

TSU05

TSU05 DIAG PART 3
CZTSCAO

COPYRIGHT (c) 1983
AH-T721A-MC
FICHE 01 OF 02

JUL 1984
digital
Made In USA

The main body of the document is a large grid of 16 columns and 20 rows of small, illegible text blocks. Each block appears to be a technical specification or a small table, but the text is too faint to read. The layout is consistent across the entire page, with a small white mark at the bottom center.

.REM_
IDENTIFICATION

PRODUCT ID: AC-T720A-MC
PRODUCT TITLE: CZTSCAO TSU05 DIAG PART 3
DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PPG
DATE: JUNE 08, 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) 1983 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

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

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS IS A PDP-11 RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSU05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A PDP-11//23 SYSTEM (UNIBUS). 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 PROCESSOR AND MEMORY
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE I.E. 4K FOR I/O PAGE)
TSU05 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)
CONSOLE TERMINAL
PDP-11 DIAGNOSTIC SUPERVISOR (HSAAA.SYS VERSION 34 OR LATER)
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. CHQUS XXDP+ USERS MANUAL; DOCUMENT NUMBER AC-F348E-MC
DATE: 14 JULY 1980.
2. TSU05 TRANSPORT SUBSYSTEM USER'S GUIDE; DOCUMENT NUMBER EK-TSU05-UG-001
DATE: AUGUST 1982
3. TSU05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSU05-TM-001
DATE: AUGUST 1982
4. TSU05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSU05-IN-001
DATE: AUGUST 1982

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

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

1.5 ASSUMPTIONS

ALL HARDWARE EXCEPT THE HARDWARE UNDER TEST IS ASSUMED TO WORK PROPERLY OR FALSE ERRORS CAN BE REPORTED.
THE TAPE BEING USED ON THE TS05 TRANSPORT IS A KNOWN GOOD REEL OF TAPE.
CVTSAA AND CVTSBA 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 TSU05 DIAGNOSTIC IS A PDP-11 DIAGNOSTIC SUPERVISOR COMPATIBLE PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE CHQUS XXDP+ USERS MANUAL, DOCUMENT NUMBER AC-F348E-MC. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC MEDIA

```
.R VTSC??
DIAG. RUN-TIME SERVICES REV D. APR 79
CVTSC-A-0
****TSU05 LOGIC DIAGNOSTIC****
UNIT IS TSU05
```

>DR

2.2 SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

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

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

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

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

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

2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?" YOU MUST ANSWER "Y" AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 6 OF THE XXDP* USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A "Y", THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL).

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

ON A "N" (NO) RESPONSE TO THE "CHANGE HW?" QUESTION, THE DIAGNOSTIC WILL

RUN USING THE DEFAULT VALUES FOR ALL QUESTIONS. THE DEFAULT ADDRESS AND VECTOR ARE:

TSBA/TSDB = 172520, VECTOR = 224

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

UNITS (D) ? <ENTER THE NUMBER OF M7455 CONTROLLERS
PRESENT TO BE TESTED>

UNIT 0

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

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

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

UP TO 4 TSU05 CONTROLLERS PER PDP-11 AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

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

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

UNITS (D) ? 8<CR>

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

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

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

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

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

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

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

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

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

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

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

```
# UNITS (D) ? 8<CR>

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

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

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

AS YOU CAN SEE IN THE ABOVE DIALOGUE, THE RUNTIME SERVICES WILL BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. IN THE FIRST PASS, TWO ENTRIES ARE BUILT SINCE TWO SUB-DEVICES AND Q-FACTORS WERE SPECIFIED. THE SERVICES ASSUME THAT THE CSR ADDRESS IS 160000 FOR BOTH SINCE IT WAS SPECIFIED ONLY ONCE. IN THE SECOND PASS, FOUR ENTRIES WERE BUILT. THIS IS BECAUSE FOUR SUB-DEVICES WERE SPECIFIED. THE "-" CONSTRUCT TELLS THE RUNTIME SERVICES TO INCREMENT THE DATA FROM THE FIRST NUMBER TO THE SECOND. IN THIS CASE, SUB-DEVICES 2, 3, 4 AND 5 WERE SPECIFIED. (IF THE SUB-DEVICE WERE SPECIFIED BY ADDRESSES, THE INCREMENT WOULD BE BY 2 SINCE ADDRESSES MUST BE ON AN EVEN BOUNDARY.) THE CSR ADDRESSES AND Q-FACTORS FOR THE FOUR ENTRIES ARE ASSUMED TO BE 160000 AND 0 RESPECTIVELY SINCE THEY WERE ONLY SPECIFIED ONCE. THE LAST TWO UNITS ARE SPECIFIED IN THE THIRD PASS.

THE WHOLE PROCESS COULD HAVE BEEN ACCOMPLISHED IN ONE PASS AS SHOWN BELOW.

```
# UNITS (D) ? 8<CR>

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

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING

A NULL FIELD) TELL THE RUNTIME SERVICES TO REPEAT THE LAST REPLY.

2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK) QUESTIONS
3. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM
4. TYPE "START"
5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. THESE DEFAULTS ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY A DIAGNOSTIC: GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES ARE ALWAYS PRINTED UNLESS THE "IER" FLAG IS SET (SECTION 2.3). THE GENERAL ERROR MESSAGE IS OF THE FORM:

```
NAME TYPE NUMBER ON UNIT NUMBER TST NUMBER PC:XXXXXX  
ERROR MESSAGE
```

.WHERE: NAME = DIAGNOSTIC NAME
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
NUMBER = ERROR NUMBER
UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS THE "IER" OR "IBR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBR" OR "IXR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 SPECIFIC ERROR MESSAGES

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

ERROR MESSAGE EXAMPLE 1

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

TST: 016 FIFO EXERCISER TEST
CVTSC HRD ERR 01610 ON UNIT 00 TST 016 SUB 002 PC: 040624
FIFO STATUS (IN WORD 9) INCORRECT AFTER WRITE FIFO

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

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

MESSAGE BUFFER ADDRESS = 047352

MESSAGE BUFFER CONTENTS:

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

ERROR MESSAGE EXAMPLE 2

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

CVTSC HRD ERR 00159 ON UNIT 00 TST 001 SUB 005 PC: 026202
TSSR NOT CORRECT AFTER SPACE RECORDS COMMAND

TSSR = 100214

TSSR BITS SET: SC,SSR

TERMINATION CLASS CODE = UNRECOVERABLE ERROR

PACKET ADDRESS = 026420

PACKET WORD # = 140010

PACKET WORD # = 000010

PACKET WORD # = 000000

PACKET WORD # = 000024

ERROR MESSAGE EXAMPLE 3

THIS ERROR SHOWS THAT THE MOTION BIT DID NOT GET SET WHILE DOING A REWIND WITH EXTENDED FEATURES MODE ENABLED.

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

4.0 PERFORMANCE AND PROGRESS REPORTS

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

SUCCESSFUL RUN EXAMPLE (PDP-11)

```
DR>STA/FLA:PNT:HOE
UNITS (D) ? 1
UNIT 0
DEVICE ADDRESS (0) 172520 ? <CR>
VECTOR (0) 224 ? <CR>
CHANGE SW (L) ? N<CR>
```

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

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

0 ERRORS

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

PROGRAM RUN TIMES

THE AVERAGE RUN TIMES OF THE PROGRAM ARE LISTED BELOW. THESE FIGURES ARE TO BE USED AS A GUIDE. THE TIMING WAS DONE ON A PDP-11 PROCESSOR WITH A LA34 CONSOLE.

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

TEST NUMBER	N/I SECS.	NUMBER ITER	DEF SECS.
1	3	10	7
2	3	8	5
3	38	250	212
4	60	300	240
5	60	300	240
6	120	360	240
7	120	600	480
8	22	90	68

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

Q.V. 7 MINUTES
 DEFAULT 31 MINUTES

5.0 DEVICE INFORMATION TABLES

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

CHANGE HW (L) ?

UNITS (D) ? <ENTER THE NUMBER OF M7455 CONTROLLERS
 PRESENT TO BE TESTED>

UNIT 0

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

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

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

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

BUFFER ADDRESSES, ILLEGAL CODES IN THE MODE FIELD OF THE BASIC READ COMMAND, AND DATA BUFFERS IN NON-EXISTANT MEMORY.

CAUTION

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

TEST 5: SPACE RECORDS

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

TEST 6: REREADS

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

CAUTION

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

TEST 7: WRITE DATA RETRY

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

TEST 8: WRITE/READ TAPE MARK

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

D2

USER DOCUMENTATION

MACRO M1113 01-FEB-84 17:54

SEQ 016

FORMAT COMMANDS WITH ILLEGAL MODE CODES ARE REJECTED.

7.0 MAINTENANCE HISTORY

REVISION A - JUNE 1983

```

1          .TITLE  TSV2 - PROGRAM HEADER
2          .SBTTL  PROGRAM HEADER
3 000000   .PSECT  ABS
4
10         .MCALL  SVC
11 000000   SVC          ; INITIALIZE SUPERVISOR MACROS
12         .ENABLE LC
13         .NLIST  BEX,CND
19         .ENABL  AMA
20         :::@@@ . =2000
21         002000' . =.+2000
22 002000   BGNMOD  TSV2
23         TSV2::
24
25         ;**
26         ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
27         ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
28         ;--
29
30         POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
31 002000   HEADER  CZTSC,A,0,655.,0
002000   L$NAME::      ;DIAGNOSTIC NAME
002000   .ASCII  /C/
002001   .ASCII  /Z/
002002   .ASCII  /T/
002003   .ASCII  /S/
002004   .ASCII  /C/
002005   .BYTE   0
002006   .BYTE   0
002007   .BYTE   0
002010   L$REV::      ;REVISION LEVEL
002010   .ASCII  /A/
002011   L$DEPO::     ;0
002011   .ASCII  /0/
002012   L$UNIT::     ;NUMBER OF UNITS
002012   .WORD   0
002014   L$TIML::     ;LONGEST TEST TIME
002014   .WORD   655.
002016   L$HPCP::     ;POINTER TO H.W. QUES.
002016   .WORD   L$HARD
002020   L$SPCP::     ;POINTER TO S.W. QUES.
002020   .WORD   L$SOFT
002022   L$HPTP::     ;PTR. TO DEF. H.W. PTABLE
002022   .WORD   L$HW
002024   L$SPTP::     ;PTR. TO S.W. PTABLE
002024   .WORD   L$SW
002026   L$LADP::     ;DIAG. END ADDRESS
002026   .WORD   L$LAST
002030   L$STA::      ;RESERVED FOR APT STATS
002030   .WORD   0
002032   L$CO::       ;
002032   .WORD   0
002034   L$DTYP::     ;DIAGNOSTIC TYPE
002034   .WORD   0
002036   L$APT::      ;APT EXPANSION
002036   .WORD   0

```

002040
002040 002124'
002042
002042 000000
002044
002044 000000
002046
002046 000000
002050
002050 003
002051 003
002052
002052 000000
002054 000000
002056
002056 000000
002060
002060 003372'
002062
002062 022524'
002064
002064 000000
002066
002066 000000
002070
002070 022212'
002072
002072 022310'
002074
002074 000000
002076
002076 003400'
002100
002100 104035
002102
002102 000000
002104
002104 021416'
002106
002106 022476'
002110
002110 022416'
002112
002112 021406'
002114
002114 000000
002116
002116 000000
002120
002120 000000

L\$DTP:: .WORD L\$DISPATCH ;PTR. TO DISPATCH TABLE
L\$PRIO:: .WORD 0 ;DIAGNOSTIC RUN PRIORITY
L\$ENVI:: .WORD 0 ;FLAGS DESCRIBE HOW IT WAS SETUP
L\$EXP1:: .WORD 0 ;EXPANSION WORD
L\$MREV:: .WORD 0 ;SVC REV AND EDIT #
C\$REVISION
C\$EDIT
L\$EF:: .WORD 0 ;DIAG. EVENT FLAGS
L\$SPC:: .WORD 0
L\$DEVP:: .WORD 0 ; POINTER TO DEVICE TYPE LIST
L\$DV TYP
L\$REPP:: .WORD L\$RPT ;PTR. TO REPORT CODE
L\$EXP4:: .WORD 0
L\$EXP5:: .WORD 0
L\$AUT:: .WORD L\$AU ;PTR. TO ADD UNIT CODE
L\$DUT:: .WORD L\$DU ;PTR. TO DROP UNIT CODE
L\$LUN:: .WORD 0 ;LUN FOR EXERCISERS TO FILL
L\$DESP:: .WORD L\$DESC ;POINTER TO DIAG. DESCRIPTION
L\$LOAD:: .WORD L\$LOAD ;GENERATE SPECIAL AUTOLOAD EMT
EMT
L\$ETP:: .WORD 0 ;POINTER TO ERR TBL
L\$ICP:: .WORD L\$INIT ;PTR. TO INIT CODE
L\$CCP:: .WORD L\$CLEAN ;PTR. TO CLEAN-UP CODE
L\$ACP:: .WORD L\$AUTO ;PTR. TO AUTO CODE
L\$PRT:: .WORD L\$PROT ;PTR. TO PROTECT TABLE
L\$TEST:: .WORD 0 ;TEST NUMBER
L\$DLY:: .WORD 0 ;DELAY COUNT
L\$HIME:: .WORD 0 ;PTR. TO HIGH MEM

.SBTTL DISPATCH TABLE

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

32
33
34
35
36
37
38

```

39
40 002122          DISPATCH 8
    002122 000010      .WORD 8
    002124          L$DISPATCH::
    002124 023306'      .WORD T1
    002126 024422'      .WORD T2
    002130 027102'      .WORD T3
    002132 034252'      .WORD T4
    002134 046506'      .WORD T5
    002136 055434'      .WORD T6
    002140 075006'      .WORD T7
    002142 105044'      .WORD T8

41
42
43          .SBTTL  DEFAULT HARDWARE P-TABLE
44
45          ;**
46          ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
47          ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
48          ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
49          ;--
50 002144          BGNHW   DFPTBL      ;DEFAULT HARD-P-TABLE
    002144 000003      .WORD   L10000-L$HW/2
    002146          L$HW::
    002146          DFPTBL::
51
52 002146 172520      .WORD   172520      ; 1ST (OF 2) REGISTERS.
53 002150 000224      .WORD   224        ; INTERRUPT VECTOR
54 002152 000200      .WORD   PRI04       ; INTERRUPT PRIORITY.
55 002154          ENDHW
    002154          L10000:

56
57          .SBTTL  SOFTWARE P-TABLE
58
59          ;**
60          ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
61          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
62          ;--
63 002154          BGNSW   SFPTBL
    002154 000004      .WORD   L10001-L$SW/2
    002156          L$SW::
    002156          SFPTBL::
64
65 002156 000000      TRANSTST:: .WORD 0      ; ENABLE TEST OF TRANSPORT(S) IF =1
66 002160 000000      NOITS::   .WORD 0      ; INHIBIT ITERATION OPTION.
67
68
69 002162 000017      LERRMAX:: .WORD 15.   ; ... 0 = ITERATE.
70 002164 000310      GERRMAX:: .WORD 200.  ; ...NZ = INHIBIT ITERATE.
71 002166          ENDSW      ; LOCAL (PER TEST) ERROR LIMIT
    002166          L10001:   ; GLOBAL (PER UNIT) ERROR LIMIT
72
73 002166          ENDMOD
74

```

7
8
13
19
20 002166
002166
21
22
23
24
25
26
27
28
29
33 002166

.TITLE TSV3 - GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

BGNMOD TSV3
TSV3::

.SBTTL GLOBAL EQUATES SECTION

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

EQUALS ; GET STANDARD EQUATES.

; BIT DIFINITIONS

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

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

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

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

;
;

```

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

```

```

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

```

34
35 002166

```

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

```

```

000250
177572
177574
177576
172516

```

```
.ENDC
;*USER "I" PAGE ADDRESS REGISTERS
UIPAR0= 177640
UIPAR1= 177642
UIPAR2= 177644
UIPAR3= 177646
UIPAR4= 177650
UIPAR5= 177652
UIPAR6= 177654
UIPAR7= 177656
. IF NB
;*USER "D" PAGE ADDRESS REGISTERS
UDPAR0= 177660
UDPAR1= 177662
UDPAR2= 177664
UDPAR3= 177666
UDPAR4= 177670
UDPAR5= 177672
UDPAR6= 177674
UDPAR7= 177676
.ENDC
. 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
40
41
42
43
44
45
46
47

```

      .SBTTL TSU05 REGISTER AND PACKET DEFINITIONS
;
; SOME GENERAL EQUATES.
;
000004 ERRVEC== 4 ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.

```

```

48      000060      TTIVEC==      60      ; INTERRUPT VECTOR FOR CONSOLE INPUT
49      177560      TTICSR==      177560    ; BUS ADDRESS OF CONSOLE INPUT
50      177562      TTIBFR==      177562    ; CONSOLE INPUT DATA BUFFER
51      177520      BDVPCR==      177520    ; BDV11 PAGE CONTROL REGISTER
52
53      ;+
54      ;BIT DEFINITIONS FOR TSSR REGISTER
55      ;-
56
57      100000      SC=      BIT15      ;SPECIAL CONDITION
58      040000      BIE=      BIT14      ;BUS INTERFACE ERROR
59      020000      SCE=      BIT13      ;SANITY CHECK ERROR
60      010000      RMR=      BIT12      ;MODIFICATION REFUSED
61      004000      NXM=      BIT11      ;NONEXISTANT MEMORY ERROR
62      002000      NBA=      BIT10      ;NEED BUFFER ADDRESS
63      001400      HIADDR= BIT9!BIT8    ;EXTENDED ADDRESS BITS
64      000200      SSR=      BIT7      ;SUB SYSTEM READY
65      000100      OFL=      BIT6      ;OFF LINE BIT
66      000060      FATERR= BIT4!BIT5    ;FATAL TERMINATION ERROR CODES
67      000016      TERCLS= BIT3!BIT2!BIT1 ;TERMINATION CODES
68
69
70      ;+
71      ;
72      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
73      ;(XST0)
74      ;
75      ;-
76
77      100000      XSOTMK= BIT15      ;TAPE MARK DETECTED
78      040000      XSORLS= BIT14      ;RECORD LENGTH SHORT
79      020000      XSOLET= BIT13      ;LOGICAL END OF TAPE
80      010000      XSORLL= BIT12      ;RECORD LENGTH LONG
81      004000      XSOWLE= BIT11      ;WRITE LOCK ERROR
82      002000      XSONEF= BIT10      ;NON EXECUTABLE FUNCTION
83      001000      XSOILC= BIT9      ;ILLEGAL COMMAND
84      000400      XSOILA= BIT8      ;ILLEGAL ADDRESS
85      000200      XSOMOT= BIT7      ;TAPE IN MOTION
86      000100      XSOONL= BIT6      ;TRANSPORT ON LINE
87      000040      XSOIE=  BIT5      ;INTERRUPT ENABLE
88      000020      XSOVCK= BIT4      ;VOLUME CHECK BIT
89      000010      XSOPED= BIT3      ;PHASE ENCODED DRIVE
90      000004      XSOWLK= BIT2      ;WRITE LOCKED
91      000002      XSOBOT= BIT1      ;BEGINNING OF TAPE
92      000001      XSOEOT= BIT0      ;END OF TAPE
93
94
95      ;+
96      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
97      ;(XST1)
98      ;-
99      100000      X1.DLT = BIT15      ;DATA LATE
100     040000      X1.SPARE= BIT14      ;NOT USED
101     020000      X1.COR = BIT13      ;CORRECTABLE DATA ERROR
102     017375      X1.MBZ = BIT12*BIT11*BIT10*BIT9*BIT7*BIT6*BIT5*BIT4*BIT3*BIT2*BIT0 ;ALWAYS 0
103     000400      X1.RBP = BIT8      ;READ BUS PARITY ERROR
104     000002      X1.UNC = BIT1      ;UNCORRECTABLE DATA OR HARD ERROR

```

```

105
106
107      ;+
108      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
109      ;(XST2)
110      ;-
110      100000      X2.OPM = BIT15          ;OPERATION IN PROGRESS (TAPE MOVING)
111      040000      X2.RCE = BIT14          ;RAM CHECKSUM ERROR
112      035400      X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8 ;NOT USED BY TSU05 (ALWAYS=0)
113      002000      X2.WCF = BIT10         ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
114      000200      X2.EXTF = BIT7         ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
115      000100      X2.BUFE = BIT6         ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
116      000077      X2.REV = 000077        ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
117      000007      X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
118
119      ;+
120      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
121      ;(XST3)
122      ;-
123      177400      X3.MDE = 177400         ;MICRO-DIAGNOSTIC ERROR CODE
124      000200      X3.SPARE= BIT7         ;NOT USED BY TSU05
125      000100      X3.OPI = BIT6         ;OPERATION INCOMPLETE
126      000040      X3.REV = BIT5         ;REVERSE
127      000020      X3.TRF = BIT4         ;TRANSPORT RESPONSE FAILURE
128      000010      X3.DCK = BIT3         ;DENSITY CHECK
129      000006      X3.MBZ =BIT2+BIT1     ;NOT USED ALWAYS 0
130      000001      X3.RIB = BIT0         ;REVERSE INTO BOT
131
132      ;+
133      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
134      ;(XST4)
135      ;-
136      100000      X4.HSP = BIT15         ;HIGH SPEED
137      040000      X4.RCE = BIT14         ;RETRY COUNT EXCEEDED
138      020000      X4.TSM = BIT13         ;TRANSPORT SPECIAL MODE
139      017400      X4.MBZ = BIT12+BIT11+BIT10+BIT9+BIT8 ;NOT USED ALWAYS 0
140      000377      X4.WRC = 000377        ;WRITE RETRY COUNT FIELD
141
142
143      ;+
144      ;
145      ;TSSR TERMINATION CODES (BIT 0-2)
146      ;
147      ;-
148
149      000006      TSREJ= 3+2             ;COMMAND REJECTED
150      000006      UNREC= 6              ;UNRECOVERABLE ERROR
151
152      ;+
153      ;
154      ;DEVICE REGISTER OFFSETS
155      ;
156      ;-
157
158      000000      TSBA== 0
159      000000      TSDB== 0              ;TSDB/TSBA REGISTER
160      000001      TSBAH== 1
161      000001      TSDBH== 1            ;TSDB/TSBA REGISTER HIGH BYTE

```

```

162      000002      TSSR== 2      ;TSSR REGISTER
163      000003      TSSRH== 3      ;TSSR REGISTER HIGH BYTE
164
165      ;+
166      ; TSDB ADDRESS BIT DEFINITIONS
167      ;-
168      000003      A1716 = BIT1+BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
169
170      ;+
171      ; COMMAND DEFINITIONS
172      ;-
173      000017      P.GETSTAT = 17      ;GET STATUS
174      000013      P.INIT = 13      ;INITIALIZE
175      000012      P.CONTROL = 12      ;CONTROL COMMANDS
176      000011      P.FORMAT = 11      ;FORMAT
177      000010      P.POSITION = 10      ;POSITION
178      000006      P.WRTSUB = 6      ;SUBSYSTEM WRITE
179      000005      P.WRITE = 5      ;WRITE
180      000004      P.WRTCHAR = 4      ;WRITE CHARACTERISTICS
181      000001      P.READ = 1      ;READ
182
183      ;+
184      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
185      ;-
186      100000      P.ACK = BIT15      ;BUFFER AVAIL FOR CONTROLLER
187      040000      P.CVC = BIT14      ;CLEAR VOLUME CHECK
188      020000      P.OPP = BIT13      ;REVERSE SEQUENCE OF DATA BITS
189      010000      P.SWB = BIT12      ;SWAP BYTES IN MEMORY
190      007400      P.MODE = BIT11!BIT10!BIT9!BIT8 ;EXTENDED COMMAND MODE FIELD
191      000200      P.IE = BIT7      ;INTERRUPT ENABLE
192      000140      P.FMT= BIT6!BIT5      ;PACKET HEADER TYPE (ALWAYS=0)
193      000037      P.CMD = 37      ;MAJOR COMMAND FIELD
194
195      ;+
196      ; CONTROL COMMAND MODE CODES
197      ;-
198      000000      PC.RELEASE = 0*256.      ;RELEASE BUFFER
199      000400      PC.REWIND = 1*256.      ;REWIND
200      001000      PC.NOOP = 2*256.      ;NO-OP
201      002000      PC.IEREW = 4*256.      ;REWIND IMMEDIATE INTERRUPT
202      002400      PC.ERASE = 5*256.      ;SECURITY ERASE
203
204      ;+
205      ; CONTROLLER RAM DEFINITIONS
206      ;-
207      000167      RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
208      000200      RMCHEND = 200      ;CHARACTERISTICS IO DATA END RAM ADDRESS
209      000201      RMPKTBEG= 201      ;COMMAND PACKET BEGIN RAM ADDRESS
210      000210      RMPKTEND= 210      ;COMMAND PACKET END RAM ADDRESS
211      000215      RMMSGBEG= 215      ;MESSAGE BUFFER BEGIN RAM ADDRESS
212      000234      RMMSGEND= 234      ;MESSAGE BUFFER END RAM ADDRESS
213
214      ;+
215      ; REGISTER DEFINITIONS IN THE MESSAGE BUFFER
216      ;-
217
218      000006      XSTO== 6      ;EXTENDED STATUS REGISTER 0 (WORD 4)

```

```

219      000010      XST1==  8.      ;EXTENDED STATUS REGISTER 1 (WORD 5)
220      000012      XST2== 10.     ;EXTENDED STATUS REGISTER 2 (WORD 6)
221      000014      XST3== 12.     ;EXTENDED STATUS REGISTER 3 (WORD 7)
222      000016      XST4== 14.     ;EXTENDED STATUS REGISTER 4 (WORD 8)
223
224
225      ;*
226      ;
227      ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
228      ;
229      ;-
230
231      000002      PKLOW   = 2      ;LOW ORDER CHARACTERISTIC DATA POINTER
232      000004      PKHI    = 4      ;HIGH ORDER CHARACTERISTIC DATA POINTER
233      000006      PKBCNT  = 6      ;NUMBER OF BYTES IN DATA PACKET
234
235      000010      EXBCNT=10      ;NUMBER OF BYTES IN EXTENDED DATA PACKET
236
237      ;*
238      ;DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
239      ;-
240      000000      BSELO   = 0      ;BYTE 0
241      000001      BSEL1   = 1      ;BYTE 1
242      000002      SEL2    = 2      ;WORD 2
243      000004      SELDATA = 4      ;WORD 3
244
245      ;*
246      ;BSELO SELECT CODES FOR WRITE SUBSYSTEM COMMAND
247      ;-
248      000000      PW.NOP   = 0      ;NO-OP
249      000001      PW.RDRAM = 1      ;READ RAM
250      000002      PW.WTRAM = 2      ;WRITE RAM
251      000003      PW.RFIFO = 3      ;READ FIFO
252      000004      PW.WFIFO = 4      ;WRITE FIFO
253      000005      PW.RDSTAT = 5     ;READ STATUS
254      000006      PW.WCTL  = 6     ;WRITE TAPE CONTROL
255      000007      PW.WFMT  = 7     ;WRITE TAPE FORMAT
256      000010      PW.WMISC = 10     ;WRITE MISCELLANEOUS
257      000011      PW.WNPR  = 11     ;WRITE NPR CONTROL
258      000020      PW.D22   = 20     ;DO MICROTEST 22
259      000021      PW.D11   = 21     ;DO MICROTEST 11
260      000022      PW.D13   = 22     ;DO MICROTEST 13
261      000023      PW.NO1311 = 23    ;DISABLE MICROTEST 11 AND 13
262      000024      PW.RDXT  = 24     ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
263
264      ;*
265      ;BSEL1 CODES FOR WRITE TAPE CONTROL
266      ;-
267      000200      WC.IFAD   = BIT7   ;IFAD - FORMATTER ADDRESS
268      000100      WC.IOTAD  = BIT6   ;ITADO - TRANSPORT ADDRESS BIT 0
269      000040      WC.I1TAD  = BIT5   ;ITAD1 - TRANSPORT ADDRESS BIT 1
270      000020      WC.ISRESV = BIT4   ;IRESV5 - RESERVED #5
271      000010      WC.IREW   = BIT3   ;IREW - REWIND
272      000004      WC.IRWU   = BIT2   ;IRWU - REWIND AND UNLOAD
273      000002      WC.IFEN   = BIT1   ;IFEN - FORMATTER ENABLE
274      000001      WC.IGO    = BIT0   ;GO
275

```

```

276
277      ;*
278      ;BSEL1 CODES FOR WRITE FORMAT
279      ;-
280      000200      WF.IHISP      = BIT7      ;IHISP - HIGH SPEED
281      000100      WF.IWRT      = BIT6      ;IWRT  - WRITE
282      000040      WF.IREV      = BIT5      ;IREV  - REVERSE
283      000020      WF.IWFM      = BIT4      ;IWFM  - WRITE FILE MARK
284      000010      WF.IEDIT     = BIT3      ;IEDIT - EDIT
285      000004      WF.IERASE    = BIT2      ;IERASE - ERASE
286      000002      WF.I3RESV    = BIT1      ;IRESV3 - RESERVED #3
287      000001      WF.I4RESV    = BIT0      ;IRESV4 - RESERVED #4
288
289      ;*
290      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
291      ;-
292      000200      MS.EXT      = BIT7      ;INVERT SENSE OF EXTENDED FEATURES SWITCH
293      000020      MS.RSFIFO    = BIT4      ;RESET FIFO AND INPUT PARITY ERRORR
294      000010      MS.RSTAPE    = BIT3      ;RESET TAPE STATUS IN 2 FLIP-FLOPS
295      000006      MS.ATTN     = BIT2!BIT1 ;ATTENTION TRIGGER FIELD
296      000001      MS.RSD      = BIT0      ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
297      ;*
298      ; MS.ATTN SUBCODES
299      ;-
300      000000      MSA.NOP     = 0*2      ;NO-OP (NOTHING TRIGGERED)
301      000002      MSA.VOL     = 1*2      ;SIMULATE ON-LINE/OFF-LINE TRANSIS'ION
302      000004      MSA.NRAM    = 2*2      ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
303      000006      MSA.FRAME   = 3*2      ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
304
305      ;*
306      ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
307      ;-
308      000200      NP.IR       = BIT7      ;INTERRUPT REQUEST (0-1 TRANSITION)
309      000100      NP.OUT      = BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
310      000040      NP.LOOP     = BIT5      ;ENABLE TRANSPORT LOOPBACK
311      000020      NP.WRP      = BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
312      ;*
313      ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
314      ;-
315      000200      S2.DIM      = BIT7      ;WORD #9 BYTE 2 DATA IN MISS
316      000100      S2.ILW     = BIT6      ;
317      000040      S2.OUTRDY   = BIT5      ;
318      000020      S2.INRDY   = BIT4      ;
319      000010      S2.ATIMR    = BIT3      ;
320      000004      S2.BTIMR    = BIT2      ;
321      000003      S2.UNDEF    = BIT1.BIT0 ;(UNDEFINED)
322      100000      S1.PARIN    = BIT15     ;WORD #8 BYTE 1 PARIN H
323      040000      S1.I2RESV   = BIT14     ;
324      020000      S1.I1RESV   = BIT13     ;
325      010000      S1.IEOT     = BIT12     ;
326      004000      S1.IIDENT   = BIT11     ;
327      002000      S1.ICER     = BIT10     ;
328      001000      S1.IFMK     = BIT9      ;
329      000400      S1.IHER     = BIT8      ;
330      000200      SO.ISPEED   = BIT7      ;WORD #8 BYTE 0 ISPEED H
331      000100      SO.IRDY     = BIT6      ;
332      000040      SO.IONL     = BIT5      ;

```

```

333      000020      SO.ILDP      = BIT4      ;           ILDP L
334      000010      SO.IDBY      = BIT3      ;           IDBY L
335      000004      SO.IRWD      = BIT2      ;           IRWD L
336      000002      SO.IFBY      = BIT1      ;           IFBY L
337      000001      SO.IFPT      = BIT0      ;           IFPT L
338
339      ;*
340      ;UNIBUS MAP DEFINATIONS
341      ;-
342      MMRO= 170200
343
344      .SBTTL SPECIAL MACROS AND OPDEFS.
345
346
347      ;*
348      ;SAVE GENERAL REGS 1 TO 5
349      ;-
350
351      .MACRO SAVREG
352      JSR R5,REGSAV
353      .ENDM
354
355      ;*
356      ; MACRO TO FORCE AN ERROR
357      ;-
358      .MACRO FORCERROR TAG,NOTSSR
359      .NLIST
360      .IF NDF LISTALL, .NLIST
361      .LIST
362      .IF B NOTSSR
363      MOV TSSR(R5),R1 ;READ TSSR
364      .ENDC
365      MOV FORCER,FORCER ;IS FORCER SET? (LEAVE C BIT ALONE)
366      BNE TAG ;BR IF YES
367      .NLIST
368      .IF NDF LISTALL, .LIST
369      .LIST
370      .ENDM
371
372      ;*
373      ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
374      ; WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
375      ; SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
376      ; FORCER TO 177777
377      ; TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
378      ;-
379      .MACRO FORCEEXIT TAG
380      .NLIST
381      .IF NDF LISTALL, .NLIST
382      .LIST
383      MOV FORCER,FORCER ;IS FORCER NEGATIVE?
384      BMI TAG ;BR IF YES
385      .NLIST
386      .IF NDF LISTALL, .LIST
387      .LIST
388      .ENDM
389      ;*

```

```

390 ; MACRO TO INCREMENT ERROR COUNTS
391 :-
392 .MACRO NEXT.ERRNO
393 .NLIST
394 :::.IIF NDF LISTALL, .NLIST
395 ERRNO=ERRNO+1
396 :::.IIF NDF LISTALL, .LIST
397 .LIST
398 .ENDM
399
400 ;+
401 ;MACRO TO PERFORM XOR
402 :-
403
404 .MACRO XOR A,B
405 MOV A,-(SP)
406 BIC B,(SP)
407 BIC A,B
408 BIS (SP)+,B
409 .ENDM
410
411 000000 EN=0 ; INITIALIZE ERROR NUMBER
412 .SBTTL FORCER - FORCE ERROR FLAG
413
414 ;
415 ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
416 ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
417 ;
418
419 002166 000000 FORCER:: 0 ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
420 ; - BY THE MACRO "IFERROR"). AN ERROR NEED NOT -
421 ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
422
423
424
425 .SBTTL GLOBAL DATA SECTION
426
427 ;++
428 ;THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
429 ;IN MORE THAN ONE TEST.
430 ;--
431
432 ;
433 ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
434 ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
435 ;
436 002170 000000 EPRTSW:: .WORD 0 ;PRINT SWITCH
437 002172 000000 UNITN:: .WORD 0 ;UNIT # UNDER TEST.
438 002174 000000 QVP:: .WORD 0 ;QUICK VERIFY FLAG.
439 002176 000000 CSRADDR:: .WORD 0 ;ADDRESS OF CSR FOR CURRENT DEVICE
440 002200 000224 IVEC:: .WORD 224 ;INTERRUPT VECTOR
441 002202 000200 IPRI:: .WORD PRI04 ;INTERRUPT PRIORITY.
442 002204 000000 TSTCNT:: .WORD 0 ;NUMBER OF TESTS RUN IN THIS PASS
443 002206 000000 LOOPCNT:: .WORD 0 ;REMAINING ITERATION COUNT FOR TEST
444 002210 000000 DEVCNT:: .WORD 0 ;NUMBER OF DEVICE UNDER TEST
445 002212 000000 FATFLG:: .WORD 0 ;SET IF FATAL ERROR IS DETECTED IN TEST
446 002214 000000 INTRECV:: .WORD 0 ;SET IF TAPE INTERRUPT WAS RECEIVED

```



```

447 002216 000000 EXTFEA:: .WORD 0 ;EXTENDED FEATURES SOFTWARE SW 0-OFF;1-ON
448 002220 000000 BENBSW:: .WORD 0 ;BUFFER ENABLE SWITCH SW 0-OFF;1-ON
449 002222 000000 EXPD:: .WORD 0 ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
450 002224 000000 RECV:: .WORD 0 ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
451 002226 000000 ERRHI:: .WORD 0 ;HIGH ADDRESS MEMORY ERROR
452 002230 000000 ERRLO:: .WORD 0 ;LOW ADDRESS MEMORY ERROR
453 002232 000000 RAMDATA:: .BLKW 16. ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
454 002272 000000 RAMSIZ:: .WORD 0 ;RAM DATA SIZE FOR PRAMPKT ROUTINE
455 002274 000000 RCVHIADD:: .WORD 0 ;RECEIVED BUFFER HIGH ADDRESS
456 002276 000000 RCVLOADD:: .WORD 0 ;RECEIVED BUFFER LOW ADDRESS
457 002300 000000 COUNT:: .WORD 0 ;TEST COUNT PATTERN
458 002302 000000 DATA:: .WORD 0 ;TEST DATA
459 002304 000000 TSTFLAG:: .WORD 0 ;TEST FLAG WORD
460 002306 000000 TSTPTR:: .WORD 0 ;TSTBLK POINTER
461 002310 000000 PRMNO:: .WORD 0 ;PRINT ROUTINE TEMP
462 002312 000000 EXPMSG:: .BLKB 100. ;EXPECTED MESSAGE BUFFER DATA
463 002456 000000 RECMSG:: .BLKB 100. ;RECEIVED MESSAGE BUFFER DATA
464 002622 000000 TMPBFR:: .BLKB 80. ;TEMPORARY STORAGE FOR PRINT

```

.SBTTL TSTBLK - TEST DATA TABLE

```

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

```

```

483 002742 TSTBLK::
484 002742 000000 .WORD 0 ;ALL ZEROS
485 002744 177777 .WORD 177777 ;ALL ONES
486 002746 000001 .WORD BIT0 ;DATA FOR WALKING ONES
487 002750 000002 .WORD BIT1
488 002752 000004 .WORD BIT2
489 002754 000010 .WORD BIT3
490 002756 000020 .WORD BIT4
491 002760 000040 .WORD BIT5
492 002762 000100 .WORD BIT6
493 002764 000200 .WORD BIT7
494 002766 000400 .WORD BIT8
495 002770 001000 .WORD BIT9
496 002772 002000 .WORD BIT10
497 002774 004000 .WORD BIT11
498 002776 010000 .WORD BIT12
499 003000 020000 .WORD BIT13
500 003002 040000 .WORD BIT14
501 003004 100000 .WORD BIT15
502 003006 177776 .WORD †CBIT0 ;DATA FOR WALKING ZEROS
503 003010 177775 .WORD †CBIT1

```

```

504 003012 177773          .WORD  †CBIT2
505 003014 177767          .WORD  †CBIT3
506 003016 177757          .WORD  †CBIT4
507 003020 177737          .WORD  †CBIT5
508 003022 177677          .WORD  †CBIT6
509 003024 177577          .WORD  †CBIT7
510 003026 177377          .WORD  †CBIT8
511 003030 176777          .WORD  †CBIT9
512 003032 175777          .WORD  †CBIT10
513 003034 173777         .WORD  †CBIT11
514 003036 167777         .WORD  †CBIT12
515 003040 157777         .WORD  †CBIT13
516 003042 137777         .WORD  †CBIT14
517 003044 077777         .WORD  †CBIT15
518 003046 125252         .WORD  125252          ;ALTERNATING ONES, ZEROS
519 003050 052525         .WORD  052525          ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
520
521
522
523
524
525
526
527 003052 000000 100000 000000 DUMMY: 0,100000,0,0 ;DUMMY DEVICE REGISTERS...
528 003062 000000 000000 000000 0,0,0,0,0,0,0,0,0 ;...FOR MULTI-UNIT CHECKOUT.
529
530
531
532 003102 000000          DUFLG::          .WORD  0          ;"DROPPED UNIT" FLAG.
533
534 003104 000000          NODEV::          .WORD  0          ;INHIBITS CODE IN "CLEAN-UP".
535
536 003106 000000          TEMP1::          .WORD  0          ;FLAG TO SAY NO DEVICE.
537 003110 000000          TEMP2::          .WORD  0          ;SOME TEMP LOCATIONS.
538 003112 000000          XXCOMM::          .WORD  0          ;XXDP+ COMM BLOCK POINTER.
539 003114 000000          FREE::          .WORD  0          ;1ST FREE MEMORY ADDRESS...
540 003116 000000          FRESIZ::          .WORD  0          ;...AND SIZE (IN WORDS).
541 003120 000000          FREEHI: .WORD  0          ;LAST WORD IN FREE SPACE
542 003122 000000          KTFLG::          .WORD  0          ;KT11, MEM AVAIL FLAG -
543
544
545 003124 000000          KTENABLE::          .WORD  0          ;- .WORD 0 = <24K OR NO KT -
546 003126 000000          NXMFLG::          .WORD  0          ;- NZ = >24K AND KT.
547 003130 000000          NXMLO::          .WORD  0          ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
548 003132 000000          NXMHI::          .WORD  0          ;SET IF WE CAN TEST CLEARED OTHERWISE
549 003134 000000          T23A::          .WORD  0          ;NXM LO ADDRESS BITS
550 003136 000000          T23B::          .WORD  0          ;NXM HI ADDRESS BITS FOR DAL'S 16-21
551 003140 000000          T3BFLG::          .WORD  0          ;PROCESSOR TYPE FLAG
552 003142 002000          PST32W::          .WORD  2000        ;PROCESSOR TYPE FLAG B
553 003144 000000          SIFLAG::          .WORD  0          ;TEST 3B FLAG †0
554 003146 000000          BADDAT::          .WORD  0          ;32W BLOCK ADDRESS FOR 32K START
555 003150 000000          GDDAT::          .WORD  0          ;
556 003152 000000          LOOPFL::          .WORD  0          ;ACTUAL DATA
557 003154
558 003154 000000          CTAB::          .WORD  0          ;EXPECTED DATA
559 003156 000000          CTABM::          .WORD  0          ;CONFIGURATION TABLES.
560 003160 000000          .WORD  0          ;CONFIG WORK.
          .WORD  0
          .WORD  0

```

```

561 003162 000000          .WORD 0
562 003164 177777          .WORD -1          ;END OF MEM TABLE.
563 003166
564          CTABE::
;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
565          :
566          :          0          =          UNIT NOT TESTED
567          :          100000 =          UNIT ONLINE, NO ERRORS
568          :          10XXXX =          UNIT ONLINE, ENCOUNTERED XXXX ERRORS
569          :          160000 =          UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
570          :          160001 =          UNIT DROPPED, NOT IDLE AT START
571          :          14XXXX =          UNIT DROPPED, ENCOUNTERED XXXX ERRORS
572          :
573 003166          ERTABL:          .BLKW 64.
574 003366 000000          ERTABE:          .WORD 0
575
576 003370 000000          SKIPT:          .WORD 0          ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST
577
578          .SBTTL GLOBAL TEXT MESSAGES
579          ;++
580          ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
581          ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
582          ; MORE THAN ONE TEST.
583          ;--
584
585
586
587          ;+
588          ;NAMES OF DEVICES SUPPORTED
589          ;-
590
591 003372          DEVTYP <TSU05>
003372          L$DVTYP::
003372          124          123          125          .ASCIZ /TSU05/
          .EVEN
592
593
594
595
596
597
598          ;+
599          ;TEST DESCRIPTION
600          ;-
601 003400          DESCRIPT <***** TSU05 DIAG PART 3 - CHK CABLES-TRANSPORT IF ERR *****>
003400          L$DESC::
003400          052          052          052          .ASCIZ /***** TSU05 DIAG PART 3 - CHK CABLES-TRANSPORT IF ERR *****/
          .EVEN
602
603
604
605
606
607
608          ;+
609          ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
610          ;-
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625 003472 003532' 003535' 003541' TSSRBIT::          .WORD 1$,2$,3$,4$,5$,6$,7$,8$
626 003512 003573' 003577' 003603'          .WORD 9$,10$,11$,12$,13$,14$,15$,16$
627 003532          123          103          000 1$:          .ASCIZ 'SC'
628 003535          102          111          105 2$:          .ASCIZ 'BIE'
629 003541          123          103          105 3$:          .ASCIZ 'SCE'
630 003545          122          115          122 4$:          .ASCIZ 'RMR'
631 003551          116          130          115 5$:          .ASCIZ 'NXM'
632 003555          116          102          101 6$:          .ASCIZ 'NBA'

```

```

633 003561 102 111 124 7$: .ASCIZ 'BIT9'
634 003566 102 111 124 8$: .ASCIZ 'BIT8'
635 003573 123 123 122 9$: .ASCIZ 'SSR'
636 003577 117 106 114 10$: .ASCIZ 'OFL'
637 003603 102 111 124 11$: .ASCIZ 'BIT5'
638 003610 102 111 124 12$: .ASCIZ 'BIT4'
639 003615 102 111 124 13$: .ASCIZ 'BIT3'
640 003622 102 111 124 14$: .ASCIZ 'BIT2'
641 003627 102 111 124 15$: .ASCIZ 'BIT1'
642 003634 102 111 124 16$: .ASCIZ 'BIT0'
643 .EVEN
644 003642 124 123 123 SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
645 003675 124 123 123 SFHERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
646 003730 040 040 116 NXR: .ASCIZ / NON-EXISTANT DEVICE REGISTER/
647 003767 045 101 040 NXR: .ASCIZ /#A ADDRESS: #06/
648 004010 045 101 040 TSSX: .ASCIZ /#A TSBA,TSSR EXP'D: #06#A,#06#N/
649 004050 045 101 040 TSSX: .ASCIZ /#A TSBA,TSSR REC'D: #06#A,#06#N/
650 004107 045 116 045 FUSI: .ASCIZ /#N#A/
651 004113 040 040 125 USI: .ASCIZ / UNEXPECTED INTERRUPT/
652 004142 040 040 111 NSI: .ASCIZ / INTERRUPT EXPECTED, NOT RECEIVED/
653 004205 045 116 045 FNOINTR: .ASCIZ /#N#A/
654 004211 040 040 116 NOINTR: .ASCIZ / NO INTERRUPT WAS GENERATED/
655 004246 040 040 111 IFAULT: .ASCIZ / INTERRUPT FAULT/
656 004270 045 101 040 INTX: .ASCIZ /#A CPU PC: #06#A TSBA: #06/
657 004325 040 040 042 NOINIT: .ASCIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
658 004377 040 040 042 NSINIT: .ASCIZ / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
659 004447 040 040 042 BRINIT: .ASCIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
660
661 004517 000 NUL: .ASCIZ //
662 004520 045 116 000 NULCR: .ASCIZ /#N/
663 004523 045 101 040 EXPGOT: .ASCIZ /#A EXP'D: #06#A, REC'D: #06/
664 004557 045 116 045 EXPGT2: .ASCIZ /#N#A EXP'D: #06#A, #06#N#A REC'D: #0#A, #06/
665 004633 045 101 040 DUAD12: .ASCIZ /#A REG(W) WRITTEN TO: #06#A REG(R) READ; EXP'D: #06#A, REC'D: #06/
666 004735 122 101 115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
667 005003 040 040 103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
668 005046 127 122 111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
669 005103 124 123 123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
670 005176 124 123 123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
671 005270 106 101 124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
672 005362 105 122 122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
673 005450 045 116 045 NOMEM: .ASCIZ '#N#A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****#N'
674 .EVEN
675
676 .SBTTL GLOBAL ERROR REPORT SECTION
677
678
679 ;**
680 ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
681 ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
682 ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
683 ;--
684 005544 BGNMSG NXRERR ;NON-EXISTANT DEVICE REGISTER.
005544
685 005544 PRINTX #NXRX,NODEV ;NODEV = NEXM ADDRESS.
005544 013746 003104'
005550 012746 003767'
005554 012746 000002
MOV NODEV,-(SP)
MOV #NXRX,-(SP)
MOV #2,-(SP)

```

005560	010600		MOV	SP,R0	
005562	104415		TRAP	C\$PNTX	
005564	062706	000006	ADD	#6,SP	
686 005570	004737	005576'	JSR	PC,EXTEND	; PRINT EXTENSION IF REQUIRED.
687 005574			ENDMSG		
005574			L10002:		
005574	104423		TRAP	C\$MSG	
688					
689					
690					
691					
692					
693					
694 005576	005727				
695 005600	000000				
696 005602	001402				
697 005604	004777	177770			
698 005610					
005610	012746	004520'			
005614	012746	000001			
005620	010600				
005622	104415				
005624	062706	000004			
699 005630	000207				
700					
701					
702					
703					
704					
705					
706					
707					
708					
709					
710					
711					
712					
713					
714					
715					
716					
717					
718					
719 005632					
720 005632					
721 005636	010104				
722 005640					
005640	010446				
005642	012746	006315'			
005646	012746	000002			
005652	010600				
005654	104414				
005656	062706	000006			
723 005662	010400				
724 005664	004737	015744'			
725 005670	103410				
726 005672					

```

MOV SP,R0
TRAP C$PNTX
ADD #6,SP
JSR PC,EXTEND ; PRINT EXTENSION IF REQUIRED.
ENDMSG
L10002:
TRAP C$MSG

; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
; TO ANY OF THE ABOVE ERROR SIGNATURES.
;
; EXTEND: TST (PC)+
; EXTA: 0 ; 0 = NO EXTENSION.
; BEQ 1$
; JSR PC,EXTA ; APPEND EXTENSION TEXT.
1$: PRINTX #NULCR ; PRINT A BLANK LINE
MOV #NULCR,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #4,SP
RTS PC

.SBTTL PRITSSR - PRINT TSSR CONTENTS

; *
; ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
; THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
; BY A MESSAGE PRINTING ROUTINE
;
; INPUTS:
; R1 CONTENTS OF TSSR
;
; SUBORDINATE ROUTINES:
; CHKAMB CHECK FOR AMBIGUOUS CONTENTS
;
; -
PRITSSR:
SAVREG ; SAVE GENERAL REGISTERS
MOV R1,R4 ; SAVE THE TSSR CONTENTS
PRINTB #TSSRFOR,R4 ; PRINT THE CONTENTS OF TSSR
MOV R4,-(SP)
MOV #TSSRFOR,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #6,SP
MOV R4,R0 ; GET TSSR BACK FOR CHKAMB
JSR PC,CHKAMB ; ARE CONTENTS AMBIGUOUS ?
BCS 5$ ; BRANCH IF NOT
PRINTX #AMBTSSR ; SHOW CONTENTS ARE AMBIGUOUS

```

```

005672 012746 006535'      MOV      #AMBTSSR,-(SP)
005676 012746 000001      MOV      #1,-(SP)
005702 010600      MOV      SP,R0
005704 104415      TRAP     C$PNTX
005706 062706 000004      ADD      #4,SP
727 005712 010403      5$:     MOV      R4,R3      ;CONTENTS OF TSSR
728 005714 042703 001476      BIC      #HIADDR!FATERR!TERCLS,R3 ;CLEAR ALL MULTIPLE BIT FIELDS
729 005720 001434      BEQ      20$      ;NO BITS ARE SET
730 005722 012702 002622'      MOV      #TMPBFR,R2      ;TEMPORARY ASCII BUFFER
731 005726 012701 003472'      MOV      #TSSRBIT,R1      ;ASCII EQUIVALENT OF BITS
732 005732 005703      10$:    TST      R3      ;REMAINING BITS TO CONVERT
733 005734 001413      BEQ      15$      ;BRANCH WHEN ALL ARE DONE
734 005736 000241      CLC      ;CLEAR CARRY FOR SHIFT
735 005740 006103      ROL      R3      ;SHIFT NEXT BIT TO CARRY
736 005742 103006      BCC      13$      ;BRANCH IF BIT NOT SET
737 005744 011100      MOV      (R1),R0      ;POINTER TO BIT DEFINITION
738 005746 112022      11$:    MOVVB   (R0)+,(R2)+      ;MOVE ASCII TO BUFFER
739 005750 001376      BNE      11$      ;MOVE ALL BITS
740 005752 112762 000054 177777      MOVVB   #' ,,-1(R2)      ;INSERT A COMMA TO TERMINATE
741 005760 005721      13$:    TST      (R1)+      ;POINT TO NEXT DESCRIPTION
742 005762 000763      BR       10$      ;GET THE REMAINING BITS
743 005764 105042      15$:    CLRB    -(R2)      ;TERMINATE THE LINE
744 005766      PRINTX  #TSSDEF,#TMPBFR ;PRINT THE BIT DEFINITIONS
005766 012746 002622'      MOV      #TMPBFR,-(SP)
005772 012746 006506'      MOV      #TSSDEF,-(SP)
005776 012746 000002      MOV      #2,-(SP)
006002 010600      MOV      SP,R0
006004 104415      TRAP     C$PNTX
006006 062706 000006      ADD      #6,SP
745
746 006012 010403      20$:    MOV      R4,R3      ;GET THE TSSR CONTENTS
747 006014 042703 177761      BIC      #+CTERCLS,R3      ;CLEAR ALL BUT TERMINATION
748 006020 016303 006576'      MOV      TCOCOD(R3),R3      ;GET THE TERMINATION CODE MEANING
749 006024      PRINTX  #TCOASC,R3      ;PRINT THE TERMINATION CODE
006024 010346      MOV      R3,-(SP)
006026 012746 006376'      MOV      #TCOASC,-(SP)
006032 012746 000002      MOV      #2,-(SP)
006036 010600      MOV      SP,R0
006040 104415      TRAP     C$PNTX
006042 062706 000006      ADD      #6,SP
750 006046 010403      MOV      R4,R3      ;TSSR CONTENTS AGAIN
751 006050 042703 177717      BIC      #+CFATERR,R3      ;CLEAR ALL BUT FATAL TERMINATION
752 006054 001416      BEQ      25$      ;DON'T PRINT IF ZERO
753 006056 006203      ASR      R3
754 006060 006203      ASR      R3
755 006062 006203      ASR      R3
756 006064 016303 007136'      MOV      TSFCOD(R3),R3      ;ALINE TERMINATION CODE FOR INDEX
757 006070      PRINTX  #TFCASC,R3      ;GET THE FATAL TERMINATION CODE
006070 010346      MOV      R3,-(SP)      ;PRINT THE FATAL TERMINATION CODE
006072 012746 006437'      MOV      #TFCASC,-(SP)
006076 012746 000002      MOV      #2,-(SP)
006102 010600      MOV      SP,R0
006104 104415      TRAP     C$PNTX
006106 062706 000006      ADD      #6,SP
758 006112 042704 176377      25$:    BIC      #+CHIADDR,R4      ;CLEAR ALL BUT EXTENDED ADDRESS
759 006116 001411      BEQ      30$      ;DON'T PRINT IF ZERO
760 006120      PRINTX  #TEXASC,R4      ;PRINT THE EXTENDED ADDRESS BITS

```

```

006120 010446          MOV      R4,-(SP)
006122 012746 006335'  MOV      #TEXASC,-(SP)
006126 012746 000002   MOV      #2,-(SP)
006132 010600          MOV      SP,R0
006134 104415          TRAP     C$PNTX
006136 062706 000006   ADD      #6,SP
761 006142 013703 002170' 30$:  MOV      EPRTSW,R3          ;PRINT MEASGE BUFFER ADDRESS
762 006146          PRINTX  R3          ;PRINT PROPER MESSAGE
006146 010346          MOV      R3,-(SP)
006150 012746 000001   MOV      #1,-(SP)
006154 010600          MOV      SP,R0
006156 104415          TRAP     C$PNTX
006160 062706 000004   ADD      #4,SP
763 006164 000207          RTS      PC          ;RETURN TO CALLER
764
775 006166          045      116      045  EPRT1: .ASCIZ  'N$A *****CHECK CABLES BETWEEN M7455 AND TRANSPORT*****'
776 006256          045      116      045  EPRT2: .ASCIZ  'N$A *****CHECK TRANSPORT*****'
782 006315          045      116      045  TSSRFOR: .ASCIZ  'N$A TSSR = #06'
783 006335          045      116      045  TEXASC: .ASCIZ  'N$A Extended Address Bits = #06'
784 006376          045      116      045  TCOASC: .ASCIZ  'N$A Termination Class Code = #T'
785 006437          045      116      045  TFCASC: .ASCIZ  'N$A Fatal Termination Class Code = #T'
786 006506          045      116      045  T$SDEF: .ASCIZ  'N$A TSSR Bits Set: #T'
787 006535          045      116      045  AMBTSSR: .ASCIZ  'N$A TSSR Contents Are Ambiguous'
788
789 006576 006616' 006641' 006667' TCOCOD: .EVEN
790 006616          116      157      162  1$: .WORD  1$,2$,3$,4$,5$,6$,7$,8$
791 006641          124      145      162  1$: .ASCIZ  'Normal Termination'
792 006667          124      141      160  2$: .ASCIZ  'Termination Condition'
793 006711          106      165      156  3$: .ASCIZ  'Tape Status Alert'
794 006731          122      145      143  4$: .ASCIZ  'Function Reject'
795 007013          122      145      143  5$: .ASCIZ  'Recoverable Error - Tape Position One Record Down'
796 007062          125      156      162  6$: .ASCIZ  'Recoverable Error - Tape Was Not Moved'
797 007106          106      141      164  7$: .ASCIZ  'Unrecoverable Error'
798
799
800 007136 007146' 007202' 007213' TSFCOD: .EVEN
801 007146          111      156      164  1$: .ASCIZ  'Fatal Controller Error'
802 007202          122      145      163  1$: .ASCIZ  'Internal Diagnostic Failure'
803 007213          102      165      163  2$: .ASCIZ  'Reserved'
804 007257          122      145      163  3$: .ASCIZ  'Bus Interface or Sanity Check Error'
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821

```

.SBTTL PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

```

;+
;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
;

```

;INPUT:

```

;
;      R0      NUMBER OF WORDS IN PACKET
;      R3      HIGH ORDER COMMAND PACKET ADDRESS
;      R4      ADDRESS OF COMMAND PACKET
;

```

; NOTE: R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.

```

822 007270          PRIPKT::
823 007270          SAVREG          ;SAVE THE REGISTERS
824 007274 010005   MOV          R0,R5 ;SAVE NO. OF WORDS IN PACKET
825 007276 005737 003124'  TST          KTENABLE ;ABOVE 28K UNDER TEST?
826 007302 001001   BNE          10$ ;BR IF YES
827 007304 005003   CLR          R3 ;SET HIGH ORDER ADDRESS TO 0
828 007306 010301 10$:  MOV          R3,R1 ;COPY HIGH ORDER ADDRESS
829 007310 010400   MOV          R4,R0 ;GET LOWER ADDRESS
830 007312 006100   ROL          R0 ;SHIFT BIT 15 INTO C BIT
831 007314 006101   ROL          R1 ;AND INTO HIGH ORDER.
832 007316          PRINTB         #PKTADD,R1,R4 ;PRINT PACKET ADDRESS
      007316 010446   MOV          R4,-(SP)
      007320 010146   MOV          R1,-(SP)
      007322 012746 007454'  MOV          #PKTADD,-(SP)
      007326 012746 000003   MOV          #3,-(SP)
      007332 010600   MOV          SP,R0
      007334 104414   TRAP         C$PNTB
      007336 062706 000010   ADD          #10,SP
833 007342 010300 15$:  MOV          R3,R0 ;GET HIGH ORDER ADDRESS
834 007344 001404   BEQ          20$ ;BR IF NOT ABOVE 28K.
835 007346 010401   MOV          R4,R1 ;GET LOW ORDER ADDRESS
836 007350 004737 017220'  JSR          PC,SETMAP ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
837 007354 010004   MOV          R0,R4 ;GET RETURNED PAR6 ADDRESS BIAS
838 007356 005001 20$:  CLR          R1 ;SAVE WORD NUMBER
839 007360 012402 25$:  MOV          (R4)+,R2 ;GET PACKET CONTENTS
840 007362          PRINTB         #PKTFRM,R1,R2 ;PRINT THE DATA
      007362 010246   MOV          R2,-(SP)
      007364 010146   MOV          R1,-(SP)
      007366 012746 007416'  MOV          #PKTFRM,-(SP)
      007372 012746 000003   MOV          #3,-(SP)
      007376 010600   MOV          SP,R0
      007400 104414   TRAP         C$PNTB
      007402 062706 000010   ADD          #10,SP
841 007406 005201   INC          R1 ;NEXT WORD NUMBER
842 007410 020105   CMP          R1,R5 ;DONE ALL PACKET WORDS?
843 007412 002762   BLT          25$ ;LOOP TILL ALL DONE
844 007414 000207   RTS          PC ;RETURN
845
846 007416          045          116          045 PKTFRM: .ASCIZ  '#N$A Packet Word #D1$A = #06'
847 007454          045          116          045 PKTADD: .ASCIZ  '#N$A Packet Address = #01$05'
848
849
850
851          .SBTTL  PRIBXOR - PRINT EXPD, RECV AND XOR BYTE
852
853          ;*
854          ;
855          ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
856          ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
857          ;
858          ;INPUTS:
859          ;
860          ;          R1          RECEIVED DATA
861          ;          R2          EXPECTED DATA
862          ;
863          ;OUTPUT:
864          ;
    
```



```

865          ;      RO      XOR OF EXPECTED/RECEIVED DATA
866          ;
867          ; -
868
869 007512    PRIBXOR::
870 007512    SAVREG          ;SAVE THE REGISTERS
871 007516    010203        MOV      R2,R3          ;EXPECTED DATA
872 007520    XOR          R1,R3          ;FORM THE EXCLUSIVE OR
873 007530    012700    177400    MOV      #+C<377>,RO      ;BYTE MASK
874 007534    040001        BIC      RO,R1          ;SAVE LOW BYTE RECV
875 007536    040002        BIC      RO,R2          ;SAVE LOW BYTE EXPD
876 007540    040003        BIC      RO,R3          ;SAVE LOW BYTE XOR
877 007542    PRINTB      #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
      007542    010346        MOV      R3,-(SP)
      007544    010146        MOV      R1,-(SP)
      007546    010246        MOV      R2,-(SP)
      007550    012746    007574'    MOV      #XORBFOR,-(SP)
      007554    012746    000004        MOV      #4,-(SP)
      007560    010600        MOV      SP,RO
      007562    104414        TRAP     C$PNTB
      007564    062706    000012        ADD      #12,SP
878 007570    010300        MOV      R3,RO          ;RO HAS XOR ON RETURN
879 007572    000207        RTS       PC           ;RETURN TO CALLER
880
881 007574    045      116      045 XORBFOR: .ASCIZ 'NNA EXPD: 03A RECV: 03A XOR: 03'
882          .EVEN
883
884
885          .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR
886
887          ;+
888          ;
889          ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
890          ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
891          ;
892          ;INPUTS:
893          ;
894          ;      R1      RECEIVED DATA
895          ;      R2      EXPECTED DATA
896          ;
897          ;OUTPUT:
898          ;
899          ;      RO      XOR OF EXPECTED/RECEIVED DATA
900          ;
901          ; -
902
903 007642    PRIBXOR::
904 007642    SAVREG          ;SAVE THE REGISTERS
905 007646    010203        MOV      R2,R3          ;EXPECTED DATA
906 007650    XOR          R1,R3          ;FORM THE EXCLUSIVE OR
907 007660    PRINTB      #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
      007660    010346        MOV      R3,-(SP)
      007662    010146        MOV      R1,-(SP)
      007664    010246        MOV      R2,-(SP)
      007666    012746    007712'    MOV      #XORFOR,-(SP)
      007672    012746    000004        MOV      #4,-(SP)
      007676    010600        MOV      SP,RO
    
```

```

007700 104414 TRAP C:PNTB
007702 062706 000012 ADD #12,SP
908 007706 010300 MOV R3,R0 ;R0 HAS XOR ON RETURN
909 007710 000207 RTS PC ;RETURN TO CALLER
910
911 007712 045 116 045 XORFOR: .ASCIZ '##N##A EXPD: ##06##A RECV: ##06##A XOR: ##06'
912 .EVEN
913
914 .SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
915
916 ;*
917 ;
918 ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
919 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
920 ;
921 ;INPUTS:
922 ;
923 ; R0 OCTAL VALUE TO CONVERT
924 ; R1 TABLE OF POINTERS TO ASCII EQUIVALENT
925 ;
926 ;-
927
928 007760 PRIEQU:
929 007760 SAVREG ;SAVE THE REGISTERS
930 007764 000207 RTS PC ;RETURN TO CALLER
931
932
933
934
935 .SBTTL PRIRAM - PRINT RAM ADDRESS
936
937 ;*
938 ;
939 ;PRINT CONTROLLER RAM ADDRESS.
940 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
941 ;
942 ;INPUTS:
943 ;
944 ; R4 RAM ADDRESS
945 ;-
946
947 007766 PRIRAM:
948 007772 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
007772 010446 PRINTB @RAMFOR,R4 ;PRINT RAM ADDRESS IN ERROR
007774 012746 010016' MOV R4,-(SP)
010000 012746 000002 MOV @RAMFOR,-(SP)
010004 010600 MOV #2,-(SP)
010006 104414 TRAP C:PNTB
010010 062706 000006 ADD #6,SP
949 010014 000207 RTS PC ;RETURN
950
951 010016 045 116 045 RAMFOR: .ASCIZ '##N##A CONTROLLER RAM ADDRESS * ##06'
952 .EVEN
953
954
955 .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
956 ;*
    
```

```

957
958 ;PRINT MEMORY ADDRESS
959 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
960
961 ; IMPLICIT INPUTS
962
963 ; ERRHI - HIGH ORDER ADDRESS
964 ; ERRLO - LOW ORDER ADDRESS
965
966 ;-
967 PRIADD:
968 010060 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
969 010060 MOV ERRHI,R0 ;GET HIGH ADDRESS
970 010064 013700 002226' MOV ERRLO,R1 ;GET LOW ADDRESS
971 010070 013701 002230' MOV R1,R2 ;COPY LOW ADDRESS
972 010074 010102 ROL R1 ;SHIFT BIT 15 TO C BIT
973 010076 006101 ROL R0 ;SHIFT INTO HIGH ORDER
974 010100 006100 PRINTB @PRIA0,R0,R2 ;PRINT MEMORY ADDRESS IN ERROR
975 010102 010246 MOV R2,-(SP)
976 010104 010046 MOV R0,-(SP)
977 010106 012746 010130' MOV @PRIA0,-(SP)
978 010112 012746 000003 MOV @3,-(SP)
979 010116 010600 MOV SP,R0
980 010120 104414 TRAP C:PNTB
981 010122 062706 000010 ADD @10,SP
982 010126 000207 RTS PC ;RETURN
983
984 977 010130 045 116 045 PRIA0: .ASCIZ 'MEMORY ERROR ADDRESS = 0105'
985 .EVEN
986
987 .SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
988
989 ;*
990 ;PRINT MEMORY ADDRESS
991 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
992
993 ; IMPLICIT INPUTS
994
995 ; ERRHI - HIGH ORDER ADDRESS
996 ; ERRLO - LOW ORDER ADDRESS
997
998 ;-
999 PRITADD:
1000 010174 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
010174 MOV ERRHI,R2 ;GET HIGH ADDRESS
010200 013702 002226' MOV ERRLO,R1 ;GET LOW ADDRESS
010204 013701 002230' MOV R1,R2 ;COPY LOW ADDRESS
;MOV R1,R2 ;COPY LOW ADDRESS
;ROL R1 ;SHIFT BIT 15 TO C BIT
;ROL R0 ;SHIFT INTO HIGH ORDER
PRINTB @PRIT0,R1 ;PRINT MEMORY ADDRESS LOW IN ERROR
MOV R1,-(SP)
MOV @PRIT0,-(SP)
MOV @2,-(SP)
MOV SP,R0
TRAP C:PNTB
ADD @6,SP

```

```

1001 010232          PRINTB @PRIT1,R2          ;PRINT MEMORY ADDRESS HIGH IN ERROR
      010232 010246      MOV R2,-(SP)
      010234 012746 010321'  MOV @PRIT1,-(SP)
      010240 012746 000002    MOV @2,-(SP)
      010244 010600      MOV SP,R0
      010246 104414      TRAP C#PNTB
      010250 062706 000006    ADD @6,SP
1002 010254 000207      RTS PC          ;RETURN
1003
1004 010256      045      116      045 PRIT0: .ASCIZ '#N#A MEMORY TEST ADDRESS LOW = #06'
1005 010321      045      116      045 PRIT1: .ASCIZ '#N#A MEMORY TEST ADDRESS HIGH = #06'
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044 010366          .SBTTL SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND
1045 010366          ;*
1046 010372 012737 000764 010560' ;ROUTINE TO ISSUE A SPACE RECORDS
1047 010400 012737 140010 010550' ;COMMAND (FORWARD OR REVERSE)
1048 010406 005703      ;INPUT:
1049 010410 100403      ;
1050 010412 010337 010552' ; R3 NUMBER OF RECORDS TO BE SPACED OVER
1051 010416 000407      ; BIT15 CONTROLS DIRECTION
                        ; BIT15 = 0 IS FORWARD
                        ; BIT15 = 1 IS REVERSE
                        ; R5 FIRST DEVICE UNIBUS ADDRESS
                        ; REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
                        ;OUTPUT:
                        ; CARRY SET - SPACE RECORDS COMMAND OK
                        ; CLR - SPACE RECORDS FAILED
                        ;
                        ; R0 THE CONTENTS OF R4 IS MOVED TO R0
                        ;IMPLICIT OUTPUT:
                        ; TAPE HAS BEEN MOVED
                        ;SIDE EFFECTS:
                        ;
                        ;-
                        SPACE::
1045 010366          SAVREG          ;SAVE THE GENERAL REGISTERS
1046 010372          MOV @500.,SDELAY ;SET UP DELAY
1047 010400          MOV @140010,80$ ;SET UP COMMAND, SPACE FORWARD
1048 010406          TST R3          ;CHECK FOR DIRECTION
1049 010410          BMI S$          ;BR, IF REVERSE INDICATED
1050 010412          MOV R3,90$      ;LOAD UP NUMBER OF RECORDS TO SPACE
1051 010416          BR 10$          ;GO DO COMMAND

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:54
 SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

SEQ 043

```

1052 010420 042703 100000      5$:   BIC   #BIT15,R3           ;CLEAR DIRECTION BIT
1053 010424 010337 010552'    MOV   R3,90$           ;LOAD UP NUMBER OF RECORDS TO SPACE
1054 010430 052737 000400 010550'  BIS   #BIT8,80$       ;SET REVERSE BIT IN COMMAND PACKET
1055 010436 012704 010550'    10$:  MOV   #80$,R4         ;SET UP R4 WITH PACKET ADDRESS
1056 010442 010465 000000      MOV   R4,TSDB(R5)     ;SEND OUT COMMAND
1057 010446 004737 016150'    15$:  JSR   PC,WAITF      ;WAIT FOR SSR
1058 010452 103420      BCS   20$             ;BR, IF SSR IS SET AND OK
1059 010454      DELAY  250           ;DELAY ABOUT .25 SECONDS
      010454 012727 000250      MOV   #250,(PC)+
      010460 000000      .WORD  0
      010462 013727 002116'    MOV   L$DLY,(PC)+
      010466 000000      .WORD  0
      010470 005367 177772      DEC   -6(PC)
      010474 001375      BNE   .-4
      010476 005367 177756      DEC   -22(PC)
      010502 001367      BNE   .-20
1060 010504 005337 010560'    DEC   SDELAY         ;BUMP DELAY COUNTER DOWN
1061 010510 001356      BNE   15$           ;BR, IF MORE DELAY
1062 010512 000411      BR    60$           ;BR IF TROUBLE CARRY = CLEAR
1063 010514 016501 000002    20$:  MOV   TSSR(R5),R1 ;READ TSSR
1064 010520 012702 000200      MOV   #SSR,R2       ;SET UP EXPECTED
1065 010524 020201    25$:  CMP   R2,R1         ;ARE THEY OK
1066 010526 001401      BEQ   40$           ;BR, IF EQUAL = OK
1067 010530 000402      BR    60$           ;TROUBLE EXIT
1068 010532 000261    40$:  SEC             ;SET CARRY NO TROUBLE
1069 010534 000401      BR    70$           ;EXIT
1070 010536 000241    60$:  CLC             ;CARRY CLEAR = ERROR
1071 010540      70$:
1072 010540 010400      MOV   R4,R0         ;PASS PACKET ADDRESS
1073 010542 000207      RTS   PC           ;RETURN
1074
1075
1076
1077
1078      ;PACKET FOR SPACE COMMAND
1079
1081 010544      .BLKB  10-<.-TSV2&7>
1083
1084      ;COMMAND WORD
1085 010550 000000    80$:  .WORD
1086      ;NUMBER OF RECORDS TO BE SPACED OVER WORD
1087 010552 000000    90$:  .WORD
1088 010554 000000      .WORD
1089 010556 000000      .WORD
1090 010560 000000    SDELAY: .WORD  0           ;DELAY COUNTER
1091      .EVEN
1092
1093
1094      .SBTTL  WRCHR - WRITE CHARACTERISTICS COMMAND
1095
1096
1097
1098      ;*
1099      ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1100      ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1101      ;INPUT:
1102

```

```

1103      :      R4      ADDRESS OF PACKET FROM TEST
1104      :      R5      FIRST DEVICE UNIBUS ADDRESS
1105      :      REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1106      :
1107      :OUTPUT:
1108      :
1109      :      RO      TSSR CONTENTS
1110      :      CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
1111      :      CLR     WRITE CHARACTERISTICS FAILED
1112      :
1113      :IMPLICIT OUTPUT:
1114      :
1115      :      MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
1116      :      SOFTWARE SWITCHES SET AS FOLLOWS:
1117      :      EXTFEA = EXTENDED FEATURES PRESENT
1118      :      BENBSW = BUFFER ENABLE SWITCH ON OR OFF
1119      :
1120      :
1121      :SIDE EFFECTS:
1122      :
1123      :
1124      : -
1125      :
  
```

```

1126 010562 WRTCHR::
1127 010562      SAVREG
1128 010566 005037 002220'      CLR      BENBSW      ;SAVE THE GENERAL REGISTERS
1129 010572 005037 002216'      CLR      EXTFEA      ;CLEAR BUFFER ENABLE SWITCH
1130 010576 010465 000000      10$: MOV      R4,TSDB(R5)  ;CLEAR EXTENDED FEATURES SW SWITCH
1131 010602 004737 016236'      JSR      PC,CHKTSSR  ;SEND OUT COMMAND
1132 010606 103401              BCS      20$         ;WAIT FOR SSR
1133 010610 000435              BR       60$         ;BR, IF SSR IS SET AND OK
1134 010612 016501 000002      20$: MOV      TSSR(R5),R1 ;BR IF TROUBLE CARRY = CLEAR
1135 010616 012702 000200      MOV      #SSR,R2    ;READ TSSR
1136 010622 032701 000100      MOV      #OFL,R1   ;SET UP EXPECTED
1137 010626 001402              BEQ      25$         ;WAS OFF LINE SET IN TSSR
1138 010630 052702 000100      BIS      #OFL,R2   ;BR, IF NO OFL SET
1139 010634 020201              CMP      R2,R1      ;MAKE THEM LOOK ALIKE
1140 010636 001401              BEQ      40$         ;ARE THEY OK
1141 010640 000421              BR       60$         ;BR, IF EQUAL = OK
1142 010642 062704 000010      40$: ADD      #8,R4    ;TROUBLE EXIT
1143 010646 011403              MOV      (R4),R3   ;POINT TO WRT CHARA DATA PACKET
1144 010650 032763 000200 000012  BIT      #X2.EXTF,XST2(R3) ;GET ADDRESS OF MESSAGE BUFFER
1145 010656 001402              BEQ      45$         ;EXTENDED FEATURES BIT SET?
1146 010660 005237 002216'      INC      EXTFEA    ;BR IF NO
1147 010664              ;SET EXTENDED FEATURES SW SWITCH
1148 010664 032763 000100 000012  45$: BIT      #X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
1149 010672 001402              BEQ      50$         ;BR, IF SWITCH NOT SET
1150 010674 005237 002220'      INC      BENBSW    ;SET SOFTWARE SWITCH FOR ENABLED
1151 010700              50$: SEC
1152 010700 000261              BR       70$         ;SET CARRY NO TROUBLE
1153 010702 000401              BR       70$         ;EXIT
1154 010704 000241              CLC
1155 010706 016500 000002      60$: MOV      TSSR(R5),RO ;CARRY CLEAR = ERROR
1156 010712 000207      70$: MOV      TSSR(R5),RO ;RETURN TSSR CONTENTS
1157      :      RTS      PC      ;RETURN
1158
1159      .SBTTL REWIND - POSITION TAPE (REWIND) COMMAND
  
