE,CVC-1 COMMAND
2173	042714	012704	043310		MOV	#T31PK3,R4		;SET UP R4 WITH PACKET ADDRESS
2174	042720	010337	043316		MOV	R3,T31SZ		;SET UP RECORD SIZE IN PACKET
2175	042724	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND
2176	042730	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET
2177	042734	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
2178	042740	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED
2179	042744	020102			CMP	R1,R2		;ARE THEY EQUAL
2180	042746	001406			BEQ	280\$;BR, IF OK
2181	042750	005237	002214		INC	FATFLG		;ERROR COUNT
2185	042754				ERRHRD	ERRNO,T31RDF,PKTSSR		;TSSR INCORRECT AFTER READ DATA
	042754	104156					TRAP	C\$ERHRD
	042756	000477					.WORD	319
	042760	043543					.WORD	T31RDF
	042762	012126					.WORD	PKTSSR

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 3: NO OP ("CLEAN TAPE") AND INITIALIZE

SEQ 137

```

043164 104432
043166 003600
2235
2236
2237
2241 043170
2242 043170 100004
2243 043172 043200
2244 043174 000000
2245 043176 000012
2246 043200
2247 043200 043212
2248 043202 000000
2249 043204 000024
2250 043206 000000
2251 043210 000000
2252 043212
2253
2254
2255
2257 043300 043300
2259 043300 100006
2261 043302 043320
2262 043304 000000
2263 043306 000006
2264
2268 043310
2269 043310 100005
2270 043312
2271 043312 003116
2272 043314 000000
2273 043316 000000
2274
2275
2276
2277
2278 043320
2279 043320 010
2280 043321 200
2281 043322 000000
2282 043324 000000
2283
2284
2285
2286
2287
2288 043326 100205
2289 043330 100605
2290 043332 102205
2291 043334 177777
2292
2293
2294 043336 000000
2295 043340 000000
2296 043342 000000
2297

;
;LOCAL STORAGE FOR THIS TEST
;
T31PACKET:
;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>E177770
T31PK2:
;WRITE SUB SYS MEM COMMAND, AND ACK
;ADDRESS OF SELECT BLOCK DATA
;SIZE OF DATA PACKET
;
T31PK3:
;REREAD COMMAND, AND ACK
;ADDRESS OF WRITE BUFFER
;SIZE OF BUFFER (EXTENT)
;
;
;
T31BF2:
T31BS0: .BYTE 10 ;BSELO AREA
T31BS1: .BYTE 200 ;BSEL1 AREA
T31S2: .WORD 0 ;SEL 2 AREA
T31S3: .WORD 0 ;DATA AREA
;
;
;EVEN
;TAPE MOTION PACKET COMMAND VALUES
;
T31RN: .WORD 100205 ;REREAD DATA (NEXT)
T31WR: .WORD 100605 ;REREAD DATA RETRY
T31CON: .WORD 102205 ;WRITE CONTINUOUS
;END OF DATA
;
;
T31CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T31CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T31DLT: .WORD 0 ;DELAY COUNTER
;

```

TRAP C\$EXIT
 .WORD L10050 .

```

2298 ;LOCAL TEXT MESSAGES FOR TEST
2299 ;
2300
2301 043344 124 123 123 T31RDE: .ASCIZ 'TSSR Not Correct After READ Command'
2302 043410 124 141 160 T31WNH: .ASCIZ 'Tape Position Incorrect After INITIALIZE Command'
2303 043471 124 141 160 T31WNG: .ASCIZ 'Tape Position Incorrect After NO? Command'
2304 043543 124 123 123 T31RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
2305 043612 122 105 122 T31RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
2306 043707 120 117 123 T31SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
2307 043771 122 111 102 T31LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
2308 044041 124 123 123 T31WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
2309 044116 111 154 154 T31LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
2310 044177 122 105 122 T31SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
2311 044233 124 123 123 T31WDE: .ASCIZ 'TSSR Not Correct After NO-OP ("CLEAN TAPE") AND INITIALIZE Command, At BOT'
2312 044345 124 141 160 T31BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XSTO)'
2313 044440 116 117 055 T31TIM: .ASCIZ 'NO-OP ("CLEAN TAPE") AND INITIALIZE''S Erase Tape Not Long Enough'
2314 044540 122 105 122 T31EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
2315 044617 124 123 123 T31TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
2316 044674 122 145 167 T31RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
2317 044743 122 101 115 T31RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
2318 045016 124 123 123 T31AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
2319 045065 104 162 151 T31OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
2320 045140 124 123 123 T31WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
2321 045230 124 123 123 T31WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
2322 045303 103 126 103 T31VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
2323 045356 124 123 102 T31BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
2324 045431 127 122 111 T31WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
2325 045520 122 145 141 T31LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XSTO'
2326 045602 122 145 141 T31LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XSTO'
2327 045664 122 145 163 T31PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
2328 045752 122 145 141 T31TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
2329 046040 116 117 055 T31NEF: .ASCIZ 'NO OP ("CLEAN TAPE") AND INITIALIZE, At First Record, Failed To Set RIB Bit'
x
2330 046161 124 123 123 T31SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
2331 046236 124 123 123 T31TSA: .ASCIZ 'TSSR Not Correct After NO OP ("CLEAN TAPE") AND INITIALIZE, Into BOT'
2332 046343 124 123 123 T31WRF: .ASCIZ 'TSSR Not Correct After NO-OP ("CLEAN TAPE") AND INITIALIZE Command'
2333 046446 104 141 164 T31DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
2334 046543 116 117 055 T31ID: .ASCIZ 'NO OP ("Clean Tape") And INITIALIZE'
2335 .EVEN
2336 ;*
2337 ;
2338 ;ROUTINE TO RESTORE COMMAND PACKET TO START UP (DEFAULT) VALUES
2339 ;WRITE SUBSYSTEM MEMORY COMMAND
2340 ;
2341 ;
2342 ;
2343 046610 T31REST:
2344 046610 SAVREG ;SAVE THE REGISTERS
2345 046614 012701 043170 MOV #T31PACKET,R1 ;START OF THE PACKET
2346 046620 012721 100004 MOV #100004,(R1); ;WRITE SUBSYSTEM MEM. WITH ACK.
2347 046624 012721 043200 MOV #T31DATA,(R1); ;ADDRESS OF CHARAISTICS DATA BLOCK
2348 046630 005021 CLR (R1); ;EXTENDED ADDRESS
2349 046632 012721 000012 MOV #10.,(R1); ;SIZE OF DATA BLOCK IN BYTES
2350 046636 012721 043212 MOV #T31BFR,(R1); ;ADDRESS OF MESSAGE BUFFER
2351 046642 005021 CLR (R1);
2352 046644 012721 000024 MOV #20.,(R1); ;LENGTH OF MESSAGE BUFFER
2353 046650 005021 CLR (R1);
2354 046652 012711 000000 MOV #0,(R1) ;SELECT DRIVE ZERO

```

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

SEQ 139

```

2355 046656 012702 000030      MOV      #24.,R2          ;NUMBER OF LOCATIONS TO BE CLEARED
2356 046662 012762 177777 043212 64$: MOV      #177777,T31BFR(R2) ;ALL ONES TO MESSAGE BUFFER
2357 046670 005742              TST      -(R2)           ;NEXT LOCATION
2358 046672 022702 000000      CMP      #0,R2          ;AT END OF LOOP YET
2359 046676 001371              BNE      64$           ;KEEP GOING UNTIL DONE
2360 046700 000207              RTS      PC             ;RETURN
2361
2362 046702              T31RT2:
2363 046702              SAVREG                ;SAVE THE REGISTERS
2364 046706 012701 043300      MOV      #T31PK2,R1    ;START OF THE PACKET
2365 046712 012721 100006      MOV      #100006,(R1). ;WRITE SUBSYSTEM MEM. WITH ACK.
2366 046716 012721 043320      MOV      #T31BF2,(R1). ;ADDRESS OF DATA BLOCK
2367 046722 005021              CLR      (R1).         ;EXTENDED ADDRESS
2368 046724 012721 000006      MOV      #6.,(R1).    ;SIZE OF DATA BLOCK IN BYTES
2369 046730 005021              CLR      (R1).
2370 046732 012701 043320      MOV      #T31BF2,R1    ;POINT TO DATA SEL AREA
2371 046736 005021              CLR      (R1).
2372 046740 005011              CLR      (R1)
2373 046742 000207              RTS      PC             ;RETURN
2374 046744              T31RT3:
2375 046744              SAVREG                ;SAVE REGISTERS
2376 046750 012701 043310      MOV      #T31PK3,R1    ;SET UP POINTER ADDRESS
2377 046754 005021              CLR      (R1).         ;COMMAND SPACE
2378 046756 005021              CLR      (R1).         ;ADDRESS OF DATA BLOCK
2379 046760 005021              CLR      (R1).         ;EXTENDED ADDRESS
2380 046762 005011              CLR      (R1)         ;SIZE OF DATA TRANSFER BLOCK
2381 046764 000207              RTS      PC             ;RETURN
2382 046766
                                L10050:
                                TRAP      C$ETST
                                046766
                                104401
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
    
```

.SBTTL TEST 4: Erase And Operation Incomplete

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

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

2659	050052	004737	017502		JSR	PC,FILLMEM		;CALL MEMORY FILLER
2660	050056	013737	003116	051452	MOV	FREE,T32WB		;STARTING WRITE BUFFER ADDRESS
2661	050064	012737	140005	051450 65:	MOV	#140005,T32PK3		;WRITE DATA,CVC-1,ACK COMMAND
2662	050072	012704	051450		MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS
2663	050076	010300			MOV	R3,R0		;SET PATTERN IN CORRECT REGISTER
2664	050100	004737	017502		JSR	PC,FILLMEM		;FILL MEMORY WITH RECORD SIZE
2665	050104	010337	051456		MOV	R3,T32SZ		;SET UP RECORD SIZE IN PACKET
2666	050110	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND
2667	050114	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET
2668	050120	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
2669	050124	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED
2670	050130	020102			CMP	R1,R2		;ARE THEY EQUAL
2671	050132	001406			BEQ	80:		;BR. IF OK
2672	050134	005237	002214		INC	FATFLG		;ERROR COUNT
2676	050140				ERRHRD	ERRNO,T32WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA
	050140	104456					TRAP	C\$ERHRD
	050142	000637					.WORD	415
	050144	052536					.WORD	T32WDC
	050146	012126					.WORD	PKTSSR
2677	050150			80:	CKLOOP			;LOOP IF SELECTED
	050150	104406					TRAP	C\$CLP1
2678	050152	005723			TST	(R3).		;BUMP RECORD SIZE COUNTER
2679	050154	020327	000156		CMP	R3,#110.		;AT 160 SIZE YET
2680	050160	001341			BNE	65:		;BR. IF MORE RECORDS TO WRITE
2681	050162	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND
2682	050166	103407			BCS	230:		;BR. IF NO PROBLEM
2683	050170	010001			MOV	R0,R1		;SAVE TSSR
2684	050172	005237	002214		INC	FATFLG		;ERROR COUNT
2688	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
2689	050206			230:	CKLOOP			;LOOP IF SELECTED
	050206	104406					TRAP	C\$CLP1
2690	050210	013701	051360		MOV	T32BFR*6,R1		;PICK UP XSTO
2691	050214	010102			MOV	R1,R2		;SET UP EXPECTED
2692	050216	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED
2693	050222	020102			CMP	R1,R2		;DOES EXP = REC'D
2694	050224	001406			BEQ	240:		;BR. IF EQUAL (OK)
2695	050226	005237	002214		INC	FATFLG		;ERROR COUNT
2699	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
2700	050242			240:	CKLOOP			;LOOP IF SELECTED
	050242	104406					TRAP	C\$CLP1
2701	050244	012703	000001		MOV	#1,R3		;SET UP FOR SPACE COMMAND
2702	050250	004737	010544		JSR	PC,SPACE		;ISSUE SPACE COMMAND 1 FORWARD
2703	050254	012737	140411	051450 265:	MOV	#140411,T32PK3		;ERASE DATA,ACK COMMAND
2704	050262	012704	051450		MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS
2705	050266	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND
2706	050272	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET
2707	050276	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
2708	050302	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED
2709	050306	020102			CMP	R1,R2		;ARE THEY EQUAL

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 4: ERASE AND OPERATION INCOMPLETE

SEQ 148

2805	050612	012704	051330		MOV	#T32PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS	
2806	050616	004737	010742		JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS	
2807	050622	103407			BCS	23\$;BR, IF COMMAND ISSUED OK	
2808	050624	005237	002214		INC	FATFLG		;ERROR COUNT	
2812	050630	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
2813	050632				ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTICSC FAILED	
	050632	104456						TRAP	C\$ERHRD
	050634	000646						.WORD	422
	050636	005052						.WORD	WRTMSG
	050640	012114						.WORD	SFIMSG
2814	050642			23\$:	CKLOOP			;LOOP IF SELECTED	
	050642	104406						TRAP	C\$CLP1
2815	050644	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND	
2816	050650	103411			BCS	30\$;BR, IF NO PROBLEM	
2817	050652	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS	
2818	050656	010004			MOV	R0,R4		;GET PACKET ADDRESS	
2819	050660	005237	002214		INC	FATFLG		;ERROR COUNT	
2823	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
2824	050674			30\$:	CKLOOP			;LOOP IF SELECTED	
	050674	104406						TRAP	C\$CLP1
2825	050676	013701	051360		MOV	T32BFR+6,R1		;PICK UP XSTO	
2826	050702	010102			MOV	R1,R2		;SET UP EXPECTED	
2827	050704	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED	
2828	050710	020102			CMP	R1,R2		;DOES EXP = REC'D	
2829	050712	001406			BEQ	40\$;BR, IF EQUAL (OK)	
2830	050714	005237	002214		INC	FATFLG		;ERROR COUNT	
2834	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
2835	050730			40\$:	CKLOOP			;LOOP IF SELECTED	
	050730	104406						TRAP	C\$CLP1
2836	050732	012737	140411	051450	65\$:	MOV	#140411,T32PK3	;ERASE DATA,CVC=1,ACK COMMAND	
2837	050740	012704	051450		MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS	
2838	050744	010337	051456		MOV	R3,T32SZ		;SET UP RECORD SIZE IN PACKET	
2839	050750	010465	000000		MOV	R4,T32DB(R5)		;ISSUE COMMAND	
2840	050754	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET	
2841	050760	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS	
2842	050764	012702	000200		MOV	#SSR,R2		;SET UP EXPECTED	
2843	050770	020102			CMP	R1,R2		;ARE THEY EQUAL	
2844	050772	001757			BEQ	65\$;BR, IF OK	
2845	050774	032701	000004		BIT	#BIT2,R1		;CHECK FOR TAPE STATUS ALERT	
2846	051000	001006			BNE	80\$;BR, IF TAPE STATUS ALERT SET	
2847	051002	005237	002214		INC	FATFLG		;ERROR COUNT	
2851	051006				ERRHRD	ERRNO,T32WDC,PKTSSR		;TSSR INCORRECT AFTEF WRITE DATA	
	051006	104456						TRAP	C\$ERHRD
	051010	000651						.WORD	425
	051012	052536						.WORD	T32WDC
	051014	012126						.WORD	PKTSSR
2852	051016	104406		80\$:	CKLOOP			;LOOP IF SELECTED	
	051016	104406						TRAP	C\$CLP1
2853	051020	013701	051360		MOV	T32BFR+6,R1		;PICK UP XSTO	

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 4: ERASE AND OPERATION INCOMPLETE

SEQ 151

```

2957
2958
2959
2960
2961 051460
2962 051460 140410
2963 051462 141410
2964 051464 140401
2965 051466 141001
2966 051470 161401
2967 051472 161001
2968 051474 141401
2969 051476 140001
2970 051500 141410
2971 051502 141010
2972 051504 141005
2973 051506 177777
2974
2975
2976 051510 000000
2977 051512 000000
2978 051514 000000
2979
2980
2981
2982
2983 051516 124 141 160
2984 051611 124 141 160
2985 051700 122 145 167
2986 051747 124 123 123
2987 052016 124 123 123
2988 052063 124 123 102
2989 052136 122 105 101
2990 052234 124 123 123
2991 052311 124 123 123
2992 052366 102 117 124
2993 052455 105 122 101
2994 052536 124 123 123
2995 052603 117 120 111
2996 052640 105 162 141
2997
2998
2999
3000
3001
3002
3003
3004
3005 052700
3006 052700
3007 052704 012701 051330
3008 052710 012721 100004
3009 052714 012721 051340
3010 052720 005021
3011 052722 012721 000012
3012 052726 012721 051352
3013 052732 005021

;
; .EVEN
; TAPE MOTION PACKET COMMAND VALUES
T32CMD:
; .WORD 140410 ; SPACE RECORDS REVERSE
; .WORD 141410 ; SKIP TAPE MARKS REVERSE
; .WORD 140401 ; READ REVERSE
; .WORD 141001 ; REREAD PREVIOUS (OPP=0)
; .WORD 161401 ; REREAD NEXT (OPP=1)
; .WORD 161001 ; REREAD PREVIOUS (OPP=1)
; .WORD 141401 ; REREAD NEXT (OPP=0)
; .WORD 140001 ; READ NEXT
; .WORD 141410 ; SKIP TAPE MARKS REVERSE
; .WORD 141010 ; SKIP RECORDS FORWARD
; .WORD 141005 ; WRITE DATA RETRY
; .WORD 177777 ; END OF DATA

;
T32CNT: .WORD 0 ; TAPE TIMER COUNTER STORAGE AREA
T32CNU: .WORD 0 ; TAPE TIMER COUNTER STORAGE AREA
T32DLY: .WORD 0 ; DELAY COUNTER
;
; *
; LOCAL TEXT MESSAGES FOR TEST
; -
T32BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
T32EOT: .ASCIZ 'Tape Status Alert During Erase To EOT, But EOT Not Set'
T32RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
T32AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
T32ERA: .ASCIZ 'TSSR Not Correct After ERASE Command'
T32BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
T32RIB: .ASCIZ 'READ REVERSE, After ERASE From BOT, Failed To Set RIB In XST3'
T32SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
T32TSA: .ASCIZ 'TSSR Not Correct After READ REVERSE Into BOT'
T32BOE: .ASCIZ 'BOT (XST0) Still Set After Erase From Tape's BOT Marker'
T32ECF: .ASCIZ 'ERASE Failed To Clear Tape (Erase) Tape Properly'
T32WDC: .ASCIZ 'TSSR Not Correct After ERASE Command'
T32OPI: .ASCIZ 'OPI Bit (XST3) Failed To Set'
T32ID: .ASCIZ 'Erase And Operation Incomplete'
; .EVEN
;
; *
; ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
; WRITE SUBSYSTEM MEMORY COMMAND
;
;
T32REST:
; SAVREG ; SAVE THE REGISTERS
MOV #T32PACKET,R1 ; START OF THE PACKET
MOV #100004,(R1); ; WRITE SUBSYSTEM MEM. WITH ACK.
MOV #T32DATA,(R1); ; ADDRESS OF CHARAISTICS DATA BLOCK
CLR (R1); ; EXTENDED ADDRESS
MOV #10.,(R1); ; SIZE OF DATA BLOCK IN BYTES
MOV #T32BFR,(R1); ; ADDRESS OF MESSAGE BUFFER
CLR (R1);

```

```

3014 052734 012721 000024      MOV      #20.,(R1),      ;LENGTH OF MESSAGE BUFFER
3015 052740 005021              CLR      (R1),
3016 052742 012711 000000      MOV      #0,(R1)        ;SELECT DRIVE ZERO
3017 052746 012702 000030      MOV      #24.,R2       ;NUMBER OF LOCATIONS TO BE CLEARED
3018 052752 012762 177777 051352 64$: MOV      #177777,T32BFR(R2) ;ALL ONES TO MESSAGE BUFFER
3019 052760 005742              TST     -(R2)           ;NEXT LOCATION
3020 052762 022702 000000      CMP     #0,R2          ;AT END OF LOOP YET
3021 052766 001371              BNE     64$            ;KEEP GOING UNTIL DONE
3022 052770 000207              RTS     PC              ;RETURN
3023
3024 052772                    T32RT2:
3025 052772                    SAVREG                ;SAVE THE REGISTERS
3026 052776 012701 051440      MOV     #T32PK2,R1     ;START OF THE PACKET
3027 053002 012721 100006      MOV     #100006,(R1),  ;WRITE SUBSYSTEM MEM. WITH ACK.
3028 053006 005021              CLR     (R1),          ;ADDRESS OF DATA BLOCK
3029 053010 005021              CLR     (R1),          ;EXTENDED ADDRESS
3030 053012 012721 000006      MOV     #6.,(R1),     ;SIZE OF DATA BLOCK IN BYTES
3031 053016 005021              CLR     (R1),
3032 053020 000207              RTS     PC              ;RETURN
3033 053022                    T32RT3:
3034 053022                    SAVREG                ;SAVE REGISTERS
3035 053026 012701 051450      MOV     #T32PK3,R1     ;SET UP POINTER ADDRESS
3036 053032 005021              CLR     (R1),          ;COMMAND SPACE
3037 053034 005021              CLR     (R1),          ;ADDRESS OF DATA BLOCK
3038 053036 005021              CLR     (R1),          ;EXTENDED ADDRESS
3039 053040 005011              CLR     (R1),          ;SIZE OF DATA TRANSFER BLOCK
3040 053042 000207              RTS     PC              ;RETURN
3041 053044                    L10053:
3041 053044                    TRAP     C$ETST
3041 053044 104401

```

```

3042                    .SBTTL TEST 5: DATA PARITY TEST
3043
3044
3045
3046
3047
3048
3049 ;TEST 5 -- Data Parity Test
3050
3051
3052 ;This test verifies that the data parity circuitry in both the controller and the
3053 ;transport is operating properly by forcing data records with wrong parity to be
3054 ;written onto tape and checking the results obtained when the data is read. The
3055 ;following test sequence is performed:
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068

```

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.

