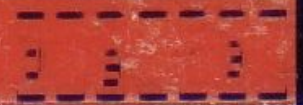


TS11

TS11 DATA RELIAB  
CZTSHCO

AH E455C MC  
FICHE 1 OF 1

MAR 98  
COPYR  
MADE IN USA



A large grid of approximately 15 columns and 20 rows of small, illegible data tables. Each cell contains a small table with multiple columns and rows of text, likely representing individual data points or small reports. The text is too small to be read.





IDENTIFICATION

PRODUCT CODE: AC-E454C-MC  
PRODUCT NAME: CZTSHCO TS11 DATA RELIAB  
PRODUCT DATE: 01-OCT-79  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: ROBERT F. WERY

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) 1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	



## USER DOCUMENTATION

USER DOCUMENTATION TABLE OF CONTENTS

## GLOSSARY

## 1.0 GENERAL INFORMATION

## 1.1 PROGRAM ABSTRACT

- 1.1.1 FUNCTIONAL DESCRIPTION
- 1.1.2 STRUCTURE OF PROGRAM
- 1.1.3 MEMORY MAP
- 1.1.4 DIAGNOSTIC INFORMATION
  - 1.1.4.1 SCOPE
  - 1.1.4.2 ERROR RECOVERY
  - 1.1.4.3 WRITE ERROR RECOVERY
    - 1.1.4.3.1 MEDIA/OPERATIONAL  
SELECTIVE WRITE-ERROR-RECOVERY
    - 1.1.4.3.2 OPERATIONAL WRITE-ERROR-RECOVERY
  - 1.1.4.4 DIAGNOSTIC TIMING ADJUSTMENT

## 1.2 SYSTEM REQUIREMENTS

- 1.2.1 HARDWARE REQUIREMENTS
- 1.2.2 SOFTWARE REQUIREMENTS

## 1.3 RELATED DOCUMENTS AND STANDARDS

## 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

## 1.5 ASSUMPTIONS

## 1.6 DIAGNOSTIC HISTORY

## 2.0 OPERATING INSTRUCTIONS

## 2.1 HARDWARE PARAMETERS

## 2.2 SOFTWARE PARAMETERS

- 2.2.1 TS04 COMMAND LIST
- 2.2.2 DATA PATTERNS

## 2.3 EXAMPLES OF SOFTWARE PARAMETER DIALOGUE

- 2.3.1 BASIC FUNCTION AND DATA RELIABILITY  
WITH ALL ERROR REPORTING ENABLED
- 2.3.2 SCOPE LOOP SET UP IN BASIC FUNCTIONS
- 2.3.3 SCOPE LOOP SET UP IN DATA RELIABILITY



## 2.4 EXECUTION TIMES

- 2.4.1 SYSTEM CONFIGURATION
- 2.4.2 TEST EXECUTION TIMES

## 3.0 ERROR INFORMATION

## 3.1 ERROR REPORTING

- 3.1.1 ERROR #1 - COMMAND PACKET ADDRESS IS NOT ON A MODULO 4 BOUNDRY
- 3.1.2 ERROR #2 - TS04 NOT READY
- 3.1.3 ERROR #3 - NO RESPONSE ERRORS
- 3.1.4 ERROR #4 - NO INTERRUPT ERROR
- 3.1.5 SPECIAL CONDITION ERRORS
  - 3.1.5.1 ERROR #5 - TCC0, UNDEFINED SPECIAL CONDITION
  - 3.1.5.2 ERROR #6 - TCC1, ATTENTION CONDITION
  - 3.1.5.3 ERROR #7 - TCC2, TAPE STATUS ALERT
  - 3.1.5.4 ERROR #8 - TCC3, FUNCTION REJECT
  - 3.1.5.5 ERROR #9 - TCC4, RECOVERABLE ERROR
  - 3.1.5.6 ERROR #10- TCC5, RECOVERABLE ERROR
  - 3.1.5.7 ERROR #11- TCC6, UNRECOVERABLE ERROR
  - 3.1.5.8 ERROR #12- TCC7, FATAL SUBSYSTEM ERROR
- 3.1.6 ERROR #13 - RFC NON-ZERO ERROR
- 3.1.7 ERROR #14 - RETRY LIMIT EXCEEDED
- 3.1.8 ERROR #15 - TOO MANY INTERRUPTS
- 3.1.9 ERROR #16 - CAPSTAN RUNAWAY
- 3.1.10 ERROR #17 - DATA COMPARE ERRORS

## 3.2 ERROR HALTS

## 4.0 PERFORMANCE REPORT

## 5.0 TEST SUMMARIES

- 5.1 TEST 1 - BASIC FUNCTIONS
- 5.2 TEST 2 - DATA RELIABILITY
- 5.3 TEST 3 - WRITE COMPATABILITY/WRITE UTILITY
- 5.4 TEST 4 - READ COMPATABILITY/READ UTILITY
- 5.5 TEST 5 - EXECUTE OPERATOR SELECTED COMMAND SEQUENCE



## 6.0 DEVICE INFORMATION

- 6.1 GENERAL
- 6.2 UNIBUS INTERFACE SPECIFICATIONS
- 6.3 BIT DEFINITIONS FOR TS11/TS04 REGISTERS
  - 6.3.1 TS11/TS04 REGISTER SUMMARY
  - 6.3.2 TS11 STATUS REGISTER (TSSR)
  - 6.3.3 EXTENDED STATUS REGISTER 0 (XSTAT0)
  - 6.3.4 EXTENDED STATUS REGISTER 1 (XSTAT1)
  - 6.3.5 EXTENDED STATUS REGISTER 2 (XSTAT2)
  - 6.3.6 EXTENDED STATUS REGISTER 3 (XSTAT3)



## GLOSSARY

ACT	AUTOMATED COMPUTER TEST SYSTEM
APT	AUTOMATED PRODUCT TEST SYSTEM
BYTE/RECORD/FILE COUNT BRF	IS STORED IN THE 4TH WORD OF THE COMMAND PACKET AND IT'S USE BY THE TS04 DEPENDS ON THE TYPE OF COMMAND.
CMD	TS04 COMMAND (SEE 2.3.14.1 FOR LIST OF COMMANDS)
COMMAND PACKET CMDPKT	FOUR WORD PACKET IN THE CPU MEMORY WHICH CONTAINS ALL INFORMATION NEEDED BY THE TS04 TO EXECUTE A COMMAND.
EXTENDED STATUS	FOUR WORDS OF TS04 STATUS WHICH ARE TRANSFERRED AS PART OF THE MESSAGE PACKET AT THE COMPLETION OF A COMMAND.
MESSAGE PACKET	SEVEN WORD PACKET IN THE CPU MEMORY INTO WHICH THE TS04 STORES STATUS AT THE COMPLETION OF A COMMAND.
PC	PROGRAM COUNTER
PSW	PROCESSOR STATUS WORD
RESIDUAL FRAME COUNT RFC	THIS COUNT IS PART OF THE MESSAGE PACKET AND CONTAINS THE NUMBER OF BYTES/RECORDS /FILES REMAINING TO BE PROCESSED AT THE COMPLETION OF A COMMAND.
SPECIAL CONDITION SPEC COND	TSS4 BIT15. WHEN SET, INDICATES THAT THE LAST COMMAND DID NOT COMPLETE WITHOUT INCIDENT.
TERMINATION CLASS CODE TCC	THREE BIT CODE IN THE TSSR WHICH INDICATES THE TYPE OF COMMAND TERMINATION.
TSBA	TAPE SYSTEM BUS ADDRESS REGISTER.
TSDB	TAPE SYSTEM DATA BUFFER REGISTER.
TSSR	TAPE SYSTEM STATUS REGISTER.
XST0	EXTENDED STATUS REGISTER 0
XST1	EXTENDED STATUS REGISTER 1
XST2	EXTENDED STATUS REGISTER 2
XST3	EXTENDED STATUS REGISTER 3
XXDP+	XXDP+ IS A "CATCH-ALL" NAME FOR A GROUP OF PDP-11 DIAGNOSTIC PACKAGES AVAILABLE ON MULTIMEDIA.



## 1.0 GENERAL INFORMATION

---

### 1.1 PROGRAM ABSTRACT

---

#### 1.1.1 FUNCTIONAL DESCRIPTION

---

THIS PROGRAM CAN BE USED AS A BASIC FUNCTION TEST, A DATA RELIABILITY TEST, A COMPATABILITY TEST, OR TO EXECUTE A SEQUENCE OF OPERATOR SELECTED COMMANDS.

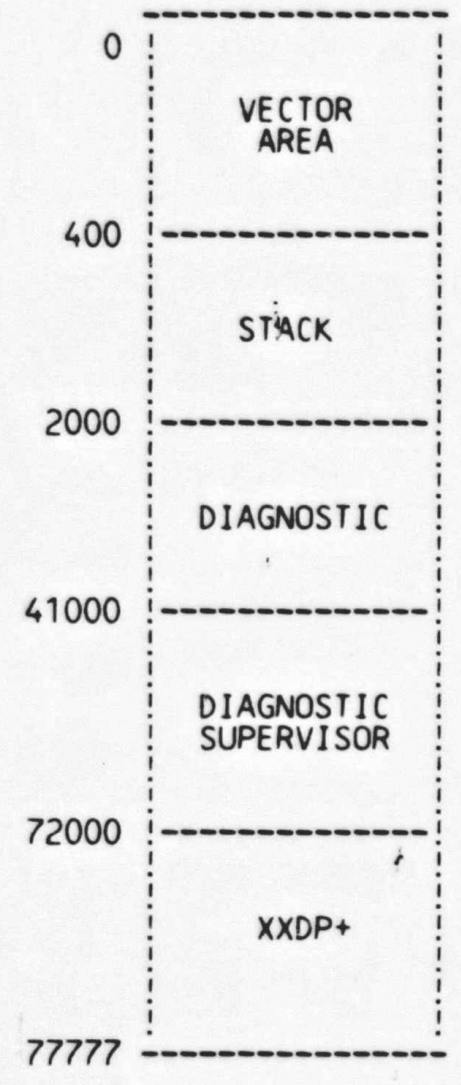
#### 1.1.2 STRUCTURE OF PROGRAM

---

THIS DIAGNOSTIC IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, BUT IT CONTAINS A CONTROL MODULE RELEASED INDEPENDENTLY AS A DIAGNOSTIC SUPERVISOR.



1.1.3 MEMORY MAP



FREE MEMO SPACE FOR WR/RD BFRS OR OTHER PUROSES IS ALLOCATED BY THE SUPERVISOR ON REQUEST OR CHOSEN BY PROGRAMMER TO RESIDE BETWEEN THE DIAG AND THE SUPERVISOR.



## 1.1.4 DIAGNOSTIC INFORMATION

### 1.1.4.1 SCOPE

THIS DIAGNOSTIC CAN TEST UP TO 4 UNITS SIMULTANEOUSLY. THE 4 UNITS ARE ASSIGNED LOGICAL UNIT NUMBERS 0 - 3 BY THE DIAGNOSTIC.

THERE ARE 5 TESTS IN THIS PROGRAM:

- TEST 1 - BASIC FUNCTIONS.
- TEST 2 - DATA RELIABILITY.
- TEST 3 - WRITE COMPATABILITY/WRITE UTILITY.
- TEST 4 - READ COMPATABILITY/READ UTILITY.
- TEST 5 - OPERATOR SELECTED SEQUENCE UTILITY.

### 1.1.4.2 ERROR RECOVERY

ERROR RECOVERY IS PERFORMED ON READ, WRITE AND WRITE TAPE MARK ERRORS UNLESS RECOVERY IS INHIBITED BY THE OPERATOR. THE READ FORWARD/READ REVERSE RETRY LIMIT IS 16 (8 IN THE SAME DIRECTION AND 8 IN THE OPPOSITE DIRECTION). FOR MORE INFORMATION ON ERROR RECOVERY PROCEDURES, SEE SECTION 3.0 (ERROR REPORTING).

### 1.1.4.3 WRITE ERROR RECOVERY

THERE ARE 2 DISTINCT, SELECTABLE WRITE-ERROR-RECOVERY ALGORITHMS:

1. MEDIA/OPERATIONAL SELECTIVE ALGORITHM
2. OPERATIONAL ALGORITHM

BY DEFAULT THE DIAGNOSTIC SELECTS THE FIRST ALGORITHM TO DISCERN MEDIA RELATED WRITE ERRORS FROM OPERATIONAL ONES.

TO SELECT THE SECOND ALGORITHM:

- ANSWER 'Y' TO CHANGE SW (L) ?
- ANSWER 'N' TO BAD TAPE SPOT DETECTION (L) Y ?

WHEN ERROR RECOVERY IS INHIBITED, THE LATTER QUESTION IS NOT ASKED AND BOTH ALGORITHMS ARE BYPASSED.

#### 1.1.4.3.1 MEDIA/OPERATIONAL SELECTIVE WRITE-ERROR-RECOVERY ALGORITHM

##### SCOPE

THE ALGORITHM DISCERNS MEDIA RELATED WRITE ERRORS FROM OPERATIONAL ONES.

##### ALGORITHM :

A WRITE RETRY SUBROUTINE IS CALLED BY THE RECOVERABLE ERROR SUBROUTINE ENTERED UPON DETECTION OF A WRITE RECOVERABLE ERROR.

THE WRITE RETRY SUBROUTINE REWRITES RECORD IN SAME SPOT ON TAPE: REPEAT 4 TIMES.

IF ALL 4 REPEATS ARE GOOD, RECORD IS CONSIDERED AS RECOVERED AND A RECOVERABLE WRITE ERROR IS LOGGED AT THAT RECORD NUMBER.

IF ANY OF THE 4 REPEAT FAILS, ERASE BAD RECORD, LOGG SUSPECTED BAD SPOT AT THAT RECORD NUMBER, RETRY AGAIN 3 INCHES FURTHER DOWN TAPE. RETRY 4 TIMES, UP TO 4 REPEATS EACH.

IF RECORD CANNOT BE WRITTEN WITHOUT RECOVERABLE ERROR AFTER 4 RETRIES, ERASE RECORD, REPORT RETRY FAILED ON BAD SPOT.

THE RECOVERABLE ERROR SUBROUTINE THEN CONTINUES TO CALL THE WRITE RETRY SUBROUTINE, WHICH REISSUES THE GROUP OF 4 RETRIES, UNTIL THE RECORD IS RECOVERED OR 20 BAD SPOTS HAVE BEEN LOGGED .



TWENTY (20) BAD SPOTS MAXIMUM ARE ALLOWED PER TAPE PASS. WHEN 20 BAD SPOTS HAVE BEEN LOGGED, ON SAME RECORD NUMBER OR NOT, TAPE IS CONSIDERED DEFECTIVE: A BAD TAPE OVERFLOW MESSAGE IS PRINTED AND UNIT IS REWOUND, THEN DROPPED.

DURING THE RECOVERY PROCESS, IT IS NECESSARY TO PERFORM SEVERAL TAPE POSITION OPERATIONS: SPACE REVERSE, ERASE. IF A POSITION ERROR STATUS IS DETECTED DURING THOSE OPERATIONS, THEN THE RECOVERY ATTEMPT IS ABORTED, AN APPROPRIATE UNRECOVERABLE MESSAGE IS PRINTED AND UNIT IS DROPPED.

ALL BADLY WRITTEN RECORDS FLAGGED WITH RECOVERABLE ERRORS ARE ERASED UNTIL RECOVERED, INCLUDING THE RECORD AT THE 20TH BAD SPOT, SO THAT ALL RECORDS LEFT ON TAPE ARE GOOD WRITTEN RECORDS. BAD SPOTS ARE ERASED, WITH ERASE GAPS FROM 3 TO 12 INCHES PER RETRY GROUP. UP TO 20 FEET OF ERASE GAP COULD RESULT WHEN RETRYING TO RECOVER A SINGLE RECORD, IF NO BAD SPOT WERE PREVIOUSLY DETECTED. THAT LONG STRETCH OF BAD TAPE WOULD THEN BE FLAGGED WITH 20 BAD SPOTS AT SAME RECORD NUMBER AND THE TAPE CONSIDERED DEFECTIVE.

#### BAD SPOTS REPORTS

IF THE PRINT OF RECOVERABLE ERRORS IS ENABLED, THE BAD SPOTS ON TAPE ARE IDENTIFIED AS THEY ARE DETECTED. SINCE THE BAD RECORDS ARE ERASED UNTIL RECOVERED, THE BAD SPOTS ACTUALLY PRECEDES THE RECORD NUMBER THAT IDENTIFIES THEM. THE NUMBER OF REPEATS AND RETRIES ATTEMPTED IS PRINTED, FROM WHICH THE LENGTH OF ERASE GAPS CAN BE DETERMINED: APPROXIMATELY 3 INCHES PER RETRY.

THE STATISTICAL REPORT PRINTED AT THE END OF TEST 2 OR UPON A 'PRINT' REQUEST, CONTAINS A SUMMARY OF THE BAD SPOTS LOGGED ON THE CURRENT TAPE PASS. IN THAT REPORT, ALL COUNTS ARE CUMULATIVE FROM PASS TO PASS, EXCEPT FOR THE NUMBER OF BAD SPOTS: IT RELATES TO A 'TAPE PASS' ONLY. FOR THIS PURPOSE, A 'TAPE PASS' IS A WRITE PASS FROM BOT TO EOT, OR FROM BOT TO WHERE THE DIAGNOSTIC IS HALTED BEFORE REACHING EOT. A PASS IS DEFINED BY THE SUPERVISOR AS A RUN THROUGH ALL THE TESTS REQUESTED ON ALL UNITS SELECTED. THOSE PASSES ARE IDENTIFIED AS 'PASS' AND 'EOP'.

THE NUMBER OF WRITE RETRIES, CUMULATIVE FROM PASS TO PASS, IS A GLOBAL COUNT OF HOW MANY TIMES THE GROUP OF 4 RETRIES HAS BEEN CALLED.

THE NUMBER OF WRITE RECOVERABLE ERRORS EXCLUDES BAD TAPE SPOTS AND REFLECTS THE SPECIFICATIONS OF THE HARDWARE UNDER TEST. PER TAPE PASS, THE NUMBER OF WRITE RETRIES EQUALS THE SUM OF THE NUMBER OF RECOVERABLE WRITE ERRORS AND BAD SPOTS, MOST OF THE TIME.

TO CLEAR CUMULATIVE COUNTS, ANSWER 'Y' TO: CLEAR COUNTERS (L) Y ?. BAD TAPE SPOTS COUNT IS CLEARED WHEN WRITING FROM BOT.

IF TEST 2 IS HALTED, THEN RESTARTED OR CONTINUED, THE RECORD COUNT IS RESET TO ZERO AND THE BAD SPOT ID SHALL FOLLOW THAT RESET COUNT.

SINCE ALL WRITTEN RECORDS ARE KNOWN GOOD, THE READ ERRORS CAN BE ATTRIBUTED TO TRANSIENT NOISE, TRANSIENT ELECTRICAL MALFUNCTIONS, OR CONTAMINANTS ON TAPE AS OPPOSED TO TAPE DEFECTS.

THE SAME RECORDS MUST BE WRITTEN FORM TAPE PASS TO TAPE PASS FOR THE BAD SPOTS ID TO REMAIN CONSISTENT IN THOSE TAPE PASSES.

EXAMPLE OF A TAPE PASS PRINTS:

CZTSH SFT ERR 00009 ON UNIT 00 TST 002 SUB 000 PC: 012100  
RECOVERABLE ERROR



WRT CMD FAILED - UNIT 0 PASS: 1 RECORD: 6  
 PREVIOUS CMD WAS WRT  
 CNDPKT TSBA RFC TSSR TCC  
 100205 002406 000000 100210 4  
 026600  
 000000  
 003107  
 XST0 XST1 XST2 XST3  
 000350 000002 100400 000000  
 SUSPECT BAD SPOT AFTER 1 RETRY, 2 REPEAT  
 SUSPECT BAD SPOT AFTER 2 RETRY, 1 REPEAT  
 SUSPECT BAD SPOT AFTER 3 RETRY, 1 REPEAT  
 SUSPECT BAD SPOT AFTER 4 RETRY, 3 REPEAT  
 RETRY FAILED ON BAD SPOT...ERASED!  
 SUSPECT BAD SPOT AFTER 1 RETRY, 1 REPEAT  
 SUSPECT BAD SPOT AFTER 2 RETRY, 1 REPEAT

CZTSH SFT ERR 00009 ON UNIT 00 TST 002 SUB 000 PC: 012100  
 RECOVERABLE ERROR

WRT CMD FAILED - UNIT 0 PASS: 1 RECORD:10210  
 PREVIOUS CMD WAS WRT  
 CNDPKT TSBA RFC TSSR TCC  
 100205 002406 000000 100210 4  
 026600  
 000000  
 004000  
 XST0 XST1 XST2 XST3  
 000350 000002 100010 000000  
 RECOVERED ON RETRY # 1

^C  
 DR>PRI

UNIT 0 PASS: 1 RECORD:10210  
 BYTES WRITTEN 0,272,279,691  
 BYTES READ REV 0,301,123,654  
 BYTES READ REV 0,301,120,381

	WRT	RDR	RDF
RECOVERABLE ERRORS	1	0	0
UNRECOVERABLE ERRORS	0	0	0
WRITE RETRIES	3		

2 BAD SPOTS THIS TAPE PASS PRECEDING RECORD #:

SPEC COND	HARD	FATAL	COMPARE
2	0	0	0

DR>

THIS EXAMPLE SHOWS:

RECORD 6 RECOVERED ON 2ND RETRY GROUP  
 THE 2 BAD SPOTS RESIDE IN A 18 INCH ERASE GAP BETWEEN RECORDS 5 AND 6  
 RECORD 10210 RECOVERED ON 1ST RETRY OF 4 GOOD REPEATS  
 3 WRITE GROUP RETRIES ATTEMPTED, RESULTING IN:  
 1 RECOVERABLE WRT ERR FROM RECORD 10210  
 2 BAD SPOTS BETWEEN RECORDS 5 AND 6

#### 1.1.4.3.2 OPERATIONAL WRITE-ERROR-RECOVERY ALGORITHM

WHEN THIS ALGORITHM IS SELECTED, THE TS11 WRITE RETRY COMMAND IS ISSUED UP TO 16 TIMES OR UNTIL RECORD IS RECOVERED, ON



A WRITE RECOVERABLE ERROR. THE WRITE RETRY COMMAND CONSISTS OF A SPACE REVERSE OVER THE BAD RECORD, THEN AN ERASE OF 3 INCHES OF TAPE AND REWRITE OF THE RECORD. THAT COMPOSITE COMMAND DOES NOT ALLOW TO DETECT BAD SPOTS ON TAPE. THEREFORE NO BAD TAPE SPOTS STATUS IS PRINTED.

IF RECORD CANNOT BE RECOVERED AFTER 16 WRITE RETRY COMMANDS, A RETRY LIMIT EXCEEDED IS FLAGGED AND UNIT IS DROPPED.

#### 1.1.4.4 DIAGNOSTIC TIMING ADJUSTMENT

A NUMBER OF SUPERVISOR TIMING DELAYS MACROS, KNOWN AS WATCH DOG DELAYS, ARE CALLED BY THE DIAGNOSTIC TO WAIT FOR VARIOUS COMMANDS COMPLETION. THESE DELAYS ARE NOT CALIBRATED AND SIMPLY EXPANDS INTO AN INLINE NESTED LOOP PAIR. THE COUNT FOR THE OUTER LOOP COMES FROM THE VARIABLE ARGUMENT SUPPLIED BY THE DELAY CALLS. THE COUNT FOR THE INNER LOOP COMES FROM THE FIXED 'HEADER' ELEMENT 'LSDLY'. AS THE DIAGNOSTIC IS RUN ON DIFFERENT CPU'S, THESE DELAYS WILL VARY IN LENGTH WITH MEMORY SPEED.

IF TIME-OUT OCCURS WHEN NO APPARENT MALFUNCTIONS IN THE TAPE UNIT IS EVIDENT, ALL TIMINGS OF THE DIAGNOSTIC MAY BE ADJUSTED TO MATCH MEMORY SPEED AND NOT RESULT IN TIME-OUTS, BY PATCHING THAT FIXED DELAY ELEMENT 'LSDLY'.

A PRESET COUNT OF 500 RESIDES AT 'LSDLY' IN LOCATION 2116 OF THE 'HEADER' SECTION.

### 1.2 SYSTEM REQUIREMENTS

#### 1.2.1 HARDWARE REQUIREMENTS

PDP-11 PROCESSOR WITH 16K OR MORE OF MEMORY  
CONSOLE DEVICE (LA30,LA36,VT50,ETC.)  
PROGRAM LOAD DEVICE

#### 1.2.2 SOFTWARE REQUIREMENTS

DIAGNOSTIC SUPERVISOR



### 1.3 RELATED DOCUMENTS AND STANDARDS

---

XXDP+ USERS MANUAL MD-11-CHQUS  
DIAGNOSTIC SUPERVISOR PROGRAM LISTING  
PDP-11 DIAGNOSTIC SUPERVISOR INTERFACE SPECIFICATION.  
PDP-11 DIAGNOSTIC SUPERVISOR PROGRAMMER'S GUIDE.  
TS11/TS04 PROGRAMMING SPECIFICATION.  
TS11/TS04 ENGINEERING SPECIFICATION.  
TS11/TS04 COMMAND PACKET SPECIFICATION.

### 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

---

ORDER OF HOST CPU DIAGNOSTIC USAGE:

- 1) CONTROL LOGIC PROGRAM - ALL TESTS.
- 2) DATA RELIABILITY PROGRAM:
  - A) BASIC FUNCTION TEST.
  - B) DATA RELIABILITY TEST.

### 1.5 ASSUMPTIONS

---

THE HARDWARE OTHER THAN THE SUBSYSTEM BEING TESTED IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, MEMORY, ETC., DO NOT FUNCTION PROPERLY.



## 1.6 DIAGNOSTIC HISTORY

N 1

- REVISION A - OCT 1978
  - ORIGINAL RELEASE
- REVISION B - FEB 1979
  - CORRECTED END OF TAPE PROBLEMS IN TESTS 3-5.
  - CHANGED DEFAULT VECTOR ADDRESS FROM 150 TO 224.
  - DECREASED MAXIMUM RECORD LENGTH FROM 4096 TO 2048 BYTES.
- REVISION B - AUG 1979
  - DO NOT PRINT RECOVERABLE ERRORS UNLESS REQUESTED BY OPERATOR
  - WARN OPERATOR OF UNIT(S) BEING NOT READY OR OFF-LINE.
  - DROP UNIT(S) LEFT NOT READY OR OFF-LINE FOR 3.5 MINUTES.
  - IMPROVE BEHAVIOR AT EOT
  - IN TEST 2, FREEZE UNITS REACHING EOT UNTIL OTHERS CATCH-UP INSTEAD OF ALLOWING THEM TO SHUTTLE AT EOT
  - WHEN ALL UNITS REACH EOT, WRITE ONE RECORD BEYOND EOT. READ REV THAT EXTRA RECORD TO POSITION TAPE SO THAT THE NEXT COMMAND REQUESTED CAN BE EXECUTED. THAT EXTRA RECORD SHALL LEAVE A CLEAN IRG GAP AND A VALID RECORD TO READ WHEN SHORTER READ STOP DISTANCE MIGHT CAUSE UNIT TO FLAG EOT ON THAT EXTRA RECORD INSTEAD OF THE PREVIOUS ONE. THIS SHOULD ELIMINATE MANY READ ERRORS AT EOT AND TAPES RUNNING OFF THE WHEELS.
  - WRITE RECORD COUNT ON TAPE.
  - PRINT RECORD COUNT READ FROM TAPE IN READ ERROR PRINTS TO INDICATE IF POSITION WAS LOST.

### \* CAUTION \*

- INTERPRET THAT 'RECORD READ' COUNT WITH CAUTION. IF VERY DIFFERENT FROM RECORD COUNT TRACKED BY THE DIAGNOSTIC, POSITION IS NOT NECESSARELY LOST. ERRORS IN READING THAT RECORD MIGHT HAVE CAUSED RECORD COUNT TO BE ERRONEOUSLY READ FROM TAPE.
- IN TEST 2, IF DIAGNOSTIC IS RESTARTED OR CONTINUED, RECORD COUNT IS RESET TO ZERO ALTHOUGH TAPE WAS NOT REWOUND. THIS IS NECESSARY BECAUSE THERE IS NO ACCURATE WAY TO DETERMINE ON WHAT RECORD COUNT OF WHAT UNIT THE DIAGNOSTIC WAS HALTED BEFORE RESTARTING OR CONTINUING.
- IT IS SUGGESTED THAT A 'PRINT' BE REQUESTED WHEN HALTING DIAG TO GET A PRINT OF THE RECORD COUNT WHEN HALTED.
- VERIFY RECORD OF 4000 BYTES INSTEAD OF 22 BYTES.
  - WHEN COMPARING DATA, CHECK AND PRINT IF NO DATA WAS READ OR RECORD WAS LONGER THAN EXPECTED.
  - FREEZE TSSR REG WHEN A COMMAND IS COMPLETED TO AVOID DIFFERENCES BETWEEN TSSR AND TCC FETCHED AT DIFFERENT TIMES.
  - WHEN DROPPING A UNIT, FLAG SECOND PRINT OF EXTENDED STATUS AS BEING THE RESULT OF A GET STATUS COMMAND.
  - WAIT FOR SSR UP BEFORE PRINTING THAT STATUS.
  - ADJUST 'PASS' COUNT OF DIAG TO MATCH 'EOP' PASS COUNT OF SUPERVISOR.
  - INCREASE NUMBER OF SELECTABLE COMMANDS IN TEST 5 FROM 4 TO 7. DEFAULT COMMAND 6 IS NOW REWIND.
  - CONVERT DIAG TO REV C OF SUPERVISOR.
- ADD SEVERAL SECTIONS:
- PROTECT TABLE
  - AUTO-DROP CODE
  - HARD CODED PARAMETER TABLE
- REVISION C - OCT 79
  - ADD MEDIA/OPERATIONAL SELECTIVE WRITE-ERROR-RECOVERY ALGORITHM TO DETECT BAD SPOTS ON TAPE.



## 2.0 OPERATING INSTRUCTIONS

---

FOR OPERATING INSTRUCTIONS, PLEASE SEE CHAPTER 5 OF XXDP+ OPERATOR'S MANUAL.

## 2.1 HARDWARE PARAMETERS

---

ON A 'N' RESPONSE TO 'CHANGE HW?', THE DIAG SHALL RUN ASSUMING ONE UNIT AT TSSR = 172522 WITH A VECTOR = 224.

ON A 'Y' RESPONSE TO 'CHANGE HW?' QUESTION, THEN THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

TSSR ADDRESS (172522) ?

VECTOR (224) ?

THE VALIDITY OF THESE PARAMETERS CAN BE CHECKED BEFORE RUNNING THE TESTS BY SETTING THE FLAG 'ADR' ON A STA, RES OR CON COMMAND. THE SO CALLED AUTO DROP CODE SHALL THEN BE EXECUTED AFTER THE INIT CODE AND BEFORE THE HARDWARE TESTS ARE RUN. THAT CODE FIRST TESTS THE ADDRESS OF THE TSSR(S). IF NO RESPONSE, IT DROPS THE UNIT(S) IMMEDIATELY WITH THE FOLLOWING MESSAGE:

BUS TRAP AT XXXXXX ( XXXXXX = TSSR AD )

INTERFACE BAD OR NOT SET TO ABOVE AD.

ON A RESPONSE FROM THE INTERFACE, THE UNITS THAT ARE NOT READY OR NOT ON-LINE ARE DROPPED IMMEDIATELY. THE HARDWARE TESTS SHALL THEN BE RUN ON RESPONDING UNITS.

IF THE 'ADR' FLAG IS NOT SET, THE READY AND OFF-LINE STATUS OF THE UNITS ARE CHECKED. A MESSAGE SHALL BE PRINTED EVERY SO OFTEN TO WARN THE OPERATOR OF UNITS BEING NOT READY OR OFF-LINE. THESE UNITS SHALL BE DROPPED AFTER A REASONABLE AMOUNT OF TIME (3 MIN ON A 11/70).



## 2.2 SOFTWARE PARAMETERS

C 2

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

CLEAR COUNTERS (L) Y ?

RESET RANDOM VARIABLES (L) N ?

PRINT RECOVERABLE ERRORS (L) N ?

HALT AFTER EACH CMD (L) N ?

INHIBIT RECOVERY (L) N ?

BAD TAPE SPOT DETECTION (L) Y ?

DISABLE INTERRUPTS (L) N ?

INHIBIT RFC ERROR REPORTS (L) N ?

CHANGE CMD SEQUENCE (L) N ?

NOTE: THIS QUESTION SHOULD BE ANSWERED (N) UNLESS AN OPERATOR SELECTED SEQUENCE IS TO BE EXECUTED. IF THIS QUESTION WAS ANSWERED (N), NO MORE QUESTIONS WILL BE ASKED. IF THIS QUESTION WAS ANSWERED Y, THE FOLLOWING QUESTIONS MUST BE ANSWERED OR DEFAULTED WITH A <CR> ONLY:

CHARACTERISTICS CODE (O) 40 ?	(0,20,40,200) (OCTAL)	
CMD/2 (D) 13 ?	(1-27)	(DECIMAL)
BRF COUNT (D) 1 ?	(1-2K)	(DECIMAL)
# OF OPERATIONS (D) 1 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/3 (D) 4 ?	(1-27)	(DECIMAL)
BRF COUNT (D) 2048 ?	(1-2K)	(DECIMAL)
# OF OPERATIONS (D) 32000 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/4 (D) 3 ?	(1-27)	(DECIMAL)
BRF COUNT (D) 2048 ?	(1-2K)	(DECIMAL)
# OF OPERATIONS (D) 32000 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/5 (D) 2 ?	(1-27)	(DECIMAL)
BRF COUNT (D) 2048 ?	(1-2K)	(DECIMAL)
# OF OPERATIONS (D) 32000 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/6 (D) 13 ?	(1-27)	(DECIMAL)
BRF COUNT (D) 1 ?	(1-2K)	(DECIMAL)
# OF OPERATIONS (D) 1 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/7 (D) 27 ?	(1-27)	(DECIMAL)
BRF COUNT (D) 2048 ?	(1-2K)	(DECIMAL)
# OF OPERATIONS (D) 32000 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/8 (D) 27 ?	(1-27)	(DECIMAL)
BRF COUNT (D) 2048 ?	(1-2K)	(DECIMAL)
# OF OPERATIONS (D) 32000 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)

NOTE: THE PROGRAM AUTOMATICALLY INSERTS AN CHARACTERISTIC 40 AS THE FIRST COMMAND IN THE SEQUENCE TABLE. IF A



DIFFERENT CHARACTERISTIC IS DESIRED, THE OPERATOR SHOULD ENTER THAT CHARACTERISTIC CODE. A TOTAL OF 7 COMMANDS MAY BE ENTERED IN ADDITION TO THE SET CHARACTERISTICS COMMAND. IF THE OPERATOR WISHES TO USE LESS THAN 7 COMMANDS, AN END COMMAND MUST BE ENTERED AND THEN A CONTROL Z (^Z) CAN BE ENTERED TO TERMINATE SOFTWARE DIALOGUE.