```

```

1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187 010714
1188 010714
1189 010720 012704 011010'
1190 010724 010465 000000
1191 010730 012703 000550
1192 010734 004737 016150'
1193 010740 103417
1194 010742
    010742 012727 000372
    010746 000000
    010750 013727 002116'
    010754 000000
    010756 005367 177772
    010762 001375
    010764 005367 177756
    010770 001367
1195 010772 005303
1196 010774 001357
1197 010776 000241
1198 011000 010400
1199 011002 000207
1200
1201
1203 011004
1205 011010
1206 011010 102010
1207 011012 000000
1208
1209
1210

```

```

; *
; THIS ROUTINE WILL REWIND THE SELECTED TAPE.
;
; CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
; TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
; SSR TO SET IN THE TSSR
;
; CALLING SEQUENCE:
;
; DO A SOFT INIT
; DO A WRITE CHARACTERISTICS
; JSR PC,REWIND
;
; INPUT:
;
; R5 FIRST DEVICE UNIBUS ADDRESS
;
; OUTPUT
;
; R0 THE CONTENTS OF R4 IS PASSED TO R0
;
; -
REWIND::
    SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
    MOV #RWPACK,R4                        ;GET PACKET ADDRESS
    MOV R4,TSDB(R5)                       ;SEND PACKET ADDRESS TO EXECUTE
    MOV #360.,R3                          ;ENOUGH TIME FOR 2400' REEL TO REWIND
10$: JSR PC,WAITF                          ;WAIT FOR SSR TO SET
    BCS 20$                               ;LEAVE WHEN SSR IS SET
    DELAY 250.                            ;WAIT FOR .25 SECONDS
    MOV #250.,(PC)+
    .WORD 0
    MOV L$DLY,(PC)+
    .WORD 0
    DEC -6(PC)
    BNE -.4
    DEC -22(PC)
    BNE -.20
    DEC R3                                ;BUMP COUNTER DOWN
    BNE 10$                               ;KEEP GOING
    CLC                                  ;CLEAR CARRY TO SET ERROR
20$: MOV R4,R0                            ;PASS THE PACKET ADDRESS
    RTS PC                                ;RETURN

RWPACK: .BLKB 10-<.-TSV2&7>
    .WORD 102010                          ;POSTION COMMAND (REWIND)
    .WORD 0                               ;NOT USED

.SBTTL CKRAM - COMPARE RAM TO I/O PACKET

```

```

1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239 011014
1240 011014
1241 011020 012701 002232'
1242 011024 012702 000201
1243 011030 005003
1244 011032 004737 016236'
1245 011036 112765 000000 000000
1246 011044 004737 016236' 10$:
1247 011050 010265 000000
1248 011054 004737 016236'
1249 011060 116511 000000
1250 011064 122124
1251 011066 001401
1252 011070 005203
1253 011072 005202 20$:
1254 011074 020227 000210
1255 011100 003761
1256 011102 005703
1257 011104 001402
1258 011106 000241
1259 011110 000401
1260 011112 000261 30$:
1261 011114 012737 000010 002272' 50$:
1262 011122 000207
1263
1264
1265
1266
1267

; *
;
; 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
MOVB #0,TSDB(R5) ;SET MAINTENANCE MODE
10$: JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
MOV R2,TSDB(R5) ;SELECT NEXT RAM ADDRESS
JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
MOVB TSBA(R5),(R1) ;READ THE RAM DATA
CMPB (R1),.(R4) ;COMPARE TO EXPECTED
BEQ 20$ ;BRANCH IF OK
INC R3 ;SET ERROR FLAG
20$: INC R2 ;ADDRESS OF NEXT RAM LOCATION
CMP R2,@RMPKTEND ;REACHED END YET ?
BLE 10$ ;BRANCH TILL ALL READ
TST R3 ;WAS AN ERROR FOUND ?
BEQ 30$ ;BRANCH IF NOT
CLC ;CLEAR CARRY TO SHOW ERROR
BR 50$ ;AND EXIT
30$: SEC ;SHOW GOOD COMPARE
50$: MOV #8,,RAMSIZ ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
RTS PC ;RETURN

.SBTTL CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA
; *
;

```



```

1268 ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
1269 ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
1270 ;
1271 ;INPUT:
1272 ;
1273 ;       R4      ADDRESS OF THE CHARACTERISTICS DATA
1274 ;       R5      FIRST DEVICE UNIBUS ADDRESS
1275 ;
1276 ;OUTPUT:
1277 ;
1278 ;       CARRY   SET - RAM MATCHES PACKET
1279 ;             CLR - RAM DOES NOT MATCH PACKET
1280 ;
1281 ;IMPLICIT OUTPUT:
1282 ;
1283 ;       THE TABLE RAMDATA IS FILLED WITH THE
1284 ;       DATA HELD IN RAM.
1285 ;       RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
1286 ;
1287 ;SIDE EFFECTS:
1288 ;
1289 ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1290 ;
1291 ;-
1292 ;

```

```

1293 011124 CKRAM2:: SAVREG ;SAVE THE GENERAL REGISTERS
1294 011124 MOV #RAMDATA,R1 ;ADDRESS TO SAVE THE RAM DATA
1295 011130 012701 002232' MOV #RMCHBEG,R2 ;BYTE ADDRESS OF FIRST RAM DATA
1296 011134 012702 000167 CLR R3 ;CLEAR THE ERROR FLAG
1297 011140 005003 JSR PC,CHKTSSR ;WAIT FOR SSR
1298 011142 004737 016236' MOVB #0,TSDB(R5) ;SET MAINTENANCE MODE
1299 011146 112765 000000 000000 10$: JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
1300 011154 004737 016236' MOV R2,TSDB(R5) ;SELECT NEXT RAM ADDRESS
1301 011160 010265 000000 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
1302 011164 004737 016236' MOVB TSBA(R5),(R1) ;READ THE RAM DATA
1303 011170 116511 000000 CMPB (R1)+,(R4)+ ;COMPARE TO EXPECTED
1304 011174 122124 BEQ 20$ ;BRANCH IF OK
1305 011176 001401 INC R3 ;SET ERROR FLAG
1306 011200 005203 INC R2 ;ADDRESS OF NEXT RAM LOCATION
1307 011202 005202 20$: MOV #8.,RAMSIZ ;ASSUME EXTFEA NOT SET
1308 011204 012737 000010 002272' TST EXTFEA ;IS THE SOFTWARE EXTENDED FEATURES SET
1309 011212 005737 002216' BEQ 25$ ;BR, IF NOT SET
1310 011216 001407 000012 002272' MOV #10.,RAMSIZ ;SET RAMSIZ FOR EXTEND FEATURES
1311 011220 012737 000012 002272' CMP R2,#RMCHEND ;AT END OF EXTENDED BUFFER
1312 011226 020227 000200 BLE 10$ ;BR, IF NOT AT END YET
1313 011232 003750 BR 27$ ;AT END BRANCH
1314 011234 000403 25$: CMP R2,#RMCHEND-2 ;REACHED END YET ?
1315 011236 020227 000176 BLE 10$ ;BRANCH TILL ALL READ
1316 011242 003744 27$: TST R3 ;WAS AN ERROR FOUND ?
1317 011244 005703 BEQ 30$ ;BRANCH IF NOT
1318 011246 001402 CLC ;CLEAR CARRY TO SHOW ERROR
1319 011250 000241 BR 50$ ;AND EXIT
1320 011252 000401 30$: SEC ;SHOW GOOD COMPARE
1321 011254 000261 50$: RTS PC ;RETURN
1322 011256 000207
1323
1324

```

```

1325 .SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS
1326
1327
1328 ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
1329 ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1330 ;ERROR PRINT ROUTINES.
1331
1332 ;INPUT:
1333
1334 ; R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1335 ; R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
1336 ; R2 EXPD MESSAGE BUFFER ADDRESS
1337 ;OUTPUT:
1338
1339 ; CARRY SET - MESSAGE BUFFERS MATCH
1340 ; CLR -MESSAGE BUFFERS DON'T MATCH
1341
1342 ;IMPLICIT OUTPUT:
1343
1344 ; EXPMSG BUFFER IS SET TO EXPD DATA
1345 ; RECMMSG BUFFER IS SET TO RECV DATA
1346 ; RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1347 ; RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
1348
1349 ;-
1350 CKMSG::
1351 SAVREG ;SAVE R1-P5 UNTIL NEXT RETURN
1352 MOV RO,RCVHIADD ;SAVE RECV HIGH ADDRESS
1353 MOV R1,RCVLOADD ;SAVE RECV LOW ADDRESS
1354 TST KTENABLE ;TESTING ABOVE 28K?
1355 BEQ 10$ ;BR IF NO
1356 JSR PC,SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN RO
1357 MOV RO,R1 ;GET RETURNED ADDRESS BIASED TO PAR6
1358 10$: CLR R4 ;WORD IN BUFFER
1359 CLR R3 ;CLEAR ERROR SEEN FLAG
1360 MOV R2,R5 ;GET EXPD BUFFER ADDRESS
1361 15$: MOV (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1362 MOV (R1),RCMSG(R4) ;SAVE RECV FOR ERROR REPORT
1363 CMP (R2)+,(R1)+ ;EXPD EQUAL RECV?
1364 BEQ 25$ ;BR IF YES
1365 INC R3 ;SET ERROR SEEN FLAG
1366 25$: ADD #2,R4 ;POINT TO NEXT WORD ADDRESS
1367 CMP R4,#14 ;DONE FIRST 7 WORDS?
1368 BLE 15$ ;BR IF NO
1369 000200 000012 BIT #X2.EXTF,XST2(R5); IS EXTENDED FEATURES SET IN EXPD?
1370 BEQ 50$ ;BR IF NO
1371 000016 CMP R4,#16 ;DONE EXTENDED FEATURES WORD?
1372 BLE 15$ ;BR IF NO
1373 50$: TST R3 ;ANY ERRORS SEEN?
1374 BEQ 55$ ;BR IF NO
1375 CLC ;SET FAILURE
1376 BR 60$ ;
1377 55$: SEC ;SET SUCCESS
1378 60$: RTS PC ;RETURN
1379
1380 .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
1381

```

```

1382
1383
1384 ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
1385 ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1386 ;ERROR PRINT ROUTINES.
1387
1388 ;INPUT:
1389
1390 ;           R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1391 ;           R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1392 ;           R2      EXPD MESSAGE BUFFER ADDRESS
1393 ;           R3      NUMBER OF BYTES TO COMPARE
1394
1395 ;OUTPUT:
1396
1397 ;           CARRY   SET - MESSAGE BUFFERS MATCH
1398 ;                   CLR - MESSAGE BUFFERS DON'T MATCH
1399
1400 ;IMPLICIT OUTPUT:
1401
1402 ;           EXPMSG   BUFFER IS SET TO EXPD DATA
1403 ;           RECVMSG  BUFFER IS SET TO RECV DATA
1404 ;           RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1405 ;           RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
1406
1407 ;-
1408 CKMSG2::
1409     SAVREG                ;SAVE R1-R5 UNTIL NEXT RETURN
1410     CMP R3,#RECVMSG-EXPMSG ;@@D IS COUNT ABOVE MAX ALLOWED?
1411     BLE 5$                ;@@D BR IF NO
1412     MOV #RECVMSG-EXPMSG,R3 ;@@D
1413     PRINTF #DEBUGMSG      ;@@D
1414     MOV #DEBUGMSG,-(SP)
1415     MOV #1,-(SP)
1416     MOV SP,R0
1417     TRAP C$PRINTF
1418     ADD #4,SP
1419     MOV R0,RCVHIADD      ;SAVE RECV HIGH ADDRESS
1420     MOV R1,RCVLOADD     ;SAVE RECV LOW ADDRESS
1421     TST KTENABLE        ;TESTING ABOVE 28K?
1422     BEQ 10$             ;BR IF NO
1423     JSR PC,SETMAP       ;RETURN ADDRESS BIASED TO PAR6 IN R0
1424     MOV R0,R1           ;GET RETURNED ADDRESS BIASED TO PAR6
1425     CLR R4              ;WORD IN BUFFER
1426     CLR R5              ;CLEAR ERROR SEEN FLAG
1427     MOVB (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1428     MOVB (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
1429     CMPB (R2)+,(R1)+    ;EXPD EQUAL RECV?
1430     BEQ 25$            ;BR IF YES
1431     INC R5              ;SET ERROR SEEN FLAG
1432     ADD #1,R4           ;POINT TO NEXT BYTE
1433     CMP R4,R3           ;DONE ALL BYTES?
1434     BGE 50$            ;BR IF YES
1435     BR 15$             ;DO NEXT BYTE
1436     TST R5              ;ANY ERRORS SEEN?
1437     BEQ 55$            ;BR IF NO
1438     CLC                ;SET FAILURE

```

```

1434 011524 000401          BR      60$          ;
1435 011526 000261          55$:   SEC          ;SET SUCCESS
1436 011530 000207          60$:   RTS      PC          ;RETURN
1437
1438 011532      120      122      117  DEBUGMSG:   .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-' ;@@D
1439 011622      045      116      045  FERCM:   .ASCII /%N%A ***/
1440 011633      040      040      124  ERCM:   .ASCIZ / TSSR ERROR CODE REC'D = /
1441 011666      056      056      056  SIMSG:   .ASCIZ /... AFTER DOING SOFT INIT/
1442 011721      124      105      123  TINERR: .ASCIZ /TEST: .../
1443          .EVEN
1444
1445
1446
1447
1448          ;*
1449          ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
1450          ;
1451          ;INPUT:
1452          ;
1453          ;      R1      CONTENTS OF TSSR AT ERROR
1454          ;
1455          ;SIDE EFFECTS:
1456          ;
1457          ;      EXECUTES DROP UNIT TO CEASE TESTING
1458          ;
1459          ;-
1460 011734          BGNMSG  SFIMSG
1461 011734          SFIMSG:: JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1462 011740 004737 005632' JSR      PC,CKDROP      ;DROP UNIT, IF ALLOWED
1463 011744          ENDMSG
1464 011744 104423          L10003: TRAP      C$MSG
1465
1466          ;*
1467          ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1468          ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
1469          ;
1470          ;INPUTS:
1471          ;
1472          ;      R1      TSSR CONTENTS
1473          ;      R4      ADDRESS OF COMMAND PACKET
1474          ;
1475          ;-
1476 011746          BGNMSG  PKTSSR
1477 011746          PKTSSR:: JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1478 011752 004737 005632' MOV      #4,R0          ;NO. OF WORDS IN PACKET
1479 011756 004737 007270' JSR      PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1480 011762          ENDMSG
1481 011762          L10004: TRAP      C$MSG
1482 011762 104423
1483
1484          ;*
1485          ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1486          ;TSSR AND A GET STATUS COMMAND PACKET.

```

```

1485
1486
1487
1488
1489
1490
1491
1492
1493 011764
      011764
1494 011764 004737 005632'
1495 011770 012700 000002
1496 011774 004737 007270'
1497 012000
      012000
      012000 104423
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509 012002
      012002
1510 012002 004737 005632'
1511 012006
      012006
      012006 104423
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527 012010
      012010
1528 012010 004737 005632'
1529 012014 010200
1530 012016 010301
1531 012020 004737 014142'
1532 012024
      012024
      012024 104423

;
;INPUTS:
;
; R1 TSSR CONTENTS
; R4 ADDRESS OF COMMAND PACKET
;
;-

      BGNMSG PKTGETS
PKTGETS:
  JSR PC,PRITSSR ;PRINT THE CONTENTS OF TSSR REGISTER
  MOV #2,R0 ;NO. OF WORDS IN GET STATUS PACKET
  JSR PC,PRIPKT ;PRINT THE CONTENTS OF COMMAND PACKET
  ENDMSG
L10005:
  TRAP C$MSG

;+
;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
;
;INPUTS:
;
; R1 TSSR CONTENTS
; R4 ADDRESS OF COMMAND PACKET
;
;-

      BGNMSG SFFMSG
SFFMSG:
  JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR REGISTER
  ENDMSG
L10006:
  TRAP C$MSG

      .SBTTL PKTMES - PRINT TSSR AND MESSAGE BUFFER
;+
;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
;BUFFER FOR ERROR REPORTS
;
;INPUTS:
;
; R1 CONTENTS OF TSSR
; R2 LOW ORDER MESSAGE BUFFER
; R3 HIGH ORDER MESSAGE BUFFER ADDRESS
; NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
;
;-

      BGNMSG PKTMES
PKTMES:
  JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR
  MOV R2,R0 ;LOW ORDER ADDRESS
  MOV R3,R1 ;HIGH ORDER ADDRESS
  JSR PC,PRMESS ;PRINT THE MESSAGE BUFFER
  ENDMSG
L10007:
  TRAP C$MSG

```

```

1533
1534
1535          .SBTTL  ADDSSR - PRINT TEST ADDRESS AND TSSR
1536          ;+
1537          ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1538          ;TSSR AND A MEMORY TEST ADDRESS
1539          ;
1540          ;INPUTS:
1541          ;
1542          ;      R5      FIRST DEVICE UNIBUS ADDRESS
1543          ;      ERRHI   HIGH ORDER MEMORY TEST ADDRESS
1544          ;      ERRLO   LOW ORDER MEMORY TEST ADDRESS
1545          ;-
1546
1547 012026          BGNMSG  ADDSSR
1548 012026          ADDSSR::
1549 012026 004737 010174'      JSR      PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
1550 012032 016501 000002      MOV      TSSR(R5),R1     ;GET CURRENT TSSR
1551 012036 004737 005632'      JSR      PC,PRITSSR     ;PRINT THE CONTENTS OF TSSR REGISTER
1552 012042          ENDMSG
1553 012042 104423      L10010:
1554          TRAP      C$MSG
1555
1556          .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RECV MESSAGE BUFFERS
1557          ;+
1558          ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
1559          ;
1560          ;IMPLICIT INPUTS:
1561          ;
1562          ;      EXPMSG - EXPECTED MESSAGE BUFFER
1563          ;      RECMG  - RECEIVED MESSAGE BUFFER
1564          ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1565          ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1566          ;-
1567 012044          BGNMSG  MSGEXP
1568 012044          MSGEXP::
1569 012044 012700 000007      MOV      #7,R0          ;ASSUME NO EXT FEATURES
1570 012050 005737 002216'      TST      EXTFEA        ;EXT FEATURES SET?
1571 012054 001402          BEQ      5$                ;BR IF NO
1572 012056 012700 000010      MOV      #8.,R0        ;EXT FEATURE BUFFER IS 8 WORDS
1573 012062 004737 014452'      JSR      PC,PRMSGEXP   ;PRINT EXPD/RECV MESSAGE BUFFERS
1574 012066          ENDMSG
1575 012066 104423      L10011:
1576          TRAP      C$MSG
1577
1578          .SBTTL  FIFEXP - PRINT FIFO EXP/RECV DATA
1579          ;+
1580          ;PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
1581          ;
1582          ;      R1      - BYTE COUNT
1583          ;
1584          ;IMPLICIT INPUTS:
1585          ;
1586          ;      EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY

```

```

1584
1585
1586 012070
      012070
1587 012070
      012070 010146
      012072 012746 012142'
      012076 012746 000002
      012102 010600
      012104 104415
      012106 062706 000006
1588 012112
      012112 012746 012211'
      012116 012746 000001
      012122 010600
      012124 104415
      012126 062706 000004
1589 012132 010100
1590 012134 004737 015022'
1591 012140
      012140
      012140 104423
1592 012142 045 116 045 FIF1MSG:
1593 012211 045 116 045 FIF2MSG:
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609 012250
      012250
1610 012250 012701 012312'
1611 012254 012100
1612 012256 001410
1613 012260
      012260 010046
      012262 012746 000001
      012266 010600
      012270 104415
      012272 062706 000004
1614 012276 000766
1615 012300 012700 000012
1616 012304 004737 014452'
1617 012310
      012310
      012310 104423
1618

```

```

; RECMSG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
;
; BGNMSG FIFEXP
FIFEXP::
PRINTX @FIF1MSG,R1 ;PRINT BYTES TRANSFERRED
MOV R1,-(SP)
MOV @FIF1MSG,-(SP)
MOV @2,-(SP)
MOV SP,R0
TRAP C:PNTX
ADD @6,SP
PRINTX @FIF2MSG ;PRINT HEADER MSG
MOV @FIF2MSG,-(SP)
MOV @1,-(SP)
MOV SP,R0
TRAP C:PNTX
ADD @4,SP
MOV R1,R0 ;GET BYTE COUNT
JSR PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
ENDMSG

L10012:
TRAP C:MSG
;ASCIZ 'N#A NUMBER OF BYTES TRANSFERRED = #D2'
;ASCIZ 'N#A FIFO DATA BYTES IN ERROR:'
.EVEN

.SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
;
;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
;
;IMPLICIT INPUTS:
;
; EXPMSG - EXPECTED MESSAGE BUFFER
; RECMSG - RECEIVED MESSAGE BUFFER
; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
;
; BGNMSG MSGSTAT
MSGSTAT::
MOV @STATCOD,R1 ;ASCII ADDRESS TABLE
10$: MOV (R1),R0 ;DONE ALL MSG LINES?
BEQ 20$ ;BR IF YES
PRINTX R0 ;PRINT STATUS BIT NAMES
MOV R0,-(SP)
MOV @1,-(SP)
MOV SP,R0
TRAP C:PNTX
ADD @4,SP
BR 10$ ;DO ANOTHER MSG LINE
20$: MOV @10,R0 ;NUMBER OF WORDS IN A READ STATUS BUFFER
JSR PC,PRMSGEXP ;PRINT EXPD/RECV MESSAGE BUFFERS
ENDMSG

L10013:
TRAP C:MSG

```

1619	012312	012330'	012372'	012463'	STATCOD:	.WORD	1\$,2\$,3\$,4\$,5\$,6\$,0
1620	012330	045	116	045	1\$:.ASCIZ	'\$N\$A	Tape Bus Signals in Word #8:'
1621	012372	045	116	045	2\$:.ASCIZ	'\$N\$A	PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
1622	012463	045	116	045	3\$:.ASCIZ	'\$N\$A	IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
1623	012554	045	116	045	4\$:.ASCIZ	'\$N\$A	IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
1624	012645	045	116	045	5\$:.ASCIZ	'\$N\$A	Tape Bus Signals in Word #9:'
1625	012707	045	116	045	6\$:.ASCIZ	'\$N\$A	DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'

1626

1627

1628

1629

1630

1631

1632

1633

1634

1635

1636

1637

1638

1639

1640

1641

1642

1643

1644

1645

1646

1647

1648

1649

1650

1651

1652

1653

1654

1655

1656

1657

1658

1659

1660

1661

1662

1663

1664

1665

1666

1667

.SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

```

;
;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
;
;IMPLICIT INPUTS:
;
;   EXPMSG - EXPECTED MESSAGE BUFFER
;   RECMSG - RECEIVED MESSAGE BUFFER
;   RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
;   RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
;
;
;BGNMSG MSGLOOP
MSGLOOP::
10$: MOV     @LOOPCOD,R1      ;ASCII ADDRESS TABLE
      MOV     (R1)+,R0      ;DONE ALL MSG LINES?
      BEQ     20$           ;BR IF YES
      PRINTX R0             ;PRINT STATUS BIT NAMES
      MOV     R0,-(SP)
      MOV     @1,-(SP)
      MOV     SP,R0
      TRAP   C$PNTX
      ADD     @4,SP
      BR     10$           ;DO ANOTHER MSG LINE
20$: MOV     @10,,R0       ;NUMBER OF WORDS IN A READ STATUS BUFFER
      JSR    PC,PRMSGEXP   ;PRINT EXPD/RCV MESSAGE BUFFERS
      ENDMSG
L10014: TRAP   C$MSG

```

1652	013026	013046'	013121'	013220'	LOOPCOD:	.WORD	1\$,2\$,3\$,4\$,5\$,6\$,7\$,0
1653	013046	045	116	045	1\$:.ASCIZ	'\$N\$A	Tape Bus Loopback Signals in Word #8:'
1654	013121	045	116	045	2\$:.ASCIZ	'\$N\$A	PARERR<15> IRESV2<14> IRESV1<13>'
1655	013220	045	116	045	3\$:.ASCIZ	'\$N\$A	IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
1656	013317	045	116	045	4\$:.ASCIZ	'\$N\$A	IWFM =>IFMK<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
1657	013416	045	116	045	5\$:.ASCIZ	'\$N\$A	ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDP <04>'
1658	013515	045	116	045	6\$:.ASCIZ	'\$N\$A	IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
1659	013614	045	116	045	7\$:.ASCIZ	'\$N\$A	IGO =>IFPT<00>'

.EVEN

.SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```

;
;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
;
;
;

```



```

1668
1669
1670
1671
1672
1673
1674
1675 013642
      013642
1676 013642 012700 000012
1677 013646 004737 014452'
1678 013652
      013652
      013652 104423
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696 013654
      013654
1697 013654 004737 010060'
1698 013660 013701 002222'
1699 013664 013702 002224'
1700 013670 004737 007642'
1701 013674
      013674
      013674 104423
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718

;IMPLICIT INPUTS:
;
;   EXPMSG - EXPECTED MESSAGE BUFFER
;   RECMSG - RECEIVED MESSAGE BUFFER
;   RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
;   RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
;
;
;   BGNMSG MSGSUB
MSGSUB::
;   MOV   #10.,R0           ;SIZE OF WRITE SUBSYSTEM BUFFER
;   JSR   PC,PRMSGEXP      ;PRINT EXPD/RCV MESSAGE BUFFERS
;   ENDMSG
L10015:
;   TRAP  C$MSG

;
;   .SBTTL MEMADD - PRINT MEMORY ADDRESS DATA ERROR
;
;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
;
;IMPLICIT INPUTS:
;
;   ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
;   ERRLO - MEMORY ERROR LOW ORDER ADDRESS
;   EXP   - EXPECTED DATA
;   RECV  - RECEIVED DATA
;
;
;   BGNMSG MEMADD
MEMADD::
;   JSR   PC,PRIADD        ;PRINT MEMORY ADDRESS IN ERROR
;   MOV   EXPD,R1          ;GET EXPD DATA
;   MOV   RECV,R2          ;GET RECEIVED DATA
;   JSR   PC,PRIXOR        ;PRINT EXPD/RCV
;   ENDMSG
L10016:
;   TRAP  C$MSG

;
;   .SBTTL PRAMPKT - PRINT RAM AND PACKET DATA
;
;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
;WHEN THE RAM DATA DOES NOT MATCH.
;
;INPUTS:
;
;   R4    POINTER TO COMMAND PACKET
;
;IMPLICIT INPUTS:
;
;   RAMDATA  DATA AS READ FROM THE RAM
;   RAMSIZ   NUMBER OF BYTES IN PACKET
;            IF RAMSIZ=0 THEN DEFAULT TO 8.

```

```

1719                                     ;IMPLICIT OUTPUTS:
1720                                     ;
1721                                     ;       RAMSIZ  SET TO 0
1722                                     ;
1723
1724 013676
1725 013676
1726 013702 012701 002232'
1727 013706 005002
1728 013710 122124
1729 013712 001005
1730 013714
1731 013724 000436
1732 013726 116105 177777
1733 013732 116403 177777
1734 013736
1735 013746 042703 177400
1736 013752 116137 177777 002224'
1737 013760 116437 177777 002222'
1738 013766
      013766 010346
      013770 013746 002222'
      013774 013746 002224'
      014000 010246
      014002 012746 014056'
      014006 012746 000005
      014012 010600
      014014 104414
      014016 062706 000014
1739 014022 005202
1740 014024 005737 002272'
1741 014030 001404
1742 014032 020237 002272'
1743 014036 003724
1744 014040 000403
1745 014042 020227 000010
1746 014046 002720
1747 014050 005037 002272'
1748 014054 000207
1749
1750 014056      045      116      045  RAMASC: .ASCIZ 'N#A BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03'
1751                                     .EVEN
1752
1753                                     .SBTTL  PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
1754
1755                                     ;*
1756                                     ;
1757                                     ;THIS ROUTINE PRINTS THE CONTENTS OF
1758                                     ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE
1759                                     ;TSV-05.
1760                                     ;
1761                                     ;INPUT:
1762                                     ;
1763                                     ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
1764                                     ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
1765                                     ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
1766                                     ;
1767                                     ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE

```

```

1767      ;
1768      ; -
1769
1770 014142      PRMESS:
1771 014142      SAVREG
1772 014146 010005      MOV R0,R5 ;SAVE THE REGISTERS
1773 014150 005737 003124' TST KTENABLE ;SAVE LOW ORDER ADDRESS
1774 014154 001001      BNE 10$ ;ADDRESS ABOVE 28K?
1775 014156 005001      CLR R1 ;BR IF YES
1776 014160 010103      10$: MOV R1,R3 ;SET HIGH ORDER ADDRESS TO 0
1777 014162 006100      ROL R0 ;SAVE HIGH ORDER ADDRESS
1778 014164 006101      ROL R1 ;SHIFT BIT15 TO C BIT
1779 014166      PRINTX @PROASC,R1,R5 ;SHIFT TO HIGH ORDER FOR PRINTOUT
      014166 010546      MOV R5,-(SP) ;PRINT MESSAGE BUFFER ADDRESS
      014170 010146      MOV R1,-(SP)
      014172 012746 014320' MOV @PROASC,-(SP)
      014176 012746 000003 MOV @3,-(SP)
      014202 010600      MOV SP,R0
      014204 104415      TRAP C$PNTX
      014206 062706 000010 ADD @10,SP
1780 014212      PRINTX @PR1ASC ;PRINT HEADER FOR CONTENTS
      014212 012746 014365' MOV @PR1ASC,-(SP)
      014216 012746 000001 MOV @1,-(SP)
      014222 010600      MOV SP,R0
      014224 104415      TRAP C$PNTX
      014226 062706 000004 ADD @4,SP
1781 014232 005004      CLR R4 ;NUMBER OF THE NEXT WORD
1782 014234 010501      MOV R5,R1 ;COPY LOW ORDER ADDRESS
1783 014236 010300      MOV R3,R0 ;COPY HIGH ORDER ADDRESS
1784 014240 001403      BEQ 20$ ;BR IF NOT ABOVE 28K
1785 014242 004737 017220' JSR PC,SETMAP ;SETUP PAR ADDRESS IN R0
1786 014246 010005      MOV R0,R5 ;GET PAR FORMAT ADDRESS ABOVE 28K
1787 014250      20$: PRINTX @PRASC,R4,(R5) ;PRINT THE CONTENTS OF MEMORY BUFFER
      014250 012546      MOV (R5),-(SP)
      014252 010446      MOV R4,-(SP)
      014254 012746 014423' MOV @PRASC,-(SP)
      014260 012746 000003 MOV @3,-(SP)
      014264 010600      MOV SP,R0
      014266 104415      TRAP C$PNTX
      014270 062706 000010 ADD @10,SP
1788 014274 005204      INC R4 ;NUMBER OF THE NEXT
1789 014276 020427 000007 CMP R4,@7 ;DONE ALL YET ?
1790 014302 003005      BGT 50$ ;BRANCH IF ALL DONE
1791 014304 002761      BLT 20$ ;PRINT FIRST 7 WORDS
1792 014306 032763 000200 000012 BIT @X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
1793 014314 001355      BNE 20$ ;PRINT EXTENDED STATUS WORD
1794 014316 000207      50$: RTS PC ;RETURN
1795
1796 014320 045 116 045 PROASC: .ASCIZ 'N$A Message Buffer Address = 0105'
1797 014365 045 116 045 PR1ASC: .ASCIZ 'N$A Message Buffer Contents:'
1798 014423 045 116 045 PRASC: .ASCIZ 'N$A WordD1$A: 0'
1799      .EVEN
1800
1801      .SBTTL PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
1802      ;*
1803      ;
1804      ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
    
```

```

1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815 014452
1816 014452
1817 014456 010005
1818 014460 013700 002276'
1819 014464 010004
1820 014466 013701 002274'
1821 014472 006100
1822 014474 006101
1823 014476
    014476 010446
    014500 010146
    014502 012746 014632'
    014506 012746 000003
    014512 010600
    014514 104415
    014516 062706 000010
1824 014522
    014522 012746 014677'
    014526 012746 000001
    014532 010600
    014534 104415
    014536 062706 000004
1825 014542 005004
1826 014544 012701 002312'
1827 014550 012702 002456'
1828 014554 011100
1829 014556 011203
1830 014560
1831 014570
    014570 010346
    014572 012246
    014574 012146
    014576 010446
    014600 012746 014735'
    014604 012746 000005
    014610 010600
    014612 104415
    014614 062706 000014
1832 014620 005204
1833 014622 020405
1834 014624 002001
1835 014626 000752
1836 014630 000207
1837
1838 014632 045 116 045 PRMSG0: .ASCIZ 'N#A Message Buffer Address = #01#05'
1839 014677 045 116 045 PRMSG1: .ASCIZ 'N#A Message Buffer Contents:'
1840 014735 045 116 045 PRMSG2: .ASCIZ 'N#A WORD #D2#A EXPD: #06#A RECV: #06#A XOR: #06#A

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


TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:54
 PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

SEQ 060

```

015224 010600          MOV    SP,R0
015226 104415          TRAP   C$PNTX
015230 062706 000006   ADD    #6,SP
1886 015234 000207    RTS    PC                ;RETURN
1887
1888 015236    045    116    045  PRBMSG: .ASCIZ 'N#A  BYTE #D2#A  EXPD: #03#A  RECV: #03#A  XOR: #03'
1889 015323    045    116    045  PRBTOT: .ASCIZ 'N#A  NUMBER OF BYTES IN ERROR = #D2'
1890
1891 015370 000000          PRBEXP: .WORD 0                ;EXPD
1892 015372 000000          PRBREC: .WORD 0                ;RECV
1893
1894                      .SBTTL  EXPREC  - PRINT EXPD/RECV WORD DATA
1895
1896                      ;*
1897                      ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1898                      ;
1899                      ;INPUTS:
1900                      ;
1901                      ;    R1    RECEIVED DATA
1902                      ;    R2    EXPECTED DATA
1903                      ;
1904                      ;-
1905
1906 015374          BGNMSG  EXPREC
015374          EXPREC::
1907 015374 004737 007642'   JSR    PC,PRIXOR          ;PRINT THE DATA
1908 015400          ENDMSG
015400          L10017:
015400 104423          TRAP    C$MSG
1909
1910
1911
1912
1913                      .SBTTL  EXPBREC  - PRINT EXPD/RECV BYTE DATA
1914                      ;*
1915                      ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
1916                      ;
1917                      ;INPUTS:
1918                      ;
1919                      ;    R1    RECEIVED DATA BYTE
1920                      ;    R2    EXPECTED DATA BYTE
1921                      ;
1922                      ;-
1923
1924
1925
1926 015402          BGNMSG  EXPBREC
015402          EXPBREC::
1927 015402 004737 007512'   JSR    PC,PRIBXOR        ;PRINT THE DATA
1928 015406          ENDMSG
015406          L10020:
015406 104423          TRAP    C$MSG
1929
1930
1931
1932
1933                      .SBTTL  RAMERR  - PRINT RAM AND PACKET DATA

```

1934				;	+	
1935				;	PRINT ROUTINE TO DISPLAY RAM/PACKET DATA	
1936				;	INPUTS:	
1937				;		
1938				;	R4 POINTER TO COMMAND PACKET	
1939				;	IMPLICIT INPUTS:	
1940				;		
1941				;	RAMDATA DATA AS READ FROM THE RAM	
1942				;	RAMSIZ NUMBER OF BYTES IN PACKET	
1943				;		IF RAMSIZ=0 THEN DEFAULT TO 8.
1944				;	IMPLICIT OUTPUTS:	
1945				;		
1946				;	RAMSIZ SET TO 0	
1947				;		
1948				;		
1949				;		
1950				;		
1951				;		
1952				;		
1953	015410				BGNMSG RAMERR	
	015410			RAMERR::		
1954	015410	004737	013676'		JSR PC,PRAMPKT ;PRINT RAM/PACKET DATA	
1955	015414				ENDMSG	
	015414			L10021:		
	015414	104423			TRAP C\$MSG	
1956						
1957						
1958					.SBTTL RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA	
1959				;	+	
1960				;	PRINT ROUTINE TO DISPLAY RAM/PACKET DATA	
1961				;	INPUTS:	
1962				;		
1963				;	R4 POINTER TO COMMAND PACKET	
1964				;	IMPLICIT INPUTS:	
1965				;		
1966				;	RAMDATA DATA AS READ FROM THE RAM	
1967				;	RAMSIZ NUMBER OF BYTES IN PACKET	
1968				;		IF RAMSIZ=0 THEN DEFAULT TO 8.
1969				;	ERRHI HIGH ORDER TEST ADDRESS	
1970				;	ERRLO LOW ORDER TEST ADDRESS	
1971				;	IMPLICIT OUTPUTS:	
1972				;		
1973				;	RAMSIZ SET TO 0	
1974				;		
1975				;		
1976				;		
1977				;		
1978				;		
1979				;		
1980	015416				BGNMSG RAMTADD	
	015416			RAMTADD::		
1981	015416	004737	010174'		JSR PC,PRITADD ;PRINT TEST ADDRESS	
1982	015422	004737	013676'		JSR PC,PRAMPKT ;PRINT RAM/PACKET DATA	
1983	015426				ENDMSG	
	015426			L10022:		
	015426	104423			TRAP C\$MSG	
1984						


```

2031      ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
2032      ;
2033      ;INPUTS:
2034      ;
2035      ;       R1      CONTENTS OF TSSR
2036      ;       R2      DATA WRITTEN (8 BITS)
2037      ;
2038      ;-
2039
2040 015600      BGNMSG  BADSSR
015600      BADSSR::
2041 015600 010246      MOV      R2,-(SP)          ;SAVE DATA TRANSFERRED
2042 015602 042702 177400      BIC      #177400,R2          ;GET JUST ONE BYTE
2043 015606      PRINTB  #XFERASC,R2
015606 010246      MOV      R2,-(SP)
015610 012746 015640'      MOV      #XFERASC,-(SP)
015614 012746 000002      MOV      #2,-(SP)
015620 010600      MOV      SP,R0
015622 104414      TRAP    C#PNTB
015624 062706 000006      ADD      #6,SP
2044 015630 012602      MOV      (SP)+,R2          ;RESTORE R2
2045 015632 004737 005632'      JSR      PC,PRITSSR          ;DECODE TSSR CONTENTS
2046 015636      ENDMMSG
015636      L10025:
015636 104423      TRAP    C#MSG
2047 015640 045 116 045 XFERASC: .ASCIZ  '#N#A Data Transferred = #03'
2048
2049
2050      .SBTTL  GLOBAL SUBROUTINES SECTION
2051
2052      ;**
2053      ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2054      ; THAT ARE USED IN MORE THAN ONE TEST.
2055      ;--
2056
2057      .SBTTL  SOFINIT - SOFT INITIALIZE OF CONTROLLER
2058
2059      ;+
2060      ;
2061      ;ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
2062      ;BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
2063      ;THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
2064      ;DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
2065      ;
2066      ;INPUTS:
2067      ;
2068      ;       R5      ADDRESS OF FIRST REGISTER
2069      ;
2070      ;OUTPUTS:
2071      ;
2072      ;       R0      CONTENTS OF TSSR, IF ERROR
2073      ;       CARRY   SET IF INIT WAS OKAY
2074      ;               CLEAR IF FATAL ERROR
2075      ;
2076      ;CALLING SEQUENCE:
2077      ;
2078      ;       MOV      #ADDRESS,R5

```

```

2079      ;      JSR      PC,SOFINIT
2080      ;      BCS      CONTINUE
2081      ;      ERRDF          ;REPORT FATAL ERROR
2082      ;
2083      ;-
2084
2085 015674      SOFINIT::
2086 015674      SAVREG          ; SAVE THE REGISTERS
2087 015700 012765 000000 000002      MOV      #0,TSSR(R5)      ; DO THE INIT.
2088 015706 004737 016150'      JSR      PC,WAITF          ; WAIT FOR SSR
2089 015712 016500 000002      MOV      TSSR(R5),R0      ;GET THE TSSR REGISTER
2090 015716 010004      MOV      R0,R4          ;TSSR CONTENTS
2091 015720 042704 176277      BIC      #+C<HIADDR!OFL>,R4
2092 015724 052704 002200      BIS      #SSR!NBA,R4      ;R4 HAS EXPECTED CONTENTS
2093 015730 020400      CMP      R4,R0          ;ONLY EXPECTED BITS SET ?
2094 015732 001402      BEQ      5$          ;BRANCH IF OKAY
2095 015734 000241      CLC          ;CLEAR THE CARRY FOR ERROR
2096 015736 000401      BR      10$          ;GO TO EXIT
2097 015740 000261      5$: SEC          ;SET THE CARRY BIT
2098 015742 000207      10$: RTS      PC      ;RETURN TO CALLER
2099
2100      .SBTTL  CHKAMB - CHECK TSSR FOR AMBIGUITY
2101
2102      ;+
2103      ;
2104      ; THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
2105      ; FOR AMBIGUITY
2106      ;
2107      ; INPUT:
2108      ;
2109      ;      R0      CONTENTS OF TSSR
2110      ;
2111      ; OUTPUT:
2112      ;
2113      ;      R0      CONTENTS OF TSSR
2114      ;
2115      ;      CARRY  SET - NO AMBIGUITY
2116      ;              CLR - AMBIGUOUS CONTENTS
2117      ;
2118      ;-
2119
2120 015744      CHKAMB:
2121 015744      SAVREG          ;SAVE THE GENERAL REGISTERS
2122 015750 010004      MOV      R0,R4          ;CONTENTS OF TSSR
2123 015752 032700 100000      BIT      #SC,R0          ;IS BIT 15 SET ?
2124 015756 001004      BNE      5$          ;BRANCH IF YES
2125 015760 032700 174077      BIT      #+C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
2126 015764 001023      BNE      40$          ;MUST BE AN ERROR
2127 015766 000424      BR      45$          ;RETURN WITH SUCCESS
2128 015770 032700 000200      5$: BIT      #SSR,R0      ;IS READY BIT SET ?
2129 015774 001011      BNE      10$          ;BRANCH IF READY BIT IS SET.
2130 015776 032700 000040      BIT      #BITS,R0      ;IS FATAL ERROR BIT SET ?
2131 016002 001414      BEQ      40$          ;ERROR IF NOT
2132 016004 042704 177761      BIC      #+CTERCLS,R4      ;CLEAR ALL BUT TERMINATION CODE
2133 016010 020427 000016      CMP      R4,#16          ;ALL THREE BITS MUST BE SET
2134 016014 001007      BNE      40$          ;ERROR IF NOT SET
2135 016016 000410      BR      45$          ;OK IF ALL ARE SET

```

```

2136 016020 032700 000040      10$: BIT    #BIT5,RO      ;IS FATAL ERROR BIT SET ?
2137 016024 001405              BEQ     45$             ;ERROR IF BIT IS SET WITH SSR
2138 016026 032700 000006      BIT    #BIT2!BIT1,RO    ;IS THIS A FUNCTION REJECT
2139 016032 001002              BNE     45$             ;BR, IF TSSR IS OK
2140 016034 000241      40$: CLC                    ;AMBIGUOUS CONTENTS
2141 016036 000401              BR      50$
2142 016040 000261      45$: SEC                    ;SHOW SUCCESS - NO AMBIGUITY
2143 016042 000207      50$: RTS     PC          ;RETURN TO CALLER
2144
2145              .SBTTL ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
2146
2147              ;
2148              ; DEFAULT DISPLAY INTERRUPT HANDLERS.
2149              ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2150              ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
2151              ;
2152              ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2153              ;
2154              000200          IOKCKIN=BIT7      ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
2155              000001          IOKSTP=BIT0       ; EXPECT "STOP" INTERRUPT.
2156              ;
2157              ;INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
2158 016044          000          INTMASK: .BYTE    0
2159              ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2160 016045          000          INTFLAG: .BYTE    0
2161
2162              ;SAVED INTERRUPT VECTOR:
2163 016046          000000          INTVEC: .WORD    0
2164              ;SAVE CPU PC
2165 016050          000000          INTCPC: .WORD    0
2166
2167              ;SUBROUTINE TO ENABLE INTERRUPTS:
2168 016052          010046          ENAINT: MOV     RO,-(SP)      ;SAVE RO
2169 016054          013700          002200'        MOV     IVEC,RO      ;GET POINTER TO VECTORS
2170 016060          012720          016116'        MOV     #INTR,(RO)+ ;SET UP INTERRUPT VECTOR
2171 016064          012720          000340          MOV     #PRI07,(RO)+
2172 016070          012600          MOV     (SP)+,RO      ;RESTORE RO
2173 016072          011646          MOV     (SP),-(SP)
2174 016074          012766          000000 000002  MOV     #0,2(SP)    ;SET CPU TO LEVEL 0
2175 016102          000002          RTI
2176
2177              ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2178 016104          011646          DSBINT: MOV     (SP),-(SP)
2179 016106          012766          000340 000002  MOV     #PRI07,2(SP)
2180 016114          000002          RTI
2181
2182              .SBTTL INTR - INTERRUPT HANDLERS
2183
2184 016116          016116          BGNSRV INTR      ;DEFINE INTERRUPT ENTRY
2185 016116          012737          000001 002214'  INTR:: MOV     #1,INTRECV ;SET FLAG TO SHOW INTERRUPT RECEIVED
2186 016124          105037          016045'        CLRB   INTFLAG      ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2187 016130          132737          000001 016044'  BITB   #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2188 016136          001003          BNE     1$           ;BR IF YES
2189 016140          152737          000001 016045'  BISB   #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2190
2191              ;SAVE REGISTERS, MSG BUFFER, ETC.

```

```

2192 016146
2193 016146
      016146
      016146 000002
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209 016150 000401
2210 016152
      016152 104422
2211 016154 012746 011000
2212 016160 016500 000002
2213 016164 105700
2214
2215 016166 100420
2216 016170
      016170 012727 000001
      016174 000000
      016176 013727 002116
      016202 000000
      016204 005367 177772
      016210 001375
      016212 005367 177756
      016216 001367
2217 016220 005316
2218 016222 001356
2219 016224 000241
2220 016226 000401
2221 016230 000261
2222 016232 005326
2223 016234 000207
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237

1$:      ENDSRV
L10026:    RTI
          .SBTTL WAITF    - WAIT FOR SUBSYSTEM READY
;
; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
;
; INPUTS:
;
;      R5      ADDRESS OF FIRST DEVICE REGISTER
;
; OUTPUTS:
;
;      R0      CONTENTS OF LAST TSSR READ
;      CARRY    SET - READY BIT SET
;              CLR - TIMEOUT WAITING FOR READY
;
WAITF:: BR      1$                    ;NOP WHEN SUPER FIXED
         BREAK                    ; DO A SUPVSR BREAK FIRST.
         TRAP      C$BRK
1$:      MOV      @11000,-(SP)      ;25-APRIL-83 REV B    - 1100 MSEC TIMER
2$:      MOV      TSSR(R5),R0      ;READ THE TSSR REGISTER
         TSTB      R0              ;TEST FOR READY BIT SET
         BMI      3$              ; EXIT ON STOP FLAG.
         DELAY    1              ; WAIT 100 USEC
         MOV      @1,(PC)
         .WORD    0
         MOV      L$DLY,(PC)
         .WORD    0
         DEC      -6(PC)
         BNE      .-4
         DEC      -22(PC)
         BNE      .-20
         DEC      (SP)            ;REDUCE DELAY COUNT
         BNE      2$            ;RETRY UNTIL TIMER EXPIRES
         CLC                    ; C = 0, CONTROLLER STILL RUNNING...
         BR      4$            ;...OR HUNG-UP AFTER 300 MSEC.
3$:      SEC                    ; C = 1, CONTROLLER IS STOPPED.
4$:      DEC      (SP)            ;RESTORE STACK WITHOUT CHANGING CARRY BIT
         RTS      PC
                                  .SBTTL CHKTSSR - CHECK TSSR FOR READY
;
; *
; THIS ROUTINE WAITS FOR READY IN THE TSSR
; AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
;
; INPUT:
;
;      R5      ADDRESS OF CSR REGISTERS
;
; OUTPUT:
;

```

```

2238      ;      RO      CONTENTS OF TSSR
2239      ;      CARRY   SET - OKAY
2240      ;
2241      ;
2242      ;
2243      ;
2244 016236      CHKTSSR:
2245 016236 004737 016150'      JSR      PC, WAITF      ;WAIT FOR READY
2246 016242 103014      BCC      20$      ;BRANCH IF TIME OUT
2247 016244 004737 015744'      JSR      PC, CHKAMB      ;TSSR AMBIGUOUS?
2248 016250 103006      BCC      10$      ;BR IF YES
2249 016252 032700 100000      BIT      @SC, RO      ;SPECIAL CONDITION SET?
2250 016256 001405      BEQ      15$      ;BR IF NO
2251 016260 032700 074000      BIT      @<SCE!BIE!RMR!NXM>, RO ;ANY ERROR BITS SET?
2252 016264 001402      BEQ      15$      ;BR IF NO
2253 016266 000241      10$: CLC      ;SET FAILURE
2254 016270 000401      BR      20$      ;
2255 016272 000261      15$: SEC      ;SET SUCCESS
2256 016274 000207      20$: RTS      PC      ;RETURN TO CALLER
2257
2258      .SBTTL  XNXM      - CHECK FOR NONEXISTENT MEMORY
2259
2260      ;*
2261      ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
2262      ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
2263      ; "C" = 0, ALL ADDRESSES OK.
2264
2265      ;CALL:  MOV ADR1,R1
2266      ;        MOV ADR2,R2
2267      ;        JSR PC,NXM
2268      ;        RETURN      ;TEST "C" AND PROCEED.
2269 016276 012737 016332' 000004 XNXM:  MOV      @2$, @04      ; SET BUSERR VECTOR.
2270 016304 012737 000200 000006      MOV      @PRI04, @06
2271 016312 005003      CLR      R3      ;FLAG.
2272 016314 000241      CLC      ;CLEAR THE CARRY FOR NO NXM FOUND
2273 016316 005711      1$:  TST      (R1)      ;TEST THE ADDRESS(ES).
2274      ;IF ANY TRAP, CONTINUE AT 2$.
2275 016320 020102      CMP      R1,R2      ;OTHERWISE, CONTINUE HERE.
2276 016322 001407      BEQ      3$      ;BR IF FINISHED (NO NEXM'S).
2277 016324 062701 000002      ADD      @2,R1      ;SET NEXT ADDRESS...
2278 016330 000772      BR      1$      ;...AND CONTINUE.
2279
2280 016332 005103      2$:  COM      R3      ;GOT ONE, SET FLAG...
2281 016334 012716 016342'      MCV      @3$, (SP)
2282 016340 000002      RTI      ;...AND DISMISS INTERRUPT...
2283 016342      3$:  CLRVEC  @4      ;...AND GIVE BACK THE VECTOR.
2284      016346 012700 000004      MOV      @4, RO
2285      016350 104436      TRAP    C$CVEC
2286 016352 005703      TST      R3
2287 016354 001401      BEQ      .+4      ;DID WE CATCH ONE ??
2288 016356 000261      SEC      ;NO, "C" = 0, SKIP NEXT.
2289      000207      RTS      PC      ;YES, "C" = 1, (R1) = NEXM ADDR.
2290
2291
2292      .SBTTL  TSTLOOP - CHECK ITERATION COUNT

```

```

2293
2294
2295
2296
2297
2298
2299
2300 016360
2301 016360 005737 002160'
2302 016364 001006
2303 016366 005737 002174'
2304 016372 100403
2305 016374 005337 002206'
2306 016400 001002
2307 016402 000241
2308 016404 000401
2309 016406 000261
2310 016410 000207
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338 016412
2339 016412 010046
2340 016414 005037 003144'
2341 016420 005037 016660'
2342 016424 005037 005600'
2343 016430 105037 016044'
2344 016434 013700 002172'
2345 016440 006300
2346 016442 005737 003104'
2347 016446 001430
2348 016450 100010
2349 016452 052760 160000 003166'
    
```

```

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

        .SBTTL  TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
;*
; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
; IN THE CURRENT RUN SEQUENCE.
; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
;
; INPUT:
;
;     R0     POINTER TO TEST ID ASCIZ STRING
;
; OUTPUT:
;
;     R5     ADDRESS OF FIRST DEVICE REGISTER
;
; IMPLICIT OUTPUTS:
;
;     TSTCNT UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
;
; SIDE EFFECTS:
;
;     INTERRUPT LEVEL IS RASIED TO LEVEL OF
;     THE DEVICE UNDER TEST
;
;
TSTSETUP::
    MOV     R0, -(SP)    ; SAVE THE TEST ID MESSAGE
    CLR     SIFLAG      ; CLEAR "SOFT INIT" FLAG
    CLR     ERRK        ; CLEAR LOCAL ERROR COUNTER.
    CLR     EXTA        ; CLEAR ERROR EXTENSION FLAG.
    CLR     INTMASK     ; CLEAR INTERRUPT MASK (CHECK ERROR)
    MOV     UNITN, R0   ; GET THE UNIT NUMBER.
    ASL     R0          ; ... AND MAKE IT A WORD OFFSET.
    TST     NODEV       ; DID STARTUP FIND THE DEVICE?
    BEQ     4$          ; BR IF YES
    BPL     3$          ; BR IF NOT IDLE
    BIS     @160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
    
```

```

2350 016460          ERRDF  1,NXR,NXRERR  ; NO DEVICE HERE -- PRINT IT
      016460 104455  TRAP    C#ERDF
      016462 000001  .WORD  1
      016464 003730' .WORD  NXR
      016466 005544' .WORD  NXRERR
2351 016470 000407  BR      2#
2352 016472 052760 160001 003166' 3# : BIS    @160001,ERTABL(RO)  ; FLAG ERROR IN THE ERROR TABLE
2353 016500          ERRDF  2,NOINIT  ; DEVICE NOT IDLE
      016500 104455  TRAP    C#ERDF
      016502 000002  .WORD  2
      016504 004325' .WORD  NOINIT
      016506 000000  .WORD  0
2354 016510 012737 177777 003102' 2# : MOV    @-1,DUFLG  ; DROP THE UNIT
2355 016516          DODU    UNITN
      016516 013700 002172' MOV    UNITN,RO
      016522 104451  TRAP    C#DODU
2356 016524          DOCLN
      016524 104444  TRAP    C#DCLN  ; ABORT THE PASS
2357 016526 000423  BR      5#
2358
2359          4# : RFLAGS  RO  ; GET THE OPERATOR FLAGS.
      016530 104421  TRAP    C#RFLA
2360 016532 032700 001000  BIT    @PNT,RO  ; PRINT THE TEST NUMBERS?
2361 016536 001412  BEQ    1#  ; BR IF NO
2362 016540 011600  MOV    (SP),RO  ;GET THE ID MESSAGE
2363 016542          PRINTF @TNAM,RO  ;DISPLAY THE TEST ID
      016542 010046  MOV    RO,-(SP)
      016544 012746 016606' MOV    @TNAM,-(SP)
      016550 012746 000002  MOV    @2,-(SP)
      016554 010600  MOV    SP,RO
      016556 104417  TRAP    C#PNTF
      016560 062706 000006  ADD    @6,SP
2364 016564 005237 002204' 1# : INC    TSTCNT  ; BUMP TEST COUNTER.
2365 016570          SETPRI IPRI  ;PRIORITY THAT OF DEVICE
      016570 013700 002202' MOV    IPRI,RO
      016574 104441  TRAP    C#SPRI
2366 016576 005726 5# : TST    (SP)+  ;FIX UP THE STACK
2367 016600 013705 002176' MOV    CSRADDR,RS  ; ADDRESS OF TSV REGISTERS ON UNIBUS
2368 016604 000207  RTS    PC
2369 016606 045 123 045 TNAM: .ASCIZ  '#S#T#A Test'
2370          .EVEN
2371
2372          .SBTTL  TSTEND - PRINT ERRORS RECEIVED
2373          ;
2374          ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2375          ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2376          ;
2377 TSTEND: RFLAGS  RO
      016622 104421  TRAP    C#RFLA
2378 016624 030027 020000  BIT    RO,@IER
2379 016630 001412  BEQ    1#  ; BR IF "IER" NOT SET.
2380 016632          PRINTF @ESUM,ERRK  ; PRINT ERROR COUNT.
      016632 013746 016660' MOV    ERRK,-(SP)
      016636 012746 016662' MOV    @ESUM,-(SP)
      016642 012746 000002  MOV    @2,-(SP)
      016646 010600  MOV    SP,RO
      016650 104417  TRAP    C#PNTF

```

```

016652 062706 000006
2381 016656 000207          1$: ADD #6,SP
2382                                RTS PC
2383 016660 000000          ERRK: 0 ; LOCAL ERROR COUNT.
2384 016662 045 101 040 ESUM: .ASCIZ /#A #D#A ERRORS/
2385 016701 105 122 122 EMAXDU: .ASCIZ /ERROR LIMIT REACHED -- DROPPING UNIT/
2386                                .EVEN
2387
2388                                .SBTTL INCERK - INCREMENT LOCAL ERROR COUNT
2389
2390                                ;*
2391                                ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
2392                                ;-
2392 016746 005237 016660' INCERK: INC ERRK ; INCREMENT LOCAL ERROR COUNT
2393 016752 010046 MOV RO,-(SP) ; SAVE RO
2394 016754 013700 002172' MOV UNITN,RO ; GET UNIT NUMBER,
2395 016760 006300 ASL RO ; ... AND MAKE IT A WORD OFFSET.
2396 016762 062700 003166' ADD @ERTABL,RO ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
2397 016766 005210 INC (RO) ; INCREMENT THE DEVICE ERROR COUNT
2398 016770 032710 007777 BIT @7777,(RO) ; DID WE OVERFLOW THE FIELD?
2399 016774 001001 BNE 1$ ; BR IF NO.
2400 016776 005310 DEC (RO) ; YES -- BACK IT UP TO 7777.
2401 017000 012600 1$: MOV (SP)+,RO ; RESTORE RO
2402 017002 000207 RTS PC ; RETURN TO CALLER.
2403
2404 017004 010046 CKEMAX: MOV RO,-(SP) ; SAVE RO
2405 017006 013700 002172' MOV UNITN,RO ; GET UNIT NUMBER
2406 017012 006300 ASL RO ; ... AND MAKE IT A WORD OFFSET
2407 017014 016000 003166' MOV ERTABL(RO),RO ; GET ERROR TABLE ENTRY
2408 017020 042700 170000 BIC @170000,RO ; EXTRACT ERROR COUNT FIELD
2409 017024 020037 002164' CMP RO,GERRMAX ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
2410 017030 103004 BHIS 1$ ; BR IF YES
2411 017032 023737 016660' 002162' CMP ERRK,LERRMAX ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2412 017040 103417 BLO 2$ ; BR IF NO
2413 017042 1$: RFLAGS RO ; GET OPERATOR FLAGS
2414 017042 104421 TRAP C#RFLA
2415 017044 032700 000040 BIT @IDU,RO ; IS DROPPING INHIBITED?
2416 017050 001013 BNE 2$ ; BR IF YES.
2417 017052 012737 177777 003102' MOV @-1,DUFLG ; NO -- DROP THE UNIT
2418 017060 ERRDF 4,EMAXDU
2419 017062 104455 TRAP C#ERDF
2420 017064 000004 .WORD 4
2421 017066 016701' .WORD EMAXDU
2422 017070 000000 .WORD 0
2423 017070 DODU UNITN
2424 017070 013700 002172' MOV UNITN,RO
2425 017074 104451 TRAP C#DODU
2426 017076 DOCLN
2427 017076 104444 TRAP C#DCLN
2428 017100 012600 2$: MOV (SP)+,RO ; RESTORE RO
2429 017102 000207 RTS PC ; RETURN TO CALLER
2430
2431                                .SBTTL CKDROP - CHECK IF UNIT SHOULD BE DROPPED
2432                                ;*
2433                                ; CHECK IF UNIT SHOULD BE DROPPED
2434                                ;-
2435 017104 010046 CKDROP: MOV RO,-(SP)
2436 017106 FORCERROR 1$,NOTSSR
    
```



```

2429 017116          RFLAGS RO
      017116 104421    TRAP   C#RFLA
2430 017120 032700 000040 BIT   @IDU,RO
2431 017124 001010    BNE   1$
2432 017126 011600    MOV   (SP),RO
2433 017130 012737 177777 003102' MOV   @-1,DUFLG
2434 017136          DODU   UNITN
      017136 013700 002172' MOV   UNITN,RO
      017142 104451    TRAP   C#DODU
2435 017144          DOCLN          ;ABORT THE PASS
      017144 104444    TRAP   C#DCLN
2436 017146 012600 1$: MOV   (SP)+,RO
2437 017150 000207    RTS    PC
2438
2439          .SBTTL  CONFIG - DETERMINE CONFIGURATION OF SYSTEM
2440          ;
2441          ; SUBROUTINE - DETERMINE CONFIGURATION OF TSU05 SYSTEM.
2442          ;
2443 017152          CONFIG:
2444 017152 004737 015674' JSR   PC,SOFINIT
2445 017156 000207    RTS    PC
2446
2447          .SBTTL  KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
2448          ;
2449          ; SUBROUTINE - ENABLE MEM MGT.
2450          ;
2451 017160 005737 003122' KTON: TST   KTFLG          ; GOT KT?
2452 017164 001403    BEQ   1$              ; NO.
2453 017166 012737 000001 177572 MOV   @1,SRO          ; YES. ENABLE KT11.
2454 017174 000207 1$: RTS    PC
2455
2456
2457
2458          ;
2459          ; SUBROUTINE - DISABLE MEM MGT.
2460          ;
2461 017176 005737 003122' KTOFF: TST   KTFLG          ; GOT KT11?
2462 017202 001405    BEQ   1$              ; NO.
2463 017204 000240    NOP
2464 017206 000240    NOP
2465 017210 012737 000000 177572 MOV   @0,SRO          ; DISABLE KT.
2466 017216 000207 1$: RTS    PC
2467
2468          .SBTTL  SETMAP - SETUP PAR6 MAPPING
2469
2470          ;*
2471          ;
2472          ; THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
2473          ; AN 22 BIT ADDRESS. THE OFFSET INTO THE PAGE
2474          ; IS RETURNED BIASED TO PAR6.
2475          ;
2476          ; INPUTS:
2477          ;
2478          ;     RO     HIGH ORDER ADDRESS BITS
2479          ;     R1     LOW ORDER ADDRESS BITS
2480          ;
2481          ; OUTPUTS:

```

```

2482
2483
2484
2485
2486
2487 017220
2488 017220
2489 017224 005737 003122'
2490 017230 001433
2491 017232 010102
2492
2493
2494
2495
2496 017264 042701 000177
2497 017270 020137 003122'
2498 017274 103011
2499 017276 010137 172354
2500 017302 042702 160000
2501 017306 062702 140000
2502 017312 010200
2503 017314 000261
2504 017316 000401
2505 017320 000241
2506 017322 000207
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524 017324
2525 017324
2526 017330 004737 017176'
2527 017334 010003
2528 017336 013701 003114'
2529 017342 013702 003116'
2530 017346 010321
2531 017350 005302
2532 017352 003375
2533 017354 005737 003122'
2534 017360 001502
2535 017362 004737 017160'
2536 017366 005000
2537 017370 013701 003142'
2538

```

```

;
;      RO      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
;      CARRY   SET IF SUCCESS
;              CLR IF ERROR
;
;--
SETMAP:
  SAVREG
  TST      KTFLG
  BEQ      10$
  MOV      R1,R2
  .REPT    6
  ASR      RO
  ROR      R1
  .ENDR
  BIC      @177,R1
  CMP      R1,KTFLG
  BHIS     10$
  MOV      R1,@KIPAR6
  BIC      @160000,R2
  ADD      @140000,R2
  MOV      R2,RO
  SEC
  BR       15$
10$:     CLC
15$:     RTS      PC
;
;      ;SAVE R1-R4 UNTIL NEXT RETURN
;      ;SYSTEM HAVE ABOVE 28K?
;      ;BR IF NO
;      ;SAVE LOW ORDER BITS
;
;      ;CONVERT WORD ADDRESS TO 32W BLOCKS
;      ;MAKE IT DOUBLE PRECISION
;
;      ;ALINE FOR LOWER 4K BOUNDARY
;      ;HIGHER THAN EXISTING MEMORY?
;      ;BR IF YES
;      ;SETUP MAPPING REGISTER PAR6
;      ;SETUP DISPLACEMENT IN PAGE
;      ;ADD IN PAR6 BIAS
;      ;RETURN IN RO
;      ;SET SUCCESS
;
;      ;SET FAILURE
;      ;RETURN
;
;      .SBTTL  FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
;
;      ;*
;      ; FILL MEMORY WITH A BACKGROUND PATTERN
;
;      ; INPUTS:
;
;      ;      RO = BACKGROUND PATTERN
;      ;      FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
;      ;      KTFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
;
;      ; OUTPUTS:
;
;      ;      NONE
;
;      ;--
;
;      FILLMEM:
;      SAVREG
;      JSR      PC,KTOFF
;      MOV      RO,R3
;      MOV      FREE,R1
;      MOV      FRESIZ,R2
10$:     MOV      R3,(R1)+
;      DEC      R2
;      BGT      10$
;      TST      KTFLG
;      BEQ      55$
;      JSR      PC,KTON
;      CLR      RO
;      MOV      PST32W,R1
;      .REPT    6
;
;      ;SAVE R1-R5 UNTIL NEXT RETURN
;      ;DISABLE KT.
;      ;COPY TEST PATTERN
;      ;GET FIRST FREE LOCATION
;      ;SIZE OF FREE SPACE BELOW 28K.
;      ;STORE A BACKGROUND WORD
;      ;DONE ALL MEMORY IN FREE SPACE?
;      ;BR IF NO
;      ; GOT KT?
;      ; NO. GET OUT.
;      ; YES. ENABLE KT.
;      ;HIGH ORDER ADDRESS START
;      ;GET >28K START ADDRESS (IN 32W BLOCKS)

```

```

2539          CLC          ;CLEAR C BIT
2540          ROL          R1  ;CONVERT BLOCKS TO WORDS
2541          ROL          R0  ;MAKE IT DOUBLE PRECISION
2542          .ENDR
2543 017440 004737 017220'   JSR    PC,SETMAP    ;SETUP PAR6 MAPPING REGISTER
2544 017444 010320          30$: MOV    R3,(R0)+    ;STORE TEST PATTERN IN >28K ADDRESS
2545 017446 020027 160000   CMP    R0,#160000   ;END OF PAR6 MAPPING AREA?
2546 017452 103774          BLO    30$         ;BR IF NO
2547 017454 162700 020000   SUB    #20000,R0    ;BACKUP INTO PAR6 MAPPING BEGIN
2548 017460 062737 000200 172354 ADD    #200,#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2549 017466 013705 003122'   MOV    KTFLG,R5     ;GET VALUE FROM MEMORY SIZER
2550 017472 042705 170000   BIC    #170000,R5   ;ONLY 18 BITS PASS
2551 017476 023705 172354   CMP    #KIPAR6,R5   ;END OF MEMORY?
2552 017502 001427          BEQ    50$         ;BR IF YES
2553 017504 005737 003134'   TST    T23A        ;PROCESSOR TYPE A
2554 017510 001407          BEQ    35$         ;NO KEEP GOING
2555 017512 013704 177572   MOV    SRO,R4       ;GET SRO CONTENTS
2556 017516 042704 177761   BIC    #177761,R4   ;CLEAR ALL BUT PAGE NUMBER
2557 017522 022704 000016   CMP    #16,R4       ;SEE IF PAGE 7
2558 017526 001415          BEQ    50$         ;EXIT IF THERE
2559 017530 005737 003136'   35$: TST    T23B        ;PROCESSOR TYPE B
2560 017534 001410          BEQ    45$         ;NO KEEP GOING
2561 017536 023727 172354 007600 CMP    #KIPAR6,#7600 ;REACHED 18 BITS?
2562 017544 103001          BHS    40$         ;YES
2563 017546 000403          BR     45$         ;NO KEEP GOING
2564 017550 012737 000020 172516 40$: MOV    #20,SRO     ;SET MMU RELOCATION
2565 017556 000137 017444'   45$: JMP    30$         ;KEEP GOING ON ETC.
2566 017562 004737 017176'   50$: JSR    PC,KTOFF  ;DISABLE KT.
2567 017566 000207          55$: RTS    PC
2568
2569          .SBTTL CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
2570          ;*
2571          ; COMPARE MEMORY WITH A BACKGROUND PATTERN
2572          ;
2573          ; INPUTS:
2574          ;
2575          ;     RO = BACKGROUND PATTERN
2576          ;     FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2577          ;     KTFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2578          ;
2579          ; OUTPUTS:
2580          ;
2581          ;     CARRY - SET IF NO ERROR
2582          ;     CARRY - CLR IF ERROR
2583          ;
2584          ; IMPLICIT OUTPUTS:
2585          ;
2586          ;     ERRHI - ERROR HIGH ADDRESS
2587          ;     ERRLO - ERROR LOW ADDRESS
2588          ;     EXPD  - EXPECTED DATA
2589          ;     RECV  - RECEIVED DATA
2590          ;-
2591 017570          CMPMEM: SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
2592 017570          MOV    RO,R3          ;COPY TEST PATTERN
2593 017574 010003          JSR    PC,KTOFF    ;DISABLE KT.
2594 017576 004737 017176'   MOV    FREE,R1     ;GET FIRST FREE LOCATION
2595 017602 013701 003114'

```

```

2596 017606 013702 003116'      MOV    FRESIZ,R2      ;SIZE OF FREE SPACE BELOW 28K.
2597 017612 020311      10$:  CMP    R3,(R1)      ;FREE SPACE LOCATION EQUAL TO EXPD?
2598 017614 001411      BEQ    15$           ;BR IF YES
2599 017616 010137 002230'      MOV    R1,ERRLO     ;SAVE ADDRESS IN ERROR
2600 017622 005037 002226'      CLR    ERRHI        ;NO HIGH ADDRESS
2601 017626 010337 002222'      MOV    R3,EXPD      ;SAVE EXPD FOR ERROR REPORT
2602 017632 011137 002224'      MOV    (R1),RECV    ;SAVE RECV FOR ERROR REPORT
2603 017636 000474      BR     50$          ;
2604 017640 005721      15$:  TST    (R1)+        ;POINT TO NEXT ADDRESS
2605 017642 005302      DEC    R2           ;DONE ALL MEMORY IN FREE SPACE?
2606 017644 003362      BGT    10$          ;BR IF NO
2607 017646 005737 003122'      TST    KTFLG        ; GOT KT?
2608 017652 001472      BEQ    55$          ; NO. GET OUT.
2609 017654 004737 017160'      JSR    PC,KTON      ; YES. ENABLE KT.
2610 017660 005000      CLR    R0           ;HIGH ORDER ADDRESS START
2611 017662 013701 003142'      MOV    PST32W,R1    ;GET >28K START ADDRESS (IN 32W BLOCKS)
2612      .REPT 6
2613      ROL    R1        ;CONVERT BLOCKS TO WORDS
2614      ROL    R0        ;MAKE IT DOUBLE PRECISION
2615      .ENDR
2616 017716 042701 000177      BIC    #177,R1      ;ALINE 4K BOUNDARY
2617 017722 010046      MOV    R0,-(SP)     ;SAVE HIGH ORDER
2618 017724 010146      MOV    R1,-(SP)     ;SAVE LOW ORDER
2619 017726 004737 017220'      JSR    PC,SETMAP    ;SETUP PAR6 MAPPING REGISTER
2620 017732 010004      MOV    R0,R4        ;COPY ADDRESS BIASED TO PAR6
2621 017734 012601      MOV    (SP)+,R1     ;RESTORE LOW ORDER IN NON PAR6 FORMAT
2622 017736 012600      MOV    (SP)+,R0     ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
2623 017740 020314      30$:  CMP    R3,(R4)      ;ABOVE 28K LOCATION EQUAL EXPD?
2624 017742 001411      BEQ    32$          ;BR IF YES
2625 017744 010037 002226'      MOV    R0,ERRHI    ;SAVE HIGH ORDER IN ERROR
2626 017750 010137 002230'      MOV    R1,ERRLO    ;SAVE LOW ORDER IN ERROR
2627 017754 010337 002222'      MOV    R3,EXPD     ;SAVE EXPD FOR ERROR REPORT
2628 017760 011437 002224'      MOV    (R4),RECV   ;SAVE RECV FOR ERROR REPORT
2629 017764 000421      BR     50$          ;
2630 017766 062701 000002      32$:  ADD    #2,R1        ;UPDATE NON PAR6 ADDRESS
2631 017772 005500      ADC    R0           ;MAKE IT DOUBLE PRECISION ADD
2632 017774 062704 000002      ADD    #2,R4        ;UPDATE PAR FORMAT ADDRESS
2633 020000 020427 160000      CMP    R4,#160000   ;END OF PAR6 MAPPING AREA?
2634 020004 103755      BLO    30$          ;BR IF NO
2635 020006 162704 020000      SUB    #2000,R4     ;BACKUP INTO PAR6 MAPPING BEGIN
2636 020012 062737 000200 172354      ADD    #200,#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2637 020020 023737 172354 003122'      CMP    #KIPAR6,KTFLG ;END OF MEMORY?
2638 020026 101744      BLOS   30$          ;BR IF NO
2639 020030 004737 017176'      50$:  JSR    PC,KTOFF     ;TURN OFF MEMORY MAPPING
2640 020034 000241      CLC                    ;SET FAILURE
2641 020036 000403      BR     60$          ;
2642 020040 004737 017176'      55$:  JSR    PC,KTOFF     ;TURN OFF MEMORY MAPPING
2643 020044 000261      SEC                    ;SET SUCCESS
2644 020046 000207      60$:  RTS    PC
2645
2646      .SBTTL REGSAV - SAVE R1-R5 ON STACK
2647      ;*
2648      ;
2649      ;ROUTINE TO
2650      ;SAVE R1 THROUGH R5 ON THE STACK
2651      ;
2652      ;CALLING SEQUENCE:
  