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

```

3106 053046          BGNTST
      053046
3107 053046 012737 006413 002172      MOV    #EPRT2,EPRTSW      ;SECONDARY ERROR MESSAGE
3112 053054 012700 055645              MOV    #T33IC,R0        ;ASCII MESSAGE TO IDENTIFY TEST
3113 053060 004737 016570              JSR    PC,T33SETUP      ;DO INITIAL TEST SETUP
3114 053064 012737 000005 002210      MOV    #5,LOOPCNT      ;PERFORM 5 ITERATIONS
3115 053072 005037 054716              CLR    T33CNT          ;CLEAR TAPE RECORD COUNTER
3116
3117
3118 053076          T33LOOP:
3119 053076          BGNSUB
      053076
      053076 104402
3120 053100 005037 002216              CLR    INTRECV         ;INTERRUPT INDICATOR
3121 053104 005037 054716              CLR    T33CNT         ;TIMER FOR WRITE DATA SPACING
3122 053110 005037 054720              CLR    T33CNU        ;TIMER FOR WRITE DATA RETRY SPACING
3123 053114 004737 055662              JSR    PC,T33REST     ;SET COMMAND PACKET
3124 053120 004737 055754              JSR    PC,T33RT2     ;SET UP OTHER COMMAND PACKET
3125 053124 004737 056016              JSR    PC,T33RT3     ;SET UP OTHER COMMAND PACKET
3126 053130 012737 176750 054722      MOV    #65000.,T33DLV ;SET UP DELAY COUNTER

```

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06-FEB 84 18:04
 TEST 5: DATA PARITY TEST

SEQ 154

3127	053136	004737	016054	10%:	JSR	PC,SOFINIT		;DO INITIALIZE ON CONTROLLER	
3128	053142	103426			BCS	20\$;BR IF INIT WAS OK	
3129	053144				DELAY	250		;DELAY ABOUT .25 SEC	
	053144	012727	000250					MOV	#250,(PC)+
	053150	000000						.WORD	0
	053152	013727	002116					MOV	L\$DLY,(PC)+
	053156	000000						.WORD	0
	053160	005367	177772					DEC	-6(PC)
	053164	001375						BNE	.-4
	053166	005367	177756					DEC	-22(PC)
	053172	001367						BNE	.-20
3130	053174	005337	054722		DEC	T33DLY		;BUMP COUNTER	
3131	053200	001356			BNE	10\$;BR, IF COUNTER NOT DONE	
3132	053202	005237	002214		INC	FATFLG		;ERROR COUNT	
3136	053206	010001			MOV	R0,R1		;CONTENTS OF TSSR REGISTER	
3137	053210				ERRDF	ERRNO,SFIERR,SFIMSG		;FATAL ERROR TSSR WAS NOT OK	
	053210	104455						TRAP	C\$ERDF
	053212	000765						.WORD	501
	053214	003646						.WORD	SFIERR
	053216	012114						.WORD	SFIMSG
3138	053220	013737	002174	054570	20%:	MOV	UNITN,T33DSW	;SET UP UNIT NUMBER	
3139									
3140	053226	012704	054550		MOV	#T33PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS	
3141	053232	004737	010742		JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS	
3142	053236	103407			BCS	23\$;BR, IF COMMAND ISSUED OK	
3143	053240	005237	002214		INC	FATFLG		;ERROR COUNT	
3147	053244	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
3148	053246				ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTICSC FAILED	
	053246	104456						TRAP	C\$ERHRD
	053250	000766						.WORD	502
	053252	005052						.WORD	WRTMSG
	053254	012114						.WORD	SFIMSG
3149	053256			23%:	CKLOOP			;LOOP IF SELECTED	
	053256	104406						TRAP	C\$CLP1
3150	053260	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND	
3151	053264	103411			BCS	30\$;BR, IF NO PROBLEM	
3152	053266	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS	
3153	053272	010004			MOV	R0,R4		;GET PACKET ADDRESS	
3154	053274	005237	002214		INC	FATFLG		;ERROR COUNT	
3158	053300				ERRHRD	ERRNO,T33RWN,PKTSSR		;REWIND NOT ACCEPTED	
	053300	104456						TRAP	C\$ERHRD
	053302	000767						.WORD	503
	053304	055420						.WORD	T33RWN
	053306	012126						.WORD	PKTSSR
3159	053310			30%:	CKLOOP			;LOOP IF SELECTED	
	053310	104406						TRAP	C\$CLP1
3160	053312	013701	054600		MOV	T33BFR+6,R1		;PICK UP XSTO	
3161	053316	010102			MOV	R1,R2		;SET UP EXPECTED	
3162	053320	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED	
3163	053324	020102			CMP	R1,R2		;DOES EXP = REC'D	
3164	053326	001406			BEQ	40\$;BR, IF EQUAL (OK)	
3165	053330	005237	002214		INC	FATFLG		;ERROR COUNT	
3169	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

```

3170 053344          40$:  CKLOOP          ;LOOP IF SELECTED
      053344 104406          TRAP      C$CLP1
3171 053346 005737 002220 42$:  TST      EXTFEA      ;CHECK FOR EXTENDED FEATURES SW SWITCH
3172 053352 001025          BNE      55$      ;BR IF SWITCH IS ON
3173 053354 112737 000200 054701 MOVB   #200,T33BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
3174 053362 112737 000010 054700 MOVB   #10,T33BS0 ;FUNC. SEL. BIT (TURN ON EXTFEA SWITCH)
3175 053370 012704 054660      MOV     #T33PK2,R4 ;WRITE SUBSYS MEM PACKET
3176 053374 010465 000000      MOV     R4,TSDB(R5) ;ISSUE COMMAND
3177 053400 004737 016416      JSR    PC,CHKTSSR ;WAIT FOR SSR
3178 053404 103407          BCS     50$      ;BR, IF NO ERROR
3179 053406 010001          MOV     R0,R1    ;ERROR, SAVE TSSR
3180 053410 005237 002214      INC     FATFLG   ;ERROR COUNT
3184 053414          ERRHRD  ERRNO,T33SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      053414 104456          TRAP      C$ERHRD
      053416 000771          .WORD   505
      053420 055241          .WORD   T33SSR
      053422 012126          .WORD   PKTSSR
3185 053424          50$:  CKLOOP          ;LOOP IF SELECTED
      053424 104406          TRAP      C$CLP1
3186 053426 005737 002222 55$:  TST      BENBSW      ;CHECK FOR BUFFER ENABLED
3187 053432 001426          BEQ     70$      ;BR, IF BUFFERING NOT ENABLED
3188 053434 013737 002174 054570 MOV     UNITN,T33DSW ;SET UP UNIT NUMBER
3189 053442 042737 000020 05457C BIC     #BIT4,T33DSW ;BUFFER DISABLE
3190 053450 052737 000010 054570 BIS     #BIT3,T33DSW ;BUFFER DISABLE SEND 01 TO BITS 4 AND 3
3191 053456 012704 054550      MOV     #T33PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
3192 053462 004737 010742      JSR    PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
3193 053466 103407          BCS     60$      ;BR, IF COMMAND ISSUED OK
3194 053470 005237 002214      INC     FATFLG   ;ERROR COUNT
3198 053474 010001          MOV     R0,R1    ;SAVE CONTENT OF TSSR
3199 053476          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      053476 104456          TRAP      C$ERHRD
      053500 000772          .WORD   506
      053502 005052          .WORD   WRTMSG
      053504 012114          .WORD   SFIMSG
3200 053506          60$:  CKLOOP          ;LOOP IF SELECTED
      053506 104406          TRAP      C$CLP1
3201 053510          70$:
3202 053510 112737 000100 054701 MOVB   #100,T33BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
3203 053516 112737 000011 054700 MOVB   #11,T33BS0 ;FUNC. SEL. BIT (SET WRONG PARITY)
3204 053524 012704 054660      MOV     #T33PK2,R4 ;WRITE SUBSYS MEM PACKET
3205 053530 010465 000000      MOV     R4,TSDB(R5) ;ISSUE COMMAND
3206 053534 004737 016416      JSR    PC,CHKTSSR ;WAIT FOR SSR
3207 053540 103407          BCS     80$      ;BR, IF NO ERROR
3208 053542 010001          MOV     R0,R1    ;ERROR, SAVE TSSR
3209 053544 005237 002214      INC     FATFLG   ;ERROR COUNT
3213 053550          ERRHRD  ERRNO,T33SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      053550 104456          TRAP      C$ERHRD
      053552 000773          .WORD   507
      053554 055241          .WORD   T33SSR
      053556 012126          .WORD   PKTSSR
3214 053560          80$:  CKLOOP          ;LOOP IF SELECTED
      053560 104406          TRAP      C$CLP1
3215 053562 012703 000026          MOV     #22.,R3    ;NUMBER OF RECORDS TO BE WRITTEN
3216 053566 013737 003116 054672 MOV     FREE,T33WB ;STARTING WRITE BUFFER ADDRESS
3217 053574 005037 054720      CLR    T33CNU   ;MAKE SURE ITS CLEAR
3218 053600 012737 140005 054670 110$: MOV     #140005,T33PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
3219 053606 012704 054670      MOV     #T33PK3,R4 ;SET UP R4 WITH PACKET ADDRESS

```

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 06-FEB 84 18:04
 TEST 5: DATA PARITY TEST

SEQ 156

3220	053612	012737	000024	054676	MOV	#20.,T33SZ	;SET UP RECORD SIZE IN PACKET		
3221	053620	013777	054720	.27270	MOV	T33CNU,DFREE	;MEMORY FILLED WITH DATA IN RECORD		
3222	053626	005237	054720		INC	T33CNU	;READY FOR NEXT RECORD		
3223	053632	010465	000000		MOV	R4 TSD8(R5)	;ISSUE COMMAND		
3224	053636	004737	016330		JSR	PC,WAITF	;WAIT FOR SCR TO SET		
3225	053642	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
3226	053646	012702	100210		MOV	#SSR!SC!BIT3,R2	;SET UP EXPECTED		
3227	053652	020102			CMP	R1,R2	;ARE THEY EQUAL		
3228	053654	001406			BEQ	120\$;BR. IF OK		
3229	053656	005237	002214		INC	FATFLG	;ERROR COUNT		
3233	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	
3234	053672			120\$:	CKLOOP		;LOOP IF SELECTED		
	053672	104406					TRAP	C\$CLP1	
3235	053674	013701	054602		MOV	T33BFR+10,R1	;PICK UP XST1		
3236	053700	010102			MOV	R1,R2	;SET UP EXPECTED		
3237	053702	052702	000002		BIS	#BIT1,R2	;SET UNC BIT IN EXPECTED		
3238	053706	020102			CMP	R1,R2	;DOES EXP = REC'D		
3239	053710	001406			BEQ	130\$;BR. IF EQUAL (OK)		
3240	053712	005237	002214		INC	FATFLG	;ERROR COUNT		
3244	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	
3245	053726			130\$:	CKLOOP		;LOOP IF SELECTED		
	053726	104406					TRAP	C\$CLP1	
3246	053730	005303			DEC	R3	;DEC RECORD COUNTER		
3247	053732	001322			BNE	110\$;BR. IF MORE RECORDS TO WRITE		
3248	053734	004737	011074		JSR	PC,REWIND	;CALL TAPE REWIND COMMAND		
3249	053740	103411			BCS	140\$;BR. IF NO PROBLEM		
3250	053742	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
3251	053746	010004			MOV	R0,R4	;GET PACKET ADDRESS		
3252	053750	005237	002214		INC	FATFLG	;ERROR COUNT		
3256	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	
3257	053764			140\$:	CKLOOP		;LOOP IF SELECTED		
	053764	104406					TRAP	C\$CLP1	
3258	053766	013701	054600		MOV	T33BFR+5,R1	;PICK UP XST0		
3259	053772	010102			MOV	R1,R2	;SET UP EXPECTED		
3260	053774	052702	000002		BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED		
3261	054000	020102			CMP	R1,R2	;DOES EXP = REC'D		
3262	054002	001406			BEQ	150\$;BR. IF EQUAL (OK)		
3263	054004	005237	002214		INC	FATFLG	;ERROR COUNT		
3267	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	
3268	054020			150\$:	CKLOOP		;LOOP IF SELECTED		
	054020	104406					TRAP	C\$CLP1	

3269	054022	005037	054720		CLR	T33CNU	;CLEAR DATA VALUE IN RECORD	
3270	054026	012703	000024		MOV	#20,,R3	;RECORD SIZE	
3271	054032	013737	003116	054672	155:	MOV	FREE,T33RB	;STARTING WRITE BUFFER ADDRESS
3272	054040	012737	140001	054670		MOV	#140001,T33PK3	;READ DATA,CVC=1,ACK COMMAND
3273	054046	012704	054670		MOV	#T33PK3,R4	;SET UP R4 WITH PACKET ADDRESS	
3274	054052	012737	000024	054676		MOV	#20,,T33SZ	;SET UP RECORD SIZE IN PACKET
3275	054060	010465	000000		MOV	R4,TSD8(R5)	;ISSUE COMMAND	
3276	054064	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET	
3277	054070	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS	
3278	054074	012702	100210		MOV	#SSR!SC!BIT3,R2	;SET UP EXPECTED	
3279	054100	020102			CMP	R1,R2	;ARE THEY EQUAL	
3280	054102	001406			BEQ	160:	;BR, IF OK	
3281	054104	005237	002214		INC	FATFLG	;ERROR COUNT	
3285	054110				ERRHRD	ERRNO,T33WDC,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA	
	054110	104456					TRAP C#ERRHRD	
	054112	001000					.WORD 512	
	054114	055467					.WORD T33WDC	
	054116	012126					.WORD PKTSSR	
3286	054120			160:	CKLOOP		;LOOP IF SELECTED	
	054120	104406					TRAP C#CLP1	
3287	054122	013701	054602		MOV	T33BFR+10,R1	;PICK UP XST1	
3288	054126	010102			MOV	R1,R2	;SET UP EXPECTED	
3289	054130	052702	000002		BIS	#BIT1,R2	;SET UNC BIT IN EXPECTED	
3290	054134	020102			CMP	R1,R2	;DOES EXP = REC'D	
3291	054136	001406			BEQ	170:	;BR, IF EQUAL (OK)	
3292	054140	005237	002214		INC	FATFLG	;ERROR COUNT	
3296	054144				ERRHRD	ERRNO,T33UND,EXPREC	;UNC BIT NOT SET AFTER READ CMD.	
	054144	104456					TRAP C#ERRHRD	
	054146	001001					.WORD 513	
	054150	055152					.WORD T33UND	
	054152	015554					.WORD EXPREC	
3297	054154			170:	CKLOOP		;LOOP IF SELECTED	
	054154	104406					TRAP C#CLP1	
3298	054156	013701	054602		MOV	T33BFR+10,R1	;PICK UP XST1	
3299	054162	010102			MOV	R1,R2	;SET UP EXPECTED	
3300	054164	052702	000400		BIS	#BIT8,R2	;SET RBP BIT IN EXPECTED	
3301	054170	020102			CMP	R1,R2	;DOES EXP = REC'D	
3302	054172	001406			BEQ	180:	;BR, IF EQUAL (OK)	
3303	054174	005237	002214		INC	FATFLG	;ERROR COUNT	
3307	054200				ERRHRD	ERRNO,T33RBP,EXPREC	;READ BUS PARITY ERROR BIT NOT SET	
	054200	104456					TRAP C#ERRHRD	
	054202	001002					.WORD 514	
	054204	054724					.WORD T33RBP	
	054206	015554					.WORD EXPREC	
3308	054210			180:	CKLOOP		;LOOP IF SELECTED	
	054210	104406					TRAP C#CLP1	
3309	054212	017701	126700		MOV	#FREE,R1	;GET DATA READ	
3310	054216	013702	054720		MOV	T33CNU,R2	;GET PATTERN	
3311	054222	020102			CMP	R1,R2	;ARE THEY EQUAL	
3312	054224	001406			BEQ	182:	;BR, IF OK	
3313	054226	005237	002214		INC	FATFLG	;ERROR COUNT	
3317	054232				ERRHRD	ERRNO,T33DTA,EXPREC	;DATA NOT CORRECT	
	054232	104456					TRAP C#ERRHRD	
	054234	001003					.WORD 515	
	054236	055550					.WORD T33DTA	
	054240	015554					.WORD EXPREC	
3318	054242			182:	CKLOOP		;LOOP IF SELECTED	

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06-FEB 84 18:04
 TEST 5: DATA PARITY TEST

SEQ 160

```

3427                                     .EVEN
3428                                     ;
3429                                     ;
3430                                     ;
3431 054700 T33BF2:
3432 054700      010 T33BS0: .BYTE 10 ;BSELO AREA
3433 054701      200 T33BS1: .BYTE 200 ;BSEL1 AREA
3434 054702 000000 T33S2: .WORD 0 ;SEL 2 AREA
3435 054704 000000 T33S3: .WORD 0 ;DATA AREA
3436                                     ;
3437                                     ;
3438                                     .EVEN
3439 ;TAPE MOTION PACKET COMMAND VALUES
3440
3441 054706 100205 T33RN: .WORD 100205 ;REREAD DATA (NEXT)
3442 054710 100605 T33WR: .WORD 100605 ;REREAD DATA RETRY
3443 054712 102205 T33CON: .WORD 102205 ;WRITE CONTINUOUS
3444 054714 177777 .WORD 177777 ;END OF DATA
3445
3446                                     ;
3447 054716 000000 T33CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
3448 054720 000000 T33CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
3449 054722 000000 T33DLY: .WORD 0 ;DELAY COUNTER
3450
3451 ;*
3452 ;LOCAL TEXT MESSAGES FOR TEST
3453 ;-
3454 054724      122      145      141 T33RBP: .ASCIZ 'Read Bus Parity Bit Not Set (XST1), Should Be
3455 055002      124      123      123 T33WPW: .ASCIZ 'TSSR Incorrect After Wrong Parity Write Command'
3456 055062      125      116      103 T33UNC: .ASCIZ 'UNC Bit (XST1) Not Set After Wrong Parity WRITE Command'
3457 055152      125      116      103 T33UND: .ASCIZ 'UNC Bit (XST1) Not Set After Wrong Parity READ Command'
3458 055241      127      122      111 T33SSR: .ASCIZ 'WRITE MISCELLANEOUS CONT/READ COMMAND Not Accepted'
3459 055325      124      141      160 T33BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
3460 055420      122      145      167 T33RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
3461 055467      124      123      123 T33WDC: .ASCIZ 'TSSR Not Correct After READ Wrong Parity Command'
3462 055550      104      141      164 T33DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
3463 055645      104      141      164 TST33ID: .ASCIZ 'Data Parity'
3464                                     .EVEN
3465                                     ;*
3466                                     ;
3467 ;ROUTINE TO RESTORE COMMAND PACKET TO START UP (DEFAULT) VALUES
3468 ;WRITE SUBSYSTEM MEMORY COMMAND
3469                                     ;
3470                                     ;-
3471
3472 055662 T33REST:
3473 055662      SAVREG
3474 055666 012701 054550      MOV #T33PACKET,R1 ;SAVE THE REGISTERS
3475 055672 012721 100004      MOV #100004,(R1). ;START OF THE PACKET
3476 055676 012721 054560      MOV #T33DATA,(R1). ;WRITE SUBSYSTEM MEM. WITH ACK,
3477 055702 005021      CLR (R1). ;ADDRESS OF CHARAISTICS DATA BLOCK
3478 055704 012721 000012      MOV #10.,(R1). ;EXTENDED ADDRESS
3479 055710 012721 054572      MOV #T33BFR,(R1). ;SIZE OF DATA BLOCK IN BYTES
3480 055714 005021      CLR (R1). ;ADDRESS OF MESSAGE BUFFER
3481 055716 012721 000024      MOV #20.,(R1). ;LENGTH OF MESSAGE BUFFER
3482 055722 005021      CLR (R1).
3483 055724 012711 000000      MOV #0,(R1) ;SELECT DRIVE ZERO

```

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 5: DATA PARITY TEST

SEQ 161

```

3484 055730 012702 000030          MOV      #24.,R2          ;NUMBER OF LOCATIONS TO BE CLEARED
3485 055734 012762 177777 054572 64$: MOV      #177777,T33BFR(R2) ;ALL ONES TO MESSAGE BUFFER
3486 055742 005742          TST      (R2)           ;NEXT LOCATION
3487 055744 022702 000000          CMP      #0,R2          ;AT END OF LOOP YET
3488 055750 001371          BNE     64$            ;KEEP GOING UNTIL DONE
3489 055752 000207          RTS     PC              ;RETURN
3490
3491 055754          T33RT2:
3492 055754          SAVREG          ;SAVE THE REGISTERS
3493 055760 012701 054660          MOV      #T33PK2,R1     ;START OF THE PACKET
3494 055764 012721 100006          MOV      #100006,(R1).  ;WRITE SUBSYSTEM MEM. WITH ACK.
3495 055770 012721 054700          MOV      #T33BF2,(R1). ;ADDRESS OF DATA BLOCK
3496 055774 005021          CLR      (R1).         ;EXTENDED ADDRESS
3497 055776 012721 000006          MOV      #6.,(R1).     ;SIZE OF DATA BLOCK IN BYTES
3498 056002 005021          CLR      (R1).
3499 056004 012701 054700          MOV      #T33BF2,R1    ;POINT TO DATA SEL AREA
3500 056010 005021          CLR      (R1).
3501 056012 005011          CLR      (R1)
3502 056014 000207          RTS     PC              ;RETURN
3503 056016          T33RT3:
3504 056016          SAVREG          ;SAVE REGISTERS
3505 056022 012701 054670          MOV      #T33PK3,R1    ;SET UP POINTER ADDRESS
3506 056026 005021          CLR      (R1).         ;COMMAND SPACE
3507 056030 005021          CLR      (R1).         ;ADDRESS OF DATA BLOCK
3508 056032 005021          CLR      (R1).         ;EXTENDED ADDRESS
3509 056034 005011          CLR      (R1)         ;SIZE OF DATA TRANSFER BLOCK
3510 056036 000207          RTS     PC              ;RETURN
3511 056040          ENDTST
3512 056040          L10057: TRAP      C$ETST
3513 056040 104401
3512          .SBTTL TEST 6: OPERATIONS AT EOT
3513          ;*
3514          ;
3515          ;THIS TEST VERIFIES PROPER OPERATION OF THE WRITE DATA RETRY
3516          ;COMMAND (SPACE REVERSE, ERASE, WRITE DATA)
3517          ;
3518          ;
3519          ;THE TEST CONSISTS OF THE FOLLOWING 1 SUBTEST
3520          ;
3521          ;
3522          ;
3523          ;-
3524 056042          BGNTST
3525 056042          T6::
3525 056042 012737 006354 002172          MOV      #EPRT1,EPRTSW ;PRIMARY ERROR MESSAGE
3530 056050 012700 063207          MOV      #TST34ID,R0   ;ASCII MESSAGE TO IDENTIFY TEST
3531 056054 004737 016570          JSR     PC,TSTSETUP    ;DO INITIAL TEST SETUP
3532 056060 012737 000005 002210          MOV      #5,LOOPCNT    ;PERFORM 5 ITERATIONS
3533 056066 005037 060672          CLR     T34CNT        ;CLEAR TAPE RECORD COUNTER
3534          ;*
3535          ;
3536          ;TEST 6, SUBTEST 1
3537          ;
3538          ;
3539          ; THIS TEST VERIFIES THAT THE EOT STATUS IS HANDLED PROPERLY BY
3540          ; THE VARIOUS TAPE MOTION COMMANDS. THE FOLLOWING TEST SEQUENCE
3541          ; IS PERFORMED:

```


3695	056474	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND
3696	056500	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET
3697	056504	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
3698	056510	012702	100204		MOV	#SC!SSR!BIT2,R2		;SET UP EXPECTED
3699	056514	020102			CMP	R1,R2		;ARE THEY EQUAL
3700	056516	001406			BEQ	90\$;BR, IF THEY ARE OK
3701	056520	005237	002214		INC	FATFLG		;ERROR COUNT
3705	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
3706	056534			90\$:	CKLOOP			;LOOP IF SELECTED
	056534	104406						TRAP C\$CLP1
3707	056536	013701	060570		MOV	T34BFR+6,R1		;PICK UP XSTO
3708	056542	010102			MOV	R1,R2		;SET UP EXPECTED
3709	056544	052702	000001		BIS	#BIT0,R2		;SET THE EOT BIT ON IN EXPECTED
3710	056550	020102			CMP	R1,R2		;WAS THE BIT ON
3711	056552	001406			BEQ	100\$;BR, IF EOT WAS FOUND
3712	056554	005237	002214		INC	FATFLG		;ERROR COUNT
3716	056560				ERRHRD	ERRNO,T34ETN,EXPREC		;EOT BIT (XSTO) NOT SET
	056560	104456						TRAP C\$ERHRD
	056562	001136						.WORD 606
	056564	061451						.WORD T34ETN
	056566	015554						.WORD EXPREC
3717	056570			100\$:	CKLOOP			;LOOP IF SELECTED
	056570	104406						TRAP C\$CLP1
3718	056572	012737	140011	060660	MOV	#140011,T34PK3		;WRITE TAPE MARK, ACK, CVC-1 COMMAND
3719	056600	012704	060660		MOV	#T34PK3,R4		;R4 = POINTER TO PACKET
3720	056604	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND
3721	056610	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET
3722	056614	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
3723	056620	012702	100204		MOV	#SC!SSR!BIT2,R2		;SET UP EXPECTED
3724	056624	020102			CMP	R1,R2		;ARE THEY EQUAL
3725	056626	001406			BEQ	110\$;BR, IF STATUS IS GOOD (OK)
3726	056630	005237	002214		INC	FATFLG		;ERROR COUNT
3730	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
3731	056644			110\$:	CKLOOP			;LOOP IF SELECTED
	056644	104406						TRAP C\$CLP1
3732	056646	013701	060570		MOV	T34BFR+6,R1		;PICK UP XSTO
3733	056652	010102			MOV	R1,R2		;SET UP EXPECTED
3734	056654	052702	000001		BIS	#BIT0,R2		;SET THE EOT BIT ON IN EXPECTED
3735	056660	020102			CMP	R1,R2		;WAS THE BIT ON
3736	056662	001406			BEQ	120\$;BR, IF EOT WAS FOUND
3737	056664	005237	002214		INC	FATFLG		;ERROR COUNT
3741	056670				ERRHRD	ERRNO,T34ETO,EXPREC		;EOT BIT (XSTO) NOT SET
	056670	104456						TRAP C\$ERHRD
	056672	001140						.WORD 608
	056674	061002						.WORD T34ETO
	056676	015554						.WORD EXPREC
3742	056700			120\$:	CKLOOP			;LOOP IF SELECTED
	056700	104406						TRAP C\$CLP1
3743	056702	012737	141410	060660	MOV	#141410,T34PK3		;SKIP TAPE MARK REVERSE ACK,CVC-1 COMMAND

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 6: OPERATIONS AT EOT

SEQ 166

3744	056710	012737	000001	060662	MOV	#1,T34WB	;SET NUMBER (1) OF TMS TO SKIP
3745	056716	012704	060660		MOV	#T34PK3,R4	;R4 = PCINTER TO PACKET
3746	056722	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND
3747	056726	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
3748	056732	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS
3749	056736	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED
3750	056742	020102			CMP	R1,R2	;ARE THEY EQUAL
3751	056744	001406			BEQ	130\$;BR, IF STATUS IS GOOD (OK)
3752	056746	005237	002214		INC	FATFLG	;ERROR COUNT
3756	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
3757	056762			130\$:	CKLOOP		;LOOP IF SELECTED
	056762	104406					TRAP C\$CLP1
3758	056764	013701	060570		MOV	T34BFR+6,R1	;PICK UP XSTO
3759	056770	010102			MOV	R1,R2	;SET UP EXPECTED
3760	056772	052702	000001		BIS	#BIT0,R2	;SET THE EOT BIT ON IN EXPECTED
3761	056776	020102			CMP	R1,R2	;WAS THE BIT ON
3762	057000	001406			BEQ	140\$;BR, IF EOT WAS FOUND
3763	057002	005237	002214		INC	FATFLG	;ERROR COUNT
3767	057006				ERRHRD	ERRNO,T34ETN,EXPREC	;EOT BIT (XSTO) NOT SET
	057006	104456					TRAP C\$ERHRD
	057010	001142					.WORD 610
	057012	061451					.WORD T34ETN
	057014	015554					.WORD EXPREC
3768	057016			140\$:	CKLOOP		;LOOP IF SELECTED
	057016	104406					TRAP C\$CLP1
3769	057020	013701	060570		MOV	T34BFR+6,R1	;PICK UP XSTO
3770	057024	010102			MOV	R1,R2	;SET UP EXPECTED
3771	057026	052702	100000		BIS	#BIT15,R2	;SET THE TMK BIT ON IN EXPECTED
3772	057032	020102			CMP	R1,R2	;WAS THE BIT ON
3773	057034	001406			BEQ	150\$;BR, IF TMK WAS FOUND
3774	057036	005237	002214		INC	FATFLG	;ERROR COUNT
3778	057042				ERRHRD	ERRNO,T34TMK,EXPREC	;EOT BIT (XSTO) NOT SET
	057042	104456					TRAP C\$ERHRD
	057044	001143					.WORD 611
	057046	061763					.WORD T34TMK
	057050	015554					.WORD EXPREC
3779	057052			150\$:	CKLOOP		;LOOP IF SELECTED
	057052	104406					TRAP C\$CLP1
3780	057054	012737	140410	060660	MOV	#140410,T34PK3	;SPACE RECORDS REVERSE, ACK, CVC=1 CMD
3781	057062	012737	000001	060662	MOV	#1,T34WB	;SPACE ONE RECORD REVERSE
3782	057070	012704	060660		MOV	#T34PK3,R4	;R4 = POINTER TO PACKET
3783	057074	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND
3784	057100	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
3785	057104	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS
3786	057110	012702	100204		MOV	#SC!SSR!BIT2,R2	;SET UP EXPECTED
3787	057114	020102			CMP	R1,R2	;ARE THEY EQUAL
3788	057116	001006			BNE	160\$;BR, IT MIGHT BE END OF TAPE
3789	057120	005237	002214		INC	FATFLG	;ERROR COUNT
3793	057124				ERRHRD	ERRNO,T34POS,PKTSSR	;EOT NOT FOUND (USE SHORTER TAPE?)
	057124	104456					TRAP C\$ERHRD
	057126	001144					.WORD 612
	057130	060714					.WORD T34POS
	057132	012126					.WORD PKTSSR

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 6: OPERATIONS AT EOT

SEQ 167

```

3794 057134          160$: CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      057134 104406
3795 057136 013701 060570          MOV    T34BFR+6,R1        ;PICK UP XSTO
3796 057142 010102          MOV    R1,R2             ;SET UP EXPECTED
3797 057144 052702 000001          BIS    #BIT0,R2         ;SET THE EOT BIT ON IN EXPECTED
3798 057150 020102          CMP    R1,R2            ;WAS THE BIT ON
3799 057152 001406          BEQ    163$            ;BR, IF EOT WAS FOUND
3800 057154 005237 002214          INC    FATFLG           ;ERROR COUNT
3804 057160          ERRHRD  ERRNO,T34ETN,EXPREC ;EOT BIT (XSTO) NOT SET
      057160 104456          TRAP    C$ERHRD
      057162 001145          .WORD  613
      057164 061451          .WORD  T34ETN
      057166 015554          .WORD  EXPREC

3805 057170          163$: CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      057170 104406
3806 057172 013701 060570          MOV    T34BFR+6,R1        ;PICK UP XSTO
3807 057176 010102          MOV    R1,R2             ;SET UP EXPECTED
3808 057200 042702 100000          BIC    #BIT15,R2        ;CLEAR THE TMK BIT ON IN EXPECTED
3809 057204 020102          CMP    R1,R2            ;WAS THE BIT ON
3810 057206 001406          BEQ    165$            ;BR, IF TMK WAS FOUND
3811 057210 005237 002214          INC    FATFLG           ;ERROR COUNT
3815 057214          ERRHRD  ERRNO,T34TMK,EXPREC ;EOT BIT (XSTO) NOT SET
      057214 104456          TRAP    C$ERHRD
      057216 001146          .WORD  614
      057220 061763          .WORD  T34TMK
      057222 015554          .WORD  EXPREC

3816 057224          165$: CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      057224 104406
3817 057226 012737 140410 060660          MOV    #140410,T34PK3    ;SPACE RECORDS REVERSE, ACK, CVC=1 CMD
3818 057234 012737 000001 060662          MOV    #1,T34WB         ;SPACE ONE RECORD REVERSE
3819 057242 012704 060660          MOV    #T34PK3,R4       ;R4 = POINTER TO PACKET
3820 057246 010465 000000          MOV    R4,TSDB(R5)      ;ISSUE COMMAND
3821 057252 004737 016330          JSR    PC,WAITF         ;WAIT FOR SSR TO SET
3822 057256 016501 000002          MOV    TSSR(R5),R1      ;GET TSSR CONTENTS
3823 057262 012702 000200          MOV    #SSR,R2          ;SET UP EXPECTED
3824 057266 020102          CMP    R1,R2            ;ARE THEY EQUAL
3825 057270 001406          BFQ    167$            ;BR, IT MIGHT BE END OF TAPE
3826 057272 005237 002214          INC    FATFLG           ;ERROR COUNT
3830 057276          ERRHRD  ERRNO,T34POS,PKTSSR ;EOT NOT FOUND (USE SHORTER TAPE?)
      057276 104456          TRAP    C$ERHRD
      057300 001147          .WORD  615
      057302 060714          .WORD  T34POS
      057304 012126          .WORD  PKTSSR

3831 057306          167$: CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      057306 104406
3832 057310 013701 060570          MOV    T34BFR+6,R1        ;PICK UP XSTO
3833 057314 010102          MOV    R1,R2             ;SET UP EXPECTED
3834 057316 042702 000001          BIC    #BIT0,R2         ;CLEAR THE EOT BIT ON IN EXPECTED
3835 057322 020102          CMP    R1,R2            ;WAS THE BIT OFF
3836 057324 001400          BEQ    170$            ;BR, IF EOT WAS FOUND
3837 057326          CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      057326 104406
3838 057330 012737 140010 060660          MOV    #140010,T34PK3    ;SPACE RECORDS FORWARD, ACK, CVC=1
3839 057336 012737 000002 060662          MOV    #2,T34WB         ;SPACE TWO RECORDS
3840 057344 012704 060660          MOV    #T34PK3,R4       ;R4 = POINTER TO PACKET
3841 057350 010465 000000          MOV    R4,TSDB(R5)      ;ISSUE COMMAND
3842 057354 004737 016330          JSR    PC,WAITF         ;WAIT FOR SSR TO SET

```


3893	057610				210\$:	CKLOOP				;LOOP IF SELECTED		
	057610	104406									TRAP	C\$CLP1
3894	057612	012737	140001	060660		MOV	#140001,T34PK3			;READ DATA, ACK, CVC=1		
3895	057620	013737	003116	060662		MOV	FREE,T34RB			;SET UP WRITE BUFFER ADDRESS		
3896	057626	012737	006654	060666		MOV	#3500.,T34SZ			;SET UP BUFFER SIZE (4K BYTES)		
3897	057634	012704	060660			MOV	#T34PK3,R4			;R4 = POINTER TO PACKET		
3898	057640	010465	000000			MOV	R4,TSDB(R5)			;ISSUE COMMAND		
3899	057644	004737	016330			JSR	PC,WAITF			;WAIT FOR SSR TO SET		
3900	057650	016501	000002			MOV	TSSR(R5),R1			;GET TSSR CONTENTS		
3901	057654	012702	000200			MOV	#SSR,R2			;SET UP EXPECTED		
3902	057660	020102				CMP	R1,R2			;ARE THEY EQUAL		
3903	057662	001406				BEQ	230\$;BR, IT MIGHT BE END OF TAPE		
3904	057664	005237	002214			INC	FATFLG			;ERROR COUNT		
3908	057670					ERRHRD	ERRNO,T34RRE,PKTSSR			;EOT NOT FOUND (USE SHORTER TAPE?)		
	057670	104456									TRAP	C\$ERHRD
	057672	001154									.WORD	620
	057674	061066									.WORD	T34RRE
	057676	012126									.WORD	PKTSSR
3909	057700				230\$:	CKLOOP				;LOOP IF SELECTED		
	057700	104406									TRAP	C\$CLP1
3910	057702	012737	140001	060660		MOV	#140001,T34PK3			;READ DATA, ACK, CVC=1		
3911	057710	013737	003116	060662		MOV	FREE,T34RB			;SET UP WRITE BUFFER ADDRESS		
3912	057716	012737	006654	060666		MOV	#3500.,T34SZ			;SET UP BUFFER SIZE (4K BYTES)		
3913	057724	012704	060660			MOV	#T34PK3,R4			;R4 = POINTER TO PACKET		
3914	057730	010465	000000			MOV	R4,TSDB(R5)			;ISSUE COMMAND		
3915	057734	004737	016330			JSR	PC,WAITF			;WAIT FOR SSR TO SET		
3916	057740	016501	000002			MOV	TSSR(R5),R1			;GET TSSR CONTENTS		
3917	057744	012702	000200			MOV	#SSR,R2			;SET UP EXPECTED		
3918	057750	020102				CMP	R1,R2			;ARE THEY EQUAL		
3919	057752	001406				BEQ	235\$;BR, IT MIGHT BE END OF TAPE		
3920	057754	005237	002214			INC	FATFLG			;ERROR COUNT		
3924	057760					ERRHRD	ERRNO,T34RRE,PKTSSR			;EOT NOT FOUND (USE SHORTER TAPE?)		
	057760	104456									TRAP	C\$ERHRD
	057762	001155									.WORD	621
	057764	061066									.WORD	T34RRE
	057766	012126									.WORD	PKTSSR
3925	057770				235\$:	CKLOOP				;LOOP IF SELECTED		
	057770	104406									TRAP	C\$CLP1
3926	057772	013701	060570			MOV	T34BFR+6,R1			;PICK UP XSTO		
3927	057776	010102				MOV	R1,R2			;SET UP EXPECTED		
3928	060000	052702	000001			BIS	#BIT0,R2			;SET THE EOT BIT ON IN EXPECTED		
3929	060004	020102				CMP	R1,R2			;WAS THE BIT ON		
3930	060006	001406				BEQ	<40\$;BR, IF EOT WAS FOUND		
3931	060010	005237	002214			INC	FATFLG			;ERROR COUNT		
3935	060014					ERRHRD	ERRNO,T34ETZ,EXPREC			;EOT BIT (XSTO) NOT SET		
	060014	104456									TRAP	C\$ERHRD
	060016	001156									.WORD	622
	060020	061622									.WORD	T34ETZ
	060022	015554									.WORD	EXPREC
3936	060024				240\$:	CKLOOP				;LOOP IF SELECTED		
	060024	104406									TRAP	C\$CLP1
3937	060026	012737	140410	060660		MOV	#140410,T34PK3			;SPACE RECORDS REVERSE, ACK, CVC=1 CMD.		
3938	060034	012737	000005	060662		MOV	#5,T34RB			;NUMBER OF RECORDS TO SPACE		
3939	060042	012704	060660			MOV	#T34PK3,R4			;R4 = POINTER TO PACKET		
3940	060046	010465	000000			MOV	R4,TSDB(R5)			;ISSUE COMMAND		
3941	060052	004737	016330			JSR	PC,WAITF			;WAIT FOR SSR TO SET		
3942	060056	016501	000002			MOV	TSSR(R5),R1			;GET TSSR CONTENTS		

4035	060520	103002		BCC	1638				
4036	060522	000137	056072	JMP	T34LOOP				;BR, IF NO LOOP REQUIRED
4037	060526			1638:	EXIT	TST			;EXECUTE AGAIN
	060526	104432							;ALL DONE THIS TEST
	060530	002662							TRAP C\$EXIT
4038									.WORD L10061..
4039									
4040									
4042		060540							
4044	060540			T34PACKET:					;COMMAND PACKET FOR TEST
4045	060540	100004			.WORD	100004			;WRITE CHARACTERISTICS COMMAND, WITH ACK
4046	060542	060550			.WORD	T34DATA			;ADDRESS OF CHARACTERISTICS BLOCK
4047	060544	000000			.WORD	0			
4048	060546	000010			.WORD	8.			;STARTING VALUE OF BLOCK SIZE
4049	060550			T34DATA:					;CHARACTERISTICS DATA BLOCK
4050	060550	060562			.WORD	T34BFR			.ADDRESS OF MESSAGE BUFFER
4051	060552	000000			.WORD	0			
4052	060554	000012			.WORD	10.			;LENGTH OF MESSAGE BUFFER
4053	060556	000000			.WORD	0			
4054	060560	000000		T34DSW:	.WORD	0			;SELECT DRIVE 0
4055	060562			T34BFR:	.BLKW	25.			;MESSAGE BUFFER
4056									
4057									
4058									
4060		060650							
4062	060650			T34PK2:					
4063	060650	100006			.WORD	100006			;WRITE SUB SYS MEM COMMAND, AND ACK
4064	060652	060676			.WORD	T34BF2			;ADDRESS OF SELECT BLOCK DATA
4065	060654	000000			.WORD	0			
4066	060656	000006			.WORD	6.			;SIZE OF DATA PACKET
4067									
4071	060660			T34PK3:					
4072	060660	100005			.WORD	100005			;WRITE COMMAND, AND ACK
4073	060662			T34RB:					
4074	060662	000000		T34WB:	.WORD	0			;ADDRESS OF WRITE/READ BUFFER
4075	060664	000000			.WORD	0			
4076	060666	000000		T34SZ:	.WORD	0			;SIZE OF BUFFER (EXTENT)
4077					.EVEN				
4078									
4079	060670	000000		T34RSZ:	.WORD	0			;LARGEST TAPE RECORD IN BITES
4080	060672	000000		T34CNT:	.WORD	0			;TAPE RECORD COUNTER
4081	060674	000000		T34DLY:	.WORD	0			;DELAY COUNTER
4082									
4083									
4084	060676			T34BF2:					
4085	060676	010		T34BS0:	.BYTE	10			;BSEL0 AREA
4086	060677	200		T34BS1:	.BYTE	200			;BSEL1 AREA
4087	060700	000000		T34S2:	.WORD	0			;SEL 2 AREA
4088	060702	000000		T34S3:	.WORD	0			;DATA AREA
4089									
4090									
4091									
4092									
4093									
4094	060704	100005		T34WD:	.WORD	100005			;WRITE DATA (NEXT)
4095	060706	100405		T34WDR:	.WORD	100405			;WRITE DATA RETRY
4096	060710	102005		T34CON:	.WORD	102005			;WRITE CONTINUOUS

```

4097 060712 177777          .WORD 177777          ;END OF DATA
4098
4099          ; LOCAL TEXT MESSAGES FOR TEST
4100          ;
4101
4102 060714      124      123      123  T34POS: .ASCIZ 'TSSR Incorrect After Position (SPACE RECORDS) Command'
4103 061002      127      122      111  T34ETO: .ASCIZ 'WRITE TAPE MARK At EOT Failed To Set EOT Bit (XSTO)'
4104 061066      122      105      101  T34RRE: .ASCIZ 'READ Command At EOT Didn't Give Normal Termination (TSSR)'
4105 061157      125      156      141  T34ETC: .ASCIZ 'Unable To Clear EOT Indication, (XSTO) Bit 0'
4106 061234      122      105      127  T34BOT: .ASCIZ 'REWIND Failed To Set BOT (XSTO) Bit'
4107 061300      127      122      111  T34WTM: .ASCIZ 'WRITE TAPE MARK At EOT Failed To Set Tape Status Alert'
4108 061367      127      122      111  T34ET2: .ASCIZ 'WRITE DATA At EOT Failed To Set Tape Status Alert'
4109 061451      127      122      111  T34ETN: .ASCIZ 'WRITE DATA At EOT Failed To Set EOT Bit (XSTO)'
4110 061530      123      120      101  T34ETS: .ASCIZ 'SPACE RECORDS FORWARD At EOT Failed To Set EOT Bit (XSTO)'
4111 061622      122      105      101  T34ETZ: .ASCIZ 'READ DATA At EOT Failed To Set EOT Bit (XSTO)'
4112 061700      124      123      123  T34STM: .ASCIZ 'TSSR Incorrect After SKIP TAPE MARK REVERSE At EOT'
4113 061763      120      117      123  T34TMK: .ASCIZ 'POSITION Command At EOT Onto Tape Mark Failed To Set TMK (XSTO)'
4114 062063      127      122      111  T34SSR: .ASCIZ 'WRITE Command Not Accepted'
4115 062116      105      117      124  T34ET: .ASCIZ 'EOT Not Found In 65000 3.5K Writes, (Use Shorter Tape)'
4116 062205      127      122      111  T34EOT: .ASCIZ 'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
4117 062263      124      123      123  T34TM: .ASCIZ 'TSSR Not Correct After WRITE Command Reject'
4118 062337      122      145      167  T34RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
4119 062406      122      101      115  T34RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
4120 062461      124      123      123  T34AM3: .ASCIZ 'TSSR Init. Failed After WRITE Command'
4121 062527      104      162      151  T34OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
4122 062602      124      123      123  T34WDD: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, SWB Bit Set'
4123 062671      124      123      123  T34WDC: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
4124 062773      103      126      103  T34VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
4125 063046      124      123      102  T34BA: .ASCIZ 'TSBA Not Correct After WRITE DATA Command'
4126 063120      127      122      111  T34WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
4127 063207      117      160      145  TST34ID: .ASCIZ 'Operations At EOT'
4128          .EVEN
4129          ;
4130          ;
4131          ; ROUTINE TO RESTORE COMMAND PACKET TO START UP (DEFAULT) VALUES
4132          ; WRITE SUBSYSTEM MEMORY COMMAND
4133          ;
4134          ;
4135          ;
4136 063232          T34REST:
4137 063232          SAVREG          ;SAVE THE REGISTERS
4138 063236      012701 060540      MOV          @T34PACKET,R1          ;START OF THE PACKET
4139 063242      012721 100004      MOV          @100004,(R1).          ;WRITE SUBSYSTEM MEM. WITH ACK
4140 063246      012721 060550      MOV          @T34DATA,(R1).          ;ADDRESS OF CHARAISTICS DATA BLOCK
4141 063252      005021          CLR          (R1).          ;EXTENDED ADDRESS
4142 063254      012721 000012      MOV          @10.(R1).          ;SIZE OF DATA BLOCK IN BYTES
4143 063260      012721 060562      MOV          @T34BFR,(R1).          ;ADDRESS OF MESSAGE BUFFER
4144 063264      005021          CLR          (R1).
4145 063266      012721 000024      MOV          @20.(R1).          ;LENGTH OF MESSAGE BUFFER
4146 063272      005021          CLR          (R1).
4147 063274      012711 000000      MOV          @0,(R1)          ;SELECT DRIVE ZERO
4148 063300      012702 000030      MOV          @24.,R2          ;NUMBER OF LOCATIONS TO BE CLEARED
4149 063304      012762 177777 060562 64: MOV          @177777,T34BFR(R2)          ;ALL ONES TO MESSAGE BUFFER
4150 063312      005742          TST          -(R2)          ;BUMP DOWN TO NEXT LOCATION
4151 063314      020227 000000      CMP          R2,@0          ;R2 AT ZERO YET
4152 063320      001371          BNE          64:          ;KEEP GOING UNTIL DONE
4153 063322      000207          RTS          PC          ;RETURN
    
```



```

4256 063636 005237 002214          INC    FATFLG          ;ERROR COUNT
4260 063642          ERRHRD  ERRNO,T35RWN,PKTSSR ;REWIND NOT ACCEPTED
      063642 104456          TRAP    C$ERHRD
      063644 001277          .WORD  703
      063646 070644          .WORD  T35RWN
      063650 012126          .WORD  PKTSSR
4261 063652          30$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      063652 104406          MOV    T35BFR+6,R1      ;PICK UP XSTO
4262 063654 013701 067420          MOV    R1,R2            ;SET UP EXPECTED
4263 063660 010102          BIS    @BIT1,R2         ;SET BOT BIT IN EXPECTED
4264 063662 052702 000002          CMP    R1,R2            ;DOES EXP = REC'D
4265 063666 020102          BEQ    40$              ;BR, IF EQUAL (OK)
4266 063670 001406          INC    FATFLG          ;ERROR COUNT
4267 063672 005237 002214          ERRHRD  ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      063676 104456          TRAP    C$ERHRD
      063700 001300          .WORD  704
      063702 070340          .WORD  T35BOT
      063704 015554          .WORD  EXPREC
4272 063706          40$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      063706 104406          MOV    @20.,R3          ;NUMBER OF RECORDS
4273 063710 012703 000024          MOV    @256.,T35SZ      ;SET UP RECORD SIZE
4274 063714 012737 000400 067516          MOV    FREE,T35WB       ;ADDRESS OF WRITE BUFFER
4275 063722 013737 003116 067512
4276
4277          ;*****
4278          ;
4279          ;WRITE DATA,ACK,CVC=1 COMMAND
4280          ;
4281          ;*****
4282
4283 063730 012737 140005 067510          MOV    @140005,T35PK3   ;WRITE DATA,ACK,CVC=1 COMMAND
4284 063736 012704 067510          MOV    @T35PK3,R4      ;SET UP R4 WITH PACKE: ADDRESS
4285 063742 010465 000000          50$:  MOV    R4,T35DB(R5) ;ISSUE COMMAND
4286 063746 004737 016330          JSR    PC,WAITF         ;WAIT FOR SSR TO SET
4287 063752 016501 000002          MOV    TSSR(R5),R1     ;GET TSSR CONTENTS
4288 063756 012702 000200          MOV    @SSR,R2         ;SET UP EXPECTED
4289 063762 020102          CMP    R1,R2           ;ARE THEY EQUAL
4290 063764 001406          BEQ    60$              ;BR, IF OK
4291 063766 005237 002214          INC    FATFLG          ;ERROR COUNT
4295 063772          ERRHRD  ERRNO,T35WDE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      063772 104456          TRAP    C$ERHRD
      063774 001301          .WORD  705
      063776 070266          .WORD  T35WDE
      064000 012126          .WORD  PKTSSR
4296 064002          60$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      064002 104406          DEC    R3                ;BUMP RECORD COUNTER
4297 064004 005303          BNE    50$              ;BR, IF MORE RRECORDS TO COUNT
4298 064006 001355
4299
4300          ;*****
4301          ;
4302          ;WAIT FOR TAPE TO STOP ALL MOTION
4303          ;
4304          ;*****
4305
4306 064010 012737 000012 067542          MOV    @10.,T35DLY      ;SET UP DELAY COUNTER
    
```