## 2.2.1 COMMAND LIST FOR USE IN SOFTWARE DIALOGUE.

CODE	COMMAND	DESCRIPTION
1 =	DRI	DRIVE INITIATE.
2 =	RDF	READ FORWARD.
3 =	RDR	READ REVERSE.
4 =	WRT	WRITE.
5 =	WTV	WRITE/VERIFY. IE. WRITE N RECORDS; READ REVERSE AND CHECK N RECORDS OF DATA; READ FORWARD AND CHECK N RECORDS.
6 =	SRF	SPACE RECORDS FORWARD.
7 =	SRR	SPACE RECORDS REVERSE.
8 =	RNR	READ NEXT REVERSE, IE. SPACE FWD, READ REV.
9 =	RNF	READ NEXT FORWARD, IE. READ FWD, SPACE REV.
10 =	RPF	READ PREVIOUS FWD, IE. SPACE REV, READ FWD.
11 =	RPR	READ PREVIOUS REV, IE. READ REV, SPACE FWD.
12 =	WRR	WRITE RETRY.
13 =	RWD	REWIND.
14 =	MBR	MESSAGE BUFFER RELEASE.
15 =	WTM	WRITE TAPE MARK.
16 =	WTR	WRITE TAPE MARK RETRY.
17 =	SFF	SPACE FILES FORWARD.
18 =	SFR	SPACE FILES REVERSE.
19 =	GES	GET EXTENDED STATUS.
20 =	ERS	ERASE 3 INCHES OF TAPE.
21 =	UNL	UNLOAD.
22 =	CLN	CLEAN TAPE
23 =	SCH	SET DEVICE CHARACTERISTIC. WHERE BRF=200, 40, 20, 0. 200 = ENABLE SKIP TAPE MARKS STOP (STOP AT LOGICAL EOT) 40 = ENABLE ATTENTION INTERRUPTS. 20 = ENABLE MESSAGE BUFFER RELEASE INTERRUPTS. SEE TS11/TS04 PROGRAMMING SPECIFICATION FOR DESCRIPTION.
24 =	DIA	DIAGNOSTICS. SEE TS11/TS04 PROGRAMMING SPECIFICATION FOR DESCRIPTION. ODT MUST BE USED TO LOAD DIAGNOSTIC DATA INTO THE WRITE BUFFER BEFORE THIS CMD IS ISSUED.
25 =	JMP	JUMP TO THE NTH COMMAND IN THE COMMAND SEQUENCE TABLE, WHERE N IS DEFINED IN THE BRF FIELD. THE NUMBER OF JUMPS IS ENTERED IN THE # OF OPERATIONS FIELD
26 =	DLY	DELAY 'N' MILLISECONDS WHERE N IS DEFINED IN THE # OF OPERATIONS.
27 =	END	END OF COMMAND SEQUENCE.

## 2.2.2 DATA PATTERN LIST FOR USE IN SOFTWARE DIALOGUE.

PATTERN #	DESCRIPTION.
0	INCREMENTING PATTERN. 0 - 377.
1	ALL '1''S PATTERN.
2	ALL '0''S PATTERN.
3	'1'' BIT WALKING FROM R TO L IN A FIELD OF '0''S.
4	'0'' BIT WALKING FROM R TO L IF A FIELD OF '1''S.
5	ALTERNATING '1'' AND '0'' BITS WITH ALTERNATE BYTES COMPLIMENTED. (125/252)
6	ALTERNATING BYTES OF 000 AND 377.
7	RANDOM DATA PATTERN.
8	NO PATTERN GENERATION.



## 2.3 EXAMPLES OF SOFTWARE DIALOGUE

### 2.3.1 BASIC FUNCTION AND DATA RELIABILITY WITH ALL ERROR REPORTING ENABLED

- A) RECEIVE PROMPT (DR>)
- B) ENTER STA/TES:1-2<CR>
- C) ANSWER HARDWARE QUESTIONS.
- D) PROCEED WITH THE FOLLOWING DIALOGUE:

CHANGE SW (L) ?	Y<CR>
CLEAR COUNTERS (L) N ?	Y<CR>
RESET RANDOM VARIABLES (L) N ?	N<CR>
PRINT RECOVERABLE ERRORS (L) N ?	Y<CR>
HALT AFTER EACH CMD (L) N ?	N<CR>
INHIBIT RECOVERY (L) N ?	N<CR>
BAD TAPE SPOT DETECTION (L) Y ?	Y<CR>
DISABLE INTERRUPTS (L) N ?	N<CR>
INHIBIT RFC ERROR REPORT (L) N ?	N<CR>
CHANGE CMD SEQUENCE (L) N ?	N<CR>

## 2.3.2 TO SET UP A SCOPE LOOP FOR A FAILURE IN BASIC FUNCTIONS.

- A) RECEIVE PROMPT (DR>)
- B) ENTER STA/TES:1/FLA:LOE:IER:ISR:IDU<CR>
- C) ANSWER HARDWARE QUESTIONS.
- D) PROCEED WITH THE FOLLOWING DIALOGUE:

```

CHANGE SW (L) ?           Y<CR>
CLEAR COUNTERS (L) N ?   Y<CR>
RESET RANDOM VARIABLES (L) N ? N<CR>
PRINT RECOVERABLE ERRORS (L) N ? N<CR>
HALT AFTER EACH CMD (L) N ? N<CR>
INHIBIT RECOVERY (L) N ? N<CR>
BAD TAPE SPOT DETECTION (L) Y ? N<CR>
DISABLE INTERRUPTS (L) N ? N<CR>
INHIBIT RFC ERROR REPORT (L) N ? Y<CR>
CHANGE CMD SEQUENCE (L) N ? N<CR>

```

## 2.3.3 TO SET UP A SCOPE LOOP FOR A FAILURE IN DATA RELIABILITY

- A) RECEIVE PROMPT (DR>)
- B) ENTER STA/TES:5/FLA:IER:ISR:IDU/EOP:1000<CR>
- C) ANSWER HARDWARE QUESTIONS.
- D) PROCEED WITH THE FOLLOWING DIALOGUE:

```

CHANGE SW (L) ?           Y<CR>
CLEAR COUNTERS (L) N ?   Y<CR>
RESET RANDOM VARIABLES (L) N ? N<CR>
PRINT RECOVERABLE ERRORS (L) N ? N<CR>
HALT AFTER EACH CMD (L) N ? N<CR>
INHIBIT RECOVERY (L) N ? N<CR>
BAD TAPE SPOT DETECTION (L) Y ? N<CR>
DISABLE INTERRUPTS (L) N ? Y<CR>
INHIBIT RFC ERROR REPORT (L) N ? Y<CR>
CHANGE CMD SEQUENCE (L) N ? Y<CR>
CHARACTERISTICS CODE (O) 40 ? 40<CR>
CMD/2 (D) 5 ?             13<CR> (REWIND) (COULD BE ANY COMMAND)
BRF COUNT (D) 2048 ?      1<CR>
# OF OPERATIONS (D) 10 ? 1<CR>
PATTERN (D) 7 ?           1<CR>
CMD/3 (D) 5 ?             4<CR> (WRITE) (COULD BE ANY COMMAND)
BRF (D) 2048 ?            1000<CR>
# OF OPERATIONS (D) 10 ? 10000<CR>
PATTERN (D) 7 ?           1<CR>
CMD/4 (D) 5 ?             27<CR> (END) (COULD BE ANY COMMAND)
BRF (D) 2048 ?            <^Z>

```



## 2.4 EXECUTION TIMES

---

### 2.4.1 SYSTEM CONFIGURATION

---

PDP11/34  
MOS MEMORY  
LA36  
TS11/TS04

### 2.4.2 TEST EXECUTION TIMES

---

TEST 1 - BASIC FUNCTIONS - 30 SECONDS PER PASS.  
TEST 2 - DATA RELIABILITY - 45 MINUTES PER PASS.  
TEST 3 - WRITE COMPATABILITY - 20 MINUTES PER PASS.  
TEST 4 - READ COMPATABILITY - 20 MINUTES PER PASS.  
TEST 5 - OPERATOR SELECTED SEQUENCE - DEPENDS ON SEQUENCE SELECTED.

NOTE: ALL EXECUTION TIMES ARE SHOWN FOR ONE UNIT OPERATION.  
APPROXIMATELY 10% WILL BE ADDED TO ALL EXECUTION TIMES  
FOR EACH ADDITIONAL UNIT.

### 3.0 ERROR INFORMATION

#### 3.1 ERROR REPORTING

ALL ERROR REPORTS EXCEPT FOR ERRORS #1 AND #17 INCLUDE A DUMP OF THE FOLLOWING INFORMATION:

ERROR #, TEST #, SUBTEST #, PROGRAM COUNTER, UNIT #, COMMAND, PREVIOUS COMMAND, PASS COUNT, # OF RECORDS FROM BOT, RECORD READ COUNT, THE COMMAND PACKET, TSSR, TCC, TSBA, RFC, AND THE EXTENDED STATUS REGISTERS (SEE 2.3.14.1 FOR LIST OF COMMANDS).

STANDARD ERROR REPORT FORMAT:

```
CZTSH SFT ERR XXXXX TST XXX SUB XXX PC: XXXXXX
(ASCII ERROR MESSAGE)
XXX CMD FAILED - UNIT X PASS: XXXXX RECORD: XXXXX
PREVIOUS CMD WAS XXX * RECORD READ: XXXXX *
CMDPKT TSBA RFC TSSR TCC
XXXXXX XXXXXX XXXXXX XXXXXX X
XXXXXX
XXXXXX
XXXXXX
XST0 XST1 XST2 XST3
XXXXXX XXXXXX XXXXXX XXXXXX
```

\* CAUTION \*

INTERPRET THAT 'RECORD READ' COUNT WITH CAUTION. IF VERY DIFFERENT FROM RECORD COUNT TRACKED BY THE DIAGNOSTIC, POSITION IS NOT NECESSARELY LOST. ERRORS IN READING THAT RECORD MIGHT HAVE CAUSED RECORD COUNT TO BE ERRONEOUSLY READ FROM TAPE.

IN TEST 2, IF DIAGNOSTIC IS RESTARTED OR CONTINUED, RECORD COUNT IS RESET TO ZERO ALTHOUGH TAPE WAS NOT REWOUND. THIS IS NECESSARY BECAUSE THERE IS NO ACCURATE WAY TO DETERMINE ON WHAT RECORD COUNT OF WHAT UNIT THE DIAGNOSTIC WAS HALTED BEFORE RESTARTING OR CONTINUING.

IT IS SUGGESTED THAT A 'PRINT' BE REQUESTED WHEN HALTING DIAG TO GET A PRINT OF THE RECORD COUNT WHEN HALTED.

EXAMPLE OF AN ERROR REPORT:

```
CZTSH SFT ERR 00009 TST 002 SUB 000 PC: 010606
RECOVERABLE ERROR
WRT CMD FAILED - UNIT 2 PASS: 2 RECORD: 254
PREVIOUS CMD WAS WRT
CMDPKT TSBA RFC TSSR TCC
100005 002324 000000 100210 4
051766
000000
000371
XST0 XST1 XST2 XST3
000350 000002 100004 000000
```

##### 3.1.1 ERROR #1 - COMMAND PACKET ADDRESS NOT ON A MODULO 4 BOUNDARY:



IF THIS ERROR IS REPORTED, THE PROGRAM DID NOT LOAD PROPERLY. THIS IS A SYSTEM FATAL ERROR AND THE PROGRAM MUST BE RELOADED TO CORRECT IT.

### 3.1.2 ERROR #2 - TS04 NOT READY:

BEFORE ANY COMMAND IS ISSUED TO THE TS04, THE SUBSYSTEM READY BIT IN THE TSS4 IS CHECKED. IF THE SSR IS NOT SET, THE PROGRAM REPORTS THE NOT READY ERROR. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST SEQUENCE UNLESS THE IDU OPTION IS USED.

### 3.1.3 ERROR #3 - NO RESPONSE ERROR:

ONCE THE TSDB IS LOADED, THE TS04 HAS ONE MILLISECOND TO RESPOND OR THE PROGRAM REPORTS A NO RESPONSE ERROR. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST SEQUENCE UNLESS THE IDU OPTION IS USED.

### 3.1.4 ERROR #4 - NO INTERRUPT ERROR:

COMMAND WAS ISSUED AND NO INTERRUPT RECEIVED. THE PROGRAM REPORTS THAT NO INTERRUPT OCCURRED. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

### 3.1.5 SPECIAL CONDITION ERRORS:

IF, DURING EXECUTION, AN INCIDENT OCCURS FORCING THE TSSR SPECIAL CONDITION BIT TO SET, THE PROGRAM WILL SELECT ONE OF 8 ERROR HANDLING ROUTINES, DEPENDING ON THE TERMINATION CLASS CODE.

THE TERMINATION CLASS CODES IN THE TSSR ARE PROCESSED AS FOLLOWS WHEN SPECIAL CONDITION IS SET:

#### 3.1.5.1 ERROR #5 - TERMINATION CLASS CODE 0, UNDEFINED SPECIAL CONDITION

THE ERROR IS REPORTED, A HARD ERROR IS LOGGED AND THE PROGRAM PROCEEDS NORMALLY.

#### 3.1.5.2 ERROR #6 - TERMINATION CLASS CODE 1, ATTENTION CONDITION

THIS TCC INDICATES THAT THE DRIVE HAS UNDERGONE A STATUS CHANGE SUCH AS GOING OFFLINE OR COMING ONLINE. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

#### 3.1.5.3 ERROR #7 - TERMINATION CLASS CODE 2, TAPE STATUS ALERT

A STATUS CONDITION HAS BEEN ENCOUNTERED THAT MAY HAVE SIGNIFICANCE TO THE PROGRAM. BITS OF INTEREST INCLUDE TMK, RLS, LET, RLL, EOT. ACTION TAKEN DEPENDS ON THE TEST BEING EXECUTED. IF THE CONDITION IS UNEXPECTED, THE ERROR IS REPORTED AND

A HARD ERROR IS LOGGED. THE PROGRAM PROCEEDS NORMALLY.

#### 3.1.5.4 ERROR #8 - TERMINATION CLASS CODE 3, FUNCTION REJECT

THE SPECIFIED FUNCTION WAS NOT INITIATED. BITS OF INTEREST ARE RMR, OFL, VCK, BOT, ILC, WLE, ILA, AND NBA. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

#### 3.1.5.5 ERROR #9 - TERMINATION CLASS CODE 4, RECOVERABLE ERROR

TAPE POSITION IS ONE RECORD BEYOND WHAT ITS POSITION WAS WHEN THE FUNCTION WAS INITIATED. RECOVERY PROCEDURE IS TO LOG THE ERROR AND ISSUE THE APPROPRIATE RETRY COMMAND. IF RETRY LIMIT IS REACHED BEFORE THE ERROR IS RECOVERED, RETRY LIMIT EXCEEDED IS REPORTED AS DESCRIBED IN ERROR #14 BELOW.

#### 3.1.5.6 ERROR #10 - TERMINATION CLASS CODE 5, RECOVERABLE ERROR

TAPE POSITION HAS NOT CHANGED. RECOVERY PROCEDURE IS TO LOG THE ERROR AND RE-ISSUE THE ORIGINAL COMMAND. IF RETRY LIMIT IS REACHED BEFORE THE ERROR IS RECOVERED, RETRY LIMIT EXCEEDED IS REPORTED AS DESCRIBED IN ERROR #14 BELOW.

#### 3.1.5.7 ERROR #11 - TERMINATION CLASS CODE 6, UNRECOVERABLE ERROR

TAPE POSITION HAS BEEN LOST. THE ONLY VALID RECOVERY PROCEDURE IS TO REWIND AND START OVER AT BOT UNLESS THE TAPE HAS LABELS OR SEQUENCE NUMBERS. IF DENSITY CHECK IS SET THIS DIAGNOSTIC WILL REWIND AND RETRY THE COMMAND, OTHERWISE THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

#### 3.1.5.8 ERROR #12 - TERMINATION CLASS CODE 7, FATAL SUBSYSTEM ERROR

THE SUBSYSTEM IS INCAPABLE OF PROPERLY PERFORMING COMMANDS OR AT LEAST ITS INTEGRITY IS SERIOUSLY QUESTIONABLE. REFER TO THE FATAL CLASS CODE FIELD IN THE TSSR REGISTER FOR ADDITIONAL INFORMATION ON THE TYPE OF FATAL ERROR. THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

#### 3.1.6 ERROR #13 - RFC NON-ZERO ERROR:

IF, AFTER EXECUTION, THE RESIDUAL FRAME COUNT IS NON-ZERO, THE ERROR IS REPORTED AND A HARD ERROR IS LOGGED. THE PROGRAM THEN PROCEEDS NORMALLY. THE REPORTING AND LOGGING OF THESE ERRORS IS OPTIONAL.

#### 3.1.7 ERROR #14 - RETRY LIMIT EXCEEDED:

ON A WRITE COMMAND THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

ON A READ COMMAND THIS ERROR IS LOGGED AS A HARD ERROR AND THE PROGRAM PROCEEDS NORMALLY.



### 3.1.8 ERROR #15 - TOO MANY INTERRUPTS:

IF MORE THAN ONE INTERRUPT OCCURS PER COMMAND, THIS ERROR IS REPORTED. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

### 3.1.9 ERROR #16 - CAPSTAN RUNAWAY:

CAPSTAN DID NOT STOP WITHIN ACCEPTABLE WINDOW AFTER LAST COMMAND. THE PROGRAM WILL ISSUE A GET STATUS COMMAND BEFORE REPORTING THE ERROR SO THAT THE DEAD TRACK FIELD IN EXTENDED STATUS REGISTER 2 WILL CONTAIN THE TACH COUNT WHEN THE TAPE STOPPED. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

### 3.1.10 ERROR #17 - DATA COMPARE ERROR:

IF A DATA VALIDATION ERROR OCCURS DURING A WRITE/VERIFY COMMAND, THE PROGRAM PRINTS WHAT THE DATA SHOULD HAVE BEEN AND WHAT THE DATA WAS, AND PRINTS THE BYTE AND RECORD NUMBER THE ERROR OCCURRED ON. ONLY THE FIRST 10 BYTES IN ERROR PER RECORD ARE PRINTED. THE TOTAL # OF BYTES IN ERROR PER RECORD IS ALSO PRINTED. A HARD ERROR IS LOGGED AND THE PROGRAM PROCEEDS NORMALLY.

## 3.2 ERROR HALTS

-----

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION WITH /FLAG:HOE. THERE ARE NO OTHER HALTS.

## 4.0 PERFORMANCE REPORT

```

UNIT X  PASS:XXXXX  RECORD:XXXXX
BYTES WRITTEN  XXX,XXX,XXX,XXX
BYTES READ REV  XXX,XXX,XXX,XXX
BYTES READ FWD  XXX,XXX,XXX,XXX

RECOVERABLE ERRORS      WRT      RDR      RDF
UNRECOVERABLE ERRORS  XXXXX    XXXXX    XXXXX

SPEC COND  HARD  FATAL  COMPARE
          XXXXX XXXXX XXXXX XXXXX

```

## 5.0 TEST SUMMARIES

- 5.1 TEST 1 - BASIC FUNCTIONS.
- EXECUTES AND VERIFIES CORRECT COMPLETION OF ALL TS04 FUNCTIONS.
- SUBTEST 1 - SET CHAR, DRIVE INIT, GET STATUS.
- + SET CHARACTERISTIC 200.
  - + DRIVE INITIATE.
  - + SET CHARACTERISTIC 20.
  - + GET STATUS
  - + SET CHARACTERISTIC 40.
  - + PRINT TS04 MICROCODE LEVEL (PASS 1 ONLY)
- SUBTEST 2 - REWIND.
- + REWIND.
  - + REWIND AT BOT.
- SUBTEST 3 - WRITE/VERIFY.
- + WRITE/VERIFY PATTERN 1.
  - + WRITE/VERIFY PATTERN 2.
  - + WRITE/VERIFY PATTERN 3.
  - + WRITE/VERIFY PATTERN 4.
  - + WRITE/VERIFY PATTERN 5.
  - + WRITE/VERIFY PATTERN 6.
  - + WRITE/VERIFY PATTERN 0.
- SUBTEST 4 - WRITE TAPE MARK, ERASE.
- + WRITE TAPE MARK.
  - + WRITE 10 RECORDS
  - + ERASE 10 TIMES
  - + WRITE TAPE MARK.
  - + WRITE TAPE MARK RETRY.
- SUBTEST 5 - SPACE FILES.
- + SPACE 2 FILES REVERSE.
  - + SPACE 2 FILES FORWARD.
  - + SPACE 2 FILES REVERSE.
  - + SPACE 2 FILES FORWARD.
- SUBTEST 6 - SPACE RECORDS.
- + REWIND.
  - + SPACE 7 RECORDS FORWARD.
  - + SPACE 7 RECORDS REVERSE.



- + SPACE 7 RECORDS FORWARD.
- + SPACE 7 RECORDS REVERSE.

SUBTEST 7 - WRITE RETRY.  
+ REWIND.  
+ WRITE DATA.  
+ WRITE RETRY.

SUBTEST 8 - READ REV RETRY.  
+ READ REVERSE.  
+ READ NEXT REVERSE.  
+ READ NEXT FORWARD.

SUBTEST 9 - READ FWD RETRY.  
+ READ FORWARD.  
+ READ PREVIOUS FORWARD.  
+ READ PREVIOUS REVERSE.

SUBTEST 10 - CLEAN.  
+ CLEAN.  
+ REWIND.

SUBTEST 11 - WRITE/VERIFY SWAPPED DATA BYTES.  
+ WRITE/VERIFY EVEN LENGTH (RECORD 1).  
+ WRITE/VERIFY ODD LENGTH (RECORD 2).  
+ SET DATA BYTE SWAP.  
+ WRITE/VERIFY EVEN LENGTH (RECORD 3).  
+ WRITE/VERIFY ODD LENGTH (RECORD 4).  
+ CLEAR DATA BYTE SWAP.

SUBTEST 12 - READ SWAPPED DATA BYTES.  
+ READ REV RECORD 4.  
+ READ REV RECORD 3.  
+ SET DATA BYTE SWAP.  
+ READ REV RECORD 2.  
+ READ REV RECORD 1.  
+ READ FWD RECORD 1.  
+ READ FWD RECORD 2.  
+ CLEAR DATA BYTE SWAP.  
+ READ FWD RECORD 3.  
+ READ FWD RECORD 4.

## 5.2 TEST 2 -

## DATA RELIABILITY.

1. THE TAPE IS INITIATED WITH THE FOLLOWING COMMANDS:  
SET CHARACTERISTIC 40  
REWIND  
WRITE/VERIFY 31 RECORDS OF RANDOM LENGTH AND DATA
2. WRITE AND READ COMMANDS ARE SELECTED AT RANDOM AND ARE EXECUTED A RANDOM NUMBER OF TIMES WITH RANDOM LENGTHS AND RANDOM PATTERN UNTIL END OF TAPE IS REACHED.
3. AT THE END OF EACH PASS, A REWIND COMMAND IS ISSUED AND A PERFORMANCE REPORT IS PRINTED.

NOTE: IF A RESTART COMMAND IS USED TO INITIATE TEST 1, THE INITIAL REWIND COMMAND IS NOT ISSUED.

## 5.3 TEST 3 -

## WRITE COMPATABILITY/WRITE UTILITY.

REWINDS AND WRITES RECORDS OF RANDOM LENGTHS AND RANDOM DATA FROM BOT TO EOT.

## 5.4 TEST 4 -

## READ COMPATABILITY/READ UTILITY.

REWINDS AND READS ENTIRE TAPE, FORWARD AND REVERSE.

## 5.5 TEST 5 -

## EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

THE SEQUENCE OF COMMANDS ENTERED BY THE OPERATOR IS EXECUTED. IF NO COMMANDS WERE ENTERED, A DEFAULT SEQUENCE OF REWIND/WRITE/READ REV/READ FWD/REWIND OF ENTIRE TAPE IS EXECUTED WITH RANDOM PATTERN AND RECORD LENGTH OF 2048 BYTES.



## 6.0 DEVICE INFORMATION TABLES

---

### 6.1 GENERAL

---

THE TS04 TAPE SUBSYSTEM CONSISTS OF A TS11 UNIBUS TO SERIAL BUS CONTROLLER CONNECTED TO A TS04 DRIVE. FROM A SOFTWARE VIEWPOINT THIS CONFIGURATION IS UNIQUE (FOR A UNIBUS DEVICE) IN A NUMBER OF WAYS:

- A. ONLY ONE REGISTER MAY BE WRITTEN - TSDB (TAPE SYSTEM DATA BUFFER),
- B. TWO REGISTERS MAY BE READ - TSSR AND TSBA (TAPE SYSTEM STATUS REGISTER AND TAPE SYSTEM BUS ADDRESS REGISTER),
- C. COMMANDS ARE NOT WRITTEN TO THE DRIVE; RATHER, COMMAND POINTERS ARE WRITTEN WHICH POINT TO COMMAND PACKETS SOMEWHERE IN CPU MEMORY. THE COMMAND POINTER IS USED BY THE TS04 SUBSYSTEM TO FETCH THE WORD(S) WITHIN THE COMMAND PACKET. THE WORDS WITHIN THE COMMAND PACKET ARE:
  - 1. COMMAND WORD
  - 2. LOW ORDER BUFFER ADDRESS
  - 3. HIGH ORDER BUFFER ADDRESS
  - 4. BYTE COUNT
- D. THE TSSR CONTAINS ALL THE INFORMATION WHICH WILL BE NECESSARY TO DETERMINE WHETHER:
  - 1. THE DRIVE IS READY TO ACCEPT ANOTHER COMMAND,
  - 2. THE PREVIOUS COMMAND WAS EXECUTED WITHOUT ERROR.IF EITHER OF THE ABOVE CONDITIONS IS UNTRUE AT "JOB DONE" OR "COMMAND INITIATION" TIME, IT MAY BE NECESSARY TO GET THE EXTENDED STATUS REGISTERS TO DETERMINE WHAT ACTION IS TO BE TAKEN AND/OR LOG THE ERROR INFORMATION.
- E. EXTENDED STATUS REGISTERS ARE NOT READ DIRECTLY FROM DRIVE REGISTERS; RATHER, A "GET STATUS" COMMAND IS ISSUED WHICH WILL CAUSE THE TS04 TO TRANSFER EXTENDED STATUS INFORMATION TO THE MEMORY AREA POINTED TO BY THE BUFFER ADDRESS OF THE "GET STATUS" COMMAND. THERE ARE FOUR EXTENDED STATUS REGISTERS. SEE 6.3.
- F. THE TSDB MUST BE WRITTEN WITH A DATO INSTRUCTION TO PROPERLY WRITE THE COMMAND POINTER. A DATOB WILL CAUSE A MAINTENANCE FUNCTION. A DATO TO THE TSSR WILL CAUSE SUBSYSTEM INIT.
- G. COMMAND PACKETS MUST RESIDE ON DIVIDE BY FOUR MEMORY BOUNDARIES (AS OPPOSED TO DIVIDE BY 2 OR WORD BOUNDARIES) .

6.2 UNIBUS INTERFACE SPECIFICATIONS

<u>TS11/ TS04</u>	<u>INT. VECTOR</u>	<u>UNIBUS ADDRESS</u>	<u>REGISTER</u>
FIRST	224	772520 772522	TSBA/TSDB TSSR
SECOND	154	772524 772526	TSBA/TSDB TSSR
THIRD	160	772530 772532	TSBA/TSDB TSSR
FOURTH	164	772534 772536	TSBA/TSDB TSSR



## 6.3 BIT DEFINITIONS FOR TS11/TS04 REGISTERS

### 6.3.1 TS11/TS04 REGISTER SUMMARY

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
TSBA	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00	
TSDB	P15	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P17	P16	
TSSR	SC	UPE	SPE	RMR	NXM	NBA	A17	A16	SSR	OFL	FC1	FC0	TC2	TC1	TC0		
XST0	TMK	RLS	LET	RLL	WLE	NEF	ILC	ILA	MOT	ONL	IE	VCK	PED	WLK	BOT	EOT	
XST1	DLT		COR	CRS	TIG	DBF	SCK		IPR	SYN	IPO	IED	POS	POL	UNC	MTE	
						NZO				DRP	ITM	LCO	NZN	LRC	CRC	VPE	
XST2	OPM	SIP	BPE	CAF		WCF		DTP	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
XST3	MICRO DIAGNOSTIC ERROR CODE								LMX	OPI		REV	CRF	DCK	NOI	LXS	RIB

#### TERMINATION CLASS CODES (TSSR TC0-TC2):

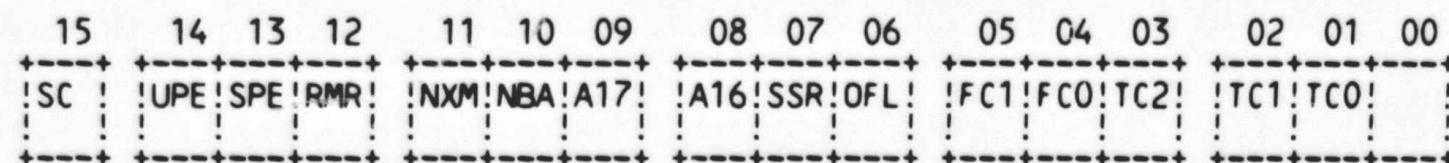
- 0 = NORMAL TERMINATION
- 1 = ATTENTION CONDITION
- 2 = TAPE STATUS ALERT
- 3 = FUNCTION REJECT
- 4 = RECOVERABLE ERROR - TAPE POSITION = ONE RECORD  
DOWN TAPE FROM START OF FUNCTION
- 5 = RECOVERABLE ERROR - TAPE NOT MOVED
- 6 = UNRECOVERABLE ERROR - TAPE POSITION LOST
- 7 = FATAL CONTROLLER ERROR

#### FATAL CLASS CODES (TSSR FC0-FC1):

- 0 = MICRO DIAGNOSTIC FAILURE (DISPLAYED IN TS04 OPERATOR PANEL AND XST3).
- 1 = I/O SEQUENCER CROM PARITY ERROR.
- 2 = MICROPROCESSOR CROM PARITY ERROR.  
SILO PARITY ERROR.  
SERIAL BUS PARITY ERROR DETECTED AT TS11 (SPE).  
SERIAL BUS PARITY ERROR DETECTED AT TS04 (BPE).  
FATAL ERROR HALTS 1750-1777 IN TS04 PROGRAM COUNTER DISPLAY.
- 3 = LOSS OF AC POWER HAS BEEN DETECTED.

## 6.3.2 TS11 STATUS REGISTER (TSSR)

UNIBUS ADDRESS + 2 - READ ONLY

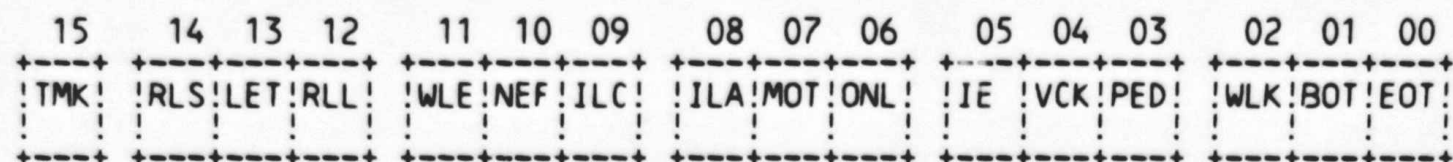


BIT	NAME	TCC	DEFINITION
15	SC	S	SPECIAL CONDITION. WHEN SET, INDICATES THAT THE LAST COMMAND DID NOT COMPLETE WITHOUT INCIDENT. SPECIFICALLY, EITHER AN ERROR WAS DETECTED OR AN EXCEPTION CONDITION OCCURRED. EXCEPTION CONDITIONS CAN BE TAPE MARKS ON READ COMMANDS, REVERSE MOTION AND AT BOT, EOT WHILE WRITING, ETC. MAY ALSO BE SET BY THE ERROR BITS CONTAINED IN THE TSSR REGISTER: UPE, SPE, RMR, AND NXM. THE TERMINATION CLASS BITS ARE SOMETHING OTHER THAN 0 (UNLESS RMR IS THE ONLY ERROR - SEE RMR).
14	UPE	4/5	UNIBUS PARITY ERROR. SET BY THE TS11 WHEN IT DETECTS A PARITY ERROR ON THE UNIBUS DATA WHEN TRANSFERRING TO OR FROM THE CPU'S MEMORY.
13	SPE	7	SERIAL BUS PARITY ERROR. THIS BIT IS SET BY THE TS11 WHEN IT DETECTS A SERIAL BUS PARITY ERROR ON DATA RECEIVED FROM THE TS04.
12	RMR	S	REGISTER MODIFICATION REFUSED. SET BY THE TS11 WHEN A COMMAND POINTER IS LOADED INTO TSDB AND SUB-SYSTEM READY (SSR) IS NOT SET. NOTE THAT THIS BIT CAUSES SPECIAL CONDITION BUT NO TERMINATION CLASS (IN FACT, THE TS04 NEVER SEES THIS ERROR) BECAUSE ON A SYSTEM WITH NO BUGS, THIS BIT MAY COME UP ON AN ATTENTION MESSAGE. IF ATTNS ARE NOT ENABLED, THIS BIT COMING UP IS AN INDICATION OF EITHER A FATAL CONTROLLER ERROR OR A SOFTWARE BUG.
11	NXM	4/5	NON-EXISTENT MEMORY. SET BY THE TS11 WHEN TRYING TO TRANSFER TO OR FROM A MEMORY LOCATION WHICH DOES NOT EXIST. MAY OCCUR WHEN FETCHING THE COMMAND PACKET, FETCHING OR STORING DATA, OR STORING THE MESSAGE PACKET.
10	NBA	S	NEED BUFFER ADDRESS. WHEN SET, INDICATES THAT THE TS04 NEEDS A MESSAGE BUFFER ADDRESS. THIS BIT IS CLEARED DURING THE SET CHARACTERISTICS COMMAND (IF A GOOD ADDRESS WAS GIVEN).
09	A17	S	BUS ADDRESS BIT 17. A17 AND A16 (BIT 08) TRACK THE VALUES OF BITS 17 AND 16 OF THE TSBA REGISTER.
08	A16	S	BUS ADDRESS BIT 16. SEE A17 (BIT 09).