```

```

2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666 020050
2667 020050 010446
2668 020052 010346
2669 020054 010246
2670 020056 010146
2671 020060 010546
2672 020062 016605 000012
2673 020066 004736
2674 020070 012601
2675 020072 012602
2676 020074 012603
2677 020076 012604
2678 020100 012605
2679 020102 000207
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700 020104
2701 020104
2702 020110
      020110 104443
      020112 000406
      020114 020140'
      020116 000022
      020120 020142'
      020122 000377
      020124 000000

```

```

;
;       JSR       R5,REGSAV
;
; THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
; THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
; THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
; REGISTERS.
;
; THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
; CALLED VIA A JSR PC INSTRUCTION
;
; -
REGSAV:
      MOV        R4,-(SP)
      MOV        R3,-(SP)
      MOV        R2,-(SP)
      MOV        R1,-(SP)
      MOV        R5,-(SP)
      MOV        10.(SP),R5
      JSR        PC,8(SP)+
      MOV        (SP)+,R1
      MOV        (SP)+,R2
      MOV        (SP)+,R3
      MOV        (SP)+,R4
      MOV        (SP)+,R5
      RTS        PC

      .SBTTL  GETPAT - GET 8 BIT PATTERN FROM OPERATOR
; *
;
; ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
;
; INPUTS:
;
;       NONE.
;
; OUTPUTS:
;
;       R0      OCTAL NUMBER FROM THE OPERATOR
;
; CALLING SEQUENCE:
;
;       JSR     PC,GETPAT
;
; -
GETPAT::
      SAVREG          ;SAVE THE GENERAL REGISTERS
1$:   GMANID  DATASC,PATDAT,0,377,0,377,NO
      TRAP    C$GMAN
      BR      10000$
      .WORD  PATDAT
      .WORD  T$CODE
      .WORD  DATASC
      .WORD  377
      .WORD  T$LOLIM

```

```

020126 000377          .WORD  T$HILIM
020130          10000$:
2703 020130          BNCOMPLETE      1$      ;RETRY IF ERROR
020130 103367          BCC      1$
2704 020132 013700 020140'  MOV      PATDAT,R0      ;DATA PATTERN FROM OPERATOR
2705 020136 000207          RTS      PC      ;RETURN TO CALLER
2706
2707          ;+
2708          ;LOCAL DATA AREA
2709          ;-
2710
2711 020140 000000          PATDAT: .WORD  0      ;TEMPORARY STORAGE FOR DATA
2712 020142 105      116 124 DATASC: .ASCIZ 'ENTER DATA PATTERN'
2713          .EVEN
2714
2715          .SBTTL  GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
2716          ;+
2717          ;
2718          ;ROUTINE TO ISSUE A MENU AND GET
2719          ;THE OPERATOR'S RESPONSE.
2720          ;
2721          ;INPUTS:
2722          ;
2723          ;      R0      ADDRESS OF ASCIZ STRING OF MENU
2724          ;      R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
2725          ;
2726          ;OUTPUTS:
2727          ;
2728          ;      R0      NUMBER OF THE OPERATOR'S SELECTION
2729          ;
2730          ;-
2731
2732          GETSEL::
2733 020166          SAVREG          ;SAVE GENERAL REGISTERS
2734 020172 010002          MOV      R0,R2      ;SAVE THE MENU ADDRESS
2735 020174 010203          1$: MOV      R2,R3      ;START OF MENU STRING
2736 020176 005713          2$: TST      (R3)      ;END OF ASCII ?
2737 020200 001412          BEQ      3$      ;BRANCH IF ALL LINES DISPLAYED
2738 020202          PRINTF  #SELASC,(R3)+ ;DISPLAY THE MENU
020202 012346          MOV      (R3)+,-(SP)
020204 012746 020352'          MOV      #SELASC,-(SP)
020210 012746 000002          MOV      #2,-(SP)
020214 010600          MOV      SP,R0
020216 104417          TRAP   C$PNTF
020220 062706 000006          ADD      #6,SP
2739 020224 000764          BR      2$
2740 020226          3$: GMANID  MENASC,MENRES,D,-1,0,-1,NO
020226 104443          TRAP   C$GMAN
020230 000406          BR      10001$
020232 020406'          .WORD  MENRES
020234 000042          .WORD  T$CODE
020236 020357'          .WORD  MENASC
020240 177777          .WORD  -1
020242 000000          .WORD  T$LOLIM
020244 177777          .WORD  T$HILIM
020246
2741 020246          10001$: BNCOMPLETE      1$      ;RETRY IF ERROR

```

```

020246 103352
2742 020250 013700 020406' BCC 1$
MOV MENRES,R0 ;GET THE OPERATOR'S REPLY
2743 020254 020001 CMP R0,R1 ;COMPARE TO MAXIMUM ALLOWED
2744 020256 101411 BLOS 5$ ;BRANCH IF OK
2745 020260 PRINTF #MENERR ;DISPLAY ERROR MESSAGE
020260 012746 020304' MOV #MENERR,-(SP)
020264 012746 000001 MOV #1,-(SP)
020270 010600 MOV SP,R0
020272 104417 TRAP C$PNTF
020274 062706 000004 ADD #4,SP
2746 020300 000735 BR 1$ ;RETRY
2747 020302 000207 5$: RTS PC ;RETURN TO CALLER
2748 020304 045 116 045 MENERR: .ASCIZ '#N#A *** Menu Selection Too Large ***'
2749 020352 045 116 045 SELASC: .ASCIZ '#N#T'
2750 020357 105 156 164 MENASC: .ASCIZ 'Enter Menu Selection: '
2751 .EVEN
2752 020406 000000 MENRES: .WORD 0
2753
2754 .SBTTL CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
2755 ;+
2756 ;
2757 ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2758 ;
2759 ;INPUT:
2760 ;
2761 ; NONE.
2762 ;
2763 ;OUTPUT:
2764 ;
2765 ; CARRY 0 MANUAL INTERVENTION NOT ALLOWED
2766 ; 1 MANUAL INTERVENTION IS OK
2767 ;
2768 ;SIDE EFFECTS:
2769 ;
2770 ; A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2771 ; NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2772 ; ALLOWED.
2773 ;
2774 ;-
2775
2776 020410 CHKMAN::
2777 020410 SAVREG ;SAVE THE REGISTERS
2778 020414 MANUAL ;SEE IF MANUAL INTERVENTION OK
020414 104450 TRAP C$MANI
2779 020416 BCOMPLETE 1$ ;BRANCH IF ALLOWED
020416 103411 BCS 1$
2780 020420 PRINTF #NOMAN ;PRINT THE WARNING MESSAGE
020420 012746 020444' MOV #NOMAN,-(SP)
020424 012746 000001 MOV #1,-(SP)
020430 010600 MOV SP,R0
020432 104417 TRAP C$PNTF
020434 062706 000004 ADD #4,SP
2781 020440 000241 CLC ;CLEAR CARRY FOR ERROR
2782 020442 000207 1$: RTS PC ;RETURN
2783
2784 020444 045 116 045 NOMAN: .ASCIZ '#N#A *** Manual Intervention not Allowed - Test Aborted ***'
2785 .even

```

```

2786
2787
2788
2789
2790
2791 020540
    020540 104431
2792 020542 010037 003114'
2793 020546 062737 000002 003114'
2794 020554 011037 003116'
2795 020560 162737 000004 003116'
2796 020566 013702 002012'
2797 020572 162737 000007 003116' 10$:
2798 020600 005302
2799 020602 001373
2800 020604 013700 003114'
2801 020610 063700 003116'
2802 020614 162700 000002
2803 020620 010037 003120'
2804 020624 000207 40$:
2805
2806
2807
2808
2809
2810
2811
2812
2813 020626
2814 020626 005037 003122'
2815 020632 005037 003124'
2816 020636 023727 002120' 001577
2817 020644 101453
2818 020646 023727 002120' 001777
2819 020654 101447
2820 020656 013700 000004
2821 020662 012737 020754' 000004
2822 020670 005737 177572
2823 020674 000240
2824 020676 013737 002120' 003122'
2825 020704 042737 000177 003122'
2826 020712 010037 000004
2827 020716 005000
2828 020720 012701 172340
2829 020724 012761 077406 177740 1$:
2830 020732 010021
2831 020734 062700 000200
2832 020740 020027 002000
2833 020744 001367
2834 020746 012741 177600
2835 020752 000410
2836
2837 020754 012716 020770' 2$:
2838 020760 000002
2839
2840
2841 020762 012716 021016' 3$:
    
```

.SBTTL ENVIRN - SETUP FREE DIAGNOSTIC SPACE

```

;
; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
;
ENVIRN: MEMORY R0
        TRAP C$MEM
        MOV R0,FREE ; GET 1ST FREE ADDRESS...
        ADD #2,FREE ; ...AND WORD COUNT.
        MOV (R0),FRESIZ
        SUB #4,FRESIZ
        MOV L$UNIT,R2 ; GET NUMBER OF UNITS
        SUB #7,FRESIZ ; TAKE AWAY 7 WORDS PER UNIT
        DEC R2
        BNE 10$
        MOV FREE,R0 ;GET FIRST FREE ADDRESS
        ADD FRESIZ,R0 ;POINT TO LAST FREE ADDRESS
        SUB #2,R0 ;BACKUP 1 WORD
        MOV RO,FREEHI ;STORE LAST FREE ADDRESS
        RTS PC ;RETURN
    
```

.SBTTL KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

;+
;
;ROUTINE TO INIT KT-11
;
;-
    
```

KTINIT:

```

        CLR KTFLG ; INIT >28K MEMORY FLAG
        CLR KTENABLE ; INIT TEST >28K FLAG
        CMP L$HIME,#1577 ; GOT ENOUGH MEMORY (>28K)?
        BLOS 9$ ; NO.
        CMP L$HIME,#1777 ; GOT ENOUGH MEMORY (>32K)?
        BLOS 9$ ; NO.
        MOV @ERRVEC,R0 ; SAVE OLD ERR VEC PTR.
        MOV #2,@ERRVEC ; SET ERR VEC PTR.
        TST @SRO ; GOT KT11?
        NOP ; (TRAP IF NO).
        MOV L$HIME,KTFLG ; YES. SET KT FLAG.
        BIC #177,KTFLG
        MOV RO,@ERRVEC ; RESTORE OLD ERR VEC PTR.
        CLR RO ; RO = AR DATA.
        MOV #KIPARO,R1 ; R1 = KI REGS PTR.
        MOV #77406,-40(R1) ; SET DESCRIPTOR REG.
        MOV RO,(R1)+ ; SET KIPAR REG.
        ADD #200,RO ; BUMP AR DATA BY "4K".
        CMP RO,#2000 ; AT "I/O"?
        BNE 1$ ; NO.
        MOV #177600,-(R1) ; YES. SET KTPAR7 FOR I/O.
        BR 9$
    
```

```

        MOV #6$,(SP) ; SET UP RETURN
        RTI ; RTI TO NEXT LOCATION
    
```

```

        MOV #10$,(SP) ; SET UP RETURN
    
```



```

2842 020766 000002          RTI          ; RTI TO NEXT LOCATION
2843
2844 020770 010037 000004    6$:    MOV    RO,@ERRVEC    ; RESTORE OLD ERR VEC PTR.
2845
2846 020774          9$:
2847 020774 013700 000004    MOV    @ERRVEC,RO    ; SAVE OLD ERR VEC PTR.
2848 021000 012737 020762' 000004    MOV    @3,@ERRVEC    ; SET ERR VEC PTR.
2849 021006 042737 000001 170200    BIC    @BIT0,@MMRO    ; BE SURE UNIBUS MAP IS OFF
2850 021014 000240          NOP
2851 021016 010037 000004    10$:  MOV    RO,@ERRVEC    ; RESET VECTOR BACK TO ERROR POINTER
2852 021022 000207          RTS    PC
2853
2854
2855
2856          ;*
2857          ; SUBROUTINE TO SET EXTENDED FEATURES SWITCH
2858          ;
2859          ; Requires that SOFINIT and WRTCHR have been done previous to call.
2860          ;
2861          ; INPUTS:
2862          ; R5 CURRENT UNIT NUMBER
2863          ; OUTPUTS:
2864          ; The Extended Features Switch is set.
2865          ;
2866          ;-
2867
2868 021024          INVERT::
2869
2870 021024 005737 002216'          TST    EXTFEA          ; IS SWITCH SET?
2871 021030 001020          BNE    1$              ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
2872 021032 012737 100206 021100'    MOV    @100206,CMDPKT ; WRT SUB-SYS MEM CMD
2873 021040 012737 021110' 021102'    MOV    @WSMBK,CMDPKT+2 ; MSG BUF ADDR
2874 021046 012737 000006 021106'    MOV    @6,CMDPKT+6     ; BYTE COUNT
2875 021054 012737 100010 021110'    MOV    @100010,WSMBK   ; INVERT THE SWITCH
2876 021062 012704 021100'    MOV    @CMDPKT,R4      ; SET CMDPKT INTO R4
2877 021066 004737 010562'    JSR    PC,WRTCHR        ; DO IT
2878 021072 000207    1$:    RTS    PC              ; RETURN
2879
2880
2881          ; COMMAND PACKET.
2882
2884 021074          .BLKB 10-<.-TSV2&7>
2886
2887 021100 000000          CMDPKT:: 0              ;1ST WORD IS TS05 COMMAND.
2888 021102 000000          0              ;2ND WORD IS THE BUFFER LOW ADDRESS.
2889 021104 000000          0              ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2890 021106 000000          0              ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2891
2892
2893          ; WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
2894
2895 021110 000000          WSMBK:: 0              ;1ST WORD:: SEL 0
2896 021112 000000          0              ;2ND WORD:: SEL 2
2897 021114 000000          0              ;3RD WORD:: SEL 4
2898          .EVEN
2899
2900          ;*

```

```

2901      ; SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
2902      ;
2903      ;
2904      ; INPUTS:
2905      ; OUTPUTS:
2906      ; The NXMFLG is set if we can test.
2907      ; The NXMLO and NXMMHI addresses are setup.
2908      ;
2909      ;
2910      MEMCK::
2911
2912      SAVREG      ;SAVE THE REGISTERS
2913      CLR        NXMFLG      ;CLEAR THE FLAG
2914      CLR        NXMLO      ;CLEAR THE TEST ADDRESS LO
2915      CLR        NXMMHI     ;CLEAR THE TEST ADDRESS HI
2916      BIT        @170000,L#HIME ;CHECK FOR MORE THAN 18 BITS INDICATED
2917      ;FROM THE SUPERVISOR
2918      BNE        14#        ;BR, IF MAP BOX ETC.
2919      TST        T23B       ;IS IT A PROCESSOR TYPE B?
2920      BEQ        1#        ;NO
2921      CMP        L#HIME,@7777 ; GREATER THAN 128K
2922      BLO        2#        ; NO
2923      JSR        PC,NXMTST  ;SETUP THE ADDRESS
2924      BR         13#       ;SET THE FLAG AND EXIT
2925      TST        T23A       ;IS IT A PROCESSOR TYPE A?
2926      BEQ        4#        ;NO
2927      CMP        L#HIME,@5777 ;GREATER THAN 96K
2928      BHI        14#       ;YES,23A/23B WITH 128K MEMORY
2929      CMP        L#HIME,@3777 ;GREATER THAN 64K BUT LESS THAN 92K?
2930      BLO        4#        ;NO, CHECK 24K
2931      JSR        PC,NXMTST  ;SETUP THE ADDRESS
2932      BR         13#       ;SET THE FLAG AND EXIT
2933      CMP        L#HIME,@1577 ;GREATER THAN 24K BUT LESS THAN 64K?
2934      BLO        14#       ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
2935      JSR        PC,NXMTST  ;SETUP THE ADDRESS
2936      ADD        @77,NXMMHI ;FOOL THE 11/02 & 11/03
2937      BIT        @177774,NXMMHI ;ANY MORE THAN 18 BITS SET?
2938      BNE        15#       ;BR, IF MORE THAN 18 BITS SET
2939      INC        NXMFLG     ;SET THE FLAG
2940      BR         15#       ;EXIT
2941      BR         14#       ;NOP FOR PRINTOUT
2942      PRINTF     @NOMEM     ;TELL THEM & EXIT ***NO PRINT*****
2943      MOV        @NOMEM,-(SP)
2944      MOV        @1,-(SP)
2945      MOV        SP,RO
2946      TRAP      C#PNTF
2947      ADD        @4,SP
2948      RTS        PC        ;RETURN
2949
2950      ;
2951      ; SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
2952      ;
2953      ;
2954      ; OUTPUTS: NXMLO, NXMMHI      ; SETUP WITH NXM ADDRESS
2955      ;
2956      ;

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:54
 KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

SEQ 081

2953	021312	013701	002120'	NXMTST: MOV	L#HIME,R1	;GET TOP OF MEMORY
2954	021316	062701	000200	ADD	#200,R1	;MAKE IT I/O BLOCK OR OTHER NXM
2955	021322	042701	000177	BIC	#177,R1	
2956	021326	010102		MOV	R1,R2	;RESAVE RESULTS
2957		000006		.REPT	6	
2958				ASL	R1	;PUT IN PLACE FOR XFER
2959				.ENDR		
2960	021344	010137	003130'	MOV	R1,NXMLO	;SAVE TEST ADDRESS LOW
2961		000012		.REPT	10.	
2962				ASR	R2	;PUT IN PLACE FOR XFER
2963				.ENDR		
2964	021374	042702	177700	BIC	#177700,R2	;DON'T WANT ILA!
2965	021400	010237	003132'	MOV	R2,NXMH1	;SAVE TEST ADDRESS HIGH
2966	021404	000207		RTS	PC	;RETURN
2967						
2968						
2969						
2970						
2971	021406			ENDMOD		

```

6          .TITLE  TSV4 - MISCELLANEOUS SECTIONS
7
8 021406   BGNMOD  TSV4
9 021406   TSV4::
10
11
12
13
14          .SBTTL  PROTECTION TABLE
15          BGNPROT
16
17 021406   L$PROT::
18 021406   177777 177777 177777 .WORD  -1, -1, -1, -1          ;NO DEVICE PROTECTION REQUIRED.
19 021416   ENDPROT
20
21          .SBTTL  INITIALIZE SECTION
22
23          ;**
24          ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
25          ;AT THE BEGINNING OF EACH PASS.
26          ;
27          ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
28          ;IF "CONTINUE", NOTHING IS REQUIRED.
29          ;
30          ;--
31          ;*
32          ;INSERT TEMPORARY JUMP TO ODT
33          ;-
34 021416   BGNINIT
35 021416   L$INIT::
36 021416   005037 002216' 40$: CLR      EXTFEA
37 021422   005037 003126' CLR      NXMFLG
38 021426   012737 006166' 002170' MOV     @EPT1,EPTSW          ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
39 021434   005037 003144' CLR      SIFLAG          ;CLEAR "SOFT INIT" FLAG
40 021440   005037 003124' CLR      KTENABLE        ;CLEAR TEST ABOVE 28K FLAG
41 021444   005037 002272' CLR      RAMSIZ          ;CLEAR RAM SIZE FOR RAMERR ROUTINE
42 021450   021450 012700 000036 READEF  @EF.CONTINUE
43 021454   104447 MOV     @EF.CONTINUE,R0
44 021456   103023 TRAP   C$REFG
45 021460   023737 002172' 002012' BNCOMPLETE 1$
46 021466   103070 BCC    1$
47 021470   005737 003102' CMP     UNITN,L$UNIT          ;UNIT IN RANGE?
48 021474   100472 BHIS   4$                   ;BR IF NO.
49 021476   013701 002172' TST     DUFLG              ;DROPPED UNIT?
50 021502   006301 BMI    NXTU                 ;BR IF YES
51 021504   005761 003166' MOV     UNITN,R1
52 021510   001516 ASL    R1
53 021512   032761 040000 003166' TST     ERTABL(R1)
54 021520   001060 BEQ    SETU
55 021522   104432 BIT    @BIT14,ERTABL(R1)          ;DROPPED?
56 021524   000416 BNE    NXTU
57 021526   012700 000035 EXIT   INIT                ;DO NOTHING IF "CONTINUE".
58 021532   104447 TRAP   C$EXIT
59 021534   103052 .WORD  L10030-.
60 021536   000035 READEF  @EF.NEW
61          MOV     @EF.NEW,R0
62          TRAP   C$REFG
63          BNCOMPLETE NXTU          ;TAKE NEXT UNIT IF NOT NEW PASS.
64          BCC    NXTU
65          READEF @EF.START

```

021536	012700	000040		MOV	#EF,START,RO	
021542	104447			TRAP	C\$REFG	
57 021544				BCOMPLETE	2\$	
021544	103404			BCS	2\$	
58 021546				READEF	#EF.RESTART	
021546	012700	000037		MOV	#EF.RESTART,RO	
021552	104447			TRAP	C\$REFG	
59 021554				BNCOMPLETE	31\$	
021554	103031			BCC	31\$	
60 021556			2\$:			;1ST PASS, BUS-INIT...
61 021556				BRESET		;BUS RESET.
021556	104433			TRAP	C\$RESET	
62 021560	005037	002204'		CLR	TSTCNT	;NUMBER OF TESTS RUN IN PASS
63 021564	005037	002212'		CLR	FATFLG	;CLEAR FATAL ERROR COUNT
64 021570	005037	003134'		CLR	T23A	;CLEAR PROCSSOR TYPE A FLAG
65 021574	005037	003136'		CLR	T23B	;CLEAR PROCSSOR TYPE B FLAG
66			:	MOV	#340,-(SP)	
67			:	MOV	#20,-(SP)	;RETURN TO DEBUGGER
68			:	JMP	O.ODT	;ENTER THE DEBUGGER
69 021600	005037	003370'		CLR	SKIPT	;CLEAR THE SUBTEST "SKIPPER"
70 021604			20\$:			
71 021604	012737	177777	002174'	MOV	#-1,QVP	;...QUICK VERIFY...
72 021612	004737	020540'		JSR	PC,ENVIRN	;SET ENVIRONMENT.
73 021616	004737	020626'		JSR	PC,KTINIT	;INITIALIZE KT MEMORY MANAGEMENT
74 021622	012700	003166'		MOV	#ERTABL,RO	
75 021626	005020		30\$:	CLR	(RO),	;CLEAR THE ERROR TABLE
76 021630	020027	003366'		CMP	RO,#ERTABE	
77 021634	103774			BLO	30\$	
78 021636	000404			BR	4\$	
79 021640	005037	002174'	31\$:	CLR	QVP	
80 021644	000137	021714'		JMP	PASRPT	;GO REPORT THE STATUS
81						
82 021650			4\$:			
83 021650	012737	177777	002172'	NEWPAS:	MOV #-1,UNITN	;INIT UNIT NUMBER...
84 021656	005037	002210'		CLR	DEV CNT	;CLEAR COUNT OF DEVICES RUNNING
85 021662			NXTU:	BREAK		
021662	104422			TRAP	C\$BRK	
86 021664	005237	002172'		INC	UNITN	;...AND SET NEXT UNIT NUMBER.
87 021670	023737	002172'	002012'	CMP	UNITN,L\$UNIT	
88 021676	103423			BLO	SETU	
89 021700	012737	177777	003102'	MOV	#-1,DUFLG	
90 021706	000401			BR	11\$	
91 021710				DOCLN		;ABORT, NO MORE UNITS.
021710	104444			TRAP	C\$DCLN	
92 021712	000240		11\$:	NOP		
93 021714			PASRPT:			
94 021714	023727	002012'	000001	CMP	L\$UNIT,#1	;HOW MANY UNITS SELECTED?
95 021722	101752			BLOS	NEWPAS	;BR IF ONLY 1
96 021724	005737	002210'		TST	DEV CNT	;ARE ANY STILL RUNNING?
97 021730	001747			BEQ	NEWPAS	;BR IF NO
98 021732				RFLAGS	RO	
021732	104421			TRAP	C\$RFLA	
99 021734	032700	000100		BIT	#ISR,RO	;SHOULD WE PRINT STATISTICS
100 021740	001343			BNE	NEWPAS	;BR IF NO
101						
102 021742				DORPT		
021742	104424			TRAP	C\$DRPT	

```

103 021744 000741
104 021746
105
106 021746
021746 013700 002172'
021752 104442
107 021754
021754 103342
108 021756 005037 003102'
109 021762 005237 002210'
110 021766 012001
111 021770 010137 002176'
112
113 021774 012001
114
115
116 021776 010137 002200'
117 022002 012721 016116'
118 022006 013721 002202'
119
120 022012
121
122
123
124
125
126
127
128 022012 013701 002172'
129 022016 006301
130 022020 052761 100000 003166'
131 022026 005037 005600'
132 022032 023727 002012' 000001
133 022040 101416
134 022042
022042 104421
135 022044 032700 001000
136 022050 001412
137 022052
022052 013746 002172'
022056 012746 022144'
022062 012746 000002
022066 010600
022070 104417
022072 062706 000006
138 022076
139 022076 005037 003104'
140 022102 013701 002176'
141 022106 010102
142 022110 062702 000002
143 022114 004737 016276'
144 022120 103005
145 022122 010137 003104'
146 022126 012737 177777 003102'
147 022134
148
149

10$: BR NEWPAS
SETU: GPHARD UNITN,R0 ;GET UNIT N P-TABLE POINTER.
MOV UNITN,R0
TRAP C:GPHRD
BNCOMPLETE NXTU ;BR IF UNIT NOT AVAILABLE.
BCC NXTU
CLR DUFLG ;CLEAR "DROPPED" FLAG.
INC DEVCNT
MOV (R0)+,R1 ;GET 1ST REGISTER ADDRESS.
MOV R1,CSRADDR ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
MOV (R0)+,R1 ;GET VECTOR ADDRESS.
;MOV (R0),R2 ;GET INTERRUPT PRIORITY
;MOV R2,IPRI ;SET INTERRUPT PRIORITY.
MOV R1,IVEC ;SET INTERRUPT VECTOR POINTER...
MOV @INTR,(R1)+ ;...VECTOR...
MOV IPRI,(R1)+ ;...AND PRIORITY.

1$:
: TST QVP ;1ST PASS ??
: BEQ 5$ ;NO, SKIP THE PASS 1 STUFF.

;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
:
MOV UNITN,R1
ASL R1
BIS @BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
CLR EXTA ;CLEAR ERROR EXTENSION FLAG.
CMP L:UNIT,@1 ;ARE WE TESTING MULTIPLE UNITS?
BLOS 10$ ;BR IF NO.
RFLAGS RO ;YES -- GET OPERATOR FLAGS.
TRAP C:RFLA
BIT @PNT,RO ;SHOULD WE PRINT UNIT @?
BEQ 10$ ;BR IF NOT.
PRINTF @PUNIT,UNITN ;PRINT THE UNIT @
MOV UNITN,-(SP)
MOV @PUNIT,-(SP)
MOV @2,-(SP)
MOV SP,RO
TRAP C:PNTF
ADD @6,SP

10$: CLR NODEV ;ADDRESS OF FIRST REGISTER
MOV CSRADDR,R1 ;START OF REGISTERS
MOV R1,R2 ;ADDRESS OF TSSR REGISTER
ADD @TSSR,R2 ;TEST BOTH CONTROLLER REGISTERS...
JSR PC,XNXM ;...AND BR IF ALL OK.
BCC 2$ ;...AND BR IF ALL OK.
MOV R1,NODEV ;FLAG DEVICE AS NON-EXISTENT
MOV @-1,DUFLG ;DROP THIS UNIT.

2$:
;FINALLY, SET CPU PRIORITY AND WE'RE DONE.

```

```

150
151 022134          5$:  SETPRI @PRI00          ;ENABLE INTERRUPTS.
      022134 012700 000000
      022140 104441
152 022142          L10030: TRAP C$SPRI
      022142
      022142 104411          TRAP C$INIT
153
154 022144          045 116 045 PUNIT: .ASCIZ /#N#N#A***** TESTING UNIT #D#A *****/
155          .EVEN
156
157          .SBTTL ADD AND DROP UNITS SECTIONS
158
159          ;**
160          ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
161          ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
162          ; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
163          ;--
164 022212          BGNAU
      022212          L$AU::
165 022212 010001          MOV RO,R1          ; GET UNIT TO BE ADDED (RO)
166 022214 006301          ASL R1          ; MAKE IT A WORD INDEX
167 022216 052761 100000 003166'  BIS #100000,ERTABL(R1)  ; SET THE "ACTIVE" BIT
168 022224 042761 040000 003166'  BIC #40000,ERTABL(R1)  ; CLEAR THE "DROPPED" BIT
169 022232          PRINTF #1$,RO
      022232 010046          MOV RO,-(SP)
      022234 012746 022260'        MOV #1,-(SP)
      022240 012746 000002        MOV #2,-(SP)
      022244 010600          MOV SP,RO
      022246 104417          TRAP C$PNTF
      022250 062706 000006        ADD #6,SP
170 022254          EXIT AU
      022254 000167          .WORD J$JMP
      022256 000026          .WORD L10031-2-
171 022260          045 116 045 1$: .ASCIZ /#N#A UNIT #D#A ADDED/
172          .EVEN
173
174 022306          ENDAU          ; UNUSED.
      022306          L10031:
      022306 104452          TRAP C$AU
175
176          ;**
177          ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
178          ; TO BE REMOVED FROM THE TEST LIST.
179          ;
180          ; SUPVSR DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN.
181          ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
182          ; COMMAND, OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
183          ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
184          ; WHICH ARE STILL ACTIVE.
185          ; UPON ENTRY, RO CONTAINS THE UNIT TO BE DROPPED.
186 022310          BGNDU
      022310          L$DU::
187 022310 012737 177777 003102'  MOV #-1,DUFLG
188 022316 010001          MOV RO,R1
189 022320 006301          ASL R1
190 022322 052761 140000 003166'  BIS #140000,ERTABL(R1)  ; SAY DROPPED

```

```

191 022330 000240 000240 000240      240,240,240      ; ??????????
192 022336      010046      PRINTF  #1$,R0
      022340 012746 022364'      MOV    RO,-(SP)
      022344 012746 000002      MOV    #1$,-(SP)
      022350 010600      MOV    #2,-(SP)
      022352 104417      MOV    SP,R0
      022354 062706 000006      TRAP  C$PNTF
193 022360      000167      ADD    #6,SP
      022362 000030      EXIT  DU
194 022364      045      116      045 1$: .WORD  J$JMP
      .WORD  L10032-2-
195      .ASCIZ /%N%A UNIT %D%A DROPPED/
196 022414      .EVEN
      022414      ENDDU
      022414 104453      L10032: TRAP  C$DU
197      ;**
198      ; AUTO-DROP CODE SECTION.
199      ;--
200 022416      BGNAUTO
      022416      L$AUTO::
201 022416 013705 002176'      MOV    CSRADDR,R5      ;POINT TO DEVICE REGISTER
202 022422 012703 000550      MOV    #360.,R3      ;ENOUGH TIME FOR 2400' REEL TO REWIND
203 022426 004737 016150'      10$: JSR    PC,WAITF      ;WAIT FOR SSR TO SET
204 022432 103420      BCS    20$      ;LEAVE WHEN SSR IS SET
205 022434      DELAY  250.      ;WAIT FOR .25 SECONDS
      022434 012727 000372      MOV    #250.,(PC)+
      022440 000000      .WORD  0
      022442 013727 002116'      MOV    L$DLY,(PC)+
      022446 000000      .WORD  0
      022450 005367 177772      DEC    -6(PC)
      022454 001375      BNE    -4
      022456 005367 177756      DEC    -22(PC)
      022462 001367      BNE    -20
206 022464 005303      DEC    R3      ;BUMP COUNTER DOWN
207 022466 001357      BNE    10$      ;KEEP GOING
208 022470 004737 017104'      JSR    PC,CKDROP      ;TRY AND DROP UNIT
209 022474
210 022474      20$: ENDAUTO      ; UNUSED.
      022474      L10033:
      022474 104461      TRAP  C$AUTO
211
212
213
214
215      ;**
216      ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
217      ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
218      ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
219      ;--
219 022476      BGNCLN
      022476      L$CLEAN::
220 022476 013705 002176'      MOV    CSRADDR,R5      ;POINT TO DEVICE REGISTER
221 022502 005737 003102'      TST    DUFLG      ;"DROPPED" FLAG IS SET ON...
222 022506 100405      BMI    1$      ;...AND GROSS CONTROLLER FAULT...
223      ;...DON'T TRY TO XCT CLEANUP CODE.
224
225 022510 012765 000000 000002      MOV    #0,TSSR(R5)      ;DO SOFT INIT

```



```

226 022516 004737 016150'      JSR    PC,WAITF
227 022522                      1$:
228 022522                      2$: ENDCLN
    022522                      L10034:
    022522 104412                TRAP    C$CLEAN

229                               ;**
230                               ; THE REPORT CODING SECTION CONTAINS THE
231                               ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
232                               ;--
233 022524                      BGNRPT
    022524                      L$RPT::
234 022524                      PRINTS  #DEVSUM
    022524 012746 022766'      MOV    #DEVSUM,-(SP)
    022530 012746 000001      MOV    #1,-(SP)
    022534 010600              MOV    SP,R0
    022536 104416              TRAP    C$PNTS
    022540 062706 000004      ADD    #4,SP
235 022544 010246              MOV    R2,-(SP)
236 022546 010346              MOV    R3,-(SP)
237 022550 010446              MOV    R4,-(SP)
238 022552 012704 003166'      MOV    #ERTABL,R4          ; GET START OF ERROR TABLE.
239 022556 005003              CLR    R3                  ; CLEAR UNIT NUMBER
240 022560 011402              1$: MOV    (R4),R2            ; GET ERROR TABLE ENTRY & TEST IT.
241 022562 001467              BEQ    4$                  ; ZERO IF UNIT NOT RUN
242 022564 100066              BPL    4$
243 022566 032702 040000      BIT    #BIT14,R2          ; WAS UNIT DROPPED?
244 022572 001015              BNE    2$                  ; BR IF YES
245 022574 042702 170000      BIC    #C7777,R2         ; GET ERROR COUNT FIELD
246 022600                      PRINTS  #DEVONL,R3,R2      ; PRINT
    022600 010246              MOV    R2,-(SP)
    022602 010346              MOV    R3,-(SP)
    022604 012746 023023'      MOV    #DEVONL,-(SP)
    022610 012746 000003      MOV    #3,-(SP)
    022614 010600              MOV    SP,R0
    022616 104416              TRAP    C$PNTS
    022620 062706 000010      ADD    #10,SP
247 022624 000446              BR     4$
248 022626 020227 160000      2$: CMP    R2,#160000        ; WAS UNIT NON-EXISTENT?
249 022632 001012              BNE    3$                  ; BR IF NO
250 022634                      PRINTS  #DEVNXR,R3
    022634 010346              MOV    R3,-(SP)
    022636 012746 023073'      MOV    #DEVNXR,-(SP)
    022642 012746 000002      MOV    #2,-(SP)
    022646 010600              MOV    SP,R0
    022650 104416              TRAP    C$PNTS
    022652 062706 000006      ADD    #6,SP
251 022656 000431              BR     4$
252 022660 020227 160001      3$: CMP    R2,#160001        ; WAS UNIT NOT READY AT STARTUP?
253 022664 001012              BNE    30$                 ; BR IF NO.
254 022666                      PRINTS  #DEVNRD,R3
    022666 010346              MOV    R3,-(SP)
    022670 012746 023155'      MOV    #DEVNRD,-(SP)
    022674 012746 000002      MOV    #2,-(SP)
    022700 010600              MOV    SP,R0
    022702 104416              TRAP    C$PNTS
    022704 062706 000006      ADD    #6,SP
255 022710 000414              BR     4$

```

```

256 022712 042702 170000      30$: BIC      #C7777,R2
257 022716 022716 010246      PRINTS #DEVDR0,R3,R2
      022720 010346      MOV      R2,-(SP)
      022722 012746 023236'  MOV      R3,-(SP)
      022726 012746 000003'  MOV      #DEVDR0,-(SP)
      022732 010600      MOV      #3,-(SP)
      022734 104416      MOV      SP,R0
      022736 062706 000010'  TRAP    C$PNTS
258 022742 062704 000002'  4$: ADD     #10,SP
259 022746 005203      INC     R3
260 022750 020427 003366'  CMP     R4,#ERTABE
261 022754 103701      BLO    1$
262 022756 012604      MOV    (SP)+,R4
263 022760 012603      MOV    (SP)+,R3
264 022762 012602      MOV    (SP)+,R2
265 022764      ENDRPT ; UNUSED.
      022764      L10035:
      022764 104425      TRAP    C$RPT

```

```

266
267
268 022766      045      116      045  DEVSUM: .ASCIZ /#N#ADEVICE STATUS SUMMARY:#N/
269 023023      045      101      040  DEVONL: .ASCIZ /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
270 023073      045      101      040  DEVNXR: .ASCIZ /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
271 023155      045      101      040  DEVNRD: .ASCIZ /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
272 023236      045      101      040  DEVDR0: .ASCIZ /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
273      .EVEN
274
275 023306      ENDMOD
276
277
278

```

```

1          .TITLE  TSV7 - HARDWARE TESTS 1-8
2
9
10 023306      BGNMOD  TSV7
11 023306      TSV7::
16
24
25          .SBTTL  TEST  1: INITIALIZE #4 TEST
26          ;+
27          ;
28          ;THIS TEST VERIFIES THAT WRITING INTO THE TSSR RETURNS THE
29          ;CONTROLLER TO ITS INITIALIZED STATE FROM VARIOUS CONDITIONS
30          ;(I.E. LOOPBACK ENABLED, FORCING WRONG PARITY, INVERTING SENSE OF
31          ;EXTENDED FEATURES SWITCH, ETC.)
32          ;
33          ;-
34 023306      BGNTST
35 023306      012737  006166'  002170'      MOV      #EPRT1,EPRTSW      ;SET UP PRIMARY ERROR MESSAGE
36
37          ;
38          ;TEST 1
39          ;
40          ;
41          ;-
42
43
48 023314      004737  016104'      JSR      PC,DSBINT      ;DISABLE INTERRUPTS
49 023320      012700  024244'      MOV      #TST21ID,R0    ;ASCII MESSAGE TO IDENTIFY TEST
50 023324      004737  016412'      JSR      PC,TSTSETUP   ;DO INITIAL TEST SETUP
51 023330      012737  000005  002206'    MOV      #5,LOOPCNT    ;PERFORM 5 ITERATIONS
52 023336      T21LOOP:
53 023336      004737  024266'      JSR      PC,T21REST    ;SET COMMAND PACKET
54 023342      004737  024356'      JSR      PC,T21RT2    ;SET UP OTHER COMMAND PACKET
55
56          ;*****
57          ;
58          ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
59          ;
60          ;*****
61
62 023346      012737  176750  023722'    MOV      #65000,,T21DLY ;SET DELAY ROUTINE
63 023354      004737  015674'    11$: JSR      PC,SOFINIT    ;DO INITIALIZE ON CONTROLLER
64 023360      103426      BCS     20$            ;BR IF INIT WAS OK
65 023362      DELAY     250            ;DELAY FOR A REWIND TO FINISH
66          MOV      #250,(PC)+
67          .WORD   0
68          MOV      L$DLY,(PC)+
69          .WORD   0
70          DEC     -6(PC)
71          BNE     .-4
72          DEC     -22(PC)
73          BNE     .-20
74 023412      005337  023722'    DEC     T21DLY        ;BUMP COUNTER DOWN
75 023416      001356      BNE     11$           ;BR, IF MORE TIME TO GO
76 023420      005237  002212'    INC     FATFLG        ;BUMP COUNT
77 023424      010001      MOV     R0,R1         ;CONTENTS OF TSSR REGISTER

```

```

73 023426          ERRDF  ERRNO,SFIERR,SFIMSG  ;FATAL ERROR TSSR WAS NOT OK
    023426 104455          TRAP  C$ERDF
    023430 000145          .WORD 101
    023432 003642'        .WORD SFIERR
    023434 011734'        .WORD SFIMSG
74 023436          20$:
75 023436 012704 023700'  MOV  #T21PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
76
77 ;*****
78 ;
79 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
80 ;
81 ;*****
82
83 023442 013737 002172' 023720'  MOV  UNITN,T21DSW          ;SET UP DRIVE NUMBER
84 023450 004737 010562'  JSR  PC,WRTCHR            ;ISSUE WRITE CHARACTERISTICS
85 023454 103407          BCS  23$                 ;BR, IF COMMAND ISSUED OK
86 023456 005237 002212'  INC  FATFLG              ;BUMP COUNT
90 023462 010001          MOV  R0,R1                ;SAVE CONTENTS OF TSSR
91 023464          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
    023464 104456          TRAP  C$ERHRD
    023466 000146          .WORD 102
    023470 005046'        .WORD WRTMSG
    023472 011734'        .WORD SFIMSG
92 023474          23$:  CKLOOP
    023474 104406          TRAP  C$CLP1
93 023476 112737 000200 024020'  MOVB #200,T21BS0          ;WRITE MISCELLANEOUS CONT/READ STATUS
94 023504 112737 000010 024021'  MOVB #10,T21BS1         ;FUNCTION SELECTION BIT
95 023512          25$:
96 023512 012704 024010'  MOV  #T21PK2,R4          ;WRITE SUBSYS MEM PACKET
97 023516 010465 000000  MOV  R4,TSDB(R5)         ;ISSUE COMMAND
98 023522 004737 016236'  JSR  PC,CHKTSSR         ;WAIT FOR SSR
99 023526 103407          BCS  30$                 ;BR, IF NO ERROR
100 023530 010001          MOV  R0,R1                ;ERROR, SAVE TSSR
101 023532 005237 002212'  INC  FATFLG              ;BUMP COUNT
105 023536          ERRHRD  ERRNO,T21SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
    023536 104456          TRAP  C$ERHRD
    023540 000147          .WORD 103
    023542 024026'        .WORD T21SSR
    023544 011746'        .WORD PKTSSR
106 023546          30$:  CKLOOP          ;SCOPE LOOP
    023546 104406          TRAP  C$CLP1
107 023550 012765 000000 000002  MOV  #0,TSSR(R5)         ;ISSUE A SOFT INITIALIZE
108 023556 004737 016150'  JSR  PC,WAITF           ;WAIT FOR JUST THE SSR BIT TO SET
109 023562 016501 000002  MOV  TSSR(R5),R1        ;READ THE TSSR BACK
110 023566 010102          MOV  R1,R2                ;WORK REGISTER
111 023570 042702 176377  BIC  #+C<HIADDR>,R2     ;CLEAR OUT OTHER BITS
112 023574 052702 002200  BIS  #SSR!NBA,R2        ;SOME OF THE BITS THAT SHOULD BE SET
113 023600 032701 000100  BIT  #OFL,R1           ;IS OFF LINE BIT SET
114 023604 001012          BNE  38$                 ;BR, IF DRIVE IS OFF LINE
115 023606 020102          35$:  CMP  R1,R2                ;EXPECTED (R2) = RECEIVED (R1)
116 023610 001406          BEQ  37$                 ;BR, IF THEY ARE EQUAL (OK)
117 023612 005237 002212'  INC  FATFLG              ;BUMP COUNT
121 023616          ERRHRD  ERRNO,T21AM3,EXPREC ;"ERROR TRYING TO INIT AFTER WRITE MISC.
    023616 104456          TRAP  C$ERHRD
    023620 000150          .WORD 104
    023622 024123'        .WORD T21AM3

```

```

122 023624 015374'
023626 104406
123 023630 000406
124 023632
128 023632
023632 104455
023634 000151
023636 024223'
023640 015374'
129 023642 004737 017104'
130 023646 000241
131 023650 106037 024021'
132 023654 001316
133 023656
023656 104406
134 023660 004737 016360'
135 023664 103002
136 023666 000137 023336'
137 023672
023672 104432
023674 000524
138
139
140
141
143 023676
145 023700
146 023700 100004
147 023702 023710'
148 023704 000000
149 023706 000012
150 023710
151 023710 023724'
152 023712 000000
153 023714 000024
154 023716 000000
155 023720 000000
156 023722 000000
157 023724
158
159
160
162 024006
164 024010
165 024010 100206
166 024012 024020'
167 024014 000000
168 024016 000006
169
170
171 024020
172 024020 000
173 024021 000
174 024022 000000
175 024024 000000
176

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

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

;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
.BLKB 10-<.-TSV2&7>
T21PK2:
.WORD 100206 ;WRITE SUB SYS MEM COMMAND, IE AND ACK
.WORD T21BF2 ;ADDRESS OF SELECT BLOCK DATA
.WORD 0
.WORD 6. ;SIZE OF DATA PACKET

.EVEN
T21BF2:
T21BS0: .BYTE 0 ;BSELO AREA --- "COMMAND" BYTE
T21BS1: .BYTE 0 ;BSEL1 AREA
T21S2: .WORD 0 ;SEL 2 AREA
T21S3: .WORD 0 ;DATA AREA

```

```

177
178
179      ;*
180      ;LOCAL TEXT MESSAGES FOR TEST
181      ;-
182 024026      127      122      111 T21SSR: .ASCIZ 'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
183 024123      124      123      123 T21AM3: .ASCIZ 'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
184 024223      104      162      151 T21OFL: .ASCIZ 'Drive is OFFLINE'
185 024244      111      156      151 TST21ID: .ASCIZ 'Initialization #4'
186
187
188      ;*
189      ;
190      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
191      ;WRITE SUBSYSTEM MEMORY COMMAND
192      ;
193      ;-
194 024266      T21REST:
195 024266      SAVREG      ;SAVE THE REGISTERS
196 024272      012701      023700'      MOV      @T21PACKET,R1      ;START OF THE PACKET
197 024276      012721      100004      MOV      @100004,(R1).      ;WRITE SUBSYSTEM MEM. WITH ACK.
198 024302      012721      023710'      MOV      @T21DATA,(R1).      ;ADDRESS OF CHARAISTICS DATA BLOCK
199 024306      005021      CLR      (R1).      ;EXTENDED ADDRESS
200 024310      012721      000010      MOV      @8.,(R1).      ;SIZE OF DATA BLOCK IN BYTES
201 024314      012721      023724'      MOV      @T21BFR,(R1).      ;ADDRESS OF MESSAGE BUFFER
202 024320      005021      CLR      (R1).
203 024322      012721      000024      MOV      @20.,(R1).      ;LENGTH OF MESSAGE BUFFER
204 024326      005021      CLR      (R1).
205 024330      005011      CLR      (R1)
206 024332      012702      000020      MOV      @20,R2      ;NUMBER OF LOCATIONS TO BE CLEARED
207 024336      012762      177777      023724' 64$:      MOV      @177777,T21BFR(R2)      ;ALL ONES TO MESSAGE BUFFER
208 024344      005742      TST      -(R2)      ;NEXT LOCATION
209 024346      020227      000000      CMP      R2,#0      ;CHECK R2 FOR ZERO
210 024352      001371      BNE      64$      ;BR. IF NOT AT ZERO YET
211 024354      000207      RTS      PC      ;RETURN
212
213
214 024356      T21RT2:
215 024356      SAVREG      ;SAVE THE REGISTERS
216 024362      012701      024010'      MOV      @T21PK2,R1      ;START OF THE PACKET
217 024366      012721      100206      MOV      @100206,(R1).      ;WRITE SUBSYSTEM MEM. WITH ACK. IE
218 024372      012721      024020'      MOV      @T21BF2,(R1).      ;ADDRESS OF DATA BLOCK
219 024376      005021      CLR      (R1).      ;EXTENDED ADDRESS
220 024400      012721      000006      MOV      @6.,(R1).      ;SIZE OF DATA BLOCK IN BYTES
221 024404      005021      CLR      (R1).
222 024406      012701      024020'      MOV      @T21BF2,R1      ;ADDRESS OF DATA FOR WRT SUB SYS MEM
223 024412      005021      CLR      (R1).
224 024414      005011      CLR      (R1)
225 024416      000207      RTS      PC      ;RETURN
226 024420      ENDTST
227
228
229      ;*
230      ;
231      ;THIS TEST VERIFIES BASIC TAPE-MOTION COMMAND DECODING AND BASIC

```

L10036: TRAP C#ETST

.SBTTL TEST 2: OFF-LINE AND REJECT REWIND


```

289
290
291
292 024514 004737 010562'      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
293 024520 103407              BCS      23$           ;BR, IF COMMAND ISSUED OK
294 024522 005237 002212'      INC      FATFLG        ;BUMP COUNT
298 024526 010001              MOV      RO,R1         ;SAVE CONTENTS OF TSSR
299 024530              ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
      024530 104456              TRAP     C$ERHRD
      024532 000312              .WORD   202
      024534 005046'              .WORD   WRTMSG
      024536 011734'              .WORD   SFIMSG
300 024540              23$:   CKLOOP              TRAP     C$CLP1
      024540 104406
301 024542 013701 026020'      MOV      T22BFR+6,R1   ;PICK UP XT50
302 024546 032701 000004'      BIT      #4,R1        ;IS UNIT WRITE-LOCKED?
303 024552 001407              BEQ      24$           ;NO,PROCEED WITH TESTING
304 024554 005237 002212'      INC      FATFLG        ;BUMP COUNT
308 024560              ERRDF  ERRNO,T22WLK,SFIMSG ;TAPE IS WRITE LOCKED
      024560 104455              TRAP     C$ERDF
      024562 000313              .WORD   203
      024564 026622'              .WORD   T22WLK
      024566 011734'              .WORD   SFIMSG
309 024570              DOCLN              TRAP     C$DCLN
      024570 104444
310 024572              24$:   CKLOOP              TRAP     C$CLP1
      024572 104406
311 024574 005737 002216'      TST      EXTFEA        ;CHECK FOR EXTENDED FEATURES SW SWITCH
312 024600 001041              BNE      50$           ;BR IF SWITCH IS ON
313 024602 112737 000200 026111'  MOVB     #200,T22BS1   ;WRITE MISCELLANEOUS CONT/READ STATUS
314 024610 112737 000010 026110'  MOVB     #10,T22BS0   ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
315 024616 012704 026100'      MOV      #T22PK2,R4   ;WRITE SUBSYS MEM PACKET
316 024622 010465 000000'      MOV      R4,T5DB(R5)  ;ISSUE COMMAND
317 024626 004737 016236'      JSR      PC,CHKTSSR   ;WAIT FOR SSR
318 024632 103407              BCS      30$           ;BR, IF NO ERROR
319 024634 010001              MOV      RO,R1         ;ERROR, SAVE TSSR
320 024636 005237 002212'      INC      FATFLG        ;BUMP COUNT
324 024642              ERRHRD  ERRNO,T22SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      024642 104456              TRAP     C$ERHRD
      024644 000314              .WORD   204
      024646 026130'              .WORD   T22SSR
      024650 011746'              .WORD   PKTSSR
325 024652              30$:   CKLOOP              ;LOOP IF SELECTED
      024652 104406              TRAP     C$CLP1
326 024654 012704 025770'      MOV      #T22PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
327
328
329
330
331
332
333
334 024660 004737 010562'      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
335 024664 103407              BCS      50$           ;BR, IF COMMAND ISSUED OK
336 024666 005237 002212'      INC      FATFLG        ;BUMP COUNT
340 024672 010001              MOV      RO,R1         ;SAVE CONTENTS OF TSSR
341 024674              ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED

```



```

488
489
490
491
492
493
494 025366          ;
      025366          ; VERIFIES THAT A REWIND COMMAND WITH CVC=1 CLEARS VCK
      025366 104402   ; AND RETURNS PROPER STATUS IN THE MESSAGE BUFFER.
495 025370 004737 026742' ;
496 025374 004737 027034' ;
497
498
499
500
501
502
503
504 025400 004737 015674' ;
      025404 103407          ;
505 025404 103407          ;
506 025406 005237 002212' ;
510 025412 010001          ;
511 025414          ;
      025414 104455          ;
      025416 000326          ;
      025420 003642'        ;
      025422 011734'        ;
512 025424          ;
513 025424 012704 025770' 20$:
      MOV          @T22PACKET,R4          ; SUBROUTINE NEEDS PACKET ADDRESS
514
515
516
517
518
519
520
521 025430 004737 010562' ;
      025434 103407          ;
522 025434 103407          ;
523 025436 005237 002212' ;
527 025442 010001          ;
528 025444          ;
      025444 104456          ;
      025446 000327          ;
      025450 005046'        ;
      025452 011734'        ;
529 025454 005737 002216' 23$:
530 025460 001041          ;
531
532 025462 112737 000200 026111' ;
533 025470 112737 000010 026110' ;
534 025476 012704 026100' ;
535 025502 010465 000000          ;
536 025506 004737 016236' ;
537 025512 103407          ;
538 025514 010001          ;
539 025516 005237 002212' ;
543 025522          ;
      ERRHRD ERRNO,T22SSR,PKTSSR          ; TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
    
```

```

025522 104456
025524 000330
025526 026130'
025530 011746'
544 025532 30$: CKLOOP ;LOOP IF SELECTED
025532 104406 TRAP C$ERHRD
545 025534 012704 025770' MOV #T22PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
546
547
548 ;*****
549 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
550 ;
551 ;*****
552
553 025540 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
554 025544 103407 BCS 50$ ;BR, IF COMMAND ISSUED OK
555 025546 005237 002212' INC FATFLG ;BUMP COUNT
559 025552 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
560 025554 ERHRD ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTIC FAILED
025554 104456 TRAP C$ERHRD
025556 000331 .WORD 217
025560 005046' .WORD WRTMSG
025562 011734' .WORD SFMSG
561 025564 50$: CKLOOP ;SCCPE LOOP
025564 104406 TRAP C$CLP1
562 025566 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
563 025572 032701 000100 BIT #OFL,R1 ;CHECK FOR THE OFFLINE BIT SET
564 025576 001006 BNE 60$ ;BR, IF OFFLINE (GOOD)
565 025600 005237 002212' INC FATFLG ;BUMP COUNT
569 025604 ERRDF ERRNO,T22OFL,SFMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
025604 104455 TRAP C$ERDF
025606 000332 .WORD 218
025610 026325' .WORD T22OFL
025612 011734' .WORD SFMSG
570 025614 60$: CKLOOP ;LOOP IF SELECTED
025614 104406 TRAP C$CLP1
571 025616 012737 142010 026100' 65$: MOV #142010,T22PK2 ;POSITION COMMAND (REWIND MODE) CVC=1
572 025624 012704 026100' MOV #T22PK2,R4 ;R4 = POINTER TO PACKET
573 025630 010465 000000 MOV R4,TSD8(R5) ;ISSUE COMMAND
574 025634 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
575 025640 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
576 025644 012702 100306 MOV #SSR!OFL!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
577 025650 020102 CMP R1,R2 ;ARE THEY EQUAL
578 025652 001406 BEQ 80$ ;BR, IF OK ESP. FUNCTION REJECT
579 025654 005237 002212' INC FATFLG ;BUMP COUNT
583 025660 ERRHRD ERRNO,T22RWJ,EXPREC ;TSSR INCORRECT AFTER TAPE MOTION CMD
025660 104456 TRAP C$ERHRD
025662 000333 .WORD 219
025664 026474' .WORD T22RWJ
025666 015374' .WORD EXPREC
584 025670 80$: CKLOOP ;LOOP IF SELECTED
025670 104406 TRAP C$CLP1
585 025672 012703 026012' MOV #T22BFR,R3 ;POINTER TO MESSAGE BUFFER
586 025676 016301 000006 MOV XSTO(R3),R1 ;PICK UP XSTO FROM MESSAGE BUFFER
587 025702 010102 MOV R1,R2 ;SET UP EXPECTED
588 025704 042702 000020 BIC #BIT4,R2 ;VCK SHOULD BE CLEAR
589 025710 020102 CMP R1,R2 ;ARE THEY EQUAL

```



```

646                                     ;TAPE MOTION PACKET COMMAND VALUES
647 026116 100201                      T22RD: .WORD 100201                      ;READ TAPE FORWARD
648 026120 100205                      T22WRT: .WORD 100205                   ;WRITE TAPE FORWARD
649 026122 100210                      T22POS: .WORD 100210                   ;POSITION TAPE
650 026124 100211                      T22FOR: .WORD 100211                   ;FORMAT TAPE
651 026126 177777                      .WORD 177777                          ;END OF DATA
652
653
654                                     ;*
655                                     ;LOCAL TEXT MESSAGES FOR TEST
656                                     ;-
657
658 026130 127 122 111 T22SSR: .ASCIZ 'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
659 026225 124 123 123 T22AM3: .ASCIZ 'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
660 026325 104 162 151 T22OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
661 026400 124 123 123 T22TM: .ASCIZ 'TSSR Incorrect After Tape Motion Command To Off-Line Device'
662 026474 124 123 123 T22RWJ: .ASCIZ 'TSSR Not Correct After REWIND With VCK Set'
663 026547 103 126 103 T22VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
664 026622 052 052 052 T22WLK: .ASCIZ '*****TAPE IS WRITE-LOCKED AND WILL CAUSE ERRORS*****'
665 026707 117 146 146 TST22ID: .ASCIZ 'Off-Line And Reject Rewind'
666                                     .EVEN
667
668                                     ;*
669                                     ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
670                                     ;WRITE SUBSYSTEM MEMORY COMMAND
671                                     ;-
672
673
674 026742                                T22REST:
675 026742                                SAVREG
676 026746 012701 025770'                MOV #T22PACKET,R1                      ;SAVE THE REGISTERS
677 026752 012721 100204                MOV #100204,(R1)+                      ;START OF THE PACKET
678 026756 012721 026000'                MOV #T22DATA,(R1)+                    ;WRITE SUBSYSTEM MEM. WITH ACK, IE
679 026762 005021                        CLR (R1)+                               ;ADDRESS OF CHARAISTICS DATA BLOCK
680 026764 012721 000012                MOV #10.,(R1)+                         ;EXTENDED ADDRESS
681 026770 012721 026012'                MOV #T22BFR,(R1)+                     ;SIZE OF DATA BLOCK IN BYTES
682 026774 005021                        CLR (R1)+                               ;ADDRESS OF MESSAGE BUFFER
683 026776 012721 000024                MOV #20.,(R1)+                         ;LENGTH OF MESSAGE BUFFER
684 027002 005021                        CLR (R1)+
685 027004 012711 000007                MOV #7,(R1)                            ;SELECT DRIVE SEVEN
686 027010 012702 000020                MOV #20,R2                             ;NUMBER OF LOCATIONS TO BE CLEARED
687 027014 012762 177777 026012' 64$:  MOV #177777,T22BFR(R2)                 ;ALL ONES TO MESSAGE BUFFER
688 027022 005742                        TST -(R2)                               ;BUMP R2 DOWN
689 027024 020227 000000                CMP R2,#0                              ;IS R2 AT ZERO YET
690 027030 001371                        BNE 64$                                 ;KEEP GOING UNTIL DONE
691 027032 000207                        RTS PC                                  ;RETURN
692
693
694 027034                                T22RT2:
695 027034                                SAVREG
696 027040 012701 026100'                MOV #T22PK2,R1                         ;SAVE THE REGISTERS
697 027044 012721 100206                MOV #100206,(R1)+                      ;START OF THE PACKET
698 027050 012721 026110'                MOV #T22BF2,(R1)+                     ;WRITE SUBSYSTEM MEM. WITH ACK, IE
699 027054 005021                        CLR (R1)+                               ;ADDRESS OF DATA BLOCK
700 027056 012721 000006                MOV #6.,(R1)+                          ;EXTENDED ADDRESS
701 027062 005021                        CLR (R1)+                               ;SIZE OF DATA BLOCK IN BYTES
702 027064 012701 026110'                MOV #T22BF2,R1                         ;POINT TO DATA SEL AREA

```



```

759 027164 103407          BCS      20$          ;BR IF INIT WAS OK
760 027166 005237 002212' INC      FATFLG      ;BUMP COUNT
764 027172 010001          MOV      R0,R1       ;CONTENTS OF TSSR REGISTER
765 027174          ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      027174 104455          TRAP    C$ERDF
      027176 000455          .WORD  301
      027200 003642'        .WORD  SFIERR
      027202 011734'        .WORD  SFIMSG
766 027204          20$:
767 027204 012737 000007 032400' MOV      #7,T23DSW    ;SET DRIVE NUMBER IN PACKET
768 027212 012704 032360' MOV      #T23PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
769
770 ;*****
771 ;
772 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
773 ;
774 ;*****
775
776 027216 004737 010562'   JSR      PC,WRTCHR    ;ISSUE WRITE CHARACTERISTICS
777 027222 103407          BCS      23$          ;BR, IF COMMAND ISSUED OK
778 027224 005237 002212' INC      FATFLG      ;BUMP COUNT
782 027230 010001          MOV      R0,R1       ;SAVE CONTENTS OF TSSR
783 027232          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      027232 104456          TRAP    C$ERHRD
      027234 000456          .WORD  302
      027236 005046'        .WORD  WRTMSG
      027240 011734'        .WORD  SFIMSG
784 027242 005737 002216'   23$:  TST      EXTFEA    ;CHECK FOR EXTENDED FEATURES SW SWITCH
785 027246 001044          BNE      50$          ;BR IF SWITCH IS ON
786
787 027250 112737 000200 032523' MOVB     #200,T23BS1  ;WRITE MISCELLANEOUS CONT/READ STATUS
788 027256 112737 000010 032522' MOVB     #10,T23BS0   ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
789 027264 012704 032470' MOV      #T23PK2,R4  ;WRITE SUBSYS MEM PACKET
790 027270 010465 000000   MOV      R4,TSDB(R5) ;ISSUE COMMAND
791 027274 004737 016236'   JSR      PC,CHKTSSR  ;WAIT FOR SSR
792 027300 103407          BCS      30$          ;BR, IF NO ERROR
793 027302 010001          MOV      R0,R1       ;ERROR, SAVE TSSR
794 027304 005237 002212'   INC      FATFLG      ;BUMP COUNT
798 027310          ERRHRD  ERRNO,T23SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      027310 104456          TRAP    C$ERHRD
      027312 000457          .WORD  303
      027314 032544'        .WORD  T23SSR
      027316 011746'        .WORD  PKTSSR
799 027320          30$:  CKLOOP          ;LOOP IF SELECTED
      027320 104406          TRAP    C$CLP1
800 027322 012737 000007 032400' MOV      #7,T23DSW    ;SET DRIVE NUMBER IN PACKET
801 027330 012704 032360' MOV      #T23PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
802
803 ;*****
804 ;
805 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
806 ;
807 ;*****
808
809 027334 004737 010562'   JSR      PC,WRTCHR    ;ISSUE WRITE CHARACTERISTICS
810 027340 103407          BCS      50$          ;BR, IF COMMAND ISSUED OK
811 027342 005237 002212'   INC      FATFLG      ;BUMP COUNT

```



```

914 027634      238:  CKLOOP                ;LOOP IF SELECTED
      027634 104406                                TRAP      C8CLP1
915
916 ;.....
917 ;
918 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
919 ;
920 ;.....
921
922 027636 004737 010714'      JSR      PC,REWIND      ;CALL THE TAPE REWIND
923 027642 012703 000024'      MOV      @20.,R3        ;STARTING RECORD SIZE
924 027646 013737 003114' 032512' 658:  MOV      FREE,T23WB     ;STARTING WRITE BUFFER ADDRESS
925
926 ;.....
927 ;
928 ;WRITE DATA,CVC=1,ACK COMMAND
929 ;
930 ;.....
931
932 027654 012737 140005 032510'      MOV      @140005,T23PK3 ;WRITE DATA,CVC=1,ACK COMMAND
933 027662 012737 140005 032532'      MOV      @140005,T23WRT ;SETUP FOR RETRY COMMAND
934 027670 052737 004000 032532'      BIS      @4000,T23WRT   ;MAKE IT A RETRY
935 027676 012704 032510'      MOV      @T23PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
936 027702 010300      MOV      R3,R0         ;SET PATTERN IN CORRECT REGISTER
937 027704 004737 017324'      JSR      PC,FILLMEM    ;FILL MEMORY WITH RECORD SIZE
938 027710 010337 032516'      MOV      R3,T23SZ      ;SET UP RECORD SIZE IN PACKET
939 027714 010465 000000      MOV      R4,T23DB(R5)  ;ISSUE COMMAND
940 027720 004737 016150'      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
941 027724 016501 000002      MOV      T23SR(R5),R1  ;GET T23SR CONTENTS
942 027730 012702 000200      MOV      @23SR,R2      ;SET UP EXPECTED
943 027734 020102      CMP      R1,R2         ;ARE THEY EQUAL
944 027736 001402      BEQ      808           ;BR, IF OK
945 027740 004737 034062'      JSR      PC,T23CHK     ;CHECK SPECIAL CONDITION
946 027744      808:  CKLOOP                ;LOOP IF SELECTED
      027744 104406                                TRAP      C8CLP1
947 027746 016501 000000      MOV      TSBA(R5),R1   ;GET TSBA CONTENTS
948 027752 012702 032402'      MOV      @T23BFR,R2   ;SET UP EXPECTED
949 027756 062702 000016      ADD      @16,R2        ;SET TO END OF MESSAGE BUFFER
950 027762 005737 002216'      TST      EXTFEA       ;CHECK FOR EXTENDED FEATURES SW SET
951 027766 001402      BEQ      858           ;BR, IF IT NOT SET
952 027770 062702 000002      ADD      @2,R2         ;BUMP R2 FOR EXTRA DATA
953 027774 020102      858:  CMP      R1,R2        ;ARE THEY EQUAL
954 027776 001406      BEQ      908           ;BR, IF TSBA IS CORRECT
955 030000 005237 002212'      INC      FATFLG       ;BUMP COUNT
959 030004      ERRHRD  ERRNO,T23BA,EXPREC ;TSBA WAS NOT CORRECT AFTER WRITE DATA
      030004 104456                                TRAP      C8ERHRD
      030006 000465                                .WORD    309
      030010 033525'                                .WORD    T23BA
      030012 015374'                                .WORD    EXPREC
960 030014      908:  CKLOOP                ;LOOP IF SELECTED
      030014 104406                                TRAP      C8CLP1
961 030016 020327 007376      CMP      R3,@7376     ;ONLY CHECK RAM UNTIL ITS FULL
962 030022 002114      BGE      1158         ;IT WRAPS AROUND ETC.
963 030024 004737 033774'      JSR      PC,T23RT2    ;MAKE SURE PACKET AND DATA ARE CLEAN
964 030030 012737 000400 032524'      MOV      @256.,T2352  ;STARTING RAM ADDRESS
965 030036 112737 000000 032522'      MOV      @0,T23B50    ;STOP INTERNAL TSV05 DIAGNOSTICS
966 030044 112737 000000 032523'      MOV      @0,T23B51    ;SIZE OF RAM READ

```



```

1069 030426 013737 002172' 032400'      MOV      UNITN,T23DSW      ;SET UP UNIT NUMBER
1070 030434 012704 032360'              MOV      @T23PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
1071
1072      ;*****
1073      ;
1074      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
1075      ;
1076      ;*****
1077
1078 030440 004737 010562'      JSR      PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
1079 030444 103407              BCS     23$              ;BR, IF COMMAND ISSUED OK
1080 030446 005237 002212'      INC     FATFLG          ;BUMP COUNT
1084 030452 010001              MOV     R0,R1           ;SAVE CONTENTS OF TSSR
1085 030454              ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP     C$ERHRD
                                .WORD    315
                                .WORD    WRTMSG
                                .WORD    SFIMSG
                                030454 104456
                                030456 000473
                                030460 005046'
                                030462 011734'
1086 030464              23$:
1087 030464 012703 000024      MOV     @20.,R3         ;STARTING RECORD SIZE
1088 030470 013737 003114' 032512' 65$:      MOV     FREE,T23WB      ;STARTING WRITE BUFFER ADDRESS
1089
1090      ;*****
1091      ;
1092      ;WRITE DATA,CVC=1,ACK,SWB COMMAND
1093      ;
1094      ;*****
1095
1096 030476 012737 150005 032510'      MOV     @150005,T23PK3   ;WRITE DATA,CVC=1,ACK,SWB COMMAND
1097 030504 012737 150005 032532'      MOV     @150005,T23WRT   ;SETUP FOR RETRY COMMAND
1098 030512 052737 004000 032532'      BIS     @4000,T23WRT     ;MAKE IT A RETRY
1099 030520 012704 032510'      MOV     @T23PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
1100 030524 010300              MOV     R3,R0           ;SET PATTERN IN CORRECT REGISTER
1101 030526 004737 017324'      JSR     PC,FILLMEM       ;FILL MEMORY WITH RECORD SIZE
1102 030532 010337 032516'      MOV     R3,T23SZ        ;SET UP RECORD SIZE IN PACKET
1103 030536 010465 000000      MOV     R4,TSDB(R5)     ;ISSUE COMMAND
1104 030542 004737 016150'      JSR     PC,WAITF        ;WAIT FOR SSR TO SET
1105 030546 016501 000002      MOV     TSSR(R5),R1     ;GET TSSR CONTENTS
1106 030552 012702 000200      MOV     @SSR,R2         ;SET UP EXPECTED
1107 030556 020102              CMP     R1,R2           ;ARE THEY EQUAL
1108 030560 001402              BEQ     80$             ;BR, IF OK
1109 030562 004737 034062'      JSR     PC,T23CHK       ;CHECK SPECIAL CONDITION
1110 030566              80$:      CKLOOP                ;LOOP IF SELECTED
                                TRAP     C$CLP1
                                030566 104406
1111 030570 016501 000000      MOV     TSBA(R5),R1     ;GET TSBA CONTENTS
1112 030574 012702 032402'      MOV     @T23BFR,R2     ;SET UP EXPECTED
1113 030600 062702 000016      ADD     @16,R2          ;SET TO END OF MESSAGE BUFFER
1114 030604 005737 002216'      TST     EXTFEA          ;CHECK FOR EXTENDED FEATURES SW SET
1115 030610 001402              BEQ     85$             ;BR, IF IT NOT SET
1116 030612 062702 000002      ADD     @2,R2           ;BUMP R2 FOR EXTRA DATA
1117 030616 020102              85$:      CMP     R1,R2           ;ARE THEY EQUAL
1118 030620 001406              BEQ     90$             ;BR, IF TSBA IS CORRECT
1119 030622 005237 002212'      INC     FATFLG          ;BUMP COUNT
1123 030626              ERRHRD  ERRNO,T23BA,EXPREC ;TSBA WAS NOT CORRECT AFTER WRITE DATA
                                TRAP     C$ERHRD
                                .WORD    316
                                .WORD    T23BA
                                030626 104456
                                030630 000474
                                030632 033525'

```

```

1124 030634 015374'          90$:  CKLOOP          ;LOOP IF SELECTED          .WORD  EXPREC
030636 104406                TRAP  C$CLP1
1125 030640 020327 007376    CMP    R3,#7376          ;ONLY CHECK RAM UNTIL ITS FULL
1126 030644 002115          BGE    115$              ;IT WRAPS AROUND ETC.
1127 030646 004737 033774'  JSR    PC,T23RT2        ;MAKE SURE PACKET AND DATA ARE CLEAN
1128 030652 012737 000400 032524'  MOV    #256.,T23S2      ;STARTING RAM ADDRESS
1129 030660 112737 000000 032522'  MOVB   #0,T23BS0        ;STOP INTERNAL TSV05 DIAGNOSTICS
1130 030666 112737 000000 032523'  MOVB   #0,T23BS1        ;SIZE OF RAM READ
1131 030674 012704 032470'  MOV    #T23PK2,R4       ;SET R4 WITH PACKET ADDRESS
1132 030700 010465 000000    MOV    R4,TSDB(R5)      ;ISSUE WRITE SUB SYS MEM COMMAND
1133 030704 004737 016236'  JSR    PC,CHKTSSR       ;CHECK TSSR AND WAIT FOR SSR TO SET
1134 030710 103407          BCS    92$              ;BR, IF NO ERRORS IN TSSR
1135 030712 010001          MOV    R0,R1            ;SAVE TSSR
1136 030714 005237 002212'  INC    FATFLG           ;BUMP COUNT
1140 030720          ERRHRD  ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
030720 104456                TRAP  C$ERHRD
030722 000475                .WORD 317
030724 033577'            .WORD T23WSS
030726 011746'            .WORD PKTSSR
1141 030730          92$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
030730 104406                ;MAKE SURE PACKET AND DATA ARE CLEAN
1142 030732 004737 033774'  JSR    PC,T23RT2        ;STARTING RAM ADDRESS
1143 030736 012737 000400 032524'  MOV    #256.,T23S2      ;READ RAM COMMAND FOR WRITE SUB SYS M.
1144 030744 112737 000001 032522'  MOVB   #1,T23BS0        ;SIZE OF RAM READ
1145 030752 112737 000002 032523'  MOVB   #2,T23BS1        ;SET R4 WITH PACKET ADDRESS
1146 030760 012704 032470'  MOV    #T23PK2,R4       ;ISSUE WRITE SUB SYS MEM CMD (READ RAM)
1147 030764 010465 000000 95$:  MOV    R4,TSDB(R5)      ;CHECK TSSR AND WAIT FOR SSR TO SET
1148 030770 004737 016236'  JSR    PC,CHKTSSR       ;BR, IF NO ERRORS IN TSSR
1149 030774 103407          BCS    100$            ;SAVE TSSR
1150 030776 010001          MOV    R0,R1            ;BUMP COUNT
1151 031000 005237 002212'  INC    FATFLG           ;TSSR BAD AFTER WRITE SUB SYS MEM
1155 031004          ERRHRD  ERRNO,T23WSS,PKTSSR ;TSSR BAD AFTER WRITE SUB SYS MEM
031004 104456                TRAP  C$ERHRD
031006 000476                .WORD 318
031010 033577'            .WORD T23WSS
031012 011746'            .WORD PKTSSR
1156 031014          100$: CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
031014 104406                ;CLEAR REGISTERS
1157 031016 005001          CLR    R1               ;CLEAR REGISTERS
1158 031020 005002          CLR    R2               ;PICK UP BYTE READ FROM RAM
1159 031022 013701 032422'  MOV    T23BFR+20,R1     ;SET UP EXPECTED
1160 031026 010302          MOV    R3,R2            ;SWAP BYTES
1161 031030 000302          SWAB   R2               ;IS RAM DATA CORRECT
1162 031032 020102          CMP    R1,R2            ;BR, IF OK (EQUAL)
1163 031034 001406          BEQ    110$            ;BUMP COUNT
1164 031036 005237 002212'  INC    FATFLG           ;RNC=RAM NOT CORRECT
1168 031042          ERRHRD  ERRNO,T23RNC,EXPREC ;TSSR BAD AFTER WRITE SUB SYS MEM
031042 104456                TRAP  C$ERHRD
031044 000477                .WORD 319
031046 033065'            .WORD T23RNC
031050 015374'            .WORD EXPREC
1169 031052          110$: CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
031052 104406                ;BUMP RAM ADDRESS TO BE CHECKED
1170 031054 005237 032524'  INC    T23S2            ;BUMP RAM ADDRESS TO BE CHECKED
1171 031060 005237 032524'  INC    T23S2            ;GET SIZE OF RECORD
1172 031064 010301          MOV    R3,R1