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06-FEB 84 18:04
 TEST 7: EXTENDED MODE FEATURES

SEQ 177

```

4307 064016          70$:  DELAY  250          ;WAIT ABOUT .25 SEC
      064016 012727 000250
      064022 000000
      064024 013727 002116
      064030 000000
      064032 005367 177772
      064036 001375
      064040 005367 177756
      064044 001367
4308 064046 005337 067542          DEC  T35DLY          ;BUMP COUNTER DOWN
4309 064052 001361          BNE  70$          ;BR, IF MORE TO DELAY
4310 064054 005737 002220          TST  EXTFEA          ;CHECK FOR EXTENDED FEATURES SW SWITCH
4311 064060 001042          BNE  110$         ;BR IF SWITCH IS ON
4312 064062 112737 000200 067521  MOVB  #200,T35BS1    ;WRITE MISCELLANEOUS CONT/READ STATUS
4313 064070 112737 000010 067520  MOVB  #10,T35BS0    ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4314 064076 012704 067500          MOV  #T35PK2,R4    ;WRITE SUBSYS MEM PACKET
4315 064102 010465 000000          MOV  R4,TSD8(R5)   ;ISSUE COMMAND
4316 064106 004737 016416          JSR  PC,CHKTSSR    ;WAIT FOR SSR
4317 064112 103407          BCS  90$          ;BR, IF NO ERROR
4318 064114 010001          MOV  R0,R1         ;ERROR, SAVE TSSR
4319 064116 005237 002214          INC  FATFLG        ;ERROR COUNT
4323 064122          ERRHRD  ERRNO,T35SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      064122 104456          TRAP  C$ERHRD
      064124 001302          .WORD 706
      064126 072422          .WORD T35SSR
      064130 012126          .WORD PKTSSR
4324 064132          90$:  CKLOOP          ;LOOP IF SELECTED
      064132 104406          TRAP  C$CLP1
4325 064134 012704 067370          MOV  #T35PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4326 064140 004737 010742          JSR  PC,WRTCHR     ;ISSUE WRITE CHARACTERISTICS
4327 064144 103407          BCS  100$         ;BR, IF COMMAND ISSUED OK
4328 064146 005237 002214          INC  FATFLG        ;ERROR COUNT
4332 064152 010001          MOV  R0,R1         ;SAVE CONTENTS OF TSSR
4333 064154          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      064154 104456          TRAP  C$ERHRD
      064156 001303          .WORD 707
      064160 005052          .WORD WRTMSG
      064162 012114          .WORD SFIMSG
4334 064164          100$: CKLOOP          ;SCOPE LOOP
      064164 104406          TRAP  C$CLP1
4335 064166 012737 176750 067542 110$:  MOV  #65000.,T35DLY ;SET UP DELAY COUNTER
4336 064174 005037 067536          CLR  T35CNT        ;DELAY COUNTER
4337
4338
4339
4340
4341
4342
4343
      ;*****
      ;
      ;REWIND IMED. INTERRUPT,ACK,CVC=1,IE=0 COMMAND
      ;
      ;*****
4344 064200 012737 142012 067510          MOV  #142012,T35PK3 ;REWIND IMED. INTERRUPT,ACK,CVC=1,IE=0 COMMAND
4345 064206 012704 067510          MOV  #T35PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
4346 064212 010465 000000          MOV  R4,TSD8(R5)   ;ISSUE COMMAND
4347 064216 016501 000002          MOV  TSSR(R5),R1   ;GET TSSR CONTENTS
4348 064222 032701 000200          BIT  #SSR,R1       ;CHECK FOR SSR SET
4349 064226 001021          BNE  130$         ;BR, WHEN SSR IS SET
4350 064230 005237 067536          INC  T35CNT        ;BUMP THE CYCLE COUNTER
4351 064234          DELAY  1          ;DELAY TO KEEP COUNTER DOWN

```

```

064234 012727 000001          MOV      #1,(PC)+
064240 000000          .WORD   0
064242 013727 002116          MOV      L$DLY,(PC)+
064246 000000          .WORD   0
064250 005367 177772          DEC      -6(PC)
064254 001375          BNE     .-4
064256 005367 177756          DEC      -22(PC)
064262 001367          BNE     . 20
4352 064264 005337 067542      DEC      T35DLY          ;DROP DEAD TIMER BUMP DOWN
4353 064270 001352          BNE     120$           ;BR. IF MORE TIME TO GO
4354 064272 012702 000200      130$:  MOV      #SSR,R2      ;SET UP EXPECTED
4355 064276 020102          CMP      R1,R2        ;ARE THEY EQUAL
4356 064300 001406          BEQ     140$           ;BR. IF OK
4357 064302 005237 002214      INC      FATFLG        ;ERROR COUNT
4361 064306          ERRHRD  ERRNO,T35RWE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
          064306 104456          TRAP    C$ERHRD
          064310 001304          .WORD   708
          064312 072770          .WORD   T35RWE
          064314 012126          .WORD   PKTSSR
4362 064316          140$:  CKLOOP          ;LOOP IF SELECTED
          064316 104406          TRAP    C$CLP1
4363 064320 005737 002216      TST     INTRECV        ;CHECK FOR INTERRUPTS
4364 064324 001410          BEQ     150$           ;BR. IF NO INTERRUPTS DETECTED
4365 064326 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR STATUS FOR PRINTOUT
4366 064332 005237 002214      INC      FATFLG        ;ERROR COUNT
4370 064336          ERRHRD  ERRNO,T35INT,PKTSSR ;INTERRUPT RECEIVED (BAD)
          064336 104456          TRAP    C$ERHRD
          064340 001305          .WORD   709
          064342 072601          .WORD   T35INT
          064344 012126          .WORD   PKTSSR
4371 064346          150$:  CKLOOP          ;LOOP IF SELECTED
          064346 104406          TRAP    C$CLP1
4372
4373      ;*****
4374      ;
4375      ;NOW CHECK FOR THE MOTION BITS SET
4376      ;
4377      ;*****
4378
4379 064350 013701 067420      MOV      T35BFR+6,R1   ;PICK UP XST0
4380 064354 010102          MOV      R1,R2        ;SET UP EXPECTED
4381 064356 052702 000200      BIS      #BIT7,R2      ;SET MOT BIT IN EXPECTED
4382 064362 020102          CMP      R1,R2        ;DOES EXP = REC'D
4383 064364 001406          BEQ     160$           ;BR. IF EQUAL (OK)
4384 064366 005237 002214      INC      FATFLG        ;ERROR COUNT
4388 064372          ERRHRD  ERRNO,T35MOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          064372 104456          TRAP    C$ERHRD
          064374 001306          .WORD   710
          064376 072503          .WORD   T35MOT
          064400 015554          .WORD   EXPREC
4389 064402          160$:  CKLOOP          ;LOOP IF SELECTED
          064402 104406          TRAP    C$CLP1
4390 064404 013701 067424      MOV      T35BFR+12,R1  ;PICK UP XST2
4391 064410 010102          MOV      R1,R2        ;SET UP EXPECTED
4392 064412 052702 100000      BIS      #BIT15,R2     ;SET OPM BIT IN EXPECTED
4393 064416 020102          CMP      R1,R2        ;DOES EXP = REC'D
4394 064420 001406          BEQ     170$           ;BR. IF EQUAL (OK)
    
```


TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06-FEB 84 18:04
 TEST 7: EXTENDED MODE FEATURES

SEQ 180

```

4436 064572          DELAY 250          ;DELAY ABOUT .25 SEC
      064572 012727 000250          MOV      #250,(PC)+
      064576 000000          .WORD 0
      064600 013727 002116          MOV      L$DLY,(PC)+
      064604 000000          .WORD 0
      064606 005367 177772          DEC      -6(PC)
      064612 001375          BNE     .-4
      064614 005367 177756          DEC     -22(PC)
      064620 001367          BNE     . 20
4437 064622 005337 067542          DEC     T35DLY
4438 064626 001356          BNE     10$
4439 064630 005237 002214          INC     FATFLG
4443 064634 010001          MOV     RO,R1
4444 064636          ERRDF  ERRNO,SFIERR,SFIMSG ;CONTENTS OF TSSR REGISTER
      064636 104455          ;FATAL ERROR TSSR WAS NOT OK
      064640 001310          TRAP   C$ERDF
      064642 003646          .WORD 712
      064644 012114          .WORD SFIERR
      064644 012114          .WORD SFIMSG
4445 064646 013737 002174 067410 20$: MOV     UNITN,T35DSW
4446 064654 012704 067370          MOV     #T35PACKET,R4
4447 064660 004737 010742          JSR    PC,WRTCHR
4448 064664 103407          BCS    25$
4449 064666 005237 002214          INC     FATFLG
4453 064672 010001          MOV     RO,R1
4454 064674          ERRHRD ERRNO,WRTMSG,SFIMSG ;SAVE CONTENTS OF TSSR
      064674 104456          ;WRITE CHARACTERISTICS FAILED
      064676 001311          TRAP   C$ERHRD
      064700 005052          .WORD 713
      064702 012114          .WORD WRTMSG
      064702 012114          .WORD SFIMSG
4455 064704          25$: CKLOOP          ;LOOP IF SELECTED
      064704 104406          TRAP   C$CLP1
4456 064706 004737 011074          JSR    PC,REWIND
4457 064712 103411          BCS    30$
4458 064714 010004          MOV     RO,R4
4459 064716 016501 000002          MOV     TSSR(R5),R1
4460 064722 005237 002214          INC     FATFLG
4464 064726          ERRHRD ERRNO,T35RWN,PKTSSR ;REWIND NOT ACCEPTED
      064726 104456          TRAP   C$ERHRD
      064730 001312          .WORD 714
      064732 070644          .WORD T35RWN
      064734 012126          .WORD PKTSSR
4465 064736          30$: CKLOOP          ;LOOP IF SELECTED
      064736 104406          TRAP   C$CLP1
4466 064740 013701 067420          MOV     T35BFR+6,R1
4467 064744 010102          MOV     R1,R2
4468 064746 052702 000002          BIS     #BIT1,R2
4469 064752 020102          CMP     R1,R2
4470 064754 001406          BEQ    40$
4471 064756 005237 002214          INC     FATFLG
4475 064762          ERRHRD ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      064762 104456          TRAP   C$ERHRD
      064764 001313          .WORD 715
      064766 070340          .WORD T35BOT
      064770 015554          .WORD EXPREC
4476 064772          40$: CKLOOP          ;LOOP IF SELECTED
      064772 104406          TRAP   C$CLP1
4477 064774 012703 000024          MOV     #20.,R3
      ;NUMBER OF RECORDS

```

```

4478 065000 012737 000400 067516      MOV      #256.,T35SZ      ;SET UP RECORD SIZE
4479 065006 013737 003116 067512      MOV      FREE,T35WB     ;ADDRESS OF WRITE BUFFER
4480
4481      ;*****
4482      ;
4483      ;WRITE DATA,ACK,CVC=1 COMMAND
4484      ;
4485      ;*****
4486
4487 065014 012737 140005 067510      MOV      #140005,T35PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
4488 065022 012704 067510      MOV      #T35PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
4489 065026 010465 000000      50$:    MOV      R4,T35DB(R5)   ;ISSUE COMMAND
4490 065032 004737 016330      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
4491 065036 016501 000002      MOV      T35SR(R5),R1   ;GET T35SR CONTENTS
4492 065042 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED
4493 065046 020102      CMP      R1,R2         ;ARE THEY EQUAL
4494 065050 001406      BEQ      60$           ;BR, IF OK
4495 065052 005237 002214      INC      FATFLG        ;ERROR COUNT
4499 065056      ERRHRD  ERRNO,T35WDE,PKTSSR ;T35SR INCORRECT AFTER WRITE DATA
      065056 104456      TRAP    C$ERHRD
      065060 001314      .WORD  716
      065062 070266      .WORD  T35WDE
      065064 012126      .WORD  PKTSSR
4500 065066      60$:    CKLOOP          ;LOOP IF SELECTED
      065066 104406      TRAP    C$CLP1
4501
4502      ;*****
4503      ;
4504      ;WAIT FOR TAPE TO STOP ALL MOTION
4505      ;
4506      ;*****
4507
4508 065070 012737 000012 067542      70$:    MOV      #10.,T35DLY ;SET UP DELAY COUNTER
4509 065076      DELAY  250           ;WAIT ABOUT .25 SEC
      065076 012727 000250      MOV      #250,(PC)+
      065102 000000      .WORD  0
      065104 013727 002116      MOV      L$DLY,(PC)+
      065110 000000      .WORD  0
      065112 005367 177772      DEC      -6(PC)
      065116 001375      BNE     -.4
      065120 005367 177756      DEC      -22(PC)
      065124 001367      BNE     .-20
4510 065126 005337 067542      DEC      T35DLY        ;BUMP COUNTER DOWN
4511 065132 001361      BNE     70$           ;BR, IF MORE TO DELAY
4512 065134 005737 002220      TST     EXTFEA        ;CHECK FOR EXTENDED FEATURES SW SWITCH
4513 065140 001042      BNE     110$         ;BR IF SWITCH IS ON
4514 065142 112737 000200 067521      MOVB    #200,T35BS1   ;WRITE MISCELLANEOUS CONT/READ STATUS
4515 065150 112737 000010 067520      MOVB    #10,T35BS0   ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4516 065156 012704 067500      MOV     #T35PK2,R4    ;WRITE SUBSYS MEM PACKET
4517 065162 010465 000000      MOV     R4,T35DB(R5) ;ISSUE COMMAND
4518 065166 004737 016416      JSR     PC,CHKTSSR   ;WAIT FOR SSR
4519 065172 103407      BCS    90$          ;BR, IF NO ERROR
4520 065174 010001      MOV     R0,R1       ;ERROR, SAVE T35SR
4521 065176 005237 002214      INC     FATFLG      ;ERROR COUNT
4525 065202      ERRHRD  ERRNO,T35SSR,PKTSSR ;T35SR NOT CORRECT AFTER WRT. MISCELLANEOUS
      065202 104456      TRAP    C$ERHRD
      065204 001315      .WORD  717
    
```

```

065206 072422 .WORD T35SSR
065210 012126 .WORD PKTSSR
4526 90$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
065212 104406 ;SUBROUTINE NEEDS PACKET ADDRESS
4527 065214 012704 067370 MOV #T35PACKET,R4 ;ISSUE WRITE CHARACTERISTICS
4528 065220 004737 010742 JSR PC,WRTCHR ;BR, IF COMMAND ISSUED OK
4529 065224 103407 BCS 100$ ;ERROR COUNT
4530 065226 005237 002214 INC FATFLG ;SAVE CONTENTS OF TSSR
4534 065232 010001 MOV R0,R1 ;WRITE CHARACTERISTICSC FAILED
4535 065234 ERRHRD ERRNO,WRTMSG,SFIMSG ;RAP C$ERHRD
065234 104456 .WORD 718
065236 001316 .WORD WRTMSG
065240 005052 .WORD SFIMSG
065242 012114
4536 100$: CKLOOP ;SCOPE LOOP TRAP C$CLP1
065244 104406
4537 065246 012737 176750 067542 110$: MOV #65000.,T35DLY ;SET UP DELAY COUNTER
4538 065254 005037 067536 CLR T35CNT ;DELAY COUNTER
4539
4540 ;*****
4541 ;
4542 ;REWIND IMMEDIATE,ACK,CVC=1,IE=1 COMMAND
4543 ;
4544 ;*****
4545
4546 065260 012737 142212 067510 MOV #142212,T35PK3 ;REWIND IMMEDIATE,ACK,CVC=1,IE=1 COMMAND
4547 065266 012704 067510 MOV #T35PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
4548 065272 010465 000000 MOV R4,T5DB(R5) ;ISSUE COMMAND
4549 065276 016501 000002 120$: MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4550 065302 032701 000200 BIT #SSR,R1 ;CHECK FOR SSR SET
4551 065306 001021 BNE 130$ ;BR, WHEN SSR IS SET
4552 065310 005237 067536 INC T35CNT ;BUMP THE CYCLE COUNTER
4553 065314 DELAY 1 ;DELAY TO KEEP COUNTER DOWN
065314 012727 000001 MOV #1,(PC)+
065320 000000 .WORD 0
065322 013727 002116 MOV L$DLY,(PC)+
065326 000000 .WORD 0
065330 005367 177772 DEC 6(PC)
065334 001375 BNE .4
065336 005367 177756 DEC -22(PC)
065342 001367 BNE .-20
4554 065344 005337 067542 DEC T35DLY ;DROP DEAD TIMER BUMP DOWN
4555 065350 001352 BNE 120$ ;BR, IF MORE TIME TO GO
4556 065352 012702 000200 130$: MOV #SSR,R2 ;SET UP EXPECTED
4557 065356 020102 CMP R1,R2 ;ARE THEY EQUAL
4558 065360 001406 BEQ 140$ ;BR, IF OK
4559 065362 005237 002214 INC FATFLG ;ERROR COUNT
4563 065366 ERRHRD ERRNO,T35RWE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
065366 104456 TRAP C$ERHRD
065370 001317 .WORD 719
065372 072770 .WORD T35RWE
065374 012126 .WORD PKTSSR
4564 140$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
065376 104406
4565 065400 005737 002216 TST INTRECV ;CHECK FOR INTERRUPTS
4566 065404 001010 BNE 150$ ;BR, IF INTERRUPTS DETECTED
4567 065406 016501 000002 MOV TSSR(R5),R1 ;GET TSSR STATUS FOR PRINTOUT
    
```



```

4568 065412 005237 002214          INC    FATFLG          ;ERROR COUNT
4572 065416          ERRHRD  ERRNO,T35NIN,PKTSSR ;INTERRUPT NOT RECEIVED (BAD)
                                TRAP   C%ERHRD
                                .WORD  720
                                .WORD  T35NIN
                                .WORD  PKTSSR
4573 065426          150$:  CKLOOP          ;LOOP IF SELECTED          TRAP   C%CLP1
                                065426  104406
4574
4575          ;*****
4576          ;
4577          ;NOW CHECK FOR THE MOTION BITS SET
4578          ;
4579          ;*****
4580
4581 065430 013701 067420          MOV    T35BFR+6,R1      ;PICK UP XSTO
4582 065434 010102          MOV    R1,R2           ;SET UP EXPECTED
4583 065436 052702 000200          BIS    #BIT7,R2        ;SET MOT BIT IN EXPECTED
4584 065442 020102          CMP    R1,R2           ;DOES EXP = REC'D
4585 065444 001406          BEQ    160$            ;BR, IF EQUAL (OK)
4586 065446 005237 002214          INC    FATFLG          ;ERROR COUNT
4590 065452          ERRHRD  ERRNO,T35MOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP   C%ERHRD
                                .WORD  721
                                .WORD  T35MOT
                                .WORD  EXPREC
4591 065462          160$:  CKLOOP          ;LOOP IF SELECTED          TRAP   C%CLP1
                                065462  104406
4592 065464 013701 067424          MOV    T35BFR+12,R1    ;PICK UP XST2
4593 065470 010102          MOV    R1,R2           ;SET UP EXPECTED
4594 065472 052702 100000          BIS    #BIT15,R2       ;SET OPM BIT IN EXPECTED
4595 065476 020102          CMP    R1,R2           ;DOES EXP = REC D
4596 065500 001406          BEQ    170$            ;BR, IF EQUAL (OK)
4597 065502 005237 002214          INC    FATFLG          ;ERROR COUNT
4601 065506          ERRHRD  ERRNO,T35OPM,EXPREC ;OPM BIT NOT SET
                                TRAP   C%ERHRD
                                .WORD  722
                                .WORD  T35OPM
                                .WORD  EXPREC
4602 065516          170$:  CKLOOP          ;LOOP IF SELECTED          TRAP   C%CLP1
                                065516  104406
4603 065520 012737 000027 067542          MOV    #23.,T35DLY     ;SET UP DELAY COUNTER
4604 065526          175$:  DELAY    250      ;START DELAY
                                MOV    #250,(PC)+
                                .WORD  0
                                MOV    L%DLY,(PC)+
                                .WORD  0
                                DEC    6(PC)
                                BNE    4
                                DEC    22(PC)
                                BNE    20
4605 065556 005337 067542          DEC    T35DLY          ;BUMP DELAY COUNTER
4606 065562 001361          BNE    175$            ;BR, IF MORE DELAY
4607 065564          ENDSUB
                                L10065:
4608 065566 023727 002214 000017          CMP    FATFLG,#15.    ;IS ERROR COUNT AT 15          TRAP   C%ESUB