07	SSR	S	SUB-SYSTEM READY. WHEN SET, INDICATES THAT THE TS11/TS04 SUBSYSTEM IS NOT BUSY AND IS READY TO ACCEPT A NEW COMMAND POINTER.
06	OFL	S,1,3	OFF-LINE. WHEN SET, INDICATES THAT THE TS04 IS OFF-LINE AND UNAVAILABLE FOR ANY TAPE MOTION COMMANDS. THIS BIT CAN CAUSE A TERMINATION CLASS OF 1 (ON ATTN INTERRUPT) OR 3 (RESULTS IN NEF).
05	FC1	7	FATAL TERMINATION CLASS 01. FC1 AND FC0 (BIT 04) ARE USED TO INDICATE THE TYPE OF FATAL ERROR WHICH HAS OCCURRED ON THE TS04. THESE BITS ARE VALID ONLY WHEN SC IS SET AND THE TERMINATION CLASS CODE BITS ARE ALL SET (111).
04	FC0	7	FATAL TERMINATION CLASS 00. SEE FC1 (BIT 05).
03	TC2	S	TERMINATION CLASS BIT 02. THIS BIT, ALONG WITH THE TC1 AND TC0 BITS, ACT AS AN OFFSET VALUE WHENEVER AN ERROR OR EXCEPTION CONDITION OCCURS ON A COMMAND. EACH OF THE EIGHT POSSIBLE VALUES OF THIS FIELD REPRESENT A PARTICULAR CLASS OF ERRORS OR EXCEPTIONS. THE CONDITIONS IN EACH CLASS HAVE SIMILAR SIGNIFICANCE AND, AS APPLICABLE, RECOVERY PROCEDURES. THE CODE PROVIDED IN THIS FIELD IS EXPECTED TO BE UTILIZED AS AN OFFSET INTO A DISPATCH TABLE FOR HANDLING OF THE CONDITION.
02	TC1	S	TERMINATION CLASS BIT 01. SEE TC2 (BIT 03).
01	TC0	S	TERMINATION CLASS BIT 00. SEE TC2 (BIT 03).
00	-	-	NOT USED.

UNIBUS ADDRESS + 2 - WRITE ONLY  
SUBSYSTEM INITIALIZE



BIT	NAME	TCC	DEFINITION
15	TMK	5,2	TAPE MARK DETECTED. SET WHENEVER A TAPE MARK WAS DETECTED DURING A READ, SPACE, OR SKIP COMMAND AND AS A RESULT OF THE WRITE TAPE MARK OR WITE TAPE MARK RETRY COMMANDS.
14	RLS	2	RECORD LENGTH SHORT. THIS BIT INDICATES THAT EITHER THE RECORD'S LENGTH WAS SHORTER THAN THE BYTE COUNT ON READ OPERATIONS, A SPACE RECORD OPERATION ENCOUNTERED A TAPE MARK OR BOT BEFORE THE POSITION COUNT WAS EXHAUSTED, OR A SKIP TAPE MARKS COMMAND WAS TERMINATED BY ENCOUNTERING BOT OR A DOUBLE TAPE MARK (IF THAT OPERATIONAL MODE IS ENABLED, SEE LET) PRIOR TO EXHAUSTING THE POSITION COUNTER.
13	LET	2	LOGICAL END OF TAPE. SET ONLY ON THE SKIP TAPE MARKS COMMAND WHEN EITHER TWO CONTIGUOUS TAPE MARKS ARE DETECTED OR WHEN MOVING OFF OF BOT AND THE FIRST RECORD ENCOUNTERED IS A TAPE MARK. THE SETTING OF THIS BIT WILL NOT OCCUR UNLESS THIS MODE OF TERMINATION IS ENABLED THROUGH USE OF THE SET CHARACTERISTICS COMMAND.
12	RLL	2	RECORD LENGTH LONG. WHEN SET, THIS BIT INDICATES THAT THE RECORD READ WAS LONGER THAN THE BYTE COUNT SPECIFIED.
11	WLE	3,6	WRITE LOCK ERROR. WHEN SET, INDICATES THAT A WRITE OPERATION WAS ISSUED BUT THE MOUNTED TAPE DID NOT CONTAIN A WRITE ENABLE RING OR THE WRT LOCK SWITCH ACTIVATED DURING THE OPERATION.
10	NEF	3	NON-EXECUTABLE FUNCTION. WHEN SET, INDICATES THAT THE COMMAND COULD NOT BE EXECUTED DUE TO ONE OF THE FOLLOWING CONDITIONS: <ul style="list-style-type: none"> <li>- THE COMMAND SPECIFIED REVERSE TAPE DIRECTION BUT THE TAPE WAS ALREADY POSITIONED AT BOT.</li> <li>- THE ISSUING OF ANY COMMAND, EXCEPT REWIND,</li> </ul>



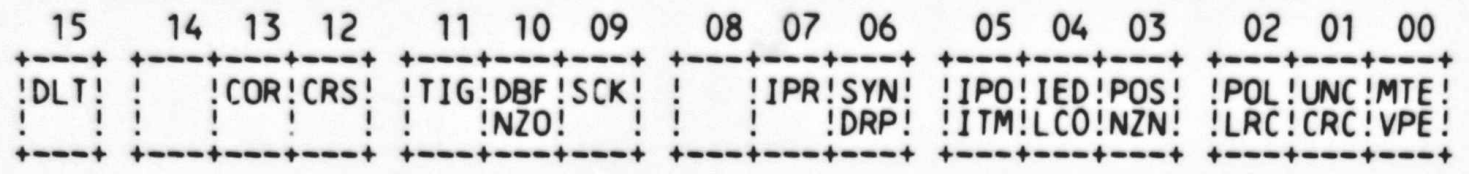
I 3

UNLOAD, OR A COMMAND WITH THE CLEAR VOLUME CHECK (CVC) BIT SET, WHEN THE VOLUME CHECK BIT IS SET.

- ANY COMMAND, EXCEPT GET STATUS OR DRIVE INITIALIZE, WHEN THE TS04 IS OFF-LINE.
- ANY WRITE COMMAND WHEN THE TAPE DOES NOT CONTAIN A WRITE ENABLE RING (WRITE LOCK STATUS - WLS).

09	ILC	3	ILLEGAL COMMAND. SET WHEN A COMMAND IS ISSUED AND EITHER ITS COMMAND FIELD OR ITS COMMAND MODE FIELD CONTAINS CODES WHICH ARE NOT SUPPORTED BY THE TS04.
08	ILA	3	ILLEGAL ADDRESS. (MORE THAN 18 BITS OR ODD WHEN AN EVEN ADDRESS IS REQUIRED.)
07	MOT	S	TAPE IS MOVING.
06	ONL	S	ON LINE. WHEN SET, INDICATES THAT THE TS04 IS ON-LINE AND OPERABLE.
05	IE	S	INTERRUPT ENABLE. REFLECTS THE STATE OF THE INTERRUPT ENABLE BIT SUPPLIED ON THE LAST COMMAND.
04	VCK	S	VOLUME CHECK. WHEN SET, INDICATES THAT THE DRIVE HAS BEEN EITHER POWERED DOWN OR TURNED OFF-LINE. CLEARED BY THE CLEAR VOLUME CHECK (CVC) BIT IN THE COMMAND HEADER WORD. THIS BIT CAN CAUSE A TERMINATION CLASS OF 3.
03	PED	S	PHASE ENCODED DRIVE. WHEN SET, INDICATES THAT THE TS04 IS CAPABLE OF READING AND WRITING ONLY 1600 BPI PHASE ENCODED DATA. WHEN RESET, INDICATES THAT THE TS04 HAS ONLY 800 BPI NRZI DATA CAPABILITIES.
02	WLK	S,3	WRITE LOCKED. WHEN SET, INDICATES THAT THE MOUNTED REEL OF TAPE DOES NOT HAVE A WRITE-ENABLE RING INSTALLED. THE TAPE IS, THEREFORE, WRITE PROTECTED.
01	BOT	S,3	BEGINNING OF TAPE. WHEN SET, INDICATES THAT THE TAPE IS POSITIONED AT THE LOAD POINT AS DENOTED BY THE BOT REFLECTIVE STRIP ON THE TAPE.
00	EOT	S,2	END OF TAPE. THIS BIT IS SET WHENEVER THE TAPE IS POSITIONED AT OR BEYOND THE END OF TAPE REFLECTIVE STRIP. DOES NOT RESET UNTIL THE TAPE PASSES OVER THE REFLECTIVE STRIP IN THE REVERSE DIRECTION UNDER PROGRAM CONTROL.

6.3.4 EXTENDED STATUS REGISTER 1 (XSTAT1)



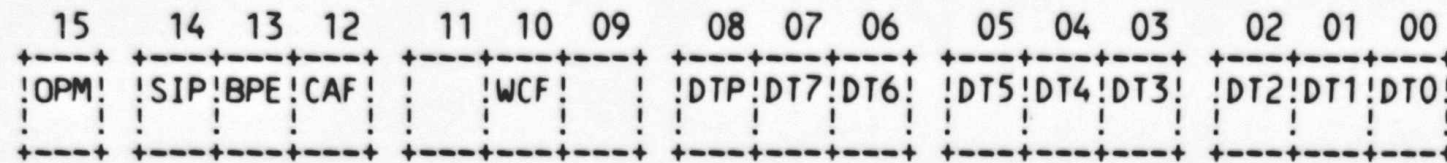
BIT	NAME	TCC	DEFINITION
15	DLT	4	DATA LATE. SET WHEN THE I/O SILO IS FULL ON A READ OR EMPTY ON A WRITE. THESE CONDITIONS OCCUR WHENEVER THE UNIBUS LATENCY EXCEEDS THE DATA TRANSFER RATE OF THE TS04.
14	-	-	NOT USED.
13	COR	S	CORRECTABLE DATA. IN PHASE ENCODED MODE, A CORRECTABLE DATA ERROR HAS BEEN ENCOUNTERED.
12	CRS	4	CREASE DETECTED. FOR NRZI, ALL DATA TRACKS DROPPED OUT FOR MORE THAN THREE CHARACTER TIMES BUT FOR LESS THAN .1 INCHES OF TAPE. FOR PE, EIGHT OUT OF NINE DATA TRACKS WENT DEAD FOR LESS THAN .1 INCHES BEFORE A VALID POSTAMBLE WAS DETECTED.
11	TIG	4	TRASH IN THE GAP. NON-ERASED DATA WAS DETECTED IN A GAP DURING A READ, WRITE, WRITE TAPE MARK, OR ERASE COMMAND.
10	DBF	4	DESKEW BUFFER FAIL. ONE OF THE DESKEW BUFFERS FAILED TO ASSERT 'OUTPUT READY' WITHIN 20 MICROSECONDS AFTER BEING ENABLED. THE DEAD TRACK BITS WILL INDICATE ON WHICH TRACKS THIS FAILURE OCCURRED. THIS ERROR IS PROBABLY A RESULT OF A BROKEN FORMATTER.
	NZO	4	NRZ FIFO OVERRUN.
09	SCK	4	SPEED CHECK. TAPE SPEED WAS OFF BY MORE THAN 5% DURING A WRITE DATA OPERATION. NOTE THAT SPEED AVERAGED OVER 8 TICKS AND THE AVERAGE MUST BE OFF 5% TO CAUSE THIS ERROR.
08	-	-	NOT USED.
07	IPR	S,4	INVALID PREAMBLE. SET ON A PE DRIVE IF THE PREAMBLE APPEARS TO BE SHORTER THAN 36 CHARACTERS OR LONGER THAN 44 CHARACTERS. ALSO SET IF THE PREAMBLE IS INCORRECTLY ENCODED BEYOND THE FIFTEENTH CHARACTER IN READ OR THE TENTH CHARACTER IN READ-AFTER-WRITE.
06	SYN	4	SYNCH FAILURE. SET ON A PE DRIVE IF THE FORMATTER WAS UNABLE TO ACHIEVE SYNCHRONIZATION IN THE PREAMBLE.
	DRP	4	NRZ RECORD DROPPED A CHARACTER (THE NEXT CHARACTER



WAS TO BE CONSIDERED CRC).

05	IPO	S,4	INVALID POSTAMBLE. SET ON A PE DRIVE DURING READ OR WRITE IF ANY OF THE FIRST 39 CHARACTERS OF THE POSTAMBLE ARE NOT READ CORRECTLY.
	ITM	S,4	ILLEGAL TAPE MARK FOR NRZ.
04	IED	4	INVALID END DATA. FOR PE, EIGHT OUT OF NINE TRACKS WENT DEAD BEFORE THE POSTAMBLE WAS DETECTED.
	LRO	4	FOR NRZI, DATA WAS NOT DETECTED IN EITHER THE LRCC OR CRCC WINDOWS. (LRC WAS ZERO)
03	POS	S,4	POSTAMBLE SHORT. SET ON PE DRIVES DURING A READ OR WRITE WHEN LESS THAN 38 ALL-ZEROES CHARACTERS ARE READ FOLLOWING THE ALL-ONES CHARACTER.
	NZN	S,4	NRZ NOISE RECORD (FEWER THAN 13(10) FRAMES).
02	POL	4	POSTAMBLE LONG. SET ON PE DRIVES DURING READ OR WRITE OPERATIONS WHEN THE POSTAMBLE EXCEEDS 42 CHARACTERS.
	LRC	4	LRC ERROR. SET ON NRZI DRIVES WHEN THE LRCC CHARACTER WAS FOUND IN ERROR.
01	UNC	4	UNCORRECTABLE DATA. SET ON PE DRIVES WHEN A PARITY ERROR OCCURRED WITHOUT A CORRESPONDING DEAD TRACK INDICATION.
	CRC	4	CRC ERROR. SET ON NRZI DRIVES WHEN THE CRC CHARACTER WAS FOUND TO BE IN ERROR.
00	MTE	4	MULTI-TRACK ERROR. SET ON PE DRIVES WHEN MORE THAN ONE DEAD TRACK OCCURRED IN THE PREAMBLE OR IN THE DATA FIELD.
	VPE	4	VERTICAL PARITY ERROR. SET ON NRZI DRIVES WHEN A CHARACTER DID NOT CONTAIN AN ODD NUMBER OF ONE BITS.

## 6.3.5 EXTENDED STATUS REGISTER 2 (XSTAT2)



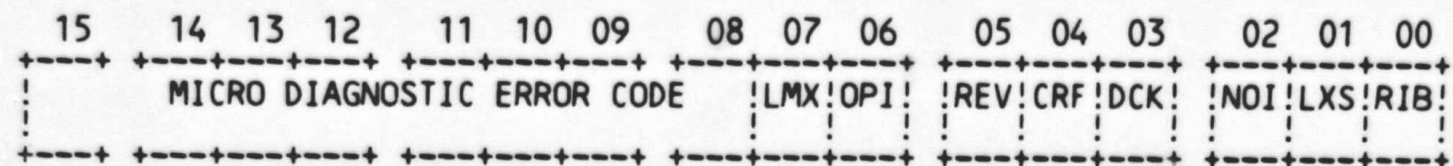
BIT	NAME	TCC	DEFINITION
15	OPM	S	OPERATION IN PROGRESS. (TAPE MOVING)
14	SIP	7,F2	SILO PARITY ERROR. CAUSES FATAL CLASS 2 BECAUSE THE ERROR MIGHT HAVE OCCURRED DURING THE TRANSMISSION OF THE MESSAGE PACKET.
13	BPE	7,F2	SERIAL BUS PARITY ERROR AT DRIVE. SET BY THE TS04 WHEN A PARITY ERROR IS DETECTED ON DATA TRANSMITTED FROM THE TS11 TO THE TS04. CAUSES FATAL CLASS 2 BECAUSE THE ERROR MIGHT HAVE OCCURRED DURING THE TRANSMISSION OF THE MESSAGE PACKET.
12	CAF	7	CAPSTAN ACCELERATION FAIL. AFTER ACCELERATING TAPE FOR .2 INCHES, THE TAPE SPEED WAS CHECKED AND FOUND TO BE OUT OF TOLERANCE BY MORE THAN 10%.
11	-	-	NOT USED.
10	WCF	7	THE WRITE BOARD IS NOT EMPTYING THE I/O SILO AT THE PROPER RATE. THIS ERROR CAN BE THE RESULT OF THE WRITE BOARD CLOCK NOT BEING TURNED ON (BROKEN HARDWARE).
09	-	-	NOT USED.
08	DTP	S	DEAD TRACK PARITY. THE BITS DTP THROUGH DT0 INDICATE WHICH TRACK(S) WENT DEAD, IF ANY, DURING THE LAST DATA TRANSFER OPERATION. IF DESKEW BUFFER FAIL (DBF) IS SET, THESE BITS INDICATE WHICH CHANNEL FAILED.
07	DT7	S	DEAD TRACK 7. SEE DTP.
06	DT6	S	DEAD TRACK 6. SEE DTP.
05	DT5	S	DEAD TRACK 5. SEE DTP.
04	DT4	S	DEAD TRACK 4. SEE DTP.
03	DT3	S	DEAD TRACK 3. SEE DTP.
02	DT2	S	DEAD TRACK 2. SEE DTP.
01	DT1	S	DEAD TRACK 1. SEE DTP.
00	DT0	S	DEAD TRACK 0. SEE DTP.



NOTE: ON A SET CHARACTERISTICS COMMAND, THE UCODE LEVEL IS RETURNED IN DT7 THRU DT0. ON A GET STATUS COMMAND, THE RESIDUAL CAPSTAN TICK COUNT (INTERNALLY R7) IS RETURNED IN DT7 THRU DT0.

6.3.6 EXTENDED STATUS REGISTER 3 (XSTAT3)

N 3



BIT	NAME	TCC	DEFINITION
15 TO 08			MICRO DIAGNOSTIC ERROR CODE. (SEE LIST OF CODES BELOW). ALL ERROR CODES IN THE TABLE WILL BE DISPLAYED ON THE TSO4 CONTROL PANEL BUT ONLY CODES HIGHER THAN 110 WILL BE AVAILABLE TO CPU DIAGNOSTICS FOR PRINTOUT IN THE MICRO DIAGNOSTIC ERROR CODE FIELD OF XSTAT3. THIS ERROR CODE FIELD IS VALID ONLY WHEN THE TERMINATION CLASS CODE IN THE TSSR EQUALS 7 AND THE FATAL CLASS CODE IN THE TSSR EQUALS 0, INDICATING AN INTERNAL DIAGNOSTIC FAILURE.
07	NTL	6	LIMIT EXCEEDED. SET WHEN THE TAPE TENSION ARMS HAVE EXCEEDED THEIR ALLOWABLE TRAVEL AND HAVE CAUSED THE ACTIVATION OF THE LIMIT SWITCHES. NO TENSION EXISTS ON THE MOUNTED TAPE.
06	OPI	6	OPERATION INCOMPLETE. SET WHEN A READ, SPACE, OR SKIP OPERATION HAS MOVED 25 FEET OF TAPE WITHOUT DETECTING ANY DATA ON THE TAPE.
05	REV	S	DIRECTION OF CURRENT OPERATION WAS REVERSE (BUT IS 0 IF REWIND OR FORWARD)
04	CRF	7	CAPSTAN RESPONSE FAILURE. A MOTION COMMAND WAS GIVEN TO THE CAPSTAN BUT WE DID NOT GET A TICK BACK WITHIN A REASONABLE AMOUNT OF TIME.
03	DCK	S,6	DENSITY CHECK. SET ON PE DRIVES WHEN A PE IDENTIFICATION BURST WAS NOT DETECTED WHEN MOVING OFF OF BOT. SET ON NRZI DRIVES WHEN A NON-NRZI IDENTIFICATION BURST WAS FOUND WHEN MOVING OFF OF BOT.
02	NOI	6	NOISE RECORD. SET DURING A READ OR SPACE OPERATION WHEN A BURST OF FLUX CHANGES, WHICH DO NOT QUALIFY AS A RECORD (BUT TOO MANY TO IGNORE), ARE DETECTED:  NRZI: AT LEAST TWO CHARACTERS IN A ROW BUT LESS THAN TWELVE, FOLLOWED BY A CHARACTER IN EITHER THE CRCC OR LRCC WINDOWS.  PE: AT LEAST 24 CHARACTERS IN A ROW THAT DO NOT QUALIFY AS A TAPE MARK OR A DATA PREAMBLE.
01	LXS	S	LIMIT EXCEEDED STATICALLY. THIS BIT IS SET ANY TIME THE LIMIT SWITCHES ARE EXCEEDED. THIS BIT CAN ONLY BE CLEARED BY MANUALLY LOADING THE TAPE.



COMMAND ALREADY IN PROGRESS HAS ENCOUNTERED THE BOT MARKER WHEN MOVING TAPE IN THE REVERSE DIRECTION. TAPE MOTION WILL BE HALTED AT BOT.

MICRO DIAGNOSTIC ERROR CODES

FOLLOWING IS A LIST OF THE ERRORS WHICH ARE DISPLAYED IN THE MICRO DIAGNOSTIC ERROR CODE (XSTAT3 BITS 15 - 08) AND ALSO IN THE LIGHTS ON THE TSO4 CONTROL PANEL, DUE TO FAILURES ON THE CAPSTAN BOARD, I/O BOARDS, WRITE BOARD, READ BOARD, OR FORMATTER BOARD. THE MICRO WILL BE IN A TIGHT LOOP IN THE DISPM PROGRAM, WAITING FOR OPERATOR OR CPU INTERVENTION WHILE THE ERROR IS BEING DISPLAYED IN THE CONSOLE LIGHTS. IT IS APPARENT THAT AN ERROR IS BEING DISPLAYED IF THE 'UOK' LIGHT IS NOT LIGHTED, THE PROCESSOR IS NOT STOPPED, AND AN OCTAL NUMBER (100-377) IS BEING DISPLAYED IN THE LIGHTS. TO SCOPE LOOP THESE TESTS, ENTER MAINTENANCE MODE (ON-LINE SWITCH TO 'OFF' POSITION, MAINTENANCE SWITCH UP, PRESS RESET), ENTER THE OFF-LINE TEST NUMBER (SEE SCOPE LOOP COLUMN BELOW) IN THE OPERATOR CONSOLE LIGHTS (ENTER ONES WITH LEFT-MOST SWITCH, ENTER ZEROES WITH RIGHT-MOST SWITCH), AND PRESS ON-LINE BUTTON. TEST WILL LOOP UNTIL ON-LINE SWITCH IS RETURNED TO OFF-LINE POSITION, ERRORS WILL BE DISPLAYED CONTINUOUSLY.

ERROR (DISPLAY)	PROGRAM	ERROR DESCRIPTION	LIKELY MODULE	SCOPE LOOP
337	OPERATIONAL CODE	CAPSTAN RUNAWAY ERROR (H3.RNY). CAPSTAN DIDN'T STOP WITHIN ACCEPTABLE WINDOW AFTER LAST COMMAND.		
100	IOTSM	BASIC I/O MICRO FAILURE (PARITY ERROR, IQATN, HANDSHAKING, AND DATA WINDOW TEST BETWEEN THE I/O AND MAIN MICROS.  NOTE: CAN ALSO BE CAUSED BY THE SERIAL BUS .SHIN (SHIFT IN) STUCK ASSERTED.	M8967	14
101	IOTSM	ERROR IN I/O CONTROL REGISTER TEST	M8966 M8967	15
102	IOTSM	FAILURE OF FRAME COUNTER TEST	M8966	15
103	IOTSM	FAILURE OF I/O SILO NON-PARITY ERROR DATA TEST OR THE WRITE FLAG.	M8966 M8963	16
104	IOTSM	FAILURE OF I/O SILO PARITY ERROR TEST OR DATA LATE TEST.	M8966	17
105	IOTSM	FAILURE OF SHIFT LOOP WITH ZEROES.	M8965	20
106	IOTSM	FAILURE OF SHIFT LOOP WITH ONES.	M8965	21
107	IOTSM	FAILURE OF SHIFT LENGTH MUX.	M8965	22
110	IOTSM	FAILURE TO RECEIVE CORRECT OP-CODE FROM TS11 WHEN WE SENT DATA OVER THE SERIAL BUS.	M8965 TS11 MOTHER BD	47

## SBUS CABLE

111	CATSM	FAILURE OF 1 KHZ CLOCK TEST. TSTS TAC SYNC FLOP AND ATTN, TOO.	G159 2 CBUS CABLE M8963
112	CATSM	LIGHT REGISTER CHANGED WHEN MOTION REGISTER WAS CLEARED.	G159 3,4
113	CATSM	FWD OR MVG BITS WRONG AFTER 1 TICK OF SIMULATED COMMAND AND TACH PULSES.	G159 3,4
114	CATSM	FAILURE OF SIMULATED CAPSTAN SPEED TEST. THE CAPSTAN SPEED COUNTER WAS OUT OF RANGE WHEN TAPE MOTION AT SPEED WAS SIMULATED.	G159 3,4
115	CATSM	FAILURE OF SIMULATED SLOW CAPSTAN TEST. SPEED COUNTER DID NOT LATCH UP WITH MAX COUNT WHEN SLOW TACH TICKS WERE SIMULATED.	G159 3,4
116	CATSM	FAILURE OF SIMULATED CAPSTAN DECEL TEST. COUNTER NOT ZERO FOR FORWARD OR 377 FOR REVERSE WHILE DECELERATING, OR MVG BIT NOT 1.	G159 3,4
117	CATSM	FAILURE OF MOVING FLOP TO GO TO ZERO AFTER STOPPING (DIRECTION REVERSAL FOR ONE TACH TICK).	G159 3,4
120	PETSM	FAILURE OF WRITE BOARD TO TURN ON AND EMPTY THE SILO, OR DATA LATE BIT DOESN'T WORK.	M8929 23 M8966
121	PETSM	FAILURE OF WRITE BOARD TO EMPTY SILO AT CORRECT SPEED.	M8929 23
124	PETSM	FORMATTER FLAG DOESN'T WORK ON THE M8922.	M8922 24
125	PETSM	FORMATTER SILO FILLING AND DATA ERROR	M8922 24 M8923 M8924
126	PETSM	PEAK SHIFT TEST ERROR	M8922 25 M8923 M8924
127	PETSM	FORMATTER TABLE LOOKUP ROM CHECKSUM TEST ERROR	M8922 26 M8923 M8924



3	PROGRAM HEADER
104	DISPATCH TABLE
121	DESCRIPTIVE TEXT
138	DEFAULT HARDWARE P-TABLE
157	SOFTWARE P-TABLE
217	GLOBAL EQUATES SECTION
492	GLOBAL DATA SECTION
802	GLOBAL TEXT SECTION
817	GLOBAL ERROR REPORT SECTION
958	GLOBAL SUBROUTINES SECTION
3396	REPORT CODING SECTION
3597	LOAD DEVICE PROTECTION TABLE
3610	INITIALIZE SECTION
3992	AUTO DROP SECTION
4118	CLEANUP CODING SECTION
4151	DROP UNIT SECTION
4186	ADD UNIT SECTION
4237	TEST 1: BASIC FUNCTIONS.
4690	TEST 2: DATA RELIABILITY.
4893	TEST 3: WRITE COMPATABILITY/WRITE UTILITY.
4961	TEST 4: READ COMPATABILITY/READ UTILITY.
5016	TEST 5: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
5246	HARDWARE PARAMETER CODING SECTION
5284	SOFTWARE PARAMETER CODING SECTION
5550	HARD CODED P-TBL

1



```

2      .TITLE PROGRAM HEADER AND TABLES
3      .SBTTL PROGRAM HEADER
4
5      .ENABL APS.AMA
6      =      2000
7      002000 002000 BGNMOD
8
9      :++
10     : THE PROGRAM HEADER IS THE INTERFACE BETWEEN
11     : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
12     :--
13
14     002000          POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU,BGNSETUP
15
16
17     002000          HEADER CZTSH,C,0,5000,1,#INTPRI
18     002000 L$NAME::          ;DIAGNOSTIC NAME
19     002000          103      .ASCII /C/
20     002001          132      .ASCII /Z/
21     002002          124      .ASCII /T/
22     002003          123      .ASCII /S/
23     002004          110      .ASCII /H/
24     002005          000      .BYTE 0
25     002006          000      .BYTE 0
26     002007          000      .BYTE 0
27     002010 L$REV::          ;REVISION LEVEL
28     002010          103      .ASCII /C/
29     002011 L$DEPO::          ;0
30     002011          060      .ASCII /O/
31     002012 L$UNIT::          ;NUMBER OF UNITS
32     002012          000001   .WORD T$PTHV
33     002014 L$TIML::          ;LONGEST TEST TIME
34     002014          005000   .WORD 5000
35     002016 L$HPCP::          ;POINTER TO H.W. QUES.
36     002016          025074   .WORD L$HARD
37     002020 L$SPCP::          ;POINTER TO S.W. QUES.
38     002020          025146   .WORD L$SOFT
39     002022 L$HPTP::          ;PTR. TO DEF. H.W. PTABLE
40     002022          002174   .WORD L$HW
41     002024 L$SPTP::          ;PTR. TO S.W. PTABLE
42     002024          002202   .WORD L$SW
43     002026 L$LADP::          ;DIAG. END ADDRESS
44     002026          026554   .WORD L$LAST
45     002030 L$STA::          ;RESERVED FOR APT STATS
46     002030          000000   .WORD 0
47     002032 L$CO::          ;
48     002032          000000   .WORD 0
49     002034 L$DTYP::          ;DIAGNOSTIC TYPE
50     002034          000001   .WORD 1
51     002036 L$APT::          ;APT EXPANSION
52     002036          000000   .WORD 0
53     002040 L$DTP::          ;PTR. TO DISPATCH TABLE
54     002040          002124   .WORD L$DISPATCH
55     002042 L$PRIO::          ;DIAGNOSTIC RUN PRIORITY
56     002042          000340   .WORD #INTPRI
57     002044 L$EXP1::          ;EXPANSION WORDS

```

58	002044	000000			.WORD	0
59	002046		L\$EXP2::			
60	002046	000000			.WORD	0
61	002050		L\$MREV::	;SVC REV AND EDIT #		
62	002050	003			.BYTE	C\$REVISION
63	002051	002			.BYTE	C\$EDIT
64	002052		L\$EF::	;DIAG. EVENT FLAGS		
65	002052	000000			.WORD	0
66	002054	000000			.WORD	0
67	002056		L\$SPC::			
68	002056	000000			.WORD	0
69	002060		L\$DEVP::	; POINTER TO DEVICE TYPE LIST		
70	002060	002164			.WORD	L\$DVTYP
71	002062		L\$REPP::	;PTR. TO REPORT CODE		
72	002062	016014			.WORD	L\$RPT
73	002064		L\$EXP4::			
74	002064	000000			.WORD	0
75	002066		L\$EXP5::			
76	002066	000000			.WORD	0
77	002070		L\$AUT::	;PTR. TO ADD UNIT CODE		
78	002070	021576			.WORD	L\$AU
79	002072		L\$DUT::	;PTR. TO DROP UNIT CODE		
80	002072	021524			.WORD	L\$DU
81	002074		L\$LUN::	; LUN FOR EXERCISERS TO FILL		
82	002074	000000			.WORD	0
83	002076		L\$DESP::	; POINTER TO DIAG. DESCRIPTION		
84	002076	002136			.WORD	L\$DESC
85	002100		L\$LOAD::	;GENERATE SPECIAL AUTOLOAD EMT		
86	002100	104035			EMT	E\$LOAD
87	002102		L\$ETP::	;POINTER TO ERR TBL		
88	002102	000000			.WORD	0
89	002104		L\$IICP::	;PTR. TO INIT CODE		
90	002104	017550			.WORD	L\$INIT
91	002106		L\$CCP::	;PTR. TO CLEAN-UP CODE		
92	002106	021462			.WORD	L\$CLEAN
93	002110		L\$ACP::	;PTR. TO AUTO CODE		
94	002110	021040			.WORD	L\$AUTO
95	002112		L\$PRT::	;PTR. TO PROTECT TABLE		
96	002112	017542			.WORD	L\$PROT
97	002114		L\$TEST::	;TEST NUMBER		
98	002114	000000			.WORD	0
99	002116		L\$DLY::	;DELAY COUNT		
100	002116	000000			.WORD	0
101	002120		L\$HIME::	;PTR. TO HIGH MEM		
102	002120	000000			.WORD	0
103						



```
104 .SBTTL DISPATCH TABLE
105
106 :++
107 : THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
108 : IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
109 :--
110
111 DISPATCH 5
112 002122 000005 .WORD 5
113 002124 L$DISPATCH::
114 002124 021672 .WORD T1
115 002126 023260 .WORD T2
116 002130 023734 .WORD T3
117 002132 024100 .WORD T4
118 002134 024232 .WORD T5
119
120
121 .SBTTL DESCRIPTIVE TEXT
122
123 :++
124 : 2 LINES OF TEXT PRINTED TO THE OPERATOR TO IDENTIFY THE DIAGNOSTIC AND THE DEVICE UNDER
125 :--
126
127 002136 DESCRIPT <DATA RELIABILITY TEST>
128 002136 L$DESC::
129 002136 040504 040524 051040 .ASCIZ /DATA RELIABILIT
130 002144 046105 040511 044502
131 002152 044514 054524 052040
132 002160 051505 000124
133
134 002164 DEVTYP <TS11> .EVEN
135 002164 L$DVTYP::
136 002164 051524 030461 000 .ASCIZ /TS11/
137 002172 .EVEN
```

138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156

002172  
002172 000002  
002174  
002174  
  
002174 172522  
002176 000224  
  
002200  
002200

.SBTTL DEFAULT HARDWARE P-TABLE  
  
:++  
: THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF  
: THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE  
: IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.  
:--  
  
BGNHW DFPTBL  
  
L\$HW::  
DFPTBL::  
  
172522 :TSSR ADDRESS.  
224 :VECTOR ADDRESS.  
  