```



```

031534 011734'                                .WORD  SFIMSG
1330
1331 ;*****
1332 ;
1333 ;WRITE DATA, ACK, CVC=1
1334 ;
1335 ;*****
1336
1337 031536 123$:
1338 031536 005737 002216' TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
1339 031542 001026 BNE 130$ ;BR IF SWITCH IS ON
1340 031544 005237 002216' INC EXTFEA ;ONLY ONE TIME
1341 031550 112737 000200 032523' MOVB #200,T23BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
1342 031556 112737 000010 032522' MOVB #10,T23BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
1343 031564 012704 032470' MOV #T23PK2,R4 ;WRITE SUBSYS MEM PACKET
1344 031570 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
1345 031574 004737 016236' JSR PC,CHKTSSR ;WAIT FOR SSR
1346 031600 103407 BCS 130$ ;BR, IF NO ERROR
1347 031602 010001 MOV RO,R1 ;ERROR, SAVE TSSR
1348 031604 005237 002212' INC FATFLG ;BUMP COUNT
1352 031610 ERRHRD ERRNO,T22SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
    031610 104456 TRAP C$ERHRD
    031612 000506 .WORD 326
    031614 026130' .WORD T22SSR
    031616 011746' .WORD  PKTSSR
1353 031620 130$: CKLOOP ;LOOP IF SELECTED
    031620 104406 TRAP C$CLP1
1354
1355 031622 012701 160000 MOV #160000,R1 ;NXM LOW ADDRESS START
1356 031626 012702 177776 MOV #177776,R2 ;LIMIT CHECK FOR NXM (HIGHEST)
1357 031632 004737 016276' JSR PC, NXM ;LOOK FOR NXM ADDRESS
1358 031636 103045 BCC 80$ ;BR, IF NON FOUND
1359 031640 010137 003130' MOV R1,NXMLO ;SET ADDRESS UP FOR TEST
1360
1361
1362 031644 005037 032514' CLR T23WB+2 ;CLEAR OUT THE HIGH BITS AREA
1363 031650
1364 031650 012737 140005 032510' 24$: MOV #140005,T23PK3 ;WRITE DATA, ACK, CVC=1
1365 031656 013737 003130' 032512' MOV NXMLO,T23WB ;SET UP WRITE BUFFER ADDRESS
1366 031664 012737 000100 032516' MOV #64.,T23SZ ;SET UP BUFFER SIZE
1367 031672 012704 032510' MOV #T23PK3,R4 ;R4 = POINTER TO PACKET
1368 031676 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
1369 031702 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
1370 031706 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
1371 031712 012702 104210 MOV #SC!NXM!SSR!BIT3,R2 ;SET UP EXPECTED
1372 031716 020102 CMP R1,R2 ;ARE THEY EQUAL
1373 031720 001414 BEQ 80$ ;BR, IF OK ESP. FUNCTION REJECT
1374 031722 005237 032514' INC T23WB+2 ;BUMP TO NEXT ADDRESS BIT
1375 031726 023727 032514' 000004 CMP T23WB+2,#4 ;CHECK TO SEE IF OVERFLOW INTO 19 BIT
1376 031734 001345 BNE 24$ ;BR, IF BITS 17 AND 18
1377 031736 005237 002212' 25$: INC FATFLG ;BUMP COUNT
1381 031742 ERRHRD ERRNO,T23TM,PKTSSR ;TSSR INCORRECT AFTER WRITE COMMAND
    031742 104456 TRAP C$ERHRD
    031744 000507 .WORD 327
    031746 032742' .WORD T23TM
    031750 011746' .WORD  PKTSSR
1382 031752 80$: CKLOOP ;LOOP IF SELECTED

```



```

1434 032064 103407          BCS      23$          ;BR, IF COMMAND ISSUED OK
1435 032066 005237 002212'  INC      FATFLG      ;BUMP COUNT
1439 032072 010001          MOV      R0,R1       ;SAVE CONTENTS OF TSSR
1440 032074          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      032074 104456          TRAP     C$ERHRD
      032076 000511          .WORD   329
      032100 005046'        .WORD   WRTMSG
      032102 011734'        .WORD   SFIMSG

1441
1442
1443          ;*****
1444          ;WRITE DATA, ACK,CVC=1
1445          ;*****
1446
1447
1448 032104          23$:
1449 032104 012701 160000      MOV      #160000,R1   ;NXM LOW ADDRESS START
1450 032110 012702 177776      MOV      #177776,R2   ;LIMIT CHECK FOR NXM (HIGHEST)
1451 032114 004737 016276'    JSR      PC, XNXM     ;LOOK FOR NXM ADDRESS
1452 032120 103051          BCC      80$          ;BR, IF NON FOUND
1453 032122 010137 003130'    MOV      R1, NXML0    ;SET ADDRESS UP FOR TEST
1454 032126 012737 000000 032514'  MOV      #0, T23WB+2  ;SET TO 16 BIT ADDRESS
1455 032134          24$:
1456 032134 012737 140005 032510'  MOV      #140005, T23PK3 ;WRITE DATA, ACK,CVC=1
1457 032142 013701 003130'    MOV      NXML0, R1    ;HIGHEST MEMORY ADDRESS LOW BITS
1458 032146 162701 000500      SUB      #500, R1     ;SET ADDRESS A LITTLE LOWER
1459 032152 010137 032512'    MOV      R1, T23WB    ;LOAD INTO THE PACKET
1460 032156 012737 000000 032516'  MOV      #0., T23SZ   ;SET UP BUFFER SIZE (64K BYTES)
1461 032164 012704 032510'    MOV      #T23PK3, R4  ;R4 = POINTER TO PACKET
1462 032170 010465 000000      MOV      R4, TSDB(R5) ;ISSUE COMMAND
1463 032174 004737 016150'    JSR      PC, WAITF    ;WAIT FOR SSR TO SET
1464 032200 016501 000002      MOV      TSSR(R5), R1 ;GET TSSR CONTENTS
1465 032204 012702 104210      MOV      #SC!NXM!SSR!BIT3, R2 ;SET UP EXPECTED
1466 032210 020102          CMP      R1, R2      ;ARE THEY EQUAL
1467 032212 001414          BEQ      80$          ;BR, IF OK ESP. FUNCTION REJECT
1468 032214 005237 032514'    INC      T23WB+2     ;BUMP TO NEXT ADDRESS RANGE
1469 032220 023727 032514' 000004  CMP      T23WB+2, #4  ;CHECK TO SEE IF WE WENT TO HIGH
1470 032226 001342          BNE      24$          ;BR, IF NO OVER FLOW
1471 032230 005237 002212'    25$: INC      FATFLG      ;BUMP COUNT
1475 032234          ERRHRD  ERRNO, T23TM, PKTSSR ;TSSR INCORRECT AFTER WRITE COMMAND
      032234 104456          TRAP     C$ERHRD
      032236 000512          .WORD   330
      032240 032742'        .WORD   T23TM
      032242 011746'        .WORD   PKTSSR

1476 032244          80$: CKLOOP          ;LOOP IF SELECTED
      032244 104406          TRAP     C$CLP1
1477 032246 004737 033774'    JSR      PC, T23RT2   ;CLEAN UP PACKET
1478 032252 004737 034036'    JSR      PC, T23RT3   ;RESTORE PACKET
1479 032256 012737 102010 032470'  MOV      #102010, T23PK2 ;REWIND (POSITION) COMMAND
1480 032264 012704 032470'    MOV      #T23PK2, R4  ;LOAD R4 WITH PACKET ADDRESS
1481 032270 010465 000000      MOV      R4, TSDB(R5) ;ISSUE REWIND COMMAND
1482 032274 004737 016236'    JSR      PC, CHKTSSR  ;WAIT FOR SSR TO SET
1483 032300 103407          BCS      130$         ;BR, IF TSSR IS OK (GOOD)
1484 032302 010001          MOV      R0, R1       ;SAVE TSSR CONTENTS
1485 032304 005237 002212'    INC      FATFLG      ;BUMP COUNT
1489 032310          ERRHRD  ERRNO, T23RWN, PKTSSR ;TSSR IS INCORRECT AFTER REWIND
      032310 104456          TRAP     C$ERHRD

```


1546 032522 010
1547 032523 200
1548 032524 000000
1549 032526 000000
1550
1551
1552 032530 000000
1553 032532 000000
1554
1555
1556
1557
1558 032534 100005
1559 032536 100405
1560 032540 102005
1561 032542 177777
1562
1563
1564
1565
1566

1567 032544 127 122
1568 032577 105 117
1569 032664 127 122
1570 032742 124 123
1571 033016 122 145
1572 033065 122 101
1573 033140 124 123
1574 033206 104 162
1575 033261 124 123
1576 033350 124 123
1577 033452 103 126
1578 033525 124 123
1579 033577 127 122
1580 033666 102 141
1581
1582
1583
1584
1585
1586
1587
1588

1589 033702
1590 033702
1591 033706 012701 032360'
1592 033712 012721 100004
1593 033716 012721 032370'
1594 033722 005021
1595 033724 012721 000012
1596 033730 012721 032402'
1597 033734 005021
1598 033736 012721 000024
1599 033742 005021
1600 033744 012711 000000
1601 033750 012702 000030
1602 033754 012762 177777 032402' 64:

```

T23BS0: .BYTE 10 ;BSELO AREA
T23BS1: .BYTE 200 ;BSEL1 AREA
T23S2: .WORD 0 ;SEL 2 AREA
T23S3: .WORD 0 ;DATA AREA
;
;
T23TMP: .WORD 0 ;TEMPORARY REGISTER
T23WRT: .WORD 0 ;RETRY COMMAND
;
.EVEN
;TAPE MOTION PACKET COMMAND VALUES
T23WD: .WORD 100005 ;WRITE DATA (NEXT)
T23WDR: .WORD 100405 ;WRITE DATA RETRY
T23CON: .WORD 102005 ;WRITE CONTINUOUS
        .WORD 177777 ;END OF DATA
;
;*
;LOCAL TEXT MESSAGES FOR TEST
;
111 T23SSR: .ASCIZ 'WRITE Command Not Accepted'
124 T23ET: .ASCIZ 'EOT Not Found In 12000 4k Writes, (Use Shorter Tape)'
111 T23EOT: .ASCIZ 'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
123 T23TM: .ASCIZ 'TSSR Not Correct After WRITE Command Reject'
167 T23RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
115 T23RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
123 T23AM3: .ASCIZ 'TSSR Init. Failed After WRITE Command'
151 T23OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
123 T23WDD: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, SWB Bit Set'
123 T23WDC: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
103 T23VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
102 T23BA: .ASCIZ 'TSBA Not Correct After WRITE DATA Command'
111 T23WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
163 TST23ID: .ASCIZ 'Basic Write'
.EVEN
;
;
;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
;WRITE SUBSYSTEM MEMORY COMMAND
;
;
T23REST:
        SAVREG ;SAVE THE REGISTERS
        MOV #T23PACKET,R1 ;START OF THE PACKET
        MOV #100004,(R1); ;WRITE SUBSYSTEM MEM. WITH ACK
        MOV #T23DATA,(R1); ;ADDRESS OF CHARAISTICS DATA BLOCK
        CLR (R1); ;EXTENDED ADDRESS
        MOV #10.,(R1); ;SIZE OF DATA BLOCK IN BYTES
        MOV #T23BFR,(R1); ;ADDRESS OF MESSAGE BUFFER
        CLR (R1);
        MOV #20.,(R1); ;LENGTH OF MESSAGE BUFFER
        CLR (R1);
        MOV #0,(R1) ;SELECT DRIVE ZERO
        MOV #24.,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
        MOV #177777,T23BFR(R2) ;ALL ONES TO MESSAGE BUFFER
    
```



```

1603 033762 005742          TST      -(R2)          ;BUMP DOWN TO NEXT LOCATION
1604 033764 020227 000000    CMP      R2,#0         ;R2 AT ZERO YET
1605 033770 001371          BNE     64$           ;KEEP GOING UNTIL DONE
1606 033772 000207          RTS     PC            ;RETURN
1607
1608
1609 033774          T23RT2:
1610 033774          SAVREG          ;SAVE THE REGISTERS
1611 034000 012701 032470'    MOV     @T23PK2,R1    ;START OF THE PACKET
1612 034004 012721 100006    MOV     @100006,(R1). ;WRITE SUBSYSTEM MEM. WITH ACK
1613 034010 012721 032522'    MOV     @T23BF2,(R1). ;ADDRESS OF DATA BLOCK
1614 034014 005021          CLR     (R1).        ;EXTENDED ADDRESS
1615 034016 012721 000006    MOV     @6.,(R1).    ;SIZE OF DATA BLOCK IN BYTES
1616 034022 012701 032522'    MOV     @T23BF2,R1   ;POINT TO DATA SEL AREA
1617 034026 005021          CLR     (R1).
1618 034030 005021          CLR     (R1).
1619 034032 005011          CLR     (R1).
1620 034034 000207          RTS     PC            ;RETURN
1621 034036          T23RT3:
1622 034036          SAVREG          ;SAVE THE REGISTERS
1623 034042 012701 032510'    MOV     @T23PK3,R1    ;START OF THE PACKET
1624 034046 012721 100005    MOV     @100005,(R1). ;WRITE TAPE. WITH ACK
1625 034052 005021          CLR     (R1).        ;ADDRESS OF DATA BLOCK
1626 034054 005021          CLR     (R1).        ;EXTENDED ADDRESS
1627 034056 005011          CLR     (R1).        ;SIZE OF DATA BLOCK
1628 034060 000207          RTS     PC            ;RETURN
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638 034062          ;*
1639 034062          ;ROUTINE TO RETRY WRITE DATA IN CASE OF BAD TAPE FOR TEST
1640 034066 005037 032530'    ;3.SUBTEST 2 & 3
1641 034072 032701 100000    ;
1642 034076 001452          ;INPUTS:          R1=TSSR
1643 034100 013702 032412'    ;SUBROUTINE SETS UP T23WRT FOR RETRY
1644 034104 032702 000002    ;
1645 034110 001401          ;
1646 034112 000405          ;
1647 034114 032702 020000    ;
1648 034120 001002          ;
1649 034122 000440          ;
1650 034124 000207          ;
1651
1652 034126          T23CHK:
1653 034126 012703 000024    SAVREG          ;SAVE THE REGISTERS
1654 034132 013737 003114'    CLR     T23TMP       ;CLEAR LOCAL REGISTER
1655 034140 012737 032532'    BIT     @SC,R1        ;IS SC SET IN TSSR?
1656 034146 012704 032510'    BEQ     FATAL        ;NO, YOU GOT PROBLEMS!
1657 034152 010300          MOV     T23BFR+10,R2 ;YES,GET XSTAT1
1658 034154 004737 017324'    BIT     @X1.UNC,R2   ;IS UNC SET IN XSTAT1?
1659 034160 010337 032516'    BEQ     1$           ;NO, CHECK COR
1660          BR     RETRY    ;YES,DO WRITE DATA RETRY
1661          BIT     @X1.COR,R2 ;IS COR SET IN XSTAT1 THEN?
1662          BNE     RETRY    ;YES SO RETRY
1663          BR     FATAL    ;NO, YOU GOT PROBLEMS
1664          RTS     PC     ;RETURN
1665
1666          RETRY:
1667          2$: MOV     @20.,R3      ;STARTING RECORD SIZE
1668          MOV     FREE,T23WB ;STARTING WRITE BUFFER ADDRESS
1669          MOV     @T23WRT,T23PK3 ;WRITE DATA RETRY COMMAND SETUP BY SUBROUTINE
1670          MOV     @T23PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
1671          MOV     R3,R0       ;SET PATTERN IN CORRECT REGISTER
1672          JSR     PC,FILLMEM  ;FILL MEMORY WITH RECORD SIZE
1673          MOV     R3,T23SZ    ;SET UP RECORD SIZE IN PACKET

```



```

1766 034444 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
1767 034446          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
                                TRAP    C$ERHRD
                                .WORD   402
                                .WORD   WRTMSG
                                .WORD   SFIMSG
1768 034456 005737 002216' 24$:  TST    EXTFEA      ;CHECK FOR EXTENDED FEATURES SW SWITCH
1769 034462 001044          BNE    50$          ;BR IF SWITCH IS ON
1770
1771 034464 112737 000200 043771'  MOVB   #200,T24BS1    ;WRITE MISCELLANEOUS CONT/READ STATUS
1772 034472 112737 000010 043770'  MOVB   #10,T24BS0    ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
1773 034500 012704 043740'  MOV    #T24PK2,R4    ;WRITE SUBSYS MEM PACKET
1774 034504 010465 000000          MOV    R4,TSDB(R5)   ;ISSUE COMMAND
1775 034510 004737 016236'  JSR    PC,CHKTSSR    ;WAIT FOR SSR
1776 034514 103407          BCS    30$          ;BR, IF NO ERROR
1777 034516 010001          MOV    R0,R1        ;ERROR, SAVE TSSR
1778 034520 005237 002212'  INC    FATFLG        ;BUMP COUNT
1782 034524          ERRHRD  ERRNO,T24SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
                                TRAP    C$ERHRD
                                .WORD   403
                                .WORD   T24SSR
                                .WORD   PKTSSR
1783 034534          CKLOOP              ;LOOP IF SELECTED
                                TRAP    C$CLP1
1784 034536 012737' 000007 043650'  MOV    #7,T24DSW     ;SET DRIVE NUMBER IN PACKET
1785 034544 012704 043630'  MOV    #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
1786
1787 ;*****
1788 ;
1789 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
1790 ;
1791 ;*****
1792
1793 034550 004737 010562'  JSR    PC,WRTCHR     ;ISSUE WRITE CHARACTERISTICS
1794 034554 103407          BCS    50$          ;BR, IF COMMAND ISSUED OK
1795 034556 005237 002212'  INC    FATFLG        ;BUMP COUNT
1799 034562 010001          MOV    R0,R1        ;SAVE CONTENTS OF TSSR
1800 034564          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
                                TRAP    C$ERHRD
                                .WORD   404
                                .WORD   WRTMSG
                                .WORD   SFIMSG
1801 034574          CKLOOP              ;SCOPE LOOP
                                TRAP    C$CLP1
1802 034576 016501 000002          MOV    TSSR(R5),R1  ;GET TSSR CONTENTS
1803 034602 032701 000100          BIT    #OFL,R1     ;CHECK FOR THE OFFLINE BIT SET
1804 034606 001006          BNE    60$          ;BR, IF OFFLINE (GOOD)
1805 034610 005237 002212'  INC    FATFLG        ;BUMP COUNT
1809 034614          ERRDF  ERRNO,T24OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
                                TRAP    C$ERDF
                                .WORD   405
                                .WORD   T24OFL
                                .WORD   SFIMSG
1810 034624          CKLOOP              ;LOOP IF SELECTED
                                TRAP    C$CLP1
1811 034626 012703 043776'  MOV    #T24RN,R3    ;POINTER FOR COMMANDS
1812
    
```



```

1916 ;*****
1917 ;
1918 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
1919 ;
1920 ;*****
1921
1922 035064 013701 043660'      MOV      T24BFR+6,R1      ;PICK UP XSTO
1923 035070 010102              MOV      R1,R2           ;SET UP EXPECTED
1924 035072 052702 000002      BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
1925 035076 020102              CMP      R1,R2           ;DOES EXP = REC'D
1926 035100 001406              BEQ      40$             ;BR, IF EQUAL (OK)
1927 035102 005237 002212'      INC      FATFLG          ;BUMP COUNT
1931 035106              ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD     410
                                .WORD     T24BOT
                                .WORD     EXPREC
1932 035116              40$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
1933 035120 012703 000400      MOV      @256.,R3        ;RECORD SIZE
1934 035124 013737 003114' 043762'  MOV      FREE,T24RB      ;STARTING WRITE BUFFER ADDRESS
1935
1936 ;*****
1937 ;
1938 ;WRITE DATA,CVC=1,ACK COMMAND
1939 ;
1940 ;*****
1941
1942 035132 012737 140005 043760'  MOV      @140005,T24PK3  ;WRITE DATA,CVC=1,ACK COMMAND
1943 035140 012704 043760'      MOV      @T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
1944 035144              65$:
1945 035144 010300              MOV      R3,R0           ;SET PATTERN IN CORRECT REGISTER
1946 035146 004737 017324'      JSR      PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
1947 035152 010337 043766'      MOV      R3,T24SZ        ;SET UP RECORD SIZE IN PACKET
1948 035156 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
1949 035162 004737 016150'      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
1950 035166 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
1951 035172 012702 000200      MOV      @SSR,R2         ;SET UP EXPECTED
1952 035176 020102              CMP      R1,R2           ;ARE THEY EQUAL
1953 035200 001406              BEQ      75$             ;BR, IF OK
1954 035202 005237 002212'      INC      FATFLG          ;BUMP COUNT
1958 035206              ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD     411
                                .WORD     WRTErr
                                .WORD     PKTSSR
1959 035216              75$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
1960 035220 005723              TST      (R3)+           ;BUMP RECORD SIZE
1961 035222 022703 000414      CMP      @268.,R3        ;END OF RECORD YET
1962 035226 001346              BNE      65$             ;BR, IF MORE RECORDS TO WRITE
1963 035230              80$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
1964 035232              120$:
1965
1966 ;*****
1967 ;

```

```

1968 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
1969 ;
1970 ;*****
1971
1972 035232 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
1973 035236 004737 016236' JSR PC,CHKTSSR ;SEE HOW TSSR IS
1974 035242 103407 BCS 130$ ;BR, IF NO PROBLEM
1975 035244 010001 MOV R0,R1 ;SAVE TSSR
1976 035246 005237 002212' INC FATFLG ;BUMP COUNT
1980 035252 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
    035252 104456 TRAP C$ERHRD
    035254 000634 .WORD 412
    035256 045116' .WORD T24RWN
    035260 011746' .WORD PKTSSR
1981 035262 130$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
    035262 104406
1982 ;*****
1983 ;
1984 ;
1985 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
1986 ;
1987 ;*****
1988
1989 035264 013701 043660' MOV T24BFR+6,R1 ;PICK UP XSTO
1990 035270 010102 MOV R1,R2 ;SET UP EXPECTED
1991 035272 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
1992 035276 020102 CMP R1,R2 ;DOES EXP = REC'D
1993 035300 001406 BEQ 140$ ;BR, IF EQUAL (OK)
1994 035302 005237 002212' INC FATFLG ;BUMP COUNT
1998 035306 ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
    035306 104456 TRAP C$ERHRD
    035310 000635 .WORD 413
    035312 044633' .WORD T24BOT
    035314 015374' .WORD EXPREC
1999 035316 140$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
    035316 104406
2000 035320 012703 000400 MOV #256.,R3 ;RECORD SIZE
2001 035324 013737 003114' 043762' MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
2002 ;*****
2003 ;
2004 ;
2005 ;READ DATA,CVC=1,ACK COMMAND
2006 ;
2007 ;*****
2008
2009 035332 012737 140001 043760' MOV #140001,T24PK3 ;READ DATA,CVC=1,ACK COMMAND
2010 035340 012704 043760' 165$: MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2011 035344 010337 043766' MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
2012 035350 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
2013 035354 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
2014 035360 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
2015 035364 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
2016 035370 020102 CMP R1,R2 ;ARE THEY EQUAL
2017 035372 001406 BEQ 170$ ;BR, IF OK
2018 035374 005237 002212' INC FATFLG ;BUMP COUNT
2022 035400 ERRHRD ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
    035400 104456 TRAP C$ERHRD
    
```



```

2069 ;*****
2070 ;
2071 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
2072 ;
2073 ;*****
2074
2075 035532 004737 015674' JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
2076 035536 103407 BCS 20$ ;BR IF INIT WAS OK
2077 035540 005237 002212' INC FATFLG ;BUMP COUNT
2081 035544 010001 MOV RO,R1 ;CONTENTS OF TSSR REGISTER
2082 035546 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
035546 104455 TRAP C$ERDF
035550 000640 .WORD 416
035552 003642' .WORD SFIERR
035554 011734' .WORD SFIMSG
2083 035556 20$: MOV UNITN,T24DSW ;SET DRIVE NUMBER IN PACKET
2084 035556 013737 002172' 043650' MOV #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
2085 035564 012704 043630'
2086
2087 ;*****
2088 ;
2089 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
2090 ;
2091 ;*****
2092
2093 035570 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
2094 035574 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
2095 035576 005237 002212' INC FATFLG ;BUMP COUNT
2099 035602 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
2100 035604 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
035604 104456 TRAP C$ERHRD
035606 000641 .WORD 417
035610 005046' .WORD WRTMSG
035612 011734' .WORD SFIMSG
2101 035614 24$: CKLOOP ;LOOP IF SELECTED
035614 104406 TRAP C$CLP1
2102
2103 ;*****
2104 ;
2105 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2106 ;
2107 ;*****
2108
2109 035616 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2110 035622 103407 BCS 30$ ;BR, IF NO PROBLEM
2111 035624 010001 MOV RO,R1 ;SAVE TSSR
2112 035626 005237 002212' INC FATFLG ;BUMP COUNT
2116 035632 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
035632 104456 TRAP C$ERHRD
035634 000642 .WORD 418
035636 045116' .WORD T24RWN
035640 011746' .WORD PKTSSR
2117 035642 30$: CKLOOP ;LOOP IF SELECTED
035642 104406 TRAP C$CLP1
2118
2119 ;*****
2120 ;

```

```

2121 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
2122 ;
2123 ;*****
2124
2125 035644 013701 043660'          MOV      T24BFR+6,R1          ;PICK UP XSTO
2126 035650 010102                MOV      R1,R2              ;SET UP EXPECTED
2127 035652 052702 000002          BIS      #BIT1,R2          ;SET BOT BIT IN EXPECTED
2128 035656 020102                CMP      R1,R2              ;DOES EXP = REC'D
2129 035660 001406                BEQ     40$                 ;BR, IF EQUAL (OK)
2130 035662 005237 002212'          INC     FATFLG              ;BUMP COUNT
2131 035666                ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP   C$ERHRD
                                .WORD  419
                                .WORD  T24BOT
                                .WORD  EXPREC
2132 035666 104456
2133 035670 000643
2134 035672 044633'
2135 035674 015374'
2135 035676                40$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP   C$CLP1
2136 035700 012703 000400          MOV     #256.,R3           ;RECORD SIZE
2137 035704 013737 003114' 043762'  MOV     FREE,T24RB        ;STARTING WRITE BUFFER ADDRESS
2138
2139 ;*****
2140 ;
2141 ;WRITE DATA,ACK,SWB,CVC=1 COMMAND
2142 ;
2143 ;*****
2144
2145 035712 012737 150005 043760'    MOV     #150005,T24PK3    ;WRITE DATA,ACK,SWB,CVC=1 COMMAND
2146 035720 012704 043760'          MOV     #T24PK3,R4       ;SET UP R4 WITH PACKET ADDRESS
2147 035724
2148 035724 010300                65$:   MOV     R3,R0         ;SET PATTERN IN CORRECT REGISTER
2149 035726 004737 017324'          JSR    PC,FILLMEM        ;FILL MEMORY WITH RECORD SIZE
2150 035732 010337 043766'          MOV     R3,T24SZ         ;SET UP RECORD SIZE IN PACKET
2151 035736 010465 000000          MOV     R4,TSDB(R5)     ;ISSUE COMMAND
2152 035742 004737 016150'          JSR    PC,WAITF         ;WAIT FOR SSR TO SET
2153 035746 016501 000002          MOV     TSSR(R5),R1     ;GET TSSR CONTENTS
2154 035752 012702 000200          MOV     #SSR,R2         ;SET UP EXPECTED
2155 035756 020102                CMP     R1,R2           ;ARE THEY EQUAL
2156 035760 001406                BEQ    75$              ;BR, IF OK
2157 035762 005237 002212'          INC     FATFLG          ;BUMP COUNT
2161 035766                ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP   C$ERHRD
                                .WORD  420
                                .WORD  WRERR
                                .WORD  PKTSSR
2162 035776                75$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP   C$CLP1
2163 036000 005723                TST    (R3)+            ;BUMP RECORD SIZE
2164 036002 022703 000414          CMP     #268.,R3        ;END OF RECORD YET
2165 036006 001346                BNE    65$              ;BR, IF MORE RECORDS TO WRITE
2166 036010                80$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP   C$CLP1
2167 036012                120$:
2168
2169 ;*****
2170 ;
2171 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2172 ;

```

```

2173
2174
2175 036012 004737 010714'          JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
2176 036016 103407                  BCS      130$              ;BR, IF NO PROBLEM
2177 036020 010001                  MOV      R0,R1             ;SAVE TSSR
2178 036022 005237 002212'          INC      FATFLG           ;BUMP COUNT
2182 036026                  ERRHRD  ERRNO,T24RWN,EXPREC ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     421
                                .WORD     T24RWN
                                .WORD     EXPREC
2183 036036 104406          130$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
2184
2185
2186
2187
2188
2189
2190
2191 036040 013701 043660'          MOV      T24BFR+6,R1      ;PICK UP XSTO
2192 036044 010102                  MOV      R1,R2            ;SET UP EXPECTED
2193 036046 052702 000002          BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
2194 036052 020102                  CMP      R1,R2            ;DOES EXP = REC'D
2195 036054 001406                  BEQ      140$             ;BR, IF EQUAL (OK)
2196 036056 005237 002212'          INC      FATFLG           ;BUMP COUNT
2200 036062                  ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD     422
                                .WORD     T24BOT
                                .WORD     EXPREC
2201 036072 104406          140$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
2202 036074 012703 000400          MOV      #256.,R3        ;RECORD SIZE
2203 036100 013737 003114' 043762'  MOV      FREE,T24RB      ;STARTING READ BUFFER ADDRESS
2204
2205
2206
2207
2208
2209
2210
2211 036106 012737 110001 043760'  165$:  MOV      #110001,T24PK3  ;READ DATA,IE,ACK,SWB COMMAND
2212 036114 012704 043760'          MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
2213 036120 010337 043766'          MOV      R3,T24SZ        ;SET UP RECORD SIZE IN PACKET
2214 036124 010465 000000          MOV      R4,TSDB(R5)     ;ISSUE COMMAND
2215 036130 004737 016150'          JSR      PC,WAITF        ;WAIT FOR SSR TO SET
2216 036134 016501 000002          MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
2217 036140 012702 000200          MOV      #SSR,R2        ;SET UP EXPECTED
2218 036144 020102                  CMP      R1,R2            ;ARE THEY EQUAL
2219 036146 001406                  BEQ      170$             ;BR, IF OK
2220 036150 005237 002212'          INC      FATFLG           ;BUMP COUNT
2224 036154                  ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD     423
                                .WORD     RDERR
                                .WORD     PKTSSR
036154 104456
036156 000647
036160 005176'
036162 011746'

```



```

2274
2275 ;
2276 ;.....
2277 036306 004737 015674' JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
2278 036312 103407 BCS 20$ ;BR IF INIT WAS OK
2279 036314 005237 002212' INC FATFLG ;BUMP COUNT
2283 036320 010001 MOV R0,R1 ;CONTENTS OF TSSR REGISTER
2284 036322 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      036322 104455 TRAP C$ERDF
      036324 000651 .WORD 425
      036326 003642' .WORD SFIERR
      036330 011734' .WORD SFIMSG
2285 036332 20$: MOV UNITN,T24DSW ;SET DRIVE NUMBER IN PACKET
2286 036332 013737 002172' 043650' MOV @T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
2287 036340 012704 043630'
2288
2289 ;.....
2290 ;
2291 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
2292 ;
2293 ;.....
2294
2295 036344 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
2296 036350 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
2297 036352 005237 002212' INC FATFLG ;BUMP COUNT
2301 036356 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
2302 036360 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      036360 104456 TRAP C$ERHRD
      036362 000652 .WORD 426
      036364 005046' .WORD WRTMSG
      036366 011734' .WORD SFIMSG
2303 036370 24$: CKLOOP ;LOOP IF SELECTED
      036370 104406 TRAP C$CLP1
2304
2305 ;.....
2306 ;
2307 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2308 ;
2309 ;.....
2310
2311 036372 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2312 036376 103407 BCS 30$ ;BR, IF NO PROBLEM
2313 036400 010001 MOV R0,R1 ;SAVE TSSR
2314 036402 005237 002212' INC FATFLG ;BUMP COUNT
2318 036406 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      036406 104456 TRAP C$ERHRD
      036410 000653 .WORD 427
      036412 045116' .WORD T24RWN
      036414 011746' .WORD PKTSSR
2319 036416 30$: CKLOOP ;LOOP IF SELECTED
      036416 104406 TRAP C$CLP1
2320
2321 ;.....
2322 ;
2323 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
2324 ;
2325 ;.....
    
```

```

2326
2327 036420 013701 043660'      MOV      T24BFR+6,R1      ;PICK UP XSTO
2328 036424 010102              MOV      R1,R2           ;SET UP EXPECTED
2329 036426 052702 000002      BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
2330 036432 020102              CMP      R1,R2           ;DOES EXP = REC'D
2331 036434 001406              BEQ      40$             ;BR, IF EQUAL (OK)
2332 036436 005237 002212'      INC      FATFLG          ;BUMP COUNT
2336 036442              ERRHRD   ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD     428
                                .WORD     T24BOT
                                .WORD     EXPREC
                                TRAP      C$CLP1
                                .WORD     104456
                                .WORD     000654
                                .WORD     044633'
                                .WORD     015374'
2337 036452              40$:   CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD     104406
2338 036454 012703 001000      MOV      @512.,R3        ;RECORD SIZE
2339 036460 013737 003114' 043762'  MOV      FREE,T24RB      ;STARTING WRITE BUFFER ADDRESS
2340
2341 ;*****
2342 ;
2343 ;WRITE DATA,ACK,CVC=1 COMMAND
2344 ;
2345 ;*****
2346
2347 036466 012737 140005 043760'      MOV      @140005,T24PK3  ;WRITE DATA,ACK,CVC=1 COMMAND
2348 036474 012704 043760'      MOV      @T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
2349 036500              65$:
2350 036500 010337 043766'      MOV      R3,T24SZ        ;SET UP RECORD SIZE IN PACKET
2351 036504 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
2352 036510 004737 016150'      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
2353 036514 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
2354 036520 012702 000200      MOV      @SSR,R2        ;SET UP EXPECTED
2355 036524 020102              CMP      R1,R2           ;ARE THEY EQUAL
2356 036526 001406              BEQ      75$             ;BR, IF OK
2357 036530 005237 002212'      INC      FATFLG          ;BUMP COUNT
2361 036534              ERRHRD   ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD     104456
                                .WORD     000655
                                .WORD     005103'
                                .WORD     011746'
                                .WORD     104406
2362 036544              75$:   CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD     036544
2363 036546              120$:
2364 ;*****
2365 ;
2366 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2367 ;
2368 ;*****
2369
2370
2371 036546 004737 010714'      JSR      PC,REWIND       ;CALL TAPE REWIND COMMAND
2372 036552 103407              BCS     130$             ;BR, IF NO PROBLEM
2373 036554 010001              MOV      R0,R1           ;SAVE TSSR
2374 036556 005237 002212'      INC      FATFLG          ;BUMP COUNT
2378 036562              ERRHRD   ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     104456
                                .WORD     000656
                                .WORD     045116'
                                .WORD     T24RWN
    
```

```

2379 036570 011746'          130$:  CKLOOP                ;LOOP IF SELECTED          .WORD  PKTSSR
      036572                ;*****
      036572 104406          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
2380 ;*****
2381 ;
2382 ;
2383 ;
2384 ;
2385 ;
2386 ;*****
2387 036574 013701 043660'    MOV      T24BFR+6,R1      ;PICK UP XSTO
2388 036600 010102          MOV      R1,R2           ;SET UP EXPECTED
2389 036602 052702 000002    BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
2390 036606 020102          CMP      R1,R2           ;DOES EXP = REC'D
2391 036610 001406          BEQ      140$            ;BR, IF EQUAL (OK)
2392 036612 005237 002212'    INC      FATFLG          ;BUMP COUNT
2396 036616                ERRHRD  ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      036616 104456          TRAP                                C$ERHRD
      036620 000657          .WORD                                431
      036622 044633'        .WORD                                T24BOT
      036624 015374'        .WORD                                EXPREC
2397 036626                140$:  CKLOOP                ;LOOP IF SELECTED          .WORD  PKTSSR
      036626 104406          TRAP                                C$CLP1
2398 036630 012703 000400    MOV      @256.,R3        ;RECORD SIZE
2399 036634 013737 003114' 043762'  MOV      FREE,T24RB      ;STARTING READ BUFFER ADDRESS
2400 ;*****
2401 ;
2402 ;
2403 ;READ DATA,ACK,CVC=1 COMMAND
2404 ;
2405 ;*****
2406 ;
2407 036642 012737 140001 043760'  MOV      @140001,T24PK3 ;READ DATA,ACK,CVC=1 COMMAND
2408 036650 012704 043760'  165$:  MOV      @T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2409 036654 010337 043766'    MOV      R3,T24SZ        ;SET UP RECORD SIZE IN PACKET
2410 036660 010465 000000    MOV      R4,TSDB(R5)     ;ISSUE COMMAND
2411 036664 004737 016150'    JSR      PC,WAITF        ;WAIT FOR SSR TO SET
2412 036670 016501 000002    MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
2413 036674 012702 100204    MOV      @SSR!SC!BIT2,R2 ;SET UP EXPECTED
2414 036700 020102          CMP      R1,R2           ;ARE THEY EQUAL
2415 036702 001406          BEQ      170$            ;BR, IF OK
2416 036704 005237 002212'    INC      FATFLG          ;BUMP COUNT
2420 036710                ERRHRD  ERRNO,T24TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      036710 104456          TRAP                                C$ERHRD
      036712 000660          .WORD                                432
      036714 046164'        .WORD                                T24TRL
      036716 011746'        .WORD                                PKTSSR
2421 036720                170$:  CKLOOP                ;LOOP IF SELECTED          .WORD  PKTSSR
      036720 104406          TRAP                                C$CLP1
2422 ;*****
2423 ;
2424 ;
2425 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
2426 ;
2427 ;*****
2428 ;
2429 036722 013701 043660'    MOV      T24BFR+6,R1     ;GET MESSAGE BUFFER

```



```

037032 011734' .WORD SFIMSG
2482 037034
2483 037034 013737 002172' 043650' 20$: MOV UNITN,T24DSW ;SET DRIVE NUMBER IN PACKET
2484 037042 012704 043630' MOV @T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
2485
2486 ;*****
2487 ;
2488 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
2489 ;
2490 ;*****
2491
2492 037046 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
2493 037052 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
2494 037054 005237 002212' INC FATFLG ;BUMP COUNT
2498 037060 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
2499 037062 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
037062 104456 TRAP C$ERHRD
037064 000663 .WORD 435
037066 005046' .WORD WRTMSG
037070 011734' .WORD SFIMSG
2500 037072 24$: CKLOOP ;LOOP IF SELECTED
037072 104406 TRAP C$CLP1
2501
2502 ;*****
2503 ;
2504 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2505 ;
2506 ;*****
2507
2508 037074 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2509 037100 103407 BCS 30$ ;BR, IF NO PROBLEM
2510 037102 010001 MOV R0,R1 ;SAVE TSSR
2511 037104 005237 002212' INC FATFLG ;BUMP COUNT
2515 037110 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
037110 104456 TRAP C$ERHRD
037112 000664 .WORD 436
037114 045116' .WORD T24RWN
037116 011746' .WORD PKTSSR
2516 037120 30$: CKLOOP ;LOOP IF SELECTED
037120 104406 TRAP C$CLP1
2517 037122 012703 000400 MOV @256.,R3 ;RECORD SIZE
2518 037126 013737 003114' 043762' MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
2519
2520 ;*****
2521 ;
2522 ;WRITE DATA,ACK,CVC-1 COMMAND
2523 ;
2524 ;*****
2525
2526 037134 012737 140005 043760' MOV @140005,T24PK3 ;WRITE DATA,ACK,CVC-1 COMMAND
2527 037142 012704 043760' MOV @T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
2528 037146
2529 037146 010337 043766' 65$: MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
2530 037152 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
2531 037156 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
2532 037162 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
2533 037166 012702 000200 MOV @SSR,R2 ;SET UP EXPECTED
    
```

```

2534 037172 020102          CMP      R1,R2          ;ARE THEY EQUAL
2535 037174 001406          BEQ      75$           ;BR, IF OK
2536 037176 005237 002212'  INC      FATFLG       ;BUMP COUNT
2540 037202          ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      037202 104456          TRAP    C$ERHRD
      037204 000665          .WORD  437
      037206 005103'        .WORD  WRTErr
      037210 011746'        .WORD  PKTSSR
2541 037212          75$:   CKLOOP          ;LOOP IF SELECTED
      037212 104406          TRAP    C$CLP1
2542 037214          120$:
2543
2544          ;*****
2545          ;
2546          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2547          ;
2548          ;*****
2549
2550 037214 004737 010714'        JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
2551 037220 103407          BCS      130$         ;BR, IF NO PROBLEM
2552 037222 010001          MOV      R0,R1        ;SAVE TSSR
2553 037224 005237 002212'  INC      FATFLG       ;BUMP COUNT
2557 037230          ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      037230 104456          TRAP    C$ERHRD
      037232 000666          .WORD  438
      037234 045116'        .WORD  T24RWN
      037236 011746'        .WORD  PKTSSR
2558 037240          130$:  CKLOOP          ;LOOP IF SELECTED
      037240 104406          TRAP    C$CLP1
2559 037242 012703 001000        MOV      #512.,R3     ;RECORD SIZE
2560 037246 013737 003114' 043762'  MOV      FREE,T24RB   ;STARTING READ BUFFER ADDRESS
2561
2562          ;*****
2563          ;
2564          ;READ DATA,ACK,CVC=1 COMMAND
2565          ;
2566          ;*****
2567
2568 037254 012737 140001 043760'  MOV      #140001,T24PK3 ;READ DATA,ACK,CVC=1 COMMAND
2569 037262 012704 043760'  165$:  MOV      #T24PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
2570 037266 010337 043766'  MOV      R3,T24SZ     ;SET UP RECORD SIZE IN PACKET
2571 037272 010465 000000        MOV      R4,TSDB(R5)  ;ISSUE COMMAND
2572 037276 004737 016150'  JSR      PC,WAITF     ;WAIT FOR SSR TO SET
2573 037302 016501 000002        MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
2574 037306 012702 100204        MOV      #SSR!SC!BIT2,R2 ;SET UP EXPECTED
2575 037312 020102          CMP      R1,R2        ;ARE THEY EQUAL
2576 037314 001406          BEQ      170$         ;BR, IF OK
2577 037316 005237 002212'  INC      FATFLG       ;BUMP COUNT
2581 037322          ERRHRD  ERRNO,T24TRL,EXPREC ;TSSR INCORRECT AFTER READ DATA
      037322 104456          TRAP    C$ERHRD
      037324 000667          .WORD  439
      037326 046164'        .WORD  T24TRL
      037330 015374'        .WORD  EXPREC
2582 037332          170$:  CKLOOP          ;LOOP IF SELECTED
      037332 104406          TRAP    C$CLP1
2583
2584          ;*****
    
```



```

2635 037440 004737 046454'      JSR    PC,T24RT3      ;SET UP OTHER COMMAND PACKET
2636 037444 004737 046320'      JSR    PC,T24REST    ;SET COMMAND PACKET
2637 037450 004737 046412'      JSR    PC,T24RT2     ;SET UP OTHER COMMAND PACKET
2638
2639 ;*****
2640 ;
2641 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
2642 ;
2643 ;*****
2644
2645 037454 004737 015674'      JSR    PC,SOFINIT    ;DO INITIALIZE ON CONTROLLER
2646 037460 103407                BCS    20$           ;BR IF INIT WAS OK
2647 037462 005237 002212'      INC    FATFLG        ;BUMP COUNT
2651 037466 010001                MOV    R0,R1         ;CONTENTS OF TSSR REGISTER
2652 037470                ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                TRAP    C$ERDF
                                .WORD   442
                                .WORD   SFIERR
                                .WORD   SFIMSG
    037470 104455
    037472 000672
    037474 003642'
    037476 011734'
2653 037500                20$:  MOV    UNITN,T24DSW      ;SET DRIVE NUMBER IN PACKET
2654 037500 013737 002172' 043650'  MOV    #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
2655 037506 012704 043630'
2656
2657 ;*****
2658 ;
2659 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
2660 ;
2661 ;*****
2662
2663 037512 004737 010562'      JSR    PC,WRTCHR     ;ISSUE WRITE CHARACTERISTICS
2664 037516 103407                BCS    24$           ;BR, IF COMMAND ISSUED OK
2665 037520 005237 002212'      INC    FATFLG        ;BUMP COUNT
2669 037524 010001                MOV    R0,R1         ;SAVE CONTENTS OF TSSR
2670 037526                ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP    C$ERHRD
                                .WORD   443
                                .WORD   WRTMSG
                                .WORD   SFIMSG
    037526 104456
    037530 000673
    037532 005046'
    037534 011734'
2671 037536                24$:  CKLOOP           ;LOOP IF SELECTED
    037536 104406                TRAP    C$CLP1
2672
2673 ;*****
2674 ;
2675 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2676 ;
2677 ;*****
2678
2679 037540 004737 010714'      JSR    PC,REWIND     ;CALL TAPE REWIND COMMAND
2680 037544 103407                BCS    30$           ;BR, IF NO PROBLEM
2681 037546 010001                MOV    R0,R1         ;SAVE TSSR
2682 037550 005237 002212'      INC    FATFLG        ;BUMP COUNT
2686 037554                ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD   444
                                .WORD   T24RWN
                                .WORD   PKTSSR
    037554 104456
    037556 000674
    037560 045116'
    037562 011746'
2687 037564                30$:  CKLOOP           ;LOOP IF SELECTED
    
```

```

037564 104406
2688 037566 012703 000400          MOV    #256.,R3          ;RECORD SIZE          TRAP    C$CLP1
2689 037572 013737 003114' 043762'  MOV    FREE,T24RB      ;STARTING WRITE BUFFER ADDRESS
2690
2691 ;*****
2692 ;
2693 ;WRITE DATA,ACK,CVC=1 COMMAND
2694 ;
2695 ;*****
2696
2697 037600 012737 140005 043760'    MOV    #140005,T24PK3  ;WRITE DATA,ACK,CVC=1 COMMAND
2698 037606 012704 043760'          MOV    #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
2699 037612          65$:
2700 037612 010300          MOV    R3,R0           ;SET PATTERN IN CORRECT REGISTER
2701 037614 004737 017324'        JSR    PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
2702 037620 010337 043766'        MOV    R3,T24SZ        ;SET UP RECORD SIZE IN PACKET
2703 037624 010465 000000        MOV    R4,TSDB(R5)    ;ISSUE COMMAND
2704 037630 004737 016150'        JSR    PC,WAITF        ;WAIT FOR SSR TO SET
2705 037634 016501 000002        MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
2706 037640 012702 000200        MOV    #SSR,R2        ;SET UP EXPECTED
2707 037644 020102          CMP    R1,R2          ;ARE THEY EQUAL
2708 037646 001406          BEQ    75$            ;BR, IF OK
2709 037650 005237 002212'        INC    FATFLG         ;BUMP COUNT
2713 037654          ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
          037654 104456          TRAP    C$ERHRD
          037656 000675          .WORD  445
          037660 005103'          .WORD  WRERR
          037662 011746'          .WORD  PKTSSR
2714 037664          75$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
          037664 104406
2715 037666 005723          TST    (R3)+          ;BUMP RECORD SIZE
2716 037670 022703 000414        CMP    #268.,R3      ;END OF RECORD YET
2717 037674 001346          BNE    65$            ;BR, IF MORE RECORDS TO WRITE
2718 037676          80$:  CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
          037676 104406
2719 037700 005743          TST    -(R3)          ;SET BACK TO 512.
2720 037702 013737 003114' 043762'  MOV    FREE,T24RB      ;STARTING READ BUFFER ADDRESS
2721
2722 ;*****
2723 ;
2724 ;READ REVERSE DATA,ACK COMMAND
2725 ;
2726 ;*****
2727
2728 037710 012737 100401 043760'    MOV    #100401,T24PK3  ;READ REVERSE DATA,ACK COMMAND
2729 037716 012704 043760'        MOV    #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
2730 037722 010337 043766'        MOV    R3,T24SZ        ;SET UP RECORD SIZE IN PACKET
2731 037726 010465 000000        MOV    R4,TSDB(R5)    ;ISSUE COMMAND
2732 037732 004737 016150'        JSR    PC,WAITF        ;WAIT FOR SSR TO SET
2733 037736 016501 000002        MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
2734 037742 012702 000200        MOV    #SSR,R2        ;SET UP EXPECTED
2735 037746 020102          CMP    R1,R2          ;ARE THEY EQUAL
2736 037750 001406          BEQ    170$          ;BR, IF OK
2737 037752 005237 002212'        INC    FATFLG         ;BUMP COUNT
2741 037756          ERRHRD  ERRNO,T24WDC,PKTSSR ;TSSR INCORRECT AFTER READ DATA
          037756 104456          TRAP    C$ERHRD
          037760 000676          .WORD  446

```



```

2789
2790 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
2791 ;
2792 ;*****
2793
2794 040110 004737 015674' JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
2795 040114 103407 BCS 20$ ;BR IF INIT WAS OK
2796 040116 005237 002212' INC FATFLG ;BUMP COUNT
2800 040122 010001 MOV R0,R1 ;CONTENTS OF TSSR REGISTER
2801 040124 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
; TRAP C$ERDF
; .WORD 448
; .WORD SFIERR
; .WORD SFIMSG
2802 040134 20$:
2803 040134 013737 002172' 043650' MOV UNITN,T24DSW ;SET DRIVE NUMBER IN PACKET
2804 040142 012704 043630' MOV #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
;*****
2806 ;
2807 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
2808 ;
2809 ;*****
2810 ;
2811 ;
2812 040146 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
2813 040152 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
2814 040154 005237 002212' INC FATFLG ;BUMP COUNT
2818 040160 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
2819 040162 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
; TRAP C$ERHRD
; .WORD 449
; .WORD WRTMSG
; .WORD SFIMSG
2820 040172 24$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
; .WORD 104406
2821 ;
2822 ;*****
2823 ;
2824 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2825 ;
2826 ;*****
2827 ;
2828 040174 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
2829 040200 103407 BCS 30$ ;BR, IF NO PROBLEM
2830 040202 010001 MOV R0,R1 ;SAVE TSSR
2831 040204 005237 002212' INC FATFLG ;BUMP COUNT
2835 040210 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERHRD
; .WORD 450
; .WORD T24RWN
; .WORD PKTSSR
2836 040220 30$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
; .WORD 104406
2837 040222 012703 000400 MOV #256.,R3 ;RECORD SIZE
2838 040226 013737 003114' 043762' MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
2839 ;
2840 ;*****
    
```



```

2841
2842 ;WRITE DATA,ACK,CVC=1,SWB COMMAND
2843 ;
2844 ;*****
2845
2846 040234 012737 150005 043760'      MOV      #150005,T24PK3      ;WRITE DATA,ACK,CVC=1,SWB COMMAND
2847 040242 012704 043760'      MOV      #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
2848 040246                                65$:
2849 040246 010300                                MOV      R3,R0              ;SET PATTERN IN CORRECT REGISTER
2850 040250 004737 017324'      JSR      PC,FILLMEM        ;FILL MEMORY WITH RECORD SIZE
2851 040254 010337 043766'      MOV      R3,T24SZ          ;SET UP RECORD SIZE IN PACKET
2852 040260 010465 000000      MOV      R4,T24SZ          ;ISSUE COMMAND
2853 040264 004737 016150'      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
2854 040270 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
2855 040274 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
2856 040300 020102                                CMP      R1,R2             ;ARE THEY EQUAL
2857 040302 001406                                BEQ      75$               ;BR, IF OK
2858 040304 005237 002212'      INC      FATFLG            ;BUMP COUNT
2862 040310                                ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERHRD
                                .WORD   451
                                .WORD   WRTERR
                                .WORD   PKTSSR
2863 040320                                75$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP    C$CLP1
2864 040322 005723                                TST      (R3)+             ;BUMP RECORD SIZE
2865 040324 022703 000414      CMP      #268.,R3         ;END OF RECORD YET
2866 040330 001346                                BNE     65$               ;BR, IF MORE RECORDS TO WRITE
2867 040332                                80$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP    C$CLP1
2868 040334 005743                                TST      -(R3)            ;SET RECORD SIZE BACK TO 512.
2869 040336 013737 003114' 043762'  MOV      FREE,T24RB        ;STARTING READ BUFFER ADDRESS
2870
2871 ;*****
2872 ;
2873 ;READ REVERSE DATA,ACK,SWB COMMAND
2874 ;
2875 ;*****
2876
2877 040344 012737 110401 043760'      MOV      #110401,T24PK3    ;READ REVERSE DATA,ACK,SWB COMMAND
2878 040352 012704 043760'      MOV      #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
2879 040356 010337 043766'      MOV      R3,T24SZ          ;SET UP RECORD SIZE IN PACKET
2880 040362 010465 000000      MOV      R4,T24SZ          ;ISSUE COMMAND
2881 040366 004737 016150'      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
2882 040372 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
2883 040376 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
2884 040402 020102                                CMP      R1,R2             ;ARE THEY EQUAL
2885 040404 001406                                BEQ      170$             ;BR, IF OK
2886 040406 005237 002212'      INC      FATFLG            ;BUMP COUNT
2890 040412                                ERRHRD  ERRNO,T24WDC,EXPREC ;TSSR INCORRECT AFTER READ DATA
                                TRAP    C$ERHRD
                                .WORD   452
                                .WORD   T24WDC
                                .WORD   EXPREC
2891 040422                                170$: CKLOOP              ;LOOP IF SELECTED
                                TRAP    C$CLP1
2892 040424 013702 003114'      MOV      FREE,R2          ;GET BUFFER ADDRESS
    
```



```

2943 040550 103407          BCS      20:          ;BR IF INIT WAS OK
2944 040552 005237 002212' INC      FATFLG      ;BUMP COUNT
2948 040556 010001          MOV      R0,R1       ;CONTENTS OF TSSR REGISTER
2949 040560          ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      040560 104455          TRAP    C:ERDF
      040562 000706          .WORD  454
      040564 003642'        .WORD  SFIERR
      040566 011734'        .WORD  SFIMSG
2950 040570          20:
2951 040570 013737 002172' 043650' MOV      UNITN,T24DSW ;SET DRIVE NUMBER IN PACKET
2952 040576 012704 043630' MOV      @T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
2953
2954 ;.....
2955 ;
2956 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
2957 ;
2958 ;.....
2959
2960 040602 004737 010562'      JSR      PC,WRTPHR   ;ISSUE WRITE CHARACTERISTICS
2961 040606 103407          BCS      24:          ;BR, IF COMMAND ISSUED OK
2962 040610 005237 002212' INC      FATFLG      ;BUMP COUNT
2966 040614 010001          MOV      R0,R1       ;SAVE CONTENTS OF TSSR
2967 040616          ERRHRD  ERRNO,WRTPMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
      040616 104456          TRAP    C:ERHRD
      040620 000707          .WORD  455
      040622 005046'        .WORD  WRTPMSG
      040624 011734'        .WORD  SFIMSG
2968 040626          24:   CKLOOP          ;LOOP IF SELECTED
      040626 104406          TRAP    C:CLP1
2969
2970 ;.....
2971 ;
2972 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
2973 ;
2974 ;.....
2975
2976 040630 004737 010714'      JSR      PC,REWIND   ;CALL TAPE REWIND COMMAND
2977 040634 004737 016236'      JSR      PC,CHKTSSR ;SEE HOW TSSR IS
2978 040640 103407          BCS      30:          ;BR, IF NO PROBLEM
2979 040642 010001          MOV      R0,R1       ;SAVE TSSR
2980 040644 005237 002212' INC      FATFLG      ;BUMP COUNT
2984 040650          ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      040650 104456          TRAP    C:ERHRD
      040652 000710          .WORD  456
      040654 045116'        .WORD  T24RWN
      040656 011746'        .WORD  PKTSSR
2985 040660          30:   CKLOOP          ;LOOP IF SELECTED
      040660 104406          TRAP    C:CLP1
2986 040662 012703 001000      MOV      @512.,R3    ;RECORD SIZE
2987 040666 013737 003114' 043762' MOV      FREE,T24RB  ;STARTING WRITE BUFFER ADDRESS
2988
2989 ;.....
2990 ;
2991 ;WRITE DATA,ACK,CVC=1 COMMAND
2992 ;
2993 ;.....
2994
    
```

```

2995 040674 012737 140005 043760'      MOV      #140005,T24PK3      ;WRITE DATA,ACK,CVC=1 COMMAND
2996 040702 012704 043760'      MOV      #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
2997 040706                                65$:
2998 040706 010337 043766'      MOV      R3,T24SZ          ;SET UP RECORD SIZE IN PACKET
2999 040712 010465 000000      MOV      R4,TSDB(R5)       ;ISSUE COMMAND
3000 040716 004737 016150'      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
3001 040722 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
3002 040726 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
3003 040732 020102                                CMP      R1,R2             ;ARE THEY EQUAL
3004 040734 001406                                BEQ      75$               ;BR, IF OK
3005 040736 005237 002212'      INC      FATFLG            ;BUMP COUNT
3009 040742                                ERRHRD   ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERHRD
                                .WORD   457
                                .WORD   WRERR
                                .WORD   PKTSSR
                                TRAP    C$CLP1
                                .WORD   104456
                                .WORD   000711
                                .WORD   005103
                                .WORD   011746
3010 040752                                75$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
3011 040754 012703 000400      MOV      #256.,R3          ;SIZE OF RECORD
3012 040760 013737 003114' 043762'  MOV      FREE,T24RB        ;STARTING READ BUFFER ADDRESS
3013
3014 ;.....
3015 ;
3016 ;READ DATA,ACK COMMAND
3017 ;
3018 ;.....
3019
3020 040766 012737 100401 043760'      MOV      #100401,T24PK3    ;READ DATA,ACK COMMAND
3021 040774 012704 043760'      MOV      #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
3022 041000 010337 043766'      MOV      R3,T24SZ          ;SET UP RECORD SIZE IN PACKET
3023 041004 010465 000000      MOV      R4,TSDB(R5)       ;ISSUE COMMAND
3024 041010 004737 016150'      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
3025 041014 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
3026 041020 012702 100204      MOV      #SSR!SC!BIT2,R2   ;SET UP EXPECTED
3027 041024 020102                                CMP      R1,R2             ;ARE THEY EQUAL
3028 041026 001406                                BEQ      170$              ;BR, IF OK
3029 041030 005237 002212'      INC      FATFLG            ;BUMP COUNT
3033 041034                                ERRHRD   ERRNO,T24TRL,EXPREC ;TSSR INCORRECT AFTER READ DATA
                                TRAP    C$ERHRD
                                .WORD   458
                                .WORD   T24TRL
                                .WORD   EXPREC
                                TRAP    C$CLP1
                                .WORD   104456
                                .WORD   000712
                                .WORD   046164
                                .WORD   015374
3034 041044                                170$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
3035
3036 ;.....
3037 ;
3038 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
3039 ;
3040 ;.....
3041
3042 041046 013701 043660'      MOV      T24BFR+6,R1       ;GET MESSAGE BUFFER (XST0)
3043 041052 010102      MOV      R1,R2             ;SET UP EXPECTED
3044 041054 052702 010000      BIS      #BIT12,R2         ;SET THE RLL BIT IN EXPECTED
3045 041060 020102      CMP      R1,R2             ;ARE THEY EQUAL
3046 041062 001406      BEQ      180$              ;BR, IF EQUAL (ALL IS WELL)
3047 041064 005237 002212'      INC      FATFLG            ;BUMP COUNT
    
```



```

3098 ;*****
3099
3100 041206 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
3101 041212 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
3102 041214 005237 002212' INC FATFLG ;BUMP COUNT
3106 041220 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
3107 041222 ERRHRD ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICSC FAILED
    041222 104456 TRAP C$ERHRD
    041224 000715 .WORD 461
    041226 005046' .WORD WRTMSG
    041230 011734' .WORD SFMSG
3108 041232 24$: CKLOOP ;LOOP IF SELECTED
    041232 104406 TRAP C$CLP1
3109 ;*****
3110 ;
3111 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
3112 ;
3113 ;
3114 ;*****
3115
3116
3117 041234 005737 002216' TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW SWITCH
3118 041240 001024 BNE 27$ ;BR IF SWITCH IS ON
3119 041242 112737 000200 043771' MOVB #200,T24BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
3120 041250 112737 000010 043770' MOVB #10,T24BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
3121 041256 012704 043740' MOV #T24PK2,R4 ;WRITE SUBSYS MEM PACKET
3122 041262 010465 000000 MOV R4,T5DB(R5) ;ISSUE COMMAND
3123 041266 004737 016236' JSR PC,CHKTSSR ;WAIT FOR SSR
3124 041272 103407 BCS 28$ ;BR, IF NO ERROR
3125 041274 010001 MOV RO,R1 ;ERROR, SAVE TSSR
3126 041276 005237 002212' INC FATFLG ;BUMP COUNT
3130 041302 ERRHRD ERRNO,T22SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
    041302 104456 TRAP C$ERHRD
    041304 000716 .WORD 462
    041306 026130' .WORD T22SSR
    041310 011746' .WORD PKTSSR
3131 041312 27$:
3132 041312 28$: CKLOOP ;LOOP IF SELECTED
    041312 104406 TRAP C$CLP1
3133
3134
3135
3136 041314 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
3137 041320 004737 016236' JSR PC,CHKTSSR ;SEE HOW TSSR IS
3138 041324 103407 BCS 30$ ;BR, IF NO PROBLEM
3139 041326 010001 MOV RO,R1 ;SAVE TSSR
3140 041330 005237 002212' INC FATFLG ;BUMP COUNT
3144 041334 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
    041334 104456 TRAP C$ERHRD
    041336 000717 .WORD 463
    041340 045116' .WORD T24RWN
    041342 011746' .WORD PKTSSR
3145 041344 30$: CKLOOP ;LOOP IF SELECTED
    041344 104406 TRAP C$CLP1
3146 041346 012703 000005 MOV #5.,R3 ;NUMBER OF RECORDS
3147 041352 013737 003114' 043762' MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
3148 ;*****
    
```

```

3149
3150      ;WRITE DATA,ACK,CVC=1 COMMAND
3151      ;
3152      ;*****
3153
3154 041360 012737 140005 043760'      MOV      #140005,T24PK3      ;WRITE DATA,ACK,CVC=1 COMMAND
3155 041366 012704 043760'      MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
3156 041372
3157 041372 012737 000256 043766' 65$:  MOV      #256,T24SZ      ;SET UP RECORD SIZE IN PACKET
3158 041400 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
3159 041404 004737 016150'      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
3160 041410 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
3161 041414 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
3162 041420 020102      CMP      R1,R2      ;ARE THEY EQUAL
3163 041422 001406      BEQ      75$      ;BR, IF OK
3164 041424 005237 002212'      INC      FATFLG      ;BUMP COUNT
3168 041430      ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      041430 104456      TRAP      C$ERHRD
      041432 000720      .WORD    464
      041434 005103'      .WORD    WRERR
      041436 011746'      .WORD    PKTSSR
3169 041440      75$:  CKLOOP      ;LOOP IF SELECTED
      041440 104406      TRAP      C$CLP1
3170 041442 005303      DEC      R3      ;BUMP DOWN RECORD COUNTER
3171 041444 001352      BNE      65$      ;BR, IF NOT AT 5 RECORDS YET
3172 041446 012703 000400      MOV      #256.,R3      ;RECORD SIZE
3173 041452 012701 160000      MOV      #160000,R1      ;NXM LOW ADDRESS START
3174 041456 012702 177776      MOV      #177776,R2      ;LIMIT CHECK FOR NXM (HIGHEST)
3175 041462 004737 016276'      JSR      PC, NXM      ;LOOK FOR NXM ADDRESS
3176 041466 103046      BCC      180$      ;BR, IF NON FOUND
3177 041470 010137 003130'      MOV      R1,NXMLO      ;SET ADDRESS UP FOR TEST
3178 041474 013737 003130' 043762'  MOV      NXML0,T24RB      ;STARTING READ BUFFER ADDRESS
3179 041502 005037 043764'      CLR      T24RB+2      ;SET TO 16 BIT ADDRESSING
3180
3181      ;*****
3182      ;
3183      ;READ DATA,ACK COMMAND
3184      ;
3185      ;*****
3186
3187 041506 012737 100001 043760'      MOV      #100001,T24PK3      ;READ DATA,ACK COMMAND
3188 041514 012704 043760' 165$:  MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
3189 041520 012737 000400 043766'  MOV      #256.,T24SZ      ;SET UP RECORD SIZE IN PACKET
3190 041526 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
3191 041532 004737 016150'      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
3192 041536 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
3193 041542 012702 104210      MOV      #SSR!NXM!SC!BIT3,R2 ;SET UP EXPECTED
3194 041546 020102      CMP      R1,R2      ;ARE THEY EQUAL
3195 041550 001414      BEQ      170$      ;BR, IF OK
3196 041552 005237 043764'      INC      T24RB+2      ;SET TO NEXT HIGHER ADDRESSING MODE
3197 041556 023727 043764' 000004  CMP      T24RB+2,#4      ;DID WE OVERFLOW INTO 19 BITS
3198 041564 001353      BNE      165$      ;BR, IF STILL IN 16-18 BITS RANGE
3199 041566 005237 002212'      INC      FATFLG      ;BUMP COUNT
3203 041572      ERRHRD  ERRNO,T24NXM,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      041572 104456      TRAP      C$ERHRD
      041574 000721      .WORD    465
      041576 044221'      .WORD    T24NXM

```



```

042624 104402
3510 042626 004737 046454' JSR PC,T24RT3 ;SET COMMAND PACKET UP CLEAR TRAP C$BSUB
3511 042632 004737 046320' JSR PC,T24REST ;SET COMMAND PACKET
3512 042636 004737 046412' JSR PC,T24RT2 ;SET UP OTHER COMMAND PACKET
3513
3514 ;*****
3515 ;
3516 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
3517 ;
3518 ;*****
3519
3520 042642 004737 015674' JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
3521 042646 103407 BCS 20$ ;BR IF INIT WAS OK
3522 042650 005237 002212' INC FATFLG ;BUMP COUNT
3526 042654 010001 MOV RO,R1 ;CONTENTS OF TSSR REGISTER
3527 042656 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
042656 104455 TRAP C$ERDF
042660 000735 .WORD 477
042662 003642' .WORD SFIERR
042664 011734' .WORD SFIMSG
3528 042666 20$: MOV UNITN,T24DSW ;SET UP DRIVE NUMBER
3529 042666 013737 002172' 043650' MOV #T24PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
3530 042674 012704 043630'
3531
3532 ;*****
3533 ;
3534 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
3535 ;
3536 ;*****
3537
3538 042700 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
3539 042704 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
3540 042706 005237 002212' INC FATFLG ;BUMP COUNT
3544 042712 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
3545 042714 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
042714 104456 TRAP C$ERHRD
042716 000736 .WORD 478
042720 005046' .WORD WRTMSG
042722 011734' .WORD SFIMSG
3546 042724 24$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
042724 104406
3547
3548 ;*****
3549 ;
3550 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
3551 ;
3552 ;*****
3553
3554 042726 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
3555 042732 004737 016236' JSR PC,CHKTSSR ;SEE HOW TSSR IS
3556 042736 103407 BCS 30$ ;BR, IF NO PROBLEM
3557 042740 010001 MOV RO,R1 ;SAVE TSSR
3558 042742 005237 002212' INC FATFLG ;BUMP COUNT
3562 042746 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
042746 104456 TRAP C$ERHRD
042750 000737 .WORD 479
042752 045116' .WORD T24RWN
    
```

```

042754 011746'
3563 042756 30$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
042756 104406 ;TRAP C$CLP1
3564 042760 012703 000400 MOV #256.,R3 ;RECORD SIZE
3565 042764 013737 003114' 043762' MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
3566
3567 ;*****
3568 ;
3569 ;READ REVERSE DATA,ACK COMMAND
3570 ;
3571 ;*****
3572
3573 042772 012737 100401 043760' MOV #100401,T24PK3 ;READ REVERSE DATA,ACK COMMAND
3574 043000 012704 043760' MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
3575 043004
3576 043004 010337 043766' 65$: MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
3577 043010 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
3578 043014 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
3579 043020 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
3580 043024 012702 100206 MOV #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
3581 043030 020102 CMP R1,R2 ;ARE THEY EQUAL
3582 043032 001406 BEQ 75$ ;BR, IF OK
3583 043034 005237 002212' INC FATFLG ;BUMP COUNT
3587 043040 ERRHRD ERRNO,T24WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
043040 104456 TRAP C$ERHRD
043042 000740 .WORD 480
043044 044561' .WORD T24WDE
043046 011746' .WORD PKTSSR
3588 043050 75$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
043050 104406 TRAP C$CLP1
3589
3590 ;*****
3591 ;
3592 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
3593 ;
3594 ;*****
3595
3596 043052 013701 043660' MOV T24BFR+6,R1 ;GET MESSAGE BUFFER
3597 043056 010102 MOV R1,R2 ;SET UP EXPECTED
3598 043060 052702 002000 BIS #BIT10,R2 ;SET THE NEF BIT IN EXPECTED
3599 043064 020102 CMP R1,R2 ;ARE THEY EQUAL
3600 043066 001406 BEQ 180$ ;BR, IF EQUAL (ALL IS WELL)
3601 043070 005237 002212' INC FATFLG ;BUMP COUNT
3605 043074 ERRHRD ERRNO,T24NEF,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
043074 104456 TRAP C$ERHRD
043076 000741 .WORD 481
043100 044010' .WORD T24NEF
043102 015374' .WORD EXPREC
3606 043104 180$: CKLOOP TRAP C$CLP1
043104 104406 ;*****
3607 043106 ENDSUB ;***** END SUBTEST *****
043106 104403 L10067: TRAP C$ESUB
3608 043110 023727 002212' 000017 CMP FATFLG,#15. ;IS ERROR COUNT AT 25
3609 043116 103402 BLO 999$ ;BR, IF LESS THAN 25
3610 043120 004737 017104' JSR PC,CKDROP ;TRY TO DROP THE UNIT
3611 043124 999$:

```



```

3664
3665
3666
3667
3668
3669
3670 043226 004737 010714'      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
3671 043232 004737 016236'      JSR      PC,CHKTSSR     ;SEE HOW TSSR IS
3672 043236 103407                BCS      30$            ;BR, IF NO PROBLEM
3673 043240 010001                MOV      R0,R1         ;SAVE TSSR
3674 043242 005237 002212'      INC      FATFLG        ;BUMP COUNT
3678 043246                ERRMRD   ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    484
                                .WORD    T24RWN
                                .WORD    PKTSSR
                                043246 104456
                                043250 000744
                                043252 045116'
                                043254 011746'
3679 043256                30$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                043256 104406
3680 043260 012703 000400        MOV      #256.,R3      ;RECORD SIZE
3681 043264 013737 003114' 043762'  MOV      FREE,T24RB    ;STARTING WRITE BUFFER ADDRESS
3682
3683
3684
3685
3686
3687
3688
3689 043272 012737 140005 043760'  MOV      #140005,T24PK3 ;WRITE DATA,ACK,CVC-1 COMMAND
3690 043300 012704 043760'        MOV      #T24PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
3691 043304
3692 043304 010337 043766'        65$:   MOV      R3,T24SZ   ;SET UP RECORD SIZE IN PACKET
3693 043310 010465 000000        MOV      R4,TSDB(R5)  ;ISSUE COMMAND
3694 043314 004737 016150'        JSR      PC,WAITF      ;WAIT FOR SSR TO SET
3695 043320 016501 000002        MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
3696 043324 012702 000200        MOV      #SSR,R2     ;SET UP EXPECTED
3697 043330 020102                CMP      R1,R2        ;ARE THEY EQUAL
3698 043332 001406                BEQ     75$           ;BR, IF OK
3699 043334 005237 002212'      INC      FATFLG        ;BUMP COUNT
3703 043340                ERRMRD   ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    485
                                .WORD    WRERR
                                .WORD    PKTSSR
                                043340 104456
                                043342 000745
                                043344 005103'
                                043346 011746'
3704 043350                75$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                043350 104406
3705 043352 012703 000400        MOV      #256.,R3      ;RECORD SIZE
3706 043356 013737 003114' 043762'  MOV      FREE,T24RB    ;STARTING READ BUFFER ADDRESS
3707
3708
3709
3710
3711
3712
3713
3714 043364 012737 100401 043760'  165$:   MOV      #100401,T24PK3 ;READ REVERSE DATA,ACK COMMAND
3715 043372 012704 043760'        MOV      #T24PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
3716 043376 010337 043766'        MOV      R3,T24SZ     ;SET UP RECORD SIZE IN PACKET
    
```



```

3766 043602 103402          BLO  999$          ;BR, IF LESS THAN 25
3767 043604 004737 017104' JSR  PC,CKDROP    ;TRY TO DROP THE UNIT
3768 043610          999$:
3769          :
3770          :
3771          :
3772 043610 004737 016360' JSR  PC,TSTLOOP   ;DO WE NEED TO ITERATE TEST
3773 043614 103002          BCC  163$        ;BR, IF NO LOOP REQUIRED
3774 043616 000137 034312' JMP  T24LOOP     ;EXECUTE AGAIN
3775 043622          163$:
3776 043622          EXI'  TST          ;ALL DONE THIS TEST
      043622 104432          TRAP
      043624 002660          .WORD  C$EXIT
                                   L10052-.
3777
3778
3779          ;*
3780          ;LOCAL STORAGE FOR THIS TEST
3781          ;-
3782 043626          .BLKB  10-<.-TSV2&7>
3783          T24PACKET:
3784 043630          .WORD  100204      ;COMMAND PACKET FOR TEST
3785 043630 100204          .WORD  T24DATA    ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
3786 043632 043640'        .WORD  0          ;ADDRESS OF CHARACTERISTICS BLOCK
3787 043634 000000          .WORD  10.      ;STARTING VALUE OF BLOCK SIZE
3788 043636 000012          .WORD  T24BFR    ;CHARACTERISTICS DATA BLOCK
3789 043640          .WORD  0          ;ADDRESS OF MESSAGE BUFFER
3790 043640 043652'        .WORD  20.      ;LENGTH OF MESSAGE BUFFER
3791 043642 000000          .WORD  0          ;DRIVE SELECTION BITS 2-0
3792 043644 000024          .WORD  0          ;MESSAGE BUFFER
3793 043646 000000          T24DSW: .WORD  0
3794 043650 000000          T24BFR: .BLKW  25.
3795 043652
3796          ;
3797          ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
3798          ;
3799          .BLKB  10-<.-TSV2&7>
3800 043734          T24PK2:
3801 043740          .WORD  100206      ;WRITE SUB SYS MEM COMMAND, IE AND ACK
3802 043740 100206          .WORD  T24BF2    ;ADDRESS OF SELECT BLOCK DATA
3803 043740 043770'        .WORD  0          ;SIZE OF DATA PACKET
3804 043742 043770'        .WORD  6.
3805 043744 000000          .WORD  10-<.-TSV2&7>
3806 043746 000006          T24PK3:
3807          .WORD  100205
3808          T24RB:
3809 043750          .WORD  FREE        ;READ COMMAND, IE AND ACK
3810 043760          .WORD  0          ;ADDRESS OF WRITE BUFFER
3811 043760 100205          T24WB:
3812 043762 003114'        .WORD  0          ;SIZE OF BUFFER (EXTENT)
3813 043762 000000          T24SZ:
3814 043764 000000          .EVEN
3815 043766 000000
3816 043766
3817
3818          ;
3819          ;
3820          ;
3821 043770          T24BF2:
3822 043770          T24BS0: .BYTE  10      ;BSELO AREA
3823 043771          T24BS1: .BYTE  200     ;BSEL1 AREA
3824 043772 000000          T24S2: .WORD  0          ;SEL 2 AREA
3825 043774 000000          T24S3: .WORD  0          ;DATA AREA
3826
    
```

```

3827
3828
3829
3830
3831 043776 100005
3832 044000 100405
3833 044002 102005
3834 044004 177777
3835 044006 000000
3836
3837
3838
3839
3840
3841
3842 044010 116 105 106
3843 044062 122 111 102
3844 044132 124 123 123
3845 044221 124 123 123
3846 044305 124 123 123
3847 044362 111 154 154
3848 044446 111 154 154
3849 044527 122 105 101
3850 044561 124 123 123
3851 044633 124 141 160
3852 044700 104 141 164
3853 044766 122 105 101
3854 045043 124 123 123
3855 045116 122 145 167
3856 045165 122 101 115
3857 045240 124 123 123
3858 045305 104 162 151
3859 045360 124 123 123
3860 045446 124 123 123
3861 045517 103 126 103
3862 045572 124 123 102
3863 045643 127 122 111
3864 045732 122 145 141
3865 046014 122 145 141
3866 046076 122 145 163
3867 046164 122 145 141
3868 046252 102 141 163
3869
3870
3871
3872
3873
3874
3875
3876
3877 046320
3878 046320
3879 046324 012701 043630'
3880 046330 012721 100004
3881 046334 012721 043640'
3882 046340 005021
3883 046342 012721 000012