```



```

065724 104456
065726 001325
065730 070644
065732 012126
4664 065734 304: CKLOOP ;LOOP IF SELECTED TRAP C$ERHRD
065734 104406 ;PICK UP XSTO TRAP C$CLP1
4665 065736 013701 067420 MOV T35BFR+6,R1 ;SET UP EXPECTED
4666 065742 010102 MOV R1,R2 ;SET BOT BIT IN EXPECTED
4667 065744 052702 000002 BIS #BIT1,R2 ;DOES EXP = REC'D
4668 065750 020102 CMP R1,R2 ;BR. IF EQUAL (OK)
4669 065752 001406 BEQ 404 ;ERROR COUNT
4670 065754 005237 002214 INC FATFLG ;TAPE NOT AT BOT AFTER REWIND
4674 065760 ERRHRD ERRNO,T35BOT,EXPREC TRAP C$ERHRD
065760 104456 .WORD 726
065762 001326 .WORD T35BOT
065764 070340 .WORD EXPREC
065766 015554
4675 065770 404: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
065770 104406 ;STARTING RECORD SIZE
4676 065772 012703 000024 MOV #20.,R3 ;STARTING WRITE BUFFER ADDRESS
4677 065776 013737 003116 067512 MOV FREE,T35WB
4678
4679 ;*****
4680 ;
4681 ;WRITE DATA,CVC=1,ACK COMMAND
4682 ;
4683 ;*****
4684
4685 066004 012737 140005 067510 654: MOV #140005,T35PK3 ;WRITE DATA,CVC=1,ACK COMMAND
4686 066012 012704 067510 MOV #T35PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
4687 066016 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
4688 066020 004737 017502 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
4689 066024 010337 067516 MOV R3,T35SZ ;SET UP RECORD SIZE IN PACKET
4690 066030 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
4691 066034 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
4692 066040 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4693 066044 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
4694 066050 020102 CMP R1,R2 ;ARE THEY EQUAL
4695 066052 001406 BEQ 804 ;BR. IF OK
4696 066054 005237 002214 INC FATFLG ;ERROR COUNT
4700 066060 ERRHRD ERRNO,T35WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
066060 104456 TRAP C$ERHRD
066062 001327 .WORD 727
066064 071200 .WORD T35WDC
066066 012126 .WORD PKTSSR
4701 066070 804: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
066070 104406 ;WRITE DATA RETRY,CVC=1,ACK COMMAND
4702 ;*****
4703 ;
4704 ;WRITE DATA RETRY,CVC=1,ACK COMMAND
4705 ;
4706 ;*****
4707 ;
4708
4709 066072 012737 141005 067510 MOV #141005,T35PK3 ;WRITE DATA RETRY,CVC=1,ACK COMMAND
4710 066100 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
4711 066104 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
    
```

```

4712 066110 016501 000002      MOV     TSSR(R5),R1      ;GET TSSR CONTENTS
4713 066114 012702 000200      MOV     #SSR,R2        ;SET UP EXPECTED
4714 066120 020102              CMP     R1,R2          ;ARE THEY EQUAL
4715 066122 001406              BEQ     90$            ;BR, IF OK
4716 066124 005237 002214      INC     FATFLG         ;ERROR COUNT
4720 066130              ERRHRD  ERRNO,T35WRF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA RETRY
                                TRAP     C$ERHRD
                                .WORD   728
                                .WORD   T35WRF
                                .WORD   PKTSSR
    066130 104456
    066132 001330
    066134 072245
    066136 012126
4721 066140              90$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP1
    066140 104406
4722 066142 005723              TST     (R3),          ;BUMP RECORD SIZE COUNTER
4723 066144 020327 000052      CMP     R3,#42.       ;AT 42 SIZE YET
4724 066150 001315              BNE     65$            ;BR, IF MORE RECORDS TO WRITE
4725 066152 004737 011074      JSR     PC,REWIND     ;CALL TAPE REWIND COMMAND
4726 066156 103411              BCS     230$          ;BR, IF NO PROBLEM
4727 066160 010001              MOV     R0,R1         ;SAVE TSSR
4728 066162 016501 000002      MOV     TSSR(R5),R1   ;GET TSSR CONTENTS
4729 066166 005237 002214      INC     FATFLG         ;ERROR COUNT
4733 066172              ERRHRD  ERRNO,T35RWN,EXPREC ;REWIND NOT ACCEPTED
                                TRAP     C$ERHRD
                                .WORD   729
                                .WORD   T35RWN
                                .WORD   EXPREC
    066172 104456
    066174 001331
    066176 070644
    066200 015554
4734 066202              230$: CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP1
    066202 104406
4735 066204 013701 067420      MOV     T35BFR+6,R1   ;PICK UP XSTO
4736 066210 010102              MOV     R1,R2        ;SET UP EXPECTED
4737 066212 052702 000002      BIS     #BIT1,R2     ;SET BOT BIT IN EXPECTED
4738 066216 020102              CMP     R1,R2        ;LJES EXP = REC'D
4739 066220 001406              BEQ     240$          ;BR, IF EQUAL (OK)
4740 066222 005237 002214      INC     FATFLG         ;ERROR COUNT
4744 066226              ERRHRD  ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP     C$ERHRD
                                .WORD   730
                                .WORD   T35BOT
                                .WORD   EXPREC
    066226 104456
    066230 001332
    066232 070340
    066234 015554
4745 066236              240$: CKLOOP          ;LOOP IF SELECTED
                                TRAP     C$CLP1
    066236 104406
4746 066240 012703 000024      MOV     #20.,R3       ;STARTING RECORD SIZE
4747 066244 013737 003116 067512  MOV     FREE,T35RB    ;STARTING READ BUFFER ADDRESS
4748
4749 ;*****
4750 ;
4751 ;READ DATA,ACK COMMAND
4752 ;
4753 ;*****
4754
4755 066252 012737 100001 067510 265$: MOV     #100001,T35PK3 ;READ DATA,ACK COMMAND
4756 066260 012704 067510      MOV     #T35PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
4757 066264 012700 177777      MOV     #177777,R0   ;SET PATTERN IN CORRECT REGISTER
4758 066270 004737 017502      JSR     PC,FILLMEM   ;FILL MEMORY WITH RECORD SIZE
4759 066274 010337 067516      MOV     R3,T35SZ     ;SET UP RECORD SIZE IN PACKET
4760 066300 010465 000000      MOV     R4,TSDB(R5)  ;ISSUE COMMAND
4761 066304 004737 016330      JSR     PC,WAITF     ;WAIT FOR SSR TO SET
4762 066310 016501 000002      MOV     TSSR(R5),R1  ;GET TSSR CONTENTS
    
```



```

066444
066444 104402
4813 066446 004737 073164 JSR PC,T35REST ;SET COMMAND PACKET
4814 066452 004737 073256 JSR PC,T35RT2 ;SET UP OTHER COMMAND PACKET
4815 066456 004737 073320 JSR PC,T35RT3 ;SET UP OTHER COMMAND PACKET
4816 066462 012737 176750 067542 MOV #65000.,T35DLY ;SET UP DELAY COUNTER
4817 066470 004737 016054 10$: JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
4818 066474 103426 BCS 20$ ;BR IF INIT WAS OK
4819 066476 DELAY 250 ;DELAY ABOUT .25 SEC
066476 012727 000250 MOV #250,(PC).
066502 000000 .WORD 0
066504 013727 002116 MOV L$DLY,(PC).
066510 000000 .WORD 0
066512 005367 177772 DEC 6(PC)
066516 001375 BNE .4
066520 005367 177756 DEC 22(PC)
066524 001367 BNE .20
4820 066526 005337 067542 DEC T35DLY ;BUMP COUNTER
4821 066532 001356 BNE 10$ ;BR, IF COUNTER NOT DONE
4822 066534 005237 002214 INC FATFLG ;ERROR COUNT
4826 066540 010001 MOV RO,P1 ;CONTENTS OF TSSR REGISTER
4827 066542 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
066542 104455 TRAP C$ERDF
066544 001335 .WORD 733
066546 003646 .WORD SFIERR
066550 012'14 .WORD SFIMSG
4828 066552 013737 002174 067410 20$: MOV UNITN,T35DSW ;SET UP UNIT (DRIVE) NUMBER
4829 066560 012704 067370 MOV #T35PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4830 066564 004737 010742 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
4831 066570 103407 BCS 23$ ;BR, IF COMMAND ISSUED OK
4832 066572 005237 002214 INC FATFLG ;ERROR COUNT
4836 066576 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
4837 066600 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
066600 104456 TRAP C$ERHRD
066602 001336 .WORD 734
066604 005052 .WORD WRTMSG
066606 012114 .WORD SFIMSG
4838 066610 23$: CKLOOP ;LOOP IF SELECTED
066610 104406 TRAP C$CLP1
4839 066612 004737 011074 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
4840 066616 103411 BCS 30$ ;BR, IF NO PROBLEM
4841 066620 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4842 066624 010004 MOV RO,R4 ;GET PACKET ADDRESS
4843 066626 005237 002214 INC FATFLG ;ERROR COUNT
4847 066632 ERRHRD ERRNO,T35RWN,PKTSSR ;REWIND NOT ACCEPTED
066632 104456 TRAP C$ERHRD
066634 001337 .WORD 735
066636 070644 .WORD T35RWN
066640 012126 .WORD PKTSSR
4848 066642 30$: CKLOOP ;LOOP IF SELECTED
066642 104406 TRAP C$CLP1
4849 066644 013701 067420 MOV T35BFR+6,R1 ;PICK UP XSTO
4850 066650 010102 MOV R1,R2 ;SET UP EXPECTED
4851 066652 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
4852 066656 020102 CMP R1,R2 ;DOES EXP = REC'D
4853 066660 001406 BEQ 40$ ;BR, IF EQUAL (OK)
4854 066662 005237 002214 INC FATFLG ;ERROR COUNT

```

```

4858 066666          ERRHRD  ERRNO,T35BOT,EXPREC          ;TAPE NOT AT BOT AFTER REWIND
      066666 104456          TRAP          C$ERHRD
      066670 001340          .WORD          736
      066672 070340          .WORD          T35BOT
      066674 015554          .WORD          EXPREC
4859          40$:  CKLOOP          ;LOOP IF SELECTED          TRAP          C$CLP1
      066676 104406
4860 066700 012703 000024          MOV          #20.,R3          ;STARTING RECORD SIZE
4861 066704 013737 003116 067512  MOV          FREE,T35WB          ;STARTING WRITE BUFFER ADDRESS
4862
4863          ;*****
4864          ;
4865          ;WRITE DATA,CVC=1,ACK COMMAND
4866          ;
4867          ;*****
4868
4869 066712 012737 140005 067510 65$:  MOV          #140005,T35PK3          ;WRITE DATA,CVC=1,ACK COMMAND
4870 066720 012704 067510          MOV          #T35PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
4871 066724 010300          MOV          R3,R0          ;SET PATTERN IN CORRECT REGISTER
4872 066726 004737 017502          JSR          PC,FILLMEM          ;FILL MEMORY WITH RECORD SIZE
4873 066732 010337 067516          MOV          R3,T35SZ          ;SET UP RECORD SIZE IN PACKET
4874 066736 010465 000000          MOV          R4,T35DB(R5)          ;ISSUE COMMAND
4875 066742 004737 016330          JSR          PC,WAITF          ;WAIT FOR SSR TO SET
4876 066746 016501 000002          MOV          T35SR(R5),R1          ;GET T35SR CONTENTS
4877 066752 012702 000200          MOV          #SSR,R2          ;SET U' EXPECTED
4878 066756 020102          CMP          R1,R2          ;ARE THEY EQUAL
4879 066760 001406          BEQ          80$          ;BR, IF OK
4880 066762 005237 002214          INC          FATFLG          ;ERROR COUNT
4884 066766          ERRHRD  ERRNO,T35WDC,PKTSSR          ;T35SR INCORRECT AFTER WRITE DATA
      066766 104456          TRAP          C$ERHRD
      066770 001341          .WORD          737
      066772 071200          .WORD          T35WDC
      066774 012126          .WORD          PKTSSR
4885          80$:  CKLOUP          ;LOOP IF SELECTED          TRAP          C$CLP1
      066776 104406
4886
4887          ;*****
4888          ;
4889          ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
4890          ;
4891          ;*****
4892
4893 067000 012737 111005 067510          MOV          #111005,T35PK3          ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
4894 067006 010465 000000          MOV          R4,T35DB(R5)          ;ISSUE COMMAND
4895 067012 004737 016330          JSR          PC,WAITF          ;WAIT FOR SSR TO SET
4896 067016 016501 000002          MOV          T35SR(R5),R1          ;GET T35SR CONTENTS
4897 067022 012702 000200          MOV          #SSR,R2          ;SET UP EXPECTED
4898 067026 020102          CMP          R1,R2          ;ARE THEY EQUAL
4899 067030 001406          BEQ          90$          ;BR, IF OK
4900 067032 005237 002214          INC          FATFLG          ;ERROR COUNT
4904 067036          ERRHRD  ERRNO,T35WRF,EXPREC          ;T35SR INCORRECT AFTER WRITE DATA RETRY
      067036 104456          TRAP          C$ERHRD
      067040 001342          .WORD          738
      067042 072245          .WORD          T35WRF
      067044 015554          .WORD          EXPREC
4905          90$:  CKLOOP          ;LOOP IF SELECTED          TRAP          C$CLP1
      067046 104406

```

```

4906 067050 005723          TST      (R3)+          ;BUMP RECORD SIZE COUNTER
4907 067052 020327 000052          CMP      R3,#42.          ;AT 42 SIZE YET
4908 067056 001315          BNE      65$              ;BR. IF MORE RECORDS TO WRITE
4909 067060 004737 011074          JSR      PC,REWIND        ;CALL TAPE REWIND COMMAND
4910 067064 103411          BCS      230$             ;BR. IF NO PROBLEM
4911 067066 016501 000002          MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
4912 067072 010004          MOV      R0,R4           ;GET PACKET ADDRESS
4913 067074 005237 002214          INC      FATFLG          ;ERROR COUNT
4917 067100          ERRHRD  ERRNO,T35RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     739
                                .WORD     T35RWN
                                .WORD     PKTSSR
                                TRAP      C$CLP1
4918 067110          230$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD     104406
4919 067112 013701 067420          MOV      T35BFR+6,R1     ;PICK UP XSTO
4920 067116 010102          MOV      R1,R2           ;SET UP EXPECTED
4921 067120 052702 000002          BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
4922 067124 020102          CMP      R1,R2           ;DOES EXP = REC'D
4923 067126 001406          BEQ      240$             ;BR. IF EQUAL (OK)
4924 067130 005237 002214          INC      FATFLG          ;ERROR COUNT
4928 067134          ERRHRD  ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD     740
                                .WORD     T35BOT
                                .WORD     EXPREC
                                TRAP      C$CLP1
4929 067144          240$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD     104406
4930 067146 012703 000024          MOV      @20.,R3         ;STARTING RECORD SIZE
4931 067152 013737 003116 067512          MOV      FREE,T35RB      ;STARTING READ BUFFER ADDRESS
4932
4933          ;*****
4934          ;
4935          ;READ DATA,ACK COMMAND
4936          ;
4937          ;*****
4938
4939 067160 012737 100001 067510 265$: MOV      @100001,T35PK3    ;READ DATA,ACK COMMAND
4940 067166 012704 067510          MOV      @T35PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
4941 067172 010337 067516          MOV      R3,T35SZ       ;SET UP RECORD SIZE IN PACKET
4942 067176 010465 000000          MOV      R4,TSD8(R5)    ;ISSUE COMMAND
4943 067202 004737 016330          JSR      PC,WAITF        ;WAIT FOR SSR TO SET
4944 067206 016501 000002          MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
4945 067212 012702 000200          MOV      @SSR,R2         ;SET UP EXPECTED
4946 067216 020102          CMP      R1,R2           ;ARE THEY EQUAL
4947 067220 001406          BEQ      280$             ;BR. IF OK
4948 067222 005237 002214          INC      FATFLG          ;ERROR COUNT
4952 067226          ERRHRD  ERRNO,T35RDF,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD     741
                                .WORD     T35RDF
                                .WORD     PKTSSR
                                TRAP      C$CLP1
4953 067236          280$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD     104406
4954 067240 013702 003116          MOV      FREE,R2         ;GET BUFFER ADDRESS
4955 067244 010304          MOV      R3,R4           ;GET RECORD SIZE
4956 067246 162704 000024          SUB      @20.,R4         ;POINT BACK TO 1ST RECORD
    
```



```

5009 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
5010 ;
5012 067500 .=<.+10>E177770
5014 067500 T35PK2:
5015 067500 100006 .WORD 100006 ;WRITE SUB SYS MEM COMMAND, AND ACK
5016 067502 067520 .WORD T35BF2 ;ADDRESS OF SELECT BLOCK DATA
5017 067504 000000 .WORD 0
5018 067506 000006 .WORD 6. ;SIZE OF DATA PACKET
5019
5023 067510 T35PK3:
5024 067510 100005 .WORD 100005 ;REREAD COMMAND, AND ACK
5025 067512 T35RB:
5026 067512 003116 T35WB: .WORD FREE ;ADDRESS OF WRITE BUFFER
5027 067514 000000 .WORD 0
5028 067516 000000 T35SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
5029 .EVEN
5030 ;
5031 ;
5032 ;
5033 067520 T35BF2:
5034 067520 010 T35BS0: .BYTE 10 ;BSELO AREA
5035 067521 200 T35BS1: .BYTE 200 ;BSEL1 AREA
5036 067522 000000 T35S2: .WORD 0 ;SEL 2 AREA
5037 067524 000000 T35S3: .WORD 0 ;DATA AREA
5038 ;
5039 ;
5040 .EVEN
5041 ;TAPE MOTION PACKET COMMAND VALUES
5042
5043 067526 100205 T35RN: .WORD 100205 ;REREAD DATA (NEXT)
5044 067530 100605 T35WDR: .WORD 100605 ;REREAD DATA RETRY
5045 067532 102205 T35CON: .WORD 102205 ;WRITE CONTINUOUS
5046 067534 177777 .WORD 177777 ;END OF DATA
5047
5048 ;
5049 067536 000000 T35CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
5050 067540 000000 T35CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
5051 067542 000000 T35DLY: .WORD 0 ;DELAY COUNTER
5052 ;
5053 ;LOCAL TEXT MESSAGES FOR TEST
5054 ;-
5055
5056 067544 124 141 160 T35WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
5057 067632 124 123 123 T35RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
5058 067701 122 105 122 T35RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
5059 067776 120 117 123 T35SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
5060 070060 122 111 102 T35LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
5061 070130 124 123 123 T35WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
5062 070205 111 154 154 T35LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
5063 070266 124 123 123 T35WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
5064 070340 124 141 160 T35BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
5065 070433 127 122 111 T35TIM: .ASCIZ 'WRITE DATA RETRY'S Erase Tape Not Long Enough'
5066 070510 122 105 122 T35EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
5067 070567 124 123 123 T35TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
5068 070644 122 145 167 T35RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
5069 070713 122 101 115 T35RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
5070 070766 124 123 123 T35AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'