ENDHW  
L10000:

.WORD L10000-L\$HW/2



```

157      .SBTTL  SOFTWARE P-TABLE
158
159      :++
160      : THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
161      : PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
162      :--
163
164      002200      BGNSW  SFPTBL
165      002200      000043
166      002202
167      002202
168
169      002202      001      CLRFLG:: .BYTE 1      ;CLEAR COUNTERS FLAG.
170      002203      000      RRANV:: .BYTE 0      ;RESET RANDOM VARIABLES EACH PASS FLAG.
171      002204      000      HAE:: .BYTE 0      ;HALT AFTER EACH COMMAND FLAG.
172      002205      000      ERCVER:: .BYTE 0      ;ENABLE RECOVERABLE ERROR PRINTS FLAG.
173      002206      001      BADTSW:: .BYTE 1      ;BAD TAPE SWITCH TO REWRITE ON SAME SPOT & DETECT BAD TA
174      002207      000      ;SPARE
175      002210      000      DINT:: .BYTE 0      ;DISABLE INTERRUPTS FLAG.
176      002211      000      IREC:: .BYTE 0      ;INHIBIT ERROR RECOVERY FLAG.
177      002212      000      CHGFLG:: .BYTE 0      ;CHANGE CMD SEQ TABLE FLAG.
178      002213      000      ;SPARE.
179      002214      000      PIRE:: .BYTE 0      ;INHIBIT RESIDUAL FRAMECOUNT ERROR REPORT FLAG.
180      002215      000      ;SPARE.
181      002216      000040      CHAR:: CH.EAI      ;CHARACTERISTICS CODE (DEFAULT = 40).
182      002220      000015      CMDD:: .WORD 13.      ;COMMAND 2 (DEFAULT = REWIND).
183      002222      000001      .WORD 1      ;BYTE COUNT
184      002224      000001      .WORD 1      ;NUMBER OF OPERATIONS
185      002226      000007      .WORD RANP      ;PATTERN
186      002230      000004      .WORD 4      ;COMMAND 3 (DEFAULT = WRITE)
187      002232      004000      .WORD DATCNT      ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
188      002234      076400      .WORD 32000.      ;NUMBER OF OPERATIONS (DEFAULT = 32000).
189      002236      000007      .WORD RANP      ;PATTERN (DEFAULT = RANDOM).
190      002240      000003      .WORD 3      ;COMMAND 4 (DEFAULT = READ REV).
191      002242      004000      .WORD DATCNT      ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
192      002244      076400      .WORD 32000.      ;NUMBER OF OPERATIONS (DEFAULT = 32,000).
193      002246      000007      .WORD RANP      ;PATTERN (DEFAULT = RANDOM).
194      002250      000002      .WORD 2      ;COMMAND 5 (DEFAULT = READ FWD).
195      002252      004000      .WORD DATCNT      ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
196      002254      076400      .WORD 32000.      ;NUMBER OF OPERATIONS (DEFAULT = 32,000).
197      002256      000007      .WORD RANP      ;PATTERN (DEFAULT = RANDOM).
198      002260      000015      .WORD 13.      ;COMMAND 6 (DEFAULT = REWIND).
199      002262      000001      .WORD 1      ;BYTE COUNT
200      002264      000001      .WORD 1      ;NUMBER OF OPERATIONS
201      002266      000007      .WORD RANP      ;PATTERN
202      002270      000033      .WORD 27.      ;END OF CMD SEQ TABLE CODE (DEF) OR CMD 7
203      002272      004000      .WORD DATCNT      ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
204      002274      076400      .WORD 32000.      ;NUMBER OF OPERATIONS (DEFAULT = 32000).
205      002276      000007      .WORD RANP      ;PATTERN (DEFAULT = RANDOM).
206      002300      000033      .WORD 27.      ;END OF CMD SEQ TABLE CODE (DEF) OR CMD 8
207      002302      004000      .WORD DATCNT      ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
208      002304      076400      .WORD 32000.      ;NUMBER OF OPERATIONS (DEFAULT = 32000).
209      002306      000007      .WORD RANP      ;PATTERN (DEFAULT = RANDOM).
210
211      002310      ENDSW
212      002310      L10001:

```

PROGRAM HEADER AND TABLES  
CZTSHC.P11 11-OCT-79 13:59

MACY11 30(1046) 11-OCT-79 14:02 K 4  
SOFTWARE P-TABLE PAGE 11

213  
214 002310

ENDMOD



```
215  
216 .TITLE GLOBAL AREAS  
217 .SBTTL GLOBAL EQUATES SECTION  
218  
219 002310 BGNMOD  
220  
221 :++  
222 : THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT  
223 : ARE USED IN MORE THAN ONE TEST.  
224 :--  
225  
226 002310 EQUALS  
227 :  
228 : BIT DIFINITIONS  
229 :  
230 100000 BIT15== 100000  
231 040000 BIT14== 40000  
232 020000 BIT13== 20000  
233 010000 BIT12== 10000  
234 004000 BIT11== 4000  
235 002000 BIT10== 2000  
236 001000 BIT09== 1000  
237 000400 BIT08== 400  
238 000200 BIT07== 200  
239 000100 BIT06== 100  
240 000040 BIT05== 40  
241 000020 BIT04== 20  
242 000010 BIT03== 10  
243 000004 BIT02== 4  
244 000002 BIT01== 2  
245 000001 BIT00== 1  
246 :  
247 001000 BIT9== BIT09  
248 000400 BIT8== BIT08  
249 000200 BIT7== BIT07  
250 000100 BIT6== BIT06  
251 000040 BIT5== BIT05  
252 000020 BIT4== BIT04  
253 000010 BIT3== BIT03  
254 000004 BIT2== BIT02  
255 000002 BIT1== BIT01  
256 000001 BIT0== BIT00  
257 :  
258 : EVENT FLAG DEFINITIONS  
259 : EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION  
260 :  
261 000040 EF.START== 32. ; START COMMAND WAS ISSUED  
262 000037 EF.RESTART== 31. ; RESTART COMMAND WAS ISSUED  
263 000036 EF.CONTINUE== 30. ; CONTINUE COMMAND WAS ISSUED  
264 000035 EF.NEW== 29. ; A NEW PASS HAS BEEN STARTED  
265 000034 EF.PWR== 28. ; A POWER-FAIL/POWER-UP OCCURRED  
266 :  
267 :  
268 : PRIORITY LEVEL DEFINITIONS  
269 :  
270 000340 PRI07== 340
```

```

271      000300      PRI06== 300
272      000240      PRI05== 240
273      000200      PRI04== 200
274      000140      PRI03== 140
275      000100      PRI02== 100
276      000040      PRI01== 40
277      000000      PRI00== 0
278
279      ;OPERATOR FLAG BITS
280
281      000004      EVL==      4
282      000010      LOT==      10
283      000020      ADR==      20
284      000040      IDU==      40
285      000100      ISR==     100
286      000200      UAM==     200
287      000400      BOE==     400
288      001000      PNT==    1000
289      002000      PRI==    2000
290      004000      IXE==    4000
291      010000      IBE==   10000
292      020000      IER==   20000
293      040000      LOE==   40000
294      100000      HOE==  100000
295
296
297      ; REGISTER USAGE.
298
299      :
300      :      R0 - PASSES PARAMETERS TO/FROM DIAGNOSTIC SUPERVISOR.
301      :      R1 - COMMAND SEQUENCE TABLE POINTER.
302      :      R2 - GENERAL PURPOSE REGISTER.
303      :      R3 - GENERAL PURPOSE REGISTER.
304      :      R4 - GENERAL PURPOSE REGISTER.
305      :      R5 - CURRENT LOGICAL DEVICE NUMBER X 2.
306      :      R6 - STACK POINTER.
307      :      R7 - PROGRAM COUNTER.
308
309      ;THE FOLLOWING ARE BIT DEFINITIONS FOR THE TSSR REGISTERS.
310      100000      TS.SC==100000      ;SPECIAL CONDITION BIT.
311      040000      TS.UPE==40000      ;UNIBUS PARITY ERROR
312      020000      TS.SPE==20000      ;SERIAL BUS PARITY ERROR.
313      010000      TS.RMR==10000      ;REGISTER MODIFICATION REFUSED.
314      004000      TS.NXM==4000      ;NON-EXISTENT MEMORY.
315      002000      TS.NBA==2000      ;NEED BUFFER ADDRESS.
316      001000      TS.A17==1000      ;BUS ADDRESS BIT 17.
317      000400      TS.A16==400      ;BUS ADDRESS BIT 16.
318      000200      TS.SSR==200      ;UNIT READY BIT.
319      000100      TS.OFL==100      ;OFF LINE.
320      177717      TSC.FCC==177717      ;FATAL CLASS CODE MASK.
321      177761      TSC.TCC==177761      ;TERMINATION CLASS CODE MASK.
    
```



```

322      ;THE FOLLOWING ARE BIT DEFINITIONS FOR THE COMMAND WORD
323
324      100000      ACK.C==100000      ;ACKNOWLEDGE BIT
325      040000      CVC.C==40000      ;CLEAR VOLUME CHECK.
326      020000      OPP.C==20000      ;OPPOSITE BIT
327      010000      SWB.C==10000      ;SWAP BYTE BIT
328      004000      MOD.C3==4000      ;MODE BIT 3
329      004000      BRP.C==4000      ;BYTE/RECORD/FILE COUNT FLAG BIT. NOT USED
330      ;BY TS04 BUT USED INTERNALLY BY THIS PROGRAM ONLY.
331      002000      MOD.C2==2000      ;MODE BIT 2
332      001000      MOD.C1==1000      ;MODE BIT 1
333      000400      MOD.C0==400      ;MODE BIT 0
334      000200      IE.C==200      ;INTERRUPT ENABLE
335      000100      FMT.C1==100      ;FORMAT BIT 1
336      000100      VFY.C==100      ;WRITE VERIFY FLAG BIT. INTERNAL USE ONLY.
337      ;NOT USED BY TS04.
338      000040      FMT.C0==40      ;FORMAT BIT 0.
339      000040      JMP.C==40      ;JUMP BIT-TO DIRECT THIS PROGRAM TO JUMP TO
340      ;A CERTAIN LOCATION IN THE COMMAND SEQUENCE
341      ;TABLE. INTERNAL USE ONLY.
342      000020      CMD.C4==20      ;COMMAND BIT 4
343      000020      DLY.C==20      ;INSERT DELAY. INTERNAL USE ONLY.
344      000010      CMD.C3==10      ;COMMAND BIT 3
345      000004      CMD.C2==4      ;COMMAND BIT 2
346      000002      CMD.C1==2      ;COMMAND BIT 1
347      000001      CMD.C0==1      ;COMMAND BIT 0
348
349      ; BIT DEFINITIONS FOR DEVICE CHARACTERISTICS.
350
351      000200      CH.ESS==200      ;ENABLE SKIP TAPE MARKS STOP (STOP AT LOGICAL EOT).
352      000040      CH.EAI==40      ;ENABLE ATTENTION INTERRUPTS.
353      000020      CH.ERI==20      ;ENABLE MESSAGE BUFFER RELEASE INTERRUPTS.
354      000040      DFTSCH==CH.EAI      ;DEFAULT CHARACTERISTICS CODE.
355
356      ;THE FOLLOWING INDICATES THE RELATIVE POSITIONS OF THE STATUS WORDS
357      ;IN THE MESSAGE BUFFER.
358
359      000004      MS.RFC==4      ;RESIDUAL FRAME COUNT.
360      000006      MS.XS0==6      ;EXT STATUS REG 0
361      000010      MS.XS1==10      ;EXT STATUS REG 1
362      000012      MS.XS2==12      ;EXT STATUS REG 2
363      000014      MS.XS3==14      ;EXT STATUS REG 3
364
365      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0.
366
367      100000      X0.TMK==100000      ;TAPE MARK.
368      040000      X0.RLS==40000      ;RECORD LENGTH SHORT.
369      020000      X0.LET==20000      ;LOGICAL EOT.
370      010000      X0.RLL==10000      ;RECORD LENGTH LONG.
371      000100      X0.ONL==100      ;ON LINE BIT.
372      000002      X0.BOT==2      ;BOT BIT.
373      000001      X0.EOT==1      ;EOT BIT.
374
375      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2.
376
377      100000      X2.OPM==100000      ;OPERATION IN PROGRESS, TAPE MOVING

```



```

378
379
380
381      000010      X3.DCK==10      ;DENSITY CHECK.
382      157400      X3.RNY==157400    ;CAPSTAN RUNAWAY UDIAG ERROR CODE.
383
384
385      ;THE FOLLOWING DEFINITIONS SHOW THE RELATIVE POSITIONS OF THE COMMAND
386      ;PACKET ENTRIES.
387
387      000000      CP.CMD==0      ;CMDPKT+0==TS04 COMMAND.
388      000002      CP.ADL==2      ;CMDPKT+2==BUFFER ADDRESS LOW.
389      000004      CP.ADH==4      ;CMDPKT+4==BUFFER ADDRESS HIGH.
390      000006      CP.CNT==6      ;CKDPKT+6==BYTE/FILE/RECORD COUNT
391
392      ;      MISCELLANEOUS DEFINITIONS.
393
394      000340      INTPRI==PRI07    ;PRIORITY TO BE USED IN INTERRUPT STATE.
395      002452      TSBA==TSDB      ;DATA BUFFER ADDRESS REGISTER.
396      000010      SCHCNT==10      ;ARBITRARY BYTE LENGTH FOR CHARACTERISTIC
397      ;BUFFER LENGTH. (EVEN #)
398      000016      MSGCNT==16      ;MESSAGE BUFFER LENGTH IN BYTES. (EVEN #)
399      003334      DIABLK==DATAWT  ;WRITE BUFFER ALSO USED FOR DIAG CMD.
400      000020      DIACNT==20      ;DIAGNOSTIC COMMAND BUFFER EXTENT.
401      004000      DATCNT==2048.   ;MAXIMUM RECORD LENGTH IN BYTES.
402      ;THIS COUNT SHOULD BE A MULTIPLE OF 256 TO INSURE
403      ;PROPER READ/WRITE BUFFER ALLOCATION BY THE SUPER.
404      000550      CNTLEN==CNTEND-CNTBGN ;LENGTH OF STATISTICAL COUNTER AREA.
405      177740      RNOPSC==177740  ;RANDOM # OF OPERATIONS MASK.
406      000007      RANP==7         ;CODE TO SELECT RANDOM PATTERN.
407      000020      RRECL==16.      ;READ RECOVERY ATTEMPT LIMIT.
408      000020      WRECL==16.      ;WRITE RECOVERY ATTEMPT LIMIT.
409      153624      RANBC==153624   ;CONSTANT USED TO RESET RANDOM # GENERATOR BASE.
410      032561      RANSC==32561    ;CONSTANT USED TO RESET RANDOM # SAVE LOCATION.
411      177774      NINUSE==177774 ;NOT IN USE CODE FOR DEVICE STATE TABLE.
412      177740      NCMD.C==ACK.C!CVC.C!OPP.C!SWB.C!MOD.C3!MOD.C2!MOD.C1!MOD.CO!IE.C!FMT.C1!FMT.CO
413      ;NOT 'COMMAND' BITS.
414
415      ;THE FOLLOWING DEFINES THE COMMAND WORD FOR EACH TS04 COMMAND.
416
417      100013      DRI==  ACK.C!CMD.C3!CMD.C1!CMD.CO    ;DRIVE INIT.
418
419
420      104001      RDF==  ACK.C!BRF.C!CMD.CO          ;READ FORWARD
421
422
423      104401      RDR==  ACK.C!BRF.C!MOD.CO!CMD.CO    ;READ REVERSE
424
425
426      104005      WRT==  ACK.C!BRF.C!CMD.CO!CMD.C2    ;WRITE COMMAND
427
428
429      104105      WTV==  ACK.C!BRF.C!VFY.C!CMD.CO!CMD.C2 ;WRITE VERIFY
430
431
432      104010      SRF==  ACK.C!BRF.C!CMD.C3          ;SPACE RECORD FORWARD
433

```



GLOBAL AREAS  
CZTSHC.P11MACY11 30(1046)  
11-OCT-79 13:5911-OCT-79 14:02 PAGE 16  
GLOBAL EQUATES SECTION

434			
435	104410	SRR==	ACK.C!BRF.C!MOD.CO!CMD.C3 ;SPACE RECORD REVERSE
436			
437			
438	105401	RNR==	ACK.C!BRF.C!MOD.C1!MOD.CO!CMD.CO ;READ REV RETRY1 - REREAD NEXT REVERSE, IE. SPACE FWD, READ REVERSE
439			
440			
441	125401	RNF==	ACK.C!BRF.C!OPP.C!MOD.C1!MOD.CO!CMD.CO ;READ REV RETRY2 - REREAD NEXT FORWARD, IE.READ FORWARD, SPACE REVERSE
442			
443			
444	105001	RPF==	ACK.C!BRF.C!MOD.C1!CMD.CO ;READ FWD RETRY1 - REREAD PREVIOUS FORWARD, IE. SPACE REVERSE, READ FORW
445			
446			
447	125001	RPR==	ACK.C!BRF.C!OPP.C!MOD.C1!CMD.CO ;READ FWD RETRY2 - REREAD PREVIOUS REVERSE, IE. READ REVERSE, SPACE FORW
448			
449			
450	105005	WRR==	ACK.C!MOD.C1!BRF.C!CMD.C2!CMD.CO ;WRITE RETRY
451			
452			
453	102010	RWD==	ACK.C!MOD.C2!CMD.C3 ;REWIND COMMAND
454			
455			
456	100012	MBR==	ACK.C!CMD.C3!CMD.C1 ;MESSAGE BUFFER RELEASE
457			
458			
459	100011	WTM==	ACK.C!CMD.C3!CMD.CO ;WRITE TAPE MARK.
460			
461			
462	101011	WTR==	ACK.C!MOD.C1!CMD.C3!CMD.CO ;WRITE TAPE MARK RETRY.
463			
464			
465	105010	SFF==	ACK.C!BRF.C!MOD.C1!CMD.C3 ;SPACE FILE FORWARD
466			
467			
468	105410	SFR==	ACK.C!BRF.C!MOD.CO!MOD.C1!CMD.C3 ;SPACE FILE REVERSE
469			
470			
471	100017	GES==	ACK.C!CMD.CO!CMD.C1!CMD.C2!CMD.C3 ;GET EXTENDED STATUS
472			
473			
474	100411	ERS==	ACK.C!MOD.CO!CMD.C3!CMD.CO ;ERASE 3 INCHES OF TAPE
475			
476			
477	100412	UNL==	ACK.C!MOD.CO!CMD.C3!CMD.C1 ;UNLOAD COMMAND
478			
479			
480	101012	CLN==	ACK.C!MOD.C1!CMD.C3!CMD.C1 ;ERASE TAPE.
481			
482			
483	140004	SCH==	ACK.C!CVC.C!CMD.C2 ;SET DEVICE CHARACTERISTICS.
484			
485	100006	DIA==	ACK.C!CMD.C2!CMD.C1 ;DIAGNOSTICS.
486			
487	000040	JMP==	JMP.C ;JUMP TO 'N'TH COMMAND
488			
489	000020	DLY==	DLY.C ;DELAY 'N' MS.

GLOBAL AREAS MACY11 30(1046) 11-OCT-79 14:02 PAGE 17  
CZTSHC.P11 11-OCT-79 13:59 GLOBAL EQUATES SECTION

490  
491

177777

END== 177777

;END OF COMMAND SEQUENCES



492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538

.SBTTL GLOBAL DATA SECTION

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

: COMMAND PACKET.

.CMDPKT:: = .+3&177774 ;MUST BE ON MOD 4 BOUNDARY.  
: 0 ;1ST WORD IS TSO4 COMMAND.  
: 0 ;2ND WORD IS THE BUFFER LOW ADDRESS.  
: 0 ;3RD WORD IS THE BUFFER HIGH ADDRESS.  
: 0 ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.

: GET STATUS COMMAND PACKET.

.GSCP:: = .+3&177774 ;MUST BE ON MOD 4 BOUNDARY.  
: .WORD GES

: MESSAGE BUFFER RELEASE COMMAND PACKET.

.BRCPK:: = .+3&177774 ;MUST BE ON MOD 4 BOUNDARY.  
: .WORD MBR

: REWIND COMMAND PACKET (USED IN ERROR RECOVERY ONLY)

.RWCPK:: = .+3&177774 ;MUST BE ON A MODULE 4 BOUNDARY.  
: .WORD RWD  
: .WORD 1

: WORK AREA FOR ANALYSIS OF MESSAGE PACKET CONTENTS.

.MSGPKT:: .BLKW 7 ;1ST WORD:: MESSAGE TYPE.  
: ;2ND WORD:: DATA FIELD LENGTH.  
: ;3RD WORD:: RESIDUAL FRAME COUNT.  
: ;4TH WORD:: XSTAT0  
: ;5TH WORD:: XSTAT1  
: ;6TH WORD:: XSTAT2  
: ;7TH WORD:: XSTAT3

```

539 ; MESSAGE PACKETS.
540
541 002352 000007 MSGPK0:: .BLKW 7 ;MESSAGE PACKET FOR DEVICE #0
542 002370 000007 MSGPK1:: .BLKW 7 ;MESSAGE PACKET FOR DEVICE #1
543 002406 000007 MSGPK2:: .BLKW 7 ;MESSAGE PACKET FOR DEVICE #2
544 002424 000007 MSGPK3:: .BLKW 7 ;MESSAGE PACKET FOR DEVICE #3
545
546 ; SET CHARACTERISTIC BLOCK.
547
548 002442 002352 SCHBK:: MSGPK0 ;1ST WORD:: MSGPKT ADDR LO(SET UP BY EXECUTE ROUTINE).
549 002444 000000 0 ;2ND WORD:: MSGPKT ADDR HI.
550 002446 000016 MSGCNT ;3RD WORD:: MSG BUFFER LENGTH (BYTES)
551 002450 000040 CH.EAI ;4TH WORD:: CHARACTERISTICS WORD(SET BY SETUP ROUTINE).
552
553 ; TS04 REGISTER ADDRESSES.
554
555 002452 000004 TSDB:: .BLKW 4 ;TS04 DATA BUFFER ADDRESSES.
556 002462 000004 TSSR:: .BLKW 4 ;TS04 STATUS REGISTER ADDRESSES.
557 002472 000004 TSVCT:: .BLKW 4 ;TS04 VECTOR ADDRESSES.
558
559 ; ADDRESSES OF MESSAGE PACKETS.
560
561 002502 002352 MSGPKA:: MSGPK0 ;DEVICE 0.
562 002504 002370 MSGPK1 ;DEVICE 1.
563 002506 002406 MSGPK2 ;DEVICE 2.
564 002510 002424 MSGPK3 ;DEVICE 3.
565
566 ; ADDRESSES OF INTERRUPT HANDLING ROUTINES.
567
568 002512 006316 TS4INT:: TS4IN0 ;DEVICE 0.
569 002514 006324 TS4IN1 ;DEVICE 1.
570 002516 006332 TS4IN2 ;DEVICE 2.
571 002520 006340 TS4IN3 ;DEVICE 3.
572
573 ; TS04 CODE LEVELS, WILL BE STORED AFTER SCH CMD IN BASIC FUNCTION TEST
574
575 002522 000000 TS4CL:: 0 ;DEVICE 0
576 002524 000000 0 ;DEVICE 1
577 002526 000000 0 ;DEVICE 2
578 002530 000000 0 ;DEVICE 3
579
580 ; UNIT NUMBERS OF ALL DEVICES BEING TESTED(1-4).
581 ; WHEN DEVICE IS NOT IN USE, IT,S LOCATION WILL = -3.
582 ; R5 WILL ALWAYS CONTAIN THE PRESENT LOGICAL UNIT NUMBER X 2.
583
584 002532 177774 DEVTBL:: .WORD NINUSE
585 002534 177774 .WORD NINUSE
586 002536 177774 .WORD NINUSE
587 002540 177774 .WORD NINUSE
588 002542 177777 .WORD END
589
590 ; BAD TAPE TABLE POINTER: USED BY WRITE RETRY ROUTINE
591 ; 'WRTY' TO LOG BAD TAPE SPOTS ON UNITS UNDER TEST
592
593
594 002544 002774 BTADDR:: BT0

```



GLOBAL AREAS MACY11 30(1046) 11-OCT-79 14:02 PAGE 20  
CZTSHC.P11 11-OCT-79 13:59 GLOBAL DATA SECTION

595	002546	003046	BT1
596	002550	003120	BT2
597	002552	003172	BT3

```

598 ; COUNTER AREA.
599
600 CNTBGN=.
601 002554 000020 WRBC:: .BLKW 20 ;BYTES WRITTEN.
602 002614 000020 RRBC:: .BLKW 20 ;BYTES READ REV.
603 002654 000020 RFBC:: .BLKW 20 ;BYTES READ FWD.
604 002714 000004 WRREC:: .BLKW 4 ;RECOVERABLE WRITE ERRORS.
605 002724 000004 WRUNR:: .BLKW 4 ;UNRECOVERABLE WRITE ERRORS.
606 002734 000004 RRREC:: .BLKW 4 ;RECOVERABLE READ REV ERRORS.
607 002744 000004 RRUNR:: .BLKW 4 ;UNRECOVERABLE READ REV ERRORS.
608 002754 000004 RFREC:: .BLKW 4 ;RECOVERABLE READ FWD ERRORS.
609 002764 000004 RFUNR:: .BLKW 4 ;UNRECOVERABLE READ FWD ERRORS.
610 002774 000025 BT0:: .BLKW 21. ;UNIT 0 BAT TAPE SPOTS LOG
611 003046 000025 BT1:: .BLKW 21. ;UNIT 1 BAT TAPE SPOTS LOG
612 003120 000025 BT2:: .BLKW 21. ;UNIT 2 BAT TAPE SPOTS LOG
613 003172 000025 BT3:: .BLKW 21. ;UNIT 3 BAT TAPE SPOTS LOG
614 003244 000004 WRTYCT:: .BLKW 4 ;WRITE RETRY COUNTER
615 003254 000004 PASCNT:: .BLKW 4 ;PASS COUNT.
616 003264 000004 SCCNT:: .BLKW 4 ;SPECIAL CONDITION COUNT.
617 003274 000004 VFYCNT:: .BLKW 4 ;COUNT OF TS04 DATA COMPARE ERRORS.
618 003304 000004 HRDCNT:: .BLKW 4 ;COUNT OF HARD ERRORS.
619 003314 000004 FTLCNT:: .BLKW 4 ;COUNT OF FATAL ERRORS.
620 003324 000004 CNTEND=. ;END OF STATICTICAL COUNTERS.
621 003324 000004 RECcnt:: .BLKW 4 ;NUMBER OF RECORDS FROM BOT: CLEARED ON REWIND
622 ; AND WHEN RESTARTING OR CONTINUING TEST 2.
623
624
625 ; THE FOLLOWING ARE THE DEFINITIONS OF VARIABLES
626 ; USED BY THE PROGRAM.
627
628 003334 000000 DATAWT:: .WORD 0 ;WRITE BUFFER ADDRESS.
629 003336 000000 DATARD:: .WORD 0 ;READ BUFFER ADDRESS.
630 003340 000000 NCNT:: .WORD 0 ;STORAGE FOR VALUE OF N.
631 003342 000000 NCNT1:: .WORD 0 ;TEMP STORAGE FOR VALUE OF N.
632 003344 000000 BRFCNT:: .WORD 0 ;STORAGE FOR BPCR VALUE.
633 003346 177777 CMDWRD:: .WORD END ;CONTAINS COMMAND WORD BEING EXECUTED PRESENTLY.
634 003350 177777 CMDSAV:: .WORD END ;SAVE LOCATION FOR CMD WORD DURING ERROR RECOVERY
635 003352 177777 PCMDWD:: .WORD END ;CONTAINS PREVIOUS COMMAND WORD.
636 003354 000000 CMDLG:: .WORD 0 ;CURRENT COMMAND LOGGING CODE.
637 003356 000000 LENMSK:: .WORD 0 ;RANDOM WRITE LENGTH MASK, TO BE SET UP BY TESTS
638 003360 153624 RANB:: .WORD 153624 ;RANDOM # GENERATOR BASE.
639 003362 032561 RANS:: .WORD 32561 ;RANDOM # SAVE LOCATION.
640 003364 000000 TIME1:: .WORD 0 ;TIME COUNT 1.
641 003366 000000 TIME2:: .WORD 0 ;TIME COUNT 2.
642 003370 000000 JLOOP:: .WORD 0 ;JMP COMMAND LOOP COUNT.
643 003372 000000 JLOC:: .WORD 0 ;JMP COMMAND LOCATION COUNT.
644 003374 000000 PATTERN:: .WORD 0 ;PATTERN SELECT CODE.
645 003376 000000 CTCC:: .WORD 0 ;CURRENT TERMINATION CLASS CODE.
646 003400 000000 R5SAVE:: .WORD 0 ;LOCATION FOR SAVING CURRENT DEVICE POINTER.
647 003402 000000 TSSREG:: .WORD 0 ;CURRENT STATUS REGISTER.

```



```

648 ; ERROR FLAG AREA, THESE FLAGS ARE CLEARED DURING INITIALIZATION AND
649 ; AFTER EACH COMMAND IS COMPLETED.
650
651 003404 BGNFLG=.
652 003404 000000 RETRYC:: .WORD 0 ;# OF RECOVERY ATTEMPTS EXECUTED.
653 003406 000 RPTCNT:: .BYTE 0 ;WRITE REPEAT ON SAME SPOT CNTR: 4 PER WRITE RETRY
654 003407 000 WRTYFG:: .BYTE 0 ;WRITE RETRY ON SAME SPOT IN PROGRESS FLAG
655 003410 000 WRTYER:: .BYTE 0 ;WRITE RETRY ON SAME SPOT ERROR FLAG
656 003411 000 RECLOG:: .BYTE 0 ;RECORD COUNT HAS BEEN UPDATED FOR THIS RECORD.
657 003412 000 ERLOG:: .BYTE 0 ;DATA BYTES AND ERRORS HAVE BEEN LOGGED FOR THIS RECORD.
658 003413 000 RWERR:: .BYTE 0 ;READ/WRITE ERROR HAS OCCURED.
659 003414 000 UNREC:: .BYTE 0 ;UNRECOVERABLE ERROR HAS OCCURED.
660 003415 000 ERRREC:: .BYTE 0 ;ERROR RECOVERY MODE.
661 .EVEN
662 003416 ENDERF=.
663
664 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED DURING INITIALIZATION.
665
666 003416 000004 INTFLG:: .BLKW 4 ;INTERRUPT OCCURRED FLAGS FOR EACH DEVICE.
667 003426 000004 EOTFLG:: .BLKW 4 ;EOT/BOT FLAGS FOR EACH DEVICE (XSTAT0).
668 003436 000000 BTPT:: .WORD 0 ;BAD TAPE SPOT POINTER TO BT0-BT3 VIA BTADDR
669 003440 000 EXPBOT:: .BYTE 0 ;BOT IS EXPECTED, DO NOT ABORT ON BOT/FUNC RTI.
670 003441 000 RANDOM:: .BYTE 0 ;RANDOM EVERYTHING FLAG.
671 003442 000 VFYFLG:: .BYTE 0 ;SET DURING WRITE/VERIFY COMMAND.
672 003443 000 RPTFLG:: .BYTE 0 ;PERFORMANCE REPORT HAS BEEN REQUESTED.
673 003444 000 SWBFLG:: .BYTE 0 ;ENABLES SWAP BYTE FUNCTION WHEN NOT EQUAL TO ZERO.
674 003445 000 IRE:: .BYTE 0 ;INHIBIT RESIDUAL FRAME COUNT ERROR REPORT.
675 003446 000 DROPED:: .BYTE 0 ;CURRENT UNIT HAS BEEN DROPPED
676 003447 000 T1SWB:: .BYTE 0 ;TEST1 SWAP BYTES FLAG
677 003450 000 ALLEOT:: .BYTE 0 ;ALL UNITS @ EOT FLAG
678 003451 000 ERSFLG:: .BYTE 0 ;ERASE FLAG: DO ERASE AFTER A SPACE REV TO DELETE
679 ;BADLY WRITTEN RECORD. 1 TO 4 ERASES LEAVING
680 ;A 3 TO 12 INCH GAP MAY RESULT.
681 .EVEN
682 003452 ENDFLG=.
683
684 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED ONLY AFTER BEING CHECKED.
685
686 003452 000 STAFLG:: .BYTE 0 ;START FLAG - SET BY INIT CODE IF STARTING.
687 003453 000 PWRFLG:: .BYTE 0 ;POWER FAILURE FLAG - SET ONLY DURING INIT.
688 003454 000 TRAPD4:: .BYTE 0 ;TRAPED AT 4 FLAG
689 003455 000 MISCFG:: .BYTE 0 ;MISCELLANEOUS FLAG
690
691 ; OPERATOR FLAG SETTINGS PASSED BY DIAG. SUPERVISOR IN A 16 BIT WORD
692 ; SEE GLOBAL EQUATES SECTION FOR FLAG BIT LIST
693
694 003456 000000 OPFLAG:: .WORD 0 ;READ ONLY OPERATOR FLAG WORD
695 .EVEN

```

```

696                                     :THE FOLLOWING IS THE COMMAND SEQUENCE TABLE. THE TABLE
697                                     :HAS DEFAULT VALUES AT PROGRAM LOAD AS SHOWN. THESE VALUES
698                                     :CAN BE UPDATED BY A TEST OR BY OPERATOR INPUT.
699
700 003460 140004      CMDSEQ:: .WORD SCH           ;SET CHARACTERISTICS.
701 003462 000040      .WORD CH.EAI
702 003464 000001      .WORD 1
703 003466 000000      .WORD 0
704 003470 102010      CMDSE2:: .WORD RWD           ;REWIND.
705 003472 000001      .WORD 1                   ;BYTE COUNT.
706 003474 000001      .WORD 1                   ;ONCE.
707 003476 000007      .WORD RANP                ;PATTERN.
708 003500 104005      .WORD WRT                 ;WRITE.
709 003502 004000      .WORD DATCNT              ;MAX BUFFER LENGTH.
710 003504 076400      .WORD 2000.              ;32,000 RECORDS.
711 003506 000007      .WORD RANP                ;RANDOM PATTERN.
712 003510 104401      .WORD RDR                 ;READ REV.
713 003512 004000      .WORD DATCNT              ;MAX BUFFER LENGTH.
714 003514 076400      .WORD 32000.             ;32,000 RECORDS
715 003516 000007      .WORD RANP                ;RANDOM PATTERN.
716 003520 104001      .WORD RDF                 ;READ FWD.
717 003522 004000      .WORD DATCNT              ;MAX BUFFER LENGTH.
718 003524 076400      .WORD 32000.             ;32,000 RECORDS.
719 003526 000007      .WORD RANP                ;RANDOM PATTERN.
720 003530 102010      .WORD RWD                 ;REWIND.
721 003532 000001      .WORD 1                   ;BYTE COUNT.
722 003534 000001      .WORD 1                   ;ONCE.
723 003536 000007      .WORD RANP                ;PATTERN.
724 003540 000004      .BLKW 4                   ;EXTENSTION TO HOLD 1 MORE CMD.
725 003550 177777      SEQEND:: .WORD END         ;SOFT END OF SEQUENCE TABLE.
726 003552 177777      .WORD END
727 003554 177777      .WORD END
728 003556 177777      .WORD END
729 003560 177777      .WORD END                 ;HARD END OF SEQUENCE TABLE.