```

```

;
; .EVEN
; TAPE MOTION PACKET COMMAND VALUES
T24RN: .WORD 100005 ;READ DATA (NEXT)
T24WDR: .WORD 100405 ;READ DATA RETRY
T24CON: .WORD 102005 ;WRITE CONTINOUS
; .WORD 177777 ;END OF DATA
T24DLY: .WORD 0 ;DELAY STORAGE AREA

;+
; LOCAL TEXT MESSAGES FOR TEST
;-
T24NEF: .ASCIZ 'NEF Not Set After NON-EXECUTABLE FUNCTION'
T24LOR: .ASCIZ 'RIB Not Set After READ REVERSE Into BOT'
T24WDG: .ASCIZ 'TSSR Not Correct After Illegal Buffer Address Bits Set'
T24NXM: .ASCIZ 'TSSR Not Correct After NXM Memory Address In Packet'
T24WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
T24ILA: .ASCIZ 'Illegal Address Bits, Failed To Set ILA Bit In XST0'
T24LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
T24SSR: .ASCIZ 'READ COMMAND Not Accepted'
T24WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
T24BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
T24DTA: .ASCIZ 'Data Written To Tape Not Equal To Data Read From Tape'
T24EOT: .ASCIZ 'READ DATA OVER EOT GAVE NO TAPE STATUS ALERT'
T24TM: .ASCIZ 'TSSR Not Correct After READ COMMAND Reject'
T24RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
T24RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
T24AM3: .ASCIZ 'TSSR Init. Failed After READ COMMAND'
T24OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
T24WDD: .ASCIZ 'TSSR Not Correct After READ DATA Command, SWB Bit Set'
T24WDC: .ASCIZ 'TSSR Not Correct After READ DATA Command'
T24VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
T24BA: .ASCIZ 'TSBA Not Correct After READ DATA Command'
T24WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
T24LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
T24LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
T24PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
T24TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
TST24ID: .ASCIZ 'Basic Read Data (Forward and Reverse)'

; .EVEN
;+
; ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
; WRITE SUBSYSTEM MEMORY COMMAND
;-
T24REST:
SAVREG ;SAVE THE REGISTERS
MOV #T24PACKET,R1 ;START OF THE PACKET
MOV #100004,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK.
MOV #T24DATA,(R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
CLR (R1)+ ;EXTENDED ADDRESS
MOV #10.,(R1)+ ;SIZE OF DATA BLOCK IN BYTES

```

```

3884 046346 012721 043652'      MOV      #T24BFR,(R1)+      ;ADDRESS OF MESSAGE BUFFER
3885 046352 005021                CLR      (R1)+              ;
3886 046354 012721 000024      MOV      #20.,(R1)+        ;LENGTH OF MESSAGE BUFFER
3887 046360 005021                CLR      (R1)+              ;
3888 046362 012711 000000      MOV      #0,(R1)           ;SELECT DRIVE ZERO
3889 046366 012702 000030      MOV      #24.,R2           ;NUMBER OF LOCATIONS TO BE CLEARED
3890 046372 012762 177777 043652' 64$: MOV      #177777,T24BFR(R2) ;ALL ONES TO MESSAGE BUFFER
3891 046400 005742                TST      -(R2)              ;NEXT LOCATION
3892 046402 022702 000000      CMP      #0,R2             ;CHECK FOR END OF LOOP
3893 046406 001371                BNE      64$                ;KEEP GOING UNTIL DONE
3894 046410 000207                RTS      PC                  ;RETURN
3895
3896
3897 046412                T24RT2:
3898 046412                SAVREG                      ;SAVE THE REGISTERS
3899 046416 012701 043740'      MOV      #T24PK2,R1        ;START OF THE PACKET
3900 046422 012721 100206      MOV      #100206,(R1)+     ;WRITE SUBSYSTEM MEM. WITH ACK, IE
3901 046426 012721 043770'      MOV      #T24BF2,(R1)+    ;ADDRESS OF DATA BLOCK
3902 046432 005021                CLR      (R1)+              ;EXTENDED ADDRESS
3903 046434 012721 000006      MOV      #6.,(R1)+        ;SIZE OF DATA BLOCK IN BYTES
3904 046440 005021                CLR      (R1)+              ;
3905 046442 012701 043770'      MOV      #T24BF2,R1        ;POINT TO DATA SEL AREA
3906 046446 005021                CLR      (R1)+              ;
3907 046450 005011                CLR      (R1)               ;
3908 046452 000207                RTS      PC                  ;RETURN
3909 046454
3910 046454                T24RT3:
3911 046460 012701 043760'      SAVREG                      ;SAVE THE REGISTERS
3912 046464 012721 000000      MOV      #T24PK3,R1        ;START OF THE PACKET
3913 046470 012721 000000      MOV      #0,(R1)+         ;CLEAR AREA OUT
3914 046474 005021                MOV      #0,(R1)+         ;ADDRESS OF DATA BLOCK
3915 046476 012711 000000      CLR      (R1)+              ;EXTENDED ADDRESS
3916 046502 000207                MOV      #0,(R1)          ;SIZE OF DATA BLOCK IN BYTES
3917 046504                RTS      PC                  ;RETURN
      046504                ENDTST
      046504                L10052: TRAP C$ETST
3918
3919
3920                .SBTTL TEST 5: SPACE RECORDS
3921                ;*
3922                ;
3923                ;THIS TEST VERIFIES THAT THE SPACE RECORDS FORWARD AND SPACE
3924                ;RECORDS REVERSE POSITION COMMANDS OPERATE PROPERLY WHEN SPACING
3925                ;OVER NORMAL DATA RECORDS. OPERATION WHEN SPACING OVER TAPE MARKS
3926                ;IS VERIFIED IN A SUBSEQUENT TEST. THE BASIC WRITE DATA TEST
3927                ;SHOULD HAVE BEEN RUN SUCCESSFULLY FOR THIS TEST TO PRODUCE MEANINGFUL
3928                ;RESULTS. THIS TEST CONSISTS OF A SERIES OF SUBTESTS. IN EACH
3929                ;OF THE SUBTESTS, THE TAPE IS ENTIRELY WRITTEN WITH RECORDS
3930                ;OF VARYING SIZES AND DATA CONTENT; THE FIRST 4 BYTES OF EACH
3931                ;RECORD INDICATE THAT RECORD'S RELATIVE POSITION ON TAPE. AFTER
3932                ;EACH SPACING OPERATION, THE TAPE POSITION IS VERIFIED BY READING
3933                ;THE NEXT OR PREVIOUS RECORD AND COMPARING THE POSITION DATA WITH
3934                ;THE EXPECTED RESULT.
3935                ;
3936                ;THE TEST CONSISTS OF THE FOLLOWING 8 SUBTESTS
3937                ;
3938                ;

```

```

3939
3940
3941 046506          ;
      046506          ; -
                        BGNTST
3942 046506 012737 006256' 002170'      MOV    #EPRT2,EPRTSW      ;SECONDARY ERROR MESSAGE
3947 046514 004737 017176'              JSR    PC,KTOFF           ;DON'T NEED LOTS OF MEMORY
3948 046520 012700 055230'              MOV    #TST25ID,R0       ;ASCII MESSAGE TO IDENTIFY TEST
3949 046524 004737 016412'              JSR    PC,TSTSETUP        ;DO INITIAL TEST SETUP
3950 046530 012737 000005 002206'      MOV    #5,LOOPCNT        ;PERFORM 5 ITERATIONS
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963 046536          ;
3964 046536          ;
      046536          ;
      104402          ;
3965 046540 004737 055246'              JSR    PC,T25REST        ;SET COMMAND PACKET
3966 046544 005037 054060'              CLR    T25CNT            ;CLEAR THE RECORD COUNTER AREA
3967 046550 004737 055340'              JSR    PC,T25RT2         ;SET UP OTHER COMMAND PACKET
3968 046554 004737 055402'              JSR    PC,T25RT3         ;SET UP OTHER COMMAND PACKET
3969 046560 012737 176750 054062'      MOV    #65000.,T25DLY    ;SET UP LOOP COUNTER
3970
3971
3972
3973
3974
3975
3976
3977 046566 004737 015674'      5$:  JSR    PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
3978 046572 103427              BCS   10$                ;BR IF INIT WAS OK
3979 046574              DELAY  250                ;DELAY IF REQUIRED
      046574 012727 000250              MOV    #250,(PC)+
      046600 000000              .WORD 0
      046602 013727 002116'              MOV    L$DLY,(PC)+
      046606 000000              .WORD 0
      046610 005367 177772              DEC    -6(PC)
      046614 001375              BNE   -.4
      046616 005367 177756              DEC    -22(PC)
      046622 001367              BNE   .-20
3980 046624 005337 054062'      DEC    T25DLY            ;DEC DELAY COUNTER
3981 046630 001356              BNE   5$                 ;BR, IF LOOP IS REQUIRED
3982 046632 005237 002212'      INC    FATFLG            ;BUMP COUNT
3986 046636 016501 000002      MOV    TSSR(R5),R1       ;CONTENTS OF TSSR REGISTER
3987 046642          ERRDF   ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      046642 104455              TRAP  C$ERDF
      046644 000765              .WORD 501
      046646 003642'              .WORD SFIERR
      046650 011734'              .WORD SFIMSG

```

```

3988 046652
3989 046652 013737 002172' 053720' 10$:
3990 046660 012704 053700'
3991
3992
3993
3994
3995
3996
3997
3998 046664 004737 010562'
3999 046670 103407
4000 046672 005237 002212'
4004 046676 010001
4005 046700
      046700 104456
      046702 000766
      046704 005046'
      046706 011734'
4006
4007
4008
4009
4010
4011
4012
4013 046710
      046710 104406
4014 046712 004737 010714'
4015 046716 103407
4016 046720 010001
4017 046722 005237 002212'
4021 046726
      046726 104456
      046730 000767
      046732 055035'
      046734 011746'
4022 046736
      046736 104406
4023
4024
4025
4026
4027
4028
4029
4030 046740 013701 053730'
4031 046744 010102
4032 046746 052702 000002
4033 046752 020102
4034 046754 001406
4035 046756 005237 002212'
4039 046762
      046762 104456
      046764 000770
      046766 054225'
      046770 015374'

      MOV UNITN,T25DSW ;SET UP DRIVE NUMBER
      MOV #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS

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

      JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
      BCS 15$ ;BR, IF COMMAND ISSUED OK
      INC FATFLG ;BUMP COUNT
      MOV RO,R1 ;SAVE CONTENTS OF TSSR
      ERRHRD ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTIC FAILED

      TRAP C$ERHRD
      .WORD 502
      .WORD WRTMSG
      .WORD SFMSG

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

15$: CKLOOP
      JSR PC,REWIND ;CALL TAPE REWIND COMMAND TRAP C$CLP1
      BCS 30$ ;BR, IF NO PROBLEM
      MOV RO,R1 ;SAVE TSSR
      INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED

      TRAP C$ERHRD
      .WORD 503
      .WORD T25RWN
      .WORD PKTSSR

30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1

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

      MOV T25BFR+6,R1 ;PICK UP XSTO
      MOV R1,R2 ;SET UP EXPECTED
      BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
      CMP R1,R2 ;DOES EXP = REC'D
      BEQ 40$ ;BR, IF EQUAL (OK)
      INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND

      TRAP C$ERHRD
      .WORD 504
      .WORD T25BOT
      .WORD EXPREC

```

```

4040 046772          40$:  CKLOOP                      ;LOOP IF SELECTED
      046772 104406          TRAP  C$CLP1
4041 046774 012703 000400      MOV  #256.,R3      ;RECORD SIZE
4042 047000 013737 003114' 054032'  MOV  FREE,T25RB   ;STARTING WRITE BUFFER ADDRESS
4043
4044 ;*****
4045 ;
4046 ;WRITE DATA,ACK,CVC=1 COMMAND
4047 ;
4048 ;*****
4049
4050 047006 012737 140005 054030'  MOV  #140005,T25PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
4051 047014 012704 054030'      MOV  #T25PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
4052 047020          65$:
4053 047020 010337 054036'      MOV  R3,T25SZ      ;SET UP RECORD SIZE IN PACKET
4054 047024 013777 054060' 134062  MOV  T25CNT,@FREE  ;LOAD UP RECORD COUNTER IN WRT BUFFER
4055 047032 062737 000001 054060'  ADD  #1,T25CNT     ;GET READY FOR NEXT RECORD
4056 047040 010465 000000      MOV  R4,T5DB(R5)   ;ISSUE COMMAND
4057 047044 004737 016150'      JSR  PC,WAITF      ;WAIT FOR SSR TO SET
4058 047050 016501 000002      MOV  T5SR(R5),R1   ;GET T5SR CONTENTS
4059 047054 012702 000200      MOV  #SSR,R2       ;SET UP EXPECTED
4060 047060 020102      CMP  R1,R2         ;ARE THEY EQUAL
4061 047062 001411      BEQ  75$           ;BR, IF OK
4062 047064 032701 000004      BIT  #BIT2,R1      ;CHECK FOR TAPE STATUS ALERT
4063 047070 001014      BNE  120$          ;BR, IF TSA IS SET (SUSPECT IS EOT)
4064 047072 005237 002212'      INC  FATFLG        ;BUMP COUNT
4068 047076          ERRHRD  ERRNO,WRTERR,PKTSSR ;T5SR INCORRECT AFTER WRITE DATA
      047076 104456          TRAP  C$ERHRD
      047100 000771          .WORD 505
      047102 005103'        .WORD WRTERR
      047104 011746'        .WORD PKTSSR
4069 047106          75$:  CKLOOP                      ;LOOP IF SELECTED
      047106 104406          TRAP  C$CLP1
4070 047110          INC  R3                          ;BUMP RECORD SIZE
4071 047112 022703 001000      CMP  #512.,R3     ;END OF RECORD YET
4072 047116 001340      BNE  65$           ;BR, IF MORE RECORDS TO WRITE
4073 047120 000415      BR   125$         ;ENOUGH RECORDS
4074 047122          120$:
4075 ;*****
4076 ;
4077 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
4078 ;
4079 ;*****
4080
4081
4082 047122 013701 053730'      MOV  T25BFR+6,R1   ;QUICK CHECK FOR EOT SET
4083 047126 010102      MOV  R1,R2         ;SET UP EXPECTED
4084 047130 052702 000001      BIS  #BIT0,R2      ;SET THE EOT BIT XSTO
4085 047134 020102      CMP  R1,R2         ;IS THE EOT BIT SET IN XSTO
4086 047136 001406      BEQ  125$         ;BR, IF SET (GOOD)
4087 047140 005237 002212'      INC  FATFLG        ;BUMP COUNT
4091 047144          ERRDF  ERRNO,T25NET,EXPREC ;DEVICE FATAL NOT EOT FOUND ETC.
      047144 104455          TRAP  C$ERDF
      047146 000772          .WORD 506
      047150 054361'        .WORD T25NET
      047152 015374'        .WORD EXPREC
4092 047154          125$:

```



```

4093
4094
4095
4096
4097
4098
4099
4100 047154 004737 010714'      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
4101 047160 103407                BCS      130$              ;BR, IF NO PROBLEM
4102 047162 010001                MOV      R0,R1            ;SAVE TSSR
4103 047164 005237 002212'      INC      FATFLG          ;BUMP COUNT
4107 047170                ERRHRD  ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    507
                                .WORD    T25RWN
                                .WORD    PKTSSR
4108 047200                130$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
4109 047202 012737 000007 053720'  MOV      #7,T25DSW        ;SET UP DRIVE NUMBER
4110 047210 012704 053700'      MOV      #T25PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
4111
4112
4113
4114
4115
4116
4117
4118 047214 004737 010562'      JSR      PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
4119 047220 103407                BCS      140$              ;BR, IF COMMAND ISSUED OK
4120 047222 005237 002212'      INC      FATFLG          ;BUMP COUNT
4124 047226 010001                MOV      R0,R1            ;SAVE CONTENTS OF TSSR
4125 047230                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP      C$ERHRD
                                .WORD    508
                                .WORD    WRTMSG
                                .WORD    SFIMSG
4126 047240                140$:  CKLOOP              ;SCOPE LOOP
                                TRAP      C$CLP1
4127 047242 005737 002216'      TST      EXTFEA          ;CHECK FOR EXTENDED FEATURES
4128 047246 001044                BNE     160$              ;BR IF SWITCH IS ON
4129
4130 047250 112737 000200 054041'  MOVB    #200,T25BS1      ;WRITE MISCELLANEOUS CONT/READ STATUS
4131 047256 112737 000010 054040'  MOVB    #10,T25BS0      ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4132 047264 012704 054010'      MOV      #T25PK2,R4     ;WRITE SUBSYS MEM PACKET
4133 047270 010465 000000                MOV      R4,TSDB(R5)     ;ISSUE COMMAND
4134 047274 004737 016236'      JSR      PC,CHKTSSR      ;WAIT FOR SSR
4135 047300 103407                BCS      150$              ;BR, IF NO ERROR
4136 047302 010001                MOV      R0,R1            ;ERROR, SAVE TSSR
4137 047304 005237 002212'      INC      FATFLG          ;BUMP COUNT
4141 047310                ERRHRD  ERRNO,T25SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
                                TRAP      C$ERHRD
                                .WORD    509
                                .WORD    T25SSR
                                .WORD    PKTSSR
4142 047320                150$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
4143 047322 012737 000007 053720'  MOV      #7,T25DSW        ;SET UP DRIVE NUMBER

```



```

4246
4247
4248
4249
4250
4251
4252
4253 047612 150: CKLOOP
      047612 104406
4254 047614 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND TRAP C:CLP1
4255 047620 103407 BCS 300 ;BR, IF NO PROBLEM
4256 047622 010001 MOV RO,R1 ;SAVE TSSR
4257 047624 005237 002212' INC FATFLG ;BUMP COUNT
4261 047630 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      047630 104456
      047632 001003 TRAP C:ERHRD
      047634 055035' .WORD 515
      047636 011746' .WORD T25RWN
      .WORD PKTSSR
4262 047640 300: CKLOOP ;LOOP IF SELECTED
      047640 104406
4263 047642 005737 002216' 1400: TST EXTFEA ;CHECK FOR EXTENDED FEATURES SW TRAP C:CLP1
4264 047646 001044 BNE 1600 ;BR IF SWITCH IS ON
4265
4266 047650 112737 000200 054041' MOVB #200,T25BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
4267 047656 112737 000010 054040' MOVB #10,T25BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4268 047664 012704 054010' MOV #T25PK2,R4 ;WRITE SUBSYS MEM PACKET
4269 047670 010465 000000 MOV R4,T5DB(R5) ;ISSUE COMMAND
4270 047674 004737 016236' JSR PC,CHKTSSR ;WAIT FOR SSR
4271 047700 103407 BCS 1500 ;BR, IF NO ERROR
4272 047702 010001 MOV RO,R1 ;ERROR, SAVE TSSR
4273 047704 005237 002212' INC FATFLG ;BUMP COUNT
4277 047710 ERRHRD ERRNO,T25SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      047710 104456 TRAP C:ERHRD
      047712 001004 .WORD 516
      047714 054064' .WORD T25SSR
      047716 011746' .WORD PKTSSR
4278 047720 1500: CKLOOP ;LOOP IF SELECTED
      047720 104406 TRAP C:CLP1
4279 047722 012737 000007 053720' MOV #7,T25DSW ;SET UP DRIVE NUMBER
4280 047730 012704 053700' MOV #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4281
4282
4283
4284
4285
4286
4287
4288 047734 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
4289 047740 103407 BCS 1600 ;BR, IF COMMAND ISSUED OK
4290 047742 005237 002212' INC FATFLG ;BUMP COUNT
4294 047746 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
4295 047750 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      047750 104456 TRAP C:ERHRD
      047752 001005 .WORD 517
      047754 005046' .WORD WRTMSG
      047756 011734' .WORD SFIMSG
4296 047760 1600: CKLOOP ;SCOPE LOOP

```



```

4345 050114 004737 055246'      JSR    PC,T25REST      ;SET COMMAND PACKET
4346 050120 004737 055340'      JSR    PC,T25RT2      ;SET UP OTHER COMMAND PACKET
4347 050124 004737 055402'      JSR    PC,T25RT3      ;SET UP OTHER COMMAND PACKET
4348
4349
4350
4351
4352
4353
4354
4355 050130 004737 015674'      JSR    PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
4356 050134 103407                BCS    10$             ;BR IF INIT WAS OK
4357 050136 005237 002212'      INC    FATFLG          ;BUMP COUNT
4361 050142 010001                MOV    RO,R1           ;CONTENTS OF TSSR REGISTER
4362 050144                ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                TRAP   C$ERDF
                                .WORD  520
                                .WORD  SFIERR
                                .WORD  SFIMSG
    050144 104455
    050146 001010
    050150 003642'
    050152 011734'
4363 050154 013737 002172' 053720' 10$:  MOV    UNITN,T25DSW      ;SET UP DRIVE NUMBER
4364
4365 050162 012704 053700'      MOV    @T25PACKET,R4  ;SUBROUTINE NEEDS PACKET ADDRESS
4366
4367
4368
4369
4370
4371
4372
4373 050166 004737 010562'      JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
4374 050172 103407                BCS    15$             ;BR, IF COMMAND ISSUED OK
4375 050174 005237 002212'      INC    FATFLG          ;BUMP COUNT
4379 050200 010001                MOV    RO,R1           ;SAVE CONTENTS OF TSSR
4380 050202                ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
                                TRAP   C$ERHRD
                                .WORD  521
                                .WORD  WRTMSG
                                .WORD  SFIMSG
    050202 104456
    050204 001011
    050206 005046'
    050210 011734'
4381
4382
4383
4384
4385
4386
4387
4388 050212                15$:  CKLOOP
    050212 104406
4389 050214 004737 010714'      JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
4390 050220 103407                BCS    30$             ;BR, IF NO PROBLEM
4391 050222 010001                MOV    RO,R1           ;SAVE TSSR
4392 050224 005237 002212'      INC    FATFLG          ;BUMP COUNT
4396 050230                ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP   C$ERHRD
                                .WORD  522
                                .WORD  T25RWN
                                .WORD  PKTSSR
    050230 104456
    050232 001012
    050234 055035'
    050236 011746'
4397 050240                30$:  CKLOOP      ;LOOP IF SELECTED

```

```

050240 104406
4398
4399
4400
4401
4402
4403
4404
4405 050242 013701 053730'
4406 050246 010102
4407 050250 052702 000002
4408 050254 020102
4409 050256 001406
4410 050260 005237 002212'
4414 050264
      050264 104456
      050266 001013
      050270 054225'
      050272 015374'
4415 050274
      050274 104406
4416 050276 012737 000001 054032'
4417
4418
4419
4420
4421
4422
4423
4424 050304 012737 140010 054030'
4425 050312 012704 054030'
4426 050316
4427 050316 010465 000000
4428 050322 004737 016150'
4429 050326 016501 000002
4430 050332 012702 000200
4431 050336 020102
4432 050340 001411
4433 050342 032701 000004
4434 050346 001006
4435 050350 005237 002212'
4439 050354
      050354 104456
      050356 001014
      050360 054145'
      050362 015374'
4440 050364
      050364 104406
4441 050366
4442
4443
4444
4445
4446
4447
4448
4449 050366 013701 053730'

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

      MOV      T25BFR+6,R1      ;PICK UP XSTO
      MOV      R1,R2           ;SET UP EXPECTED
      BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
      CMP      R1,R2           ;DOES EXP = REC'D
      BEQ      40$             ;BR, IF EQUAL (OK)
      INC      FATFLG          ;BUMP COUNT
      ERRHRD   ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                     TRAP   C$ERHRD
                                     .WORD  523
                                     .WORD  T25BOT
                                     .WORD  EXPREC
40$:   CKLOOP                  ;LOOP IF SELECTED
      MOV      @000001,T25RB    ;NUMBER OF RECORDS TO SPACE OVER
                                     TRAP   C$CLP1

;*****
;
;SPACE FORWARD,ACK,CVC=1 COMMAND
;
;*****

      MOV      @140010,T25PK3   ;SPACE FORWARD,ACK,CVC=1 COMMAND
      MOV      @T25PK3,R4       ;SET UP R4 WITH PACKET ADDRESS
65$:   MOV      R4,TSDB(R5)     ;ISSUE COMMAND
      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
      MOV      @SSR,R2         ;SET UP EXPECTED
      CMP      R1,R2           ;ARE THEY EQUAL
      BEQ      75$             ;BR, IF OK
      BIT      @BIT2,R1        ;CHECK FOR TAPE STATUS ALERT
      BNE      75$             ;BR, IF TSA IS SET (SUSPECT IS EOT)
      INC      FATFLG          ;BUMP COUNT
      ERRHRD   ERRNO,T25WDE,EXPREC ;TSSR INCORRECT AFTER READ DATA
                                     TRAP   C$ERHRD
                                     .WORD  524
                                     .WORD  T25WDE
                                     .WORD  EXPREC
75$:   CKLOOP                  ;LOOP IF SELECTED
                                     TRAP   C$CLP1
120$:

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

      MOV      T25BFR+6,R1      ;QUICK CHECK FOR BOT SET

```



```

4551 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
4552 ;
4553 ;*****
4554
4555 050670 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
4556 050674 103407 BCS 30$ ;BR, IF NO PROBLEM
4557 050676 010001 MOV R0,R1 ;SAVE TSSR
4558 050700 005237 002212' INC FATFLG ;BUMP COUNT
4562 050704 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      050704 104456 TRAP C$ERHRD
      050706 001022 .WORD 530
      050710 055035' .WORD T25RWN
      050712 011746' .WORD PKTSSR
4563 050714 30$: CKLOOP ;LOC IF SELECTED TRAP C$CLP1
      050714 104406
4564
4565 ;*****
4566 ;
4567 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
4568 ;
4569 ;*****
4570
4571 050716 013701 053730' MOV T25BFR+6,R1 ;PICK UP XSTO
4572 050722 010102 MOV R1,R2 ;SET UP EXPECTED
4573 050724 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
4574 050730 020102 CMP R1,R2 ;DOES EXP = REC'D
4575 050732 001406 BEQ 40$ ;BR, IF EQUAL (OK)
4576 050734 005237 002212' INC FATFLG ;BUMP COUNT
4580 050740 ERRHRD ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      050740 104456 TRAP C$ERHRD
      050742 001023 .WORD 531
      050744 054225' .WORD T25BOT
      050746 015374' .WORD EXPREC
4581 050750 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      050750 104406
4582
4583 ;*****
4584 ;
4585 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
4586 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
4587 ;
4588 ;*****
4589
4590 050752 012703 000001 MOV #000001,R3 ;NUMBER OF RECORDS TO SPACE FORWARD
4591 050756 004737 010366' JSR PC,SPACE ;CALL SPACE COMMAND
4592 050762 103410 BCS 50$ ;CHECK FOR ERROR
4593 050764 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4594 050770 005237 002212' INC FATFLG ;BUMP COUNT
4598 050774 ERRHRD ERRNO,T25WDE,SFFMSG ;SPACE FORWARD FAILED
      050774 104456 TRAP C$ERHRD
      050776 001024 .WORD 532
      051000 054145' .WORD T25WDE
      051002 012002' .WORD SFFMSG
4599 051004 50$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      051004 104406
4600 051006 012737 000001 054032' MOV #1,T25RB ;NUMBER OF RECORDS TO SPACE OVER
4601

```

```

4602 ;*****
4603 ;
4604 ;SPACE REVERSE,ACK,CVC=1 COMMAND
4605 ;
4606 ;*****
4607
4608 051014 012737 140410 054030'      MOV      #140410,T25PK3      ;SPACE REVERSE,ACK,CVC=1 COMMAND
4609 051022 012704 054030'      MOV      #T25PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4610 051026      65$:
4611 051026 010465 000000      MOV      R4,TSDB(R5)       ;ISSUE COMMAND
4612 051032 004737 016150'      JSR      PC,WAITF         ;WAIT FOR SSR TO SET
4613 051036 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
4614 051042 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED
4615 051046 020102      CMP      R1,R2           ;ARE THEY EQUAL
4616 051050 001406      BEQ      75$             ;BR, IF OK
4617 051052 005237 002212'      INC      FATFLG          ;BUMP COUNT
4621 051056      ERRHRD  ERRNO,T25WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      051056 104456      TRAP    C$ERHRD
      051060 001025      .WORD  533
      051062 054145'    .WORD  T25WDE
      051064 011746'    .WORD  PKTSSR
4622 051066      75$:  CKLOOP          ;LOOP IF SELECTED
      051066 104406      TRAP    C$CLP1
4623 051070      120$:
4624 051070 012703 000400      MOV      #256.,R3        ;RECORD SIZE
4625 051074 013737 003114' 054032'    MOV      FREE,T25RB      ;STARTING READ BUFFER ADDRESS
4626 ;*****
4627 ;
4628 ;READ DATA,ACK,CVC=1 COMMAND
4629 ;
4630 ;*****
4631 ;
4632 ;
4633 051102 012737 140001 054030'      MOV      #140001,T25PK3   ;READ DATA,ACK,CVC=1 COMMAND
4634 051110 012704 054030'      MOV      #T25PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4635 051114 010337 054036'      MOV      R3,T25SZ        ;SET UP RECORD SIZE IN PACKET
4636 051120 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
4637 051124 004737 016150'      JSR      PC,WAITF         ;WAIT FOR SSR TO SET
4638 051130 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
4639 051134 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED
4640 051140 020102      CMP      R1,R2           ;ARE THEY EQUAL
4641 051142 001406      BEQ      170$            ;BR, IF OK
4642 051144 005237 002212'      INC      FATFLG          ;BUMP COUNT
4646 051150      ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      051150 104456      TRAP    C$ERHRD
      051152 001026      .WORD  534
      051154 005176'    .WORD  RDERR
      051156 011746'    .WORD  PKTSSR
4647 051160      170$:  CKLOOP          ;LOOP IF SELECTED
      051160 104406      TRAP    C$CLP1
4648 051162 017701 131726      MOV      @FREE,R1        ;GET FIRST WORD FROM BUFFER
4649 051166 012702 000000      MOV      #0,R2          ;SET UP EXPECTED
4650 051172 020102      CMP      R1,R2           ;WAS RECORD NUMBERED 1
4651 051174 001406      BEQ      200$            ;BR, IF CORRECT RECORD
4652 051176 005237 002212'      INC      FATFLG          ;BUMP COUNT
4656 051202      ERRHRD  ERRNO,T25WNG,EXPREC ;SHOULD HAVE BEEN RECORD NUMBER 1
      051202 104456      TRAP    C$ERHRD

```



```

4701 051346          ERRDF  ERRNO,SFIERR,SFIMSG  ;FATAL ERROR TSSR WAS NOT OK
      051346 104455          TRAP  C$ERDF
      051350 001030          .WORD 536
      051352 003642'        .WORD SFIERR
      051354 011734'        .WORD SFIMSG
4702 051356          20$:
4703 051356 013737 002172' 053720'  MOV  UNITN,T25DSW  ;SET UP UNIT NUMBER
4704 051364 012704 053700'  MOV  #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4705
4706 ;*****
4707 ;
4708 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
4709 ;
4710 ;*****
4711
4712 051370 004737 010562'  JSR  PC,WRTCHR  ;ISSUE WRITE CHARACTERISTICS
4713 051374 103407  BCS  25$      ;BR, IF COMMAND ISSUED OK
4714 051376 005237 002212'  INC  FATFLG    ;BUMP COUNT
4718 051402 010001  MOV  R0,R1     ;SAVE CONTENTS OF TSSR
4719 051404          ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      051404 104456          TRAP  C$ERHRD
      051406 001031          .WORD 537
      051410 005046'        .WORD WRTMSG
      051412 011734'        .WORD SFIMSG
4720 051414          25$:  CKLOOP          ;LOOP IF SELECTED
      051414 104406          TRAP  C$CLP1
4721
4722 ;*****
4723 ;
4724 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
4725 ;
4726 ;*****
4727
4728 051416 004737 010714'  JSR  PC,REWIND  ;CALL TAPE REWIND COMMAND
4729 051422 103407  BCS  30$      ;BR, IF NO PROBLEM
4730 051424 010001  MOV  R0,R1     ;SAVE TSSR
4731 051426 005237 002212'  INC  FATFLG    ;BUMP COUNT
4735 051432          ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
      051432 104456          TRAP  C$ERHRD
      051434 001032          .WORD 538
      051436 055035'        .WORD T25RWN
      051440 011746'        .WORD PKTSSR
4736 051442          30$:  CKLOOP          ;LOOP IF SELECTED
      051442 104406          TRAP  C$CLP1
4737 051444 013701 054056'  MOV  T25CN2,R1  ;NUMBER OF RECORDS ON TAPE
4738 051450 012702 177776  MOV  #65534.,R2 ;MAX IT CAN SPACE OVER
4739 051454 020201  CMP   R2,R1     ;WHICH VALUE CAN WE USE
4740 051456 003002  BGT  46$      ;BR, IF # WRITTEN > 64K
4741 051460 010103  MOV  R1,R3     ;# WRITTEN CAN BE USED
4742 051462 000401  BR   47$      ;MOVE ON
4743 051464 010203  46$:  MOV  R2,R3  ;USE MAX NUMBER
4744 051466 162703 000001  47$:  SUB  #1,R3   ;DON'T GO ALL THE WAY YET
4745 051472 010337 054032'  MOV  R3,T25RB  ;NUMBER OF RECORDS TO SPACE OVER
4746
4747 ;*****
4748 ;
4749 ;SPACE FORWARD,ACK,CVC=1 COMM/ND

```

```

4750
4751
4752
4753 051476 012737 140010 054030'      MOV      #140010,T25PK3      ;SPACE FORWARD,ACK,CVC=1 COMMAND
4754 051504 012704 054030'      MOV      #T25PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4755 051510
4756 051510 013737 054056' 054062' 65$:      MOV      T25CN2,T25DLY      ;NUMBER OF RECORDS USED AS DELAY COUNTER
4757 051516 010465 000000      MOV      R4,TSDB(R5)        ;ISSUE COMMAND
4758 051522 004737 016150'      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
4759 051526 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
4760 051532 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
4761 051536 020102      CMP      R1,R2            ;ARE THEY EQUAL
4762 051540 001425      BEQ      75$              ;BR, IF OK
4763 051542      DELAY    250              ;DELAY .25 SECONDS
      MOV      #250,(PC)+
      .WORD    0
      MOV      L$DLY,(PC)+
      .WORD    0
      DEC     -6(PC)
      BNE     .-4
      DEC     -22(PC)
      BNE     .-20
4764 051572 005337 054062'      DEC     T25DLY            ;BUMP DOWN COUNTER
4765 051576 001351      BNE     67$              ;BR, IF NOT AT END OF DELAY
4766 051600 005237 002212'      INC     FATFLG           ;BUMP COUNT
4770 051604      ERRHRD  ERRNO,T25WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      TRAP    C$ERHRD
      .WORD   539
      .WORD   T25WDE
      .WORD   PKTSSR
4771 051614      75$:      CKLOOP                    ;LOOP IF SELECTED
      TRAP    C$CLP1
4772 051616 012703 010000      MOV     #4096.,R3         ;RECORD SIZE
4773 051622 013737 003114' 054032'  MOV     FREE,T25RB        ;STARTING READ BUFFER ADDRESS
4774
4775
4776
4777
4778
4779
4780
4781 051630 012737 100001 054030'      MOV     #100001,T25PK3     ;READ DATA,ACK COMMAND
4782 051636 012704 054030' 165$:      MOV     #T25PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4783 051642 010337 054036'      MOV     R3,T25SZ          ;SET UP RECORD SIZE IN PACKET
4784 051646 010465 000000      MOV     R4,TSDB(R5)       ;ISSUE COMMAND
4785 051652 004737 016150'      JSR     PC,WAITF          ;WAIT FOR SSR TO SET
4786 051656 016501 000002      MOV     TSSR(R5),R1       ;GET TSSR CONTENTS
4787 051662 012702 000200      MOV     #SSR,R2          ;SET UP EXPECTED
4788 051666 020102      CMP     R1,R2            ;ARE THEY EQUAL
4789 051670 001411      BEQ     170$              ;BR, IF OK
4790 051672 032701 000004      BIT     #BIT2,R1          ;CHECK FOR TAPE STATUS ALERT
4791 051676 001006      BNE     170$              ;IF SET ALL IS WELL
4792 051700 005237 002212'      INC     FATFLG           ;BUMP COUNT
4796 051704      ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      TRAP    C$ERHRD
      .WORD   540
      .WORD   RDERR
051704 104456
051706 001034
051710 005176'

```



```

052044 000000
052046 005367 177772
052052 001375
052054 005367 177756
052060 001367
4843 052062 005337 054062'
4844 052066 001356
4845 052070 016501 000002
4846 052074 005237 002212'
4850 052100
052100 104455
052102 001036
052104 003642'
052106 011734'
4851 052110 013737 002172' 053720' 20$:
4852
4853 052116 012704 053700'
4854
4855
4856
4857
4858
4859
4860
4861 052122 004737 010562'
4862 052126 103407
4863 052130 005237 002212'
4867 052134 010001
4868 052136
052136 104456
052140 001037
052142 005046'
052144 011734'
4869 052146
052146 104406
4870
4871
4872
4873
4874
4875
4876
4877 052150 004737 010714'
4878 052154 103407
4879 052156 010001
4880 052160 005237 002212'
4884 052164
052164 104456
052166 001040
052170 055035'
052172 011746'
4885 052174
052174 104406
4886
4887
4888
4889

```

.WORD 0
 DEC -6(PC)
 BNE -4
 DEC -22(PC)
 BNE -20

DEC T25DLY ;DEC COUNTER
 BNE 10\$;BR, IF MORE LOOPS REQUIRED
 MOV TSSR(R5),R1 ;CONTENTS OF TSSR REGISTER
 INC FATFLG ;BUMP COUNT
 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK

TRAP C\$ERDF
 .WORD 542
 .WORD SFIERR
 .WORD SFIMSG

MOV UNITN,T25DSW ;SET UP UNIT NUMBER
 MOV #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS

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

JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
 BCS 25\$;BR, IF COMMAND ISSUED OK
 INC FATFLG ;BUMP COUNT
 MOV R0,R1 ;SAVE CONTENTS OF TSSR
 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED

TRAP C\$ERHRD
 .WORD 543
 .WORD WRTMSG
 .WORD SFIMSG

25\$: CKLOOP ;LOOP IF SELECTED
 TRAP C\$CLP1

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

JSR PC,REWIND ;CALL TAPE REWIND COMMAND
 BCS 30\$;BR, IF NO PROBLEM
 MOV R0,R1 ;SAVE TSSR
 INC FATFLG ;BUMP COUNT
 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED

TRAP C\$ERHRD
 .WORD 544
 .WORD T25RWN
 .WORD PKTSSR

30\$: CKLOOP ;LOOP IF SELECTED
 TRAP C\$CLP1

;*****
 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)


```

4890
4891
4892
4893 052176 013701 053730'          MOV      T25BFR*6,R1          ;PICK UP XSTO
4894 052202 010102                    MOV      R1,R2              ;SET UP EXPECTED
4895 052204 052702 000002          BIS      @BIT1,R2          ;SET BOT BIT IN EXPECTED
4896 052210 020102                    CMP      R1,R2              ;DOES EXP = REC'D
4897 052212 001406                    BEQ      40$                ;BR, IF EQUAL (OK)
4898 052214 005237 002212'          INC      FATFLG            ;BUMP COUNT
4902 052220                    ERRHRD  ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    545
                                .WORD    T25BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
                                .WORD    546
                                .WORD    T25WDE
052220 104456
052222 001041
052224 054225'
052226 015374'
4903 052230                    40$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    545
052230 104406
4904 052232 013701 054056'          MOV      T25CN2,R1          ;NUMBER OF RECORDS ON TAPE
4905 052236 012702 177776          MOV      @65534.,R2        ;MAX IT CAN SPACE OVER
4906 052242 020201                    CMP      R2,R1              ;WHICH VALUE CAN WE USE
4907 052244 003002                    BGT      46$                ;BR, IF @ WRITTEN > 64K
4908 052246 010103                    MOV      R1,R3              ;@ WRITTEN CAN BE USED
4909 052250 000401                    BR       47$                ;MOVE ON
4910 052252 010203                    46$:  MOV      R2,R3          ;USE MAX NUMBER
4911 052254                    47$:
4912 052254 010337 054032'          MOV      R3,T25RB          ;NUMBER OF RECORDS TO SPACE OVER
4913
4914
4915
4916
4917
4918
4919
4920 052260 012737 140010 054030'    MOV      @140010,T25PK3    ;SPACE FORWARD,ACK,CVC=1 COMMAND
4921 052266 012704 054030'          MOV      @T25PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4922 052272 010465 000000          MOV      R4,T5DB(R5)       ;ISSUE COMMAND
4923 052276 013737 054056' 054062'  MOV      T25CN2,T25DLY     ;SET UP DELAY COUNTER
4924 052304 004737 016150'          48$:  JSR      PC,WAITF      ;WAIT FOR SSR TO SET
4925 052310 016501 000002          MOV      T5SR(R5),R1       ;GET T5SR CONTENTS
4926 052314 012702 000200          MOV      @5SR,R2           ;SET UP EXPECTED
4927 052320 020102                    CMP      R1,R2              ;ARE THEY EQUAL
4928 052322 001425                    BEQ      50$                ;BR, IF OK
4929 052324                    DELAY  250                  ;WAIT .25 SECONDS
                                MOV      @250,(PC)-
                                .WORD    0
                                MOV      L$DLY,(PC)-
                                .WORD    0
                                DEC      -6(PC)
                                BNE     -.4
                                DEC      -22(PC)
                                BNE     -.20
                                MOV      @250,(PC)-
                                .WORD    0
                                MOV      L$DLY,(PC)-
                                .WORD    0
                                DEC      -6(PC)
                                BNE     -.4
                                DEC      -22(PC)
                                BNE     -.20
052324 012727 000250
052330 000000
052332 013727 002116'
052336 000000
052340 005367 177772
052344 001375
052346 005367 177756
052352 001367
4930 052354 005337 054062'          DEC      T25DLY            ;DEC THE DELAY COUNTER
4931 052360 001351                    BNE     48$                ;BR, IF COUNTER HASN'T EXPIRED
4932 052362 005237 002212'          INC      FATFLG            ;BUMP COUNT
4936 052366                    ERRHRD  ERRNO,T25WDE,EXPREC ;T5SR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD    546
                                .WORD    T25WDE
052366 104456
052370 001042
052372 054145'

```

```

4937 052374 015374'
052376 104406
4938 052400 013701 054056'
4939 052404 012702 177776
4940 052410 020201
4941 052412 003002
4942 052414 010103
4943 052416 000401
4944 052420 010203
4945 052422 162703 000001
4946 052426 010337 054032'
4947
4948
4949
4950
4951
4952
4953
4954 052432 012737 140410 054030'
4955 052440 012704 054030'
4956 052444 010465 000000
4957 052450 013737 054056' 054062'
4958 052456 004737 016150'
4959 052462 016501 000002
4960 052466 012702 000200
4961 052472 020102
4962 052474 001425
4963 052476
052476 012727 000250
052502 000000
052504 013727 002116'
052510 000000
052512 005367 177772
052516 001375
052520 005367 177756
052524 001367
4964 052526 005337 054062'
4965 052532 001351
4966 052534 005237 002212'
4970 052540
052540 104456
052542 001043
052544 054145'
052546 015374'
4971 052550
052550 104406
4972 052552 012703 010000
4973 052556 013737 003114' 054032'
4974
4975
4976
4977
4978
4979
4980
4981 052564 012737 100001 054030'

```

```

50$: CKLOOP
MOV T25CN2,R1 ;NUMBER OF RECORDS ON TAPE
MOV #65534.,R2 ;MAX IT CAN SPACE OVER
CMP R2,R1 ;WHICH VALUE CAN WE USE
BGT 55$ ;BR, IF # WRITTEN > 64K
MOV R1,R3 ;# WRITTEN CAN BE USED
BR 60$ ;MOVE ON
55$: MOV R2,R3 ;USE MAX NUMBER
60$: SUB #1,R3 ;DON'T GO ALL THE WAY YET
MOV R3,T25RB ;NUMBER OF RECORDS TO SPACE OVER

;*****
;SPACE REVERSE,ACK,CVC=1 COMMAND
;*****
MOV #140410,T25PK3 ;SPACE REVERSE,ACK,CVC=1 COMMAND
MOV #T25PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
MOV R4,T25DB(R5) ;ISSUE COMMAND
MOV T25CN2,T25DLY ;SET UP COUNTER
70$: JSR PC,WAITF ;WAIT FOR SSR TO SET
MOV T25SR(R5),R1 ;GET T25SR CONTENTS
MOV #SSR,R2 ;SET UP EXPECTED
CMP R1,R2 ;ARE THEY EQUAL
BEQ 75$ ;BR, IF OK
DELAY 250 ;WAIT ABOUT .25 SECONDS
MOV #250,(PC)+
WORD 0
MOV L$DLY,(PC)+
WORD 0
DEC -6(PC)
BNE -.4
DEC -22(PC)
BNE -.20
DEC T25DLY ;BUMP COUNTER DOWN
BNE 70$ ;BR, IF COUNTER HASN'T EXPIRED
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T25WDE,EXPREC ;T25SR INCORRECT AFTER READ DATA
TRAP C$ERRRD
WORD 547
WORD T25WDE
WORD EXPREC
75$: CKLOOP ;LOOP IF SELECTED
TRAP C$CLP1
MOV #4096.,R3 ;RECORD SIZE
MOV FREE,T25RB ;STARTING READ BUFFER ADDRESS

;*****
;READ DATA,ACK COMMAND
;*****
MOV #100001,T25PK3 ;READ DATA,ACK COMMAND

```



```

5031
5032 ;*****
5033 ;
5034 ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
5035 ;
5036 ;*****
5037
5038 052750 004737 015674' JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
5039 052754 103407 BCS 20$ ;BR IF INIT WAS OK
5040 052756 005237 002212' INC FATFLG ;BUMP COUNT
5044 052762 010001 MOV RO,R1 ;CONTENTS OF TSSR REGISTER
5045 052764 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
5046 052764 104455 TRAP C$ERDF
5047 052766 001046 .WORD 550
5048 052770 003642' .WORD SFIERR
5049 052772 011734' .WORD SFIMSG
5046 052774 013737 002172' 053720' 20$: MOV UNITN,T25DSW ;SET UP UNIT NUMBER
5047
5048 053002 012704 053700' MOV #T25PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
5049
5050 ;*****
5051 ;
5052 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
5053 ;
5054 ;*****
5055
5056 053006 004737 010562' JSR PC,WRTPHR ;ISSUE WRITE CHARACTERISTICS
5057 053012 103407 BCS 25$ ;BR, IF COMMAND ISSUED OK
5058 053014 005237 002212' INC FATFLG ;BUMP COUNT
5062 053020 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
5063 053022 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
5064 053022 104456 TRAP C$ERHRD
5065 053024 001047 .WORD 551
5066 053026 005046' .WORD WRTMSG
5067 053030 011734' .WORD SFIMSG
5064 053032 25$: CKLOOP ;LOOP IF SELECTED
5065 053032 104406 TRAP C$CLP1
5066
5067 ;*****
5068 ;
5069 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5070 ;
5071 ;*****
5072
5072 053034 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
5073 053040 103407 BCS 30$ ;BR, IF NO PROBLEM
5074 053042 010001 MOV RO,R1 ;SAVE TSSR
5075 053044 005237 002212' INC FATFLG ;BUMP COUNT
5079 053050 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
5080 053050 104456 TRAP C$ERHRD
5081 053052 001050 .WORD 552
5082 053054 055035' .WORD T25RWN
5083 053056 011746' .WORD PKTSSR
5080 053060 30$: CKLOOP ;LOOP IF SELECTED
5081 053060 104406 TRAP C$CLP1
5082 ;*****

```

```

5083
5084 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5085 ;
5086 ;*****
5087
5088 053062 013701 053730'      MOV      T25BFR+6,R1      ;PICK UP XSTO
5089 053066 010102              MOV      R1,R2           ;SET UP EXPECTED
5090 053070 052702 000002      BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
5091 053074 020102              CMP      R1,R2           ;DOES EXP = REC'D
5092 053076 001406              BEQ      40$             ;BR, IF EQUAL (OK)
5093 053100 005237 002212'      INC      FATFLG          ;BUMP COUNT
5097 053104              ERRHRD  ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD     553
                                .WORD     T25BOT
                                .WORD     EXPREC
5098 053114              40$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
5099 053116 012737 000001 054032'  MOV      @1,T25RB        ;NUMBER OF RECORDS TO SPACE OVER
5100
5101 ;*****
5102 ;
5103 ;SPACE REVERSE,ACK COMMAND
5104 ;
5105 ;*****
5106
5107 053124 012737 100410 054030'  MOV      @100410,T25PK3  ;SPACE REVERSE,ACK COMMAND
5108 053132 012704 054030'      MOV      @T25PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
5109 053136              65$:  MOV      R4,T5DB(R5)      ;ISSUE COMMAND
5110 053136 010465 000000      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
5111 053142 004737 016150'      MOV      T5SR(R5),R1     ;GET T5SR CONTENTS
5112 053146 016501 000002      MOV      @5SR!5C!BIT1!BIT2,R2 ;SET UP EXPECTED
5113 053152 012702 100206      CMP      R1,R2           ;ARE THEY EQUAL
5114 053156 020102              BEQ      75$             ;BR, IF OK
5115 053160 001406              INC      FATFLG          ;BUMP COUNT
5116 053162 005237 002212'      ERRHRD  ERRNO,T25WDE,PKT5SR ;T5SR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD     554
                                .WORD     T25WDE
                                .WORD     PKT5SR
5121 053176              75$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
5122
5123 ;*****
5124 ;
5125 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5126 ;
5127 ;*****
5128
5129 053200 013701 053730'      MOV      T25BFR+6,R1     ;GET XSTO STATUS WORD
5130 053204 010102              MOV      R1,R2           ;SET UP EXPECTED
5131 053206 052702 002000      BIS      @BIT10,R2       ;SET THE NEF BIT
5132 053212 020102              CMP      R1,R2           ;ARE THEY EQUAL
5133 053214 001406              BEQ      170$            ;BR, IF EQUAL (GOOD)
5134 053216 005237 002212'      INC      FATFLG          ;BUMP COUNT
5138 053222              ERRHRD  ERRNO,T25NEF,EXPREC ;NEF SHOULD BE SET

```



```

5186 ;*****
5187 ;
5188 053326 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
5189 053332 103407 BCS 25$ ;BR, IF COMMAND ISSUED OK
5190 053334 005237 002212' INC FATFLG ;BUMP COUNT
5194 053340 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
5195 053342 ERRHRD ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICSC FAILED
; TRAP C$ERHRD
; .WORD 557
; .WORD WRTMSG
; .WORD SFMSG
5196 053352 25$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
5197 ;*****
5198 ;
5199 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5200 ;
5201 ;
5202 ;*****
5203 ;
5204 053354 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
5205 053360 103407 BCS 30$ ;BR, IF NO PROBLEM
5206 053362 010001 MOV R0,R1 ;SAVE TSSR
5207 053364 005237 002212' INC FATFLG ;BUMP COUNT
5211 053370 ERRHRD ERRNO,T25RWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERHRD
; .WORD 558
; .WORD T25RWN
; .WORD PKTSSR
5212 053400 30$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
5213 ;*****
5214 ;
5215 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5216 ;
5217 ;
5218 ;*****
5219 ;
5220 053402 013701 053730' MOV T25BFR+6,R1 ;PICK UP XSTO
5221 053406 010102 MOV R1,R2 ;SET UP EXPECTED
5222 053410 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
5223 053414 020102 CMP R1,R2 ;DOES EXP = REC'D
5224 053416 001406 BEQ 40$ ;BR, IF EQUAL (OK)
5225 053420 005237 002212' INC FATFLG ;BUMP COUNT
5229 053424 ERRHRD ERRNO,T25BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
; TRAP C$ERHRD
; .WORD 559
; .WORD T25BOT
; .WORD EXPREC
5230 053434 40$: CKLOOP ;NUMBER OF RECORDS TO SPACE OVER
; TRAP C$CLP1
5231 053436 012737 000001 054032' MOV #1,T25RB
5232 ;*****
5233 ;
5234 ;SPACE FORWARD,IE,ACK,CVC=1 COMMAND
5235 ;
5236 ;

```



```

5347 054044 000000      T25S3: .WORD 0          ;DATA AREA
5348                    ;
5349                    ;
5350                    ;
5351                    ;
5352                    ;
5353 054046 100005      T25RN: .WORD 100005     ;READ DATA (NEXT)
5354 054050 100405      T25WDR: .WORD 100405    ;READ DATA RETRY
5355 054052 102005      T25CON: .WORD 102005    ;WRITE CONTINOUS
5356 054054 177777      .WORD 177777           ;END OF DATA
5357                    ;
5358 054056 000000      T25CN2: .WORD 0         ;COUNTER FOR RECORDS
5359 054060 000000      T25CNT: .WORD 0         ;COUNTER FOR RECORDS
5360 054062 000000      T25DLY: .WORD 0         ;COUNTER FOR RECORDS
5361                    ;
5362                    ;
5363                    ;*
5364                    ;LOCAL TEXT MESSAGES FOR TEST
5365                    ;-
5366 054064      127      122      111  T25SSR: .ASCIZ 'WRITE SUBSYSTEM Miscellaneous Read Status Failed'
5367 054145      124      123      123  T25WDE: .ASCIZ 'TSSR Not Correct After POSITION (SPACE) Command'
5368 054225      124      141      160  T25BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
5369 054272      124      123      123  T25TM: .ASCIZ 'TSSR Not Correct After POSITION (Space Command) Reject'
5370 054361      127      162      151  T25NET: .ASCIZ 'Write Tape, Status Alert, But No EOT Sensed'
5371 054435      123      160      141  T25WNG: .ASCIZ 'Space Forward Failed To Position On Correct Record'
5372 054520      123      160      141  T25BNC: .ASCIZ 'Space Forward, From BOT, Failed To Clear BOT Indication'
5373 054610      123      160      141  T25WNH: .ASCIZ 'Space Reverse Failed To Position On Correct Record'
5374 054673      123      160      141  T25NEF: .ASCIZ 'Space Reverse, At BOT, Failed To Set NEF (XST0)'
5375 054753      123      160      141  T25RIB: .ASCIZ 'Space Reverse, Into BOT, Failed To Set RIB (XST3)'
5376 055035      122      145      167  T25RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
5377 055104      104      162      151  T25OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
5378 055157      124      123      123  T25WDC: .ASCIZ 'TSSR Not Correct After READ DATA Command'
5379 055230      123      160      141  TST25ID: .ASCIZ 'Space Records'
5380                    ;
5381                    ;
5382                    ;
5383                    ;*
5384                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5385                    ;WRITE SUBSYSTEM MEMORY COMMAND
5386                    ;
5387                    ;-
5388 055246      T25REST:
5389 055246      SAVREG
5390 055252      012701 053700'      MOV #T25PACKET,R1          ;SAVE THE REGISTERS
5391 055256      012721 100004      MOV #100004,(R1)+         ;START OF THE PACKET
5392 055262      012721 053710'      MOV #T25DATA,(R1)+       ;WRITE SUBSYSTEM MEM. WITH ACK
5393 055266      005021              CLR (R1)+                 ;ADDRESS OF CHARAISTIC'S DATA BLOCK
5394 055270      012721 000012      MOV #10.,(R1)+           ;EXTENDED ADDRESS
5395 055274      012721 053722'      MOV #T25BFR,(R1)+        ;SIZE OF DATA BLOCK IN BYTES
5396 055300      005021              CLR (R1)+                 ;ADDRESS OF MESSAGE BUFFER
5397 055302      012721 000024      MOV #20.,(R1)+           ;LENGTH OF MESSAGE BUFFER
5398 055306      005021              CLR (R1)+
5399 055310      012711 000000      MOV #0,(R1)              ;SELECT DRIVE ZERO
5400 055314      012702 000030      MOV #24.,R2              ;NUMBER OF LOCATIONS TO BE CLEARED
5401 055320      012762 177777 053722' 64$: MOV #177777,T25BFR(R2)    ;ALL ONES TO MESSAGE BUFFER
5402 055326      005742              TST -(R2)                 ;NEXT LOCATION
5403 055330      022702 000000      CMP #0,R2                ;IS R2 AT ZERO YET

```

```

5404 055334 001371          BNE      64$          ;KEEP GOING UNTIL DONE
5405 055336 000207          RTS       PC           ;RETURN
5406
5407 055340                T25RT2:
5408 055340                SAVREG
5409 055344 012701 054010'  MOV      #T25PK2,R1    ;SAVE THE REGISTERS
5410 055350 012721 100006  MOV      #100006,(R1)+ ;START OF THE PACKET
5411 055354 012721 054040'  MOV      #T25BF2,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
5412 055360 005021          CLR      (R1)+         ;ADDRESS OF DATA BLOCK
5413 055362 012721 000006  MOV      #6.,(R1)+     ;EXTENDED ADDRESS
5414 055366 005021          CLR      (R1)+         ;SIZE OF DATA BLOCK IN BYTES
5415 055370 012701 054040'  MOV      #T25BF2,R1    ;POINT TO DATA SEL AREA
5416 055374 005021          CLR      (R1)+
5417 055376 005011          CLR      (R1)
5418 055400 000207          RTS       PC           ;RETURN
5419 055402                T25RT3:
5420 055402                SAVREG
5421 055406 012701 054030'  MOV      #T25PK3,R1    ;SAVE THE REGISTERS
5422 055412 012721 000000  MOV      #0,(R1)+      ;START OF THE PACKET
5423 055416 012721 000000  MOV      #0,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,
5424 055422 005021          CLR      (R1)+         ;ADDRESS OF DATA BLOCK
5425 055424 012721 000000  MOV      #0,(R1)+     ;EXTENDED ADDRESS
5426 055430 000207          RTS       PC           ;SIZE OF DATA BLOCK IN BYTES
5427 055432                ENDTST          ;RETURN
5428 055432                L10071:
5429 055432 104401          TRAP      C$ETST

```

.SBTTL TEST 6: REREADS

```

5430 ;+
5431 ;
5432 ; THIS TEST VERIFIES THAT THE REREAD PREVIOUS AND REREAD NEXT
5433 ; COMMANDS OPERATE PROPERLY. VARIOUS COMBINATIONS OF ODD AND EVEN
5434 ; DATA BUFFER BOUNDRIES, RECORD SIZES (UP TO 64K BYTES IF MEMORY
5435 ; SPACE IS AVAILIABLE), AND BYTE-SWAP (SWP) AND OPPOSITE (OPP)
5436 ; CONRTOL ARE USED. ALSO TESTED ARE PROPER TERMINATIONS ON
5437 ; EXCEPTIONAL OR ERROR CONDITIONS: RECORD LENGTH LONG, RECORD
5438 ; LENGTH SHORT, READ REVERSE AT BOT, ILLEGAL DATA BUFFER ADDRESSES,
5439 ; AND DATA BUFFERS IN NONEXISTENT MEMORY.
5440 ;
5441 ;
5442 ; THE TEST CONSISTS OF THE FOLLOWING 15 SUBTESTS
5443 ;
5444 ;
5445 ;
5446 ;-
5447 055434                BGNTST
5448 055434                T6::
5448 055434 012737 006256' 002170' MOV      #EPRT2,EPRTSW ;SECONDARY ERROR MESSAGE
5449 055442 004737 017176' JSR      PC,KTOFF      ;DON'T NEED KT11
5454 055446 012700 074607' MOV      #TST26ID,RO   ;ASCII MESSAGE TO IDENTIFY TEST
5455 055452 004737 016412' JSR      PC,TSTSETUP  ;DO INITIAL TEST SETUP
5456 055456 012737 000005 002206' MOV      #5,LOOPCNT   ;PERFORM 5 ITERATIONS
5457 055464 004737 021116' JSR      PC,MEMCK     ;CHECK FOR MEMORY
5458 055470 005037 072056' CLR      T26CNT       ;CLEAR TAPE RECORD COUNTER
5459 ;+
5460 ;
5461 ; TEST 6, SUBTEST 1

```



```

5508
5509 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
5510 ;
5511 ;*****
5512
5513 055614 004737 010562'      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
5514 055620 103407              BCS      26$           ;BR, IF COMMAND ISSUED OK
5515 055622 005237 002212'      INC      FATFLG        ;BUMP COUNT
5519 055626 010001              MOV      R0,R1         ;SAVE CONTENTS OF TSSR
5520 055630              ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
5521 055630 104456              TRAP                                C$ERHRD
5522 055632 001132              .WORD                                602
5523 055634 005046'            .WORD                                WRTMSG
5524 055636 011734'            .WORD                                SFIMSG
5525 26$:      CKLOOP              ;LOOP IF SELECTED
5526 TRAP                                C$CLP1
5527 ;*****
5528 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5529 ;
5530 ;*****
5531 055642 004737 010714'      JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
5532 055646 103413              BCS      30$           ;BR, IF NO PROBLEM
5533 055650 016501 000002      MOV      TSSR(R5),R1  ;GET TSSR
5534 055654 012702 000200      MOV      #SSR,R2     ;SET UP EXPECTED TSSR
5535 055660 010004              MOV      R0,R4        ;PACKET ADDRESS SET UP
5536 055662 005237 002212'      INC      FATFLG        ;BUMP COUNT
5537 055666              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
5538 055666 104456              TRAP                                C$ERHRD
5539 055670 001133              .WORD                                603
5540 055672 073364'            .WORD                                T26RWN
5541 055674 011746'            .WORD                                PKTSSR
5542 30$:      CKLOOP              ;LOOP IF SELECTED
5543 TRAP                                C$CLP1
5544 ;*****
5545 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
5546 ;
5547 ;*****
5548 055700 013701 071730'      MOV      T26BFR+6,R1 ;PICK UP XST0
5549 055704 010102              MOV      R1,R2        ;SET UP EXPECTED
5550 055706 052702 000002      BIS      #BIT1,R2    ;SET BOT BIT IN EXPECTED
5551 055712 020102              CMP      R1,R2        ;DOES EXP = REC'D
5552 055714 001406              BEQ      40$          ;BR, IF EQUAL (OK)
5553 055716 005237 002212'      INC      FATFLG        ;BUMP COUNT
5554 055722              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
5555 055722 104456              TRAP                                C$ERHRD
5556 055724 001134              .WORD                                604
5557 055726 073075'            .WORD                                T26BOT
5558 055730 015374'            .WORD                                EXPREC
5559 40$:      CKLOOP              ;LOOP IF SELECTED
5560 TRAP                                C$CLP1
5561 055732 104406              MOV      #256.,R3     ;RECORD SIZE
5562 055734 012703 000400

```

```

5559 055740 013737 003114' 072032'      MOV      FREE,T26RB      ;STARTING WRITE BUFFER ADDRESS
5560
5561      ;*****
5562      ;
5563      ;WRITE DATA,ACK,CVC=1 COMMAND
5564      ;
5565      ;*****
5566
5567 055746 012737 140005 072030'      MOV      #140005,T26PK3  ;WRITE DATA,ACK,CVC=1 COMMAND
5568 055754 012704 072030'      MOV      #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
5569 055760
5570 055760 010300      65$:    MOV      R3,R0          ;SET PATTERN IN CORRECT REGISTER
5571 055762 004737 017324'      JSR      PC,FILLMEM     ;FILL MEMORY WITH RECORD SIZE
5572 055766 010337 072036'      MOV      R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
5573 055772 010465 000000      MOV      R4,TSSB(R5)    ;ISSUE COMMAND
5574 055776 004737 016150'      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
5575 056002 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
5576 056006 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED
5577 056012 020102      CMP      R1,R2         ;ARE THEY EQUAL
5578 056014 001406      BEQ      75$           ;BR, IF OK
5579 056016 005237 002212'      INC      FATFLG        ;BUMP COUNT
5583 056022      ERRHRD  ERRNO,WRTErr,EXPREC ;TSSR INCORRECT AFTER WRITE DATA
      056022 104456      TRAP    C$ERRHRD
      056024 001135      .WORD  605
      056026 005103'    .WORD  WRTErr
      056030 015374'    .WORD  EXPREC
5584 056032      75$:    CKLOOP          ;LOOP IF SELECTED
      056032 104406      TRAP    C$CLP1
5585 056034 005723      TST     (R3),         ;BUMP RECORD SIZE
5586 056036 022703 000414      CMP     #268.,R3     ;END OF RECORD YET
5587 056042 001346      BNE    65$           ;BR, IF MORE RECORDS TO WRITE
5588 056044      80$:    CKLOOP          ;LOOP IF SELECTED
      056044 104406      TRAP    C$CLP1
5589 056046
5590
5591      ;*****
5592      ;
5593      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5594      ;
5595      ;*****
5596
5597 056046 004737 010714'      JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
5598 056052 103413      BCS     130$         ;BR, IF NO PROBLEM
5599 056054 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR
5600 056060 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED TSSR
5601 056064 010004      MOV      R0,R4         ;PACKET ADDRESS SET UP
5602 056066 005237 002212'      INC      FATFLG        ;BUMP COUNT
5606 056072      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      056072 104456      TRAP    C$ERRHRD
      056074 001136      .WORD  606
      056076 073364'    .WORD  T26RWN
      056100 011746'    .WORD  PKTSSR
5607 056102      130$:  CKLOOP          ;LOOP IF SELECTED
      056102 104406      TRAP    C$CLP1
5608
5609      ;*****
5610      ;

```

```

5611 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5612 ;
5613 ;*****
5614
5615 056104 013701 071730'      MOV      T26BFR+6,R1      ;PICK UP XSTO
5616 056110 010102      MOV      R1,R2          ;SET UP EXPECTED
5617 056112 052702 000002      BIS      @BIT1,R2       ;SET BOT BIT IN EXPECTED
5618 056116 020102      CMP      R1,R2         ;DOES EXP = REC'D
5619 056120 001406      BEQ     140$           ;BR, IF EQUAL (OK)
5620 056122 005237 002212'      INC      FATFLG        ;BUMP COUNT
5624 056126      ERRHRD  ERRNO,T26BOT,PKTSSR ;TAPE NOT AT BOT AFTER REWIND
      056126 104456      TRAP      C$ERHRD
      056130 001137      .WORD    607
      056132 073075'    .WORD    T26BOT
      056134 011746'    .WORD    PKTSSR
5625 056136      140$:  CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
      056136 104406
5626 056140 012737 000400 072062'  MOV      @256.,T26RSZ  ;SET RECORD SIZE
5627
5628 ;*****
5629 ;
5630 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
5631 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
5632 ;
5633 ;*****
5634
5635 056146 012703 000001      145$:  MOV      @1,R3      ;SPACE ONE RECORD PARAMETER
5636 056152 004737 010366'    JSR     PC,SPACE      ;CALL SPACE ROUTINE
5637 056156 103412      BCS     150$         ;BR, IF NO PROBLEM WITH SPACE COMMAND
5638 056160 016501 000002      MOV     TSSR(R5),R1   ;GET TSSR
5639 056164 012702 000200      MOV     @SSR,R2      ;SET UP EXPECTED TSSR
5640 056170 005237 002212'    INC     FATFLG        ;BUMP COUNT
5644 056174      ERRHRD  ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
      056174 104456      TRAP      C$ERHRD
      056176 001140      .WORD    608
      056200 072477'    .WORD    T26SC
      056202 015374'    .WORD    EXPREC
5645 056204      150$:  CKLOOP      ;RECORD SIZE      TRAP      C$CLP1
      056204 104406
5646 056206 013703 072062'    MOV     T26RSZ,R3    ;RECORD SIZE
5647 056212 013737 003114' 072032'  MOV     FREE,T26RB   ;STARTING READ BUFFER ADDRESS
5648
5649 ;*****
5650 ;
5651 ;REREREAD DATA,CVC=1,ACK COMMAND
5652 ;
5653 ;*****
5654
5655 056220 012737 141001 072030'  MOV     @141001,T26PK3 ;REREREAD DATA,CVC=1,ACK COMMAND
5656 056226 012704 072030'  165$:  MOV     @T26PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
5657 056232 010337 072036'    MOV     R3,T26SZ     ;SET UP RECORD SIZE IN PACKET
5658 056236 010465 000000      MOV     R4,TSDB(R5)  ;ISSUE COMMAND
5659 056242 004737 016150'    JSR     PC,WAITF     ;WAIT FOR SSR TO SET
5660 056246 016501 000002      MOV     TSSR(R5),R1  ;GET TSSR CONTENTS
5661 056252 012702 000200      MOV     @SSR,R2      ;SET UP EXPECTED
5662 056256 020102      CMP     R1,R2        ;ARE THEY EQUAL
5663 056260 001406      BEQ     170$         ;BR, IF OK

```



```

056406 104402
5713 056410 004737 074620'      JSR      PC,T26REST      ;SET COMMAND PACKET      TRAP      C$BSUB
5714 056414 004737 074712'      JSR      PC,T26RT2      ;SET UP OTHER COMMAND PACKET
5715 056420 004737 074754'      JSR      PC,T26RT3      ;SET UP OTHER COMMAND PACKET
5716
5717      ;*****
5718      ;
5719      ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
5720      ;
5721      ;*****
5722
5723 056424 004737 015674'      JSR      PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
5724 056430 103407      BCS      20$      ;BR IF INIT WAS OK
5725 056432 005237 002212'      INC      FATFLG      ;BUMP COUNT
5729 056436 010001      MOV      RO,R1      ;CONTENTS OF TSSR REGISTER
5730 056440      ERRDF      ERRNO,SFIERR,SFIMSG      ;FATAL ERROR TSSR WAS NOT OK
      056440 104455      TRAP      C$ERDF
      056442 001143      .WORD      611
      056444 003642'      .WORD      SFIERR
      056446 011734'      .WORD      SFIMSG
5731 056450 013737 002172' 071720' 20$:      MOV      UNITN,T26DSW      ;SET UP UNIT NUMBER
5732
5733 056456 012704 071700'      MOV      @T26PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
5734
5735      ;*****
5736      ;
5737      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
5738      ;
5739      ;*****
5740
5741 056462 004737 010562'      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
5742 056466 103407      BCS      26$      ;BR, IF COMMAND ISSUED OK
5743 056470 005237 002212'      INC      FATFLG      ;BUMP COUNT
5747 056474 010001      MOV      RO,R1      ;SAVE CONTENTS OF TSSR
5748 056476      ERRHRD      ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTICSC FAILED
      056476 104456      TRAP      C$ERHRD
      056500 001144      .WORD      612
      056502 005046'      .WORD      WRTMSG
      056504 011734'      .WORD      SFIMSG
5749 056506      26$:      CKLOOP      ;LOOP IF SELECTED      TRAP      C$CLP1
      056506 104406
5750
5751      ;*****
5752      ;
5753      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5754      ;
5755      ;*****
5756
5757 056510 004737 010714'      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
5758 056514 103413      BCS      30$      ;BR, IF NO PROBLEM
5759 056516 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR
5760 056522 012702 000200      MOV      @SSR,R2      ;SET UP EXPECTED TSSR
5761 056526 010004      MOV      RO,R4      ;PACKET ADDRESS SET UP
5762 056530 005237 002212'      INC      FATFLG      ;BUMP COUNT
5766 056534      ERRHRD      ERRNO,T26RWN,PKTSSR      ;REWIND NOT ACCEPTED
      056534 104456      TRAP      C$ERHRD
      056536 001145      .WORD      613

```

```

056540 073364' .WORD T26RWN
056542 011746' .WORD PKTSSR
5767 056544 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
056544 104406
5768
5769 ;*****
5770 ;
5771 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5772 ;
5773 ;*****
5774
5775 056546 013701 071730' MOV T26BFR+6,R1 ;PICK UP XSTO
5776 056552 010102 MOV R1,R2 ;SET UP EXPECTED
5777 056554 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
5778 056560 020102 CMP R1,R2 ;DOES EXP = REC'D
5779 056562 001406 BEQ 40$ ;BR, IF EQUAL (OK)
5780 056564 005237 002212' INC FATFLG ;BUMP COUNT
5784 056570 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
056570 104456 TRAP C$ERHRD
056572 001146 .WORD 614
056574 073075' .WORD T26BOT
056576 015374' .WORD EXPREC
5785 056600 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
056600 104406
5786 056602 012703 000400 MOV #256.,R3 ;RECORD SIZE
5787 056606 013737 003114' 072032' MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
5788
5789 ;*****
5790 ;
5791 ;WRITE DATA,ACK,SWB COMMAND
5792 ;
5793 ;*****
5794
5795 056614 012737 110005 072030' MOV #110005,T26PK3 ;WRITE DATA,ACK,SWB COMMAND
5796 056622 012704 072030' MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
5797 056626
5798 056626 010300 65$: MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
5799 056630 004737 017324' JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
5800 056634 010337 072036' MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
5801 056640 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
5802 056644 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
5803 056650 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
5804 056654 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
5805 056660 020102 CMP R1,R2 ;ARE THEY EQUAL
5806 056662 001406 BEQ 75$ ;BR, IF OK
5807 056664 005237 002212' INC FATFLG ;BUMP COUNT
5811 056670 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
056670 104456 TRAP C$ERHRD
056672 001147 .WORD 615
056674 005103' .WORD WRERR
056676 011746' .WORD PKTSSR
5812 056700 75$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
056700 104406
5813 056702 005723 TST (R3)+ ;BUMP RECORD SIZE
5814 056704 022703 000414 CMP #268.,R3 ;END OF RECORD YET
5815 056710 001346 BNE 65$ ;BR, IF MORE RECORDS TO WRITE
5816 056712 80$: CKLOOP ;LOOP IF SELECTED