```

5071	071035	104	162	151	T35OFL:	.ASCIZ	'Drive 7 Select Failed To Set "OFL" In TSSR'
5072	071110	124	123	123	T35WDD:	.ASCIZ	'TSSR Not Correct After REREAD DATA Command, SWB B't Set'
5073	071200	124	123	123	T35WDC:	.ASCIZ	'TSSR Not Correct After REREAD DATA Command'
5074	071253	103	126	103	T35VCK:	.ASCIZ	'CVC Set, Didn't Reset VCK In Message Buffer'
5075	071326	124	123	102	T35BA:	.ASCIZ	'TSBA Not Correct After REREAD DATA Command'
5076	071401	127	122	111	T35WSS:	.ASCIZ	'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
5077	071470	122	145	141	T35LON:	.ASCIZ	'Reading Long Record Failed To Set RLL Bit In XST0'
5078	071552	122	145	141	T35LOP:	.ASCIZ	'Reading Long Record Failed To Set RLS Bit In XST0'
5079	071634	122	145	163	T35PBP:	.ASCIZ	'Residual Byte Count Incorrect After Short Record Read'
5080	071722	122	145	141	T35TRL:	.ASCIZ	'Reading Long Record Failed To Give Tape Status Alert'
5081	072010	127	122	111	T35NEF:	.ASCIZ	'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
5082	072106	124	123	123	T35SCF:	.ASCIZ	'TSSR Not Correct After SPACE RECORDS Command'
5083	072163	124	123	123	T35TSA:	.ASCIZ	'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
5084	072245	124	123	123	T35WRF:	.ASCIZ	'TSSR Not Correct After WRITE DATA RETRY Command'
5085	072325	104	141	164	T35DTA:	.ASCIZ	'Data Compare Error, Data Read From Tape Not Equal To Written'
5096	072422	124	123	123	T35SSR:	.ASCIZ	'TSSR Incorrect After WRITE MISCELLANEOUS Command'
5087	072503	115	117	124	T35MOT:	.ASCIZ	'MOT Bit (XST0) Not Set During Rewind (Extended Features Mode)'
5088	072601	111	156	164	T35INT:	.ASCIZ	'Interrupt Received After REWIND Command (IE Bit Not Set)'
5089	072672	117	120	115	T35OPM:	.ASCIZ	'OPM Bit (XST2) Not Set During Rewind (Extended Features Mode)'
5090	072770	124	123	123	T35RWE:	.ASCIZ	'TSSR Incorrect After Extended Features REWIND Command'
5091	073056	116	157	040	T35NIN:	.ASCIZ	'No Interrupt Detected After REWIND IMMEDIATE'
5092	073133	105	170	164	TST35ID:	.ASCIZ	'Extended Mode Functions'
5093						.EVEN	
5094							
5095							
5096							
5097							
5098							
5099							
5100							
5101	073164				T35RES1:		
5102	073164				SAVREG		;SAVE THE REGISTERS
5103	073170	012701	067370		MOV	#T35PACKET,R1	;START OF THE PACKET
5104	073174	012721	100004		MOV	#100004,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK,
5105	073200	012721	067400		MOV	#T35DATA,(R1)+	;ADDRESS OF CHARAISTICS DATA BLOCK
5106	073204	005021			CLR	(R1)+	;EXTENDED ADDRESS
5107	073206	012721	000012		MOV	#10.,(R1)+	;SIZE OF DATA BLOCK IN BYTES
5108	073212	012721	067412		MOV	#T35BFR,(R1)+	;ADDRESS OF MESSAGE BUFFER
5109	073216	005021			CLR	(R1)+	
5110	073220	012721	000024		MOV	#20.,(R1)+	;LENGTH OF MESSAGE BUFFER
5111	073224	005021			CLR	(R1)+	
5112	073226	012711	000000		MOV	#0,(R1)	;SELECT DRIVE ZERO
5113	073232	012702	000030		MOV	#24.,R2	;NUMBER OF LOCATIONS TO BE CLEARED
5114	073236	012762	177777	067412	MOV	#177777,T35BFR(R2)	;ALL ONES TO MESSAGE BUFFER
5115	073244	005742			TST	-(R2)	;NEXT LOCATION
5116	073246	022702	000000		CMP	#0,R2	;AT END OF LOOP YET
5117	073252	001371			BNE	64\$;KEEP GOING UNTIL DONE
5118	073254	000207			RTS	PC	;RETURN
5119							
5120	073256				T35RT2:		
5121	073256				SAVREG		;SAVE THE REGISTERS
5122	073262	012701	067500		MOV	#T35PK2,R1	;START OF THE PACKET
5123	073266	012721	100006		MOV	#100006,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK,
5124	073272	012721	067520		MOV	#T35BF2,(R1)+	;ADDRESS OF DATA BLOCK
5125	073276	005021			CLR	(R1)+	;EXTENDED ADDRESS
5126	073300	012721	000006		MOV	#6.,(R1)+	;SIZE OF DATA BLOCK IN BYTES
5127	073304	005021			CLR	(R1)+	

```

5128 073306 012701 067520      MOV      #T35BF2,R1      ;POINT TO DATA SEL AREA
5129 073312 005021              CLR      (R1)+
5130 073314 005011              CLR      (R1)+
5131 073316 000207              RTS      PC              ;RETURN
5132 073320                      T35RT3:
5133 073320                      SAVREG      ;SAVE REGISTERS
5134 073324 012701 067510      MOV      #T35PK3,R1      ;SET UP POINTER ADDRESS
5135 073330 005021              CLR      (R1)+          ;COMMAND SPACE
5136 073332 005021              CLR      (R1)+          ;ADDRESS OF DATA BLOCK
5137 073334 005021              CLR      (R1)+          ;EXTENDED ADDRESS
5138 073336 005011              CLR      (R1)+          ;SIZE OF DATA TRANSFER BLOCK
5139 073340 000207              RTS      PC              ;RETURN
5140 073342                      ENDTST
      073342                      L10063:
      073342 104401                      TRAP      C$ETST

5141                      .SBTTL TEST 8 RECORD BUFFERING
5142
5143
5144
5145
5146
5147
5148
5149
5150
5151
5152
5153
5154
5155
5156
5157
5158
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5180
5181
5182

```

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.

5282	073700				50:	CKLOOP			;LOOP IF SELECTED		
	073700	104406								TRAP	C:CLP1
5283	073702	012737	003720	075706		MOV	#2000.,T36SZ		;SET UP RECORD SIZE		
5284	073710	013737	003116	075702		MOV	FREE,T36WB		;ADDRESS OF WRITE BUFFER		
5285	073716	012737	140005	075700		MOV	#140005,T36PK3		;WRITE DATA,ACK,CVC=1 COMMAND		
5286	073724	012704	015700			MOV	#T36PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
5287	073730	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND		
5288	073734	004737	016330			JSR	PC,WAITF		;WAIT FOR SSR TO SET		
5289	073740	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
5290	073744	012702	000200			MOV	#SSR,R2		;SET UP EXPECTED		
5291	073750	020102				CMP	R1,R2		;ARE THEY EQUAL		
5292	073752	001406				BEQ	60:		;BR, IF OK		
5293	073754	005237	002214			INC	FATFLG		;ERROR COUNT		
5297	073760					ERRHRD	ERRNO,WRTErr,PKTSSR		;TSSR INCORRECT AFTER READ DATA		
	073760	104456								TRAP	C:ERHRD
	073762	001446								.WORD	806
	073764	005107								.WORD	WRTErr
	073766	012126								.WORD	PKTSSR
5298	073770				60:	CKLOOP			;LOOP IF SELECTED		
	073770	104406								TRAP	C:CLP1
5299	073772	012737	000005	075732		MOV	#05.,T36DLY		;25 APR-83 REV B	DELAY FOR TAPE TO STOP	
5300	074000				70:	DELAY	1		;25-APR-83 REV B	DELAY ROUTINE CALL	
	074000	012727	000001							MOV	#1.(PC).
	074004	000000								.WORD	0
	074006	013727	002116							MOV	L%DLY.(PC).
	074012	000000								.WORD	0
	074014	005367	177772							DEC	-6(PC)
	074020	001375								BNE	. 4
	074022	005367	177756							DEC	22(PC)
	074026	001367								BNE	.-20
5301	074030	005337	075732			DEC	T36DLY		;BUMP COUNTER DOWN		
5302	074034	001361				BNE	70:		;BR, IF MORE DELAY TO GO		
5303	074036	012737	006642	075706		MOV	#3490.,T36SZ		;SET SIZE OF TRANSFER		
5304	074044	012737	140005	075700		MOV	#140005,T36PK3		;WRITE DATA,ACK,CVC=1 COMMAND		
5305	074052	012704	075700			MOV	#T36PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
5306	074056	005037	075726			CLR	T36CNT		;CLEAR COUNTER		
5307	074062	012737	001750	075732		MOV	#1000.,T36DLY		;SET DROP DEAD COUNTER VALUE		
5308	074070	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND		
5309	074074	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
5310	074100	032701	000200		80:	BIT	#SSR,R1		;CHECK FOR SSR SET		
5311	074104	001021				BNE	90:		;BR, IF SSR IS SET		
5312	074106	005237	075726			INC	T36CNT		;BUMP CYCLE COUNTER		
5313	074112					DELAY	1		;CUT NUMBER OF LOOPS DOWN		
	074112	012727	000001							MOV	#1.(PC).
	074116	000000								.WORD	0
	074120	013727	002116							MOV	L%DLY.(PC).
	074124	000000								.WORD	0
	074126	005367	177772							DEC	6(PC)
	074132	001375								BNE	. 4
	074134	005367	177756							DEC	22(PC)
	074140	001367								BNE	.-20
5314	074142	005337	075732			DEC	T36DLY		;BUMP DROP DEAD COUNTER		
5315	074146	001352				BNE	80:		;BR, IF THERE IS STILL TIME		
5316	074150	012702	000200		90:	MOV	#SSR,R2		;SET UP EXPECTED		
5317	074154	020102				CMP	R1,R2		;ARE THEY EQUAL		
5318	074156	001406				BEQ	100:		;BR, IF OK		
5319	074160	005237	002214			INC	FATFLG		;ERROR COUNT		

5323	074164			ERRHRD	ERRNO,T36WDE,PKTSSR		;TSSR INCORRECT AFTER READ DATA
	074164	104456					TRAP C:ERHRD
	074166	001447					.WORD 807
	074170	076563					.WORD T36WDE
	074172	012126					.WORD PKTSSR
5324	074174			100:	CKLOOP		;LOOP IF SELECTED
	074174	104406					TRAP C:CLP1
5325	074176	013737	002174	075600	MOV UNITN,T36DSW		;SET UP DRIVE NUMBER
5326	074204	052737	000010	075600	BIS #BIT3,T36DSW		;25 APR-83 REV B - TURN OFF BUFFERING
5327	074212	012704	075560		MOV #T36PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS
5328	074216	004737	010742		JSR PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS
5329	074222	103407			BCS 110:		;BR, IF COMMAND ISSUED OK
5330	074224	005237	002214		INC FATFLG		;ERROR COUNT
5334	074230	010001			MOV R0,R1		;SAVE CONTENTS OF TSSR
5335	074232				ERRHRD ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTIC FAILED
	074232	104456					TRAP C:ERHRD
	074234	001450					.WORD 808
	074236	005052					.WORD WRTMSG
	074240	012114					.WORD SFIMSG
5336	074242			110:	CKLOOP		;LOOP IF SELECTED
	074242	104406					TRAP C:CLP1
5337	074244	012737	006642	075706	MOV #3490.,T36S7		;SET SIZE OF TRANSFER
5338	074252	012737	140005	075700	MOV #140005,T36PK		;WRITE DATA,ACK,CVC=1 COMMAND
5339	074260	012704	075700		MOV #T36PK3,R4		;SET UP R4 WITH PACKET ADDRESS
5340	074264	005037	075730		CLR T36CNU		;CLEAR COUNTER
5341	074270	012737	001750	075732	MOV #1000.,T36DLY		;SET DROP DEAD COUNTER VALUE
5342	074276	010465	000000		MOV R4,T36DB(R5)		;ISSUE COMMAND
5343	074302	016501	000002	120:	MOV TSSR(R5),R1		;GET TSSR CONTENTS
5344	074306	032701	000200		BIT #SSR,R1		;CHECK FOR SSR SET
5345	074312	001021			BNE 130:		;BR, IF SSR IS SET
5346	074314	005237	075730		INC T36CNU		;BUMP CYCLE COUNTER
5347	074320				DELAY 1		;CUT NUMBER OF LOOPS DOWN
	074320	012727	000001				MOV #1,(PC).
	074324	000000					.WORD 0
	074326	013727	002116				MOV L:DLY,(PC).
	074332	000000					.WORD 0
	074334	005367	177772				DEC -6(PC)
	074340	001375					BNE .4
	074342	005367	177756				DEC -22(PC)
	074346	001367					BNE -.20
5348	074350	005337	075732		DEC T36DLY		;BUMP DROP DEAD COUNTER
5349	074354	001352			BNE 120:		;BR, IF THERE IS STILL TIME
5350	074356	012702	000200	130:	MOV #SSR,R2		;SET UP EXPECTED
5351	074362	020102			CMP R1,R2		;ARE THEY EQUAL
5352	074364	001406			BEQ 140:		;BR, IF OK
5353	074366	005237	002214		INC FATFLG		;ERROR COUNT
5357	074372				ERRHRD ERRNO,WRTERR,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA
	074372	104456					TRAP C:ERHRD
	074374	001451					.WORD 809
	074376	005107					.WORD WRTERR
	074400	012126					.WORD PKTSSR
5358	074402			140:	CKLOOP		;LOOP IF SELECTED
	074402	104406					TRAP C:CLP1
5359	074404	013701	075726		MOV T36CNT,R1		;GET FIRST COUNTER
5360	074410	013702	075730		MOV T36CNU,R2		;GET SECOND COUNTER
5361	074414	020102			CMP R1,R2		;25 APR 83 REV B - COMPARE EM
5362	074416	003406			BLE 300:		;BR, IF VALUES ARE CORRECT (OK)


```

5363 074420 005237 002214          INC    FATFLG          ;ERROR COUNT
5367 074424          ERRHRD  ERRNO,T36NAS,EXPREC ;TAPE NOT AT CORRECT SPEED
      074424 104456          TRAP    C:ERHRD
      074426 001452          .WORD  810
      074430 075734          .WORD  T36NAS
      074432 015554          .WORD  EXPREC
5368 074434          300$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C:CLP1
      074434 104406          ENDSUB
5369 074436          L10071:
      074436 104403          TRAP    C:ESUB
5370 074440 023727 002214 000017    CMP    FATFLG,015.    ;IS ERROR COUNT AT 25
5371 074446 103402          BLO    999$          ;BR, IF LESS THAN 25
5372 074450 004737 017262          JSR    PC,CKDROP     ;TRY TO DROP THE UNIT
5373 074454          999$:
5374          ;
5375          ;
5376          ;TEST 8, SUBTEST 2
5377          ;
5378          ;
5379          ;
5380          ;
5381          ;
5382          ;
5383          ;
5384          ;
5385          ;
5386          ;
5387          ;
5388          ;
5389          ;
5390          ;
5391          ;
5392          ;
5393          ;
5394          ;
5395          ;
5396          ;
5397          ;
5398          ;
5399          ;
5400          ;
5401          ;
5402          ;
5403          ;
5404          ;
5405          ;
5406          ;
5407          ;
5408          ;
5409          ;
5410          ;
5411          ;
5412          ;
5413          ;
5414          ;
5415          ;
    
```

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.

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 8: RECORD BUFFERING

SEQ 203

5546	075342	012704	075700		MOV	#T36PK3,R4		;SET UP R4 WITH PACKET ADDRESS	
5547	075346	005037	075730		CLR	T36CNU		;CLEAR COUNTER	
5548	075352	012737	001750	075732	MOV	#1000.,T36DLY		;SET DROP DEAD COUNTER VALUE	
5549	075360	010465	000000		MOV	R4,TSD8(R5)		;ISSUE COMMAND	
5550	075364	016501	000002	120\$:	MOV	TSSR(R5),R1		;GET TSSR CONTENTS	
5551	075370	032701	000200		BIT	#SSR,R1		;CHECK FOR SSR SET	
5552	075374	001021			BNE	130\$;BR, IF SSR IS SET	
5553	075376	005237	075730		INC	T36CNU		;BUMP CYCLE COUNTER	
5554	075402				DELAY	1		;CUT NUMBER OF LOOPS DOWN	
	075402	012727	000001						MOV #1,(PC).
	075406	000000							.WORD 0
	075410	013727	002116						MOV L\$DLY,(PC).
	075414	000000							.WORD 0
	075416	005367	177772						DEC 6(PC)
	075422	001375							BNE . 4
	075424	005367	177756						DEC 22(PC)
	075430	001367							BNE . 20
5555	075432	005337	075732		DEC	T36DLY		;BUMP DROP DEAD COUNTER	
5556	075436	001352			BNE	120\$;BR, IF THERE IS STILL TIME	
5557	075440	012702	000200	130\$:	MOV	#SSR,R2		;SET UP EXPECTED	
5558	075444	020102			CMP	R1,R2		;ARE THEY EQUAL	
5559	075446	001406			BEQ	140\$;BR, IF OK	
5560	075450	005237	002214		INC	FATFLG		;ERROR COUNT	
5564	075454				ERRHRD	ERRNO,WRERR,PKTSSR		;SSR INCORRECT AFTER WRITE DATA	
	075454	104456							TRAP C\$ERHRD
	075456	001463							.WORD 819
	075460	005107							.WORD WRERR
	075462	012126							.WORD PKTSSR
5565	075464			140\$:	CKLOOP			;LOOP IF SELECTED	
	075464	104406							TRAP C\$CLP1
5566	075466	013701	075726		MOV	T36CNT,R1		;GET FIRST COUNTER	
5567	075472	013702	075730		MOV	T36CNU,R2		;GET SECOND COUNTER	
5568	075476	020102			CMP	R1,R2		;25 APR-83 REV B COMPARE EM	
5569	075500	003406			BLE	300\$;BR, IF VALUES ARE CORRECT (OK)	
5570	075502	005237	002214		INC	FA,FLG		;ERROR COUNT	
5574	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
5575	075516			300\$:	CKLOOP			;LOOP IF SELECTED	
	075516	104406							TRAP C\$CLP1
5576	075520				ENDSUB				
	075520	104403							L10072: TRAP C\$ESUB
5577	075522	023727	002214	000017	CMP	FATFLG,#15.		;IS ERROR COUNT AT 25	
5578	075530	103402			BLO	999\$;BR, IF LESS THAN 25	
5579	075532	004737	017262		JSI:	PC,CKDROP		;TRY TO DROP THE UNIT	
5580	075536			999\$:					
5581				:					
5582				:					
5583				:					
5584	075536	004737	016536		JSR	PC,TSTLOOP		;DO WE NEED TO ITERATE TEST	
5585	075542	103002			BCC	163\$;BR, IF NO LOOP REQUIRED	
5586	075544	000137	073400		JMP	T36LOOP		;EXECUTE AGAIN	
5587	075550			163\$:					
5588	075550				EXIT	TST		;ALL DONE THIS TEST	

TRAP C\$EXIT
.WORD L10070 .

```

075550 104432
075552 003344
5589
5590
5591
5593 075560
5595 075560
5596 075560 100704
5597 075562 075570
5598 075564 000000
5599 075566 000012
5600 075570
5601 075570 075602
5602 075572 000000
5603 075574 000024
5604 075576 000000
5605 075600 000000
5606 075602
5607
5608
5609
5611 075670
5613 075670
5614 075670 100006
5615 075672 075710
5616 075674 000000
5617 075676 000006
5618
5622 075700
5623 075700 100005
5624 075702
5625 075702 003116
5626 075704 000000
5627 075706 000000
5628
5629
5630
5631
5632 075710
5633 075710 010
5634 075711 200
5635 075712 000000
5636 075714 000000
5637
5638
5639
5640
5641
5642 075716 100205
5643 075720 100605
5644 075722 102205
5645 075724 177777
5646
5647
5648 075726 000000
5649 075730 000000
5650 075732 000000

```

```

;*
;LOCAL STORAGE FOR THIS TEST
;
;
T36PACKET: .=<.*10>E177770
           .WORD 100004
           .WORD T36DATA1
           .WORD 0
           .WORD 10.
T36DATA:  .WORD T36BFR
           .WORD 0
           .WORD 20.
           .WORD 0
T36DSW:  .WORD 0
T36BFR:  .BLKW 25.
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
;
T36PK2:  .=<.*10>E177770
           .WORD 100006
           .WORD T36BF2
           .WORD 0
           .WORD 6.
T36PK3:  .WORD 100005
T36RB:
T36WB:  .WORD FREE
           .WORD 0
T36SZ:  .WORD 0
           .EVEN
;
;
T36BF2:
T36BS0: .BYTE 10
T36BS1: .BYTE 200
T36S2:  .WORD 0
T36S3:  .WORD 0
;
;
           .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T36RN:  .WORD 100205
T36WDR: .WORD 100605
T36CON: .WORD 102205
           .WORD 177777
;
T36CNT: .WORD 0
T36CNU: .WORD 0
T36DLY: .WORD 0

```

```

;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH . ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;
;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
;
;LENGTH OF MESSAGE BUFFER
;
;SELECT DRIVE 0
;MESSAGE BUFFER
;
;WRITE SUB SYS MEM COMMAND, AND ACK
;ADDRESS OF SELECT BLOCK DATA
;
;SIZE OF DATA PACKET
;
;REREAD COMMAND, AND ACK
;
;ADDRESS OF WRITE BUFFER
;
;SIZE OF BUFFER (EXTENT)
;
;BSEL0 AREA
;BSEL1 AREA
;SEL 2 AREA
;DATA AREA
;
;REREAD DATA (NEXT)
;REREAD DATA RETRY
;WRITE CONTINOUS
;END OF DATA
;
;TAPE TIMER COUNTER STORAGE AREA
;TAPE TIMER COUNTER STORAGE AREA
;DELAY COUNTER