```



;THE FOLLOWING IS THE TS04 COMMAND TABLE

Line	Code	Address	Command	Description
730				
731				
732	003562	100013	CMDTBL:: .WORD DRI	:DRIVE INIT.
733	003564	104001	.WORD RDF	:READ FORWARD.
734	003566	104401	.WORD RDR	:READ REVERSE.
735	003570	104005	.WORD WRT	:WRITE
736	003572	104105	.WORD WTV	:WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND
737				:CHECK DATA ON ALL RECORDS, RDF AND
738				:CHECK DATA ON ALL RECORDS.)
739	003574	104010	.WORD SRF	:SPACE 'N' RECORDS FORWARD.
740	003576	104410	.WORD SRR	:SPACE 'N' RECORDS REVERSE.
741	003600	105401	.WORD RNR	:READ NEXT REVERSE. I.E., SPACE FWD, READ REVERSE.
742	003602	125401	.WORD RNF	:READ NEXT FORWARD, I.E., READ FORWARD, SPACE REVERSE.
743	003604	105001	.WORD RPF	:READ PREVIOUS FORWARD. I.E., SPACE REVERSE, READ FORWAR
744	003606	125001	.WORD RPR	:READ PREVIOUS REVERSE. I.E., READ REVERSE, SPACE FORWAR
745	003610	105005	.WORD WRR	:WRITE RETRY.
746	003612	102010	.WORD RWD	:REWIND.
747	003614	100012	.WORD MBR	:MESSAGE BUFFER RELEASE
748	003616	100011	.WORD WTM	:WRITE TAPE MARK
749	003620	101011	.WORD WTR	:WRITE TAPE MARK RETRY.
750	003622	105010	.WORD SFF	:SPACE 'N' FILES FORWARD.
751	003624	105410	.WORD SFR	:SPACE 'N' FILES REVERSE.
752	003626	100017	.WORD GES	:GET EXTENDED STATUS.
753	003630	100411	.WORD ERS	:ERASE 3 INCHES OF TAPE.
754	003632	100412	.WORD UNL	:REWIND AND UNLOAD.
755	003634	101012	.WORD CLN	:CLEAR TAPE.
756	003636	140004	.WORD SCH	:SET CHARACTERISTICS.
757	003640	100006	.WORD DIA	:DIAGNOSTIC COMMAND.
758	003642	000040	.WORD JMP	:JUMP TO THE NTH COMMAND IN THE SEQUENCE.
759	003644	000020	.WORD DLY	:DELAY 'N' MS.
760	003646	177777	.WORD END	:END OF COMMAND TABLE
761				

762  
 763  
 764 003650 051104 111  
 765 003653 122 043104  
 766 003656 042122 122  
 767 003661 127 052122  
 768 003664 052127 126  
 769  
 770 003667 123 043122  
 771 003672 051123 122  
 772 003675 122 051116  
 773 003700 047122 106  
 774 003703 122 043120  
 775 003706 050122 122  
 776 003711 127 051122  
 777 003714 053522 104  
 778 003717 115 051102  
 779 003722 052127 115  
 780 003725 127 051124  
 781 003730 043123 106  
 782 003733 123 051106  
 783 003736 042507 123  
 784 003741 105 051522  
 785 003744 047125 114  
 786 003747 103 047114  
 787 003752 041523 110  
 788  
 789 003755 104 040511  
 790  
 791  
 792 003760 046512 120  
 793  
 794  
 795 003763 104 054514  
 796  
 797 003766 047105 104  
 798 003772  
 799  
 800  
 801

: THE FOLLOWING TABLE CONTAINS THE ASCII FOR EACH COMMAND.

CMDASC:: .ASCII /DRI/ :DRIVE INIT.  
 .ASCII /RDF/ :READ FORWARD.  
 .ASCII /RDR/ :READ REVERSE.  
 .ASCII /WRT/ :WRITE  
 .ASCII /WTV/ :WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND CHECK DATA  
 :ON ALL RECORDS, RDF AND CHECK DATA ON ALL RECORDS.)  
 .ASCII /SRF/ :SPACE 'N' RECORDS FORWARD.  
 .ASCII /SRR/ :SPACE 'N' RECORDS REVERSE.  
 .ASCII /RNR/ :READ NEXT REVERSE. I.E., SPACE FWD READ REVERSE.  
 .ASCII /RNF/ :READ NEXT FORWARD, I.E., READ FORWARD, SPACE REVERSE.  
 .ASCII /RPF/ :READ PREVIOUS FORWARD. IE., SPACE REVERSE, READ FORWARD  
 .ASCII /RPR/ :READ PREVIOUS REVERSE. IE., READ REVERSE, SPACE FORWARD  
 .ASCII /WRR/ :WRITE RETRY.  
 .ASCII /RWD/ :REWIND.  
 .ASCII /MBR/ :MESSAGE BUFFER RELEASE  
 .ASCII /WTM/ :WRITE TAPE MARK  
 .ASCII /WTR/ :WRITE TAPE MARK RETRY.  
 .ASCII /SFF/ :SPACE 'N' FILES FORWARD.  
 .ASCII /SFR/ :SPACE 'N' FILES REVERSE.  
 .ASCII /GES/ :GET EXTENDED STATUS.  
 .ASCII /ERS/ :ERASE 3 INCHES OF TAPE.  
 .ASCII /UNL/ :REWIND AND UNLOAD.  
 .ASCII /CLN/ :CLEAN TAPE.  
 .ASCII /SCH/ :SET CHARACTERISTICS. WHERE BRF=200, 40, 20, 0.  
 :SEE TS11/TS04 PROGRAMMING SPECIFICATION FOR DESCRIPTION  
 :DIAGNOSTICS. SEE TS11/TS04 PROGRAMMING SPECIFICATION  
 :FOR DESCRIPTION. ODT MUST BE USED TO LOAD DIAGNOSTIC D  
 :INTO THE WRITE BUFFER BEFORE THIS CMD IS ISSUED.  
 .ASCII /JMP/ :JUMP TO THE NTH COMMAND IN THE COMMAND  
 :SEQUENCE TABLE, WHERE N IS DEFINED IN  
 :THE # OF OPERATIONS.  
 .ASCII /DLY/ :DELAY 'N' MS, WHERE N IS DEFINED IN  
 :THE # OF OPERATIONS.  
 .ASCII /END/ :END OF COMMAND SEQUENCE.  
 .EVEN



802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815

.SBTTL GLOBAL TEXT SECTION

;++  
: THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,  
: MESSAGES, AND ASCII INFORMATION THAT ARE USED IN  
: MORE THAN ONE TEST.  
:--

:  
: FORMAT STATEMENTS USED IN PRINT CALLS  
:

.NLIST BEX

```
003772 047045 040445 047125 CODELM:: .ASCIZ /%N%UNIT %D1%A TS11 CODE LEVEL P%03%N%N/  
                                .EVEN  
004042 054130 020130 046503 HALTM:: .ASCIZ /XXX CMD - TYPE <CR> TO CONTINUE/  
004102 046503 020104 040520 CMDPKM:: .ASCIZ /CMD PACKET ADR NOT ON MODULO 4 BOUNDARY: RELOAD!/  
                                .EVEN  
004164 040504 040524 041440 WTVERM:: .ASCIZ /DATA COMPARE ERROR/  
004207 116 020117 051524 TOERM:: .ASCIZ /NO TS11 RESPONSE/  
004230 047125 042504 044506 SCERM:: .ASCIZ /UNDEFINED SPEC COND/  
004254 043122 020103 047516 RFCERM:: .ASCIZ /RFC NON ZERO/  
004271 124 030523 020061 NSSRM:: .ASCIZ /TS11 NOT READY/  
004310 042522 051124 020131 RLEXM:: .ASCIZ /RETRY LIMIT EXCEEDED/  
004335 125 044516 020124 ATTNM:: .ASCIZ /UNIT OFF LINE/  
004353 106 047125 052103 FUNRM:: .ASCIZ /FUNCTION REJECT/  
004373 106 052101 046101 FATSM:: .ASCIZ /FATAL SUBSYSTEM ERROR/  
004421 116 020117 047111 NOINTM:: .ASCIZ /NO INTERRUPT/  
004436 040524 042520 051440 TSAM:: .ASCIZ /TAPE STATUS ALERT/  
004460 047524 020117 040515 TOOMM:: .ASCIZ /TOO MANY INTERRUPTS/  
004504 040503 051520 040524 RNYM:: .ASCIZ /CAPSTAN RUNAWAY-GET STATUS RESULTS:/  
004550 042522 047503 042526 RERM:: .ASCIZ /RECOVERABLE ERROR/  
004572 047125 042522 047503 URERM:: .ASCIZ /UNRECOVERABLE ERROR/  
004616 047045 040445 051104 DROPDM:: .ASCIZ /%N%ADROPPED UNIT %D1%N/  
004645 045 022516 040501 AUDRPM:: .ASCIZ /%N%AALL UNITS DROPPED%N%N/  
004677 045 022516 041101 DTAER2:: .ASCIZ '%N%ABYTE:%D4%S2%AWAS:%B8%S2%AS/B:%B8%N'  
004746 042045 022464 020101 DTAER3:: .ASCIZ '%D4%A BYTES IN ERROR OUT OF %D4%N'  
005010 040445 047516 042040 DTAER4:: .ASCIZ /%ANO DATA READ%N/  
005031 045 051101 041505 DTAER5:: .ASCIZ /%ARECORD TOO LONG: >%04%A BYTES%N/  
005073 045 051101 041505 NURTY1:: .ASCIZ /%ARECOVERED ON RETRY #%D2%N/  
005127 045 052501 044516 OFLINM:: .ASCIZ /%AUNIT %D1%A OFF LINE%N/  
005157 045 043501 052105 GETSTM:: .ASCIZ /%AGET STATUS CMD RESULTS:%N/  
005213 045 000116 CRLF:: .ASCIZ /%N/  
005216 047045 051445 000067 CRLFSP:: .ASCIZ /%N%S7/  
                                .LIST BEX  
                                .EVEN
```

816

817  
 818  
 819  
 820  
 821  
 822  
 823  
 824  
 825  
 826  
 827  
 828  
 829  
 830  
 831  
 832  
 833  
 834  
 835  
 836  
 837  
 838  
 839  
 840  
 841  
 842  
 843  
 844  
 845  
 846  
 847  
 848  
 849  
 850  
 851  
 852  
 853  
 854  
 855  
 856  
 857  
 858  
 859  
 860  
 861  
 862  
 863  
 864  
 865  
 866  
 867  
 868  
 869  
 870  
 871  
 872

005224  
 005224  
 005224 016546 003324  
 005230 016546 003254  
 005234 016546 002532  
 005240 012746 005704  
 005244 012746 000004  
 005250 010600  
 005252 104414  
 005254 062706 000012  
 005260  
 005260 012746 005776  
 005264 012746 000001  
 005270 010600  
 005272 104414  
 005274 062706 000004  
 005300  
 005300 010237 006312  
 005304  
 005304 010337 003364  
 005310  
 005310 010437 003366  
 005314 004737 006346  
 005320  
 005320 013702 006312  
 005324  
 005324 010337 006312  
 005330  
 005330 013703 003364  
 005334  
 005334 013704 003366  
 005340  
 005340 013746 006312  
 005344 012746 006026  
 005350 012746 000002  
 005354 010600  
 005356 104414  
 005360 062706 000006  
 005364  
 005364 000167  
 005366 000000  
 005370  
 005370  
 005370 104423

.SBTTL GLOBAL ERROR REPORT SECTION

```

:++
: THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
: THAT ARE USED IN MORE THAN ONE TEST. IT ALSO INCLUDES THE ASCII MESSAGES
: THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
:--
  
```

```

BGNMSG DTAERM
DTAERM::
  PRINTB #STAER1,DEVTL(R5),PASCNT(R5),RECCNT(R5)
  
```

```

  PRINTB #STAER7
  
```

```

  LET RECD := R2          ;SAVE R2
  LET TIME1 := R3        ;SAVE R3
  LET TIME2 := R4        ;SAVE R4
  JSR PC,RECTAP          ;RETRIEVE RECORD READ
  LET R2 := RECD         ;RESTORE R2
  LET RECD := R3        ;SAVE RECORD READ
  LET R3 := TIME1       ;RESTORE R3
  LET R4 := TIME2       ;RESTORE R4
  PRINTB #STAER6,RECD    ;PRINT RECORD READ
  
```

```

  EXIT MSG
  
```

```

  .EVEN
  
```

```

  ENDMSG
  
```

L10002:

```

  MOV RECCNT(R5),-(SP)
  MOV PASCNT(R5),-(SP)
  MOV DEVTL(R5),-(SP)
  MOV #STAER1,-(SP)
  MOV #4,-(SP)
  MOV SP,R0
  TRAP C$PNTB
  ADD #12,SP

  MOV #STAER7,-(SP)
  MOV #1,-(SP)
  MOV SP,R0
  TRAP C$PNTB
  ADD #4,SP

  MOV R2,RECD
  MOV R3,TIME1
  MOV R4,TIME2

  MOV RECD,R2
  MOV R3,RECD
  MOV TIME1,R3
  MOV TIME2,R4

  MOV RECD, -(SP)
  MOV #STAER6, -(SP)
  MOV #2, -(SP)
  MOV SP, R0
  TRAP C$PNTB
  ADD #6, SP

  .WORD JSJMP
  .WORD L10002-2-.

  TRAP C$MSG
  
```



873							
874	005372			BGNMSG STAERM			
875	005372			STAERM::			
876	005372			PRINTB #STAER1,DEVTBL(R5),PASCNT(R5),RECCNT(R5)			
877	005372	016546	003324			MOV	RECCNT(R5),-(SP)
878	005376	016546	003254			MOV	PASCNT(R5),-(SP)
879	005402	016546	002532			MOV	DEVTBL(R5),-(SP)
880	005406	012746	005704			MOV	#STAER1,-(SP)
881	005412	012746	000004			MOV	#4,-(SP)
882	005416	010600				MOV	SP,R0
883	005420	104414				TRAP	C\$PNTB
884	005422	062706	000012			ADD	#12,SP
885	005426			PRINTB #STAER7			
886	005426	012746	005776			MOV	#STAER7,-(SP)
887	005432	012746	000001			MOV	#1,-(SP)
888	005436	010600				MOV	SP,R0
889	005440	104414				TRAP	C\$PNTB
890	005442	062706	000004			ADD	#4,SP
891	005446			LET R2 := CMDPKT CLR.BY #177740			
892	005446	013702	002310			MOV	CMDPKT,R2
893	005452	042702	177740			BIC	#177740,R2
894	005456			LET R2 := R2 - #1			
895	005456	005302				DEC	R2
896	005460			IF R2 EQ #0 THEN	:IF CMD IS A READ		
897	005460	005702				TST	R2
898	005462	001016				BNE	50000\$
899	005464	004737	006346	JSR PC,RECTAP	:THEN RETRIEVE		
900	005470			LET RECD := R3	:AND		
901	005470	010337	006312			MOV	R3,RECD
902	005474			PRINTB #STAER6,RECD	:TYPE RECORD READ		
903	005474	013746	006312			MOV	RECD, -(SP)
904	005500	012746	006026			MOV	#STAER6, -(SP)
905	005504	012746	000002			MOV	#2, -(SP)
906	005510	010600				MOV	SP,R0
907	005512	104414				TRAP	C\$PNTB
908	005514	052706	000006			ADD	#6,SP
909	005520			ENDIF			
910	005520						50000\$:
911	005520			PRINTX #STAER2			
912	005520	012746	006062			MOV	#STAER2, -(SP)
913	005524	012746	000001			MOV	#1, -(SP)
914	005530	010600				MOV	SP,R0
915	005532	104415				TRAP	C\$PNTX
916	005534	062706	000004			ADD	#4,SP
917	005540			PRINTX #STAER3,CMDPKT,@TSDB(R5),MSGPKT+MS.RFC,TSSREG,CTCC			
918	005540	013746	003376			MOV	CTCC, -(SP)
919	005544	013746	003402			MOV	TSSREG, -(SP)
920	005550	013746	002340			MOV	MSGPKT+MS.RFC, -(
921	005554	017546	002452			MOV	@TSDB(R5), -(SP)
922	005560	013746	002310			MOV	CMDPKT, -(SP)
923	005564	012746	006141			MOV	#STAER3, -(SP)
924	005570	012746	000006			MOV	#6, -(SP)
925	005574	010600				MOV	SP,R0
926	005576	104415				TRAP	C\$PNTX
927	005600	062706	000016			ADD	#16,SP
928	005604			PRINTX #STAER4,CMDPKT+2,CMDPKT+4,CMDPKT+6			





GLOBAL SUBROUTINES SECTION

```
958      .SBTTL GLOBAL SUBROUTINES SECTION
959
960      :++
961      : THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
962      : THAT ARE USED IN MORE THAN ONE TEST.
963      :--
964
965      :      MODULES TO HANDLE TS04 INTERRUPTS.
966
967      TS4IN0: BGNSRV TS4IN0      ;DEVICE 0.
968      LET INTFLG := INTFLG + #1      ;SET INTERRUPT OCCURRED FLAG.
969      ;                               INC      INTFLG
970      005237 003416
971      ENDSRV
972      L10004:
973      006322 000002      RTI
974
975      TS4IN1: BGNSRV TS4IN1      ;DEVICE 1.
976      LET INTFLG+2 := INTFLG+2 + #1 ;SET INTERRUPT OCCURRED FLAG.
977      ;                               INC      INTFLG+2
978      005237 003420
979      ENDSRV
980      L10005:
981      006330 000002      RTI
982
983      TS4IN2: BGNSRV TS4IN2      ;DEVICE 2.
984      LET INTFLG+4 := INTFLG+4 + #1 ;SET INTERRUPT OCCURRED FLAG.
985      ;                               INC      INTFLG+4
986      005237 003422
987      ENDSRV
988      L10006:
989      006336 000002      RTI
990
991      TS4IN3: BGNSRV TS4IN3      ;DEVICE 3.
992      LET INTFLG+6 := INTFLG+6 + #1 ;SET INTERRUPT OCCURRED FLAG.
993      ;                               INC      INTFLG+6
994      005237 003424
995      ENDSRV
996      L10007:
997      006344 000002      RTI
```

```

998      :      SUBROUTINE TO RETRIEVE RECORD COUNT READ FROM TAPE FOR ERROR
999      :      PRINTS.
1000     :      INPUTS:
1001     :      OUTPUTS: R3 = RECORD COUNT READ
1002     :      REGISTERS: R2, R3, R4
1003     :      CALLS:
1004
1005     006346      RECTAP::IF #MOD.CO SETIN CMDWRD THEN      ;READ REV FETCH
1006     006346      032737 000400 003346      BIT      #MOD.CO,CMDWRD
1007     006354      001430      BEQ      50001$
1008     006356      LET R2 := MSGPKT+MS.RFC + DATARD ;FIND LAST READ AD.
1009     006356      013702 002340      MOV      MSGPKT+MS.RFC,R2
1010     006362      063702 003336      ADD      DATARD,R2
1011     006366      IF #BIT00 SETIN R2 THEN      ;ODD AD., REASSEMBLE
1012     006366      032702 000001      BIT      #BIT00,R2
1013     006372      001417      BEQ      50002$
1014     006374      LET R2 := R2 + #1      ;REC COUNT STARTING
1015     006374      005202      INC      R2
1016     006376      LET R3 :B= (R2) CLR.BY #177400 ;WITH UPPER BYTE FETCH
1017     006376      111203      MOVB     (R2),R3
1018     006400      142703 177400      BICB     #177400,R3
1019     006404      LET R3 := SWAP R3      ;
1020     006404      000303      SWAB     R3
1021     006406      LET R2 := R2 - #1      ;LOWER BYTE AD.
1022     006406      005302      DEC      R2
1023     006410      IFB SWBFLG NE #0 THEN
1024     006410      105737 003444      TSTB     SWBFLG
1025     006414      001401      BEQ      50003$
1026     006416      LET R2 := R2 - #1      ;LOWER BYTE AD. ON SWAP
1027     006416      005302      DEC      R2
1028     006420      ENDIF
1029     006420      50003$:
1030     006420      LET R4 :B= (R2) CLR.BY #177400 ;FETCH LOWER BYTE
1031     006420      111204      MOVB     (R2),R4
1032     006422      142704 177400      BICB     #177400,R4
1033     006426      LET R3 := R3 OR R4      ;MERGE BYTES
1034     006426      050403      BIS      R4,R3
1035     006430      ELSE
1036     006430      000401      BR      50004$
1037     006432      50002$:
1038     006432      LET R3 := (R2)      ;EVEN AD. FETCH
1039     006432      011203      MOV      (R2),R3
1040     006434      ENDIF
1041     006434      50004$:
1042     006434      ELSE
1043     006434      000402      BR      50005$
1044     006436      50001$:
1045     006436      LET R3 := @DATARD      ;READ FWD FETCH
1046     006436      017703 174674      MOV      @DATARD,R3
1047     006442      ENDIF
1048     006442      50005$:
1049
1050     006442      000207      RTS      PC
    
```



GLOBAL SUBROUTINES SECTION

```

1051      :      SUBROUTINE TO STORE A SET CHARACTERISTIC COMMAND AS
1052      :      THE FIRST ENTRY IN THE SEQUENCE TABLE.
1053      :      INPUTS:
1054      :      OUTPUTS:
1055      :      REGISTERS:
1056      :      CALLS:
1057
1058 006444      SETCH:: LET R1 := #CMDSEQ      ;INIT COMMAND SEQUENCE TABLE POINTER.
1059 006444 012701 003460      MOV      #CMDSEQ,R1
1060 006450 012721 140004      MOV      #SCH,(R1)+      ;THIS CODE SETS UP A SET CHARACTERISTIC
1061 006454 012721 000040      MOV      #DFTSCH,(R1)+      ;COMMAND AS THE FIRST COMMAND IN THE
1062 006460 012721 000001      MOV      #1,(R1)+      ;SEQUENCE TABLE.
1063 006464 005721      TST      (R1)+      ;SKIP PATTERN LOCATION.
1064 006466 000207      RTS PC
1065
1066
1067
1068
1069      :      SUBROUTINE TO STORE A REWIND COMMAND IN THE SEQUENCE TABLE
1070      :      INPUTS:
1071      :      OUTPUTS:
1072      :      REGISTERS:
1073      :      CALLS:
1074
1075 006470      SETRW:: LET (R1)+ := #RWD      ;CMD = REWIND.
1076 006470 012721 102010      MOV      #RWD,(R1)+
1077 006474      LET (R1)+ := #1      ;BRF.
1078 006474 012721 000001      MOV      #1,(R1)+
1079 006500      LET (R1)+ := #1      ;# OF OPERATIONS.
1080 006500 012721 000001      MOV      #1,(R1)+
1081 006504 005721      TST (R1)+      ;SKIP PATTERN.
1082 006506 000207      RTS PC      ;RETURN

```

```

1083      :      SUBROUTINE TO EXECUTE ALL COMMANDS IN THE SEQUENCE TABLE ON ALL
1084      :      DEVICES.
1085      :      INPUTS:
1086      :      OUTPUTS:      R2 = TERMINATION INDICATOR (0=END OF TABLE,1=EOT)
1087      :      REGISTERS:
1088      :      CALLS:      CMDAC,SETUP,EXSUB,CKHAE,NEXTU,FIRSTU,VFYDAT.
1089
1090 006510      EXALL:: LET R1 := #CMDSEQ      ;INIT SEQUENCE TABLE POINTER.
1091 006510 012701 003460      WHILE (R1) NE #END DO      ;WHILE THERE ARE CMDS IN THE SEQUENCE TABLE.
1092 006514      ;      50006$:
1093 006514      ;      CMP      (R1),#END
1094 006514 021127 177777      ;      BEQ      50007$
1095 006520 001527      ;
1096 006522 004737 007452      JSR PC,SETUP      ;GO SETUP THE COMMAND BLOCK.
1097 006526      WHILE NCNT LT NCNT1 DO      ;WHILE THERE ARE RECORDS REMAINING:
1098 006526      ;      50010$:
1099 006526 023737 003340 003342      ;      CMP      NCNT,NCNT1
1100 006534 002116      ;      BGE      50011$
1101 006536 004737 007344      JSR PC,CMDAC      ;STORE CMD ASCII IN ERROR MESSAGE.
1102 006542      IFB RANDOM NE #0 THEN      ;IF IN RANDOM MODE:
1103 006542 105737 003441      ;      TSTB     RANDOM
1104 006546 001435      ;      BEQ      50012$
1105 006550      ;
1106 006550 023727 003346 104005      IF CMDWRD EQ #WRT THEN      ;IF CMD IS A WRITE THEN:
1107 006556 001031      ;      CMP      CMDWRD,#WRT
1108 006560      ;      BNE      50013$
1109 006560 105737 003442      IFB VFYFLG EQ #0 THEN      ;IF DATA IS NOT TO BE VERIFIED THEN:
1110 006564 001026      ;      TSTB     VFYFLG
1111 006566      ;      BNE      50014$
1112 006566 063737 003362 003360      LET RANB := RANB + RANS      ;GENERATE
1113 006574      ;      ADD      RANS,RANB
1114 006574 063737 003360 003362      LET RANS := RANS + RANB      ;RANDOM
1115 006602      ;      ADD      RANB,RANS
1116 006602 013737 003362 003344      LET BRFCNT := RANS      ;LENGTH
1117 006610      LET BRFCNT := BRFCNT CLR.BY LENMSK      ;MASK RANDOM LENGTH.
1118 006610 043737 003356 003344      ;      MOV      RANS,BRFCNT
1119 006616      IF BRFCNT LT #18. THEN      ;DO NOT ALLOW BYTE COUNT OF LESS THAN 18
1120 006616 023727 003344 000022      ;      BIC      LENMSK,BRFCNT
1121 006624 002003      ;      CMP      BRFCNT,#18.
1122 006626      ;      BGE      50015$
1123 006626 012737 000022 003344      LET BRFCNT := #18.      ;CHANGE COUNT OF 0-17 TO 18.
1124 006634      ;      MOV      #18.,BRFCNT
1125 006634      ;      50015$:
1126 006634      LET CMDPKT+CP.CNT := BRFCNT      ;MOVE BRFCNT TO CMD PACKET.
1127 006634 013737 003344 002316      ;      MOV      BRFCNT,CMDPKT+CP
1128 006642      ;      50014$:
1129 006642      ;      50013$:
1130 006642      ;      50012$:
1131 006642      ;
1132 006642      ;
1133 006642      ;
1134 006642 004737 007004      JSR PC,EXSUB      ;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATUS.
1135 006646 004737 015724      JSR PC,CKHAE      ;CHECK HALT AFTER EACH CMD FLAG.
1136 006652      LET R2 := #1      ;SET ALL UNITS AT BOT/EOT.
1137 006652 012702 000001      ;      MOV      #1,R2
1138 006656 004737 015332      JSR PC,FIRSTU      ;FIND FIRST UNIT.

```



```
1139 006662          WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE UNITS:
1140 006662          50016$:
1141 006662 026527 002532 177777          CMP      DEVTBL(R5),#END
1142 006670 001426          BEQ      50017$
1143 006672          IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
1144 006672 032737 000400 003346          BIT      #MOD.CO,CMDWRD
1145 006700 001406          BEQ      50020$
1146 006702          IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT THEN:
1147 006702 032765 000002 003426          BIT      #X0.BOT,EOTFLG(R
1148 006710 001001          BNE      50021$
1149 006712          LET R2 := #0          ;CLEAR EOT/BOT FLAG.
1150 006712 005002          CLR      R2
1151 006714          ENDIF
1152 006714          50021$:
1153 006714          ELSE          ;ELSE IF CMD IS NOT REVERSE:
1154 006714 000411          BR      50022$
1155 006716          50020$:
1156 006716          IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
1157 006716 032765 000001 003426          BIT      #X0.EOT,EOTFLG(R
1158 006724 001404          BEQ      50023$
1159 006726 032737 000001 003346          BIT      #CMD.CO,CMDWRD
1160 006734 001001          BNE      50024$
1161 006736          50023$:
1162          ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
1163 006736          LET R2 := #0          ;CLEAR EOT/BOT FLAG.
1164 006736 005002          CLR      R2
1165 006740          ENDIF
1166 006740          ENDIF
1167 006740          50024$:
1168 006740          50022$:
1169 006740 004737 015400          JSR PC,NEXTU          ;FIND NEXT UNIT
1170 006744          ENDDO          ;
1171 006744 000746          BR      50016$
1172 006746          50017$:
1173 006746          IF R2 EQ #1 THEN          ;IF ALL UNIT ARE AT EOT/BOT THEN:
1174 006746 020227 000001          CMP      R2,#1
1175 006752 001001          BNE      50025$
1176 006754 000412          BR      EXARTN          ;RETURN WITH R2 = #1.
1177 006756          ENDIF
1178 006756          50025$:
1179 006756          LET NCNT := NCNT + #1          ;UPDATE RECORD COUNT.
1180 006756 005237 003340          INC      NCNT
1181 006762          LET PCMDWD := CMDWRD          ;SAVE PREVIOUS COMMAND WORD.
1182 006762 013737 003346 003352          MOV      CMDWRD,PCMDWD
1183 006770          ENDDO
1184 006770 000656          BR      50010$
1185 006772          50011$:
1186 006772 004737 014316          JSR PC,VFYDAT          ;IF LAST CMD WAS A WRITE VERIFY, THEN GO
1187          ENDDO          ;VERIFY THE LAST N RECORDS OF DATA.
1188 006776          50007$:
1189 006776 000646          BR      50006$
1190 007000          LET R2 := #0          ;SET NORMAL RETURN INDICATOR.
1191 007000          50007$:
1192 007000 005002          CLR      R2
1193 007002 000207          EXARTN: RTS PC          ;RETURN.
1194
```







```

1307 007272          ENDIF
1308 007272
1309 007272          ELSE
1310 007272 000420    ;ELSE IF CMD IS FORWARD:
1311 007274          ;IF NOT AT EOT OR NOT A MCTION CMD THEN:
1312 007274          ;WAIT FOR INT,CHECK STATUS.
1313 007274 032765 000001 003426    IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
1314 007302 001404    BIT #X0.EOT,EOTFLG(R
1315 007304 032737 000001 003346    BEQ 50054$
1316 007312 001003    BIT #CMD.CO,CMDWRD
1317 007314          BNE 50055$
1318
1319 007314 004737 010636    JSR PC,GOWAIT
1320 007320          ELSE
1321 007320 000405
1322 007322
1323 007322          IFB ALLEOT NE #0 THEN
1324 007322 105737 003450          TSTB ALLEOT
1325 007326 001402          BEQ 50057$
1326 007330 004737 010636    JSR PC,GOWAIT
1327 007334          ENDIF
1328 007334
1329 007334          ENDIF
1330 007334
1331 007334          ENDIF
1332 007334
1333 007334 004737 015400    JSR PC,NEXTU ;FIND NEXT UNIT IN TEST CYCLE.
1334 007340          ENDDO
1335 007340 000724          BR 50044$
1336 007342
1337 007342 000207    RTS PC ;RETURN.

```



GLOBAL SUBROUTINES SECTION

```

1338 : THIS SUBROUTINE STORES THE ASCII FOR THE CURRENT COMMAND AND PREVIOUS
1339 : COMMAND IN THE STANDARD ERROR MESSAGE. ON ENTRY LOCATION CMDWRD
1340 : CONTAINS CURRENT CMD AND LOCATION PCMDWD CONTAINS PREVIOUS CMD.
1341 : INPUTS:
1342 : OUTPUTS:
1343 : REGISTERS: R3, R4.
1344 : CALLS: GCMDA
1345
1346 CMDAC:: LET R4 := CMDWRD ;R4 = CMD BINARY.
1347 007344 013704 003346 ;R4 = CMD BINARY. MOV CMDWRD,R4
1348 007344 004737 007416 JSR PC,GCMDA ;GET CMD ASCII.
1349 007354 112337 005706 MOVB (R3)+,STAER1+2 ;MOVE CMD ASCII
1350 007360 112337 005707 MOVB (R3)+,STAER1+3 ;
1351 007364 111337 005710 MOVB (R3),STAER1+4 ;INTO MSG.
1352 007370 LET R4 := PCMDWD ;R4 = PREVIOUS CMD BINARY.
1353 007370 013704 003352 ;R4 = PREVIOUS CMD BINARY. MOV PCMDWD,R4
1354 007374 004737 007416 JSR PC,GCMDA ;GET CMD ASCII.
1355 007400 LET STAER7+24 :B= (R3)+ ;MOVE CMD ASCII
1356 007400 112337 006022 MOVB (R3)+,STAER7+24
1357 007404 LET STAER7+25 :B= (R3)+ ;
1358 007404 112337 006023 MOVB (R3)+,STAER7+25
1359 007410 LET STAER7+26 :B= (R3) ;INTO MSG.
1360 007410 111337 006024 MOVB (R3),STAER7+26
1361 007414 000207 RTS PC ;RETURN. GO EXECUTE NEXT FUNCTION.
1362
1363
1364
1365 : SUBROUTINE TO FIND THE ASCII EQUIVILENT OF THE COMMAND IN R4.
1366 : ADDRESS OF ASCII 1ST WORD IS RETURNED IN R3.
1367 : INPUTS: R4 = PRESENT COMMAND WORD.
1368 : OUTPUTS: R3 = ADDRESS OF PRESENT COMMAND ASCII.
1369 : REGISTERS:
1370 : CALLS:
1371
1372 GCMDA:: LET R3 := #0 ;INIT CMD TBL POINTER.
1373 007416 005003 ;INIT CMD TBL POINTER. CLR R3
1374 007420 WHILE CMDTBL(R3) NE R4 DO ;UNTIL CURRENT CMD IS FOUND:
1375 007420 ;UNTIL CURRENT CMD IS FOUND: 50060$:
1376 007420 026304 003562 CMP CMDTBL(R3),R4
1377 007424 001403 BEQ 50061$
1378 007426 LET R3 := R3 + #2 ;SEARCH CMD TABLE.
1379 007426 062703 000002 ADD #2,R3
1380 007432 ENDDO
1381 007432 000772 BR 50060$
1382 007434 ;SEARCH CMD TABLE. 50061$:
1383 007434 LET R4 := R3
1384 007434 010304 MOV R3,R4
1385 007436 LET R3 := R3 SHIFT -1 ;POINT TO ASCII FOR THAT COMMAND
1386 007436 006203 ASR R3
1387 007440 000240 NOP
1388 007442 060403 ADD R4,R3
1389 007444 062703 003650 ADD #CMDASC,R3
1390 007450 000207 RTS PC ;RETURN.