```

```

056712 104406                                TRAP    C$CLP1
5817 056714                                120$:
5818
5819 ;*****
5820 ;
5821 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5822 ;
5823 ;*****
5824
5825 056714 004737 010714'                    JSR     PC,REWIND                    ;CALL TAPE REWIND COMMAND
5826 056720 103413                            BCS     130$                          ;BR, IF NO PROBLEM
5827 056722 016501 000002                    MOV     TSSR(R5),R1                  ;GET TSSR
5828 056726 012702 000200                    MOV     #SSR,R2                       ;SET UP EXPECTED TSSR
5829 056732 010004                            MOV     R0,R4                          ;PACKET ADDRESS SET UP
5830 056734 005237 002212'                    INC     FATFLG                         ;BUMP COUNT
5834 056740                                ERRHRD  ERRNO,T26RWN,PKTSSR          ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD   616
                                .WORD   T26RWN
                                .WORD   PKTSSR
056740 104456
056742 001150
056744 073364'
056746 011746'
5835 056750                                130$:  CKLOOP                        ;LOOP IF SELECTED
056750 104406                                TRAP    C$CLP1
5836
5837 ;*****
5838 ;
5839 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5840 ;
5841 ;*****
5842
5843 056752 013701 071730'                    MOV     T26BFR+6,R1                  ;PICK UP XSTO
5844 056756 010102                            MOV     R1,R2                          ;SET UP EXPECTED
5845 056760 052702 000002                    BIS     #BIT1,R2                       ;SET BOT BIT IN EXPECTED
5846 056764 020102                            CMP     R1,R2                          ;DOES EXP = REC'D
5847 056766 001406                            BEQ     140$                          ;BR, IF EQUAL (OK)
5848 056770 005237 002212'                    INC     FATFLG                         ;BUMP COUNT
5852 056774                                ERRHRD  ERRNO,T26BOT,EXPREC          ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD   617
                                .WORD   T26BOT
                                .WORD   EXPREC
056774 104456
056776 001151
057000 073075'
057002 015374'
5853 057004                                140$:  CKLOOP                        ;LOOP IF SELECTED
057004 104406                                TRAP    C$CLP1
5854 057006 012737 000400 072062'          MOV     #256.,T26RSZ                  ;SET UP RECORD SIZE
5855
5856 ;*****
5857 ;
5858 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
5859 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
5860 ;
5861 ;*****
5862
5863 057014 012703 000001                    145$:  MOV     #1,R3                    ;SPACE ONE RECORD PARAMETER
5864 057020 004737 010366'                    JSR     PC,SPACE                      ;CALL SPACE ROUTINE
5865 057024 103412                            BCS     150$                          ;BR, IF NO PROBLEM WITH SPACE COMMAND
5866 057026 016501 000002                    MOV     TSSR(R5),R1                  ;GET TSSR
5867 057032 012702 000200                    MOV     #SSR,R2                       ;SET UP EXPECTED TSSR
5868 057036 005237 002212'                    INC     FATFLG                         ;BUMP COUNT

```

```

5872 057042          ERRHRD  ERRNO,T26SC,EXPREC          ;POSITION (SPACE RECORDS) FAILED
      057042 104456          TRAP          C$ERHRD
      057044 001152          .WORD          618
      057046 072477'        .WORD          T26SC
      057050 015374'        .WORD          EXPREC
5873          150$:  CKLOOP          TRAP          C$CLP1
      057052 104406
5874 057054 013703 072062'        MOV      T26RSZ,R3          ;RECORD SIZE
5875 057060 013737 003114' 072032'  MOV      FREE,T26RB        ;STARTING READ BUFFER ADDRESS
5876
5877          ;*****
5878          ;
5879          ;REREAD DATA,CVC=1,ACK,SWB COMMAND
5880          ;
5881          ;*****
5882
5883 057066 012737 151001 072030'  MOV      #151001,T26PK3    ;REREAD DATA,CVC=1,ACK,SWB COMMAND
5884 057074 012704 072030' 165$:  MOV      #T26PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
5885 057100 010337 072036'  MOV      R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
5886 057104 010465 000000  MOV      R4,TSDB(R5)     ;ISSUE COMMAND
5887 057110 004737 016150'  JSR      PC,WAITF        ;WAIT FOR SSR TO SET
5888 057114 016501 000002  MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
5889 057120 012702 000200  MOV      #SSR,R2        ;SET UP EXPECTED
5890 057124 020102  CMP      R1,R2          ;ARE THEY EQUAL
5891 057126 001406  BEQ      170$          ;BR, IF OK
5892 057130 005237 002212'  INC      FATFLG         ;BUMP COUNT
5896 057134          ERRHRD  ERRNO,T26WDC,PKTSSR    ;TSSR INCORRECT AFTER REREAD DATA
      057134 104456          TRAP          C$ERHRD
      057136 001153          .WORD          619
      057140 073720'        .WORD          T26WDC
      057142 011746'        .WORD          PKTSSR
5897 057144          170$:  CKLOOP          ;LOOP IF SELECTED          TRAP          C$CLP1
      057144 104406
5898 057146 013702 003114'  MOV      FREE,R2        ;CURRENT BUFFER ADDRESS TO R2
5899 057152 010304  MOV      R3,R4          ;CURRENT RECORD SIZE
5900 057154 162704 000400  SUB      #256.,R4       ;FIRST LOCATION IN BUFFER
5901 057160 060204 173$:  ADD      R2,R4          ;SET UP POINTER
5902 057162 021403  CMP      (R4),R3        ;CHECK DATA READ (R3=DATA ALSO)
5903 057164 001410  BEQ      180$          ;BR, IF ALL IS WELL
5904 057166 011401  MOV      (R4),R1       ;RECD DATA
5905 057170 010302  MOV      R3,R2          ;EXPECTED DATA
5906 057172 005237 002212'  INC      FATFLG         ;BUMP COUNT
5910 057176          ERRHRD  ERRNO,T26DTA,EXPREC    ;DATA READ NOT = WRITTEN
      057176 104456          TRAP          C$ERHRD
      057200 001154          .WORD          620
      057202 073142'        .WORD          T26DTA
      057204 015374'        .WORD          EXPREC
5911 057206          180$:  CKLOOP          ;LOOP IF SELECTED          TRAP          C$CLP1
      057206 104406
5912 057210 005724  TST      (R4)+          ;BUMP TO NEXT LOCATION
5913 057212 160204  SUB      R2,R4          ;CORRECT RECORDS SIZE VALUE
5914 057214 020403  CMP      R4,R3          ;END OF RECORD YET
5915 057216 001360  BNE     173$          ;BR, IF NOT AT END OF RECORD
5916 057220 005723  TST      (R3)+          ;BUMP RECORD SIZE
5917 057222 010337 072062'  MOV      R3,T26RSZ     ;STORE RECORD SIZE
5918 057226 022703 000412  CMP      #266.,R3      ;END OF RECORD YET
5919 057232 001270  BNE     145$          ;BR, IF MORE RECORDS TO READ

```



```

5972 057276 004737 015674'      JSR    PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
5973 057302 103407              BCS    20$             ;BR IF INIT WAS OK
5974 057304 005237 002212'      INC    FATFLG         ;BUMP COUNT
5978 057310 010001              MOV    RO,R1          ;CONTENTS OF TSSR REGISTER
5979 057312      ERRDF    ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      057312 104455              TRAP   C$ERDF
      057314 001155              .WORD 621
      057316 003642'            .WORD SFIERR
      057320 011734'            .WORD SFIMSG
5980 057322 013737 002172' 071720' 20$:  MOV    UNITN,T26DSW    ;SET UP UNIT NUMBER
5981
5982 057330 012704 071700'      MOV    #T26PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
5983
5984      ;*****
5985      ;
5986      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
5987      ;
5988      ;*****
5989
5990 057334 004737 010562'      JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
5991 057340 103407              BCS    26$             ;BR, IF COMMAND ISSUED OK
5992 057342 005237 002212'      INC    FATFLG         ;BUMP COUNT
5996 057346 010001              MOV    RO,R1          ;SAVE CONTENTS OF TSSR
5997 057350      ERRHRD    ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
      057350 104456              TRAP   C$ERHRD
      057352 001156              .WORD 622
      057354 005046'            .WORD WRTMSG
      057356 011734'            .WORD SFIMSG
5998 057360      26$:    CKLOOP      ;LOOP IF SELECTED
      057360 104406              TRAP   C$CLP1
5999
6000      ;*****
6001      ;
6002      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6003      ;
6004      ;*****
6005
6006 057362 004737 010714'      JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
6007 057366 103413              BCS    30$             ;BR, IF NO PROBLEM
6008 057370 016501 000002      MOV    TSSR(R5),R1    ;GET TSSR
6009 057374 012702 000200      MOV    #SSR,R2       ;SET UP EXPECTED TSSR
6010 057400 010004              MOV    RO,R4          ;PACKET ADDRESS SET UP
6011 057402 005237 002212'      INC    FATFLG         ;BUMP COUNT
6015 057406      ERRHRD    ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      057406 104456              TRAP   C$ERHRD
      057410 001157              .WORD 623
      057412 073364'            .WORD T26RWN
      057414 011746'            .WORD PKTSSR
6016 057416      30$:    CKLOOP      ;LOOP IF SELECTED
      057416 104406              TRAP   C$CLP1
6017
6018      ;*****
6019      ;
6020      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6021      ;
6022      ;*****
6023

```

```

6024 057420 013701 071730'      MOV      T26BFR+6,R1      ;PICK UP XSTO
6025 057424 010102              MOV      R1,R2           ;SET UP EXPECTED
6026 057426 052702 000002      BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
6027 057432 020102              CMP      R1,R2           ;DOES EXP = REC'D
6028 057434 001406              BEQ      40$             ;BR, IF EQUAL (OK)
6029 057436 005237 002212'      INC      FATFLG          ;BUMP COUNT
6033 057442              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    624
                                .WORD    T26BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
        057442 104456
        057444 001160
        057446 073075'
        057450 015374'
6034 057452              40$:  CKLOOP              ;LOOP IF SELECTED
        057452 104406              TRAP      C$CLP1
6035 057454 012703 000400      MOV      @256.,R3        ;RECORD SIZE
6036 057460 013737 003114' 072032'  MOV      FREE,T26RB      ;STARTING WRITE BUFFER ADDRESS
6037
6038 ;*****
6039 ;
6040 ;WRITE DATA,CVC=1,ACK COMMAND
6041 ;
6042 ;*****
6043
6044 057466 012737 140005 072030'  MOV      @140005,T26PK3  ;WRITE DATA,CVC=1,ACK COMMAND
6045 057474 012704 072030'      MOV      @T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
6046 057500              65$:
6047 057500 010300              MOV      R3,R0           ;SET PATTERN IN CORRECT REGISTER
6048 057502 004737 017324'      JSR      PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
6049 057506 010337 072036'      MOV      R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
6050 057512 013777 072056' 123374  MOV      T26CNT,@FREE    ;MOVE TAPE RECORD NUMBER TO BUFFER
6051 057520 062737 000001 072056'  ADD      @1,T26CNT       ;NUMBER READY FOR NEXT RECORD
6052 057526 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
6053 057532 004737 016150'      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
6054 057536 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
6055 057542 012702 000200      MOV      @SSR,R2        ;SET UP EXPECTED
6056 057546 020102              CMP      R1,R2           ;ARE THEY EQUAL
6057 057550 001406              BEQ      75$             ;BR, IF OK
6058 057552 005237 002212'      INC      FATFLG          ;BUMP COUNT
6062 057556              ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    625
                                .WORD    WRTErr
                                .WORD    PKTSSR
                                TRAP      C$CLP1
        057556 104456
        057560 001161
        057562 005103'
        057564 011746'
6063 057566              75$:  CKLOOP              ;LOOP IF SELECTED
        057566 104406              TRAP      C$CLP1
6064 057570 005723              TST      (R3)+           ;BUMP THE RECORD SIZE
6065 057572 022703 000414      CMP      @268.,R3        ;MAXIMUM SIZE YET
6066 057576 001401              BEQ      120$           ;BR, IF AT END OF WRITE SEQUENCE
6067 057600 000737              BR       65$            ;WRITE MORE RECORDS
6068 057602
6069 057602 005037 072056' 120$:  CLR      T26CNT          ;SET RECORD COUNTER BACK TO ZERO
6070
6071 ;*****
6072 ;
6073 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6074 ;
6075 ;*****
6076

```

```

6077 057606 004737 010714'      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
6078 057612 103413                BCS      130$              ;BR, IF NO PROBLEM
6079 057614 016501 000002        MOV      TSSR(R5),R1      ;GET TSSR
6080 057620 012702 000200        MOV      #SSR,R2         ;SET UP EXPECTED TSSR
6081 057624 010004                MOV      R0,R4           ;PACKET ADDRESS SET UP
6082 057626 005237 002212'      INC      FATFLG          ;BUMP COUNT
6086 057632                ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    626
                                .WORD    T26RWN
                                .WORD    PKTSSR
                                TRAP      C$CLP1
6087 057642                130$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
6088
6089
6090
6091
6092
6093
6094
6095 057644 013701 071730'      MOV      T26BFR+6,R1     ;PICK UP XSTO
6096 057650 010102                MOV      R1,R2           ;SET UP EXPECTED
6097 057652 052702 000002        BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
6098 057656 020102                CMP      R1,R2           ;DOES EXP = REC'D
6099 057660 001406                BEQ      140$            ;BR, IF EQUAL (OK)
6100 057662 005237 002212'      INC      FATFLG          ;BUMP COUNT
6104 057666                ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    627
                                .WORD    T26BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
6105 057676                140$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
6106
6107
6108
6109
6110
6111
6112
6113
6114 057700 012703 000001        MOV      #1,R3           ;SPACE 1 RECORD FORWARD
6115 057704 004737 010366'      JSR      PC,SPACE        ;SPACE CALL
6116 057710 012703 000400        MOV      #256.,R3        ;RECORD SIZE
6117 057714 013737 003114' 072032' 150$:  MOV      FREE,T26RB      ;STARTING READ BUFFER ADDRESS
6118
6119
6120
6121
6122
6123
6124
6125 057722 012737 161001 072030' 165$:  MOV      #161001,T26PK3    ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6126 057730 012704 072030'      MOV      #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
6127 057734 010337 072036'      MOV      R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
6128 057740 010465 000000        MOV      R4,TSDB(R5)     ;ISSUE COMMAND
6129 057744 004737 016150'      JSR      PC,WAITF        ;WAIT FOR SSR TO SET

```



```

060242 104456
060244 001170
060246 005046'
060250 011734'
6236 060252 268: CKLOOP ;LOOP IF SELECTED
060252 104406 TRAP C:ERHRD
;
6237 ;*****
6238 ;
6239 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6240 ;
6241 ;*****
6242
6243
6244 060254 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6245 060260 103413 BCS 308 ;BR, IF NO PROBLEM
6246 060262 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
6247 060266 012702 000200 MOV @SSR,R2 ;SET UP EXPECTED TSSR
6248 060272 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
6249 060274 005237 002212' INC FATFLG ;BUMP COUNT
6253 060300 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
060300 104456 TRAP C:ERHRD
060302 001171 .WORD 633
060304 073364' .WORD T26RWN
060306 011746' .WORD PKTSSR
6254 060310 308: CKLOOP ;LOOP IF SELECTED
060310 104406 TRAP C:CLP1
;
6255 ;*****
6256 ;
6257 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6258 ;
6259 ;*****
6260
6261
6262 060312 013701 071730' MOV T26BFR*6,R1 ;PICK UP XSTO
6263 060316 010102 MOV R1,R2 ;SET UP EXPECTED
6264 060320 052702 000002 BIS @BIT1,R2 ;SET BOT BIT IN EXPECTED
6265 060324 020102 CMP R1,R2 ;DOES EXP = REC'D
6266 060326 001406 BEQ 408 ;BR, IF EQUAL (OK)
6267 060330 005237 002212' INC FATFLG ;BUMP COUNT
6271 060334 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
060334 104456 TRAP C:ERHRD
060336 001172 .WORD 634
060340 073075' .WORD T26BOT
060342 015374' .WORD EXPREC
6272 060344 408: CKLOOP ;LOOP IF SELECTED
060344 104406 TRAP C:CLP1
6273 060346 012703 000400 MOV @256.,R3 ;RECORD SIZE
6274 060352 013737 003114' 072032' MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
6275 ;
6276 ;*****
6277 ;
6278 ;WRITE DATA,CVC=1,ACK COMMAND
6279 ;
6280 ;*****
6281
6282 060360 012737 140005 072030' MOV @140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
6283 060366 012704 072030' MOV @T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS

```

```

6284 060372
6285 060372 010300
6286 060374 004737 017324'
6287 060400 010337 072036'
6288 060404 013777 072056' 122502
6289 060412 062737 000001 072056'
6290 060420 010465 000000
6291 060424 004737 016150'
6292 060430 016501 000002
6293 060434 012702 000200
6294 060440 020102
6295 060442 001406
6296 060444 005237 002212'
6300 060450
      060450 104456
      060452 001173
      060454 005103'
      060456 011746'
6301 060460
      060460 104406
6302 060462 005723
6303 060464 022703 000412
6304 060470 001401
6305 060472 000737
6306 060474
6307 060474 005037 072056'
6308
6309
6310
6311
6312
6313
6314
6315 060500 004737 010714'
6316 060504 103413
6317 060506 016501 000002
6318 060512 012702 000200
6319 060516 010004
6320 060520 005237 002212'
6324 060524
      060524 104456
      060526 001174
      060530 073364'
      060532 011746'
6325 060534
      060534 104406
6326
6327
6328
6329
6330
6331
6332
6333 060536 013701 071730'
6334 060542 010102
6335 060544 052702 000002
6336 060550 020102

      65$:
      MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
      JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
      MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
      MOV T26CNT,0FREE ;MOVE TAPE RECORD NUMBER TO BUFFER
      ADD #1,T26CNT ;NUMBER READY FOR NEXT RECORD
      MOV R4,TSDB(R5) ;ISSUE COMMAND
      JSR PC,WAITF ;WAIT FOR SSR TO SET
      MOV TSSR(R5),R1 ;GET TSSR CONTENTS
      MOV #SSR,R2 ;SET UP EXPECTED
      CMP R1,R2 ;ARE THEY EQUAL
      BEQ 75$ ;BR, IF OK
      INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                                     TRAP C$ERHRD
                                     .WORD 635
                                     .WORD WRERR
                                     .WORD PKTSSR

      75$: CKLOOP ;LOOP IF SELECTED
                                     TRAP C$CLP1

      TST (R3) ;BUMP THE RECORD SIZE
      CMP #266.,R3 ;MAXIMUM SIZE YET
      BEQ 120$ ;BR, IF AT END OF WRITE SEQUENCE
      BR 65$ ;WRITE MORE RECORDS

      120$:
      CLR T26CNT ;SET RECORD COUNTER BACK TO ZERO

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

      JSR PC,REWIND ;CALL TAPE REWIND COMMAND
      BCS 130$ ;BR, IF NO PROBLEM
      MOV TSSR(R5),R1 ;GET TSSR
      MOV #SSR,R2 ;SET UP EXPECTED TSSR
      MOV R0,R4 ;PACKET ADDRESS SET UP
      INC FATFLG ;BUMP COUNT
      ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                     TRAP C$ERHRD
                                     .WORD 636
                                     .WORD T26RWN
                                     .WORD PKTSSR

      130$: CKLOOP ;LOOP IF SELECTED
                                     TRAP C$CLP1

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

      MOV T26BFR+6,R1 ;PICK UP XSTO
      MOV R1,R2 ;SET UP EXPECTED
      BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
      CMP R1,R2 ;DOES EXP = REC'D

```

```

6337 060552 001406          BEQ      140$          ;BR, IF EQUAL (OK)
6338 060554 005237 002212'  INC      FATFLG       ;BUMP COUNT
6342 060560          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    637
                                .WORD    T26BOT
                                .WORD    EXPREC
6343 060570          140$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    634
6344 060560 104456
6345 060562 001175
6346 060564 073075'
6347 060566 015374'
6348 060570          140$:  CKLOOP          ;LOOP IF SELECTED
6349 060570 104406          TRAP      C$CLP1
6350
6351
6352 060572 012703 000001          MOV      #1,R3          ;SET UP SPACE FORWARD 1
6353 060576 004737 010366'        JSR      PC,SPACE       ;ISSUE SPACE COMMAND
6354 060602 012703 000400          MOV      #256.,R3      ;RECORD SIZE
6355 060606 013737 003114' 072032' 150$:  MOV      FREE,T26RB    ;STARTING READ BUFFER ADDRESS
6356
6357
6358
6359
6360
6361
6362
6363 060614 012737 171001 072030' 165$:  MOV      #171001,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
6364 060622 012704 072030'        MOV      #T26PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
6365 060626 010337 072036'        MOV      R3,T26SZ      ;SET UP RECORD SIZE IN PACKET
6366 060632 010465 000000          MOV      R4,TSSB(R5)   ;ISSUE COMMAND
6367 060636 004737 016150'        JSR      PC,WAITF      ;WAIT FOR SSR TO SET
6368 060642 016501 000002          MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
6369 060646 012702 000200          MOV      #SSR,R2      ;SET UP EXPECTED
6370 060652 020102          CMP      R1,R2         ;ARE THEY EQUAL
6371 060654 001406          BEQ      170$          ;BR, IF OK
6372 060656 005237 002212'  INC      FATFLG       ;BUMP COUNT
6376 060662          ERRHRD  ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                                TRAP      C$ERHRD
                                .WORD    638
                                .WORD    T26RRF
                                .WORD    PKTSSR
6377 060672          170$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    637
6378 060674 017701 122214          MOV      @FREE,R1      ;FIRST WORD FROM READ BUFFER
6379 060700 013702 072056'        MOV      T26CNT,R2    ;SET UP EXPECTED
6380 060704 000302          SWAB    R2            ;SWAP BYTES IN EXPECTED
6381 060706 020102          CMP      R1,R2         ;IS TAPE POSITION CORRECT
6382 060710 001406          BEQ      190$          ;KEEP GOING POSITION OK
6383 060712 005237 002212'  INC      FATFLG       ;BUMP COUNT
6387 060716          ERRHRD  ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
                                TRAP      C$ERHRD
                                .WORD    639
                                .WORD    T26WNG
                                .WORD    EXPREC
6388 060726          190$:  CKLOOP

```



```

6491
6492 061176 004737 010714'      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
6493 061202 103413              BCS      30$                ;BR, IF NO PROBLEM
6494 061204 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR
6495 061210 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED TSSR
6496 061214 010004              MOV      R0,R4             ;PACKET ADDRESS SET UP
6497 061216 005237 002212'      INC      FATFLG           ;BUMP COUNT
6501 061222              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    644
                                .WORD    T26RWN
                                .WORD    PKTSSR
                                TRAP      C$CLP1
6502 061232 104406      30$:   CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
6503
6504 ;*****
6505 ;
6506 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6507 ;
6508 ;*****
6509
6510 061234 013701 071730'      MOV      T26BFR+6,R1      ;PICK UP XSTO
6511 061240 010102              MOV      R1,R2            ;SET UP EXPECTED
6512 061242 052702 000002      BIS      #BIT1,R2         ;SET BOT BIT IN EXPECTED
6513 061246 020102              CMP      R1,R2            ;DOES EXP = REC'D
6514 061250 001406              BEQ      40$              ;BR, IF EQUAL (OK)
6515 061252 005237 002212'      INC      FATFLG           ;BUMP COUNT
6519 061256              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    645
                                .WORD    T26BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
6520 061266 104406      40$:   CKLOOP              ;LOOP IF SELECTED
6521 061270 012703 001000      MOV      #512.,R3         ;RECORD SIZE
6522 061274 013737 003114' 072032'  MOV      FREE,T26RB        ;STARTING WRITE BUFFER ADDRESS
6523
6524 ;*****
6525 ;
6526 ;WRITE DATA,CVC=1,ACK COMMAND
6527 ;
6528 ;*****
6529
6530 061302 012737 140005 072030'  MOV      #140005,T26PK3    ;WRITE DATA,CVC=1,ACK COMMAND
6531 061310 012704 072030'      MOV      #T26PK3,R4       ;SET UP R4 WITH PACKET ADDRESS
6532 061314
6533 061314 010337 072036'  65$:   MOV      R3,T26SZ         ;SET UP RECORD SIZE IN PACKET
6534 061320 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
6535 061324 004737 016150'      JSR      PC,WAITF         ;WAIT FOR SSR TO SET
6536 061330 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
6537 061334 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
6538 061340 020102              CMP      R1,R2            ;ARE THEY EQUAL
6539 061342 001406              BEQ      75$              ;BR, IF OK
6540 061344 005237 002212'      INC      FATFLG           ;BUMP COUNT
6544 061350              ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    646

```



```

061354 005103'
061356 011746'
6545 061360 75$: CKLOOP ;LOOP IF SELECTED .WORD WRTERR
061360 104406 ;.WORD PKTSSR
6546 061362 005303 DEC R3 ;TRAP C$CLP1
6547 061364 013737 003114' 072032' MOV FREE,T26RB ;;SET RECORD SIZE TO 511.
;STARTING READ BUFFER ADDRESS
6548
6549 ;*****
6550 ;
6551 ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
6552 ;
6553 ;*****
6554
6555 061372 012737 161001 072030' 165$: MOV @161001,T26PK3 ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
6556 061400 012704 072030' MOV @T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6557 061404 010337 072036' MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6558 061410 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6559 061414 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
6560 061420 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6561 061424 012702 100204 MOV @SSR!SC!BIT2,R2 ;SET UP EXPECTED
6562 061430 020102 CMP R1,R2 ;ARE THEY EQUAL
6563 061432 001406 BEQ 170$ ;BR, IF OK
6564 061434 005237 002212' INC FATFLG ;BUMP COUNT
6568 061440 ERRHRD ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
061440 104456 TRAP C$ERHRD
061442 001207 .WORD 647
061444 074442' .WORD T26TRL
061446 011746' .WORD PKTSSR
6569 061450 170$: CKLOOP ;LOOP IF SELECTED .WORD TRAP C$CLP1
061450 104406
6570
6571 ;*****
6572 ;
6573 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6574 ;
6575 ;*****
6576
6577 061452 013701 071730' MOV T26BFR+6,R1 ;GET MESSAGE BUFFER
6578 061456 010102 MOV R1,R2 ;SET UP EXPECTED
6579 061460 052702 010000 BIS @BIT12,R2 ;SET THE RLL BIT IN EXPECTED
6580 061464 020102 CMP R1,R2 ;ARE THEY EQUAL
6581 061466 001406 BEQ 180$ ;BR, IF EQUAL (ALL IS WELL)
6582 061470 005237 002212' INC FATFLG ;BUMP COUNT
6586 061474 ERRHRD ERRNO,T26LON,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
061474 104456 TRAP C$ERHRD
061476 001210 .WORD 648
061500 074210' .WORD T26LON
061502 015374' .WORD EXPREC
6587 061504 180$: CKLOOP ;LOOP IF SELECTED .WORD TRAP C$CLP1
061504 104406
6588 061506 012703 000777 MOV @511.,R3 ;SET RECORD SIZE
6589 061512 013737 003114' 072032' MOV FREE,T26RB ;STARTING READ BUFFER ADDRESS
6590
6591 ;*****
6592 ;
6593 ;REREAD DATA,CVC=1,ACK COMMAND
6594 ;

```



```

6698 061754 004737 010714'      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
6699 061760 103413                BCS      30$                ;BR, IF NO PROBLEM
6700 061762 016501 000002        MOV      TSSR(R5),R1       ;GET TSSR
6701 061766 012702 000200        MOV      #SSR,R2          ;SET UP EXPECTED TSSR
6702 061772 010004                MOV      R0,R4            ;PACKET ADDRESS SET UP
6703 061774 005237 002212'      INC      FATFLG           ;BUMP COUNT
6707 062000                ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    653
                                .WORD    T26RWN
                                .WORD    PKTSSR
                                TRAP      C$CLP1
6708 062010 104406      30$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
6709
6710
6711
6712
6713
6714
6715
6716 062012 013701 071730'      MOV      T26BFR+6,R1      ;PICK UP XSTO
6717 062016 010102                MOV      R1,R2            ;SET UP EXPECTED
6718 062020 052702 000002        BIS      #BIT1,R2         ;SET BOT BIT IN EXPECTED
6719 062024 020102                CMP      R1,R2            ;DOES EXP = REC'D
6720 062026 001406                BEQ      40$              ;BR, IF EQUAL (OK)
6721 062030 005237 002212'      INC      FATFLG           ;BUMP COUNT
6725 062034                ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    654
                                .WORD    T26BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
6726 062044 104406      40$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
6727 062046 012703 000400        MOV      #256.,R3         ;RECORD SIZE
6728 062052 013737 003114' 072032'  MOV      FREE,T26RB       ;STARTING WRITE BUFFER ADDRESS
6729
6730
6731
6732
6733
6734
6735
6736 062060 012737 140005 072030'  MOV      #140005,T26PK3   ;WRITE DATA,CVC=1,ACK COMMAND
6737 062066 012704 072030'      MOV      #T26PK3,R4       ;SET UP R4 WITH PACKET ADDRESS
6738 062072                65$:
6739 062072 010337 072036'      MOV      R3,T26SZ         ;SET UP RECORD SIZE IN PACKET
6740 062076 010465 000000        MOV      R4,TSDB(R5)     ;ISSUE COMMAND
6741 062102 004737 016150'      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
6742 062106 016501 000002        MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
6743 062112 012702 000200        MOV      #SSR,R2         ;SET UP EXPECTED
6744 062116 020102                CMP      R1,R2            ;ARE THEY EQUAL
6745 062120 001406                BEQ      75$              ;BR, IF OK
6746 062122 005237 002212'      INC      FATFLG           ;BUMP COUNT
6750 062126                ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    655
                                .WORD    WRERR

```

```

062134 011746'
6751 062136 75$: CKLOOP ;LOOP IF SELECTED .WORD PKTSSR
062136 104406 ;RECORD SIZE TRAP C$CLP1
6752 062140 012703 001000 MOV #512.,R3 ;STARTING READ BUFFER ADDRESS
6753 062144 013737 003114' 072032' MOV FREE,T26RB
6754
6755 ;*****
6756 ;
6757 ;REREAD PREVIOUS,ACK,CVC=1,OPP=1
6758 ;
6759 ;*****
6760
6761 062152 012737 161001 072030' MOV #161001,T26PK3 ;REREAD PREVIOUS,ACK,CVC=1,OPP=1
6762 062160 012704 072030' 165$: MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6763 062164 010337 072036' MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6764 062170 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6765 062174 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
6766 062200 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6767 062204 012702 100204 MOV #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6768 062210 020102 CMP R1,R2 ;ARE THEY EQUAL
6769 062212 001406 BEQ 170$ ;BR, IF OK
6770 062214 005237 002212' INC FATFLG ;BUMP COUNT
6774 062220 ERRHRD ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
062220 104456 TRAP C$ERHRD
062222 001220 .WORD 656
062224 074442' .WORD T26TRL
062226 011746' .WORD PKTSSR
6775 062230 170$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
062230 104406
6776
6777 ;*****
6778 ;
6779 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6780 ;
6781 ;*****
6782
6783 062232 013701 071730' MOV T26BFR+6,R1 ;GET MESSAGE BUFFER
6784 062236 010102 MOV R1,R2 ;SET UP EXPECTED
6785 062240 052702 040000 BIS #BIT14,R2 ;SET THE RLS BIT IN EXPECTED
6786 062244 020102 CMP R1,R2 ;ARE THEY EQUAL
6787 062246 001406 BEQ 180$ ;BR, IF EQUAL (ALL IS WELL)
6788 062250 005237 002212' INC FATFLG ;BUMP COUNT
6792 062254 ERRHRD ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
062254 104456 TRAP C$ERHRD
062256 001221 .WORD 657
062260 074272' .WORD T26LOP
062262 015374' .WORD EXPREC
6793 062264 180$: CKLOOP TRAP C$CLP1
062264 104406
6794 062266 013701 071726' MOV T26BFR+4,R1 ;PICK UP RESIDUAL BYTE COUNTER
6795 062272 012702 000400 MOV #256.,R2 ;THIS SHOULD BE THE DIFFERENCE
6796 062276 020102 CMP R1,R2 ;IS THE DIFFERENCE CORRECT
6797 062300 001406 BEQ 190$ ;BR, IF CORRECT
6798 062302 005237 002212' INC FATFLG ;BUMP COUNT
6802 062306 ERRHRD ERRNO,T26PBP,EXPREC ;NBPCR NOT CORRECT
062306 104456 TRAP C$ERHRD
062310 001222 .WORD 658

```

```

062312 074354'
062314 015374'
6803 062316 104406 190$: CKLOOP ;LOOP IF SELECTED .WORD T26PBP
062316 104406 ;RECORD SIZE .WORD EXPREC
6804 062320 012703 001000 MOV #512.,R3 ;TRAP C$CLP1
6805 062324 013737 003114' 072032' MOV FREE,T26RB ;STARTING READ BUFFER ADDRESS
6806
6807 ;*****
6808 ;
6809 ;REREAD PREVIOUS,ACK,CVC=1,OPP=0
6810 ;
6811 ;*****
6812
6813 062332 012737 141001 072030' MOV #141001,T26PK3 ;REREAD PREVIOUS,ACK,CVC=1,OPP=0
6814 062340 012704 072030' MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6815 062344 010337 072036' MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
6816 062350 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
6817 062354 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
6818 062360 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6819 062364 012702 100204 MOV #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6820 062370 020102 CMP R1,R2 ;ARE THEY EQUAL
6821 062372 001406 BEQ 270$ ;BR, IF OK
6822 062374 005237 002212' INC FATFLG ;BUMP COUNT
6826 062400 ERRHRD ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
062400 104456 TRAP C$ERHRD
062402 001223 .WORD 659
062404 074442' .WORD T26TRL
062406 011746' .WORD PKTSSR
6827 062410 104406 270$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
062410 104406
6828
6829 ;*****
6830 ;
6831 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6832 ;
6833 ;*****
6834
6835 062412 013701 071730' MOV T26BFR+6,R1 ;GET MESSAGE BUFFER
6836 062416 010102 MOV R1,R2 ;SET UP EXPECTED
6837 062420 052702 040000 BIS #BIT14,R2 ;SET THE RLS BIT IN EXPECTED
6838 062424 020102 CMP R1,R2 ;ARE THEY EQUAL
6839 062426 001406 BEQ 280$ ;BR, IF EQUAL (ALL IS WELL)
6840 062430 005237 002212' INC FATFLG ;BUMP COUNT
6844 062434 ERRHRD ERRNO,T26LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
062434 104456 TRAP C$ERHRD
062436 001224 .WORD 660
062440 074272' .WORD T26LOP
062442 015374' .WORD EXPREC
6845 062444 104406 280$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
062444 104406
6846 062446 013701 071726' MOV T26BFR+4,R1 ;PICK UP RESIDUAL BYTE COUNTER
6847 062452 012702 000400 MOV #256.,R2 ;THIS SHOULD BE THE DIFFERENCE
6848 062456 020102 CMP R1,R2 ;IS THE DIFFERENCE CORRECT
6849 062460 001405 BEQ 290$ ;BR, IF CORRECT
6853 062464 ERRHRD ERRNO,T26PBP,EXPREC ;RBPBR NOT CORRECT
062464 104456 TRAP C$ERHRD
062466 001224 .WORD 660

```



```

6903
6904 ;*****
6905 ;
6906 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
6907 ;
6908 ;*****
6909
6910 062570 004737 010562'      JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
6911 062574 103407              BCS    26$           ;BR, IF COMMAND ISSUED OK
6912 062576 005237 002212'      INC    FATFLG        ;BUMP COUNT
6916 062602 010001              MOV    RO,R1         ;SAVE CONTENTS OF TSSR
6917 062604              ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP    C$ERHRD
                                .WORD   662
                                .WORD   WRTMSG
                                .WORD   SFIMSG
                                062604 104456
                                062606 001226
                                062610 005046'
                                062612 011734'
6918 062614              26$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                062614 104406
6919
6920 ;*****
6921 ;
6922 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6923 ;
6924 ;*****
6925
6926 062616 004737 010714'      JSR    PC,REWIND     ;CALL TAPE REWIND COMMAND
6927 062622 103413              BCS    30$           ;BR, IF NO PROBLEM
6928 062624 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR
6929 062630 012702 000200      MOV    @SSR,R2       ;SET UP EXPECTED TSSR
6930 062634 010004              MOV    RO,R4         ;PACKET ADDRESS SET UP
6931 062636 005237 002212'      INC    FATFLG        ;BUMP COUNT
6935 062642              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD   663
                                .WORD   T26RWN
                                .WORD   PKTSSR
                                062642 104456
                                062644 001227
                                062646 C73364'
                                062650 011746'
6936 062652              30$:   CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                062652 104406
6937
6938 ;*****
6939 ;
6940 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6941 ;
6942 ;*****
6943
6944 062654 013701 071730'      MOV    T26BFR+6,R1   ;PICK UP XSTO
6945 062660 010102              MOV    R1,R2         ;SET UP EXPECTED
6946 062662 052702 000002      BIS    @BIT1,R2      ;SET BOT BIT IN EXPECTED
6947 062666 020102              CMP    R1,R2         ;DOES EXP = REC'D
6948 062670 001406              BEQ    40$           ;BR, IF EQUAL (OK)
6949 062672 005237 002212'      INC    FATFLG        ;BUMP COUNT
6953 062676              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD   664
                                .WORD   T26BOT
                                .WORD   EXPREC
                                062676 104456
                                062700 001230
                                062702 073075'
                                062704 015374'
6954 062706              40$:   CKLOOP                ;LOOP IF SELECTED

```



```

6955 062706 104406          MOV      #256.,R3          ;RECORD SIZE          TRAP      C$CLP1
6956 062710 012703 000400  MOV      FREE,T26RB      ;STARTING WRITE BUFFER ADDRESS
6957 062714 013737 003114' 072032'
6958 ;*****
6959 ;
6960 ;WRITE DATA,CVC=1,ACK COMMAND
6961 ;
6962 ;*****
6963
6964 062722 012737 140005 072030'  MOV      #140005,T26PK3  ;WRITE DATA,CVC=1,ACK COMMAND
6965 062730 012704 072030'  MOV      #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
6966 062734
6967 062734 010300 65$:  MOV      R3,R0          ;SET PATTERN IN CORRECT REGISTER
6968 062736 004737 017324'  JSR      PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
6969 062742 010337 072036'  MOV      R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
6970 062746 010465 000000  MOV      R4,TSDB(R5)     ;ISSUE COMMAND
6971 062752 004737 016150'  JSR      PC,WAITF        ;WAIT FOR SSR TO SET
6972 062756 016501 000002  MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
6973 062762 012702 000200  MOV      #SSR,R2        ;SET UP EXPECTED
6974 062766 020102  CMP      R1,R2          ;ARE THEY EQUAL
6975 062770 001406  BEQ      75$            ;BR, IF OK
6976 062772 005237 002212'  INC      FATFLG         ;BUMP COUNT
6980 062776          ERRHRD  ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
6981 062776 104456          TRAP      C$ERHRD
6982 063000 001231          .WORD    665
6983 063002 005103'      .WORD    WRERR
6984 063004 011746'      .WORD    PKTSSR
6981 063006          75$:  CKLOOP          ;LOOP IF SELECTED
6982 063006 104406          TRAP      C$CLP1
6983 063010 005723          TST      (R3)+          ;BUMP RECORD SIZE
6984 063012 022703 000414  CMP      #268.,R3      ;END OF RECORD YET
6985 063016 001346  BNE      65$            ;BR, IF MORE RECORDS TO WRITE
6986 063020          80$:  CKLOOP          ;LOOP IF SELECTED
6987 063022 104406          TRAP      C$CLP1
6988 ;*****
6989 ;
6990 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6991 ;
6992 ;*****
6993
6994 063022 004737 010714'  JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
6995 063026 103413  BCS      130$          ;BR, IF NO PROBLEM
6996 063030 016501 000002  MOV      TSSR(R5),R1     ;GET TSSR
6997 063034 012702 000200  MOV      #SSR,R2        ;SET UP EXPECTED TSSR
6998 063040 010004  MOV      R0,R4          ;PACKET ADDRESS SET UP
6999 063042 005237 002212'  INC      FATFLG         ;BUMP COUNT
7003 063046          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
7004 063046 104456          TRAP      C$ERHRD
7005 063050 001232          .WORD    666
7006 063052 073364'      .WORD    T26RWN
7007 063054 011746'      .WORD    PKTSSR
7004 063056          130$: CKLOOP          ;LOOP IF SELECTED
7005 063056 104406          TRAP      C$CLP1

```

```

7006 ;*****
7007 ;
7008 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7009 ;
7010 ;*****
7011
7012 063060 013701 071730'      MOV      T26BFR+6,R1      ;PICK UP XSTO
7013 063064 010102      MOV      R1,R2           ;SET UP EXPECTED
7014 063066 052702 000002      BIS      @BIT1,R2       ;SET BOT BIT IN EXPECTED
7015 063072 020102      CMP      R1,R2          ;DOES EXP = REC'D
7016 063074 001406      BEQ      140$          ;BR, IF EQUAL (OK)
7017 063076 005237 002212'      INC      FATFLG         ;BUMP COUNT
7021 063102      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      063102 104456      TRAP      C$ERHRD
      063104 001233      .WORD    667
      063106 073075'      .WORD    T26BOT
      063110 015374'      .WORD    EXPREC
7022 063112      140$:  CKLOOP          ;LOOP IF SELECTED
      063112 104406      TRAP      C$CLP1
7023 063114 012737 000400 072062'  MOV      @256.,T26RSZ   ;STORE START RECORD SIZE
7024 063122 000420      BR       150$          ;SKIP THE SAPCE THIS TIME
7025
7026 ;*****
7027 ;
7028 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
7029 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
7030 ;
7031 ;*****
7032
7033 063124 012703 000001      145$:  MOV      @1,R3    ;SPACE ONE RECORD PARAMETER
7034 063130 004737 010366'      JSR      PC,SPACE     ;CALL SPACE ROUTINE
7035 063134 103413      BCS      150$         ;BR, IF NO PROBLEM WITH SPACE COMMAND
7036 063136 016501 000002      MOV      TSSR(R5),R1  ;GET TSSR
7037 063142 012702 000200      MOV      @SSR,R2     ;SET UP EXPECTED TSSR
7038 063146 010004      MOV      R0,R4       ;PACKET ADDRESS SET UP
7039 063150 005237 002212'      INC      FATFLG     ;BUMP COUNT
7043 063154      ERRHRD  ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
      063154 104456      TRAP      C$ERHRD
      063156 001234      .WORD    668
      063160 072477'      .WORD    T26SC
      063162 015374'      .WORD    EXPREC
7044 063164      150$:  CKLOOP          ;LOOP IF SELECTED
      063164 104406      TRAP      C$CLP1
7045 063166 013703 072062'      MOV      T26RSZ,R3    ;RECORD SIZE
7046 063172 013737 003114' 072032'  MOV      FREE,T26RB   ;STARTING READ BUFFER ADDRESS
7047
7048 ;*****
7049 ;
7050 ;REREREAD DATA,CVC=1,ACK COMMAND
7051 ;
7052 ;*****
7053
7054 063200 012737 141401 072030'      165$:  MOV      @141401,T26PK3 ;REREREAD DATA,CVC=1,ACK COMMAND
7055 063206 012704 072030'      MOV      @T26PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
7056 063212 010337 072036'      MOV      R3,T26SZ    ;SET UP RECORD SIZE IN PACKET
7057 063216 010465 000000      MOV      R4,TSDB(R5) ;ISSUE COMMAND
7058 063222 004737 016150'      JSR      PC,WAITF    ;WAIT FOR SSR TO SET

```



```

7161 063510 005237 002212'          INC    FATFLG          ;BUMP COUNT
7165 063514          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      063514 104456          TRAP    C$ERHRD
      063516 001241          .WORD  673
      063520 073364'          .WORD  T26RWN
      063522 011746'          .WORD  PKTSSR
7166 063524          30$:   CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      063524 104406
7167
7168          ;*****
7169          ;
7170          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7171          ;
7172          ;*****
7173
7174 063526 013701 071730'          MOV    T26BFR+6,R1          ;PICK UP XSTO
7175 063532 010102          MOV    R1,R2              ;SET UP EXPECTED
7176 063534 052702 000002          BIS    #8BIT1,R2          ;SET BOT BIT IN EXPECTED
7177 063540 020102          CMP    R1,R2              ;DOES EXP = REC'D
7178 063542 001406          BEQ    40$                ;BR, IF EQUAL (OK)
7179 063544 005237 002212'          INC    FATFLG          ;BUMP COUNT
7183 063550          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      063550 104456          TRAP    C$ERHRD
      063552 001242          .WORD  674
      063554 073075'          .WORD  T26BOT
      063556 015374'          .WORD  EXPREC
7184 063560          40$:   CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      063560 104406
7185 063562 012703 000400          MOV    #256.,R3          ;RECORD SIZE
7186 063566 013737 003114' 072032'  MOV    FREE,T26RB          ;STARTING WRITE BUFFER ADDRESS
7187
7188          ;*****
7189          ;
7190          ;WRITE DATA,CVC=1,ACK,SWB COMMAND
7191          ;
7192          ;*****
7193
7194 063574 012737 150005 072030'  MOV    #150005,T26PK3      ;WRITE DATA,CVC=1,ACK,SWB COMMAND
7195 063602 012704 072030'          MOV    #T26PK3,R4          ;SET UP R4 WITH PACKET ADDRESS
7196 063606          65$:
7197 063606 010300          MOV    R3,R0              ;SET PATTERN IN CORRECT REGISTER
7198 063610 004737 017324'          JSR    PC,FILLMEM          ;FILL MEMORY WITH RECORD SIZE
7199 063614 010337 072036'          MOV    R3,T26SZ          ;SET UP RECORD SIZE IN PACKET
7200 063620 010465 000000          MOV    R4,TSDB(R5)        ;ISSUE COMMAND
7201 063624 004737 016150'          JSR    PC,WAITF           ;WAIT FOR SSR TO SET
7202 063630 016501 000002          MOV    TSSR(R5),R1        ;GET TSSR CONTENTS
7203 063634 012702 000200          MOV    #SSR,R2           ;SET UP EXPECTED
7204 063640 020102          CMP    R1,R2              ;ARE THEY EQUAL
7205 063642 001406          BEQ    75$                ;BR, IF OK
7206 063644 005237 002212'          INC    FATFLG          ;BUMP COUNT
7210 063650          ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      063650 104456          TRAP    C$ERHRD
      063652 001243          .WORD  675
      063654 005103'          .WORD  WRTErr
      063656 011746'          .WORD  PKTSSR
7211 063660          75$:   CKLOOP          ;LOOP IF SELECTED          TRAP    C$CLP1
      063660 104406

```

```

7212 063662 005723          TST      (R3)+          ;BUMP RECORD SIZE
7213 063664 022703 000414  CMP      #268.,R3      ;END OF RECORD YET
7214 063670 001346          BNE      65$           ;BR, IF MORE RECORDS TO WRITE
7215 063672          80$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
7216 063674          120$:
7217
7218          ;*****
7219          ;
7220          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7221          ;
7222          ;*****
7223
7224 063674 004737 010714'   JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
7225 063700 103413          BCS      130$          ;BR, IF NO PROBLEM
7226 063702 016501 000002   MOV      TSSR(R5),R1   ;GET TSSR
7227 063706 012702 000200   MOV      #SSR,R2       ;SET UP EXPECTED TSSR
7228 063712 010004          MOV      R0,R4         ;PACKET ADDRESS SET UP
7229 063714 005237 002212'   INC      FATFLG        ;BUMP COUNT
7233 063720          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    676
                                .WORD    T26RWN
                                .WORD    PKTSSR
7234 063730          130$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
7235
7236          ;*****
7237          ;
7238          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7239          ;
7240          ;*****
7241
7242 063732 013701 071730'   MOV      T26BFR+6,R1   ;PICK UP XSTO
7243 063736 010102          MOV      R1,R2         ;SET UP EXPECTED
7244 063740 052702 000002   BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
7245 063744 020102          CMP      R1,R2         ;DOES EXP = REC'D
7246 063746 001406          BEQ      140$          ;BR, IF EQUAL (OK)
7247 063750 005237 002212'   INC      FATFLG        ;BUMP COUNT
7251 063754          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    677
                                .WORD    T26BOT
                                .WORD    EXPREC
7252 063764          140$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
7253 063766 012737 000400 072062' MOV      #256.,T26RSZ  ;START RECORD SIZE
7254 063774 000420          BR       150$          ;SKIP SAPCE THIS TIME
7255
7256          ;*****
7257          ;
7258          ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
7259          ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
7260          ;
7261          ;*****
7262
7263 063776 012703 000001   145$:  MOV      #1,R3          ;SPACE ONE RECORD PARAMETER

```

```

7264 064002 004737 010366'      JSR      PC,SPACE          ;CALL SPACE ROUTINE
7265 064006 103413              BCS      150$              ;BR, IF NO PROBLEM WITH SPACE COMMAND
7266 064010 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR
7267 064014 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED TSSR
7268 064020 010004              MOV      R0,R4           ;PACKET ADDRESS SET UP
7269 064022 005237 002212'      INC      FATFLG          ;BUMP COUNT
7273 064026              ERRHRD  ERRNO,T26SC,EXPREC ;POSITION (SPACE RECORDS) FAILED
      064026 104456              TRAP    C$ERHRD
      064030 001246              .WORD  678
      064032 072477'          .WORD  T26SC
      064034 015374'          .WORD  EXPREC
7274 064036              150$:  CKLOOP              TRAP    C$CLP1
      064036 104406
7275 064040 013703 072062'      MOV      T26RSZ,R3       ;RECORD SIZE
7276 064044 013737 003114' 072032'  MOV      FREE,T26RB      ;STARTING READ BUFFER ADDRESS
7277
7278 ;*****
7279 ;
7280 ;REREAD DATA,ACK,CVC=1,SWB COMMAND
7281 ;
7282 ;*****
7283
7284 064052 012737 151401 072030'  MOV      #151401,T26PK3  ;REREAD DATA,ACK,CVC=1,SWB COMMAND
7285 064060 012704 072030'  165$:  MOV      #T26PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
7286 064064 010337 072036'      MOV      R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
7287 064070 010465 000000      MOV      R4,TSDB(R5)     ;ISSUE COMMAND
7288 064074 004737 016150'      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
7289 064100 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
7290 064104 012702 000200      MOV      #SSR,R2         ;SET UP EXPECTED
7291 064110 020102              CMP      R1,R2           ;ARE THEY EQUAL
7292 064112 001406              BEQ      170$            ;BR, IF OK
7293 064114 005237 002212'      INC      FATFLG          ;BUMP COUNT
7297 064120              ERRHRD  ERRNO,T26WDC,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
      064120 104456              TRAP    C$ERHRD
      064122 001247              .WORD  679
      064124 073720'          .WORD  T26WDC
      064126 011746'          .WORD  PKTSSR
7298 064130              170$:  CKLOOP              ;LOOP IF SELECTED
      064130 104406              TRAP    C$CLP1
7299 064132 013702 003114'      MOV      FREE,R2         ;CURRENT BUFFER ADDRESS TO R2
7300 064136 010304              MOV      R3,R4           ;CURRENT RECORD SIZE
7301 064140 162704 000400      SUB      #256.,R4        ;FIRST LOCATION IN BUFFER
7302 064144 060204              173$:  ADD      R2,R4         ;SET UP POINTER
7303 064146 021403              CMP      (R4),R3         ;CHECK DATA READ (R3=DATA ALSO)
7304 064150 001410              BEQ      180$            ;BR, IF ALL IS WELL
7305 064152 011401              MOV      (R4),R1         ;RECD DATA
7306 064154 010302              MOV      R3,R2           ;EXPECTED DATA
7307 064156 005237 002212'      INC      FATFLG          ;BUMP COUNT
7311 064162              ERRHRD  ERRNO,T26DTA,EXPREC ;DATA READ NOT = WRITTEN
      064162 104456              TRAP    C$ERHRD
      064164 001250              .WORD  680
      064166 073142'          .WORD  T26DTA
      064170 015374'          .WORD  EXPREC
7312 064172              180$:  CKLOOP              ;LOOP IF SELECTED
      064172 104406              TRAP    C$CLP1
7313 064174 005724              TST      (R4)+           ;BUMP TO NEXT LOCATION
7314 064176 160204              SUB      R2,R4           ;CORRECT RECORDS SIZE VALUE

```



```

7367
7368
7369
7370
7371
7372
7373 064262 004737 015674'
7374 064266 103407
7375 064270 005237 002212'
7379 064274 010001
7380 064276
064276 104455
064300 001251
064302 003642'
064304 011734'
7381 064306 013737 002172' 071720' 20$:
7382
7383 064314 012704 071700'
7384
7385
7386
7387
7388
7389
7390
7391 064320 004737 010562'
7392 064324 103407
7393 064326 005237 002212'
7397 064332 010001
7398 064334
064334 104456
064336 001252
064340 005046'
064342 011734'
7399 064344
064344 104406
7400
7401
7402
7403
7404
7405
7406
7407 064346 004737 010714'
7408 064352 103413
7409 064354 016501 000002
7410 064360 012702 000200
7411 064364 010004
7412 064366 005237 002212'
7416 064372
064372 104456
064374 001253
064376 073364'
064400 011746'
7417 064402
064402 104406
7418

```

```

;*****
;
;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
;
;*****
JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
BCS 20$ ;BR IF INIT WAS OK
INC FATFLG ;BUMP COUNT
MOV RO,R1 ;CONTENTS OF TSSR REGISTER
ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
TRAP C$ERDF
.WORD 681
.WORD SFIERR
.WORD SFIMSG
MOV UNITN,T26DSW ;SET UP UNIT NUMBER
MOV @T26PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
;*****
;
;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
;
;*****
JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
BCS 26$ ;BR, IF COMMAND ISSUED OK
INC FATFLG ;BUMP COUNT
MOV RO,R1 ;SAVE CONTENTS OF TSSR
ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
TRAP C$ERHRD
.WORD 682
.WORD WRTMSG
.WORD SFIMSG
26$: CKLOOP ;LOOP IF SELECTED
TRAP C$CLP1
;*****
;
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
JSR PC,REWIND ;CALL TAPE REWIND COMMAND
BCS 30$ ;BR, IF NO PROBLEM
MOV TSSR(R5),R1 ;GET TSSR
MOV @SSR,R2 ;SET UP EXPECTED TSSR
MOV RO,R4 ;PACKET ADDRESS SET UP
INC FATFLG ;BUMP COUNT
ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
TRAP C$ERHRD
.WORD 683
.WORD T26RWN
.WORD PKTSSR
30$: CKLOOP ;LOOP IF SELECTED
TRAP C$CLP1

```

```

7419 ;*****
7420 ;
7421 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7422 ;
7423 ;*****
7424
7425 064404 013701 071730'      MOV      T26BFR+6,R1      ;PICK UP XSTO
7426 064410 010102              MOV      R1,R2           ;SET UP EXPECTED
7427 064412 052702 000002      BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
7428 064416 020102              CMP      R1,R2           ;DOES EXP = REC'D
7429 064420 001406              BEQ     40$              ;BR, IF EQUAL (OK)
7430 064422 005237 002212'      INC     FATFLG           ;BUMP COUNT
7434 064426              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP     C$ERHRD
                                .WORD    684
                                .WORD    T26BOT
                                .WORD    EXPREC
7435 064436              40$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP     C$CLP1
7436 064440 012703 000400      MOV     @256.,R3         ;RECORD SIZE
7437 064444 013737 003114' 072032'  MOV     FREE,T26RB      ;STARTING WRITE BUFFER ADDRESS
7438
7439 ;*****
7440 ;
7441 ;WRITE DATA,CVC=1,ACK COMMAND
7442 ;
7443 ;*****
7444
7445 064452 012737 140005 072030'  MOV     @140005,T26PK3  ;WRITE DATA,CVC=1,ACK COMMAND
7446 064460 012704 072030'      MOV     @T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
7447 064464              65$:
7448 064464 010337 072036'      MOV     R3,T26SZ        ;SET UP RECORD SIZE IN PACKET
7449 064470 013777 072056' 116416  MOV     T26CNT,@FREE    ;MOVE TAPE RECORD NUMBER TO BUFFER
7450 064476 062737 000001 072056'  ADD     @1,T26CNT       ;NUMBER READY FOR NEXT RECORD
7451 064504 010465 000000      MOV     R4,TSDB(R5)     ;ISSUE COMMAND
7452 064510 004737 016150'      JSR    PC,WAITF        ;WAIT FOR SSR TO SET
7453 064514 016501 000002      MOV     TSSR(R5),R1     ;GET TSSR CONTENTS
7454 064520 012702 000200      MOV     @SSR,R2         ;SET UP EXPECTED
7455 064524 020102              CMP     R1,R2           ;ARE THEY EQUAL
7456 064526 001406              BEQ     75$              ;BR, IF OK
7457 064530 005237 002212'      INC     FATFLG           ;BUMP COUNT
7461 064534              ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                                TRAP     C$ERHRD
                                .WORD    685
                                .WORD    WRTErr
                                .WORD    PKTSSR
7462 064544              75$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP     C$CLP1
7463 064546 005723              TST     (R3)+           ;BUMP THE RECORD SIZE
7464 064550 022703 000414      CMP     @268.,R3       ;MAXIMUM SIZE YET
7465 064554 001401              BEQ     120$            ;BR, IF AT END OF WRITE SEQUENCE
7466 064556 000742              BR      65$             ;WRITE MORE RECORDS
7467 064560              120$:
7468 064560 005037 072056'      CLR     T26CNT          ;SET RECORD COUNTER BACK TO ZERO
7469
7470 ;*****
7471 ;

```

```

7472      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7473      ;
7474      ;*****
7475
7476 064564 004737 010714'      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
7477 064570 103411              BCS      130$           ;BR, IF NO PROBLEM
7478 064572 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR
7479 064576 010004              MOV      R0,R4          ;PACKET ADDRESS SET UP
7480 064600 005237 002212'      INC      FATFLG         ;BUMP COUNT
7484 064604              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
          064604 104456              TRAP    C$ERHRD
          064606 001256              .WORD  686
          064610 073364'              .WORD  T26RWN
          064612 011746'              .WORD  PKTSSR
7485 064614              130$:  CKLOOP          ;LOOP IF SELECTED
          064614 104406              TRAP    C$CLP1
7486
7487      ;*****
7488      ;
7489      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7490      ;
7491      ;*****
7492
7493 064616 013701 071730'      MOV      T26BFR+6,R1   ;PICK UP XSTO
7494 064622 010102              MOV      R1,R2         ;SET UP EXPECTED
7495 064624 052702 000002      BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
7496 064630 020102              CMP      R1,R2         ;DOES EXP = REC'D
7497 064632 001406              BEQ     135$           ;BR, IF EQUAL (OK)
7498 064634 005237 002212'      INC      FATFLG         ;BUMP COUNT
7502 064640              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          064640 104456              TRAP    C$ERHRD
          064642 001257              .WORD  687
          064644 073075'              .WORD  T26BOT
          064646 015374'              .WORD  EXPREC
7503 064650              135$:  CKLOOP          ;LOOP IF SELECTED
          064650 104406              TRAP    C$CLP1
7504 064652 012737 000400 072062' MOV      #256.,T26RSZ   ;STARTING RECORD SIZE
7505 064660 000420              BR       140$          ;SKIP OVER THE SAPCE THIS TIME
7506
7507      ;*****
7508      ;
7509      ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
7510      ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
7511      ;
7512      ;*****
7513
7514 064662 012703 000001      132$:  MOV      #000001,R3 ;SET UP SPACE COMMAND (1 FORWARD)
7515 064666 004737 010366'      JSR      PC,SPACE      ;CALL SPACE ROUTINE
7516 064672 103413              BCS      140$           ;BR, IF NO TROUBLE
7517 064674 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR
7518 064700 012702 000200      MOV      #SSR,R2       ;SET UP EXPECTED TSSR
7519 064704 010004              MOV      R0,R4          ;PACKET ADDRESS SET UP
7520 064706 005237 002212'      INC      FATFLG         ;BUMP COUNT
7524 064712              ERRHRD  ERRNO,T26SC,PKTSSR ;SPACE (FORWARD) FAILED
          064712 104456              TRAP    C$ERHRD
          064714 001260              .WORD  688
          064716 072477'              .WORD  T26SC

```

```

7525 064720 011746'          140$:  CKLOOP                ;LOOP IF SELECTED                .WORD  PKTSSR
      064722                ;RECORD SIZE                      TRAP   C:CLP1
      064722 104406          ;STARTING READ BUFFER ADDRESS
7526 064724 013703 072062'   MOV    T26RSZ,R3
7527 064730 013737 003114' 072032' 150$:  MOV    FREE,T26RB
7528
7529 ;*****
7530 ;
7531 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
7532 ;
7533 ;*****
7534
7535 064736 012737 161401 072030'   MOV    @161401,T26PK3           ;REREAD DATA,CVC=1,ACK, OPP COMMAND
7536 064744 012704 072030'   MOV    @T26PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
7537 064750 010337 072036'   MOV    R3,T26SZ                ;SET UP RECORD SIZE IN PACKET
7538 064754 010463 000000      MOV    R4,TSDB(R5)             ;ISSUE COMMAND
7539 064760 004737 016150'   JSR    PC,WAITF                ;WAIT FOR SSR TO SET
7540 064764 016501 000002      MOV    TSSR(R5),R1            ;GET TSSR CONTENTS
7541 064770 012702 000200      MOV    @SSR,R2                ;SET UP EXPECTED
7542 064774 020102            CMP    R1,R2                   ;ARE THEY EQUAL
7543 064776 001406            BEQ    170$                    ;BR, IF OK
7544 065000 005237 002212'   INC    FATFLG                  ;BUMP COUNT
7548 065004            ERRHRD  ERRNO,T26RRF,PKTSSR     ;TSSR INCORRECT AFTER REREAD DATA
      065004 104456                TRAP   C:ERHRD
      065006 001261                .WORD 689
      065010 072305'                .WORD T26RRF
      065012 011746'                .WORD  PKTSSR
7549 065014            170$:  CKLOOP                ;LOOP IF SELECTED                TRAP   C:CLP1
      065014 104406                ;FIRST WORD FROM READ BUFFER
7550 065016 017701 116072      MOV    @FREE,R1                ;SET UP EXPECTED
7551 065022 013702 072056'   MOV    T26CNT,R2              ;IS TAPE POSITION CORRECT
7552 065026 020102            CMP    R1,R2                   ;KEEP GOING POSITION OK
7553 065030 001406            BEQ    190$                    ;BUMP COUNT
7554 065032 005237 002212'   INC    FATFLG                  ;TAPE POSITION INCORRECT
7558 065036            ERRHRD  ERRNO,T26WNG,EXPREC
      065036 104456                TRAP   C:ERHRD
      065040 001262                .WORD 690
      065042 072066'                .WORD T26WNG
      065044 015374'                .WORD  EXPREC
7559 065046            190$:  CKLOOP                ;LOOP IF SELECTED                TRAP   C:CLP1
      065046 104406                ;BUMP TAPE RECORD COUNTER
7560 065050 062737 000001 072056'  ADD    @1,T26CNT              ;NEXT RECORD SIZE
7561 065056 005723            TST    (R3)                    ;STORE RECORD SIZE
7562 065060 010337 072062'   MOV    R3,T26RSZ              ;AT MAX SIZE YET
7563 065064 022703 000412      CMP    @266.,R3               ;BR, IF AT END OF THE SUBTEST
7564 065070 001402            BEQ    200$                    ;KEEP GOING MORE RECORDS
7565 065072 000137 064662'   JMP    132$
7566 065076            200$:  ENDSUB                ;***** END SUBTEST *****
7567 065076                L10113:
      065076 104403                TRAP   C:ESUB
7568 065100 023727 002212' 000017  CMP    FATFLG,@15.            ;IS ERROR COUNT AT 25
7569 065106 103402            BLO    999$                    ;BR, IF LESS THAN 25
7570 065110 004737 017104'   JSR    PC,CKDROP              ;TRY TO DROP THE UNIT
7571 065114            999$:
7572
7573

```



```

065220 104406 TRAP C$CLP1
7627
7628 ;*****
7629 ;
7630 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7631 ;
7632 ;*****
7633
7634 065222 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
7635 065226 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
7636 065232 012702 000200 MOV @SSR,R2 ;SET UP EXPECTED TSSR
7637 065236 103407 BCS 30$ ;BR, IF NO PROBLEM
7638 065240 010004 MOV RO,R4 ;PACKET ADDRESS SET UP
7639 065242 005237 002212' INC FATFLG ;BUMP COUNT
7643 065246 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
065246 104456 TRAP C$ERHRD
065250 001265 .WORD 693
065252 073364' .WORD T26RWN
065254 011746' .WORD PKTSSR
7644 065256 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
065256 104406
7645
7646 ;*****
7647 ;
7648 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7649 ;
7650 ;*****
7651
7652 065260 013701 071730' MOV T26BFR+6,R1 ;PICK UP XSTO
7653 065264 010102 MOV R1,R2 ;SET UP EXPECTED
7654 065266 052702 000002 BIS @BIT1,R2 ;SET BOT BIT IN EXPECTED
7655 065272 020102 CMP R1,R2 ;DOES EXP = REC'D
7656 065274 001406 BEQ 40$ ;BR, IF EQUAL (OK)
7657 065276 005237 002212' INC FATFLG ;BUMP COUNT
7661 065302 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
065302 104456 TRAP C$ERHRD
065304 001266 .WORD 694
065306 073075' .WORD T26BOT
065310 015374' .WORD EXPREC
7662 065312 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
065312 104406
7663 065314 012703 000400 MOV @256.,R3 ;RECORD SIZE
7664 065320 013737 003114' 072032' MOV FREE,T26RB ;STARTING WRITE BUFFER ADDRESS
7665
7666 ;*****
7667 ;
7668 ;WRITE DATA,CVC-1,ACK COMMAND
7669 ;
7670 ;*****
7671
7672 065326 012737 140005 072030' MOV @140005,T26PK3 ;WRITE DATA,CVC-1,ACK COMMAND
7673 065334 012704 072030' MOV @T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
7674 065340
7675 065340 010337 072036' 65$: MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
7676 065344 013777 072056' 115542 MOV T26CNT,@FREE ;MOVE TAPE RECORD NUMBER TO BUFFER
7677 065352 062737 000001 072056' ADD @1,T26CNT ;NUMBER READY FOR NEXT RECORD
7678 065360 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND

```

```

7679 065364 004737 016150'      JSR    PC, WAITF      ;WAIT FOR SSR TO SET
7680 065370 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
7681 065374 012702 000200      MOV    #SSR,R2      ;SET UP EXPECTED
7682 065400 020102              CMP    R1,R2        ;ARE THEY EQUAL
7683 065402 001406              BEQ    75$          ;BR, IF OK
7684 065404 005237 002212'      INC    FATFLG       ;BUMP COUNT
7688 065410              ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERHRD
                                .WORD   695
                                .WORD   WRTErr
                                .WORD   PKTSSR
                                TRAP    C$CLP1
                                .WORD   695
                                .WORD   WRTErr
                                .WORD   PKTSSR
7689 065420              75$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   695
                                .WORD   WRTErr
                                .WORD   PKTSSR
7690 065422 005723              TST    (R3)+        ;BUMP THE RECORD SIZE
7691 065424 022703 000414      CMP    #268.,R3    ;MAXIMUM SIZE YET
7692 065430 001401              BEQ    120$        ;BR, IF AT END OF WRITE SEQUENCE
7693 065432 000742              BR     65$         ;WRITE MORE RECORDS
7694 065434              120$:
7695 065434 005037 072056'      CLR    T26CNT       ;SET RECORD COUNTER BACK TO ZERO
7696
7697
7698
7699
7700
7701
7702
7703 065440 004737 010714'      JSR    PC,REWIND    ;CALL TAPE REWIND COMMAND
7704 065444 103411              BCS    130$        ;BR, IF NO PROBLEM
7705 065446 016501 000002      MOV    TSSR(R5),R1 ;GET TSSR
7706 065452 010004              MOV    R0,R4       ;PACKET ADDRESS SET UP
7707 065454 005237 002212'      INC    FATFLG       ;BUMP COUNT
7711 065460              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP    C$ERHRD
                                .WORD   696
                                .WORD   T26RWN
                                .WORD   PKTSSR
7712 065470              130$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   696
                                .WORD   T26RWN
                                .WORD   PKTSSR
7713
7714
7715
7716
7717
7718
7719
7720 065472 013701 071730'      MOV    T26BFR+6,R1 ;PICK UP XSTO
7721 065476 010102              MOV    R1,R2       ;SET UP EXPECTED
7722 065500 052702 000002      BIS    #BIT1,R2    ;SET BOT BIT IN EXPECTED
7723 065504 020102              CMP    R1,R2       ;DOES EXP = REC'D
7724 065506 001406              BEQ    135$        ;BR, IF EQUAL (OK)
7725 065510 005237 002212'      INC    FATFLG       ;BUMP COUNT
7729 065514              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP    C$ERHRD
                                .WORD   697
                                .WORD   T26BOT
                                .WORD   EXPREC
7730 065524              135$:  CKLOOP          ;LOOP IF SELECTED

```

```

065524 104406
7731 065526 012737 000400 072062'      MOV    #256.,T26RSZ      ;START RECORD SIZE
7732 065534 000420                      BR     140$             ;SKIP OVER SPACE
7733
7734      ;*****
7735      ;
7736      ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
7737      ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
7738      ;
7739      ;*****
7740
7741 065536 012703 000001      136$:  MOV    #000001,R3      ;SET UP SPACE COMMAND (1 FORWARD)
7742 065542 004737 010366'      JSR    PC,SPACE        ;CALL SPACE ROUTINE
7743 065546 103413                      BCS    140$            ;BR, IF NO TROUBLE
7744 065550 016501 000002      MOV    TSSR(R5),R1     ;GET TSSR
7745 065554 012702 000200      MOV    #SSR,R2        ;SET UP EXPECTED TSSR
7746 065560 010004                      MOV    R0,R4          ;PACKET ADDRESS SET UP
7747 065562 005237 002212'      INC    FATFLG         ;BUMP COUNT
7751 065566                      ERRHRD ERRNO,T26SC,PKTSSR ;SPACE (FORWARD) FAILED
065566 104456                      TRAP   C$ERHRD
065570 001272                      .WORD 698
065572 072477'                      .WORD T26SC
065574 011746'                      .WORD PKTSSR
7752 065576                      140$:  CKLOOP          ;LOOP IF SELECTED
065576 104406                      TRAP   C$CLP1
7753 065600 013703 072062'      MOV    T26RSZ,R3      ;RECORD SIZE
7754 065604 013737 003114' 072032' 150$:  MOV    FREE,T26RB     ;STARTING READ BUFFER ADDRESS
7755
7756      ;*****
7757      ;
7758      ;REREAD DATA,CVC=1,ACK, OPP COMMAND
7759      ;
7760      ;*****
7761
7762 065612 012737 161401 072030'      MOV    #161401,T26PK3 ;REREAD DATA,CVC=1,ACK, OPP COMMAND
7763 065620 012704 072030'      MOV    #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
7764 065624 010337 072036'      MOV    R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
7765 065630 010465 000000      MOV    R4,TSDB(R5)    ;ISSUE COMMAND
7766 065634 004737 016150'      JSR    PC,WAITF       ;WAIT FOR SSR TO SET
7767 065640 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
7768 065644 012702 000200      MOV    #SSR,R2       ;SET UP EXPECTED
7769 065650 020102                      CMP    R1,R2         ;ARE THEY EQUAL
7770 065652 001406                      BEQ    170$          ;BR, IF OK
7771 065654 005237 002212'      INC    FATFLG         ;BUMP COUNT
7775 065660                      ERRHRD ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
065660 104456                      TRAP   C$ERHRD
065662 001273                      .WORD 699
065664 072305'                      .WORD T26RRF
065666 011746'                      .WORD PKTSSR
7776 065670                      170$:  CKLOOP          ;LOOP IF SELECTED
065670 104406                      TRAP   C$CLP1
7777 065672 017701 115216      MOV    @FREE,R1       ;FIRST WORD FROM READ BUFFER
7778 065676 013702 072056'      MOV    T26CNT,R2     ;SET UP EXPECTED
7779 065702 020102                      CMP    R1,R2         ;IS TAPE POSITION CORRECT
7780 065704 001406                      BEQ    190$          ;KEEP GOING POSITION OK
7781 065706 005237 002212'      INC    FATFLG         ;BUMP COUNT
7785 065712                      ERRHRD ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT

```



```

7834 066030 011734' 002172' 071720' 20$: MOV UNITN,T26DSW ;SET UP UNIT NUMBER .WORD SFIMSG
7835 066032 013737
7836 066040 012704 071700' MOV #T26PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
7837
7838 ;*****
7839 ;
7840 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
7841 ;
7842 ;*****
7843
7844 066044 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
7845 066050 103407 BCS 26$ ;BR, IF COMMAND ISSUED OK
7846 066052 005237 002212' INC FATFLG ;BUMP COUNT
7850 066056 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
7851 066060 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
; TRAP C$ERHRD
; .WORD 702
; .WORD WRTMSG
; .WORD SFIMSG
7852 066070 26$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
7853
7854 ;*****
7855 ;
7856 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7857 ;
7858 ;*****
7859
7860 066072 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
7861 066076 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
7862 066102 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
7863 066106 103407 BCS 30$ ;BR, IF NO PROBLEM
7864 066110 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
7865 066112 005237 002212' INC FATFLG ;BUMP COUNT
7869 066116 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERHRD
; .WORD 703
; .WORD T26RWN
; .WORD PKTSSR
7870 066126 30$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
7871
7872 ;*****
7873 ;
7874 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7875 ;
7876 ;*****
7877
7878 066130 013701 071730' MOV T26BFR+6,R1 ;PICK UP XSTO
7879 066134 010102 MOV R1,R2 ;SET UP EXPECTED
7880 066136 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
7881 066142 020102 CMP R1,R2 ;DOES EXP = REC'D
7882 066144 001406 BEQ 40$ ;BR, IF EQUAL (OK)
7883 066146 005237 002212' INC FATFLG ;BUMP COUNT
7887 066152 ERRHRD ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
; TRAP C$ERHRD
; .WORD 704
; .WORD T26BOT
; .WORD EXPREC
066152 104456