```

```

5651
5652          ;*
5653          ;LOCAL TEXT MESSAGES FOR TEST
5654          ;-
5655 075734    111    155    160 T36NAS: .ASCIZ 'Improper Tape Controller Buffering Speed'
5656 076005    124    141    160 T36WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)
5657 076073    124    123    123 T36RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
5658 076142    122    105    122 T36RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
5659 076237    120    117    123 T36SC:  .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
5660 076321    122    111    102 T36LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
5661 076371    124    123    123 T36WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
5662 076446    111    154    154 T36LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
5663 076527    122    105    122 T36SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
5664 076563    124    123    123 T36WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
5665 076635    124    141    160 T36BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
5666 076730    127    122    111 T36TIM: .ASCIZ 'WRITE DATA RETRY'S Erase Tape Not Long Enough'
5667 077005    122    105    122 T36EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
5668 077064    124    123    123 T36TM:  .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
5669 077141    122    145    167 T36RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
5670 077210    122    101    115 T36RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
5671 077263    124    123    123 T36AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
5672 077332    104    162    151 T36OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
5673 077405    124    123    123 T36WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
5674 077475    124    123    123 T36WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
5675 077550    103    126    103 T36VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
5676 077623    124    123    102 T36BA:  .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
5677 077676    127    122    111 T36WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
5678 077765    122    145    141 T36LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
5679 100047    122    145    141 T36LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
5680 100131    122    145    163 T36PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
5681 100217    122    145    141 T36TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
5682 100305    127    122    111 T36NEF: .ASCIZ 'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
5683 100403    124    123    123 T36SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
5684 100460    124    123    123 T36TSA: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
5685 100542    124    123    123 T36WRF: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command'
5686 100622    104    141    164 T36DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
5687 100717    122    145    143 TST36ID: .ASCIZ 'Record Buffering'
5688          .EVEN
5689          ;*
5690          ;
5691          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5692          ;WRITE SUBSYSTEM MEMORY COMMAND
5693          ;
5694          ;-
5695
5696 100740
5697 100740
5698 100744    012701 075560
5699 100750    012721 100004
5700 100754    012721 075570
5701 100760    005021
5702 100762    012721 000012
5703 100766    012721 075602
5704 100772    005021
5705 100774    012721 000024
5706 101000    005021
5707 101002    012711 000000

T36REST:
    SAVREG
    MOV     #T36PACKET,R1
    MOV     #100004,(R1)
    MOV     #T36DATA,(R1)
    CLR     (R1)
    MOV     #10,(R1)
    MOV     #T36BFR,(R1)
    CLR     (R1)
    MOV     #20,(R1)
    CLR     (R1)
    MOV     #0 (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
;SELECT DRIVE ZERO
    
```

```

5708 101006 012702 000030          MOV    #24.,R2          ;NUMBER OF LOCATIONS TO BE CLEARED
5709 101012 012762 177777 075602 64$: MOV    #177777,T36BFR(R2) ;ALL ONES TO MESSAGE BUFFER
5710 101020 005742                TST    -(R2)          ;NEXT LOCATION
5711 101022 022702 000000          CMP    #0,R2          ;AT END OF LOOP YET
5712 101026 001371                BNE    64$           ;KEEP GOING UNTIL DONE
5713 101030 000207                RTS    PC             ;RETURN
5714
5715 101032                T36RT2:
5716 101032                SAVREG                ;SAVE THE REGISTERS
5717 101036 012701 075670          MOV    #T36PK2,R1    ;START OF THE PACKET
5718 101042 012721 100006          MOV    #100006,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
5719 101046 012721 075710          MOV    #T36BF2,(R1)+ ;ADDRESS OF DATA BLOCK
5720 101052 005021                CLR    (R1)+         ;EXTENDED ADDRESS
5721 101054 012721 000006          MOV    #6.,(R1)+    ;SIZE OF DATA BLOCK IN BYTES
5722 101060 005021                CLR    (R1)+
5723 101062 012701 075710          MOV    #T36BF2,R1   ;POINT TO DATA SEL AREA
5724 101066 005021                CLR    (R1)+
5725 101070 005011                CLR    (R1)
5726 101072 000207                RTS    PC             ;RETURN
5727 101074                T36RT3:
5728 101074                SAVREG                ;SAVE REGISTERS
5729 101100 012701 075700          MOV    #T36PK3,R1   ;SET UP POINTER ADDRESS
5730 101104 005021                CLR    (R1)+         ;COMMAND SPACE
5731 101106 005021                CLR    (R1)+         ;ADDRESS OF DATA BLOCK
5732 101110 005021                CLR    (R1)+         ;EXTENDED ADDRESS
5733 101112 005011                CLR    (R1)         ;SIZE OF DATA TRANSFER BLOCK
5734 101114 000207                RTS    PC             ;RETURN
5735 101116                ENDTST
                    L10070: TRAP    C$ETST
5736 101116 104401
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751 101120                .SBTTL TEST 9: FUNCTION TIMING
                    :+
                    :
                    : THIS TEST VERIFIES THAT THE TAPE TRANSPORT SEEMS TO BE WRITING
5752 101120 012737 006354 002172          MOV    #EPRT1,EPRTSW ;PRIMARY ERROR MESSAGE
5753 101126 004737 017354                JSR    PC,KTOFF      ;TURN KT OFF
5758 101132 012700 105343                MOV    #TST37ID,R0  ;ASCII MESSAGE TO IDENTIFY TEST
5759 101136 004737 01657C                JSR    PC,TSTSETUP  ;DO INITIAL TEST SETUP
5760 101142 012737 000005 002210          MOV    #5,LOOPCNT   ;PERFORM 5 ITERATIONS
5761 101150 005037 102406                CLR    T37CNT       ;CLEAR TAPE RECORD COUNTER
5762
5763
5764
5765
                    :
                    : TEST 9, SUBTEST 1
                    :
    
```


Line	Address	Offset	Label	Instruction	Comment	Trap	Word
	101360	001607					903
	101362	103565					T37RWN
	101364	012126					PKTSSR
5812	101366		301:	CKLOOP	;LOOP IF SELECTED	TRAP	C1CLP1
	101366	104406					
5813	101370	013701	102270	MOV	T37BFR+6,R1		
5814	101374	010102		MOV	R1,R2		
5815	10137	052702	000002	BIS	#BIT1,R2		
5816	10140	020102		CMP	R1,R2		
5817	101404	001406		BEQ	401		
5818	101406	005237	002214	INC	FATFLG		
5822	101412			ERRHRD	ERRNO,T37BOT,EXPREC		
	101412	104456				TRAP	C1ERHRD
	101414	001610				.WORD	904
	101416	103261				.WORD	T37BOT
	101420	015554				.WORD	EXPREC
5823	101422		401:	CKLOOP	;LOOP IF SELECTED	TRAP	C1CLP1
	101422	104406					
5824	101424	012703	000144	MOV	#100.,R3		
5825	101430	013737	003116	MOV	FREE,T37WB		
5826	101436	012737	140005	MOV	#140005,T37PK3		
5827	101444	012704	102360	MOV	#T37PK3,R4		
5828	101450	012737	001130	MOV	#600.,T37S4		
5829	101456	010465	000000	MOV	R4,TSD8(R5)		
5830	101462	004737	016330	JSR	PC,WAITF		
5831	101466	016501	000002	MOV	TSSR(R5),R1		
5832	101472	012702	000200	MOV	#SSR,R2		
5833	101476	020102		CMP	R1,R2		
5834	101500	001406		BEQ	701		
5835	101502	005237	002214	INC	FATFLG		
5839	101506			ERRHRD	ERRNO,T37WDC,PKTSSR		
	101506	104456				TRAP	C1ERHRD
	101510	001611				.WORD	905
	101512	104121				.WORD	T37WDC
	101514	012126				.WORD	PKTSSR
5840	101516		701:	CKLOOP	;LOOP IF SELECTED	TRAP	C1CLP1
	101516	104406					
5841	101520	005303		DEC	R3		
5842	101522	001345		BNE	651		
5843	101524	004737	011074	JSR	PC,REWIND		
5844	101530	103411		BCS	1301		
5845	101532	016501	000002	MOV	TSSR(R5),R1		
5846	101536	010004		MOV	R0,R4		
5847	101540	005237	002214	INC	FATFLG		
5851	101544			ERRHRD	ERRNO,T37RWN,PKTSSR		
	101544	104456				TRAP	C1ERHRD
	101546	001612				.WORD	906
	101550	103565				.WORD	T37RWN
	101552	012126				.WORD	PKTSSR
5852	101554		1301:	CKLOOP	;LOOP IF SELECTED	TRAP	C1CLP1
	101554	104406					
5853	101556	013701	102270	MOV	T37BFR+6,R1		
5854	101562	010102		MOV	R1,R2		
5855	101564	052702	000002	BIS	#BIT1,R2		
5856	101570	020102		CMP	R1,R2		
5857	101572	001406		BEQ	1401		
5858	101574	005237	002214	INC	FATFLG		

5862	101600				ERRHRD	ERRNO,T37BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND		
	101600	104456						TRAP	C\$ERHRD	
	101602	001613						.WORD	907	
	101604	103261						.WORD	T37BOT	
	101606	015554						.WORD	EXPREC	
5863	101610			1401:	CKLOOP			;LOOP IF SELECTED		
	101610	104406						TRAP	C\$CLP1	
5864	101612	012704	102360		MOV	@T37PK3,R4		;SET UP PACKET ADDRESS		
5865	101616	012737	000037	102362	MOV	@31.,T37RB		;SET UP RECORDS TO SPACE OVER		
5866	101624	012737	140010	102360	MOV	@140010,T37PK3		;ACK,CVC=1,SPACE FORWARD COMMAND		
5867	101632	010465	000000		1501:	MOV	R4,TSD8(R5)	;ISSUE COMMAND		
5868	101636	005237	102406		1521:	INC	T37CNT	;BUMP TIMER		
5869	101642				DELAY	1		;DELAY ABOUT 100US		
	101642	012727	000001					MOV	@1,(PC).	
	101646	000000						.WORD	0	
	101650	013727	002116					MOV	L\$DLY,(PC).	
	101654	000000						.WORD	0	
	101656	005367	177772					DEC	6(PC)	
	101662	001375						BNE	.-4	
	101664	005367	177756					DEC	22(PC)	
	101670	001367						BNE	.-20	
5870	101672	016501	000002		MOV	TSSR(R5),R1		;GET TSSR		
5871	101676	032701	000200		BIT	@SSR,R1		;CHECK FOR TSSR'S SSR SET		
5872	101702	001755			BEQ	1521		;KEEP COUNTING UNTIL SET		
5873	101704	012702	000200		MOV	@SSR,R2		;SET UP EXPECTED		
5874	101710	020201			CMP	R2,R1		;WAS EVERYTHING OK		
5875	101712	001406			BEQ	1601		;BR, IF ALL IS WELL		
5876	101714	005237	002214		INC	FATFLG		;ERROR COUNT		
5880	101720				ERRHRD	ERRNO,T37SCF,PKTSSR		;SPACE FORWARD DIDN T WORK OUT		
	101720	104456						TRAP	C\$ERHRD	
	101722	001614						.WORD	908	
	101724	105027						.WORD	T37SCF	
	101726	012126						.WORD	PKTSSR	
5881	101730			1601:	CKLOOP			;LOOP IF SELECTED		
	101730	104406						TRAP	C\$CLP1	
5882	101732	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND		
5883	101736	103411			BCS	1701		;BR, IF NO PROBLEM		
5884	101740	010004			MOV	R0,R4		;GET PACKET ADDRESS		
5885	101742	016501	000002		MOV	TSSR(R5),R1		;GET STATUS FROM TSSR		
5886	101746	005237	002214		INC	FATFLG		;ERROR COUNT		
5890	101752				ERRHRD	ERRNO,T37RWN,PKTSSR		;REWIND NOT ACCEPTED		
	101752	104456						TRAP	C\$ERHRD	
	101754	001615						.WORD	909	
	101756	103565						.WORD	T37RWN	
	101760	012126						.WORD	PKTSSR	
5891	101762			1701:	CKLOOP			;LOOP IF SELECTED		
	101762	104406						TRAP	C\$CLP1	
5892	101764	013701	102270		MOV	T37BFR*6,R1		;PICK UP XSTO		
5893	101770	010102			MOV	R1,R2		;SET UP EXPECTED		
5894	101772	052702	000002		BIS	@BIT1,R2		;SET BOT BIT IN EXPECTED		
5895	101776	020102			CMP	R1,R2		;DOES EXP = REC'D		
5896	102000	001406			BEQ	1751		;BR, IF EQUAL (OK)		
5897	102002	005237	002214		INC	FATFLG		;ERROR COUNT		
5901	102006				ERRHRD	ERRNO,T37BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND		
	102006	104456						TRAP	C\$ERHRD	
	102010	001616						.WORD	910	
	102012	103261						.WORD	T37BOT	

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
TEST 9: FUNCTION TIMING

F 1

SEQ 211

5943	102226	000137	101154	MP	T37LOOP	;EXECUTE AGAIN
5944	102232			1638:		
5945	102232			EXIT	TST	;ALL DONE THIS TEST
	102232	104432				TRAP C:EXIT
	102234	003306				.WORD L10073 .
5946				;*		
5947				;LOCAL STORAGE FOR THIS TEST		
5948				;		
5950		102240		;		
5952	102240			T37PACKET: .<<.10>E177770		;COMMAND PACKET FOR TEST
5953	102240	100004		.WORD 100004		;WRITE CHARACTERISTICS COMMAND, WITH . ACK
5954	102242	102250		.WORD T37DATA		;ADDRESS OF CHARACTERISTICS BLOCK
5955	102244	000000		.WORD 0		
5956	102246	000012		.WORD 10.		;STARTING VALUE OF BLOCK SIZE
5957	102250			T37DATA:		;CHARACTERISTICS DATA BLOCK
5958	102250	102262		.WORD T37BFR		;ADDRESS OF MESSAGE BUFFER
5959	102252	000000		.WORD 0		
5960	102254	000024		.WORD 20.		;LENGTH OF MESSAGE BUFFER
5961	102256	000000		.WORD 0		
5962	102260	000000		T37DSW: .WORD 0		;SELECT DRIVE 0
5963	102262			T37BFR: .BLKW 25.		;MESSAGE BUFFER
5964				;		
5965				;WRITE SUBSYSTEM MEMORY COMMAND PACKET		
5966				;		
5968		102350		;		
5970	102350			T37PK2: .<<.10>E177770		
5971	102350	100006		.WORD 100006		;WRITE SUB SYS MEM COMMAND, AND ACK
5972	102352	102370		.WORD T37BF2		;ADDRESS OF SELECT BLOCK DATA
5973	102354	000000		.WORD 0		
5974	102356	000006		.WORD 6.		;SIZE OF DATA PACKET
5975						
5979	102360			T37PK3:		
5980	102360	100005		.WORD 100005		;REREAD COMMAND, AND ACK
5981	102362			T37RB:		
5982	102362	003116		T37WB: .WORD FREE		;ADDRESS OF WRITE BUFFER
5983	102364	000000		.WORD 0		
5984	102366	000000		T37SZ: .WORD 0		;SIZE OF BUFFER (EXTENT)
5985				.EVEN		
5986				;		
5987				;		
5988				;		
5989	102370			T37BF2:		
5990	102370	010		T37BS0: .BYTE 10		;BSEL0 AREA
5991	102371	200		T37BS1: .BYTE 200		;BSEL1 AREA
5992	102372	000000		T37S2: .WORD 0		;SEL 2 AREA
5993	102374	000000		T37S3: .WORD 0		;DATA AREA
5994				;		
5995				;		
5996				.EVEN		
5997				;TAPE MOTION PACKET COMMAND VALUES		
5998						
5999	102376	100205		T37RN: .WORD 100205		;REREAD DATA (NEXT)
6000	102400	100605		T37WR: .WORD 100605		;REREAD DATA RETRY
6001	102402	102205		T37CON: .WORD 102205		;WRITE CONTINOUS
6002	102404	177777		.WORD 177777		;END OF DATA
6003						
6004				;		

```

6005 102406 000000          T37CNT: .WORD 0          ;TAPE TIMER COUNTER STORAGE AREA
6006 102410 000000          T37CNU: .WORD 0          ;TAPE TIMER COUNTER STORAGE AREA
6007 102412 000000          T37DLY: .WORD 0          ;DELAY COUNTER
6008
6009                          ;*
6010                          ;LOCAL TEXT MESSAGES FOR TEST
6011                          ;-
6012 102414      124      141      160 T37WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
6013 102502      124      123      123 T37RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
6014 102551      122      105      122 T37RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
6015 102646      120      117      123 T37SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
6016 102730      122      111      102 T37LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
6017 103000      124      123      123 T37WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
6018 103055      111      154      154 T37LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
6019 103136      122      105      122 T375^R: .ASCIZ 'REREAD COMMAND Not Accepted'
6020 103172      124      123      123 T37WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command, At BOT'
6021 103261      124      141      160 T37BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
6022 103354      127      122      111 T37TIM: .ASCIZ 'WRITE DATA RETRY'S Erase Tape Not Long Enough'
6023 103431      122      105      122 T37EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
6024 103510      124      123      123 T37TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
6025 103565      122      145      167 T37RMN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
6026 103634      122      101      115 T37RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
6027 103707      124      123      123 T37AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
6028 103756      104      162      151 T37OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
6029 104031      124      123      123 T37WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
6030 104121      124      123      123 T37WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
6031 104174      103      126      103 T37VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
6032 104247      124      123      102 T37BA: .ASCIZ 'TSBA Not Correct After RERFAD DATA Command'
6033 104322      127      122      111 T37WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
6034 104411      122      145      141 T37LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
6035 104473      122      145      141 T37LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
6036 104555      122      145      163 T37PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
6037 104643      122      145      141 T37TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
6038 104731      127      122      111 T37NEF: .ASCIZ 'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
6039 105027      124      123      123 T37SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
6040 105104      124      123      123 T37TSA: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
6041 105166      124      123      123 T37WRF: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command'
6042 105246      104      141      164 T37DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
6043 105343      106      165      156 T37ID: .ASCIZ 'Function Timing'
6044
6045                          ;*
6046                          ;
6047                          ;ROUTINE TO RESTORE COMMAND PACKET TO START UP (DEFAULT) VALUES
6048                          ;WRITE SUBSYSTEM MEMORY COMMAND
6049                          ;
6050                          ;-
6051
6052 105364          T37REST:
6053 105364          SAVREG
6054 105370      012701 102240          MOV #T37PACKET,R1          ;SAVE THE REGISTERS
6055 105374      012721 100004          MOV #100004,(R1)          ;START OF THE PACKET
6056 105400      012721 102250          MOV #T37DATA,(R1)          ;WRITE SUBSYSTEM MEM. WITH ACK,
6057 105404      005021                    CLR (R1)          ;ADDRESS OF CHARAISTICS DATA BLOCK
6058 105406      012721 000012          MOV #10,(R1)          ;EXTENDED ADDRESS
6059 105412      012721 102262          MOV #T37BFR,(R1)          ;SIZE OF DATA BLOCK IN BYTES
6060 105416      005021                    CLR (R1)          ;ADDRESS OF MESSAGE BUFFER
6061 105420      012721 000024          MOV #20,(R1)          ;LENGTH OF MESSAGE BUFFER

```

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 06 FEB 84 18:04
 TEST 9: FUNCTION TIMING

SEQ 213

6062	105424	005021		CLR	(R1).	
6063	105426	012711	000000	MOV	#0,(R1)	;SELECT DRIVE ZERO
6064	105432	012702	000030	MOV	#24.,R2	;NUMBER OF LOCATIONS TO BE CLEARED
6065	105436	012762	177777	MOV	#177777,T37BFR(R2)	;ALL ONES TO MESSAGE BUFFER
6066	105444	005742		TST	-(R2)	;NEXT LOCATION
6067	105446	022702	000000	CMP	#0,R2	;AT END OF LOOP YET
6068	105452	001371		BNE	64#	;KEEP GOING UNTIL DONE
6069	105454	000207		RTS	PC	;RETURN
6070						
6071	105456					
6072	105456			T37RT2:	SAVREG	;SAVE THE REGISTERS
6073	105462	012701	102350	MOV	#T37PK2,R1	;START OF THE PACKET
6074	105466	012721	100006	MOV	#100006,(R1).	;WRITE SUBSYSTEM MEM. WITH ACK.
6075	105472	012721	102370	MOV	#T37BF2,(R1).	;ADDRESS OF DATA BLOCK
6076	105476	005021		CLR	(R1).	;EXTENDED ADDRESS
6077	105500	012721	000006	MOV	#6.,(R1).	;SIZE OF DATA BLOCK IN BYTES
6078	105504	005021		CLR	(R1).	
6079	105506	012701	102370	MOV	#T37BF2,R1	;POINT TO DATA SEL AREA
6080	105512	005021		CLR	(R1).	
6081	105514	005011		CLR	(R1)	
6082	105516	000207		RTS	PC	;RETURN
6083	105520					
6084	105520			T37RT3:	SAVREG	;SAVE REGISTERS
6085	105524	012701	102360	MOV	#T37PK3,R1	;SET UP POINTER ADDRESS
6086	105530	005021		CLR	(R1).	;COMMAND SPACE
6087	105532	005021		CLR	(R1).	;ADDRESS OF DATA BLOCK
6088	105534	005021		CLR	(R1).	;EXTENDED ADDRESS
6089	105536	005011		CLR	(R1)	;SIZE OF DATA TRANSFER BLOCK
6090	105540	000207		RTS	PC	;RETURN
6091	105542			ENDTST		
	105542					
	105542	104401				L10073: TRAP C#ETST
6092	105544			ENDMOD		

```

1          .TITLE  TSV6  PARAMETER CODING
7
12
18
19 105544  BGNMOD  TSV6
105544  TSV6::
20
21          .SBTTL  HARDWARE PARAMETER CODING SECTION
22
23          ;**
24          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
25          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
26          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
27          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
28          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
29          ; WITH THE OPERATOR.
30          ;
31 105544  BGNHRD
105544 000010  .WORD  L10075-L$HARD/2
105546  L$HARD::
32
33 105546  GPRMA  HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
105546 000031  .WORD  T$CODE
105550 105566  .WORD  HPM1
105552 160010  .WORD  T$LLOLIM
105554 177776  .WORD  T$HILIM
34 105556  GPRMA  HPM2,2,0,0,776,YES      ;GET VECTOR ADDRESS.
105556 001031  .WORD  T$CODE
105560 105622  .WORD  HPM2
105562 000000  .WORD  T$LLOLIM
105564 000776  .WORD  T$HILIM
35          ;GPRMD  HPM3,4,0,340,0,7,YES      ;GET INTERRUPT PRIORITY.
36 105566  ENDRD
          .EVEN
          L10075:
37 105566 104 105 126 HPM1:  .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB)
38 105622 111 116 124 HPM2:  .ASCIZ  'INTERRUPT VECTOR
39 105646 111 116 124 HPM3:  .ASCIZ  'INTERRUPT PRIORITY
40          .EVEN
    
```