```

GLOBAL SUBROUTINES SECTION

```

1391 : THIS SUBROUTINE LOADS THE TS04 COMMAND PACKET FROM ONE
1392 : ENTRY IN THE SEQUENCE TABLE.
1393 : INPUTS:
1394 : OUTPUTS:
1395 : REGISTERS: R2, R3.
1396 : CALLS: GENPAT.
1397
1398 007452 SETUP:: LET CMDLG := #0 ;CLR CMD LOGGING CODE(DISABLES LOGGING)
1399 007452 005037 003354 ; CLR CMDLG
1400 007452 012137 002310 MOV (R1)+,CMDPKT ;LOAD THE COMMAND WORD.
1401 007462 011137 002316 MOV (R1),CMDPKT+CP.CNT ;LOAD THE BYTE/RECORD/FILE COUNT.
1402 007466 011137 003344 MOV (R1),BRFCNT ;SAVE BRFCNT FOR THIS COMMAND.
1403 007472 013702 002310 MOV CMDPKT,R2 ;GET CMD.
1404 007476 042702 177740 BIC #NCMD.C,R2 ;CLR ALL BUT CMD BITS.
1405 007502 010203 MOV R2,R3 ;SAVE IT TWICE.
1406 007504 162703 000010 SUB #CMD.C3,R3 ;POSITION COMMAND?
1407 007510 001003 BNE 2$ ;BR IF NOT.
1408 007512 011137 002312 MOV (R1),CMDPKT+2 ;MOVE BPCR IN 2ND PKT WORD FOR POSITION CMD.
1409 007516 000461 BR 3$
1410 007520 2$: IF CMDPKT EQ #WTM THEN ;IF CMD IS A WRITE TAPE MARK THEN:
1411 007520 023727 002310 100011 CMP CMDPKT,#WTM
1412 007526 001003 BNE 50062$
1413 007530 LET CMDLG := #2 ;WTM LOGGING CODE IS 2.
1414 007530 012737 000002 003354 MOV #2,CMDLG
1415 007536 ENDIF
1416 007536 50062$:
1417 007536 010203 MOV R2,R3
1418 007540 162703 000001 SUB #CMD.CO,R3 ;IS IT A READ?
1419 007544 001017 BNE 1$ ;BR IF NOT.
1420 007546 013737 003336 002312 MOV DATARD,CMDPKT+CP.ADL ;IF SO, LOAD THE BUFFER ADDR.
1421 007554 IF #MOD.CO SET IN CMDPKT THEN ;IF CMD IS A READ REV THEN:
1422 007554 032737 000400 002310 BIT #MOD.CO,CMDPKT
1423 007562 001404 BEQ 50063$
1424 007564 LET CMDLG := #4 ;LOGGING CODE IS 4.
1425 007564 012737 000004 003354 MOV #4,CMDLG
1426 007572 ELSE ;ELSE - IF CMD IS A READ FWD:
1427 007572 000403 BR 50064$
1428 007574 50063$:
1429 007574 LET CMDLG := #6 ;LOGGING CODE IS 6.
1430 007574 012737 000006 003354 MOV #6,CMDLG
1431 007602 ENDIF
1432 007602 50064$:
1433 007602 000427 BR 3$ ;CONTINUE.
1434 007604 010203 1$: MOV R2,R3 ;IS IT
1435 007606 162703 000004 SUB #CMD.C2,R3 ;A SET CHARACTERISTICS CMD?
1436 007612 001011 BNE 4$ ;BR IF NOT.
1437 007614 LET CMDPKT+CP.ADL := #SCHBK ;SET UP ADR LO FOR SET CHAR.
1438 007614 012737 002442 002312 MOV #SCHBK,CMDPKT+CP
1439 007622 012737 000010 002316 MOV #SCHCNT,CMDPKT+CP.CNT ;SET BUFFER EXTENT
1440 007630 LET SCHBK+6 := (R1) ;STORE CHARACTERISTIC CODE IN SCH BLOCK.
1441 007630 011137 002450 MOV (R1),SCHBK+6
1442 007634 000412 BR 3$ ;CONTINUE.
1443 007636 010203 4$: MOV R2,R3 ;IS IT
1444 007640 162703 000006 SUB #CMD.C1!CMD.C2,R3 ;A DIAGNOSTIC (DIA) CMD?
1445 007644 001006 BNE 3$ ;BR IF NOT.
1446 007646 012737 000020 002316 MOV #DIACNT,CMDPKT+CP.CNT ;LOAD BUFFER EXTENT.

```



```
1447 007654 012737 003334 002312      MOV    #DIABLK,CMDPKT+CP.ADL  ;LOAD BUFFER ADR LOW.
1448 007662 005721                    3$:  TST    (R1)+                ;POINT TO N (NUMBER OF TIMES TO EXECUTE THIS INS
1449 007664                                LET NCNT1 := (R1)+          ;SAVE NUMBER OF OPERATIONS
1450 007664 012137 003342                    MOV    (R1)+,NCNT1
1451 007670                                ;CLEAR OPERATION COUNTER.
1452 007670 005037 003340                    CLR    NCNT
1453 007674 012137 003374                    MOV    (R1)+,PATERN        ;SAVE PATTERN CODE FOR CURRENT CMD.
1454 007700 010203                    MOV    R2,R3              ;IS IT
1455 007702 162703 000005                    SUB    #CMD.CO!CMD.C2,R3  ;A WRITE?
1456 007706 001010                    BNE    5$                 ;BR IF NOT.
1457 007710 013737 003334 002312          MOV    DATAW,CMDPKT+CP.ADL ;LOAD WRITE BUFFER LO ORDER.
1458 007716 004737 010030                    JSR    PC,GENPAT          ;GO GENERATE THE WRITE PATTERN.
1459 007722                                LET CMDLG := #2          ;WRITE LOGGING CODE IS 2.
1460 007722 012737 000002 003354          5$:  IF #VFY.C SETIN CMDPKT THEN ;IF DATA VERIFICATION IS REQUIRED:
1461 007730                                MOV    #2,CMDLG
1462 007730 032737 000100 002310          BIT    #VFY.C,CMDPKT
1463 007736 001407                                BEQ    50065$
1464 007740                                ;SET VERIFY FLAG.
1465 007740 112737 000001 003442          LET VFYFLG :B= #1
1466 007746 042737 000100 002310          BIC    #VFY.C,CMDPKT
1467 007754                                ELSE
1468 007754 000402                                ;CLEAR VERIFY BIT(NOT USED BY HARDWARE).
1469 007756                                ;IF DATA VERIFICATION IS NOT REQUIRED:
1470 007756                                BR     50066$
1471 007756 105037 003442                    50065$:
1472 007762                                CLR    VFYFLG
1473 007762                                ;CLR VERIFY FLAG.
1474 007762                                CLRB   VFYFLG
1475 007762 013737 003346 003352          50066$:
1476 007770                                ;SAVE PREVIOUS CMD WORD.
1477 007770 013737 002310 003346          LET PCMDWD := CMDWRD
1478 007776                                MOV    CMDWRD,PCMDWD
1479 007776 105737 003444                    ;SAVE PRESENT CMD WORD.
1480 010002 001403                                MOV    CMDPKT,CMDWRD
1481 010004                                ;IF SWAP BYTES IS ENABLED:
1482 010004 052737 010000 002310          IFB SWBFLG NE #0 THEN
1483 010012                                TSTB   SWBFLG
1484 010012                                BEQ    50067$
1485 010012 042737 004000 002310          LET CMDPKT := CMDPKT SET.BY #SWB.C ;SET SWAP BIT IN COMMAND.
1486 010020                                BIS    #SWB.C,CMDPKT
1487 010020 013737 002310 003350          50067$:
1488 010026 000207                                ;CLR BRF BIT (INTERNAL ONLY).
                                LET CMDSAV := CMDPKT
                                ;SAVE 1ST WORD OF COMMAND PACKET.
                                MOV    CMDPKT,CMDSAV
                                ;RETURN.
                                RTS    PC
```

```
1489 : THIS SUBROUTINE SETS UP AND CALLS THE APPROPRIATE SUBROUTINE TO GENERATE
1490 : THE DESIRED PATTERN FOR THE WRITE AND WRITE/VERIFY COMMANDS.
1491 : INPUTS:
1492 : OUTPUTS:
1493 : REGISTERS: R2, R3, R4.
1494 : CALLS: PATRO - PATR7
1495
1496 010030 GENPAT:: LET R3 := PATERN SHIFT 1 ;SETUP PATTERN ROUTINE POINTER
1497 010030 013703 003374 MOV PATERN,R3
1498 010034 006303 ASL R3
1499 010036 LET R4 := BRFCNT + #1 ;SET LENGTH OF WRITE BFR
1500 010036 013704 003344 MOV BRFCNT,R4
1501 010042 005204 INC R4
1502 010044 LET R4 := R4 CLR.BY #1 ;ROUNDED UP TO NEXT WORD
1503 010044 042704 000001 BIC #1,R4
1504 010050 LET R4 := R4 - #2 ;WITH FIRST WORD RESERVED
1505 010050 162704 000002 SUB #2,R4
1506 010054 LET R2 := DATAWT + #2 ;FOR RECORD COUNT
1507 010054 013702 003334 MOV DATAWT,R2
1508 010060 062702 000002 ADD #2,R2
1509 010064 004773 010072 JSR PC,@PATTBL(R3) ;GO GENERATE THE APPROPRIATE PATTERN.
1510 010070 000207 RTS PC ;RETURN TO SETUP SUBROUTINE.
1511
1512 ;TS04 WRITE PATTERN LOOKUP TABLE. USED TO JSR TO THE
1513 ;CORRECT DATA PATTERN GENERATING ROUTINE.
1514
1515 010072 010114 PATTBL: PATRO
1516 010074 010152 PATR1
1517 010076 010172 PATR2
1518 010100 010202 PATR3
1519 010102 010226 PATR4
1520 010104 010240 PATR5
1521 010106 010252 PATR6
1522 010110 010272 PATR7
1523 010112 010324 PATR8
1524
1525
1526 ;INCREMENTING PATTERN. 0 - 377.
1527
1528 010114 PATRO:: LET R3 := #400
1529 010114 012703 000400 MOV #400,R3
1530 010120 1$: LET R4 := R4 - #2 ;DECREMENT WORD COUNT.
1531 010120 162704 000002 SUB #2,R4
1532 010124 100411 BMI 2$ ;BR IF DONE.
1533 010126 LET (R2)+ := R3 ;STORE DATA WORD.
1534 010126 010322 MOV R3,(R2)+
1535 010130 LET R3 := R3 + #1002 ;UPDATE PATTERN.
1536 010130 062703 001002 ADD #1002,R3
1537 010134 IF R3 EQ #1000 THEN ;IF PATTERN HAS WRAPPED AROUND THEN:
1538 010134 020327 001000 CMP R3,#1000
1539 010140 001002 BNE 50070$
1540 010142 LET R3 := #400 ;INIT THE PATTERN AGAIN.
1541 010142 012703 000400 MOV #400,R3
1542 010146 ENDF
1543 010146
1544 010146 000764 BR 1$ ;DO IT AGAIN. 50070$:
```



```
1545 010150 000207          2$:   RTS    PC           ;RETURN.
1546
1547                          ;ALL ONE'S PATTERN.
1548
1549 010152 012703 177777    PATR1:: MOV    #-1,R3       ;ALL ONES PATTERN;.
1550 010156          ZROPAT: LET R4 := R4 - #2    ;DECREMENT BYTE COUNT.
1551 010156 162704 000002          BMI    1$           SUB    #2,R4
1552 010162 100402          MOV    R3,(R2)+       ;DONE?,BR IF YES.
1553 010164 010322          BR    ZROPAT         ;IF NOT LOAD NEXT BYTE WITH PATTERN.
1554 010166 000773          ;DO IT AGAIN.
1555
1556 010170 000207          1$:   RTS    PC           ;RETURN.
```

```

1557                                     ;ALL ZEROES PATTERN.
1558
1559 010172 005003 PATR2:: CLR R3 ;CLR PATTERN REGISTER.
1560 010174 004737 010156 JSR PC,ZROPAT ;GO GENERATE IT.
1561 010200 000207 RTS PC ;RETURN.
1562
1563                                     ;ONE BIT WALKING FROM R TO L IN A FIELD OF ZEROES.
1564
1565 010202 012703 000401 PATR3:: MOV #401,R3 ;INIT PATTERN REGISTER.
1566 010206 WLKZRO: LET R4 := R4 - #2 ;DECREMENT WORD COUNT.
1567 010206 162704 000002 ;SUB #2,R4
1568 010212 100404 BMI 1$ ;BR IF DONE.
1569 010214 010322 MOV R3,(R2)+ ;LOAD DATA.
1570 010216 006303 ASL R3 ;SHIFT PATTERN.
1571 010220 005503 ADC R3 ;ADD CARRY BACK INTO PATTERN.
1572 010222 000771 BR WLKZRO ;DO IT AGAIN.
1573 010224 000207 1$: RTS PC ;RETURN.
1574
1575                                     ;ZERO BIT WALKING FROM R TO L IN A FIELD OF 1'S.
1576
1577 010226 012703 177376 PATR4:: MOV #177376,R3 ;INIT PATTERN REGISTER.
1578 010232 004737 010206 JSR PC,WLKZRO ;GO GENERATE ;IT.
1579 010236 000207 RTS PC ;RETURN.
1580
1581                                     ;ALTERNATING ONE AND ZERO BITS WITH ALTERNATE BYTES
1582                                     ;COMPLEMENTED.
1583
1584 010240 012703 125125 PATR5:: MOV #125125,R3 ;INIT PATTERN REGISTER.
1585 010244 004737 010156 JSR PC,ZROPAT ;GO GENERATE IT.
1586 010250 000207 RTS PC ;RETURN.
1587
1588                                     ;ALTERNATING BYTES OF 000 AND 377.
1589
1590 010252 012703 177400 PATR6:: MOV #177400,R3 ;INIT PATTERN REGISTER.
1591 010256 1$: LET R4 := R4 - #2 ;DECREMENT WORD COUNT.
1592 010256 162704 000002 ;SUB #2,R4
1593 010262 100402 BMI 2$ ;BR IF DONE.
1594 010264 010322 MOV R3,(R2)+ ;LOAD DATA.
1595 010266 000773 BR 1$ ;DO IT AGAIN.
1596 010270 000207 2$: RTS PC ;RETURN.
1597
1598                                     ;RANDOM PATTERN GENERATOR
1599
1600 010272 PATR7:: LET R4 := R4 - #2 ;DECREMENT WORD COUNT
1601 010272 162704 000002 ;SUB #2,R4
1602 010276 100411 BMI GIT ;BR IF DONE.
1603 010300 063737 003362 003360 ADD RANS,RANB ;GET NEW #.
1604 010306 063737 003360 003362 ADD RANB,RANS ;SAVE #.
1605 010314 013722 003362 MOV RANS,(R2)+ ;CONTINUE.
1606 010320 000764 BR PATR7 ;RETURN
1607 010322 000207 GIT: RTS PC
1608
1609 ; NO PATTERN GENERATION.
1610
1611 010324 000207 PATR8:: RTS PC ;RETURN.

```



```

1612      :      THIS SUBROUTINE INITIATES TS04 COMMAND EXECUTION
1613      :      AND CHECKS FOR TS04 RESPONSE.
1614      :      INPUTS:
1615      :      OUTPUTS:
1616      :      REGISTERS:      R2, R3.
1617      :      CALLS:      DROPU, MOVMSG, FIRSTU, NEXTU, WSSR.
1618
1619      010326      EXCUTE:: LET TIME1 := #-1      ;INIT TIMEOUT COUNTER.
1620      010326      012737 177777 003364      REPEAT      ;WAIT -      MOV      #-1,TIME1
1621      010334      LET TIME1 := TIME1 - #1      ;UPDATE TIMEOUT COUNTER.      50071$:
1622      010334      IF TIME1 EQ #0 THEN      ;IF TIMED OUT:      DEC      TIME1
1623      010334      JSR PC,MOVMSG      ;MOVE CURRENT PACKET MSG.      TST      TIME1
1624      010334      005337 003364      ERRDF #2,NSSRM,STAERM      ;REPORT TS04 NOT READY      BNE      50072$
1625      010340
1626      010340      005737 003364      JSR PC,DROPU      ;DROP THE UNIT.
1627      010344      001011      BR EXCRTN      ;RETURN.
1628      010346      004737 011206      JSR PC,DROPU      ;DROP THE UNIT.
1629      010352      104455      BR EXCRTN      ;RETURN.
1630      010352      000002      ENDIF
1631      010354      004271      UNTIL #TS.SSR SETIN @TSSR(R5)      ;WAIT UNTIL DEVICE IS READY.
1632      010356      005372      IF CMDWRD EQ #SCH THEN      ;IF WE ARE DOING A SET CHAR CMD THEN:
1633      010360      004737 015430      LET R5SAVE := R5      ;SAVE CURRENT DEVICE POINTER.
1634      010362      000522      JSR PC,FIRSTU      ;FIND FIRST UNIT.
1635      010366      000522      WHILE DEVTBL(R5) NE #END DO      ;WAIT FOR UNIT READY OR TIME OUT,
1636      010370      UNTIL #TS.SSR SETIN @TSSR(R5)      ;FIND NEXT UNIT.
1637      010370      032775 000200 002462      JSR PC,WSSR      ;WAIT FOR UNIT READY OR TIME OUT,
1638      010370      001756      JSR PC,NEXTU      ;FIND NEXT UNIT.
1639      010376      023727 003346 140004      ENDDO
1640      010400      LET R5 := R5SAVE      ;RESTORE CURRENT DEVICE POINTER.
1641      010400      023727 003346 140004      LET SCHBK := MSGPKA(R5)      ;SET UP ADR OF MSG PKT IN SCH BLOCK.
1642      010406      001022      ENDIF
1643      010410      LET R3 := MSGPKA(R5)      ;ADR OF THIS UNIT'S MSG PACKET.
1644      010410      010537 003400      LET R2 := #0      ;CLR COUNTER.
1645      010414      004737 015332      WHILE R2 NE #MSGCNT DO      ;WHILE THERE ARE MORE LOCATIONS:
1646      010420      JSR PC,WSSR      ;WAIT FOR UNIT READY OR TIME OUT,
1647      010420      026527 002532 177777      JSR PC,NEXTU      ;FIND NEXT UNIT.
1648      010420      001405      ENDDO
1649      010426      004737 011152      BR      50074$
1650      010430      004737 015400      BR      50075$
1651      010434      000767      BR      50074$
1652      010440      LET R5 := R5SAVE      ;RESTORE CURRENT DEVICE POINTER.
1653      010440      013705 003400      LET SCHBK := MSGPKA(R5)      ;SET UP ADR OF MSG PKT IN SCH BLOCK.
1654      010442      016537 002502 002442      ENDIF
1655      010442      LET R3 := MSGPKA(R5)      ;ADR OF THIS UNIT'S MSG PACKET.
1656      010442      016503 002502      LET R2 := #0      ;CLR COUNTER.
1657      010446      005002      WHILE R2 NE #MSGCNT DO      ;WHILE THERE ARE MORE LOCATIONS:
1658      010446      JSR PC,WSSR      ;WAIT FOR UNIT READY OR TIME OUT,
1659      010446      016537 002502 002442      JSR PC,NEXTU      ;FIND NEXT UNIT.
1660      010454      BR      50074$
1661      010454      BR      50075$
1662      010454      BR      50074$
1663      010454      BR      50075$
1664      010460      BR      50074$
1665      010460      BR      50075$
1666      010462      BR      50074$
1667      010462      BR      50075$

```

```

1668 010462 020227 000016                                CMP      R2,#MSGCNT
1669 010466 001405                                BEQ      50077$
1670 010470                                LET (R3)+ := #-1                                ;INIT THE MSG PACKET WITH ALL 1'S
1671 010470 012723 177777                                MOV      #-1,(R3)+
1672 010474                                LET R2 := R2 + #2                                ;UPDATE COUNTER.
1673 010474 062702 000002                                ADD      #2,R2
1674 010500                                ENDDO
1675 010500 000770                                BR       50076$
1676 010502                                50077$:
1677 010502 105737 002210                                TSTB    DINT                                ;ARE INTERRUPTS DISABLED.
1678 010506 001023                                BNE     1$                                ;BR IF YES.
1679 010510                                IFB INTFLG(R5) GT #1 THEN                    ;IF MORE THAN ONE INTERRUPT HAS OCCURED:
1680 010510 126527 003416 000001                                CMPB    INTFLG(R5),#1
1681 010516 003412                                BLE     50100$
1682 010520                                LET TSSREG := @TSSR(R5)                        ;FREEZE THE CURRENT STATUS REG FOR PRINT
1683 010520 017537 002462 003402                                MOV     @TSSR(R5),TSSREG
1684 010526                                ERRDF #15,TOOMM,STAERM                        ;REPORT TOO MANY INTERRUPTS.
1685 010526 104455                                TRAP   C$ERDF
1686 010530 000017                                .WORD 15
1687 010532 004460                                .WORD TOOMM
1688 010534 005372                                .WORD STAERM
1689 010536 004737 015430                                JSR PC,DROPU                                ;DROP THE UNIT
1690 010542 000434                                BR EXCRTN                                ;RETURN - UNIT HAS BEEN DROPPED.
1691 010544                                ENDDIF
1692 010544                                50100$:
1693 010544                                LET INTFLG(R5) := #0                        ;CLR INTERRUPT FLAG FOR THIS DEV.
1694 010544 005065 003416                                CLR     INTFLG(R5)
1695 010550 052737 000200 002310                                BIS     #IE.C,CMDPKT
1696 010556 001005                                IFB ERRREC EQ #0 THEN                        ;SET INT ENABLE BIT.
1697 010556 105737 003415                                1$:                                          ;IF NOT RETRYING
1698 010562 001005                                TSTB   ERRREC
1699 010564                                BNE    50101$
1700 010564 005265 003324                                LET RECCNT(R5) := RECCNT(R5) + #1
1701 010570                                LET @DATAWT := RECCNT(R5)                    ;THEN UPDATE REC COUNT TO WRITE IT ON TAPE
1702 010570 016577 003324 172536                                INC     RECCNT(R5)
1703 010576                                ENDDIF                                MOV     RECCNT(R5),@DATA
1704 010576                                50101$:
1705 010576 012775 002310 002452                                MOV     #CMDPKT,@TSDB(R5)                    ;LOAD TSDB WITH CMDPKT ADDRESS
1706 010604                                IF #TS.SSR SETIN @TSSR(R5) THEN              ;THIS INITIATES COMMAND EXECUTION.
1707 010604 032775 000200 002462                                ;IF READY DID NOT DROP THEN:
1708 010612 001410                                BIT     #TS.SSR,@TSSR(R5)
1709 010614 004737 011206                                BEQ     50102$
1710 010620                                JSR PC,MOVMSG                                ;MOVE CURRENT MESSAGE PACKET TO COMMON.
1711 010620 104455                                ERRDF #3,TOERM,STAERM                        ;REPORT NO TS04 RESPONSE.
1712 010622 000003                                TRAP   C$ERDF
1713 010624 004207                                .WORD 3
1714 010626 005372                                .WORD TOERM
1715 010630 004737 015430                                .WORD STAERM
1716 010630 004737 015430                                JSR PC,DROPU                                ;DROP THE UNIT
1717 010634                                ENDDIF
1718 010634                                50102$:
1719 010634 000207                                EXCRTN: RTS    PC                            ;RETURN.

```



GLOBAL AREAS MACY11 30(1046) 11-OCT-79 14:02 PAGE 46  
 CZTSHC.P11 11-OCT-79 13:59 GLOBAL SUBROUTINES SECTION

```

1720 : THIS SUBROUTINE WAITS FOR THE TS04 INERRUPT OR DONE BIT TO SET AND ALLOWS THE
1721 : OPERATOR TO TRANSFER CONTROL TO THE SUPERVISOR.
1722 : UPON APPEARANCE OF THE INTERRUPT OR DONE, CHECK TSSR FOR STATUS ERRORS,
1723 : LOG BYTES AND ERRORS AND PERFORM ERROR RECOVERY IF NESSASARY.
1724 : INPUTS:
1725 : OUTPUTS:
1726 : REGISTERS: R2, R3.
1727 : CALLS: DROPU, MOVMSG, RECUD, CHKERR, LOG, CLRERR.
1728
1729 010636 GOWAIT:: LET TIME1 := #-1 ;INIT TIME OUT COUNTER.
1730 010636 012737 177777 003364 REPEAT ;REPEAT UNTIL INTERRUPT OCCURES:
1731 010644 REPEAT ;50103$:
1732 010644 BREAK ;GO TO THE SUPER TO ALLOW TTY INPUT.
1733 010644 104422 IF CMDWRD EQ #RWD THEN ;IF COMMAND WAS REWIND THEN:
1734 010644 023727 003346 102010 DELAY 10. ;WAIT EXTRA MSECS EACH LOOP.
1735 010646 001014
1736 010646 012727 000012 MOV #10.,(PC)+
1737 010654 000000 .WORD 0
1738 010656 013727 002116 MOV L$DLY,(PC)+
1739 010662 000000 .WORD 0
1740 010664 005367 177772 DEC -6(PC)
1741 010670 001375 BNE -4
1742 010672 005367 177756 DEC -22(PC)
1743 010676 001367 BNE -20
1744 010700 001367
1745 010704
1746 010706
1747 010706
1748 010706
1749 010706 IF CMDWRD EQ #SFF OR CMDWRD EQ #SFR THEN 50104$:
1750 010706 023727 003346 105010 CMP CMDWRD,#SFF
1751 010714 001404 BEQ 50105$
1752 010716 023727 003346 105410 CMP CMDWRD,#SFR
1753 010724 001014 BNE 50106$
1754 010726
1755 010726 DELAY 12. ;ADD DELAY FOR SPACE TAPE MARK COMMANDS
1756 010726 012727 000014 MOV #12.,(PC)+
1757 010732 000000 .WORD 0
1758 010734 013727 002116 MOV L$DLY,(PC)+
1759 010740 000000 .WORD 0
1760 010742 005367 177772 DEC -6(PC)
1761 010746 001375 BNE -4
1762 010750 005367 177756 DEC -22(PC)
1763 010754 001367 BNE -20
1764 010756
1765 010756
1766 010756 IFB DINT EQ #0 THEN 50106$:
1767 010756 105737 002210 ;IF INTERRUPTS ARE ENABLED.
1768 010762 001003 TSTB DINT
1769 010764 016502 003416 LET R2 := INTFLG(R5) ;FETCH INTERRUPT OCCURRED FLAG.
1770 010764 000406 ELSE ;IF IN BRUTUS MODE:
1771 010770 BR 50110$
1772 010770
1773 010772
1774 010772 LET R3 := COMP #TS.SSR ;SET UP A MASK FOR THE DONE BIT.
1775 010772 012703 000200 MOV #TS.SSR,R3

```

```

1776 010776 005103
1777 011000
1778 011000 017502 002462
1779 011004 040302
1780 011006
1781 011006
1782 011006
1783 011006 005337 003364
1784 011012
1785 011012 005702
1786 011014 001003
1787 011016 005737 003364
1788 011022 001310
1789 011024
1790 011024
1791 011024 005737 003364
1792 011030 001022
1793 011032
1794 011032 016577 003324 172274
1795 011040 005377 172270
1796 011044 004737 011206
1797 011050
1798 011050 104455
1799 011052 000004
1800 011054 004421
1801 011056 005372
1802 011060 004737 015430
1803 011064
1804 011064 012703 003416
1805 011070 004737 011136
1806 011074
1807 011074 000417
1808 011076
1809 011076 004737 011206
1810 011102 004737 011272
1811 011106 004737 011440
1812 011112
1813 011112 105737 003407
1814 011116 001006
1815 011120 004737 014016
1816 011124
1817 011124 012703 003416
1818 011130 004737 011136
1819 011134
1820 011134
1821 011134
1822 011134
1823 011134 000207

      LET R2 := @TSSR(R5) CLR.BY R3 ;FETCH DONE BIT.
      MOV @TSSR(R5),R2
      BIC R3,R2
      ENDIF
      50110$:
      LET TIME1 := TIME1 - #1 ;UPDATE TIMEOUT COUNTER.
      DEC TIME1
      UNTIL R2 NE #0 OR TIME1 EQ #0 ;REPEAT UNTIL INTERRUPT OR READY OCCURES.
      TST R2
      BNE 50111$
      TST TIME1
      BNE 50103$
      50111$:
      IF TIME1 EQ #0 THEN ;IF TIME OUT HAS OCCURRED:
      TST TIME1
      BNE 50112$
      LET @DATAWT := RECCNT(R5) - #1 ;RE-ADJUST REC COUNT DOWN
      MOV RECCNT(R5),@DATA
      DEC @DATAWT
      JSR PC,MOVMSG ;MOVE CURRENT MSG PACKET TO COMMON AREA.
      ERRDF #4,NOINTM,STAERM ;REPORT NO INTERRUPT.
      TRAP C$ERDF
      .WORD 4
      .WORD NOINTM
      .WORD STAERM
      JSR PC,DROPU ;DROP THE UNIT.
      LET R3 := #ENDERF
      MOV #ENDERF,R3
      JSR PC,CLRERR ;CLEAR ALL ERROR FLAGS
      ELSE
      BR 50113$
      50112$:
      JSR PC,MOVMSG ;MOVE CURRENT MSG. PACKET TO COMMON AREA.
      JSR PC,RECU
      ;UPDATE THE RECORD COUNT.
      JSR PC,CHKERR
      ;CHECK FOR STATUS ERRORS.
      IFB WRTYFG EQ #0 THEN
      TSTB WRTYFG
      BNE 50114$
      JSR PC,LOG ;LOG BYTES AND ERRORS.
      LET R3 := #ENDERF
      MOV #ENDERF,R3
      JSR PC,CLRERR ;CLEAR ALL ERROR FLAGS
      ENDIF
      50114$:
      ENDIF
      50113$:
      RTS PC ;RETURN IF DONE.

```



```

1824      :      SUBROUTINE TO CLEAR FLAGS.
1825      :      INPUTS:      R3 = LWA TO BE CLEARED + 2.
1826      :      OUTPUTS:
1827      :      REGISTERS:    R2
1828      :      CALLS:
1829
1830      0111      CLRERR:: LET R2 := #BGNFLG
1831      011136      MOV      #BGNFLG,R2
1832      01114      REPEAT
1833      011142      50115$:
1834      011142      LET (R2)+ := #0
1835      011142      00500      CLR      (R2)+
1836      011144      UNTIL R2 EQ R3
1837      011144      020203      CMP      R2,R3
1838      011146      001375      BNE      50115$
1839      011150      000207      RTS PC
1840
1841
1842
1843      SUBROUTINE TO WAIT UNTIL CURRENT UNIT IS READY OR UNTIL TIME OUT.
1844      INPUTS:
1845      OUTPUTS:
1846      REGISTERS:
1847      CALLS:
1848
1849      011152      012737 177777 003364  WTR:: LET TIME1 := #-1      ;INIT TIMEOUT COUNTER.
1850      011152      REPEAT      ;REPEAT UNTIL DEV READY OR TIMEOUT:
1851      011160      50116$:
1852      011160      BREAK      ;BREAK TO THE SUPERVISOR.
1853      011160      104422      TRAP      C$BRK
1854      011160      LET TIME1 := TIME1 - #1      ;UPDATE TIMEOUT COUNTER.
1855      011162      005337 003364      DEC      TIME1
1856      011162      UNTIL #TS.SSR SETIN @TSSR(R5) OR TIME1 EQ #0
1857      011166      032775 000200 002462      BIT      #TS.SSR,@TSSR(R5)
1858      011174      001003      BNE      50117$
1859      011176      005737 003364      TST      TIME1
1860      011202      001366      BNE      50116$
1861      011204      50117$:
1862      011204      ;REPEAT UNTIL DEV READY OR TIMEOUT.
1863      011204      000207      ;RETURN.
1864      011204      RTS PC
1865
1866

```

```

1867
1868
1869
1870
1871
1872
1873
1874
1875 011206
1876 011206 017537 002462 003402
1877 011214
1878 011214 013702 003402
1879 011220 042702 177761
1880 011224
1881 011224 010237 003376
1882 011230 006237 003376
1883 011234
1884 011234 016503 002502
1885 011240
1886 011240 005002
1887 011242
1888 011242
1889 011242 020227 000016
1890 011246 001405
1891 011250
1892 011250 012362 002334
1893 011254
1894 011254 062702 000002
1895 011260
1896 011260 000770
1897 011262
1898 011262
1899 011262 013765 002342 003426
1900 011270 000207

: SUBROUTINE TO MOVE THE CURRENT MESSAGE PACKET TO THE COMMON AREA AND
: TO UPDATE THE CURRENT TERMINATION CLASS CODE.
: INPUTS:
: OUTPUTS:
: REGISTERS: R2, R3.
: CALLS:

MOVMSG:: LET TSSREG := @TSSR(R5) ;FREEZE THE STATUS REG CONTENTS
LET R2 := TSSREG CLR.BY #TSC.TCC ;EXTRACT THE TERMINATION CLASS CODE,
MOV @TSSR(R5),TSSREG
MOV TSSREG,R2
BIC #TSC.TCC,R2

LET CTCC := R2 SHIFT -1 ;AND SAVE IT
MOV R2,CTCC
ASR CTCC

LET R3 := MSGPKA(R5) ;ADR OF THIS DEVICE'S MSG.
MOV MSGPKA(R5),R3

LET R2 := #0 ;CLR COUNTER.
CLR R2

WHILE R2 NE #MSGCNT DO ;WHILE THERE ARE MORE LOCATIONS:
50120$:
CMP R2,#MSGCNT
BEQ 50121$

LET MSGPKT(R2) := (R3)+ ;MOVE MSG TO COMMON AREA.
MOV (R3)+,MSGPKT(R2)

LET R2 := R2 + #2 ;UPDATE COUNTER.
ADD #2,R2

ENDDO
BR 50120$

50121$:
LET EOTFLG(R5) := MSGPKT+MS.XS0 ;MOVE XSTATO TO EOT FLAG.
MOV MSGPKT+MS.XS0,E0

RTS PC

```



```

1901      :      SUBROUTINE TO ADJUST THE RECORD COUNT.
1902      :      INPUTS:
1903      :      OUTPUTS:
1904      :      REGISTERS:
1905      :      CALLS:
1906
1907 011272      RECUD:: IFB RECLOG EQ #0 THEN      ;IF RECORD HAS NOT BEEN LOGGED:
1908 011272 105737 003411      TSTB      RECLOG
1909 011276 001057      BNE      50122$
1910 011300      LET RECcnt(R5) := RECcnt(R5) - #1
1911 011300 005365 003324      DEC      RECcnt(R5)
1912 011304      IF #BIT0 NOTSETIN CTCC AND #X2.OPM SETIN MSGPKT+MS.XS2 THEN ;IF TAPE MOVED T
1913 011304 032737 000001 003376      BIT      #BIT0,CTCC
1914 011312 001046      BNE      50123$
1915 011314 032737 100000 002346      BIT      #X2.OPM,MSGPKT+M
1916 011322 001442      BEQ      50123$
1917 011324      LET RECLOG :B= RECLOG + #1 ;SET RECORD LOGGED,
1918 011324 105237 003411      INCB      RECLOG
1919 011330      IF CMDWRD EQ #RWD THEN      ;IF THIS IS A REWIND CMD:
1920 011330 023727 003346 102010      CMP      CMDWRD,#RWD
1921 011336 001003      BNE      50124$
1922 011340      LET RECcnt(R5) := #0      ;CLEAR RECORD COUNT,
1923 011340 005065 003324      CLR      RECcnt(R5)
1924 011344      ELSE
1925 011344 000431      BR      50125$
1926 011346
1927 011346      IF #BRF.C SETIN CMDWRD THEN      ;IF BRF USED, UPDATE RECORD COUNT.
1928 011346 032737 004000 003346      BIT      #BRF.C,CMDWRD
1929 011354 001425      BEQ      50126$
1930 011356      IF #MOD.CO NOTSETIN CMDWRD THEN ;IF A FORWARD CMD:
1931 011356 032737 000400 003346      BIT      #MOD.CO,CMDWRD
1932 011364 001007      BNE      50127$
1933 011366      IF #MOD.CO NOTSETIN PCMDWD THEN ;IF PREV CMD WAS A FWD ALSO:
1934 011366 032737 000400 003352      BIT      #MOD.CO,PCMDWD
1935 011374 001002      BNE      50130$
1936 011376      LET RECcnt(R5) := RECcnt(R5) + #1 ;INCREMENT RECORD COUNT.
1937 011376 005265 003324      INC      RECcnt(R5)
1938 011402      ENDIF
1939 011402
1940 011402      ELSE      ;IF REVERSE CMD:
1941 011402 000412      BR      50131$
1942 011404
1943 011404      IF #MOD.CO SETIN PCMDWD THEN ;IF PREVIOUS CMD WAS A REV ALSO:
1944 011404 032737 000400 003352      BIT      #MOD.CO,PCMDWD
1945 011412 001406      BEQ      50132$
1946 011414      IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;WHEN NOT AT BOT THEN
1947 011414 032765 000002 003426      BIT      #X0.BOT,EOTFLG(R
1948 011422 001002      BNE      50133$
1949 011424      LET RECcnt(R5) := RECcnt(R5) - #1 ;DECREMENT RECORD COUNT.
1950 011424 005365 003324      DEC      RECcnt(R5)
1951 011430      ENDIF
1952 011430
1953 011430      ENDIF
1954 011430
1955 011430      ENDIF
1956 011430