```

```

066154 001300
066156 073075'
066160 015374'
7888 066162 104406 40$: CKLOOP ;LOOP IF SELECTED .WORD 704
066162 104406 ;RECORD SIZE .WORD T26BOT
7889 066164 012703 001000 MOV #512.,R3 ;STARTING WRITE BUFFER ADDRESS .WORD EXPREC
7890 066170 013737 003114' 072032' MOV FREE,T26RB TRAP C$CLP1
7891
7892 ;*****
7893 ;
7894 ;WRITE DATA,CVC=1,ACK COMMAND
7895 ;
7896 ;*****
7897
7898 066176 012737 140005 072030' MOV #140005,T26PK3 ;WRITE DATA,CVC=1,ACK COMMAND
7899 066204 012704 072030' MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
7900 066210
7901 066210 010337 072036' 65$: MOV R3,T26SZ ;SET UP RECORD SIZE IN PACKET
7902 066214 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
7903 066220 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
7904 066224 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
7905 066230 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
7906 066234 020102 CMP R1,R2 ;ARE THEY EQUAL
7907 066236 001406 BEQ 75$ ;BR, IF OK
7908 066240 005237 002212' INC FATFLG ;BUMP COUNT
7912 066244 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
066244 104456 TRAP C$ERHRD
066246 001301 .WORD 705
066250 005103' .WORD WRERR
066252 011746' .WORD PKTSSR
7913 066254 104406 75$: CKLOOP ;LOOP IF SELECTED .WORD C$CLP1
066254 104406 TRAP C$CLP1
7914
7915 ;*****
7916 ;
7917 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7918 ;
7919 ;*****
7920
7921 066256 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
7922 066262 016501 000002 MOV TSSR(R5),R1 ;GET TSSR
7923 066266 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED TSSR
7924 066272 103407 BCS 130$ ;BR, IF NO PROBLEM
7925 066274 010004 MOV R0,R4 ;PACKET ADDRESS SET UP
7926 066276 005237 002212' INC FATFLG ;BUMP COUNT
7930 066302 ERRHRD ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
066302 104456 TRAP C$ERHRD
066304 001302 .WORD 706
066306 073364' .WORD T26RWN
066310 011746' .WORD PKTSSR
7931 066312 104406 130$: CKLOOP ;LOOP IF SELECTED .WORD C$CLP1
066312 104406 TRAP C$CLP1
7932
7933 ;*****
7934 ;
7935 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
7936 ;

```

```

7937
7938
7939 066314 013701 071730'      MOV      T26BFR+6,R1      ;PICK UP XSTO
7940 066320 010102              MOV      R1,R2           ;SET UP EXPECTED
7941 066322 052702 000002      BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
7942 066326 020102              CMP      R1,R2          ;DOES EXP = REC'D
7943 066330 001406              BEQ      140$            ;BR, IF EQUAL (OK)
7944 066332 005237 002212'      INC      FATFLG          ;BUMP COUNT
7948 066336              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    707
                                .WORD    T26BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
066336 104456
066340 001303
066342 073075'
066344 015374'
7949 066346              140$:  CKLOOP            ;LOOP IF SELECTED
                                TRAP      C$CLP1
066346 104406
7950 066350 005303              DEC      R3              ;SET RECORD SIZE TO 511.
7951 066352 013737 003114' 072032'  MOV      FREE,T26RB      ;STARTING READ BUFFER ADDRESS
7952
7953
7954
7955 ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
7956
7957
7958
7959 066360 012737 161401 072030'  165$:  MOV      #161401,T26PK3 ;REREAD DATA,CVC=1,ACK,OPP=1 COMMAND
7960 066366 012704 072030'      MOV      #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
7961 066372 010337 072036'      MOV      R3,T26SZ       ;SET UP RECORD SIZE IN PACKET
7962 066376 010465 000000      MOV      R4,TSDB(R5)    ;ISSUE COMMAND
7963 066402 004737 016150'      JSR      PC,WAITF       ;WAIT FOR SSR TO SET
7964 066406 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
7965 066412 012702 100204      MOV      #SSR!SC!BIT2,R2 ;SET UP EXPECTED
7966 066416 020102              CMP      R1,R2          ;ARE THEY EQUAL
7967 066420 001406              BEQ      170$            ;BR, IF OK
7968 066422 005237 002212'      INC      FATFLG          ;BUMP COUNT
7972 066426              ERRHRD  ERRNO,T26TRL,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
                                TRAP      C$ERHRD
                                .WORD    708
                                .WORD    T26TRL
                                .WORD    PKTSSR
                                TRAP      C$CLP1
066426 104456
066430 001304
066432 074442'
066434 011746'
7973 066436              170$:  CKLOOP            ;LOOP IF SELECTED
066436 104406
                                TRAP      C$CLP1
7974
7975
7976
7977 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
7978
7979
7980
7981 066440 013701 071730'      MOV      T26BFR+6,R1    ;GET MESSAGE BUFFER
7982 066444 010102              MOV      R1,R2           ;SET UP EXPECTED
7983 066446 052702 010000      BIS      #BIT12,R2      ;SET THE RLL BIT IN EXPECTED
7984 066452 020102              CMP      R1,R2          ;ARE THEY EQUAL
7985 066454 001406              BEQ      180$            ;BR, IF EQUAL (ALL IS WELL)
7986 066456 005237 002212'      INC      FATFLG          ;BUMP COUNT
7990 066462              ERRHRD  ERRNO,T26LON,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
                                TRAP      C$ERHRD
                                .WORD    709
                                .WORD

```



```

      066734 005046'
      066736 011734'
8094 066740 104406      26$:  CKLOOP      ;LOOP IF SELECTED      .WORD  WRTMSG
      066740 104406      TRAP      C$CLP1      .WORD  SFIMSG
8095
8096
8097      ;*****
8098      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8099      ;
8100      ;*****
8101
8102 066742 004737 010714'      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
8103 066746 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR
8104 066752 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED TSSR
8105 066756 103407      BCS      30$      ;BR, IF NO PROBLEM
8106 066760 010004      MOV      R0,R4      ;PACKET ADDRESS SET UP
8107 066762 005237 002212'      INC      FATFLG      ;BUMP COUNT
8111 066766      ERRHRD  ERRNO,T26RWN,PKTSSR      ;REWIND NOT ACCEPTED
      066766 104456      TRAP      C$ERHRD
      066770 001312      .WORD  714
      066772 073364'      .WORD  T26RWN
      066774 011746'      .WORD  PKTSSR
8112 066776 104406      30$:  CKLOOP      ;LOOP IF SELECTED      TRAP  C$CLP1
      066776 104406      TRAP
8113
8114      ;*****
8115      ;
8116      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8117      ;
8118      ;*****
8119
8120 067000 013701 071730'      MOV      T26BFR+6,R1      ;PICK UP XSTO
8121 067004 010102      MOV      R1,R2      ;SET UP EXPECTED
8122 067006 052702 000002      BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
8123 067012 020102      CMP      R1,R2      ;DOES EXP = REC'D
8124 067014 001406      BEQ      40$      ;BR, IF EQUAL (OK)
8125 067016 005237 002212'      INC      FATFLG      ;BUMP COUNT
8129 067022      ERRHRD  ERRNO,T26BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      067022 104456      TRAP      C$ERHRD
      067024 001313      .WORD  715
      067026 073075'      .WORD  T26BOT
      067030 015374'      .WORD  EXPREC
8130 067032 104406      40$:  CKLOOP      ;LOOP IF SELECTED      TRAP  C$CLP1
      067032 104406      TRAP
8131 067034 012703 000400      MOV      #256.,R3      ;RECORD SIZE
8132 067040 013737 003114' 072032'      MOV      FREE,T26RB      ;STARTING WRITE BUFFER ADDRESS
8133
8134      ;*****
8135      ;
8136      ;WRITE DATA,CVC=1,ACK COMMAND
8137      ;
8138      ;*****
8139
8140 067046 012737 140005 072030'      MOV      #140005,T26PK3      ;WRITE DATA,CVC=1,ACK COMMAND
8141 067054 012704 072030'      MOV      #T26PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
8142 067060
8143 067060 010337 072036'      65$:  MOV      R3,T26SZ      ;SET UP RECORD SIZE IN PACKET

```

```

8144 067064 010465 000000      MOV      R4,TSDB(R5)          ;ISSUE COMMAND
8145 067070 004737 016150'     JSR      PC,WAITF            ;WAIT FOR SSR TO SET
8146 067074 016501 000002      MOV      TSSR(R5),R1        ;GET TSSR CONTENTS
8147 067100 012702 000200      MOV      #SSR,R2            ;SET UP EXPECTED
8148 067104 020102              CMP      R1,R2               ;ARE THEY EQUAL
8149 067106 001406              BEQ      75$                 ;BR, IF OK
8150 067110 005237 002212'     INC      FATFLG              ;BUMP COUNT
8154 067114              ERRHRD  ERRNO,WRTErr,PKTSSR  ;TSSR INCORRECT AFTER WRITE DATA
      067114 104456              TRAP    C$ERHRD
      067116 001314              .WORD  716
      067120 005103'            .WORD  WRTErr
      067122 011746'            .WORD  PKTSSR
8155 067124              75$:  CKLOOP                ;LOOP IF SELECTED
      067124 104406              TRAP    C$CLP1
8156 067126              120$:
8157
8158 ;*****
8159 ;
8160 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8161 ;
8162 ;*****
8163
8164 067126 004737 010714'     JSR      PC,REWIND           ;CALL TAPE REWIND COMMAND
8165 067132 016501 000002      MOV      TSSR(R5),R1        ;GET TSSR
8166 067136 012702 000200      MOV      #SSR,R2            ;SET UP EXPECTED TSSR
8167 067142 103407              BCS     130$                 ;BR, IF NO PROBLEM
8168 067144 010004              MOV      R0,R4               ;PACKET ADDRESS SET UP
8169 067146 005237 002212'     INC      FATFLG              ;BUMP COUNT
8173 067152              ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      067152 104456              TRAP    C$ERHRD
      067154 001315              .WORD  717
      067156 073364'            .WORD  T26RWN
      067160 011746'            .WORD  PKTSSR
8174 067162              130$:  CKLOOP                ;LOOP IF SELECTED
      067162 104406              TRAP    C$CLP1
8175
8176 ;*****
8177 ;
8178 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8179 ;
8180 ;*****
8181
8182 067164 013701 071730'     MOV      T26BFR+6,R1        ;PICK UP XSTO
8183 067170 010102              MOV      R1,R2               ;SET UP EXPECTED
8184 067172 052702 000002      BIS      #BIT1,R2            ;SET BOT BIT IN EXPECTED
8185 067176 020102              CMP      R1,R2               ;DOES EXP = REC'D
8186 067200 001406              BEQ     135$                 ;BR, IF EQUAL (OK)
8187 067202 005237 002212'     INC      FATFLG              ;BUMP COUNT
8191 067206              ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      067206 104456              TRAP    C$ERHRD
      067210 001316              .WORD  718
      067212 073075'            .WORD  T26BOT
      067214 015374'            .WORD  EXPREC
8192 067216              135$:  CKLOOP                ;LOOP IF SELECTED
      067216 104406              TRAP    C$CLP1
8193 067220 012703 001000      MOV      #512.,R3           ;RECORD SIZE
8194 067224 013737 003114' 072032'  MOV      FREE,T26RB         ;STARTING READ BUFFER ADDRESS

```



```

8195
8196
8197
8198
8199
8200
8201
8202 067232 012737 161401 072030'
8203 067240 012704 072030'
8204 067244 010337 072036'
8205 067250 010465 000000
8206 067254 004737 016150'
8207 067260 016501 000002
8208 067264 012702 100204
8209 067270 020102
8210 067272 001406
8211 067274 005237 002212'
8215 067300
      067300 104456
      067302 001317
      067304 074442'
      067306 011746'
8216 067310
      067310 104406
8217
8218
8219
8220
8221
8222
8223
8224 067312 013701 071730'
8225 067316 010102
8226 067320 052702 040000
8227 067324 020102
8228 067326 001406
8229 067330 005237 002212'
8233 067334
      067334 104456
      067336 001320
      067340 074272'
      067342 015374'
8234 067344
      067344 104406
8235 067346 013701 071726'
8236 067352 012702 000400
8237 067356 020102
8238 067360 001405
8242 067364
      067364 104456
      067366 001320
      067370 074354'
      067372 015374'
8243 067374
      067374 104406
8244 067376 012703 001000
8245 067402 013737 003114' 072032'

```

```

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

```



```

8352 067700          ERRHRD  ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTIC FAILED
      067700 104456                                     TRAP      C$ERHRD
      067702 001324                                     .WORD    724
      067704 005046'                                     .WORD    WRTMSG
      067706 011734'                                     .WORD    SFIMSG
8353 067710          26$:  CKLOOP                      ;LOOP IF SELECTED
      067710 104406                                     TRAP      C$CLP1
8354
8355 ;*****
8356 ;
8357 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8358 ;
8359 ;*****
8360
8361 067712 004737 021024'      JSR      PC,INVERT          ;INVERT THE EXTENDED FEATURES SWITCH
8362 067716 004737 010714'      JSR      PC,REWIND         ;CALL TAPE REWIND COMMAND
8363 067722 103411              BCS      30$               ;BR, IF NO PROBLEM
8364 067724 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR
8365 067730 010004              MOV      R0,R4             ;PACKET ADDRESS SET UP
8366 067732 005237 002212'      INC      FATFLG           ;BUMP COUNT
8370 067736          ERRHRD  ERRNO,T26RWN,PKTSSR      ;REWIND NOT ACCEPTED
      067736 104456                                     TRAP      C$ERHRD
      067740 001325                                     .WORD    725
      067742 073364'                                     .WORD    T26RWN
      067744 011746'                                     .WORD    PKTSSR
8371 067746          30$:  CKLOOP                      ;LOOP IF SELECTED
      067746 104406                                     TRAP      C$CLP1
8372
8373 ;*****
8374 ;
8375 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8376 ;
8377 ;*****
8378
8379 067750 013701 071730'      MOV      T26BFR+6,R1      ;PICK UP XSTO
8380 067754 010102              MOV      R1,R2            ;SET UP EXPECTED
8381 067756 052702 000002      BIS      @BIT1,R2         ;SET BOT BIT IN EXPECTED
8382 067762 020102              CMP      R1,R2            ;DOES EXP = REC'D
8383 067764 001406              BEQ      40$              ;BR, IF EQUAL (OK)
8384 067766 005237 002212'      INC      FATFLG           ;BUMP COUNT
8388 067772          ERRHRD  ERRNO,T26BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      067772 104456                                     TRAP      C$ERHRD
      067774 001326                                     .WORD    726
      067776 073075'                                     .WORD    T26BOT
      070000 015374'                                     .WORD    EXPREC
8389 070002          40$:  CKLOOP                      ;LOOP IF SELECTED
      070002 104406                                     TRAP      C$CLP1
8390 070004 013737 003114' 072032'  MOV      FREE,T26RB        ;STARTING WRITE BUFFER ADDRESS
8391
8392 ;*****
8393 ;
8394 ;WRITE DATA,CVC=1,ACK COMMAND
8395 ;
8396 ;*****
8397
8398 070012 012737 140005 072030'  MOV      @140005,T26PK3    ;WRITE DATA,CVC=1,ACK COMMAND
8399 070020 012704 072030'  MOV      @T26PK3,R4        ;SET UP R4 WITH PACKET ADDRESS

```

```

8400 070024 012737 000400 072036' 65$:  MOV    #256.,T26SZ      ;SET UP RECORD SIZE IN PACKET
8401 070032 013777 072056' 113054  MOV    T26CNT,0FREE    ;MOVE TAPE RECORD NUMBER TO BUFFER
8402 070040 062737 000001 072056'  ADD    #1,T26CNT      ;NUMBER READY FOR NEXT RECORD
8403 070046 010465 000000      MOV    R4,TSDB(R5)    ;ISSUE COMMAND
8404 070052 004737 016150'  JSR    PC,WAITF      ;WAIT FOR SSR TO SET
8405 070056 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR CONTENTS
8406 070062 012702 000200      MOV    #SSR,R2       ;SET UP EXPECTED
8407 070066 020102      CMP    R1,R2         ;ARE THEY EQUAL
8408 070070 001406      BEQ    75$           ;BR, IF OK
8409 070072 005237 002212'  INC    FATFLG        ;BUMP COUNT
8413 070076      ERRHRD  ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      070076 104456      TRAP    C$ERHRD
      070100 001327      .WORD  727
      070102 005103'    .WORD  WRTERR
      070104 011746'    .WORD  PKTSSR
8414 070106      75$:  CKLOOP      ;LOOP IF SELECTED
      070106 104406      TRAP    C$CLP1
8415 070110 022737 000013 072056'  CMP    #11.,T26CNT   ;CHECK NUMBER OF RECORDS WRITTEN
8416 070116 001401      BEQ    120$         ;BR, IF AT END OF WRITE SEQUENCE
8417 070120 000741      BR     65$          ;WRITE MORE RECORDS
8418 070122      120$:
8419 070122 005037 003132'  CLR    NXMMI         ;SET TO 16 BIT ADDRESS
8420 070126      125$:
8421 070126 005037 072056'  CLR    T26CNT        ;SET RECORD COUNTER BACK TO ZERO
8422
8423 ;*****
8424 ;
8425 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8426 ;
8427 ;*****
8428
8429 070132 004737 010714'  JSR    PC,REWIND     ;CALL TAPE REWIND COMMAND
8430 070136 103411      BCS    130$         ;BR, IF NO PROBLEM
8431 070140 016501 000002      MOV    TSSR(R5),R1   ;GET TSSR
8432 070144 010004      MOV    R0,R4         ;PACKET ADDRESS SET UP
8433 070146 005237 002212'  INC    FATFLG        ;BUMP COUNT
8437 070152      ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      070152 104456      TRAP    C$ERHRD
      070154 001330      .WORD  728
      070156 073364'    .WORD  T26RWN
      070160 011746'    .WORD  PKTSSR
8438 070162      130$:  CKLOOP      ;LOOP IF SELECTED
      070162 104406      TRAP    C$CLP1
8439
8440 ;*****
8441 ;
8442 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8443 ;
8444 ;*****
8445
8446 070164 013701 071730'  MOV    T26BFR+6,R1   ;PICK UP XSTO
8447 070170 010102      MOV    R1,R2         ;SET UP EXPECTED
8448 070172 052702 000002'  BIS    #BIT1,R2      ;SET BOT BIT IN EXPECTED
8449 070176 020102      CMP    R1,R2         ;DOES EXP = REC'D
8450 070200 001406      BEQ    140$         ;BR, IF EQUAL (OK)
8451 070202 005237 002212'  INC    FATFLG        ;BUMP COUNT
8455 070206      ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND

```

```

070206 104456
070210 001331
070212 073075'
070214 015374'
8456 070216 140$: CKLOOP ;LOOP IF SELECTED TRAP C$ERHRD
070216 104406 ;COMMAND BUFFER ADDRESS TRAP C$CLP1
8457 070220 012703 072046'
8458 070224 013737 003130' 072032' 150$: MOV #T26RN,R3 ;STARTING READ BUFFER ADDRESS
8459 070232 013737 003132' 072034' MOV NXML0,T26RB ;SET UP HIGH ORDER ADDRESS BITS
MOV NXMHI,T26RB+2
8460
8461 ;*****
8462 ;
8463 ;REREAD DATA,IE,ACK, OPP COMMAND
8464 ;
8465 ;*****
8466
8467 070240 011337 072030' MOV (R3),T26PK3 ;REREAD DATA,IE,ACK, OPP COMMAND
8468 070244 012704 072030' 165$: MOV #T26PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
8469 070250 012737 000400 072036' MOV #256.,T26SZ ;SET UP RECORD SIZE IN PACKET
8470 070256 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
8471 070262 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
8472 070266 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
8473 070272 012702 104210 MOV #SSR!NXM!SC!BIT3,R2 ;SET UP EXPECTED
8474 070276 020102 CMP R1,R2 ;ARE THEY EQUAL
8475 070300 001422 BEQ 170$ ;BR, IF OK
8476 070302 031327 001000 BIT (R3),#BIT9 ;CHECK FOR A READ COMMAND
8477 070306 001403 BEQ 168$ ;BR, IF IT WAS A READ COMMAND
8478 070310 030127 000002 BIT R1,#BIT1 ;WAS BIT1 SET
8479 070314 001014 BNE 170$ ;BR, IF REREAD AND BIT1 SET
8480 070316
8481 070316 005237 003132' 168$: INC NXMHI ;BUMP TO NEXT ADDRESS RANGE
8482 070322 023727 003132' 000004 CMP NXMHI,#4 ;CHECK FOR OVERFLOW
8483 070330 001276 BNE 125$ ;BR, IF MORE BITS TO GO
8484 070332 005237 002212' INC FATFLG ;BUMP COUNT
8488 070336 ERRHRD ERRNO,T26RRF,PKTSSR ;TSSR INCORRECT AFTER REREAD DATA
070336 104456 TRAP C$ERHRD
070340 001332 .WORD 730
070342 072305' .WORD T26RRF
070344 011746' .WORD PKTSSR
8489 070346 170$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
070346 104406
8490
8491 ;*****
8492 ;
8493 ;READ DATA, ACK,CVC=1 COMMAND
8494 ;
8495 ;*****
8496
8497 070350 012737 140001 072030' MOV #140001,T26PK3 ;READ DATA, ACK,CVC=1 COMMAND
8498 070356 012737 000400 072036' MOV #256.,T26SZ ;SET SIZE INTO PACKET
8499 070364 005037 072034' CLR T26RB+2 ;CLEAR OUT HIGH ADDRESS BITS
8500 070370 013737 003114' 072032' MOV FREE,T26RB ;GIVE READ A GOOD BUFFER
8501 070376 010465 000000 MOV R4,TSDB(R5) ;ISSUE READ DATA COMMAND
8502 070402 004737 016150' JSR PC,WAITF ;WAIT FOR SSR
8503 070406 016501 000002 MOV TSSR(R5),R1 ;PICK UP THE TSSR
8504 070412 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
8505 070416 020102 CMP R1,R2 ;IS THE TSSR OK

```

```

8506 070420 001406          BEQ      180$          ;BR, IF TSSR OK (GOOD)
8507 070422 005237 002212'  INC      FATFLG        ;BUMP COUNT
8511 070426          ERRHRD  ERRNO,RDERR,PKTSSR ;READ DATA COMMAND FAILED
      070426 104456          TRAP      C$ERHRD
      070430 001333          .WORD    731
      070432 005176'        .WORD    RDERR
      070434 011746'        .WORD    PKTSSR
8512 070436          180$:  CKLOOP          ;LOOP IF SELECTED
      070436 104406          TRAP      C$CLP1
8513 070440 017701 112450  MOV      @FREE,R1      ;FIRST WORD FROM READ BUFFER
8514 070444 012702 000001  MOV      @1,R2         ;SET UP EXPECTED
8515 070450 020102          CMP      R1,R2         ;IS TAPE POSITION CORRECT
8516 070452 001406          BEQ      190$          ;KEEP GOING POSITION OK
8517 070454 005237 002212'  INC      FATFLG        ;BUMP COUNT
8521 070460          ERRHRD  ERRNO,T26WNG,EXPREC ;TAPE POSITION INCORRECT
      070460 104456          TRAP      C$ERHRD
      070462 001334          .WORD    732
      070464 072066'        .WORD    T26WNG
      070466 015374'        .WORD    EXPREC
8522 070470          190$:  CKLOOP
      070470 104406          TRAP      C$CLP1
8523
8524          ;*****
8525          ;
8526          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8527          ;
8528          ;*****
8529
8530 070472 004737 010714'  JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
8531 070476 103411          BCS      194$          ;BR, IF NO PROBLEM
8532 070500 016501 000002  MOV      TSSR(R5),R1  ;GET TSSR
8533 070504 010004          MOV      R0,R4         ;PACKET ADDRESS SET UP
8534 070506 005237 002212'  INC      FATFLG        ;BUMP COUNT
8538 070512          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      070512 104456          TRAP      C$ERHRD
      070514 001335          .WORD    733
      070516 073364'        .WORD    T26RWN
      070520 011746'        .WORD    PKTSSR
8539 070522          194$:  CKLOOP          ;LOOP IF SELECTED
      070522 104406          TRAP      C$CLP1
8540
8541          ;*****
8542          ;
8543          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8544          ;
8545          ;*****
8546
8547 070524 013701 071730'  MOV      T26BFR+6,R1  ;PICK UP XSTO
8548 070530 010102          MOV      R1,R2         ;SET UP EXPECTED
8549 070532 052702 000002  BIS      @BIT1,R2     ;SET BOT BIT IN EXPECTED
8550 070536 020102          CMP      R1,R2         ;DOES EXP = REC'D
8551 070540 001406          BEQ      196$          ;BR, IF EQUAL (OK)
8552 070542 005237 002212'  INC      FATFLG        ;BUMP COUNT
8556 070546          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      070546 104456          TRAP      C$ERHRD
      070550 001336          .WORD    734
      070552 073075'        .WORD    T26BOT

```



```

8611 070654 010001          MOV    R0,R1          ;CONTENTS OF TSSR REGISTER
8612 070656          ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      070656 104455          TRAP    C$ERDF
      070660 001337          .WORD  735
      070662 003642'        .WORD  SFIERR
      070664 011734'        .WORD  SFIMSG
8613 070666 013737 002172' 071720' 20$:  MOV    UNITN,T26DSW      ;SET UP UNIT NUMBER
8614
8615 070674 012704 071700'  MOV    @T26PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
8616
8617          ;*****
8618          ;
8619          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
8620          ;
8621          ;*****
8622
8623 070700 004737 010562'    JSR    PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
8624 070704 103407          BCS    25$              ;BR, IF COMMAND ISSUED OK
8625 070706 005237 002212'  INC    FATFLG          ;BUMP COUNT
8629 070712 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
8630 070714          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      070714 104456          TRAP    C$ERHRD
      070716 001340          .WORD  736
      070720 005046'        .WORD  WRTMSG
      070722 011734'        .WORD  SFIMSG
8631 070724          25$:  CKLOOP          ;LOOP IF SELECTED
      070724 104406          TRAP    C$CLP1
8632
8633          ;*****
8634          ;
8635          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
8636          ;
8637          ;*****
8638
8639 070726 004737 010714'  26$:  JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
8640 070732 016501 000002'  MOV    TSSR(R5),R1    ;GET TSSR
8641 070736 012702 000200'  MOV    @SSR,R2        ;SET UP EXPECTED TSSR
8642 070742 103407          BCS    30$              ;BR, IF NO PROBLEM
8643 070744 010004          MOV    R0,R4          ;PACKET ADDRESS SET UP
8644 070746 005237 002212'  INC    FATFLG          ;BUMP COUNT
8648 070752          ERRHRD  ERRNO,T26RWN,PKTSSR ;REWIND NOT ACCEPTED
      070752 104456          TRAP    C$ERHRD
      070754 001341          .WORD  737
      070756 073364'        .WORD  T26RWN
      070760 011746'        .WORD  PKTSSR
8649 070762          30$:  CKLOOP          ;LOOP IF SELECTED
      070762 104406          TRAP    C$CLP1
8650
8651          ;*****
8652          ;
8653          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
8654          ;
8655          ;*****
8656
8657 070764 013701 071730'    MOV    T26BFR+6,R1    ;PICK UP XST0
8658 070770 010102          MOV    R1,R2          ;SET UP EXPECTED
8659 070772 052702 000002'  BIS    @BIT1,R2        ;SET BOT BIT IN EXPECTED

```

```

8660 070776 020102          CMP      R1,R2          ;DOES EXP = REC'D
8661 071000 001406          BEQ      40$           ;BR, IF EQUAL (OK)
8662 071002 005237 002212'  INC      FATFLG        ;BUMP COUNT
8666 071006          ERRHRD  ERRNO,T26BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      071006 104456          TRAP      C$ERHRD
      071010 001342          .WORD    738
      071012 073075'        .WORD    T26BOT
      071014 015374'        .WORD    EXPREC
8667 071016          40$:   CKLOOP          ;LOOP IF SELECTED
      071016 104406          TRAP      C$CLP1
8668 071020 012737 000400 072036'  MOV      #256.,T26SZ    ;SET UP RECORD SIZE IN PACKET
8669 071026 013737 003114' 072032'  MOV      FREE,T26RB     ;ADDRESS OF READ BUFFER
8670 071034 005703          TST      R3            ;CHECK NUMBER OF TIMES THROUGH HERE
8671 071036 001404          BEQ      50$           ;BR, IF FIRST TIME THROUGH HERE
8672
8673
8674          ;*****
8675          ;REREAD,CVC=1,ACK COMMAND
8676          ;
8677          ;*****
8678
8679 071040 012737 161001 072030'  MOV      #161001,T26PK3 ;REREAD,CVC=1,ACK COMMAND
8680 071046 000403          BR       55$           ;SKIP NEXT COMMAND
8681
8682          ;*****
8683          ;REREAD,ACK COMMAND
8684          ;
8685          ;*****
8686
8687
8688 071050 012737 141001 072030'  50$:   MOV      #141001,T26PK3 ;REREAD,ACK COMMAND
8689 071056          55$:
8690 071056 012704 072030'        MOV      #T26PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
8691 071062          65$:
8692 071062 010465 000000        MOV      R4,TSDB(R5)    ;ISSUE COMMAND
8693 071066 004737 016150'        JSR      PC,WAITF      ;WAIT FOR SSR TO SET
8694 071072 016501 000002        MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
8695 071076 012702 100206        MOV      #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
8696 071102 020102          CMP      R1,R2         ;ARE THEY EQUAL
8697 071104 001406          BEQ      75$           ;BR, IF OK
8698 071106 005237 002212'  INC      FATFLG        ;BUMP COUNT
8702 071112          ERRHRD  ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      071112 104456          TRAP      C$ERHRD
      071114 001343          .WORD    739
      071116 073023'        .WORD    T26WDE
      071120 011746'        .WORD    PKTSSR
8703 071122          75$:   CKLOOP          ;LOOP IF SELECTED
      071122 104406          TRAP      C$CLP1
8704
8705          ;*****
8706          ;
8707          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
8708          ;
8709          ;*****
8710
8711 071124 013701 071730'        MOV      T26BFR+6,R1   ;GET XSTO STATUS WORD
8712 071130 010102          MOV      R1,R2         ;SET UP EXPECTED

```



```

      071372 001350
      071374 073075'
      071376 015374'
8818 071400 40$: CKLOOP
      071400 104406
8819
8820
8821
8822
8823
8824
8825
8826
8827 071402 012703 000001
8828 071406 004737 010366'
8829 071412 103411
8830 071414 016501 000002
8831 071420 010004
8832 071422 005237 002212'
8836 071426
      071426 104456
      071430 001351
      071432 073023'
      071434 011746'
8837 071436 75$: CKLOOP
      071436 104406
8838
8839
8840
8841
8842
8843
8844
8845
8846 071440 012703 100001
8847 071444 004737 010366'
8848 071450 103411
8849 071452 016501 000002
8850 071456 010004
8851 071460 005237 002212'
8855 071464
      071464 104456
      071466 001352
      071470 073023'
      071472 011746'
8856 071474 175$: CKLOOP
      071474 104406
8857 071476 013737 003114' 072032'
8858 071504 005737 072060'
8859 071510 001404
8860
8861
8862
8863
8864
8865
8866

```

```

;*****
;
;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
;
;*****
      MOV      #000001,R3          ;SET UP SPACE FORWARD 1 RECORD
      JSR      PC,SPACE          ;ISSUE SPACE COMMAND
      BCS      75$              ;BR, IF OK
      MOV      TSSR(R5),R1       ;GET STATUS DATA
      MOV      R0,R4            ;GET PACKET ADDRESS
      INC      FATFLG           ;BUMP COUNT
      ERRHRD  ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      TRAP    C$ERHRD
      .WORD   745
      .WORD   T26WDE
      .WORD   PKTSSR
;LOOP IF SELECTED
      TRAP    C$CLP1
;*****
;
;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
;
;*****
      MOV      #100001,R3       ;SET SPACE REVERSE 1 RECORD
      JSR      PC,SPACE          ;ISSUE COMMAND
      BCS      175$             ;GO ON IF ALL IS WELL
      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
      MOV      R0,R4            ;SET UP EXPECTED (PACKET CONTENTS)
      INC      FATFLG           ;BUMP COUNT
      ERRHRD  ERRNO,T26WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      TRAP    C$ERHRD
      .WORD   746
      .WORD   T26WDE
      .WORD   PKTSSR
;LOOP IF SELECTED
      TRAP    C$CLP1
      MOV      FREE,T26RB        ;ADDRESS OF BUFFER
      TST      T26CNU           ;CHECK FOR TIMES THROUGH HERE
      BEQ     176$              ;BR, IF FIRST TIME THROUGH
;*****
;
;REREAD (PREVIOUS),IE,ACK,OPP=1 CMD.
;
;*****

```



```

071672 003112                                .WORD  L10102-.
8917
8918
8919      ; LOCAL STORAGE FOR THIS TEST
8920      ;
8922 071674      .BLKB  10-<.-TSV2&7>
8924 071700      T26PACKET:
8925 071700 014004      .WORD  14004      ;COMMAND PACKET FOR TEST
8926 071702 071710'    .WORD  T26DATA    ;WRITE CHARACTERISTICS COMMAND, WITH CVC=1, ACK
8927 071704 000000    .WORD  0          ;ADDRESS OF CHARACTERISTICS BLOCK
8928 071706 000012    .WORD  10.        ;STARTING VALUE OF BLOCK SIZE
8929 071710      T26DATA:
8930 071710 071722'    .WORD  T26BFR    ;CHARACTERISTICS DATA BLOCK
8931 071712 000000    .WORD  0          ;ADDRESS OF MESSAGE BUFFER
8932 071714 000024    .WORD  20.        ;LENGTH OF MESSAGE BUFFER
8933 071716 000000    .WORD  0
8934 071720 000000    T26DSW: .WORD  0      ;SELECT DRIVE 0
8935 071722      T26BFR: .BLKW  25.    ;MESSAGE BUFFER
8936
8937      ;
8938      ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
8939      ;
8940 072004      .BLKB  10-<.-TSV2&7>
8942 072010      T26PK2:
8943 072010 100006    .WORD  100006    ;WRITE SUB SYS MEM COMMAND, AND ACK
8944 072012 072040'    .WORD  T26BF2    ;ADDRESS OF SELECT BLOCK DATA
8945 072014 000000    .WORD  0
8946 072016 000006    .WORD  6.        ;SIZE OF DATA PACKET
8947
8949 072020      .BLKB  10-<.-TSV2&7>
8951 072030      T26PK3:
8952 072030 140005    .WORD  140005    ;REREAD COMMAND, CVC=1 AND ACK
8953 072032      T26RB:
8954 072032 003114'    T26WB: .WORD  FREE  ;ADDRESS OF WRITE BUFFER
8955 072034 000000    .WORD  0
8956 072036 000000    T26SZ: .WORD  0      ;SIZE OF BUFFER (EXTENT)
8957      .EVEN
8958
8959      ;
8960      ;
8961 072040      T26BF2:
8962 072040 010      T26BS0: .BYTE  10    ;BSELO AREA
8963 072041 200      T26BS1: .BYTE  200   ;BSEL1 AREA
8964 072042 000000    T26S2: .WORD  0      ;SEL 2 AREA
8965 072044 000000    T26S3: .WORD  0      ;DATA AREA
8966
8967      ;
8968      ;
8969      .EVEN
8970      ;TAPE MOTION PACKET COMMAND VALUES
8971 072046 140001    T26RN: .WORD  140001 ;READ DATA
8972 072050 141401    .WORD  141401    ;REREAD NEXT OPP=0
8973 072052 161401    .WORD  161401    ;REREAD NEXT OPP=1
8974 072054 177777    .WORD  177777    ;END OF DATA
8975
8976      ;
8977 072056 000000    T26CNT: .WORD  0    ;TAPE RECORD COUNTER STORAGE AREA
8978 072060 000000    T26CNU: .WORD  0    ;TAPE RECORD COUNTER STORAGE AREA

```

```

8979
8980 072062 000000          T26RSZ: .WORD 0          ;RECORD STORAGE SIZE AREA
8981
8982 072064 000000          T26DLY: .WORD 0          ;DELAY COUNTER AREA
8983
8984
8985          ;*
8986          ;LOCAL TEXT MESSAGES FOR TEST
8987          ;-
8988
8989
8990 072066          124      141      160  T26WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
8991 072154          122      105      122  T26NEF: .ASCIZ 'REREAD PREVIOUS, At BOT, Failed To Set NEF (XSTO)'
8992 072236          124      123      123  T26RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
8993 072305          122      105      122  T26RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
8994 072402          122      105      122  T26RRG: .ASCIZ 'REREAD Previous (Read Reverse, Space Forward) Command Failed'
8995 072477          120      117      123  T26SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
8996 072561          122      111      102  T26LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
8997 072631          124      123      123  T26WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
8998 072706          111      154      154  T26LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
8999 072767          122      105      122  T26SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
9000 073023          124      123      123  T26WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
9001 073075          124      141      160  T26BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
9002 073142          104      141      164  T26DTA: .ASCIZ 'Data Written To Tape Not Equal To Data Read From Tape'
9003 073230          122      105      122  T26EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
9004 073307          124      123      123  T26TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
9005 073364          122      145      167  T26RMN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
9006 073433          122      101      115  T26RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
9007 073506          124      123      123  T26AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
9008 073555          104      162      151  T26OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
9009 073630          124      123      123  T26WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
9010 073720          124      123      123  T26WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
9011 073773          103      126      103  T26VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
9012 074046          124      123      102  T268A: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
9013 074121          127      122      111  T26WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
9014 074210          122      145      141  T26LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XSTO'
9015 074272          122      145      141  T26LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XSTO'
9016 074354          122      145      163  T26PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
9017 074442          122      145      141  T26TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
9018 074530          104      141      164  T26NEQ: .ASCIZ 'Data REREAD From Tape Not Correct, After SWB=1'
9019 074607          122      145      162  TST26ID: .ASCIZ 'Rereads'
9020
9021          .EVEN
9022          ;*
9023          ;
9024          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
9025          ;WRITE SUBSYSTEM MEMORY COMMAND
9026          ;
9027          ;-
9028 074620
9029 074620
9030 074624          012701 071700'
9031 074630          012721 140004
9032 074634          012721 071710'
9033 074640          005021
9034 074642          012721 000012
9035 074646          012721 071722'

          T26REST:
          SAVREG          ;SAVE THE REGISTERS
          MOV             #T26PACKET,R1      ;START OF THE PACKET
          MOV             #140004,(R1)      ;WRITE SUBSYSTEM MEM. WITH ACK, CVC=1
          MOV             #T26DATA,(R1)    ;ADDRESS OF CHARAISTICS DATA BLOCK
          CLR             (R1)              ;EXTENDED ADDRESS
          MOV             #10,(R1)         ;SIZE OF DATA BLOCK IN BYTES
          MOV             #T26BFR,(R1)     ;ADDRESS OF MESSAGE BUFFER

```



```

9036 074652 005021          CLR      (R1)+
9037 074654 012721 000024    MOV      #20.,(R1)+      ;LENGTH OF MESSAGE BUFFER
9038 074660 005021          CLR      (R1)+
9039 074662 012711 000000    MOV      #0,(R1)        ;SELECT DRIVE ZERO (0)
9040 074666 012702 000030    MOV      #24.,R2       ;NUMBER OF LOCATIONS TO BE CLEARED
9041 074672 012762 177777 071722' 64$: MOV      #177777,T26BFR(R2) ;ALL ONES TO MESSAGE BUFFER
9042 074700 005742          TST      -(R2)         ;NEXT LOCATION
9043 074702 020227 000000    CMP      R2,#0         ;CHECK FOR END OF LOOP
9044 074706 001371          BNE      64$          ;KEEP GOING UNTIL DONE
9045 074710 000207          RTS      PC           ;RETURN
9046
9047
9048 074712          T26RT2:
9049 074712          SAVREG          ;SAVE THE REGISTERS
9050 074716 012701 072010'    MOV      #T26PK2,R1    ;START OF THE PACKET
9051 074722 012721 140006    MOV      #140006,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1.
9052 074726 012721 072040'    MOV      #T26BF2,(R1)+ ;ADDRESS OF DATA BLOCK
9053 074732 005021          CLR      (R1)+        ;EXTENDED ADDRESS
9054 074734 012721 000006    MOV      #6.,(R1)+    ;SIZE OF DATA BLOCK IN BYTES
9055 074740 005021          CLR      (R1)+
9056 074742 012701 072040'    MOV      #T26BF2,R1    ;POINT TO DATA SEL AREA
9057 074746 005021          CLR      (R1)+
9058 074750 005011          CLR      (R1)
9059 074752 000207          RTS      PC           ;RETURN
9060 074754          T26RT3:
9061 074754          SAVREG          ;SAVE THE REGISTERS
9062 074760 012701 072030'    MOV      #T26PK3,R1    ;START OF THE PACKET
9063 074764 012721 000000    MOV      #0,(R1)+    ;WRITE SUBSYSTEM MEM. WITH ACK.
9064 074770 012721 000000    MOV      #0,(R1)+    ;ADDRESS OF DATA BLOCK
9065 074774 005021          CLR      (R1)+        ;EXTENDED ADDRESS
9066 074776 012711 000000    MOV      #0,(R1)      ;SIZE OF DATA BLOCK IN BYTES
9067 075002 000207          RTS      PC           ;RETURN
9068 075004          ENDTST
9069 075004          L10102: TRAP      C$ETST
9070 075004 104401
9071          .SBTTL TEST 7: WRITE DATA RETRY
9072          ;*
9073          ;THIS TEST VERIFIES PROPER OPERATION OF THE WRITE DATA RETRY
9074          ;COMMAND (SPACE REVERSE, ERASE, WRITE DATA)
9075          ;
9076          ;
9077          ;THE TEST CONSISTS OF THE FOLLOWING 5 SUBTESTS
9078          ;
9079          ;
9080          ;
9081          ;-
9082 075006          BGNTST
9083 075006 012737 006256' 002170' MOV      #EPRT2,EPRTSW ;SECONDARY ERROR MESSAGE
9084 075014 005037 003124'          CLR      KTENABLE    ;TURN OFF KT11
9085 075020 004737 017176'          JSR      PC,KTOFF    ;TURN KT11 BACK OFF IF THERE
9090 075024 012700 104643'          MOV      #TST27ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
9091 075030 004737 016412'          JSR      PC,TSTSETUP ;DO INITIAL TEST SETUP
9092 075034 012737 000005 002206' MOV      #5,LOOPCNT   ;PERFORM 5 ITERATIONS
9093 075042 005037 101706'          CLR      T27CNT     ;CLEAR TAPE RECORD COUNTER

```



```

9140
9141 075172 004737 010562'      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
9142 075176 103407              BCS      25$            ;BR, IF COMMAND ISSUED OK
9143 075200 005237 002212'      INC      FATFLG        ;BUMP COUNT
9147 075204 010001              MOV      R0,R1         ;SAVE CONTENTS OF TSSR
9148 075206              ERRHRD   ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
                                TRAP      C$ERHRD
                                .WORD     702
                                .WORD     WRTMSG
                                .WORD     SFIMSG
                                075206 104456
                                075210 001276
                                075212 005046'
                                075214 011734'
9149 075216 25$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                075216 104406
9150
9151      ;*****
9152      ;
9153      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9154      ;
9155      ;*****
9156
9157 075220 004737 010714'      JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
9158 075224 103407              BCS      30$            ;BR, IF NO PROBLEM
9159 075226 010004              MOV      R0,R4         ;SET UP REWIND PACKET ADDRESS
9160 075230 005237 002212'      INC      FATFLG        ;BUMP COUNT
9164 075234              ERRHRD   ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     703
                                .WORD     T27RWN
                                .WORD     PKTSSR
                                075234 104456
                                075236 001277
                                075240 103065'
                                075242 011746'
9165 075244 30$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                075244 104406
9166
9167      ;*****
9168      ;
9169      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9170      ;
9171      ;*****
9172
9173 075246 013701 101560'      MOV      T27BFR+6,R1   ;PICK UP XSTO
9174 075252 010102              MOV      R1,R2         ;SET UP EXPECTED
9175 075254 052702 000002      BIS      @BIT1,R2      ;SET BOT BIT IN EXPECTED
9176 075260 020102              CMP      R1,R2         ;DOES EXP = REC'D
9177 075262 001406              BEQ      40$            ;BR, IF EQUAL (OK)
9178 075264 005237 002212'      INC      FATFLG        ;BUMP COUNT
9182 075270              ERRHRD   ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD     704
                                .WORD     T27BOT
                                .WORD     EXPREC
                                075270 104456
                                075272 001300
                                075274 102561'
                                075276 015374'
9183 075300 40$: CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                075300 104406
9184 075302 012737 000400 101666' MOV      @256.,T27SZ   ;SET UP RECORD SIZE
9185 075310 013737 003114' 101662' MOV      FREE,T27WB    ;ADDRESS OF WRITE BUFFER
9186
9187      ;*****
9188      ;
9189      ;WRITE DATA RETRY,ACK,CVC=1 COMMAND
9190      ;

```

```

9191
9192
9193 075316 012737 141005 101660'      MOV      #141005,T27PK3      ;WRITE DATA RETRY,ACK,CVC=1 COMMAND
9194 075324 012704 101660'      MOV      #T27PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
9195 075330 010465 000000      MOV      R4,TSDB(R5)       ;ISSUE COMMAND
9196 075334 004737 016150'      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
9197 075340 016501 000002      MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
9198 075344 012702 100206      MOV      #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
9199 075350 020102      CMP      R1,R2             ;ARE THEY EQUAL
9200 075352 001406      BEQ      75$              ;BR, IF OK
9201 075354 005237 002212'      INC      FATFLG           ;BUMP COUNT
9205 075360      ERRHRD  ERRNO,T27WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
9205 075360 104456      TRAP    C$ERHRD
9205 075362 001301      .WORD  705
9205 075364 102472'      .WORD  T27WDE
9205 075366 011746'      .WORD  PKTSSR
9206 075370      75$:   CKLOOP           ;LOOP IF SELECTED
9206 075370 104406      TRAP    C$CLP1
9207
9208
9209
9210
9211
9212
9213
9214 075372 013701 101560'      MOV      T27BFR+6,R1       ;GET XSTO STATUS WORD
9215 075376 010102      MOV      R1,R2             ;SET UP EXPECTED
9216 075400 052702 002000      BIS      #BIT10,R2        ;SET THE NEF BIT
9217 075404 020102      CMP      R1,R2             ;ARE THEY EQUAL
9218 075406 001406      BEQ      170$            ;BR, IF EQUAL (GOOD)
9219 075410 005237 002212'      INC      FATFLG           ;BUMP COUNT
9223 075414      ERRHRD  ERRNO,T27NEF,EXPREC ;NEF SHOULD BE SET
9223 075414 104456      TRAP    C$ERHRD
9223 075416 001302      .WORD  706
9223 075420 104231'      .WORD  T27NEF
9223 075422 015374'      .WORD  EXPREC
9224 075424      170$:  CKLOOP
9224 075424 104406      TRAP    C$CLP1
9225 075426      ENDSUB
9225 075426      L10123: TRAP    C$ESUB
9226 075430 023727 002212' 000017      CMP      FATFLG,#15.      ;IS ERROR COUNT AT 25
9227 075436 103402      BLO     999$             ;BR, IF LESS THAN 25
9228 075440 004737 017104'      JSR      PC,CKDROP        ;TRY TO DROP THE UNIT
9229 075444      999$:
9230
9231
9232
9233
9234
9235
9236
9237
9238
9239
9240
9241

```



```

075566 104456 TRAP C$ERHRD
075570 001305 .WORD 709
075572 103065' .WORD T27RWN
075574 011746' .WORD PKTSSR
9297 075576 26$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
075576 104406 ;STARTING RECORD SIZE
9298 075600 012703 000400 MOV #256.,R3 ;STARTING WRITE BUFFER ADDRESS
9299 075604 013737 003114' 101662' MOV FREE,T27WB
9300
9301 ;*****
9302 ;
9303 ;WRITE DATA,CVC=1,ACK COMMAND
9304 ;
9305 ;*****
9306
9307 075612 012737 140005 101660' MOV #140005,T27PK3 ;WRITE DATA,CVC=1,ACK COMMAND
9308 075620 012704 101660' MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
9309 075624 010337 101666' MOV R3,T27SZ ;SET UP RECORD SIZE IN PACKET
9310 075630 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
9311 075634 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
9312 075640 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9313 075644 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9314 075650 020102 CMP R1,R2 ;ARE THEY EQUAL
9315 075652 001406 BEQ 28$ ;BR, IF OK
9316 075654 005237 002212' INC FATFLG ;BUMP COUNT
9320 075660 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
075660 104456 TRAP C$ERHRD
075662 001306 .WORD 710
075664 005103' .WORD WRERR
075666 011746' .WORD PKTSSR
9321 075670 28$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
075670 104406
9322
9323 ;*****
9324 ;
9325 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9326 ;
9327 ;*****
9328
9329 075672 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
9330 075676 103411 BCS 30$ ;BR, IF NO PROBLEM
9331 075700 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9332 075704 010004 MOV R0,R4 ;SET UP REWIND PACKET ADDRESS
9333 075706 005237 002212' INC FATFLG ;BUMP COUNT
9337 075712 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
075712 104456 TRAP C$ERHRD
075714 001307 .WORD 711
075716 103065' .WORD T27RWN
075720 011746' .WORD PKTSSR
9338 075722 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
075722 104406
9339
9340 ;*****
9341 ;
9342 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9343 ;
9344 ;*****

```

```

9345
9346 075724 013701 101560'      MOV      T27BFR+6,R1      ;PICK UP XSTO
9347 075730 010102              MOV      R1,R2           ;SET UP EXPECTED
9348 075732 052702 000002      BIS      @BIT1,R2       ;SET BOT BIT IN EXPECTED
9349 075736 020102              CMP      R1,R2           ;DOES EXP = REC'D
9350 075740 001406              BEQ      40$            ;BR, IF EQUAL (OK)
9351 075742 005237 002212'      INC      FATFLG         ;BUMP COUNT
9355 075746              ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    712
                                .WORD    T27BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
                                .WORD    104456
                                .WORD    001310
                                .WORD    102561'
                                .WORD    015374'
9356 075756              40$:   CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406
9357
9358 ;*****
9359 ;
9360 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
9361 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
9362 ;
9363 ;*****
9364
9365 075760 012703 000001      MOV      @1,R3           ;PARAMETER SPACE FORWARD 1 RECORD
9366 075764 004737 010366'      JSR      PC,SPACE       ;CALL SPACE RECORDS ROUTINE
9367 075770 103413              BCS      50$            ;BR, IF NO ERRORS
9368 075772 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
9369 075776 012702 000200      MOV      @SSR,R2        ;SET UP EXPECTED
9370 076002 010004              MOV      R0,R4          ;SET UP REWIND PACKET ADDRESS
9371 076004 005237 002212'      INC      FATFLG         ;BUMP COUNT
9375 076010              ERRHRD  ERRNO,T27SCF,PKTSSR ;SPACE RECORDS COMMAND FAILED
                                TRAP      C$ERHRD
                                .WORD    713
                                .WORD    T27SCF
                                .WORD    PKTSSR
                                TRAP      C$CLP1
                                .WORD    104456
                                .WORD    001311
                                .WORD    104327'
                                .WORD    011746'
9376 076020              50$:   CKLOOP           ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406
9377
9378 ;*****
9379 ;
9380 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
9381 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
9382 ;
9383 ;*****
9384
9385 076022 012703 100001      MOV      @100001,R3     ;PARAMETER SPACE REVERSE 1 RECORD
9386 076026 004737 010366'      JSR      PC,SPACE       ;CALL SPACE RECORDS ROUTINE
9387 076032 103413              BCS      60$            ;BR, IF NO ERRORS
9388 076034 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
9389 076040 012702 000200      MOV      @SSR,R2        ;SET UP EXPECTED
9390 076044 010004              MOV      R0,R4          ;SET UP REWIND PACKET ADDRESS
9391 076046 005237 002212'      INC      FATFLG         ;BUMP COUNT
9395 076052              ERRHRD  ERRNO,T27SCF,PKTSSR ;SPACE RECORDS COMMAND FAILED
                                TRAP      C$ERHRD
                                .WORD    714
                                .WORD    T27SCF
                                .WORD    PKTSSR
                                TRAP      C$CLP1
                                .WORD    104456
                                .WORD    001312
                                .WORD    104327'
                                .WORD    011746'
9396 076062              60$:   CKLOOP           ;LOOP IF SELECTED

```



```

076346 001317
076350 103065'
076352 011746'
9501 076354 104406 30$: CKLOOP ;LOOP IF SELECTED .WORD 719
076354 104406 TRAP C$CLP1 .WORD T27RWN
9502 ;***** .WORD PKTSSR
9503 ;
9504 ;
9505 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9506 ;
9507 ;*****
9508
9509 076356 013701 101560' MOV T27BFR+6,R1 ;PICK UP XSTO
9510 076362 010102 MOV R1,R2 ;SET UP EXPECTED
9511 076364 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
9512 076370 020102 CMP R1,R2 ;DOES EXP = REC'D
9513 076372 001406 BEQ 40$ ;BR, IF EQUAL (OK)
9514 076374 005237 002212' INC FATFLG ;BUMP COUNT
9518 076400 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
076400 104456 TRAP C$ERHRD
076402 001320 .WORD 720
076404 102561' .WORD T27BOT
076406 015374' .WORD EXPREC
9519 076410 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
076410 104406
9520 076412 012703 000024 MOV #20.,R3 ;STARTING RECORD SIZE
9521 076416 013737 003114' 101662' MOV FREE,T27WB ;STARTING WRITE BUFFER ADDRESS
9522 ;*****
9523 ;
9524 ;
9525 ;WRITE DATA,CVC=1,ACK COMMAND
9526 ;
9527 ;*****
9528
9529 076424 012737 140005 101660' 65$: MOV #140005,T27PK3 ;WRITE DATA,CVC=1,ACK COMMAND
9530 076432 012704 101660' MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
9531 076436 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
9532 076440 004737 017324' JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
9533 076444 010337 101666' MOV R3,T27SZ ;SET UP RECORD SIZE IN PACKET
9534 076450 010465 000000 MOV R4,TSD8(R5) ;ISSUE COMMAND
9535 076454 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
9536 076460 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9537 076464 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
9538 076470 020102 CMP R1,R2 ;ARE THEY EQUAL
9539 076472 001406 BEQ 80$ ;BR, IF OK
9540 076474 005237 002212' INC FATFLG ;BUMP COUNT
9544 076500 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
076500 104456 TRAP C$ERHRD
076502 001321 .WORD 721
076504 005103' .WORD WRERR
076506 011746' .WORD PKTSSR
9545 076510 80$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
076510 104406
9546 ;*****
9547 ;
9548 ;
9549 ;WRITE DATA RETRY,CVC=1,ACK COMMAND

```

```

9550
9551 ;
9552 ;*****
9553 076512 012737 141005 101660'      MOV      #141005,T27PK3      ;WRITE DATA RETRY,CVC=1,ACK COMMAND
9554 076520 010465 000000                MOV      R4,TSDB(R5)        ;ISSUE COMMAND
9555 076524 004737 016150'              JSR      PC,WAITF           ;WAIT FOR SSR TO SET
9556 076530 016501 000002                MOV      TSSR(R5),R1       ;GET TSSR CONTENTS
9557 076534 012702 000200                MOV      #SSR,R2          ;SET UP EXPECTED
9558 076540 020102                        CMP      R1,R2             ;ARE THEY EQUAL
9559 076542 001406                        BEQ      90$               ;BR, IF OK
9560 076544 005237 002212'              INC      FATFLG            ;BUMP COUNT
9564 076550                                ERRHRD  ERRNO,T27WRF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA RETRY
                                TRAP  C$ERHRD
                                .WORD 722
                                .WORD T27WRF
                                .WORD PKTSSR
                                TRAP  C$CLP1
076550 104456
076552 001322
076554 104466'
076556 011746'
9565 076560                                90$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP  C$CLP1
076560 104406
9566 076562 005723                        TST      (R3),             ;BUMP RECORD SIZE COUNTER
9567 076564 020327 000050                CMP      R3,#40.          ;AT 40 SIZE YET
9568 076570 001315                        BNE      65$               ;BR, IF MORE RECORDS TO WRITE
9569
9570 ;*****
9571 ;
9572 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9573 ;
9574 ;*****
9575
9576 076572 004737 010714'              JSR      PC,REWIND         ;CALL TAPE REWIND COMMAND
9577 076576 103407                        BCS     230$              ;BR, IF NO PROBLEM
9578 076600 010001                        MOV      R0,R1            ;SAVE TSSR
9579 076602 005237 002212'              INC      FATFLG            ;BUMP COUNT
9583 076606                                ERRHRD  ERRNO,T27RWN,EXPREC ;REWIND NOT ACCEPTED
                                TRAP  C$ERHRD
                                .WORD 723
                                .WORD T27RWN
                                .WORD EXPREC
076606 104456
076610 001323
076612 103065'
076614 015374'
9584 076616                                230$: CKLOOP                ;LOOP IF SELECTED
                                TRAP  C$CLP1
076616 104406
9585
9586 ;*****
9587 ;
9588 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9589 ;
9590 ;*****
9591
9592 076620 013701 101560'              MOV      T27BFR+6,R1      ;PICK UP XSTO
9593 076624 010102                        MOV      R1,R2            ;SET UP EXPECTED
9594 076626 052702 000002                BIS      #BIT1,R2         ;SET BOT BIT IN EXPECTED
9595 076632 020102                        CMP      R1,R2            ;DOES EXP = REC'D
9596 076634 001406                        BEQ      240$              ;BR, IF EQUAL (OK)
9597 076636 005237 002212'              INC      FATFLG            ;BUMP COUNT
9601 076642                                ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP  C$ERHRD
                                .WORD 724
                                .WORD T27BOT
                                .WORD EXPREC
076642 104456
076644 001324
076646 102561'
076650 015374'

```



```

9651 077034 023727 002212' 000017      CMP    FATFLG,#15.      ;IS ERROR COUNT AT 25
9652 077042 103402                      BLO    999$            ;BR, IF LESS THAN 25
9653 077044 004737 017104'              JSR    PC,CKDROP      ;TRY TO DROP THE UNIT
9654 077050                      999$:
9655
9656
9657
9658
9659
9660
9661
9662
9663
9664
9665
9666
9667
9668 077050                      ;*
077050                      ;TEST 7, SUBTEST 4
077050 104402                      ;
077052 004737 104664'              ;VERIFIES THAT A WRITE DATA RETRY COMMAND WITH SWB=1
077056 004737 104756'              ;TERMINATES PROPERLY AND WRITES CORRECT DATA ON TAPE
077062 004737 105020'              ;(THE WRITTEN RECORD IS READ AND CHECKED). VARIOUS
077066 012737 176750 101712'        ;BYTE COUNTS AND DATA PATTERNS ARE USED.
9673
9674
9675
9676
9677
9678
9679
9680 077074 004737 015674'          ;
077100 103426                      ;*****
077102 012727 000250              ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
077106 000000                      ;*****
077110 013727 002116'              10$: JSR    PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
077114 000000                      BCS    20$            ;BR IF INIT WAS OK
077116 005367 177772              DELAY  250           ;DELAY ABOUT .25 SEC
077122 001375                      MOV    #250,(PC)+
077124 005367 177756              .WORD 0
077130 001367                      MOV    L#DLY,(PC)+
077132 005337 101712'              .WORD 0
077136 001356                      DEC    -6(PC)
077140 005237 002212'              BNE    -4
077144 010001                      DEC    -22(PC)
077146 104455                      BNE    -20
077150 001327                      DEC    T27DLY        ;BUMP COUNTER
077152 003642'                      BNE    10$          ;BR, IF COUNTER NOT DONE
077154 011734'                      INC    FATFLG        ;BUMP COUNT
9691 077156 013737 002172' 101550' 20$: MOV    RO,R1          ;CONTENTS OF TSSR REGISTER
9692 077164 012704 101530'          ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
9693
9694
9695
9696

```

```

9697
9698
9699
9700 077170 004737 010562'
9701 077174 103407
9702 077176 005237 002212'
9706 077202 010001
9707 077204
      077204 104456
      077206 001330
      077210 005046'
      077212 011734'
9708 077214
      077214 104406
9709
9710
9711
9712
9713
9714
9715
9716 077216 004737 010714'
9717 077222 103411
9718 077224 016501 000002
9719 077230 010004
9720 077232 005237 002212'
9724 077236
      077236 104456
      077240 001331
      077242 103065'
      077244 011746'
9725 077246
      077246 104406
9726
9727
9728
9729
9730
9731
9732
9733 077250 013701 101560'
9734 077254 010102
9735 077256 052702 000002
9736 077262 020102
9737 077264 001406
9738 077266 005237 002212'
9742 077272
      077272 104456
      077274 001332
      077276 102561'
      077300 015374'
9743 077302
      077302 104406
9744 077304 012703 000024
9745 077310 013737 003114' 101662'
9746
9747

```

```

;
;*****
      JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
      BCS     23$                ;BR, IF COMMAND ISSUED OK
      INC     FATFLG             ;BUMP COUNT
      MOV     RO,R1              ;SAVE CONTENTS OF TSSR
      ERRHRD ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICS FAILED
                                  TRAP      C$ERHRD
                                  .WORD     728
                                  .WORD     WRTMSG
                                  .WORD     SFMSG
23$:   CKLOOP                    ;LOOP IF SELECTED
                                  TRAP      C$CLP1
;*****
;
;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
;
;*****
      JSR     PC,REWIND          ;CALL TAPE REWIND COMMAND
      BCS     30$                ;BR, IF NO PROBLEM
      MOV     TSSR(R5),R1       ;GET TSSR CONTENTS
      MOV     RO,R4              ;GET PACKET ADDRESS
      INC     FATFLG             ;BUMP COUNT
      ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
                                  TRAP      C$ERHRD
                                  .WORD     729
                                  .WORD     T27RWN
                                  .WORD     PKTSSR
30$:   CKLOOP                    ;LOOP IF SELECTED
                                  TRAP      C$CLP1
;*****
;
;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
;
;*****
      MOV     T27BFR+6,R1       ;PICK UP XSTO
      MOV     R1,R2              ;SET UP EXPECTED
      BIS     @BIT1,R2          ;SET BOT BIT IN EXPECTED
      CMP     R1,R2              ;DOES EXP = REC'D
      BEQ     40$                ;BR, IF EQUAL (OK)
      INC     FATFLG             ;BUMP COUNT
      ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                  TRAP      C$ERHRD
                                  .WORD     730
                                  .WORD     T27BOT
                                  .WORD     EXPREC
40$:   CKLOOP                    ;LOOP IF SELECTED
                                  TRAP      C$CLP1
      MOV     @20.,R3            ;STARTING RECORD SIZE
      MOV     FREE,T27WB        ;STARTING WRITE BUFFER ADDRESS
;*****

```

```

9748                                     ;
9749                                     ;WRITE DATA,CVC=1,ACK COMMAND
9750                                     ;
9751                                     ;*****
9752                                     ;
9753 077316 012737 140005 101660' 65$:  MOV    #140005,T27PK3      ;WRITE DATA,CVC=1,ACK COMMAND
9754 077324 012704 101660'          MOV    #T27PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
9755 077330 010300                    MOV    R3,R0          ;SET PATTERN IN CORRECT REGISTER
9756 077332 004737 017324'          JSR    PC,FILLMEM     ;FILL MEMORY WITH RECORD SIZE
9757 077336 010337 101666'          MOV    R3,T27SZ       ;SET UP RECORD SIZE IN PACKET
9758 077342 010465 000000            MOV    R4,TSDB(R5)    ;ISSUE COMMAND
9759 077346 004737 016150'          JSR    PC,WAITF       ;WAIT FOR SSR TO SET
9760 077352 016501 000002            MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
9761 077356 012702 000200            MOV    #SSR,R2        ;SET UP EXPECTED
9762 077362 020102                    CMP    R1,R2          ;ARE THEY EQUAL
9763 077364 001406                    BEQ    80$            ;BR, IF OK
9764 077366 005237 002212'          INC    FATFLG         ;BUMP COUNT
9768 077372                    ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERHRD
                                .WORD   731
                                .WORD   WRTErr
                                .WORD   PKTSSR
9769 077402                    80$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
9770
9771                                     ;*****
9772                                     ;
9773                                     ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
9774                                     ;
9775                                     ;*****
9776                                     ;
9777 077404 012737 111005 101660'    MOV    #111005,T27PK3 ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
9778 077412 010465 000000            MOV    R4,TSDB(R5)    ;ISSUE COMMAND
9779 077416 004737 016150'          JSR    PC,WAITF       ;WAIT FOR SSR TO SET
9780 077422 016501 000002            MOV    TSSR(R5),R1    ;GET TSSR CONTENTS
9781 077426 012702 000200            MOV    #SSR,R2        ;SET UP EXPECTED
9782 077432 020102                    CMP    R1,R2          ;ARE THEY EQUAL
9783 077434 001406                    BEQ    90$            ;BR, IF OK
9784 077436 005237 002212'          INC    FATFLG         ;BUMP COUNT
9788 077442                    ERRHRD  ERRNO,T27WRF,EXPREC ;TSSR INCORRECT AFTER WRITE DATA RETRY
                                TRAP    C$ERHRD
                                .WORD   732
                                .WORD   T27WRF
                                .WORD   EXPREC
9789 077452                    90$:  CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
9790 077454 005723                    TST    (R3)+          ;BUMP RECORD SIZE COUNTER
9791 077456 020327 000050            CMP    R3,#40         ;AT 40 SIZE YET
9792 077462 001315                    BNE    65$            ;BR, IF MORE RECORDS TO WRITE
9793
9794                                     ;*****
9795                                     ;
9796                                     ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9797                                     ;
9798                                     ;*****
9799
9800 077464 004737 010714'          JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND

```

```

9801 077470 103411          BCS      230$          ;BR, IF NO PROBLEM
9802 077472 016501 000002  MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
9803 077476 010004          MOV      R0,R4        ;GET PACKET ADDRESS
9804 077500 005237 002212'  INC      FATFLG       ;BUMP COUNT
9808 077504          ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
          077504 104456          TRAP    C$ERHRD
          077506 001335          .WORD  733
          077510 103065'         .WORD  T27RWN
          077512 011746'         .WORD  PKTSSR
9809 077514          230$:  CKLOOP          ;LOOP IF SELECTED
          077514 104406          TRAP    C$CLP1
9810
9811          ;*****
9812          ;
9813          ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9814          ;
9815          ;*****
9816
9817 077516 013701 101560'  MOV      T27BFR+6,R1  ;PICK UP XSTO
9818 077522 010102          MOV      R1,R2        ;SET UP EXPECTED
9819 077524 052702 000002  BIS      #BIT1,R2     ;SET BOT BIT IN EXPECTED
9820 077530 020102          CMP      R1,R2        ;DOES EXP = REC'D
9821 077532 001406          BEQ      240$         ;BR, IF EQUAL (OK)
9822 077534 005237 002212'  INC      FATFLG       ;BUMP COUNT
9826 077540          ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
          077540 104456          TRAP    C$ERHRD
          077542 001336          .WORD  734
          077544 102561'         .WORD  T27BOT
          077546 015374'         .WORD  EXPREC
9827 077550          240$:  CKLOOP          ;LOOP IF SELECTED
          077550 104406          TRAP    C$CLP1
9828 077552 012703 000024  MOV      #20.,R3      ;STARTING RECORD SIZE
9829 077556 013737 003114' 101662'  MOV      FREE,T27RB   ;STARTING READ BUFFER ADDRESS
9830
9831          ;*****
9832          ;
9833          ;READ DATA,ACK COMMAND
9834          ;
9835          ;*****
9836
9837 077564 012737 100001 101660' 265$:  MOV      #100001,T27PK3 ;READ DATA,ACK COMMAND
9838 077572 012704 101660'  MOV      #T27PK3,R4   ;SET UP R4 WITH PACKET ADDRESS
9839 077576 010337 101666'  MOV      R3,T27SZ     ;SET UP RECORD SIZE IN PACKET
9840 077602 010465 000000  MOV      R4,TSDB(R5)  ;ISSUE COMMAND
9841 077606 004737 016150'  JSR      PC,WAITF     ;WAIT FOR SSR TO SET
9842 077612 016501 000002  MOV      TSSR(R5),R1  ;GET TSSR CONTENTS
9843 077616 012702 000200  MOV      #SSR,R2     ;SET UP EXPECTED
9844 077622 020102          CMP      R1,R2        ;ARE THEY EQUAL
9845 077624 001406          BEQ      280$         ;BR, IF OK
9846 077626 005237 002212'  INC      FATFLG       ;BUMP COUNT
9850 077632          ERRHRD  ERRNO,RDERR,PKTSSR ;TSSR INCORRECT AFTER READ DATA
          077632 104456          TRAP    C$ERHRD
          077634 001337          .WORD  735
          077636 005176'         .WORD  RDERR
          077640 011746'         .WORD  PKTSSR
9851 077642          280$:  CKLOOP          ;LOOP IF SELECTED
          077642 104406          TRAP    C$CLP1

```



```

9950 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTPHR)
9951 ;
9952 ;*****
9953
9954 100106 004737 010562' JSR PC,WRTPHR ;ISSUE WRITE CHARACTERISTICS
9955 100112 103407 BCS 23$ ;BR, IF COMMAND ISSUED OK
9956 100114 005237 002212' INC FATFLG ;BUMP COUNT
9960 100120 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
9961 100122 ERRHRD ERRNO,WRTPHR,SFIMSG ;WRITE CHARACTERISTICS FAILED
100122 104456 TRAP C$ERHRD
100124 001342 .WORD 738
100126 005046' .WORD WRTPHR
100130 011734' .WORD SFIMSG
9962 100132 23$: CKLOOP ;LOOP IF SELECTED
100132 104406 TRAP C$CLP1
9963 ;*****
9964 ;
9965 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
9966 ;
9967 ;*****
9968
9969
9970 100134 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
9971 100140 103411 BCS 30$ ;BR, IF NO PROBLEM
9972 100142 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
9973 100146 010004 MOV RO,R4 ;GET PACKET ADDRESS
9974 100150 005237 002212' INC FATFLG ;BUMP COUNT
9978 100154 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
100154 104456 TRAP C$ERHRD
100156 001343 .WORD 739
100160 103065' .WORD T27RWN
100162 011746' .WORD PKTSSR
9979 100164 30$: CKLOOP ;LOOP IF SELECTED
100164 104406 TRAP C$CLP1
9980 ;*****
9981 ;
9982 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
9983 ;
9984 ;*****
9985
9986
9987 100166 013701 101560' MOV T27BFR+6,R1 ;PICK UP XSTO
9988 100172 010102 MOV R1,R2 ;SET UP EXPECTED
9989 100174 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
9990 100200 020102 CMP R1,R2 ;DOES EXP = REC'D
9991 100202 001406 BEQ 40$ ;BR, IF EQUAL (OK)
9992 100204 005237 002212' INC FATFLG ;BUMP COUNT
9996 100210 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
100210 104456 TRAP C$ERHRD
100212 001344 .WORD 740
100214 102561' .WORD T27BOT
100216 015374' .WORD EXPREC
9997 100220 40$: CKLOOP ;LOOP IF SELECTED
100220 104406 TRAP C$CLP1
9998 100222 012703 000144 MOV #100.,R3 ;NUMBER OF RECORDS TO BE WRITTEN
9999 100226 013737 003114' 101662' MOV FREE,T27WB ;STARTING WRITE BUFFER ADDRESS
10000

```

```

10001 ;*****
10002 ;
10003 ;WRITE DATA,ACK,CVC=1 COMMAND
10004 ;
10005 ;*****
10006
10007 100234 012737 140005 101660' 65$:  MOV      #140005,T27PK3      ;WRITE DATA,ACK,CVC=1 COMMAND
10008 100242 012704 101660'      MOV      #T27PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
10009 100246 012737 000024 101666'      MOV      #20.,T27SZ       ;SET UP RECORD SIZE IN PACKET
10010 100254 010465 000000      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
10011 100260 004737 016150'      JSR      PC,WAITF         ;WAIT FOR SSR TO SET
10012 100264 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
10013 100270 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
10014 100274 020102      CMP      R1,R2           ;ARE THEY EQUAL
10015 100276 001406      BEQ      70$             ;BR, IF OK
10016 100300 005237 002212'      INC      FATFLG          ;BUMP COUNT
10020 100304      ERRHRD  ERRNO,WRTErr,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      100304 104456      TRAP    C$ERHRD
      100306 001345      .WORD  741
      100310 005103'      .WORD  WRTErr
      100312 011746'      .WORD  PKTSSR
10021 100314      70$:  CKLOOP                ;LOOP IF SELECTED      TRAP    C$CLP1
      100314 104406
10022 100316 005303      DEC      R3              ;DEC RECORD COUNTER
10023 100320 001345      BNE     65$             ;BR, IF MORE RECORDS TO WRITE
10024
10025 ;*****
10026 ;
10027 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10028 ;
10029 ;*****
10030
10031 100322 004737 010714'      JSR      PC,REWIND       ;CALL TAPE REWIND COMMAND
10032 100326 103411      BCS     130$            ;BR, IF NO PROBLEM
10033 100330 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
10034 100334 010004      MOV      R0,R4          ;GET PACKET ADDRESS
10035 100336 005237 002212'      INC      FATFLG          ;BUMP COUNT
10039 100342      ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
      100342 104456      TRAP    C$ERHRD
      100344 001346      .WORD  742
      100346 103065'      .WORD  T27RWN
      100350 011746'      .WORD  PKTSSR
10040 100352      130$: CKLOOP                ;LOOP IF SELECTED      TRAP    C$CLP1
      100352 104406
10041
10042 ;*****
10043 ;
10044 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
10045 ;
10046 ;*****
10047
10048 100354 013701 101560'      MOV      T27BFR+6,R1     ;PICK UP XST0
10049 100360 010102      MOV      R1,R2          ;SET UP EXPECTED
10050 100362 052702 000002      BIS     #BIT1,R2        ;SET BOT BIT IN EXPECTED
10051 100366 020102      CMP     R1,R2           ;DOES EXP = REC'D
10052 100370 001406      BEQ     140$            ;BR, IF EQUAL (OK)
10053 100372 005237 002212'      INC     FATFLG          ;BUMP COUNT