ADSSR	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	G	
ASSEMB	000010		C\$BSEG	000004	DEVONL	023243	F\$AUTO	000020		INTREC	002216	G	
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	G	
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		I\$AU	000041		
BIT03	000010	G	C\$DU	000053	EF.PWR	000034	F\$PROT	000021		I\$AUTO	000041		
BIT04	000020	G	C\$EDIT	000003	EF.RES	000037	F\$PWR	000017		I\$CLN	000041		
BIT05	000040	G	C\$ERDF	000055	EF.STA	000040	F\$RPT	000012		I\$DU	000041		
BIT06	000100	G	C\$ERHR	000056	EMAXDU	017057	F\$SEG	000003		I\$HRD	000041		
BIT07	000200	G	C\$ERRO	000060	EN	000000	F\$SOFT	000005		I\$INIT	000041		
BIT08	000400	G	C\$ERSF	000054	ENAIN	016232	F\$SRV	000010		I\$MOD	000041		
BIT09	001000	G	C\$ERSO	000057	ENVIRN	020710	F\$SUB	000002		I\$MSG	000041		
BIT1	000002	G	C\$ESCA	000010	EPRTSW	002172	F\$SW	000014		I\$PROT	000040		
BIT10	002000	G	C\$ESEG	000005	EPRT1	006354	F\$TEST	000001		I\$PTAB	000041		
BIT11	004000	G	C\$ESUB	000003	EPRT2	006413	GDDAT	003152	G	I\$PWR	000041		
BIT12	010000	G	C\$ETST	000001	ERCM	012013	GERRMA	002166	G	I\$RPT	000041		
BIT13	020000	G	C\$EXIT	000032	ERRHI	002230	GETPAT	020254	G	I\$SEG	000041		
BIT14	040000	G	C\$GETB	000026	ERRK	017036	GETSEL	020336	G	I\$SETU	000041		
BIT15	100000	G	C\$GETW	000027	ERRLO	002232	G\$CNT0	000200		I\$SFT	000041		
BIT2	000004	G	C\$GMAN	000043	ERRNO	001620	G\$DELM	000372		I\$SRV	000041		
BIT3	000010	G	C\$GPHR	000042	ERRVEC	000004	G\$DISP	000003		I\$SUB	000041		
BIT4	000020	G	C\$GPHO	000030	ERTABE	003370	G\$EXCP	000400		I\$TST	000041		
BIT5	000040	G	C\$GPRI	000040	ERTABL	003170	G\$HILI	000002		J\$JMP	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\$OF SI	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\$PRADA	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\$RADD	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	HOE	100000	G	KIPDR7	172316		
CKRAM2	011304	G	C\$RPT	000025	FERCM	012002	HPM1	105566		KTENAB	003126	G	
CMDPKT	021340	G	C\$SEFG	000046	FIFEXP	012250	HPM2	105622		KTFLG	003124	G	
CHPMEM	017740		C\$SPRI	000041	FIF1MS	012322	HPM3	105646		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	IFALT	004252		LISTAL	000001	
CTABE	003170	G	DEBUGM	011712		FREE	003116	INCERK	017124		LOE	040000	G
CTABM	003156	G	DEVcnt	002212	G	FREEM	003122	INTCPC	016230		LOOPCN	002210	G

LOOPCO	013206	L10001	002170	L10073	105542	O\$ERRT	= 000000	PST32W	003144 G
LOOPFL	003154 G	L10002	005762	L10074	102202	O\$GNSW	= 000001	PUNIT	022364
LOT	= 000010 G	L10003	012124	L10075	105566	O\$POIN	= 000001	PW.D11	= 000021
L\$ACP	002110 G	L10004	012142	L10076	105706	O\$SETU	= 000000	PW.D13	= 000022
L\$APT	002036 G	L10005	012160	MEMADD	014034 G	PASRPT	022134	PW.D22	= 000020
L\$AU	022432 G	L10006	012166	MEMCK	021356 G	PATCH	106046 G	PW.NOP	= 000000
L\$AUT	002070 G	L10007	012204	MENASC	020527	PATDAT	020310	PW.NO1	= 000023
L\$AUTO	022636 G	L10010	012222	MENERR	020454	PC.ERA	= 002400	PW.RDE	= 000024
L\$CCP	002106 G	L10011	012246	MENRES	020556	PC.IER	= 002000	PW.RDR	= 000001
L\$CLEA	022716 G	L10012	012320	MMVEC	= 000250	PC.NOO	= 001000	PW.RDS	= 000005
L\$CO	002032 G	L10013	012470	MSA.FR	= 000006	PC.REL	= 000000	PW.RFI	= 000003
L\$DEPO	002011 G	L10014	013204	MSA.NO	= 000000	PC.REW	= 000400	PW.WCT	= 000006
L\$DESC	003402 G	L10015	014032	MSA.NR	= 000004	PKBCNT	= 000006	PW.WFI	= 000004
L\$DESP	002076 G	L10016	014054	MSA.VO	= 000002	PKHI	= 000004	PW.WFM	= 000007
L\$DEVP	002060 G	L10017	015560	MSGEXP	012224 G	PKLOW	= 000002	PW.WMI	= 000010
L\$DISP	002124 G	L10020	015566	MSGLOO	013144 G	PKTADD	007632	PW.WNP	= 000011
L\$DLY	002116 G	L10021	015574	MSGSTA	012430 G	PKTFRM	007574	PW.WTR	= 000002
L\$DTP	002040 G	L10022	015606	MSGSUB	014022 G	PKTGET	012144 G	P.ACK	= 100000
L\$DTYP	002034 G	L10023	015630	MS.ATI	= 000006	PKTMES	012170 G	P.CMD	= 000037
L\$DU	022530 G	L10024	015656	MS.EXT	= 000200	PKTRAM	007741 G	P.CONT	= 000012
L\$DUT	002072 G	L10025	016016	MS.RSD	= 000001	PKTSSR	012126 G	P.CVC	= 040000
L\$DVTY	003374 G	L10026	016326	MS.RSF	= 000020	PNT	= 001000 G	P.FMT	= 000140
L\$EF	002052 G	L10030	022362	MS.RST	= 000010	PRAMPK	014056	P.FORM	= 000011
L\$ENVI	002044 G	L10031	022526	M8186	005550	PRASC	014603	P.GETS	= 000017
L\$ETP	002102 G	L10032	022634	M8189	005641	PRBEXP	015550	P.IE	= 000200
L\$EXP1	002046 G	L10033	022714	NBA	= 002000	PRBMSG	015416	P.INIT	= 000013
L\$EXP4	002064 G	L10034	022742	NEWPAS	022070	PRBREC	015552	P.MODE	= 007400
L\$EXP5	002066 G	L10035	023204	NODEV	003106 G	PRBTOT	015503	P.OPP	= 020000
L\$HARD	105546 G	L10036	032702	NOINIT	004331	PRBYTE	015202 G	P.POSI	= 000010
L\$HIME	002120 G	L10037	024170	NOINTR	004215	PRI	= 002000 G	P.READ	= 000001
L\$HPCP	002016 G	L10040	024712	NOITS	002162 G	PRIADD	010236	P.SWB	= 010000
L\$HPTP	002022 G	L10041	025436	NOMAN	020614	PRIAO	010306	P.WRIT	= 000005
L\$HW	002150 G	L10042	026260	NOMEM	005454	PRIBX0	007670 G	P.WRTC	= 000004
L\$ICP	002108 G	L10043	041430	NP.IR	= 000200	PRIEQU	010136	P.WRTS	= 000006
L\$INIT	021636 G	L10044	033734	NP.LOO	= 000040	PRIPKT	007446 G	QVP	002176 G
L\$LADP	002026 G	L10045	035360	NP.OUT	= 000100	PRIRAM	010144	RAMASC	014236
L\$LAST	106404 G	L10046	035754	NP.WRP	= 000020	PRITAD	010352	RAMDAT	002234 G
L\$LOAD	002100 G	L10047	036440	NSI	004146	PRITSS	006020	RAMERR	015570 G
L\$LUN	002074 G	L10050	046766	NSINIT	004403	PRITO	010434	RAMEXP	015610 G
L\$MREV	002050 G	L10051	042322	NUL	004523	PRITI	010477	RAMFOR	010174
L\$NAME	002000 G	L10052	043134	NULCR	004524	PRIXOR	010020 G	RAMSIZ	002274 G
L\$PRIO	002042 G	L10053	053044	NXM	= 004000	PRI00	= 000000 G	RAMTAD	015576 G
L\$PROT	021626 G	L10054	047642	NXMFLG	003130 G	PRI01	= 000040 G	RCVHIA	002276 G
L\$PRT	002112 G	L10055	050452	NXMHI	003134 G	PRI02	= 000100 G	RCVLOA	002300 G
L\$REPP	002062 G	L10056	051266	NXMLO	003132 G	PRI03	= 000140 G	RDERR	005202
L\$REV	002010 G	L10057	056040	NXMTST	021532	PRI04	= 000200 G	RECMSG	002460 G
L\$RPT	022744 G	L10060	054506	NXR	003734	PRI05	= 000240 G	RECV	002226 G
L\$SOFT	105700 G	L10061	063412	NXRERR	005732 G	PRI06	= 000300 G	REGSAV	020220
L\$SPC	002056 G	L10062	060476	NXRX	003773	PRI07	= 000340 G	RETERR	005366
L\$SPCP	002020 G	L10063	073342	NXTU	022102	PRMESS	014322	REWIND	011074 G
L\$SPTP	002024 G	L10064	064504	OFL	= 000100	PRMNO	002312 G	RMCHBE	= 000167
L\$STA	002030 G	L10065	065564	ONEFIL	= 000000	PRMSG	014632 G	RMCHEN	= 000200
L\$SW	002160 G	L10066	066426	O\$APTS	= 000000	PRMSG0	015012	RMMSGB	= 000215
L\$TEST	002114 G	L10067	067330	O\$AU	= 000001	PRMSG1	015057	RMMSG	= 000234
L\$TIML	002014 G	L10070	101116	O\$BGNR	= 000001	PRMSG2	015115	RMPKT	= 000201
L\$UNIT	002012 G	L10071	074436	O\$BGNS	= 000001	PROASC	014500	RMPKTE	= 000210
L10000	002156	L10072	075520	O\$DU	= 000001	PRIASC	014545	RMR	= 010000

RMPACK	011170	S2.INR=	000020	T\$EXCP=	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	T3OLOO	032360
SIFLAG	003146 G	TNAM	016764	T\$TAGL=	177777	T29PBP	031371	T3OLOQ	037630
SIMSG	012046	TRANST	002160 G	T\$TAGN=	010077	T29PK2	026430	T3ONEF	040576
SKIPT	003372	TSBA	= 000000 G	T\$TEMP=	000000	T29PK3	026440	T3OOFL	040307
SOFINI	016054 G	TSBAH =	000001 G	T\$TEST=	000011	T29RB	026442	T3OPAC	036500
SPACE	010544 G	TSDB =	000000 G	T\$TSTM=	177777	T29RDF	026720	T3OPK2	036610
SPM1	105706	TSDBH =	000001 G	T\$TSTS=	000001	T29RDG	031643	T3OPK3	036620
SPM4	105736	TSFCOD	007314	T\$\$AU =	010031	T29RES	032146	T3OPTB	037242
SPM6	105766	TSREJ =	000006	T\$\$AUT=	010033	T29RIB	031724	T3ORB	036622
SPM7	106016	TSSDEF	006664	T\$\$CLE=	010034	T29RN	026456	T3ORDF	037413
SRO	= 177572	TSSR =	000002 G	T\$\$DU =	010032	T29RNC	030354	T3ORDG	037471
SR1	= 177574	TSSRBI	003476 G	T\$\$HAR=	010075	T29RRF	026767	T3ORES	041252
SR2	= 177576	TSSRFO	006473	T\$\$HW =	010000	T29RRG	027103	T3ORIB	036745
SR3	= 172516	TSSRH =	000003 G	T\$\$INI=	010030	T29RRN	032024	T3ORN	036636
SSR	= 000200	TSSX	004014	T\$\$MSG=	010025	T29RSZ	026476	T3ORRM	040655
STATCO	012472	TSTBLK	002744 G	T\$\$PRD=	010027	T29RT2	032240	T3ORRN	040733
SVCGBL=	000000	TSTCNT	002206 G	T\$\$RPT=	010035	T29RT3	032302	T3ORRP	041012
SVCINS=	000000	TSTEND	017000	T\$\$SOF=	010076	T29RWN	030305	T3ORT2	041344
SVCSUB=	000001	TSTFLA	002306 G	T\$\$SRV=	010026	T29SC	027217	T3ORT3	041406
SVCTAG=	000000	TSTL00	016536 G	T\$\$SUB=	010074	T29SSR	027507	T3ORWN	040240
SVCTST=	000001	TSTPTR	002310 G	T\$\$SW =	010001	T29SZ	026446	T3OSKM	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	100717	T2.2	033752	T29WDE	027562	T30VCK	040441
S1.ICE=	002000	TST37I	105343	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	105544 G	T29AM3	030427	T29WRT	027731	T30WDF	037553
S1.I1R=	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	T3BFLG	003142 G	T31BFR	043212
S2.ATI=	000010	TTIVEC=	000060 G	T29BOT	027776	T3.1	041462	T31BF2	043320
S2.BTI=	000004	T\$ARGC=	000003	T29BS0	026450	T3.2	042340	T31BOT	044345
S2.DIM=	000200	T\$CODE=	001130	T29BS1	026451	T30BFR	036522	T31BS0	043320
S2.ILW=	000100	T\$ERRN=	001620	T29CNT	026474	T30BF2	036630	T31BS1	043321

T31CNT	043336	T32CNU	051512	T34BA	063046	T35CON	067532	T36BS1	075711
T31CNU	043340	T32DAT	051340	T34BFR	060562	T35DAT	067400	T36CNT	075726
T31CON	043332	T32DLY	051514	T34BF2	060676	T35DLY	067542	T36CNU	075730
T31DAT	043200	T32DSW	051350	T34BOT	061234	T35DSW	067410	T36CON	075722
T31DLY	043342	T32ECF	052455	T34BS0	060676	T35DTA	072325	T36DAT	075570
T31DSW	043210	T32EOT	051611	T34BS1	060677	T35EOT	070510	T36DLY	075732
T31DTA	046446	T32ERA	052016	T34CNT	060672	T35INT	072601	T36DSW	075600
T31EOT	044540	T32L00	047020	T34CON	060710	T35LON	071470	T36DTA	100622
T31LON	045520	T320PI	052603	T34DAT	060550	T35L00	063444	T36EOT	077005
T31L00	041462	T32PAC	051330	T34DLY	060674	T35L0P	071552	T36LON	077765
T31L0P	045602	T32PK2	051440	T34DSW	060560	T35L0Q	070205	T36L00	073400
T31L0Q	044116	T32PK3	051450	T34EOT	062205	T35LOR	070060	T36L0P	100047
T31LOR	043771	T32RB	051452	T34ET	062116	T35MOT	072503	T36L0Q	076446
T31NEF	046040	T32RES	052700	T34ETC	061157	T35NEF	072010	T36LOR	076321
T31OFL	045065	T32RIB	052136	T34ETN	061451	T35NIN	073056	T36NAS	075734
T31PAC	043170	T32RT2	052772	T34ETO	061002	T35OFL	071035	T36NEF	100305
T31PBP	045664	T32RT3	053022	T34ETS	061530	T35OPM	072672	T36OFL	077332
T31PK2	043300	T32RWN	051700	T34ETZ	061622	T35PAC	067370	T36PAC	075560
T31PK3	043310	T32SCF	052234	T34ET2	061367	T35PBP	071634	T36PBP	100131
T31RB	043312	T32SZ	051456	T34L00	056072	T35PK2	067500	T36PK2	075670
T31RDE	043344	T32TSA	052311	T34OFL	062527	T35PK3	067510	T36PK3	075700
T31RDF	043543	T32WB	051452	T34PAC	060540	T35RB	067512	T36RB	075702
T31RES	046610	T32WDC	052536	T34PK2	060650	T35RDF	067632	T36RDF	076073
T31RN	043326	T33BFR	054572	T34PK3	060660	T35RES	073164	T36RES	100740
T31RNC	044743	T33BF2	054700	T34POS	060714	T35RN	067526	T36RN	075716
T31RRF	043612	T33BOT	055325	T34RB	060662	T35RNC	070713	T36RNC	077210
T31RT2	046702	T33BS0	054700	T34RES	063232	T35RRF	067701	T36RRF	076142
T31RT3	046744	T33BS1	054701	T34RNC	062406	T35RT2	073256	T36RT2	101032
T31RWN	044674	T33CNT	054716	T34RRE	061066	T35RT3	073320	T36RT3	101074
T31SC	043707	T33CNU	054720	T34RSZ	060670	T35RWE	072770	T36RWN	077141
T31SCF	046161	T33CON	054712	T34RT2	063324	T35RWN	070644	T36SC	076237
T31SSR	044177	T33DAT	054560	T34RT3	063366	T35SC	067776	T36SCF	100403
T31SZ	043316	T33DLY	054722	T34RWN	062337	T35SCF	072106	T36SSR	076527
T31S2	043322	T33DSW	054570	T34SSR	062063	T35SSR	072422	T36SZ	075706
T31S3	043324	T33DTA	055550	T34STM	061700	T35SZ	067516	T36S2	075712
T31TIM	044440	T33L00	053076	T34SZ	060666	T35S2	067522	T36S3	075714
T31TM	044617	T33PAC	054550	T34S2	060700	T35S3	067524	T36TIM	076730
T31TRL	045752	T33PK2	054660	T34S3	060702	T35TIM	070433	T36TM	077064
T31TSA	046236	T33PK3	054670	T34TM	062263	T35TRL	070567	T36TRL	100217
T31VCK	045303	T33RB	054672	T34TMK	061763	T35TSA	071722	T36TSA	100460
T31WB	043312	T33RBP	054724	T34VCK	062773	T35VCK	071253	T36VCK	077550
T31WDC	045230	T33RES	055662	T34WB	060662	T35VCK	071253	T36V	075702
T31WDD	045140	T33RN	054706	T34WD	060704	T35WB	067512	T36WDC	077475
T31WDE	044233	T33RT2	054754	T34WDC	062671	T35WDC	071200	T36WDD	077405
T31WDF	044041	T33RT3	056016	T34WDD	062602	T35WDD	071110	T36WDE	076563
T31WDR	043330	T33RWN	055420	T34WDR	060706	T35WDE	070266	T36WDF	076371
T31WNG	043471	T33SSR	055241	T34WSS	063120	T35WDF	070130	T36WDR	075720
T31WNH	043410	T33SZ	054676	T34WTH	061300	T35WDR	067530	T36WNG	076005
T31WRF	046343	T33S2	054702	T35AM3	070766	T35WNG	067544	T36WRF	100542
T31WSS	045431	T33S3	054704	T35BA	071326	T35WRF	072245	T36WSS	077676
T32AM3	051747	T33UNC	055062	T35BFR	067412	T35WSS	071401	T37AM3	103707
T32BA	052063	T33UND	055152	T35BF2	067520	T36AM3	077263	T37BA	104247
T32BFR	051352	T33WB	054672	T35BOT	070340	T36BA	077623	T37BFR	102262
T32BOE	052366	T33WDC	055467	T35BS0	067520	T36BFR	075602	T37BF2	102370
T32BOT	051516	T33WDR	054710	T35BS1	067521	T36BF2	075710	T37BOT	103261
T32CMD	051460	T33WPW	055002	T35CNT	067536	T36BOT	076635	T37BS0	102370
T32CNT	051510	T34AM3	062461	T35CNU	067540	T36BS0	075710	T37BS1	102371

T37CNT	102406	T37SSR	103136	T7.4	066444	WSMBK	021350	G	X8OFFS	=	000400			
T37CNU	102410	T37SZ	102366	T8	073344	G	XFERAS	016020	X8TRUE	=	000020			
T37CON	102402	T37S2	102372	T8.1	073400	XNXM	016456	X1.COR	=	020000				
T37DAT	102250	T37S3	102374	T8.2	074454	XORBFO	007752	X1.DLT	=	100000				
T37DLY	102412	T37TIM	103354	T9	101120	G	XORFOR	010070	X1.MBZ	=	017375			
T37DSW	102260	T37TM	103510	T9.1	101154	XST0	=	000006	G	X1.RBP	=	000400		
T37DTA	105246	T37TRL	104643	UAM	=	000200	G	XST1	=	000010	G	X1.SPA	=	040000
T37EOT	103431	T37TSA	105104	UNITM	=	002174	G	XST2	=	000012	G	X1.UNC	=	000002
T37LON	104411	T37VCK	104174	UNREC	=	000006	XST3	=	000014	G	X2.BUF	=	000100	
T37LOO	101154	T37WB	102362	USI	004117	XST4	=	000016	G	X2.EXT	=	000200		
T37LOP	104473	T37WDC	104121	WAITF	016330	G	XSOBOT	=	000002	X2.OPM	=	100000		
T37LOQ	103055	T37WDD	104031	WC.IFA	=	000200	XSOEOT	=	000001	X2.RCE	=	040000		
T37LOR	102730	T37WDE	103172	WC.IFE	=	000002	XSOIE	=	000040	X2.REV	=	000077		
T37NEF	104731	T37WDF	103000	WC.IGO	=	000001	XSOILA	=	000400	X2.SPA	=	035400		
T37OFL	103756	T37WDR	102400	WC.IRE	=	000010	XSOILC	=	001000	X2.UNI	=	000007		
T37PAC	102240	T37WNG	102414	WC.IRW	=	000004	XSOLET	=	020000	X2.WCF	=	002000		
T37PBP	104555	T37WRF	105166	WC.IOT	=	000100	XSOMOT	=	000200	X3.DCK	=	000010		
T37PK2	102350	T37WSS	104322	WC.IIT	=	000040	XSONEF	=	002000	X3.MBZ	=	000006		
T37PK3	102360	T4	046770	G	WC.ISR	=	000020	XSOONL	=	000100	X3.MDE	=	177400	
T37RB	102362	T4.1	047020	WF.IED	=	000010	XSOPED	=	000010	X3.OPI	=	000100		
T37RDF	102502	T4.2	047660	WF.IER	=	000004	XSORLL	=	010000	X3.REV	=	000040		
T37RES	105364	T4.3	050470	WF.IMI	=	000200	XSORLS	=	040000	X3.RIB	=	000001		
T37RN	102376	T5	053046	G	WF.IRE	=	000040	XSOTMK	=	100000	X3.SPA	=	000200	
T37RNC	103634	T5.1	053076	WF.IWF	=	000020	XSOVCK	=	000020	X3.TRF	=	000020		
T37RRF	102551	T6	056042	G	WF.IWR	=	000100	XSOWLE	=	004000	X4.HSP	=	100000	
T37RT2	105456	T6.1	056072	WF.I3R	=	000002	XSOWLK	=	000004	X4.MBZ	=	017400		
T37RT3	105520	T7	063414	G	WF.I4R	=	000001	XXCOMM	=	003114	G	X4.RCE	=	040000
T37RWN	103565	T7.1	063444	WRCHR	010742	G	X8ALWA	=	000000	X4.TSM	=	020000		
T37SC	102646	T7.2	064522	WRERR	005107	X8FALS	=	000040	X4.WRC	=	000377			
T37SCF	105027	T7.3	065602	WRMSG	005052									

. ABS. 106404 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 30328 WORDS (119 PAGES)
DYNAMIC MEMORY: 20614 WORDS (79 PAGES)
ELAPSED TIME: 00:45:58
CVTSDB,CVTSDB.SEQ/-SP=SVC/ML,TSV10,TSV220,TSV38,TSV4,TSV7B,TSV6