```

GLOBAL AREAS  
CZTSHC.P11

MACY11 30(1046)  
11-OCT-79 13:59

11-OCT-79 14:02 PAGE 51  
GLOBAL SUBROUTINES SECTION

L 7

```
1957 011430          ENDIF
1958 011430
1959 011430          ENDIF
1960 011430
1961 011430          ENDIF
1962 011430
1963 011430          LET @DATAWT := RECCNT(R5)
1964 011430 016577 003324 171676
1965 011436          ENDIF
1966 011436
1967 011436 000207   RTS      PC          ;RETURN.

50126$:
50125$:
50123$:
MOV      RECCNT(R5),@DATA
50122$:
```



GLOBAL SUBROUTINES SECTION

```

1968 : THIS IS THE ERROR CHECK SUBROUTINE. AFTER INTERRUPT THIS
1969 : SUBROUTINE IS CALLED TO CHECK THE TS04 STATUS.
1970 : IF SPECIAL COND IS SET THEN THE TCC HANDLING SUBROUTINE IS ENTERED.
1971 : IF THE RFC IS NON ZERO FOR A COMMAND REQUIRING A BPCR,
1972 : THEN AN ERROR RFC IS REPOPTED,
1973 : INPUTS:
1974 : OUTPUTS:
1975 : REGISTERS: R2, R4.
1976 : CALLS: TCC0-TCC7.
1977 :
1978 011440 : CHKERR:: IF #TS.SC SETIN TSSREG THEN ;IF SPECIAL COND STATUS IS SET THEN:
1979 011440 032737 100000 003402 BIT #TS.SC,TSSREG
1980 011446 001441 BEQ 50134$
1981 011450 IF CTCC NE #2 THEN ;IF TCC IS NOT 2 THEN:
1982 011450 023727 003376 000002 CMP CTCC,#2
1983 011456 001405 BEQ 50135$
1984 011460 IFB ERRREC EQ #0 THEN ;IF NOT IN ERROR RECOVERY:
1985 011460 105737 003415 TSTB ERRREC
1986 011464 001002 BNE 50136$
1987 011466 005265 003264 INC SCCNT(R5) ;INC SC COUNTER.
1988 011472 ENDIF 50136$:
1989 011472 ENDIF 50135$:
1990 011472 IF #TS.NXM SETIN TSSREG OR #TS.UPE SETIN TSSREG THEN ;WHEN NON-EXISTANT MEMO
1991 011472 032737 004000 003402 BIT #TS.NXM,TSSREG
1992 011472 001004 BNE 50137$
1993 011500 032737 040000 003402 BIT #TS.UPE,TSSREG
1994 011510 001412 BEQ 50140$
1995 011512 50137$:
1996 011512 IF #X2.OPM NOTSETIN MSGPKT+MS.XS2 THEN ;AND TAPE NOT MOVED
1997 011512 032737 100000 002346 BIT #X2.OPM,MSGPKT+M
1998 011520 001003 BNE 50141$
1999 011522 LET R2 := #5 ;SET TCC5 INDEX
2000 011522 012702 000005 MOV #5,R2
2001 011526 ELSE BR 50142$
2002 011526 000402 50141$:
2003 011530 LET R2 := #4 ;TAPE MOVED, SET TCC4 INDEX
2004 011530 012702 000004 MOV #4,R2
2005 011534 ENDIF 50142$:
2006 011534 ELSE BR 50143$
2007 011534 000402 50140$:
2008 011536 LET R2 := CTCC ;SET DETECTED TCC INDEX
2009 011536 013702 003376 MOV CTCC,R2
2010 011542 ENDIF 50143$:
2011 011542 LET R2 := R2 SHIFT 1 ;CURRENT TCC X 2.
2012 011542 006302 ASL R2
2013 011542 004772 011644 JSR PC,@TCCRA(R2) ;GO TO THE TCC HANDLING SUBROUTINE.
2014 011550 ELSE BR 50144$
2015 011550 000426 50134$:
2016 011552 IF #BRF.C SETIN CMDWRD THEN ;IF BRF IS USED IN THIS CMD THEN:
2017 011552

```

```

2024 011552 032737 004000 003346
2025 011560 001422
2026 011562
2027 011562 005737 002340 IF MSGPKT+MS.RFC NE #0 THEN ;IF THERE IS AN RFC THEN:
2028 011566 001417 BIT #BRF.C,CMDWRD
2029 011570 IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN BEQ 50145$
2030 011570 105737 003441 TST MSGPKT+MS.RFC
2031 011574 001403 BEQ 50146$
2032 011576 105737 003442 TSTB RANDOM
2033 011602 001411 BEQ 50147$
2034 011604 BEQ 50150$
2035
2036 011604 ;IF NOT IN RANDOM OR IF CMD IS WTV:
2037 011604 105737 003445 IFB IRE EQ #0 THEN ;IF RFC ERROR REPORTS ARE ALLOWED:
2038 011610 001006 TSTB IRE
2039 011612 LET HRDCNT(R5) := HRDCNT(R5) + #1 ;UPDATE HARD ERROR COUNT
2040 011612 005265 003304 ERRHRD #13,RFCERM,STAERM ;REPORT RFC ERROR INC HRDCNT(R5):
2041 011616 TRAP C$ERHRD
2042 011616 104456 .WORD 13
2043 011620 000015 .WORD RFCERM
2044 011622 004254 .WORD STAERM
2045 011624 005372
2046 011626
2047 011626 ENDIF
2048 011626
2049 011626 ENDIF
2050 011626
2051 011626 ENDIF
2052 011626
2053 011626 ENDIF
2054 011626
2055 011626 ENDIF
2056 011626 IFB RWERR NE #0 THEN ;IF A READ/WRITE ERROR HAS OCCURRED THEN:
2057 011626 105737 003413 TSTB RWERR
2058 011632 001403 BEQ 50152$
2059 011634 LET CMDPKT := CMDSAV ;RESTORE CMD PACKET AFTER ERROR RECOV.
2060 011634 013737 003350 002310 MOV CMDSAV,CMDPKT
2061 011642
2062 011642
2063 011642 000207 RTS PC ;RETURN.
2064
2065 ;
2066 ; ADDRESSES OF TCC HANDLING ROUTINES FOR TERMINATION CLASS CODES 0 - 7.
2067 011644 011664 TCCRA: TCC0
2068 011646 011702 TCC1
2069 011650 011720 TCC2
2070 011652 012030 TCC3
2071 011654 012046 TCC4
2072 011656 012462 TCC5
2073 011660 012560 TCC6
2074 011662 012722 TCC7
  
```



GLOBAL SUBROUTINES SECTION

```

2075      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 0, UNDEFINED SPECIAL
2076      :      CONDITION ERROR.
2077      :      INPUTS:
2078      :      OUTPUTS:
2079      :      REGISTERS:
2080      :      CALLS:
2081
2082 011664      TCC0:: LET HRDCNT(R5) := HRDCNT(R5) + #1 ;UPDATE HARD ERROR COUNT.
2083 011664 005265 003304      INC      HRDCNT(R5)
2084 011670      ERRHRD #5,SCERM,STAERM      ;REPORT SPECIAL CONDITION ERROR.
2085 011670 104456      TRAP      C$ERHRD
2086 011672 000005      .WORD      5
2087 011674 004230      .WORD      SCERM
2088 011676 005372      .WORD      STAERM
2089 011700      RTS PC      ;RETURN.
2090
2091
2092
2093
2094
2095      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 1, ATTENTION CONDITION.
2096      :      THIS TCC INDICATES THAT THE DRIVE HAS UNDERGONE A STATUS CHANGE
2097      :      SUCH AS GOING OFFLINE OR COMING ONLINE.
2098      :      INPUTS:
2099      :      OUTPUTS:
2100      :      REGISTERS:      R2,R4
2101      :      CALLS:      DROPU
2102
2103 011702      TCC1:: ERRDF #6,ATTNM,STAERM      ;REPORT ATTENTION-UNIT OFF LINE.
2104 011702 104455      TRAP      C$ERDF
2105 011704 000006      .WORD      6
2106 011706 004335      .WORD      ATTNM
2107 011710 005372      .WORD      STAERM
2108 011712 004737 015430      JSR PC,DROPU      ;DROP THE UNIT.
2109 011716 000207      RTS PC      ;RETURN.

```

```

2110      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 2, TAPE STATUS ALERT.
2111      :      A STATUS CONDITION HAS BEEN ENCOUNTERED THAT MAY HAVE SIGNIFICANCE
2112      :      TO THE PROGRAM. BITS OF INTEREST INCLUDE TMK, RLS, LET, RLL, BOT, EOT.
2113      :      INPUTS:
2114      :      OUTPUTS:
2115      :      REGISTERS:
2116      :      CALLS:
2117
2118 011720      TCC2::      IF #X0.BOT SETIN MSGPKT+MS.XSO ANDB EXPBOT NE #0 THEN
2119 011720 032737 000002 002342      BIT      #X0.BOT,MSGPKT+M
2120 011726 001404      BEQ      50153$
2121 011730 105737 003440      TSTB     EXPBOT
2122 011734 001401      BEQ      50153$
2123      :IF AT BOT AND BOT IS EXPECTED:
2124 011736 000433      BR TC2RTN      ;RETURN-TCC2 CAUSED BY EXPECTED BOT.
2125 011740      ENDIF
2126 011740
2127 011740      50153$:
2128 011740 032737 170002 002342      IF #X0.RLS!X0.RLL!X0.TMK!X0.LET!X0.BOT SETIN MSGPKT+MS.XSO THEN
2129 011746 001427      BIT      #X0.RLS!X0.RLL!X
2130      :IF TCC2 CAUSED BY ANYTHING BUT EOT:
2131 011750      IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN
2132 011750 105737 003441      TSTB     RANDOM
2133 011754 001403      BEQ      50155$
2134 011756 105737 003442      TSTB     VFYFLG
2135 011762 001421      BEQ      50156$
2136 011764
2137      50155$:
2138 011764      IFB IRE EQ #0 THEN      ;IF NOT IN RANDOM OR IF CMD IS WTV:
2139 011764 105737 003445      ;IF RFC ERROR REPORTS ARE ALLOWED:
2140 011770 001016      TSTB     IRE
2141 011772      BNE      50157$
2142 011772 105737 003415      IFB ERRREC NE #0 THEN      ;IF WE ARE IN ERROR RECOVERY THEN:
2143 011776 001403      TSTB     ERRREC
2144 012000      LET UNREC :B= UNREC + #1      ;SET UNRECOVERABLE FLAG FOR LOG.
2145 012000 105237 003414      ELSE      ;ELSE - IF NOT IN ERROR RECOVERY:
2146 012004 000402      BEQ      50160$
2147 012004 000402      BR      50161$
2148 012006
2149 012006      50160$:
2150 012006 005265 003264      LET SCCNT(R5) := SCCNT(R5) + #1 ;INCREMENT THE SPEC COND COUNTER.
2151 012012      ENDIF      INC      SCCNT(R5)
2152 012012
2153 012012      50161$:
2154 012012 005265 003304      LET HRDCNT(R5) := HRDCNT(R5) + #1 ;UPDATE HARD ERROR COUNT.
2155 012016      ERRHRD #7,TSAM,STAERM      ;REPORT TAPE STATUS ALERT.
2156 012016 104456      TRAP     C$ERHRD
2157 012020 000007      .WORD   7
2158 012022 004436      .WORD   TSAM
2159 012024 005372      .WORD   STAERM
2160      ENDIF
2161      ENDIF
2162      ENDIF
2163      ENDIF
2164      ENDIF
2165      ENDIF

```



2166 012026 000207

TC2RTN: RTS PC

;RETURN.

2167  
2168  
2169  
2170  
2171

2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179

: SUBROUTINE TO HANDLE TERMINATION CLASS CODE 3, FUNCTION REJECT.  
: THE SPECIFIED FUNCTION WAS NOT INITIATED. BITS OF INTEREST ARE  
: RMR, OFL, VCK, BOT, ILC, WLE, ILA, AND NBA.  
: INPUTS:  
: OUTPUTS:  
: REGISTERS: R2,R4  
: CALLS: DROPU

2180 012030

TCC3:: ERRDF #8,FUNRM,STAERM

;REPORT FUNCTION REJECT.

2181 012030 104455

TRAP C\$ERDF

2182 012032 000010

.WORD 8

2183 012034 004353

.WORD FUNRM

2184 012036 005372

.WORD STAERM

2185 012040 004737 015430

JSR PC,DROPU

;DROP THE UNIT.

2186 012044 000207

RTS PC

;RETURN.

```

2187 : SUBROUTINE TO HANDLE TERMINATION CLASS CODE 4, RECOVERABLE ERROR.
2188 : TAPE POSITION IS ONE RECORD BEYOND WHAT ITS POSITION WAS WHEN
2189 : THE FUNCTION WAS INITIATED. RECOVERY PROCEDURE IS TO LOG THE
2190 : ERROR AND ISSUE THE APPROPRIATE RETRY COMMAND.
2191 : 2 WRITE-ERROR-RECOVERY ALGORITHMS CAN BE SELECTED:
2192 : THE FIRST ONE, VIA BADTSW SWITCH, DOES DETECT BAD SPOTS ON TAPE.
2193 : IT CALLS A WRITE RETRY SUBR UNTIL THE RECORD IS RECOVERED
2194 : OR 20 BAD SPOTS HAVE BEEN LOGGED. ON REACHING 20 BAD
2195 : SPOTS LOGGED, A BAD TAPE OVERFLOW MSG IS PRINTED AND THE
2196 : UNIT DROPPED.
2197 : THE SECOND ALGORITHM ISSUES THE TS11 WRITE RETRY COMMAND
2198 : UP TO 16 TIMES BEFORE DROPPING THE UNIT OR PROCEEDING
2199 : WITH THE NEXT RECORD ON RECOVERY.
2200 : INPUTS:
2201 : OUTPUTS:
2202 : REGISTERS: R2,R4.
2203 : CALLS: RTLE, EXECUTE, GOWAIT, DROPU, WRTY
2204 :
2205 012046 TCC4:: IF CMDLG EQ #2 ANDB BADTSW NE #0 THEN
2206 012046 023727 003354 000002 CMP CMDLG,#2
2207 012054 001125 BNE 50162$
2208 012056 105737 002206 TSTB BADTSW
2209 012062 001522 BEQ 50162$
2210 012064 IFB ERRREC EQ #0 ANDB ERCVER NE #0 THEN
2211 012064 105737 003415 TSTB ERRREC
2212 012070 001007 BNE 50163$
2213 012072 105737 002205 TSTB ERCVER
2214 012076 001404 BEQ 50163$
2215 012100 ERRSOFT #9,RERM,STAERM ;
2216 012100 104457 TRAP C$ERSOFT
2217 012102 000011 .WORD 9
2218 012104 004550 .WORD RERM
2219 012106 005372 .WORD STAERM
2220 012110 ENDIF
2221 012110 IFB IREC EQ #0 THEN ; 50163$:
2222 012110 105737 002211 TSTB IREC
2223 012114 001102 BNE 50164$
2224 012116 LET ERRREC :B= ERRREC + #1 ;RETRY FLAG FOR EXECUTE SUBR: DON'T UPDATE REC CN
2225 012116 105237 003415 INCB ERRREC
2226 012122 LET WRTYER :B= WRTYER + #1 ;REWRITE ERROR FLAG FOR WRTY SUBR
2227 012122 105237 003410 INCB WRTYER
2228 012126 IFB WRTYFG EQ #0 THEN ;FIRST RETRY ON THIS RECORD: SUBSEQUENT
2229 012126 105737 003407 TSTB WRTYFG
2230 012132 001072 BNE 50165$
2231 012134 LET WTYWRD := CMDWRD ;RETRIES WITH TCC4 ERRORS BY-PASS THIS SECTION
2232 012134 013737 003346 013336 ;SAVE WRITE COMMAND PACKET
2233 012142 LET WTYCMD := CMDPKT ;
2234 012142 013737 002310 013334 MOV CMDWRD,WTYWRD
2235 012150 LET WTYBRF := CMDPKT+CP.CNT ;
2236 012150 013737 002316 013340 MOV CMDPKT,WTYCMD
2237 012156 LET RWERR :B= RWERR + #1 ;LOG SUBR FLAG: COUNT WRT ERRORS
2238 012162 105237 003413 MOV CMDPKT+CP.CNT,WT
2239 012162 105237 003407 INCB RWERR
2240 012162 105237 003407 INCB WRTYFG
2241 012162 105237 003407
2242 012162 105237 003407

```



```

2243 012166 REPEAT
2244 012166
2245 012166 LET WRTYCT(R5) := WRTYCT(R5) + #1 50166$:
2246 012166 005265 003244 ;COUNT GLOBAL WRITE RETRIES
2247 012172 INC WRTYCT(R5)
2248 012172 005037 003404 LET RETRYC := #0 ;CLEAR # OF RETRIES PER RECORD
2249 012176 CLR RETRYC
2250 012176 105037 003406 LET RPTCNT :B= #0 ;CLEAR # OF REPEATS
2251 012202 004737 013066 CLR RPTCNT
2252 012206 JSR PC,WRTY ;CALL WRITE RETRY
2253 012206 105737 003410 UNTILB WRTYER EQ #0 OR @BTPT HIS #40. ;REPEAT RETRIES ON SAME RECORD
2254 012212 001404 TSTB WRTYER
2255 012214 027727 171216 000050 BEQ 50167$
2256 012222 103761 CMP @BTPT,#40.
2257 012224 BLO 50166$
2258
2259 012224 IF @BTPT HIS #40. THEN 50167$:
2260 012224 027727 171206 000050 ;UNTIL RECOVERED OR 20 BAD SPOTS
2261 012232 103423 ;WHEN 20 BAD SPOTS LOGGED
2262 012234 PRINTB #BTMSG2 ;PRINT BAD TAPE OVERFLOW MSG
2263 012234 012746 013427 CMP @BTPT,#40.
2264 012240 012746 000001 BLO 50170$
2265 012244 010600 MOV #BTMSG2,-(SP)
2266 012246 104414 MOV #1,-(SP)
2267 012250 062706 000004 MOV SP,R0
2268 012254 004737 013546 JSR PC,BORERS ;ERASE BAD RECORD
2269 012260 LET RECCNT(R5) := RECCNT(R5) - #1 ;
2270 012260 005365 003324 DEC RECCNT(R5)
2271 012264 004737 015430 JSR PC,DROPU ;DROP UNIT
2272 012270 LET RECCNT(R5) := #0 ;
2273 012270 005065 003324 CLR RECCNT(R5)
2274 012274 LET @TSDB(R5) := #RWCPK ;REWIND UNIT
2275 012274 012775 002330 002452 MOV #RWCPK,@TSDB(R5)
2276 012302 ENDIF
2277 012302
2278 012302 LET WRTYFG :B= #0 50170$:
2279 012302 105037 003407 ;RETRY COMPLETE FLAG
2280 012306 CLR WRTYFG
2281 012306 105237 003455 LET MISCFG :B= MISCFG + #1 ;DO NOT HALT ON THIS CMD FLG
2282 012312 INCB MISCFG
2283 012312 013737 013336 003352 LET PCMDWD := WTYWRD ;RESTORE ORIGINAL WRT CMD AFTER RECOVERY
2284 012320 MOV WTYWRD,PCMDWD
2285 012320 ENDIF
2286 012320 ELSE 50165$:
2287 012320 000402 BR 50171$
2288 012322 LET UNREC :B= UNREC + #1 ;
2289 012322 INCB UNREC
2290 012322 105237 003414 ENDIF
2291 012326 ELSE 50171$:
2292 012326 BR 50172$
2293 012326
2294 012326 000454
2295 012330 JSR PC,RTLE 50162$:
2296 012330 004737 012740 ;CHECK FOR RETRY LIMIT EXCEEDED.
2297 012334 IF CMDLG GT #2 THEN ;IF READ CMD THEN:
2298 012334 023727 003354 000002 CMP CMDLG,#2

```

GLOBAL AREAS MACY11 30(1046) 11-OCT-79 14:02 PAGE 59  
 CZTSHC.P11 11-OCT-79 13:59 GLOBAL SUBROUTINES SECTION

```

2299 012342 003411
2300 012344
2301 012344 012702 000020
2302 012350 006202
2303 012352
2304 012352 023702 003404
2305 012356 002403
2306 012360
2307 012360 052737 020000 002310
2308 012366
2309 012366
2310 012366
2311 012366
2312 012366
2313 012366 005737 003404
2314 012372 001007
2315 012374 105737 002205
2316 012400 001404
2317 012402
2318 012402 104457
2319 012404 00C011
2320 012406 004550
2321 012410 005372
2322 012412
2323 012412
2324 012412
2325 012412 005237 003404
2326 012416
2327 012416 052737 001000 002310
2328 012424
2329 012424 105737 002211
2330 012430 001011
2331 012432
2332 012432 105237 003415
2333 012436
2334 012436 012602
2335 012440 012602
2336 012442 004737 010326
2337 012446 000137 010636
2338 012452
2339 012452 000402
2340 012454
2341 012454
2342 012454 105237 003414
2343 012460
2344 012460
2345 012460
2346 012460
2347 012460 000207

      LET R2 := #RRECL SHIFT -1 ;R2=READ RETRY COUNT LIMIT / 2
      IF RETRYC GE R2 THEN ;IF RETRY COUNT IS MORE THAN HALF LIMIT:
          LET CMDPKT := CMDPKT SET.BY #OPP.C ;SET OPPOSITE BIT FOR RETRY2.
      ENDIF
      ENDIF
      IF RETRYC EQ #0 ANDB ERCVER NE #0 THEN ;IF THIS IS THE ORIGINAL ERROR THEN:
          ERRSOFT #9,RERM,STAERM ;REPORT RECOVERABLE ERROR
      ENDIF
      LET RETRYC := RETRYC + #1 ;UPDATE RETRY COUNT.
      LET CMDPKT := CMDPKT SET.BY #MOD.C1 ;SET RETRY BIT IN CMD PACKET.
      IFB IREC EQ #0 THEN ;IF ERROR RECOVERY ENABLED:
          LET ERRREC :B= ERRREC + #1 ;SET ERROR RECOVERY FLAG.
          POP R2,R2 ;POP 2 RTN ADRS FROM STACK.
          JSR PC,EXCUTE ;GO EXECUTE THE RETRY COMMAND.
          JMP GOWAIT ;GO WAIT FOR INTERRUPT + CHECK STATUS.
      ELSE ;ELSE IF ERROR RECOVERY IS NOT ENABLED:
          LET UNREC :B= UNREC + #1 ;SET UNRECOVERABLE ERROR FLAG.
      ENDIF
      RTS PC ;RETURN

```



GLOBAL SUBROUTINES SECTION

```

2348      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 5, RECOVERABLE ERROR.
2349      :      TAPE POSITION HAS NOT CHANGED.  RECOVERY PROCEDURE IS TO LOG THE
2350      :      ERROR AND RE-ISSUE THE ORIGINAL COMMAND.
2351      :      INPUTS:
2352      :      OUTPUTS:
2353      :      REGISTERS:      R2,R4.
2354      :      CALLS:          RTLE, EXCUTE, GOWAIT, DROPU.
2355
2356 012462 004737 012740      TCC5:: JSR PC,RTLE          ;CHECK FOR RETRY LIMIT EXCEEDED
2357 012466          IF RETRYC EQ #0 THEN      ;IF THIS IS THE ORIGINAL ERROR THEN:
2358 012466 005737 003404          TST      RETRYC
2359 012472 001004          BNE      50200$
2360 012474          ERRSOFT #10,RERM,STAERM    ;REPORT RECOVERABLE ERROR.
2361 012474 104457          TRAP      C$ERSOFT
2362 012476 000012          .WORD    10
2363 012500 004550          .WORD    RERM
2364 012502 005372          .WORD    STAERM
2365 012504          ENDIF
2366 012504          50200$:
2367 012504          LET RETRYC := RETRYC + #1    ;UPDATE RETRY COUNTER.
2368 012504 005237 003404          IFB IREC EQ #0 THEN      ;IF ERROR RECOVERY IS ENABLED:
2369 012510          TSTB      IREC
2370 012510 105737 002211          BNE      50201$
2371 012514 001016          LET ERRREC :B= ERRREC + #1    ;SET ERROR RECOVERY FLAG.
2372 012516          INCB      ERRREC
2373 012516 105237 003415          LET RECCNT(R5) := RECCNT(R5) + #1    ;UPDATE REC COUNT
2374 012522          INC      RECCNT(R5)
2375 012522 005265 003324          LET @DATAWT := RECCNT(R5)    ;AND INSERT IT INTO WRT
2376 012526          MOV      RECCNT(R5),@DATA
2377 012526 016577 003324 170600          POP R2,R2          ;POP 2 RTN ADRS FROM STACK.
2378 012534          MOV      (SP)+,R2
2379 012534 012602          MOV      (SP)+,R2
2380 012536 012602          JSR PC,EXCUTE          ;GO RE-ISSUE THE COMMAND.
2381 012540 004737 010326          JMP GOWAIT          ;GO WAIT FOR INTERRUPT + CHECK STATUS.
2382 012544 000137 010636          ELSE          ;ELSE IF ERROR RECOVERY IS NOT ENABLED:
2383 012550          BR      50202$
2384 012550 000402          50201$:
2385 012552          LET UNREC :B= UNREC + #1    ;SET UNRECOVERABLE ERROR FLAG.
2386 012552          INCB      UNREC
2387 012552 105237 003414          ENDIF
2388 012556          50202$:
2389 012556          RTS PC          ;RETURN.
2390 012556 000207
2391
2392

```

GLOBAL AREAS  
CZTSHC.P11MACY11 30(1046)  
11-OCT-79 13:59

11-OCT-79 14:02 PAGE 61

GLOBAL SUBROUTINES SECTION

```

2393      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 6, UNRECOVERABLE ERROR.
2394      :      TAPE POSITION HAS BEEN LOST.  THE ONLY VALID RECOVERY PROCEDURE
2395      :      IS TO REWIND AND START OVER AT BOT UNLESS THE TAPE HAS LABELS OR
2396      :      SEQUENCE NUMBERS.  THIS DIAGNOSTIC WILL REWIND AND RETRY THE
2397      :      COMMAND ONLY IF DENSITY CHECK IS SET, OTHERWISE THE UNIT WILL BE
2398      :      DROPPED FROM THE TEST SEQUENCE.
2399      :      INPUTS:
2400      :      OUTPUTS:
2401      :      REGISTERS:      R2, R4
2402      :      CALLS:          RTLE, WSSR, EXCUTE, GOWAIT, DROPU
2403
2404      012560      TCC6::  IF X3.DCK NOTSETIN MSGPKT+MS.XS3 THEN
2405      012560      033737 000010 002350
2406      012566      001016
2407
2408      012570      IF CMDLG NE #0 THEN
2409      012570      005737 003354
2410      012574      001404
2411      012576      LET RWERR :B= RWERR + #1
2412      012576      105237 003413
2413      012602      LET UNREC :B= UNREC + #1
2414      012602      105237 003414
2415      012606      ENDF
2416      012606
2417      012606      ERRDF #11,URERM,STAERM
2418      012606      104455
2419      012610      000013
2420      012612      004572
2421      012614      005372
2422      012616      004737 015430
2423      012622      JSR PC,DROPU
2424      012622      000436      ELSE
2425      012624
2426      012624      004737 012740      JSR PC,RTLE
2427      012630      IF RETRYC EQ #0 THEN
2428      012630      005737 003404
2429      012634      001004
2430      012636      ERRSOFT #11,URERM,STAERM
2431      012636      104457
2432      012640      000013
2433      012642      004572
2434      012644      005372
2435      012646      ENDF
2436      012646
2437      012646      LET RETRYC := RETRYC + #1
2438      012646      005237 003404
2439      012652      IFB IRE EQ #0 THEN
2440      012652      105737 003445
2441      012656      001016
2442      012660      LET ERRREC :B= ERRREC + #1
2443      012660      105237 003415
2444      012664      LET @TSDB(R5) := #RWCPK
2445      012664      012775 002330 002452
2446      012672      004737 011152      JSR PC,WSSR
2447      012676      POP R2,R2
2448      012676      012602

```

```

;IF THERE IS NO DENSITY CHECK THEN:
;IF CMD IS A READ OR WRITE THEN:
TST      CMDLG
BEQ      50204$
;SET RD/WR ERROR FLAG,
INCB     RWERR
;SET UNRECOVERABLE ERROR FLAG.
INCB     UNREC
50204$:
;REPORT UNRECOVERABLE ERROR.
TRAP     C$ERDF
.WORD    11
.WORD    URERM
.WORD    STAERM
;REPORT ERROR + DROP UNIT.
;ELSE-IF THERE IS DENSITY CHECK:
BR       50205$
50205$:
;CHECK FOR RETRY LIMIT EXCEEDED.
;IF THIS IS THE ORIGINAL ERROR THEN:
TST      RETRYC
BNE     50206$
TRAP     C$ERSOFT
.WORD    11
.WORD    URERM
.WORD    STAERM
50206$:
;UPDATE RETRY COUNT.
INC      RETRYC
;IF ERROR RECOVERY IS ENABLED THEN:
TSTB    IRE
BNE     50207$
;SET ERROR RECOVERY FLAG,
INCB     ERRREC
;ISSUE A REWIND COMMAND,
MOV      #RWCPK,@TSDB(R5)
;WAIT FOR SUBSYSTEM READY,
;POP 2 RTN ADR'S FROM STACK,
MOV      (SP)+,R2

```



```

2449 012700 012602
2450 012702 004737 010326      JSR PC,EXCUTE
2451 012706 000137 010636      JMP GOWAIT
2452 012712
2453 012712 000402      ELSE
2454 012714
2455 012714
2456 012714 105237 003414      LET UNREC :B= UNREC + #1
2457 012720      ENDIF
2458 012720
2459 012720      ENDIF
2460 012720
2461 012720 000207      RTS PC

                                MOV      (SP)+,R2
                                ;REISSUE THE COMMAND,
                                ;WAIT FOR INTERRUPT
                                ;ELSE-IF ERR REC DISABLED:
                                BR      50210$
                                50207$:
                                ;SET UNRECOVERABLE ERROR FLAG.
                                INCB   UNREC
                                ;
                                50210$:
                                ;
                                50205$:
                                ;RETURN

```

```

2462      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 7, FATAL SUBSYSTEM
2463      :      ERROR.  THE SUBSYSTEM IS INCAPABLE OF PROPERLY PERFORMING
2464      :      COMMANDS OR AT LEAST ITS INTEGRITY IS SERIOUSLY QUESTIONABLE.
2465      :      REFER TO THE FATAL CLASS CODE FIELD IN THE TSSR REGISTER FOR
2466      :      ADDITIONAL INFORMATION ON THE TYPE OF FATAL ERROR.
2467      :      INPUTS:
2468      :      OUTPUTS:
2469      :      REGISTERS:      R2, R4
2470      :      CALLS:
2471
2472      012722      TCC7::  ERRDF #12,FATSM,STAERM      ;REPORT FATAL SUBSYSTEM ERROR.
2473      012722      104455      TRAP      C$ERDF
2474      012724      000014      .WORD      12
2475      012726      004373      .WORD      FATSM
2476      012730      005372      .WORD      STAERM
2477      012732      004737      015430      JSR PC,DROPU      ;DROP THE UNIT.
2478      012736      000207      RTS PC      ;RETURN.
2479
2480
2481
2482      :      SUBROUTINE TO CHECK FOR RETRY LIMIT EXCEEDED.  PRINTS ERROR MESSAGE
2483      :      IF EXCEEDED AND DROP UNIT UNLESS COMMAND IS A READ.
2484      :      INPUTS:
2485      :      OUTPUTS:
2486      :      REGISTERS:      R2, R4.
2487      :      CALLS:      DROPU
2488
2489      012740      RTLE::  IF CMDLG EQ #0 THEN      ;IF CMD IS NOT A READ OR WRITE THEN:
2490      012740      005737      003354      TST      CMDLG
2491      012744      001010      BNE      50211$
2492      012746      ERRDF #11,URERM,STAERM      ;REPORT UNRECOVERABLE ERROR.
2493      012746      104455      TRAP      C$ERDF
2494      012750      000013      .WORD      11
2495      012752      004572      .WORD      URERM
2496      012754      005372      .WORD      STAERM
2497      012756      004737      015430      JSR PC,DROPU      ;DROP THE UNIT.
2498      012762      POP R2
2499      012762      012602      MOV      (SP)+,R2
2500      012764      000437      BR RTLRTN      ;AND RETURN.
2501      012766      ENDIF
2502      012766      50211$:
2503      012766      LET RWERR :B= RWERR + #1      ;SET READ/WRITE ERROR FLAG.
2504      012766      105237      003413      INCB      RWERR
2505      012772      IF CMDLG EQ #2 THEN      ;IF CMD IS A WRT OR WTM:
2506      012772      023727      003354      000002      CMP      CMDLG,#2
2507      013000      001016      BNE      50212$
2508      013002      IF RETRYC EQ #WRECL THEN      ;IF RETRY COUNT HAS REACHED LIMIT:
2509      013002      023727      003404      000020      CMP      RETRYC,#WRECL
2510      013010      001011      BNE      50213$
2511      013012      LET UNREC :B= UNREC + #1      ;SET UNRECOVERABLE FLAG
2512      013012      105237      003414      INCB      UNREC
2513      013016      ERRDF #14,RLEXM,STAERM      ;REPORT RETRY LIMIT EXCEEDED.
2514      013016      104455      TRAP      C$ERDF
2515      013020      000016      .WORD      14
2516      013022      004310      .WORD      RLEXM
2517      013024      005372      .WORD      STAERM

```



```

2518 013026 004737 015430          JSR PC,DROPU          ;DROP THE UNIT.
2519 013032          POP R2
2520 013032 012602
2521 013034          ENDIF
2522 013034
2523 013034          ELSE
2524 013034 000413
2525 013036
2526 013036          IF RETRYC EQ #RRECL THEN
2527 013036 023727 003404 000020    ;IF RETRY COUNT HAS REACHED LIMIT:
2528 013044 001007
2529 013046          LET UNREC :B= UNREC + #1
2530 013046 105237 003414          ;SET UNRECOVERABLE FLAG
2531 013052          ERRHRD #14,RLEXM,STAERM
2532 013052 104456          ;REPORT RECOVERABLE ERROR.
2533 013054 000016          TRAP C$ERHRD
2534 013056 004310          .WORD 14
2535 013060 005372          .WORD RLEXM
2536 013062          POP R2
2537 013062 012602          .WORD STAERM
2538 013064          ENDIF
2539 013064          50215$:
2540 013064          ENDIF
2541 013064          50214$:
2542 013064 000207          RTLRTN: RTS PC          ;RETURN