```

```

10057 100376          ERRHRD  ERRNO,T27BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      100376 104456          TRAP          C$ERHRD
      100400 001347          .WORD        743
      100402 102561'        .WORD        T27BOT
      100404 015374'        .WORD        EXPREC
10058 100406          140$:  CKLOOP                    ;LOOP IF SELECTED          TRAP          C$CLP1
      100406 104406
10059 100410 012704 101660'  MOV          #T27PK3,R4      ;SET UP PACKET ADDRESS
10060 100414 012737 000010 101662'  MOV          #10,T27RB      ;SET UP RECORDS TO SPACE OVER
10061
10062          ;*****
10063          ;
10064          ;ACK,CVC=1,SPACE FORWARD COMMAND
10065          ;
10066          ;*****
10067
10068 100422 012737 140010 101660'  MOV          #140010,T27PK3 ;ACK,CVC=1,SPACE FORWARD COMMAND
10069 100430 010465 000000          150$:  MOV          R4,TSDB(R5) ;ISSUE COMMAND
10070 100434 005237 101706'          152$:  INC          T27CNT      ;BUMP TIMER
10071 100440          DELAY          1          ;DELAY ABOUT 100US
      100440 012727 000001          MOV          #1,(PC)+
      100444 000000          .WORD        0
      100446 013727 002116'        MOV          L$DLY,(PC)+
      100452 000000          .WORD        0
      100454 005367 177772          DEC          -6(PC)
      100460 001375          BNE          -.4
      100462 005367 177756          DEC          -22(PC)
      100466 001367          BNE          .-20
10072 100470 016501 000002          MOV          TSSR(R5),R1      ;GET TSSR
10073 100474 032701 000200          BIT          #BIT7,R1        ;CHECK FOR TSSR'S SSR SET
10074 100500 001755          BEQ          152$            ;KEEP COUNTING UNTIL SET
10075 100502 016501 000002          MOV          TSSR(R5),R1      ;GET STATUS FROM TSSR
10076 100506 012702 000200          MOV          #SSR,R2         ;SET UP EXPECTED
10077 100512 020201          CMP          R2,R1           ;WAS EVERYTHING OK
10078 100514 001406          BEQ          160$            ;BR, IF ALL IS WELL
10079 100516 005237 002212'          INC          FATFLG          ;BUMP COUNT
10083 100522          ERRHRD  ERRNO,T27SCF,PKTSSR ;SPACE FORWARD DIDN'T WORK OUT
      100522 104456          TRAP          C$ERHRD
      100524 001350          .WORD        744
      100526 104327'        .WORD        T27SCF
      100530 011746'        .WORD        PKTSSR
10084 100532          160$:  CKLOOP                    ;LOOP IF SELECTED          TRAP          C$CLP1
      100532 104406
10085
10086          ;*****
10087          ;
10088          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10089          ;
10090          ;*****
10091
10092 100534 004737 010714'          JSR          PC,REWIND        ;CALL TAPE REWIND COMMAND
10093 100540 004737 016236'          JSR          PC,CHKTSSR      ;SEE HOW TSSR IS
10094 100544 103407          BCS          170$            ;BR, IF NO PROBLEM
10095 100546 010001          MOV          R0,R1           ;SAVE TSSR
10096 100550 005237 002212'          INC          FATFLG          ;BUMP COUNT
10100 100554          ERRHRD  ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
      100554 104456          TRAP          C$ERHRD

```

```

100556 001351 .WORD 745
100560 103065' .WORD T27RWN
100562 011746' .WORD PKTSSR
10101 100564 170$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
100564 104406
10102 ;*****
10103 ;
10104 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
10105 ;
10106 ;*****
10107 ;
10108 ;
10109 100566 013701 101560' MOV T27BFR+6,R1 ;PICK UP XSTO
10110 100572 010102 MOV R1,R2 ;SET UP EXPECTED
10111 100574 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
10112 100600 020102 CMP R1,R2 ;DOES EXP = REC'D
10113 100602 001406 BEQ 175$ ;BR, IF EQUAL (OK)
10114 100604 005237 002212' INC FATFLG ;BUMP COUNT
10118 100610 ERRHRD ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
100610 104456 TRAP C$ERHRD
100612 001352 .WORD 746
100614 102561' .WORD T27BOT
100616 015374' .WORD EXPREC
10119 100620 175$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
100620 04406 ;STARTING RECORD SIZE
10120 100622 012703 000144 MOV #100.,R3 ;STARTING WRITE BUFFER ADDRESS
10121 100626 013737 003114' 101662' 177$: MOV FREE,T27WB
10122 ;*****
10123 ;
10124 ;WRITE DATA,CVC=1,ACK COMMAND
10125 ;
10126 ;*****
10127 ;
10128 ;
10129 100634 012737 140005 101660' MOV #140005,T27PK3 ;WRITE DATA,CVC=1,ACK COMMAND
10130 100642 012704 101660' MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
10131 100646 012737 000024 101666' MOV #20.,T27SZ ;SET UP RECORD SIZE IN PACKET
10132 100654 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
10133 100660 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
10134 100664 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10135 100670 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
10136 100674 020102 CMP R1,R2 ;ARE THEY EQUAL
10137 100676 001406 BEQ 180$ ;BR, IF OK
10138 100700 005237 002212' INC FATFLG ;BUMP COUNT
10142 100704 ERRHRD ERRNO,WRERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
100704 104456 TRAP C$ERHRD
100706 001353 .WORD 747
100710 005103' .WORD WRERR
100712 011746' .WORD PKTSSR
10143 100714 180$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
100714 104406 ;COUNT NUMBER OF RECORDS
10144 100716 005303 DEC R3 ;BR, IF MORE RECORDS TO WRITE
10145 100720 001342 BNE 177$
10146 ;*****
10147 ;
10148 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10149 ;

```

```

10150
10151 ;
10152 ;*****
10153 100722 004737 010714' JSR PC,REWIND ;ISSUE REWIND
10154 100726 103411 BCS 182$ ;BR, IF ALL IS WELL
10155 100730 010004 MOV R0,R4 ;GET PACKET ADDRESS
10156 100732 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10157 100736 005237 002212' INC FATFLG ;BUMP COUNT
10161 100742 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND FAILED
10162 100752 182$: CKLOOP ;SELECT LOOP MAYBE TRAP C$ERHRD
10163 100752 104406 ; TRAP 748
10164 ;***** ;.WORD T27RWN
10165 ; ;.WORD PKTSSR
10166 ;ISSUE SPACE RECORDS COMMAND - VALUE IN R3 SETS NUMBER OF RECORDS
10167 ;BIT 15 SETS DIRECTION - 0=FORWARD 1=REVERSE
10168 ;
10169 ;*****
10170 ;
10171 100754 012703 000001 MOV #1.,R3 ;SPACE 1 RECORD FORWARD
10172 100760 004737 010366' JSR PC,SPACE ;ISSUE SPACE COMMAND
10173 100764 103411 BCS 185$ ;BR, IF COMMAND OK
10174 100766 010004 MOV R0,R4 ;GET PACKET ADDRESS
10175 100770 016501 000002 MOV TSSR(R5),R1 ;GET TSSR STATUS
10176 100774 005237 002212' INC FATFLG ;BUMP COUNT
10180 101000 ERRHRD ERRNO,T27SCF,PKTSSR ;SPACE FORWARD COMMAND FAILED
10181 101000 104456 TRAP C$ERHRD
10182 101002 001355 ;.WORD 749
10183 101004 104327' ;.WORD T27SCF
10184 101006 011746' ;.WORD PKTSSR
10185 185$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
10186 ;
10187 ;WRITE DATA RETRY,ACK COMMAND
10188 ;
10189 ;*****
10190 ;
10191 101024 012737 101005 101660' 190$: MOV #101005,T27PK3 ;WRITE DATA RETRY,ACK COMMAND
10192 101032 012704 101660' MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
10193 101036 012737 000024 101666' MOV #20.,T27SZ ;SET UP RECORD SIZE IN PACKET
10194 101044 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
10195 101050 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
10196 101054 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10197 101060 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
10198 101064 020102 CMP R1,R2 ;ARE THEY EQUAL
10199 101066 001406 BEQ 200$ ;BR, IF OK
10200 101070 005237 002212' INC FATFLG ;BUMP COUNT
10204 101074 ERRHRD ERRNO,T27WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
10205 101074 104456 TRAP C$ERHRD

```

```

101076 001356
101100 103421'
101102 011746'
10205 101104 2000: CKLOOP ;LOOP IF SELECTED .WORD 750
101104 104406 ;STARTING WRITE BUFFER ADDRESS .WORD T27WDC
10206 101106 013737 003114' 101662' MOV FREE,T27WB ;TRAP C1CLP1
10207
10208 ;.....
10209 ;WRITE DATA,CVC=1,ACK COMMAND
10210 ;
10211 ;.....
10212
10213
10214 101114 012737 140005 101660' MOV #140005,T27PK3 ;WRITE DATA,CVC=1,ACK COMMAND
10215 101122 012704 101660' MOV #T27PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
10216 101126 012737 000024 101666' MOV #20.,T27SZ ;SET UP RECORD SIZE IN PACKET
10217 101134 010465 000000 MOV R4,TSD8(R5) ;ISSUE COMMAND
10218 101140 004737 016150' JSR PC,WAITF ;WAIT FOR SSR TO SET
10219 101144 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10220 101150 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
10221 101154 020102 CMP R1,R2 ;ARE THEY EQUAL
10222 101156 001406 BEQ 2100 ;BR, IF OK
10223 101160 005237 002212' INC FATFLG ;BUMP COUNT
10227 101164 ERRHRD ERRNO,WRTERR,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
101164 104456 TRAP C1ERHRD
101166 001357 .WORD 751
101170 005103' .WORD WRTERR
101172 011746' .WORD PKTSSR
10228 101174 2100: CKLOOP ;LOOP IF SELECTED .WORD 752
101174 104406 ;STARTING WRITE BUFFER ADDRESS .WORD T27RWN
10229 101176 005303 DEC R3 ;BUMP DOWN RECORD COUNTER .WORD PKTSSR
10230 101200 001311 BNE 1900 ;BR, IF MORE RECORDS TO WRITE RETRY
10231
10232 ;.....
10233 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
10234 ;
10235 ;.....
10236
10237
10238 101202 004737 010714' JSR PC,REWIND ;CALL TAPE REWIND COMMAND
10239 101206 103411 BCS 2300 ;BR, IF NO PROBLEM
10240 101210 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10241 101214 010004 MOV R0,R4 ;GET PACKET ADDRESS
10242 101216 005237 002212' INC FATFLG ;BUMP COUNT
10246 101222 ERRHRD ERRNO,T27RWN,PKTSSR ;REWIND NOT ACCEPTED
101222 104456 TRAP C1ERHRD
101224 001360 .WORD 752
101226 103065' .WORD T27RWN
101230 011746' .WORD PKTSSR
10247 101232 2300: CKLOOP ;LOOP IF SELECTED .WORD 753
101232 104406 ;STARTING WRITE BUFFER ADDRESS .WORD T27RWN
10248
10249 ;.....
10250 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
10251 ;
10252 ;.....
10253

```



```

10254
10255 101234 013701 101560'      MOV      T27BFR+6,R1      ;PICK UP XSTO
10256 101240 010102      MOV      R1,R2          ;SET UP EXPECTED
10257 101242 052702 000002      BIS      @BIT1,R2       ;SET BOT BIT IN EXPECTED
10258 101246 020102      CMP      R1,R2          ;DOES EXP = REC'D
10259 101250 001406      BEQ      240$           ;BR, IF EQUAL (OK)
10260 101252 005237 002212'      INC      FATFLG         ;BUMP COUNT
10264 101256      ERRHRD  ERRNO,T27BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      101256 104456      TRAP      C$ERHRD
      101260 001361      .WORD    753
      101262 102561'     .WORD    T27BOT
      101264 015374'     .WORD    EXPREC
10265 101266      240$:  CKLOOP          ;LOOP IF SELECTED      TRAP      C$CLP1
      101266 104406
10266 101270 012704 101660'      MOV      @T27PK3,R4     ;SET UP PACKET ADDRESS
10267 101274 012737 000010 101662'  MOV      @10,T27RB      ;SET UP RECORDS TO SPACE OVER
10268
10269      ;*****
10270      ;
10271      ;ACK,CVC=1,SPACE FORWARD COMMAND
10272      ;
10273      ;*****
10274
10275 101302 012737 140010 101660'  MOV      @140010,T27PK3 ;ACK,CVC=1,SPACE FORWARD COMMAND
10276 101310 010465 000000 250$:  MOV      R4,TSDB(R5)    ;ISSUE COMMAND
10277 101314 005237 101710'  252$:  INC      T27CNU     ;BUMP TIMER
10278 101320      DELAY  1          ;DELAY ABOUT 100US
      101320 012727 000001      MOV      @1,(PC)+
      101324 000000      .WORD    0
      101326 013727 002116'     MOV      L$DLY,(PC)+
      101332 000000      .WORD    0
      101334 005367 177772      DEC      -6(PC)
      101340 001375      BNE     .-4
      101342 005367 177756      DEC      -22(PC)
      101346 001367      BNE     .-20
10279 101350 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR
10280 101354 032701 000200      BIT      @BIT7,R1       ;CHECK FOR TSSR'S SSR SET
10281 101360 001755      BEQ      252$           ;KEEP COUNTING UNTIL SET
10282 101362 016501 000002      MOV      TSSR(R5),R1    ;GET STATUS FROM TSSR
10283 101366 012702 000200      MOV      @SSR,R2        ;SET UP EXPECTED
10284 101372 020201      CMP      R2,R1          ;WAS EVERYTHING OK
10285 101374 001406      BEQ      260$           ;BR, IF ALL IS WELL
10286 101376 005237 002212'      INC      FATFLG         ;BUMP COUNT
10290 101402      ERRHRD  ERRNO,T27SCF,PKTSSR ;SPACE FORWARD DIDN'T WORK OUT
      101402 104456      TRAP      C$ERHRD
      101404 001362      .WORD    754
      101406 104327'     .WORD    T27SCF
      101410 011746'     .WORD    PKTSSR
10291 101412      260$:  CKLOOP          ;LOOP IF SELECTED      TRAP      C$CLP1
      101412 104406
10292 101414 013701 101706'      MOV      T27CNT,R1      ;TIME FOR WRITE SPACING
10293 101420 013702 101710'      MOV      T27CNU,R2      ;TIME FOR WRITE RETRY SPACING
10294 101424 160102      SUB      R1,R2          ;GET'EM PRETTY CLOSE
10295 101426 160102      SUB      R1,R2          ;GET'EM PRETTY CLOSE
10296 101430 160102      SUB      R1,R2          ;GET'EM PRETTY CLOSE
10297 101432 042702 000077      BIC      @77,R2         ;CLEAR LOW 6 BITS
10298 101436 042701 000777      BIC      @000777,R1     ;SETTING UP CONSTANTS

```


10356	101662				T27RB:			
10357	101662	003114'			T27WB:	.WORD	FREE	;ADDRESS OF WRITE BUFFER
10358	101664	000000				.WORD	0	
10359	101666	000000			T27SZ:	.WORD	0	;SIZE OF BUFFER (EXTENT)
10360						.EVEN		
10361					:			
10362					:			
10363					:			
10364	101670				T27BF2:			
10365	101670	010			T27BS0:	.BYTE	10	;BSELO AREA
10366	101671	200			T27BS1:	.BYTE	200	;BSEL1 AREA
10367	101672	000000			T27S2:	.WORD	0	;SEL 2 AREA
10368	101674	000000			T27S3:	.WORD	0	;DATA AREA
10369					:			
10370					:			
10371						.EVEN		
10372								
10373								
10374	101676	100205			T27RN:	.WORD	100205	;REREAD DATA (NEXT)
10375	101700	100605			T27WDR:	.WORD	100605	;REREAD DATA RETRY
10376	101702	102205			T27CON:	.WORD	102205	;WRITE CONTINUOUS
10377	101704	177777				.WORD	177777	;END OF DATA
10378								
10379					:			
10380	101706	000000			T27CNT:	.WORD	0	;TAPE TIMER COUNTER STORAGE AREA
10381	101710	000000			T27CNU:	.WORD	0	;TAPE TIMER COUNTER STORAGE AREA
10382	101712	000000			T27DLY:	.WORD	0	;DELAY COUNTER
10383								
10384								
10385								
10386								
10387								
10388								
10389								
10390	101714	124	141	160	T27WNG:	.ASCIZ		'Tape Position Incorrect After REREAD Previous (OPP=1)'
10391	102002	124	123	123	T27RDF:	.ASCIZ		'TSSR Incorrect After READ DATA Command'
10392	102051	122	105	122	T27RRF:	.ASCIZ		'REREAD Previous (Space Reverse, Read Forward) Command Failed'
10393	102146	120	117	123	T27SC:	.ASCIZ		'POSITION (Space Command) Failed, TSSR Not Correct'
10394	102230	122	111	102	T27LOR:	.ASCIZ		'RIB NOT SET AFTER READ REVERSE INTO BOT'
10395	102300	124	123	123	T27WDF:	.ASCIZ		'TSSR Not Correct After Illegal Mode Bits Set'
10396	102355	111	154	154	T27LOQ:	.ASCIZ		'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
10397	102436	122	105	122	T27SSR:	.ASCIZ		'REREAD COMMAND Not Accepted'
10398	102472	124	123	123	T27WDE:	.ASCIZ		'TSSR Not Correct After WRITE DATA RETRY Command, At BOT'
10399	102561	124	141	160	T27BOT:	.ASCIZ		'Tape Not At BOT After REWIND Command (BOT Not Set In XSTO)'
10400	102654	127	122	111	T27TIM:	.ASCIZ		'WRITE DATA RETRY'S Erase Tape Not Long Enough'
10401	102731	122	105	122	T27EOT:	.ASCIZ		'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
10402	103010	124	123	123	T27TM:	.ASCIZ		'TSSR Not Correct After REREAD COMMAND Reject'
10403	103065	122	145	167	T27RWN:	.ASCIZ		'Rewind (POSITION) Command Not Accepted'
10404	103134	122	101	115	T27RNC:	.ASCIZ		'RAM Error, Correct Data Pattern Not In Ram'
10405	103207	124	123	123	T27AM3:	.ASCIZ		'TSSR Init. Failed After REREAD COMMAND'
10406	103256	104	162	151	T27OFL:	.ASCIZ		'Drive 7 Select Failed To Set "OFL" In TSSR'
10407	103331	124	123	123	T27WDD:	.ASCIZ		'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
10408	103421	124	123	123	T27WDC:	.ASCIZ		'TSSR Not Correct After REREAD DATA Command'
10409	103474	103	126	103	T27VCK:	.ASCIZ		'CVC Set, Didn't Reset VCK In Message Buffer'
10410	103547	124	123	102	T27BA:	.ASCIZ		'TSBA Not Correct After REREAD DATA Command'
10411	103622	127	122	111	T27WSS:	.ASCIZ		'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
10412	103711	122	145	141	T27LON:	.ASCIZ		'Reading Long Record Failed To Set RLL Bit In XSTO'

10413	103773	122	145	141	T27LOP:	.ASCIZ	'Reading Long Record Failed To Set RLS Bit In XST0'
10414	104055	122	145	163	T27PBP:	.ASCIZ	'Residual Byte Count Incorrect After Short Record Read'
10415	104143	122	145	141	T27TRL:	.ASCIZ	'Reading Long Record Failed To Give Tape Status Alert'
10416	104231	127	122	111	T27NEF:	.ASCIZ	'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
10417	104327	124	123	123	T27SCF:	.ASCIZ	'TSSR Not Correct After SPACE RECORDS Command'
10418	104404	124	123	123	T27TSA:	.ASCIZ	'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
10419	104466	124	123	123	T27WRF:	.ASCIZ	'TSSR Not Correct After WRITE DATA RETRY Command'
10420	104546	104	141	164	T27DTA:	.ASCIZ	'Data Compare Error, Data Read From Tape Not Equal To Written'
10421	104643	127	162	151	TST27ID:	.ASCIZ	'Write Data Retry'
10422						.EVEN	
10423							
10424							
10425							
10426							
10427							
10428							
10429							
10430	104664				T27REST:		
10431	104664				SAVREG		;SAVE THE REGISTERS
10432	104670	012701	101530'		MOV	@T27PACKET,R1	;START OF THE PACKET
10433	104674	012721	100004		MOV	@100004,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK.
10434	104700	012721	101540'		MOV	@T27DATA,(R1)+	;ADDRESS OF CHARAISTICS DATA BLOCK
10435	104704	005021			CLR	(R1)+	;EXTENDED ADDRESS
10436	104706	012721	000012		MOV	@10.,(R1)+	;SIZE OF DATA BLOCK IN BYTES
10437	104712	012721	101552'		MOV	@T27BFR,(R1)+	;ADDRESS OF MESSAGE BUFFER
10438	104716	005021			CLR	(R1)+	
10439	104720	012721	000024		MOV	@20.,(R1)+	;LENGTH OF MESSAGE BUFFER
10440	104724	005021			CLR	(R1)+	
10441	104726	012711	000000		MOV	@0,(R1)	;SELECT DRIVE ZERO
10442	104732	012702	000030		MOV	@24.,R2	;NUMBER OF LOCATIONS TO BE CLEARED
10443	104736	012762	177777	101552' 64#:	MOV	@177777,T27BFR(R2)	;ALL ONES TO MESSAGE BUFFER
10444	104744	005742			TST	-(R2)	;NEXT LOCATION
10445	104746	022702	000000		CMP	@0,R2	;AT END OF LOOP YET
10446	104752	001371			BNE	64#	;KEEP GOING UNTIL DONE
10447	104754	000207			RTS	PC	;RETURN
10448							
10449							
10450	104756				T27RT2:		
10451	104756				SAVREG		;SAVE THE REGISTERS
10452	104762	012701	101640'		MOV	@T27PK2,R1	;START OF THE PACKET
10453	104766	012721	100006		MOV	@100006,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK.
10454	104772	012721	101670'		MOV	@T27BF2,(R1)+	;ADDRESS OF DATA BLOCK
10455	104776	005021			CLR	(R1)+	;EXTENDED ADDRESS
10456	105000	012721	000006		MOV	@6.,(R1)+	;SIZE OF DATA BLOCK IN BYTES
10457	105004	005021			CLR	(R1)+	
10458	105006	012701	101670'		MOV	@T27BF2,R1	;POINT TO DATA SEL AREA
10459	105012	005021			CLR	(R1)+	
10460	105014	005011			CLR	(R1)	
10461	105016	000207			RTS	PC	;RETURN
10462	105020				T27RT3:		
10463	105020				SAVREG		;SAVE REGISTERS
10464	105024	012701	101660'		MOV	@T27PK3,R1	;SET UP POINTER ADDRESS
10465	105030	005021			CLR	(R1)+	;COMMAND SPACE
10466	105032	005021			CLR	(R1)+	;ADDRESS OF DATA BLOCK
10467	105034	005021			CLR	(R1)+	;EXTENDED ADDRESS
10468	105036	005011			CLR	(R1)	;SIZE OF DATA TRANSFER BLOCK
10469	105040	000207			RTS	PC	;RETURN


```

10526 105140 011734'
10526 105142 012737 000007 110260' 20$: MOV #7,T28DSW ;SET UP DRIVE NUMBER .WORD SFIMSG
10527 105150 012704 110240' MOV #T28PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
10528 105154 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
10529 105160 103407 BCS 24$ ;BR, IF COMMAND ISSUED OK
10530 105162 005237 002212' INC FATFLG ;BUMP COUNT
10534 105166 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
10535 105170 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
105170 104456 TRAP C$ERHRD
105172 001442 .WORD 802
105174 005046' .WORD WRTMSG
105176 011734' .WORD SFIMSG
10536 105200 24$: CKLOOP TRAP C$CLP1
105200 104406 ;CHECK FOR EXTENDED FEATURES SW SWITCH
10537 105202 005737 002216' TST EXTFEA ;BR IF SWITCH IS ON
10538 105206 001044 BNE 50$
10539
10540 105210 112737 000200 110401' MOVB #200,T28BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
10541 105216 112737 000010 110400' MOVB #10,T28BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
10542 105224 012704 110350' MOV #T28PK2,R4 ;WRITE SUBSYS MEM PACKET
10543 105230 010465 000000 MOV R4,TSD8(R5) ;ISSUE COMMAND
10544 105234 004737 016236' JSR PC,CHKTSSR ;WAIT FOR SSR
10545 105240 103407 BCS 30$ ;BR, IF NO ERROR
10546 105242 010001 MOV R0,R1 ;ERROR, SAVE TSSR
10547 105244 005237 002212' INC FATFLG ;BUMP COUNT
10551 105250 ERRHRD ERRNO,T28SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
105250 104456 TRAP C$ERHRD
105252 001443 .WORD 803
105254 111075' .WORD T28SSR
105256 011746' .WORD PKTSSR
10552 105260 30$: CKLOOP ;LOOP IF SELECTED
105260 104406 TRAP C$CLP1
10553 105262 012704 110240' MOV #T28PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
10554 105266 012737 000007 110260' MOV #7,T28DSW ;SELECT DRIVE 7
10555 105274 004737 010562' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
10556 105300 103407 BCS 50$ ;BR, IF COMMAND ISSUED OK
10557 105302 005237 002212' INC FATFLG ;BUMP COUNT
10561 105306 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
10562 105310 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
105310 104456 TRAP C$ERHRD
105312 001444 .WORD 804
105314 005046' .WORD WRTMSG
105316 011734' .WORD SFIMSG
10563 105320 50$: CKLOOP ;SCOPE LOOP
105320 104406 TRAP C$CLP1
10564 105322 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
10565 105326 032701 000100 BIT #0FL,R1 ;CHECK FOR THE OFFLINE BIT SET
10566 105332 001006 BNE 60$ ;BR, IF OFFLINE (GOOD)
10567 105334 005237 002212' INC FATFLG ;BUMP COUNT
10571 105340 ERRDF ERRNO,T28OFL,SFIMSG ;OFF LINE SHOULD HAVE BEEN SET (BAD)
105340 104455 TRAP C$ERDF
105342 001445 .WORD 805
105344 111430' .WORD T28OFL
105346 011734' .WORD SFIMSG
10572 105350 60$: CKLOOP ;LOOP IF SELECTED
105350 104406 TRAP C$CLP1
10573 105352 012703 110416' MOV #T28RN,R3 ;POINTER FOR COMMANDS

```



```

10672      ;*
10673      ;
10674      ;TEST 8, SUBTEST 3
10675      ;
10676      ;VERIFIES THAT WRITE TAPE MARK COMMANDS OPERATE
10677      ;PROPERLY, AND THAT READ COMMANDS SUBSEQUENTLY ISSUED
10678      ;TO DETECT THE WRITTEN TAPE MARKS TERMINATE WITH TAPE
10679      ;STATUS ALERT WITH THE TAPE MARK DETECTED (TMK) STATUS
10680      ;BIT SET. THE FOLLOWING SEQUENCE IS EXECUTED.
10681      ;
10682      ;1.   THE CONTROLLER IS INITIALIZED AND TAPE REWOUND.
10683      ;      THIS SETS THE VOLUME CHECK (VCK) STATUS BIT.
10684      ;
10685      ;2.   A WRITE TAPE MARK COMMAND WITH CVC=1 IS ISSUED
10686      ;      AND PROPER TERMINATION AND STATUS IS VERIFIED
10687      ;      (I.E. VCK=0 AND TMK=1).
10688      ;
10689      ;3.   SEVERAL MORE WRITE TAPE MARK COMMANDS, THESE WITH
10690      ;      CVC=0 ARE ISSUED AND PROPER TERMINATION (NORMAL)
10691      ;      AND STATUS (TMK) VERIFIED.
10692      ;
10693      ;4.   A READ REVERSE COMMAND IS ISSUED AND PROPER
10694      ;      TERMINATION (TAPE STATUS ALERT) AND STATUS (TMK)
10695      ;      VERIFIED. IT IS ALSO VERIFIED THAT NO DATA IS
10696      ;      TRANSFERRED INTO MEMORY.
10697      ;
10698      ;5.   A SPACE RECORDS REVERSE COMMAND IS ISSUED AND
10699      ;      PROPER TERMINATION (TAPE STATUS ALERT) AND STATUS
10700      ;      (TMK) VERIFIED.
10701      ;
10702      ;6.   THE TAPE IS REWOUND AND A READ FORWARD COMMAND IS
10703      ;      ISSUED AND PROPER TERMINATION (TAPE STATUS ALERT)
10704      ;      AND STATUS (TMK) VERIFIED. IT IS ALSO VERIFIED
10705      ;      THAT NO DATA IS TRANSFERRED INTO MEMORY.
10706      ;
10707      ;7.   A SPACE RECORDS REVERSE COMMAND THAT CONTAINS A
10708      ;      RECORD COUNT GREATER THAN 1 IS ISSUED AND IT IS
10709      ;      VERIFIED THAT TAPE STATUS ALERT TERMINATION
10710      ;      OCCURED, TMK=1 AND THAT RBPCR (RESIDUAL
10711      ;      BYTE/RECORD COUNTER) CONTAINS THE PROPER NONZERO
10712      ;      VALUE. THIS OPERATION VERIFIES THAT DETECTION OF
10713      ;      THE TAPE MARK CAUSES THE SPACE RECORDS OPERATION
10714      ;      TO BE PREMATURELY TERMINATED. THIS SHOULD LEAVE
10715      ;      THE POSITION JUST BEFORE THE FIRST RECORD ON
10716      ;      TAPE.
10717      ;
10718      ;8.   TAPE POSITION IS VERIFIED BY ISSUING ANOTHER
10719      ;      SPACE RECORDS REVERSE COMMAND AND VERIFYING THAT
10720      ;      TAPE STATUS ALERT TERMINATION OCCURS, WITH THE
10721      ;      REVERSE INTO BOT (RIB) STATUS ERROR BIT SET.
10722      ;
10723      ;9.   A SPACE RECORDS FORWARD COMMAND THAT CONTAINS A
10724      ;      RECORD COUNT GREATER THAN 1 IS ISSUED AND IT IS
10725      ;      VERIFIED THAT TAPE STATUS ALERT TERMINATION
10726      ;      OCCURED, TMK=1, AND THAT RBPCR (RESIDUAL
10727      ;      BYTE/RECORD COUNTER) CONTAINS THE PROPER NONZERO
10728      ;      VALUE. THIS OPERATION VERIFIES THAT DETECTION OF

```


	106132	111361'					.WORD	T28RWN
	106134	011746'					.WORD	PKTSSR
10774	106136		30\$:	CKLOOP		;LOOP IF SELECTED		
	106136	104406					TRAP	C\$CLP1
10775	106140	013701	110270'	MOV	T28BFR+6,R1	;PICK UP XSTO		
10776	106144	010102		MOV	R1,R2	;SET UP EXPECTED		
10777	106146	052702	000002	BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED		
10778	106152	020102		CMP	R1,R2	;DOES EXP = REC'D		
10779	106154	001406		BEQ	40\$;BR, IF EQUAL (OK)		
10780	106156	005237	002212'	INC	FATFLG	;BUMP COUNT		
10784	106162			ERRHRD	ERRNO,T28BOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	106162	104456					TRAP	C\$ERHRD
	106164	001456					.WORD	814
	106166	111237'					.WORD	T28BOT
	106170	015374'					.WORD	EXPREC
10785	106172		40\$:	CKLOOP		;LOOP IF SELECTED		
	106172	104406					TRAP	C\$CLP1
10786	106174	005737	002216'	42\$:	TST	EXTFEA		;CHECK FOR EXTENDED FEATURES SW SWITCH
10787	106200	001024		BNE	50\$;BR IF SWITCH IS ON		
10788	106202	112737	000200	110401'	MOVB	#200,T28BS1		;WRITE MISCELLANEOUS CONT/READ STATUS
10789	106210	112737	000010	110400'	MOVB	#10,T28BS0		;FUNC. SEL. BIT (TURN ON EXTFEA SWITCH)
10790	106216	012704	110350'	MOV	#T28PK2,R4	;WRITE SUBSYS MEM PACKET		
10791	106222	010465	000000	MOV	R4,TSDB(R5)	;ISSUE COMMAND		
10792	106226	004737	016236'	JSR	PC,CHKTSSR	;WAIT FOR SSR		
10793	106232	103407		BCS	50\$;BR, IF NO ERROR		
10794	106234	010001		MOV	R0,R1	;ERROR, SAVE TSSR		
10795	106236	005237	002212'	INC	FATFLG	;BUMP COUNT		
10799	106242			ERRHRD	ERRNO,T28SSR,PKTSSR	;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS		
	106242	104456					TRAP	C\$ERHRD
	106244	001457					.WORD	815
	106246	111075'					.WORD	T28SSR
	106250	011746'					.WORD	PKTSSR
10800	106252		50\$:	CKLOOP		;LOOP IF SELECTED		
	106252	104406					TRAP	C\$CLP1
10801	106254	012737	000007	110260'	MOV	#7,T28DSW		;SET UP DRIVE NUMBER
10802	106262	012704	110240'	MOV	#T28PACKET,R4	;SUBROUTINE NEEDS PACKET ADDRESS		
10803	106266	004737	010562'	JSR	PC,WRTCHR	;ISSUE WRITE CHARACTERISTICS		
10804	106272	103407		BCS	60\$;BR, IF COMMAND ISSUED OK		
10805	106274	005237	002212'	INC	FATFLG	;BUMP COUNT		
10809	106300	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR		
10810	106302			ERRHRD	ERRNO,WRTMSG,SFIMSG	;WRITE CHARACTERISTICS FAILED		
	106302	104456					TRAP	C\$ERHRD
	106304	001460					.WORD	816
	106306	005046'					.WORD	WRTMSG
	106310	011734'					.WORD	SFIMSG
10811	106312		60\$:	CKLOOP		;SCOPE LOOP		
	106312	104406					TRAP	C\$CLP1
10812	106314	016501	000002	MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
10813	106320	032701	000100	BIT	#OFL,R1	;CHECK FOR THE OFFLINE BIT SET		
10814	106324	001006		BNE	65\$;BR, IF OFFLINE (GOOD)		
10815	106326	005237	002212'	INC	FATFLG	;BUMP COUNT		
10819	106332			ERRDF	ERRNO,T28OFL,SFIMSG	;OFF LINE SHOULD HAVE BEEN SET (BAD)		
	106332	104455					TRAP	C\$ERDF
	106334	001461					.WORD	817
	106336	111430'					.WORD	T28OFL
	106340	011734'					.WORD	SFIMSG
10820	106342		65\$:	CKLOOP		;LOOP IF SELECTED		

10918	106762	004737	017324'		JSR	PC,FILLMEM	:FILL MEM WITH ALL ONES		
10919	106766	013737	003114'	110372'	MOV	FREE,T28WB	:STARTING READ BUFFER ADDRESS		
10920	106774	012737	140401	110370'	MOV	#140401,T28PK3	:READ REVERSE,ACK, COMMAND		
10921	107002	012704	110370'		MOV	#T28PK3,R4	:SET UP R4 WITH PACKET ADDRESS		
10922	107006	013737	000024	110376'	MOV	20.,T28SZ	:SET UP RECORD SIZE IN PACKET		
10923	107014	010465	000000		MOV	R4,TSDB(R5)	:ISSUE COMMAND		
10924	107020	004737	016150'		JSR	PC,WAITF	:WAIT FOR SSR TO SET		
10925	107024	016501	000002		MOV	TSSR(R5),R1	:GET TSSR CONTENTS		
10926	107030	012702	100204		MOV	#SSR!SC!BIT2,R2	:SET UP EXPECTED		
10927	107034	020102			CMP	R1,R2	:ARE THEY EQUAL		
10928	107036	001406			BEQ	200:	:BR, IF OK		
10929	107040	005237	002212'		INC	FATFLG	:BUMP COUNT		
10933	107044				ERRHRD	ERRNO,T28RDF,PKTSSR	:TSSR INCORRECT AFTER WRITE DATA		
	107044	104456					TRAP	C:ERHRD	
	107046	001472					.WORD	826	
	107050	110574'					.WORD	T28RDF	
	107052	011746'					.WORD	PKTSSR	
10934	107054			200:	CKLOOP		:LOOP IF SELECTED		
	107054	104406					TRAP	C:CLP1	
10935	107056	013701	110270'		MOV	T28BFR*6,R1	:PICK UP XSTO		
10936	107062	010102			MOV	R1,R2	:SET UP EXPECTED		
10937	107064	052702	100000		BIS	#BIT15,R2	:TMK SHOULD BE SET		
10938	107070	020102			CMP	R1,R2	:IS TMK SET		
10939	107072	001406			BEQ	210:	:BR, IF TMK WAS SET (GOOD)		
10940	107074	005237	002212'		INC	FATFLG	:BUMP COUNT		
10944	107100				ERRHRD	ERRNO,T28RRM,EXPREC	:TMK NOT SET AFTER READ REV		
	107100	104456					TRAP	C:ERHRD	
	107102	001473					.WORD	827	
	107104	111707'					.WORD	T28RRM	
	107106	015374'					.WORD	EXPREC	
10945	107110			210:	CKLOOP		:LOOP IF SELECTED		
	107110	104406					TRAP	C:CLP1	
10946	107112	017701	073776		MOV	#FREE,R1	:FIRST LOC IN READ BUFFER		
10947	107116	012702	177777		MOV	#177777,R2	:EXPECTED IF NO DATA TRANS.		
10948	107122	020102			CMP	R1,R2	:DID ANY DATA GET TRANSFERRED		
10949	107124	001406			BEQ	220:	:BR, IF NO DATA TRANS (GOOD)		
10950	107126	005237	002212'		INC	FATFLG	:BUMP COUNT		
10954	107132				ERRHRD	ERRNO,T28DTR,EXPREC	:DATA TRANSFERRED ON READ TAPE MARK		
	107132	104456					TRAP	C:ERHRD	
	107134	001474					.WORD	828	
	107136	112122'					.WORD	T28DTR	
	107140	015374'					.WORD	EXPREC	
10955	107142			220:	CKLOOP		:LOOP IF SELECTED		
	107142	104406					TRAP	C:CLP1	
10956	107144	012737	100410	110370'	MOV	#100410,T28PK3	:SPACE REVERSE,ACK, COMMAND		
10957	107152	012737	000001	110372'	MOV	#1,T28RB	:NUMBER OF RECORDS TO SPACE BACK		
10958	107160	012704	110370'		MOV	#T28PK3,R4	:SET UP R4 WITH PACKET ADDRESS		
10959	107164	010465	000000		MOV	R4,TSDB(R5)	:ISSUE COMMAND		
10960	107170	004737	016150'		JSR	PC,WAITF	:WAIT FOR SSR TO SET		
10961	107174	016501	000002		MOV	TSSR(R5),R1	:GET TSSR CONTENTS		
10962	107200	012702	100204		MOV	#SSR!SC!BIT2,R2	:SET UP EXPECTED		
10963	107204	020102			CMP	R1,R2	:ARE THEY EQUAL		
10964	107206	001406			BEQ	222:	:BR, IF OK		
10965	107210	005237	002212'		INC	FATFLG	:BUMP COUNT		
10969	107214				ERRHRD	ERRNO,T28RDG,PKTSSR	:TSSR INCORRECT AFTER SPACE CMD.		
	107214	104456					TRAP	C:ERHRD	
	107216	001475					.WORD	829	


```

11165 110250
11166 110250 110262'
11167 110252 000000
11168 110254 000024
11169 110256 000000
11170 110260 000000
11171 110262
11172
11173
11174
11176 110344
11178 110350
11179 110350 100006
11180 110352 110400'
11181 110354 000000
11182 110356 000006
11183
11185 110360
11187 110370
11188 110370 100005
11189 110372
11190 110372 003114'
11191 110374 000000
11192 110376 000000
11193
11194
11195
11196
11197 110400
11198 110400 010
11199 110401 200
11200 110402 000000
11201 110404 000000
11202
11203
11204
11205
11206
11207 110406
11208 110406 101411
11209 110410 102011
11210 110412 103411
11211 110414 177777
11212 110416 100011
11213 110420 100411
11214 110422 101011
11215 110424 177777
11216
11217
11218 110426 000000
11219 110430 000000
11220 110432 000000
11221
11222
11223
11224
11225

T28DATA:
      .WORD T28BFR
      .WORD 0
      .WORD 20.
      .WORD 0
T28DSW: .WORD 0
T28BFR: .BLKW 25.
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
T28PK2: .BLKB 10-<.-TSV2&7>
      .WORD 100006
      .WORD T28BF2
      .WORD 0
      .WORD 6.
;
T28PK3: .BLKB 10-<.-TSV2&7>
      .WORD 100005
;
T28RB:
T28WB: .WORD FREE
      .WORD 0
T28SZ: .WORD 0
      .EVEN
;
;
T28BF2:
T28BS0: .BYTE 10
T28BS1: .BYTE 200
T28S2: .WORD 0
T28S3: .WORD 0
;
;
      .EVEN
;TAPE MOTION PACKET COMMAND VALUES
T28IMV:
      .WORD 101411
      .WORD 102011
      .WORD 103411
      .WORD 177777
;
T28RN: .WORD 100011
T28WDR: .WORD 100411
T28CON: .WORD 101011
      .WORD 177777
;
;
T28CNT: .WORD 0
T28CNU: .WORD 0
T28DLY: .WORD 0
      .EVEN
;
;*
;LOCAL TEXT MESSAGES FOR TEST
;

```

;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER

;LENGTH OF MESSAGE BUFFER

;SELECT DRIVE 0
;MESSAGE BUFFER;WRITE SUB SYS MEM COMMAND, IE AND ACK
;ADDRESS OF SELECT BLOCK DATA

;SIZE OF DATA PACKET

;REREAD COMMAND, AND ACK

;ADDRESS OF WRITE BUFFER

;SIZE OF BUFFER (EXTENT)

;BSELO AREA

;BSEL1 AREA

;SEL 2 AREA

;DATA AREA

;ILLEGAL MODE BITS TEST DATA

;WRITE TAPE MARK COMMAND

;ERASE COMMAND

;WRITE TAPE MARK RETRY

;END OF DATA

;TAPE TIMER COUNTER STORAGE AREA

;TAPE TIMER COUNTER STORAGE AREA

;DELAY COUNTER

```

11226
11227
11228 110434      124      141      160 T28RIB: .ASCIZ 'Tape Position Not Correct, RIB Should Be Set'
11229 110511      122      145      163 T28PBP: .ASCIZ 'Residual Byte Counter Register (RBPCR) Not Correct'
11230 110574      124      123      123 T28RDF: .ASCIZ 'TSSR Incorrect After READ REVERSE Into TAPE MARK'
11231 110655      124      123      123 T28RDG: .ASCIZ 'TSSR Incorrect After SPACE Command Into TAPE MARK'
11232 110737      124      123      123 T28WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
11233 111014      111      154      154 T28LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
11234 111075      127      122      111 T28SSR: .ASCIZ 'WRITE MISCELLANEOUS Command Not Accepted'
11235 111146      124      123      123 T28WDE: .ASCIZ 'TSSR Not Correct After READ DATA Command, Into TAPE MARK'
11236 111237      124      141      160 T28BOT: .ASCIZ 'Tape Not At BOT After REWIND Command'
11237 111304      124      123      123 T28TM: .ASCIZ 'TSSR Not Correct After FORMAT Command Reject'
11238 111361      122      145      167 T28RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
11239 111430      104      162      151 T28OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
11240 111503      124      123      123 T28WDC: .ASCIZ 'TSSR Not Correct After WRITE TAPE MARK Command'
11241 111562      103      126      103 T28VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
11242 111635      124      115      113 T28TMK: .ASCIZ 'TMK Not Set After WRITE TAPE MARK Command'
11243 111707      124      115      113 T28RRM: .ASCIZ 'TMK Not Set After READ REVERSE Into TAPE MARK'
11244 111765      124      115      113 T28RRN: .ASCIZ 'TMK Not Set After SPACE REVERSE Into TAPE MARK'
11245 112044      124      115      113 T28RRP: .ASCIZ 'TMK Not Set After READ FORWARD Into TAPE MARK'
11246 112122      104      141      164 T28DTR: .ASCIZ 'Data Transferred On READ REVERSE Into A TAPE MARK'
11247 112204      104      141      164 T28DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
11248 112301      127      162      151 TST28ID: .ASCIZ 'Write/Read Tape Mark'
11249
11250
11251
11252
11253
11254
11255
11256
11257 112326
11258 112326
11259 112332 012701 110240'
11260 112336 012721 100004
11261 112342 012721 110250'
11262 112346 005021
11263 112350 012721 000012
11264 112354 012721 110262'
11265 112360 005021
11266 112362 012721 000024
11267 112366 005021
11268 112370 012711 000000
11269 112374 012702 000030
11270 112400 012762 177777 110262' 64$:
11271 112406 005742
11272 112410 020227 000000
11273 112414 001371
11274 112416 000207
11275
11276
11277 112420
11278 112420
11279 112424 012701 110350'
11280 112430 012721 100006
11281 112434 012721 110400'
11282 112440 005021

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

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

```

```

11283 112442 012721 000006          MOV    #6.,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
11284 112446 005021                  CLR    (R1)+          ;POINT TO DATA SEL AREA
11285 112450 012701 110400'         MOV    #T28BF2,R1
11286 112454 005021                  CLR    (R1)+
11287 112456 005011                  CLR    (R1)
11288 112460 000207                  RTS     PC             ;RETURN
11289 112462
11290 112462          T28RT3:      SAVREG
11291 112466 012701 110370'         MOV    #T28PK3,R1    ;GET PACKET ADDRESS
11292 112472 005021                  CLR    (R1)+          ;CLEAR COMMAND AREA
11293 112474 005021                  CLR    (R1)+          ;CLEAR ADDRESS AREA
11294 112476 005021                  CLR    (R1)+          ;CLEAR EXTENDED ADDRESS AREA
11295 112500 005011                  CLR    (R1)           ;SIZE OF DATA TRANSFER
11296 112502 000207                  RTS     PC             ;RETURN
11297 112504
11298 112504 104401                  TRAP   C#ETST        L10130:
11298 112506          ENDMOD

```

```

1          .TITLE  TSV6 - PARAMETER CODING
7
12
18
19 112506  BGNMOD  TSV6
112506  TSV6::
20
21
22          .SBTTL  HARDWARE PARAMETER CODING SECTION
23
24          ;**
25          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
26          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
27          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
28          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
29          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
30          ; WITH THE OPERATOR.
31          ;--
32 112506  BGNHRD
112506 000010  .WORD L10134-L$HARD/2
112510  L$HARD::
33
34 112510  GPRMA  HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
112510 000031  .WORD  T$CODE
112512 112530' .WORD  HPM1
112514 160010  .WORD  T$LLOLM
112516 177776  .WORD  T$HILIM
35 112520  GPRMA  HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
112520 001031  .WORD  T$CODE
112522 112564' .WORD  HPM2
112524 000000  .WORD  T$LLOLM
112526 000776  .WORD  T$HILIM
36          ;GPRMD  HPM3,4,0,340,0,7,YES        ;GET INTERRUPT PRIORITY.
37 112530  ENDHRD
          .EVEN
          L10134:
38 112530      104      105      126  HPM1:  .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
39 112564      111      116      124  HPM2:  .ASCIZ  'INTERRUPT VECTOR '
40 112610      111      116      124  HPM3:  .ASCIZ  'INTERRUPT PRIORITY '
41          .EVEN
42
43          .SBTTL  SOFTWARE PARAMETER CODING SECTION
44
45          ;**
46          ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
47          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
48          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
49          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
50          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
51          ; WITH THE OPERATOR.
52          ;--
53 112640  BGNSFT
112640 000003  .WORD L10135-L$SOFT/2
112642  L$SOFT::
54          ; GPRML  SPM1,0,-1,YES                ; GET TRANSPORT TEST FLAG.
55 112642  GPRML  SPM4,2,-1,YES                ; GET ITERATION CONTROL.
112642 001130  .WORD  T$CODE

```

```

112644 112700'
112646 177777
56
57
58 112650
112650
59
60
61 112650 105 116 101 SPM1: .ASCIZ 'ENABLE TRANSPORT TESTS '
62 112700 111 116 110 SPM4: .ASCIZ 'INHIBIT ITERATIONS '
63 ;SPM6: .ASCIZ 'PER TEST ERROR LIMIT '
64 ;SPM7: .ASCIZ 'PER UNIT ERROR LIMIT '
65 .SBTTL PATCH AREA
66
67
68 ; FINALLY A GENEROUS PATCH AREA.
69
70 ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
71 ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
72
73
74 112730 PATCH::
75
76 ; .BLKW 32.
77 112730 ; .BLKW 1.
78
79 ; .IF NZ,..&377
80 ; =.!377*1
81 ; .ENDC
82 112732 LASTAD ;SET LAST USED ADDRESS.
112732 000000 .EVEN
112734 000000 .WORD 0
112736 L$LAST:: .WORD 0
83 112736 ENDMOD
84 .SBTTL HARD CODED P-TABLE
85
86 ;++
87 ;
88 112736 BGNSETUP 1
89 112736 BGNPTAB
112736 000000 .WORD 0
112740 000003 .WORD L10140-./2-1
112742 L10136:
90 112742 172522 .WORD 172522
91 112744 000224 .WORD 224
92 112746 000240 .WORD PRI05
93 112750 ENDPTAB
112750 L10140:
94 112750 ENDSETUP
95
96 000001 .END

```

ADDSSR	012026RG	002	C\$AU	=	000052	DEVDR0	023236R	002	FREE	003114RG	002	INCERK	016746R	002			
ADR	=	000020	G	C\$AUTO	=	000061	DEVNRD	023155R	002	FREEHI	003120R	002	INTCPC	016050R	002		
AMBTSS	006535R	002	C\$BRK	=	000022	DEVNXR	023073R	002	FRESIZ	003116RG	002	INTFLA	016045R	002			
ASSEMB	=	000010	C\$BSEG	=	000004	DEVONL	023023R	002	FUSI	004107R	002	INTMAS	016044R	002			
A1716	=	000003	C\$BSUB	=	000002	DEVSUM	022766R	002	F\$AU	=	000015	INTR	016116RG	002			
BADDAT	003146RG	002	C\$CEFG	=	000045	DFPTBL	002146RG	002	F\$AUTO	=	000020	INTREC	002214RG	002			
BADSSR	015600RG	002	C\$CLCK	=	000062	DIAGMC	=	000000	F\$BGN	=	000040	INTVEC	016046R	002			
BDVPCR	=	177520	G	C\$CLEA	=	000012	DICEC	=	000001	F\$CLEA	=	000007	INTX	004270R	002		
BENBSW	002220RG	002	C\$CLOS	=	000035	DSBINT	016104R	002	F\$DU	=	000016	INVERT	021024RG	002			
BIE	=	040000	C\$CLP1	=	000006	DUAD12	004633R	002	F\$END	=	000041	IOKCKI	=	000200			
BIT0	=	000001	G	C\$CVEC	=	000036	DUFLG	003102RG	002	F\$HARD	=	000004	IOKSTP	=	000001		
BIT00	=	000001	G	C\$DCLN	=	000044	DUMMY	003052R	002	F\$HW	=	000013	IPRI	002202RG	002		
BIT01	=	000002	G	C\$DODU	=	000051	EF.CON	=	000036	G	F\$INIT	=	000006	ISR	=	000100	G
BIT02	=	000004	G	C\$DRPT	=	000024	EF.NEW	=	000035	G	F\$JMP	=	000050	IVEC	002200RG	002	
BIT03	=	000010	G	C\$DU	=	000053	EF.PWR	=	000034	G	F\$MOD	=	000000	IXE	=	004000	G
BIT04	=	000020	G	C\$EDIT	=	000003	EF.RES	=	000037	G	F\$MSG	=	000011	I\$AU	=	000041	
BIT05	=	000040	G	C\$ERDF	=	000055	EF.STA	=	000040	G	F\$PROT	=	000021	I\$AUTO	=	000041	
BIT06	=	000100	G	C\$ERHR	=	000056	EMAXDU	016701R	002	F\$PWR	=	000017	I\$CLN	=	000041		
BIT07	=	000200	G	C\$ERRO	=	000060	EN	=	000000	F\$RPT	=	000012	I\$DU	=	000041		
BIT08	=	000400	G	C\$ERSF	=	000054	ENAINI	016052R	002	F\$SEG	=	000003	I\$HRD	=	000041		
BIT09	=	001000	G	C\$ERSO	=	000057	ENVIRN	020540R	002	F\$SOFT	=	000005	I\$INIT	=	000041		
BIT1	=	000002	G	C\$ESCA	=	000010	EPRTSW	002170RG	002	F\$SRV	=	000010	I\$MOD	=	000041		
BIT10	=	002000	G	C\$ESEG	=	000005	EPRT1	006166R	002	F\$SUB	=	000002	I\$MSG	=	000041		
BIT11	=	004000	G	C\$ESUB	=	000003	EPRT2	006256R	002	F\$SW	=	000014	I\$PROT	=	000040		
BIT12	=	010000	G	C\$ETST	=	000001	ERCM	011633R	002	F\$TEST	=	000001	I\$PTAB	=	000041		
BIT13	=	020000	G	C\$EXIT	=	000032	ERRHI	002226RG	002	GDDAT	003150RG	002	I\$PWR	=	000041		
BIT14	=	040000	G	C\$GETB	=	000026	ERRK	016660R	002	GERRMA	002164RG	002	I\$RPT	=	000041		
BIT15	=	100000	G	C\$GETW	=	000027	ERRLO	002230RG	002	GETPAT	020104RG	002	I\$SEG	=	000041		
BIT2	=	000004	G	C\$GMAN	=	000043	ERRNO	=	001513	GETSEL	020166RG	002	I\$SETU	=	000041		
BIT3	=	000010	G	C\$GPHR	=	000042	ERRVEC	=	000004	G	G\$CNT0	=	000200	I\$SFT	=	000041	
BIT4	=	000020	G	C\$GPLO	=	000030	ERTABE	003366R	002	G\$DELM	=	000372	I\$SRV	=	000041		
BIT5	=	000040	G	C\$GPRI	=	000040	ERTABL	003166R	002	G\$DISP	=	000003	I\$SUB	=	000041		
BIT6	=	000100	G	C\$INIT	=	000011	ESUM	016662R	002	G\$EXCP	=	000400	I\$TST	=	000041		
BIT7	=	000200	G	C\$INLP	=	000020	EVL	=	000004	G	G\$HILI	=	000002	J\$JMP	=	000167	
BIT8	=	000400	G	C\$MANI	=	000050	EXBCNT	=	000010	G\$LOLI	=	000001	KIPAR0	=	172340		
BIT9	=	001000	G	C\$MEM	=	000031	EXIT	034124R	002	G\$NO	=	000000	KIPAR1	=	172342		
BOE	=	000400	G	C\$MSG	=	000023	EXPBRE	015402RG	002	G\$OFFS	=	000400	KIPAR2	=	172344		
BRINIT	004447R	002	C\$OPEN	=	000034	EXPD	002222RG	002	G\$OF SI	=	000376	KIPAR3	=	172346			
BSELO	=	000000	C\$PNTB	=	000014	EXPGOT	004523R	002	G\$PRMA	=	000001	KIPAR4	=	172350			
BSEL1	=	000001	C\$PNTF	=	000017	EXPGT2	004557R	002	G\$PRMD	=	000002	KIPAR5	=	172352			
CHKAMB	015744R	002	C\$PNTS	=	000016	EXPMSG	002312RG	002	G\$PRML	=	000000	KIPAR6	=	172354			
CHKMAN	020410RG	002	C\$PNTX	=	000015	EXPREC	015374RG	002	G\$RADA	=	000140	KIPAR7	=	172356			
CHKTSS	016236R	002	C\$QIO	=	000377	EXTA	005600R	002	G\$RADB	=	000000	KIPDR0	=	172300			
CKDROP	017104R	002	C\$RDBU	=	000007	EXTEND	005576R	002	G\$RADD	=	000040	KIPDR1	=	172302			
CKEMAX	017004R	002	C\$REFG	=	000047	EXTFEA	002216RG	002	G\$RADL	=	000120	KIPDR2	=	172304			
CKMSG	011260RG	002	C\$RESE	=	000033	E\$END	=	002100	G\$RADO	=	000020	KIPDR3	=	172306			
CKMSG2	011400RG	002	C\$REVI	=	000003	E\$LOAD	=	000035	G\$XFER	=	000004	KIPDR4	=	172310			
CKRAM	011014RG	002	C\$RFLA	=	000021	FATAL	034224R	002	G\$YES	=	000010	KIPDR5	=	172312			
CKRAM2	011124RG	002	C\$RPT	=	000025	FATERR	=	000060	HIADDR	=	001400	KIPDR6	=	172314			
CMDPKT	021100RG	002	C\$SEFG	=	000046	FATFLG	002212RG	002	HOE	=	100000	G	KIPDR7	=	172316		
CMPMEM	017570R	002	C\$SPRI	=	000041	FERCM	011622R	002	HPM1	=	112530R	002	KTENAB	003124RG	002		
CONFIG	017152R	002	C\$SVEC	=	000037	FIFEXP	012070RG	002	HPM2	=	112564R	002	KTFLG	003122RG	002		
COUNT	002300RG	002	C\$TPRI	=	000013	FIF1MS	012142R	002	HPM3	=	112610R	002	KTINIT	020626R	002		
CSRADD	002176RG	002	DATA	002302RG	002	FIF2MS	012211R	002	IBE	=	010000	G	KTOFF	017176R	002		
CTAB	003154RG	002	DATASC	020142R	002	FILLME	017324R	002	IDU	=	000040	G	KTON	017160R	002		
CTABE	003166RG	002	DEBUGM	011532R	002	FNOINT	004205R	002	IER	=	020000	G	LERRMA	002162RG	002		
CTABM	003154RG	002	DEVcnt	002210RG	002	FORCER	002166RG	002	IFault	004246R	002	LISTAL	=	000001			

P.GETS= 000017	SPM4 112700R	002 TSSDEF 006506R	002 T8TSTM= 177777	T22RWJ 026474R	002
P.IE = 000207	SR0 = 177572	TSSR = 000002 G	T8TSTS= 000001	T22SSR 026130R	002
P.INIT= 000013	SR1 = 177574	TSSRBI 003472RG	002 T88AU = 010031	T22S2 026112R	002
P.MODE= 007400	SR2 = 177576	TSSRFO 006315R	002 T88AUT= 010033	T22S3 026114R	002
P.OPP = 020000	SR3 = 172516	TSSRM = 000003 G	T88CLE= 010034	T22TM 026400R	002
P.POSI= 000010	SSR = 000200	TSSX 004010R	002 T88DAT= 010140	T22VCK 026547R	002
P.READ= 000001	STATCO 012312R	002 TSTBLK 002742RG	002 T88DU = 010032	T22WLK 026622R	002
P.SWB = 010000	SVCGBL= 000000	TSTCNT 002204RG	002 T88HAR= 010134	T22WRT 026120R	002
P.WRIT= 000005	SVCINS= 000000	TSTEND 016622R	002 T88HW = 010000	T23A 003134RG	002
P.WRTC= 000004	SVCSUB= 000001	TSTFLA 002304RG	002 T88INI= 010030	T23AM3 033140R	002
P.WRTS= 000006	SVCTAG= 000000	TSTL00 016360RG	002 T88MSG= 010025	T23B 003136RG	002
QVP 002174RG	002 SVCTST= 000001	TSTPTR 002306RG	002 T88PC = 000001	T23BA 033525R	002
RAMASC 014056R	002 S8LSYM= 010000	TSTSET 016412RG	002 T88PRO= 010027	T23BFR 032402R	002
RANDAT 002232RG	002 S0.IDB= 000010	TST21I 024244R	002 T88PTA= 010137	T23BF2 032522R	002
RAMERR 015410RG	002 S0.IFB= 000002	TST22I 026707R	002 T88RPT= 010035	T23BS0 032522R	002
RAMEXP 015430RG	002 S0.IFP= 000001	TST23I 033666R	002 T88SOF= 010135	T23BS1 032523R	002
RAMFOR 010016R	002 S0.ILD= 000020	TST24I 046252R	002 T88SRV= 010026	T23CHK 034062R	002
RAMSIZ 002272RG	002 S0.ION= 000040	TST25I 055230R	002 T88SUB= 010133	T23CON 032540R	002
RAMTAD 015416RG	002 S0.IRD= 000100	TST26I 074607R	002 T88SW = 010001	T23DAT 032370R	002
RCVHIA 002274RG	002 S0.IRW= 000004	TST27I 104643R	002 T88TES= 010130	T23DSW 032400R	002
RCVLOA 002276RG	002 S0.ISP= 000200	TST28I 112301R	002 T1 023306RG	002 T23EOT 032664R	002
RDERR 005176R	002 S1.ICE= 002000	TSV2 002000RG	002 T2 024422RG	002 T23ET 032577R	002
RECMG 002456RG	002 S1.IEO= 010000	TSV3 002166RG	002 T2.1 024452R	002 T23L00 027146R	002
RECV 002224RG	002 S1.IFM= 001000	TSV4 021406RG	002 T2.2 025044R	002 T23OFL 033206R	002
REGSAV 020050R	002 S1.IHE= 000400	TSV6 112506RG	002 T2.3 025366R	002 T23PAC 032360R	002
RETRR 005362R	002 S1.IID= 004000	TSV7 023306RG	002 T21AM3 024123R	002 T23PK2 032470R	002
RETRY 034126R	002 S1.I1R= 020000	TTIBFR= 177562 G	T21BFR 023724R	002 T23PK3 032510R	002
REWIND 010714RG	002 S1.I2R= 040000	TTICSR= 177560 G	T21BF2 024020R	002 T23RES 033702R	002
RMCHBE= 000167	S1.PAR= 100000	TTIVEC= 000060 G	T21BS0 024020R	002 T23RNC 033065R	002
RMCHEN= 000200	S2.ATI= 000010	T8ARGC= 000003	T21BS1 024021R	002 T23RSZ 032520R	002
RMPMSGB= 000215	S2.BTI= 000004	T8CODE= 001130	T21DAT 023710R	002 T23RT2 033774R	002
RMPMSGE= 000234	S2.DIM= 000200	T8ERRN= 001513	T21DLY 023722R	002 T23RT3 034036R	002
RMPKTB= 000201	S2.ILW= 000100	T8EXCP= 000000	T21DSW 023720R	002 T23RWN 033016R	002
RMPKTE= 000210	S2.INR= 000020	T8FLAG= 000040	T21L00 023336R	002 T23SSR 032544R	002
RMR = 010000	S2.OUT= 000040	T8FREE= 112750R	002 T21OFL 024223R	002 T23S2 032516R	002
RMPACK 011010R	002 S2.UND= 000003	T8GMAN= 000000	T21PAC 023700R	002 T23S2 032524R	002
SC = 100000	TBLEND= 003052RG	002 T8HILI= 000776	T21PK2 024010R	002 T23S3 032526R	002
SCE = 020000	TCOASC 006376R	002 T8LAST= 000001	T21RES 024266R	002 T23TM 032742R	002
SCHERR 005270R	002 TCOCOD 006576R	002 T8LOLI= 000000	T21RT2 024356R	002 T23TMP 032530R	002
SCME 005003R	002 TEMP1 003106RG	002 T8LSYM= 010000	T21SSR 024026R	002 T23VCK 033452R	002
SDELAY 010560R	002 TEMP2 003110RG	002 T8LTND= 000010	T21S2 024022R	002 T23WB 032512R	002
SELASC 020352R	002 TERCLS= 000016	T8NEST= 177777	T21S3 024024R	002 T23WD 032534R	002
SELDAT= 000004	TESTNO= 000010	T8NS0 = 000000	T22AM3 026225R	002 T23WDC 033350R	002
SEL2 = 000002	TEXASC 006335R	002 T8NS1 = 000005	T22BFR 026012R	002 T23WDD 033261R	002
SETMAP 017220R	002 TFCASC 006437R	002 T8NS2 = 000002	T22BF2 026110R	002 T23WDR 032536R	002
SETU 021746R	002 TIMEXP 015452RG	002 T8PCNT= 000000	T22BS0 026110R	002 T23WRT 032532R	002
SFFMSG 012022RG	002 TIMSG0 015500R	002 T8PTAB= 010137	T22BS1 026111R	002 T23WSS 033577R	002
SFHERR 003675R	002 TINERR 011721R	002 T8PTHV= 000001	T22DAT 026000R	002 T24AM3 045240R	002
SFIERR 003642R	002 TMPBFR 002622RG	002 T8PTNU= 000001	T22FOR 026124R	002 T24BA 045572R	002
SFIMSG 011734RG	002 TNAM 016606R	002 T8SAVL= 177777	T22L00 024452R	002 T24BFR 043652R	002
SFPTBL 002156RG	002 TRANST 002156RG	002 T8SEGL= 177777	T22OFL 026325R	002 T24BF2 043770R	002
SIFLAG 003144RG	002 TSBA = 000000 G	T8SIZE= 000005	T22PAC 025770R	002 T24BOT 044633R	002
SIMSG 011666R	002 TSBAH = 000001 G	T8SUBN= 000003	T22PK2 026100R	002 T24BS0 043770R	002
SKIPT 003370R	002 TSDB = 000000 G	T8TAGL= 177777	T22POS 026122R	002 T24BS1 043771R	002
SOFINI 015674RG	002 TSDBH = 000001 G	T8TAGN= 010141	T22RD 026116R	002 T24CON 044002R	002
SPACE 010366RG	002 TSDBM = 000001 G	002 T8TEMP= 000000	T22RES 026742R	002 T24DAT 043640R	002
SPM1 112650R	002 TSFCOD 007136R	T8TEST= 000010	T22RT2 027034R	002 T24DLY 044006R	002
	002 TSREJ = 000006				

T24DSW	043650R	002	T25RB	054032R	002	T26SZ	072036R	002	T27TSA	104404R	002	T3BFLG	003140RG	002
T24DTA	044700R	002	T25RES	055246R	002	T26S2	072042R	002	T27VCK	103474R	002	T3.1	027146R	002
T24EOT	044766R	002	T25RIB	054753R	002	T26S3	072044R	002	T27MB	101662R	002	T3.2	027520R	002
T24ILA	044362R	002	T25RN	054046R	002	T26TM	073307R	002	T27WDC	103421R	002	T3.3	030370R	002
T24LON	045732R	002	T25RT2	055340R	002	T26TRL	074442R	002	T27WDD	103331R	002	T3.4	031210R	002
T24LOO	034312R	002	T25RT3	055402R	002	T26VCK	073773R	002	T27WDE	102472R	002	T3.5	031424R	002
T24LOP	046014R	002	T25RWN	055035R	002	T26WB	072032R	002	T27WDF	102300R	002	T3.6	031772R	002
T24LOQ	044446R	002	T25SSR	054064R	002	T26WDC	073720R	002	T27WDR	101700R	002	T4	034252RG	002
T24LOR	044062R	002	T25S2	054036R	002	T26WDD	073630R	002	T27WNG	101714R	002	T4.1	034312R	002
T24NEF	044010R	002	T25S2	054042R	002	T26WDE	073023R	002	T27WRF	104466R	002	T4.10	041622R	002
T24NXM	044221R	002	T25S3	054044R	002	T26WDF	072631R	002	T27WSS	103622R	002	T4.11	042066R	002
T24OFL	045305R	002	T25TM	054272R	002	T26WNG	072066R	002	T28BFR	110262R	002	T4.12	042340R	002
T24PAC	043630R	002	T25WB	054032R	002	T26WSS	074121R	002	T28BF2	110400R	002	T4.13	042624R	002
T24PBP	046076R	002	T25WDC	055157R	002	T27AM3	103207R	002	T28BOT	111237R	002	T4.14	043124R	002
T24PK2	043740R	002	T25WDE	054145R	002	T27BA	103547R	002	T28BS0	110400R	002	T4.2	034734R	002
T24PK3	043760R	002	T25WDR	054050R	002	T27BFR	101552R	002	T28BS1	110401R	002	T4.3	035514R	002
T24RB	043762R	002	T25WNG	054435R	002	T27BF2	101670R	002	T28CNT	110426R	002	T4.4	036270R	002
T24RES	046320R	002	T25WNH	054610R	002	T27BOT	102561R	002	T28CNU	110430R	002	T4.5	036772R	002
T24RN	043776R	002	T26AM3	073506R	002	T27BS0	101670R	002	T28CON	110422R	002	T4.6	037436R	002
T24RNC	045165R	002	T26BA	074046R	002	T27BS1	101671R	002	T28DAT	110250R	002	T4.7	040072R	002
T24RT2	046412R	002	T26BFR	071722R	002	T27CNT	101706R	002	T28DLY	110432R	002	T4.8	040526R	002
T24RT3	046454R	002	T26BF2	072040R	002	T27CNU	101710R	002	T28DSW	110260R	002	T4.9	041120R	002
T24RWN	045116R	002	T26BOT	073075R	002	T27CON	101702R	002	T28DTA	112204R	002	T5	046506RG	002
T24SSR	044527R	002	T26BS0	072040R	002	T27DAT	101540R	002	T28DTR	112122R	002	T5.1	046536R	002
T24SZ	043766R	002	T26BS1	072041R	002	T27DLY	101712R	002	T28IMV	110406R	002	T5.2	047512R	002
T24S2	043772R	002	T26CNT	072056R	002	T27DSW	101550R	002	T28LOO	105100R	002	T5.3	050112R	002
T24S3	043774R	002	T26CNU	072060R	002	T27DTA	104546R	002	T28LOQ	111014R	002	T5.4	050566R	002
T24TM	045043R	002	T26DAT	071710R	002	T27EOT	102731R	002	T28OFL	111430R	002	T5.5	051232R	002
T24TRL	046164R	002	T26DLY	072064R	002	T27LON	103711R	002	T28PAC	110240R	002	T5.6	051772R	002
T24VCK	045517R	002	T26DSW	071720R	002	T27LOO	075046R	002	T28PBP	110511R	002	T5.7	052732R	002
T24WB	043762R	002	T26DTA	073142R	002	T27LOP	103773R	002	T28PK2	110350R	002	T5.8	053252R	002
T24WDC	045446R	002	T26EOT	073230R	002	T27LOQ	102355R	002	T28PK3	110370R	002	T6	055434RG	002
T24WDD	045360R	002	T26LON	074210R	002	T27LOR	102230R	002	T28RB	110372R	002	T6.1	055474R	002
T24WDE	044561R	002	T26LOO	055474R	002	T27NEF	104231R	002	T28RDF	110574R	002	T6.10	065114R	002
T24WDF	044305R	002	T26LOP	074272R	002	T27OFL	103256R	002	T28RDG	110655R	002	T6.11	065770R	002
T24WDG	044132R	002	T26LOQ	072706R	002	T27PAC	101530R	002	T28RES	112326R	002	T6.12	066640R	002
T24WDR	044000R	002	T26LOR	072561R	002	T27PBP	104055R	002	T28RIB	110434R	002	T6.13	067572R	002
T24WSS	045643R	002	T26NEF	072154R	002	T27PK2	101640R	002	T28RN	110416R	002	T6.14	070622R	002
T25BFR	053722R	002	T26NEQ	074530R	002	T27PK3	101660R	002	T28RRM	111707R	002	T6.15	071202R	002
T25BF2	054040R	002	T26OFL	073555R	002	T27RB	101662R	002	T28RRN	111765R	002	T6.2	056406R	002
T25BNC	054520R	002	T26PAC	071700R	002	T27RDF	102002R	002	T28RAP	112044R	002	T6.3	057254R	002
T25BOT	054225R	002	T26PBP	074354R	002	T27RES	104664R	002	T28RT2	112420R	002	T6.4	060146R	002
T25BS0	054040R	002	T26PK2	072010R	002	T27RN	101676R	002	T28RT3	112462R	002	T6.5	061074R	002
T25BS1	054041R	002	T26PK3	072030R	002	T27RNC	103134R	002	T28RWN	111361R	002	T6.6	061652R	002
T25CNT	054060R	002	T26RB	072032R	002	T27RAF	102051R	002	T28SSR	111075R	002	T6.7	062514R	002
T25CN2	054056R	002	T26RDF	072236R	002	T27RT2	104756R	002	T28SZ	110376R	002	T6.8	063366R	002
T25CON	054052R	002	T26RES	074620R	002	T27RT3	105020R	002	T28S2	110402R	002	T6.9	064240R	002
T25DAT	053710R	002	T26RN	072046R	002	T27RWN	103065R	002	T28S3	110404R	002	T7	075006RG	002
T25DLY	054062R	002	T26RNC	073433R	002	T27SC	102146R	002	T28TM	111304R	002	T7.1	075046R	002
T25DSW	053720R	002	T26RRF	072305R	002	T27SCF	104327R	002	T28TMK	111635R	002	T7.2	075444R	002
T25LOO	046536R	002	T26RRG	072402R	002	T27SSR	102436R	002	T28VCK	111562R	002	T7.3	076226R	002
T25NEF	054673R	002	T26RSZ	072062R	002	T27SZ	101666R	002	T28WB	110372R	002	T7.4	077050R	002
T25NET	054361R	002	T26RT2	074712R	002	T27S2	101672R	002	T28WDC	111503R	002	T7.5	077752R	002
T25OFL	055104R	002	T26RT3	074754R	002	T27S3	101674R	002	T28WDE	111146R	002	T8	105044RG	002
T25PAC	053700R	002	T26RWN	073364R	002	T27TIM	102654R	002	T28WDF	110737R	002	T8.1	105100R	002
T25PK2	054010R	002	T26SC	072477R	002	T27TM	103010R	002	T28WDR	110420R	002	T8.2	105460R	002
T25PK3	054030R	002	T26SSR	072767R	002	T27TRL	104143R	002	T3	027102RG	002	T8.3	105740R	002

TSV6 - PARAMETER CODING MACRO M1113 01-FEB-84 17:54
 SYMBOL TABLE

SEQ 314

UAM = 000200 G	WF.IWF = 000020	XSOBOT = 000002	XXCOMM 003112RG	002 X2.SPA = 035400
UNITN 002172RG	002 WF.IWR = 000100	XSOEOT = 000001	X#ALWA = 000000	X2.UNI = 000007
UNREC = 000006	WF.I3R = 000002	XSOIE = 000040	X#FALS = 000040	X2.WCF = 002000
USI 004113R	002 WF.I4R = 000001	XSOILA = 000400	X#OFFS = 000400	X3.DCK = 000010
WAITF 016150RG	002 WRTCHR 010562RG	002 XSOILC = 001000	X#TRUE = 000020	X3.MBZ = 000006
WC.IFA = 000200	WRTERR 005103R	002 XSOLET = 020000	X1.COR = 020000	X3.MDE = 177400
WC.IFE = 000002	WRTMSG 005046R	002 XSOMOT = 000200	X1.DLT = 100000	X3.OPI = 000100
WC.IGO = 000001	WSMBK 021110RG	002 XSONEF = 002000	X1.MBZ = 017375	X3.REV = 000040
WC.IRE = 000010	XFERAS 015640R	002 XSOONL = 000100	X1.RBP = 000400	X3.RIB = 000001
WC.IRW = 000004	XNXM 016276R	002 XSOPED = 000010	X1.SPA = 040000	X3.SPA = 000200
WC.IOT = 000100	XORBFO 007574R	002 XSORLL = 010000	X1.UNC = 000002	X3.TRF = 000020
WC.I1T = 000040	XORFOR 007712R	002 XSORLS = 040000	X2.BUF = 000100	X4.HSP = 100000
WC.I5R = 000020	XST0 = 000006 G	XSOTMK = 100000	X2.EXT = 000200	X4.MBZ = 017400
WF.IED = 000010	XST1 = 000010 G	XSOVCK = 000020	X2.OPM = 100000	X4.RCE = 040000
WF.IER = 000004	XST2 = 000012 G	XSOWLE = 004000	X2.RCE = 040000	X4.TSM = 020000
WF.IMI = 000200	XST3 = 000014 G	XSOWLK = 000004	X2.REV = 000077	X1.WRC = 000377
WF.IRE = 000040	XST4 = 000016 G			
. ABS. 000000 000				
000000 001				
ABS 112750 002				
ERRORS DETECTED: 0				

VIRTUAL MEMORY USED: 31628 WORDS (124 PAGES)
 DYNAMIC MEMORY: 20614 WORDS (79 PAGES)
 ELAPSED TIME: 01:00:28
 CZTSCA,CZTSCA,SEQ/-SP=SVC/ML,TSV1C,TSV22C,TSV3B,TSV4,TSV7A,TSV6