```

```

MOV (SP)+,R2
50213$:
BR 50214$
50212$:
CMP RETRYC,#RRECL
BNE 50215$
INCB UNREC
TRAP C$ERHRD
.WORD 14
.WORD RLEXM
.WORD STAERM
MOV (SP)+,R2

```

```

2543 : SUBR TO REWRITE A BAD, BUT RECOVERABLE WRITTEN RECORD.
2544 : REWRITE RECORD ON SAME SPOT: REPEAT 4 TIMES.
2545 : IF ALL 4 REPEATS GOOD, RECORD IS RECOVERED
2546 : AND A RECOVERABLE WRITE ERROR IS LOGGED.
2547 : IF ANY OF 4 REPEATS BAD, ERASE BAD RECORD, LOG SUSPECTED
2548 : BAD SPOT, RETRY AGAIN. RETRY 4 TIMES, UP TO 4 REPEATS EACH.
2549 : IF RECORD NOT GOOD AFTER 4 RETRIES, ERASE IT, EXIT WITH
2550 : ERROR FLAG WRTYER SET, PRINTING RETRY FAILED.
2551 : THIS ALL SCHEME IS REENTERED 20 TIMES MAX, IE 20 BAD
2552 : SPOTS MAX ARE ALLOWED.
2553 :
2554 : INPUTS:
2555 : OUTPUTS:
2556 : REGISTERS: R3,R4
2557 : CALLS: BORERS, REWRT
2558 :
2559 013066 WRTY:: BEGIN RETRY
2560 013066 REPEAT
2561 013066
2562 013066 BEGIN REPEAT 50217$:
2563 013066 REPEAT
2564 013066
2565 013066 004737 013546 JSR PC,BORERS ;BACKSPACE/ERASE ONE RECORD
2566 013072 LET WRTYER :B= #0 ;CLEAR WRITE RETRY ERROR
2567 013072 105037 003410 CLRB WRTYER
2568 013076 004737 013722 JSR PC,REWRT ;REWRITE RECORD ON SAME SPOT
2569 013102 LET RPTCNT :B= RPTCNT + #1 ;COUNT REPEATS
2570 013102 105237 003406 UNTILB RPTCNT EQ #4 ORB WRTYER NE #0 ;LIMIT: INCB RPTCNT
2571 013106 123727 003406 000004 CMPB RPTCNT,#4
2572 013106 001403 BEQ 50222$
2573 013114 105737 003410 TSTB WRTYER
2574 013116 001761 BEQ 50221$
2575 013122
2576 013124
2577 013124 END REPEAT ;
2578 013124
2579 013124 LET RETRYC := RETRYC + #1 ;COUNT RETRIES
2580 013124 005237 003404 INC RETRYC
2581 013130 IFB WRTYER EQ #0 THEN ;
2582 013130 105737 003410 TSTB WRTYER
2583 013134 001001 BNE 50223$
2584 013136 LEAVE RETRY ;EXIT RETRY LOOP IF RECOVERED
2585 013136 000457 BR 50216$
2586 013140 ELSE ;
2587 013140
2588 013140 IFB ERCVER NE #0 THEN ;
2589 013140 105737 002205 TSTB ERCVER
2590 013144 001415 BEQ 50225$
2591 013146 PRINTB #BTMSG1,RETRYC,<B,RPTCNT> ;PRINT SUSPECTED BAD SPOT
2592 013146 005046 CLR -(SP)
2593 013150 153716 003406 BISB RPTCNT,(SP)
2594 013154 013746 003404 MOV RETRYC, -(SP)
2595 013160 012746 013342 MOV #BTMSG1, -(SP)
2596 013164 012746 000003 MOV #3, -(SP)
2597 013170 010600 MOV SP,R0
2598 013172 104414 TRAP C$PNTB

```



2599	013174	062706	000010						ADD	#10,SP
2600	013200				ENDIF				:	
2601	013200									
2602	013200									
2603	013200	023727	003404	000001	IF RETRYC EQ #1 THEN				50225\$:	
2604	013206	001021								
2605	013210				LET BTPT := BTADDR(R5)					
2606	013210	016537	002544	003436						
2607	013216				LET R4 := @BTPT + #2					
2608	013216	017704	170214							
2609	013222	062704	000002							
2610	013226				LET @BTPT := R4					
2611	013226	010477	170204							
2612	013232				IF R4 LOS #40. THEN					
2613	013232	020427	000050							
2614	013236	101005								
2615	013240				LET R3 := BTPT					
2616	013240	013703	003436							
2617	013244				LET R4 := R4 + R3					
2618	013244	060304								
2619	013246				LET (R4) := RECCNT(R5)					
2620	013246	016514	003324							
2621	013252				ENDIF					
2622	013252									
2623	013252				ENDIF					
2624	013252									
2625	013252				LET ERSFLG :B= ERSFLG + #1					
2626	013252	105237	003451							
2627	013256				LET RWERR :B= #0					
2628	013256	105037	003413							
2629	013262				LET RPTCNT :B= #0					
2630	013262	105037	003406							
2631	013266				ENDIF					
2632	013266									
2633	013266				UNTIL RETRYC EQ #4					
2634	013266	023727	003404	000004						
2635	013274	001274								
2636	013276				END RETRY					
2637	013276									
2638	013276				IFB WRTYER NE #0 THEN					
2639	013276	105737	003410							
2640	013302	001413								
2641	013304				IFB ERCVER NE #0 THEN					
2642	013304	105737	002205							
2643	013310	001410								
2644	013312				PRINTB #BTMSG3					
2645	013312	012746	013477							
2646	013316	012746	000001							
2647	013322	010600								
2648	013324	104414								
2649	013326	062706	000004							
2650	013332				ENDIF					
2651	013332									
2652	013332				ENDIF					
2653	013332									
2654	013332	000207			RTS PC					

2655  
2656  
2657  
2658  
2659  
2660  
2661  
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669  
2670  
2671  
2672  
2673  
2674  
2675  
2676  
2677  
2678  
2679  
2680  
2681  
2682  
2683  
2684  
2685  
2686  
2687  
2688  
2689

013334 000000  
013336 000000  
013340 000000  
  
013342 040445 052523 050123  
013350 041505 020124 040502  
013356 020104 050123 052117  
013364 040440 052106 051105  
013372 022440 030504 040445  
013400 051040 052105 054522  
013406 020054 042045 022461  
013414 020101 042522 042520  
013422 052101 047045 000  
013427 045 022516 041101  
013434 042101 052040 050101  
013442 020105 053117 051105  
013450 046106 053517 020072  
013456 044103 047101 042507  
013464 052040 050101 020505  
013472 047045 047045 000  
013477 045 051101 052105  
013504 054522 043040 044501  
013512 042514 020104 047117  
013520 041040 042101 051440  
013526 047520 027124 027056  
013534 051105 051501 042105  
013542 022441 000116

WTYCMD: .WORD 0 ;STORAGE FOR WRITE CMD WHILE RETRYING  
WTYWRD: .WORD 0 ;STORAGE FOR WRITE CMD WORD WHILE RETPYING  
WTYBRF: .WORD 0 ;STORAGE FOR WRITE BPCR WHILE RETRYING

BTMSG1: .ASCIZ /%ASUSPECT BAD SPOT AFTER %D1%A RETRY, %D1%A REPEAT%N/

BTMSG2: .ASCIZ /%N%ABAD TAPE OVERFLOW: CHANGE TAPE!%N%N/

BTMSG3: .ASCIZ /%ARETRY FAILED ON BAD SPOT...ERASED!%N/

.EVEN



```

2690      :      SUBR TO BACSPACE ONE RECORD
2691      :      IF THE ERASE FLAG IS SET, THEN ERASE THAT RECORD
2692      :      INPUTS:          ERSFLG 1 = DO ERASE
2693      :      OUTPUTS:
2694      :      REGISTERS:
2695      :      CALLS:          EXCUTE, GOWAIT, CKHAE
2696
2697 013546 BORERS:: LET PCMDWD := CMDWRD ;SET COMMAND TO SPACE REV      MOV      CMDWRD,PCMDWD
2698 013546 013737 003346 003352      LET CMDWRD := #SRR ;      MOV      #SRR,CMDWRD
2699 013554
2700 013554 012737 104410 003346      LET CMDPKT := CMDWRD CLR.BY #BRF.C ;      MOV      CMDWRD,CMDPKT
2701 013562
2702 013562 013737 003346 002310      BIC      #BRF.C,CMDPKT
2703 013570 042737 004000 002310
2704 013576
2705 013576 013737 002310 003350      LET CMDSAV := CMDPKT ;      MOV      CMDPKT,CMDSAV
2706 013604
2707 013604 012737 000001 002312      LET CMDPKT+CP.ADL := #1 ;      MOV      #1,CMDPKT+CP.ADL
2708 013612
2709 013612 005037 003354      LET CMDLG := #0 ;      CLR      CMDLG
2710 013616 004737 007344
2711 013622 004737 010326      JSR PC,CMDAC ;
2712 013626 004737 010636      JSR PC,EXCUTE ;
2713 013632 004737 015724      JSR PC,GOWAIT ;
2714 013636      JSR PC,CKHAE ;
2715 013636 105737 003451      IFB ERSFLG NE #0 THEN ;WHEN ERASE FLAG IS SET, DO ERASE      TSTB     ERSFLG
2716 013642 001426      BEQ      50232$
2717 013644
2718 013644 013737 003346 003352      LET PCMDWD := CMDWRD ;      MOV      CMDWRD,PCMDWD
2719 013652
2720 013652 012737 100411 003346      LET CMDWRD := #ERS ;      MOV      #ERS,CMDWRD
2721 013660
2722 013660 013737 003346 002310      LET CMDPKT := CMDWRD ;      MOV      CMDWRD,CMDPKT
2723 013666
2724 013666 013737 002310 003350      LET CMDSAV := CMDPKT ;      MOV      CMDPKT,CMDSAV
2725 013674 004737 007344
2726 013700 004737 010326      JSR PC,CMDAC ;
2727 013704 004737 010636      JSR PC,EXCUTE ;
2728 013710 004737 015724      JSR PC,GOWAIT ;
2729 013714      JSR PC,CKHAE ;
2730 013714 105037 003451      LET ERSFLG :B= #0      CLRB     ERSFLG
2731 013720
2732 013720      ENDIF
2733 013720 000207      RTS PC      50232$:
2734      ;
2735      SUBR TO REWRITE A BADLY WRITTEN RECORD
2736 013722 REWRT: LET PCMDWD := CMDWRD ;RESTORE WRITE COMMAND PACKET      MOV      CMDWRD,PCMDWD
2737 013722 013737 003346 003352      LET CMDWRD := WTYWRD ;      MOV      WTYWRD,CMDWRD
2738 013730
2739 013730 013737 013336 003346      LET CMDPKT := WTYCMD ;      MOV      WTYCMD,CMDPKT
2740 013736
2741 013736 013737 013334 002310      LET CMDSAV := CMDPKT ;      MOV      CMDPKT,CMDSAV
2742 013744
2743 013744 013737 002310 003350      LET CMDPKT+CP.ADL := DATAWT ;      MOV      DATAWT,CMDPKT+CP
2744 013752
2745 013752 013737 003334 002312

```

```

2746 013760          LET CMDPKT+CP.CNT := WTYBRF      ;
2747 013760 013737 013340 002316          MOV      WTYBRF,CMDPKT+CP
2748 013766          LET CMDLG := #2                ;
2749 013766 012737 000002 003354          MOV      #2,CMDLG
2750 013774 004737 007344          JSR PC,CMDAC
2751 014000 004737 010326          JSR PC,EXCUTE      ;RE-WRITE RECORD
2752 014004 004737 010636          JSR PC,GOWAIT
2753 014010 004737 015724          JSR PC,CKHAE
2754 014014 000207          RTS PC

```



```

2755      : SUBROUTINE TO LOG BYTES READ/WRITTEN.
2756      : ALSO UPDATES READ/WRITE ERROR COUNTERS.
2757      : INPUTS:
2758      : OUTPUTS:
2759      : REGISTERS:      R2, R3, R4.
2760      : CALLS:
2761
2762 014016 LOG:: IFB ERLOG EQ #0 THEN ;IF DATA AND ERRORS HAVE NOT BEEN LOGGED THEN:
2763 014016 105737 003412 ;IF DATA AND ERRORS HAVE NOT BEEN LOGGED THEN:
2764 014022 001126 TSTB ERLOG
2765 014024 LET ERLOG :B= ERLOG + #1 ;SET LOG DONE FLAG. BNE 50233$
2766 014024 105237 003412 INCB ERLOG
2767 014030 LET R4 := CMDLG ;GET CURRENT CMD LOGGING CODE. MOV CMDLG,R4
2768 014030 013704 003354 IF R4 NE #0 THEN ;IF THERE IS A CODE THEN:
2769 014034 TST R4
2770 014034 005704 BEQ 50234$
2771 014036 001520 LET R4 := R4 - #2 ;ADJUST THE CODE FOR TABLE INDEX.
2772 014040 SUB #2,R4
2773 014040 162704 000002 LET R2 := R5 + BINC(R4) + #CNTBGN ;R2 = ADR OF BYTE COUNT LSW.
2774 014044 MOV R5,R2
2775 014044 010502 ADD BINC(R4),R2
2776 014046 066402 014302 ADD #CNTBGN,R2
2777 014052 062702 002554 LET (R2) := (R2) + BRFCNT ;ADD BRFCNT TO LSW.
2778 014056 IF MSGPKT+MS.RFC LOS BRFCNT THEN ;IF THE RFC IS LOWER OR THE SAME AS BRFCNT THEN
2779 014056 063712 003344 ADD BRFCNT,(R2)
2780 014062 CMP MSGPKT+MS.RFC,BR
2781 014062 023737 002340 003344 BHI 50235$
2782 014070 101002 LET (R2) := (R2) - MSGPKT+MS.RFC ;SUBTRACT RFC FROM EXPECTED BRFCNT.
2783 014072 SUB MSGPKT+MS.RFC,(R2)
2784 014072 163712 002340 ENDIF
2785 014076
2786 014076 LET R3 := R2 + #10 ;R3 = ADR OF 2ND WORD. 50235$:
2787 014076 MOV R2,R3
2788 014076 010203 ADD #10,R3
2789 014100 062703 000010 WHILE (R2) GT #999. DO
2790 014104
2791 014104 50236$:
2792 014104 021227 001747 CMP (R2),#999.
2793 014110 003404 BLE 50237$
2794 014112 LET (R2) := (R2) - #1000. ;UPDATE BYTE COUNT
2795 014112 162712 001750 SUB #1000.,(R2)
2796 014116 LET (R3) := (R3) + #1 ;2ND WORD.
2797 014116 005213 INC (R3)
2798 014120 ENDDO BR 50236$
2799 014120 000771
2800 014122 LET R2 := R3 + #10 ;R2 = ADR OF 3RD WORD. 50237$:
2801 014122 MOV R3,R2
2802 014122 010302 ADD #10,R2
2803 014124 062702 000010 WHILE (R3) GT #999. DO
2804 014130
2805 014130 50240$:
2806 014130 021327 001747 CMP (R3),#999.
2807 014134 003404 BLE 50241$
2808 014136 LET (R3) := (R3) - #1000. ;UPDATE BYTE COUNT
2809 014136 162713 001750 SUB #1000.,(R3)
2810 014142 LET (R2) := (R2) + #1 ;3RD WORD.

```

2811	014142	005212			INC	(R2)
2812	014144		ENDDO			
2813	014144	000771			BR	50240\$
2814	014146					
2815	014146		LET R3 := R2 + #10	;R3 = ADR OF 4TH WORD.		
2816	014146	010203			MOV	R2,R3
2817	014150	062703	000010		ADD	#10,R3
2818	014154		WHILE (R2) GT #999. DO			
2819	014154					
2820	014154	021227	001747			
2821	014160	003404			50242\$: CMP	(R2),#999.
2822	014162		LET (R2) := (R2) - #1000.	;UPDATE BYTE COUNT	BLE	50243\$
2823	014162	162712	001750		SUB	#1000.,(R2)
2824	014166		LET (R3) := (R3) + #1	;4TH WORD.		
2825	014166	005213			INC	(R3)
2826	014170		ENDDO			
2827	014170	000771			BR	50242\$
2828	014172					
2829	014172		IFB RWERR NE #0 THEN	;IF R/W ERROR, UPDATE ERROR COUNT.	50243\$: TSTB	RWERR
2830	014172	105737	003413		BEQ	50244\$
2831	014176	001440				
2832	014200		LET R2 := R5 + EINC(R4) + #WRREC	;R2 = ADR OF COUNTER.	MOV	R5,R2
2833	014200	010502			ADD	EINC(R4),R2
2834	014202	066402	014310		ADD	#WRREC,R2
2835	014206	062702	002714			
2836	014212		IFB UNREC NE #0 THEN	;IS THE ERROR UNRECOVERABLE?	TSTB	UNREC
2837	014212	105737	003414		BEQ	50245\$
2838	014216	001404				
2839	014220		LET R2 := R2 + #10	;YES, POINT TO NEXT COUNTER.	ADD	#10,R2
2840	014220	062702	000010			
2841	014224		LET (R2) := (R2) + #1	;UPDATE THE ERROR COUNTER	INC	(R2)
2842	014224	005212				
2843	014226		ELSE	;ELSE - IF ERROR IS RECOVERABLE:	BR	50246\$
2844	014226	000424				
2845	014230				50245\$: INC	(R2)
2846	014230		LET (R2) := (R2) + #1	;UPDATE THE ERROR COUNTER		
2847	014230	005212				
2848	014232		IFB IREC EQ #0 THEN	;IF ERROR RECOVERY IS ENABLED:	TSTB	IREC
2849	014232	105737	002211		BNE	50247\$
2850	014236	001020				
2851	014240					
2852	014240	105737	003446	IFB DROPED EQ #0 ANDB ERCVER NE #0 THEN	;IF UNIT HAS NOT BEEN DROPPED:	TSTB
2853	014244	001015			BNE	50250\$
2854	014246	105737	002205		TSTB	ERCVER
2855	014252	001412			BEQ	50250\$
2856	014254		PRINTB #NURTY1,RETRYC	;PRINT # OF RETRIES TO RECOVER	MOV	RETRYC,-(SP)
2857	014254	013746	003404		MOV	#NURTY1,-(SP)
2858	014260	012746	005073		MOV	#2,-(SP)
2859	014264	012746	000002		MOV	SP,R0
2860	014270	010600			TRAP	C\$PNTB
2861	014272	104414			ADD	#6,SP
2862	014274	062706	000006			
2863	014300		ENDIF	;PROVIDED PRINT HAS BEEN ENABLED		
2864	014300				50250\$:	
2865	014300		ENDIF			
2866	014300				50247\$:	



2867 014300  
2868 014300  
2869 014300  
2870 014300  
2871 014300  
2872 014300  
2873 014300  
2874 014300  
2875 014300 000207  
2876  
2877 014302 000000  
2878 014304 000040  
2879 014306 000100  
2880  
2881 014310 000000  
2882 014312 000020  
2883 014314 000040  
2884  
2885  
2886  
2887  
2888  
2889  
2890  
2891  
2892  
2893  
2894  
2895  
2896 014316  
2897 014316 105737 003442  
2898 014322 001426  
2899 014324  
2900 014324 013737 003346 003352  
2901 014332  
2902 014332 012737 104401 003346  
2903 014340  
2904 014340 012737 000004 003354  
2905 014346 004737 014402  
2906 014352  
2907 014352 013737 003346 003352  
2908 014360  
2909 014360 012737 104001 003346  
2910 014366  
2911 014366 012737 000006 003354  
2912 014374 004737 014402  
2913 014400  
2914 014400  
2915 014400 000207

```
ENDIF
ENDIF
ENDIF
ENDIF
RTS PC
INDEXES TO BYTE COUNTERS.
;BINC: 0 ;WRITE.
40 ;READ REV.
100 ;READ FWD.
INDEXES TO READ/WRITE ERROR COUNTERS.
;EINC: 0 ;WRITE.
20 ;READ REV.
40 ;READ FWD.

: IF A WRITE/VERIFY COMMAND IS ISSUED, CONTROL IS THEN
: TRANSFERRED TO THIS SUBROUTINE TO READ REVERSE, CHECK DATA,
: READ FORWARD, CHECK DATA, THEN CONTINUE TO NEXT COMMAND.
: INPUTS:
: OUTPUTS:
: REGISTERS:
: CALLS: VFEXC.

VFYDAT:: IFB VFYFLG NE #0 THEN ;IF DATA IS TO BE VERIFIED:
TSTB VFYFLG
BEQ 50251$
;SAVE THE PREVIOUS COMMAND WORD.
MOV CMDWRD,PCMDWD
;COMMAND IS READ REV.
MOV #RDR,CMDWRD
;SET UP CMD LOGGING INDEX.
MOV #4,CMDLG
;GO READ ALL THE RECORDS REV.
;SAVE THE PREVIOUS COMMAND WORD.
MOV CMDWRD,PCMDWD
;COMMAND IS READ FWD.
MOV #RDF,CMDWRD
;SET UP CMD LOGGING INDEX.
MOV #6,CMDLG
;GO READ ALL RECORDS FWD.

ENDIF
RTS PC ;RETURN.
```

50246\$:  
50244\$:  
50234\$:  
50233\$:

50251\$:

GLOBAL SUBROUTINES SECTION

```

2916      :      SUBROUTINE TO EXECUTE THE READ AND VERIFY, FORWARD OR REVERSE.
2917      :      INPUTS:
2918      :      OUTPUTS:
2919      :      REGISTERS:      R2
2920      :      CALLS:          CMDAC, FIRSTU, VFISU, NEXTU, CKHAE.
2921
2922      014402      VFEXC:: LET CMDPKT := CMDWRD CLR.BY #BRF.C ;COMMAND PACKET = READ REV OR FWD.
2923      014402      013737      003346      002310      MOV      CMDWRD,CMDPKT
2924      014410      042737      004000      002310      BIC      #BRF.C,CMDPKT
2925      014416      IFB SWBFLG NE #0 THEN      ;IF BYTES ARE TO BE SWAPPED:
2926      014416      105737      003444      TSTB     SWBFLG
2927      014422      001403      BEQ      50252$
2928      014424      LET CMDPKT := CMDPKT SET.BY #SWB.C ;SET SWAB BIT IN CMD PACKET.
2929      014424      052737      010000      002310      BIS      #SWB.C,CMDPKT
2930      014432      ENDIF
2931      014432
2932      014432      LET CMDSAV := CMDPKT      ;SAVE COMMAND PACKET 1ST WORD.
2933      014432      013737      002310      003350      MOV      CMDPKT,CMDSAV
2934      014440      013737      003336      002312      MOV      DATARD,CMDPKT+CP.ADL ;SAVE BUFFER START ADDRESS.
2935      014446      LET NCNT := #0      ;CLEAR NUMBER OF OPERATIONS.
2936      014446      005037      003340      CLR      NCNT
2937      014452      WHILE NCNT LT NCNT1 DO      ;WHILE THERE ARE RECORDS REMAINING:
2938      014452      50253$:
2939      014452      023737      003340      003342      CMP      NCNT,NCNT1
2940      014460      002062      BGE      50254$
2941      014462      004737      007344      JSR PC,CMDAC      ;STORE CMD ASCII IN ERROR MSG.
2942      014466      004737      015332      JSR PC,FIRSTU     ;SET UP FOR FIRST UNIT.
2943      014472      WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE DEVICES REMAINING:
2944      014472      50255$:
2945      014472      026527      002532      177777      CMP      DEVTBL(R5),#END
2946      014500      001442      BEQ      50256$
2947      014502      IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
2948      014502      032737      000400      003346      BIT      #MOD.CO,CMDWRD
2949      014510      001421      BEQ      50257$
2950      014512      IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT
2951      014512      032765      000002      003426      BOT
2952      014520      001014      BIT      #X0.BOT,EOTFLG(R
2953      014522      032765      000001      003426      BNE      50260$
2954      014522      032765      000001      003426      IF #X0.EOT SETIN EOTFLG(R5) THEN ;BUT IF AT
2955      014530      001406      BIT      #X0.EOT,EOTFLG(R
2956      014532      IFB ALLEOT NE #0 THEN      ;AND ALL OTHERS AT
2957      014532      105737      003450      EOT
2958      014536      001402      TSTB     ALLEOT
2959      014540      004737      014630      BEQ      50262$
2960      014544      JSR PC,VFISU      ;THEN READ VERIFY
2961      014544      ENDIF      ;IF NOT ALL AT EOT, FREEZE UNIT(S)
2962      014544      50262$:
2963      014544      000402      ELSE      ;IF NOT AT BOT AND
2964      014546      004737      014630      JSR PC,VFISU      50261$:
2965      014552      ENDIF      ;NOT AT EOT, READ VFY
2966      014552      50263$:
2967      014552      ENDIF
2968      014552      50263$:
2969      014552      ELSE      ;ELSE IF CMD IS NOT REVERSE:
2970      014552      000412      50260$:
2971      014552      BR      50264$
  
```



2972	014554								50257\$:
2973	014554								IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
2974	014554	032765	000001	003426					BIT #X0.EOT,EOTFLG(R
2975	014562	001404							BEQ 50265\$
2976	014564	032737	000001	003346					BIT #CMD.CO,CMDWRD
2977	014572	001002							BNE 50266\$
2978	014574								50265\$:
2979									;IF NOT AT EOT OR NOT A MOTION CMD THEN:
2980	014574	004737	014630						;ISSUE CMD, CHECK STATUS AND DATA.
2981	014600								50266\$:
2982	014600								50264\$:
2983	014600								;GO FIND THE NEXT UNIT.
2984	014600								
2985	014600	004737	015400						BR 50255\$
2986	014604								50256\$:
2987	014604	000732							;CHECK FOR HALT AFTER EACH CMD.
2988	014606								;UPDATE THE RECORD COUNT.
2989	014606	004737	015724						INC NCNT
2990	014612								MOV CMDWRD,PCMDWD
2991	014612	005237	003340						
2992	014616								BR 50253\$
2993	014616	013737	003346	003352					50254\$:
2994	014624								;RETURN.
2995	014624	000712							
2996	014626								
2997	014626	000207							

```

2998      :      SUBROUTINE TO ISSUE COMMAND, AWAIT INTERRUPT,
2999      :      CHECK STATUS, CHECK DATA.
3000      :      INPUTS:
3001      :      OUTPUTS:
3002      :      REGISTERS:      R2
3003      :      CALLS:      EXECUTE, GOWAIT, CKDATA.
3004
3005 014630      VFISU::      LET R2 := DATARD + #8.      ;INIT READ BUFFER POINTER.
3006 014630 013702 003336      ;MOV DATARD,R2
3007 014634 062702 000010      ;ADD #8.,R2
3008 014640      WHILE R2 NE DATARD DO      ;UNTIL 8 BYTES HAVE BEEN SET,
3009 014640      ;50267$:
3010 014640 020237 003336      ;CMP R2,DATARD
3011 014644 001403      ;BEQ 50270$
3012 014646      LET -(R2) := #-1      ;INIT READ BUFFER.
3013 014646 012742 177777      ;MOV #-1,-(R2)
3014 014652      ENDDO
3015 014652 000772      ;BR 50267$
3016 014654      ;50270$:
3017 014654 004737 010326      JSR PC,EXECUTE      ;GO EXECUTE THE COMMAND.
3018 014660      IFB DROPEQ #0 THEN      ;IF UNIT HAS NOT BEEN DROPPED THEN:
3019 014660 105737 003446      ;TSTB DROPEQ
3020 014664 001002      ;BNE 50271$
3021 014666 004737 010636      JSR PC,GOWAIT      ;GO WAIT FOR DONE BIT.
3022 014672      ENDIF
3023 014672      ;50271$:
3024 014672      IFB DROPEQ #0 THEN      ;IF UNIT HAS NOT BEEN DROPPED THEN:
3025 014672 105737 003446      ;TSTB DROPEQ
3026 014676 001006      ;BNE 50272$
3027 014700      IF #X0.BOT NOTSETIN EOTFLG(R5) THEN      ;WHEN NOT REVERSED INTO BOT, THEN
3028 014700 032765 000002 003426      ;BIT #X0.BOT,EOTFLG(R
3029 014706 001002      ;BNE 50273$
3030 014710 004737 014716      JSR PC,CKDATA      ;GO VERIFY DATA.
3031 014714      ENDIF
3032 014714      ;50273$:
3033 014714      ENDIF
3034 014714      ;50272$:
3035 014714 000207      RTS PC
3036

```



```

3037      :      SUBROUTINE TO COMPARE DATA BETWEEN READ AND WRITE BUFFERS
3038      :      AND PRINT ERROR MESSAGE ON MISCOMPARE.
3039      :      INPUTS:
3040      :      OUTPUTS:
3041      :      REGISTERS:      R2, R3, R4.
3042      :      CALLS:          GCMDB
3043
3044      014716      CKDATA:: LET R3 := BRFCNT - MSGPKT+MS.RFC ; COMPUTE REC LENGTH READ
3045      014716      013703      003344      MOV      BRFCNT,R3
3046      014722      163703      002340      SUB      MSGPKT+MS.RFC,R3
3047      014726      IF R3 EQ #0 THEN      ;WHEN NO DATA RECEIVED
3048      014726      005703      TST      R3
3049      014730      001015      BNE      50274$
3050      014732      ERRHRD 17,WTVERM,DTAERM      ;PRINT ERROR AND EXIT
3051      014732      104456      TRAP     C$ERHRD
3052      014734      000021      .WORD   17
3053      014736      004164      .WORD   WTVERM
3054      014740      005224      .WORD   DTAERM
3055      014742      PRINTB #DTAER4      ;COMPARE ROUTINE
3056      014742      012746      005010      MOV      #DTAER4,-(SP)
3057      014746      012746      000001      MOV      #1,-(SP)
3058      014752      010600      MOV      SP,R0
3059      014754      104414      TRAP     C$PNTB
3060      014756      062706      000004      ADD      #4,SP
3061      014762      ELSE
3062      014762      000560      BR      50275$
3063      014764      50274$:
3064      014764      IF R3 HI BRFCNT THEN      ;WHEN REC READ IS LONGER
3065      014764      020337      003344      CMP      R3,BRFCNT
3066      014770      101417      BLOS     50276$
3067      014772      ERRHRD 17,WTVERM,DTAERM      ;THAN EXPECTED, PRINT
3068      014772      104456      TRAP     C$ERHRD
3069      014774      000021      .WORD   17
3070      014776      004164      .WORD   WTVERM
3071      015000      005224      .WORD   DTAERM
3072      015002      PRINTB #DTAERS,CMDPKT+CP.CNT      ;AN ERROR MESSAGE
3073      015002      013746      002316      MOV      CMDPKT+CP.CNT,-(
3074      015006      012746      005031      MOV      #DTAERS,-(SP)
3075      015012      012746      000002      MOV      #2,-(SP)
3076      015016      010600      MOV      SP,R0
3077      015020      104414      TRAP     C$PNTB
3078      015022      062706      000006      ADD      #6,SP
3079      015026      ELSE      ;AND EXIT ROUTINE
3080      015026      000536      BR      50277$
3081      015030      50276$:
3082      015030      LET CKDCNT := R3 - #1      ;SAVE VERIFICATION LENGTH - 1.
3083      015030      010337      015326      MOV      R3,CKDCNT
3084      015034      005337      015326      DEC      CKDCNT
3085      015040      005037      015330      CLR CKDFF      ;CLEAR # OF BYTES IN ERROR COUNTER.
3086      015044      005002      CLR R2      ;INIT BYTE COUNTER
3087      015046      LET R3 := DATAW      ;GET WRITE BUFFER ADDRESS.
3088      015046      013703      003334      MOV      DATAW,R3
3089      015052      LET R4 := DATARD      ;GET READ BUFFER ADDRESS.
3090      015052      013704      003336      MOV      DATARD,R4
3091      015056      IFB T1SWB NE #0 THEN      ;WHEN RUNNING TEST1-SUB
3092      015056      105737      003447      TSTB    T1SWB

```





```

3149 015246 105723      3$:      TSTB (R3)+      :UPDATE WRITE BUFFER ADDRESS.
3150 015250 105724      TSTB (R4)+      :UPDATE READ BUFFER ADDRESS.
3151 015252 105722      TSTB (R2)+      :UPDATE BYTE COUNTER.
3152 015254                UNTIL R2 GT CKDCNT      ;END OF DATA COMPARE REPEAT LOOP.
3153 015254 020237 015326                CMP      R2,CKDCNT
3154 015260 003702                BLE      50301$
3155 015262                LET CKDCNT := CKDCNT + #1      ;CKDCNT EQUALS RECORD LENGTH.
3156 015262 005237 015326                IF CKDFF NE #0 THEN      ;IF COMPARE ERROR HAS OCCURED THEN:
3157 015266                PRINTB #DTAER3,CKDFF,CKDCNT      ;PRINT # OF BYTES IN ERROR.
3158 015266 005737 015330                TST      CKDFF
3159 015272 001414                BEQ      50306$
3160 015274                MOV      CKDCNT,-(SP)
3161 015274 013746 015326                MOV      CKDFF,-(SP)
3162 015300 013746 015330                MOV      #DTAER3,-(SP)
3163 015304 012746 004746                MOV      #3,-(SP)
3164 015310 012746 000003                MOV      SP,R0
3165 015314 010600                TRAP    C$PNTB
3166 015316 104414                ADD     #10,SP
3167 015320 062706 000010                ENDIF
3168 015324                ENDIF
3169 015324                ENDIF
3170 015324                ENDIF
3171 015324                RTS      PC
3172 015324                CKDCNT: .WORD 0
3173 015324                CKDFF: .WORD 0
3174 015324 000207                ;OTHERWISE, RETURN.
3175
3176 015326 000000                ;# OF BYTES TO BE VERIFIED -1.
3177 015330 000000                ;# OF BYTES IN ERROR COUNTER.

```

GLOBAL SUBROUTINES SECTION

```

3178      :      SUBROUTINE TO FIND THE FIRST DEVICE IN THE TEST SEQUENCE.
3179      :      INPUTS:
3180      :      OUTPUTS:
3181      :      REGISTERS:
3182      :      CALLS:
3183
3184 015332 FIRSTU:: LET DROPED :B= #0          ;CLR UNIT DROPPED FLAG
3185 015332 105037 003446          ;CLR DEVICE POINTER.      CLRB  DROPED
3186 015336          LET R5 := #0          ;CLR DEVICE POINTER.      CLR  R5
3187 015336 005005          WHILE DEVTBL(R5) EQ #NINUSE DO ;WHILE DEVICES ARE NOT IN USE:
3188 015340          ;                               50307$:
3189 015340          ;                               CMP  DEVTBL(R5),#NINU
3190 015340 026527 002532 177774 ;                               BNE  50310$
3191 015346 001003          LET R5 := R5 + #2          ;POINT TO NEXT DEVICE.
3192 015350          ;                               ADD  #2,R5
3193 015350 062705 000002          ENDDO
3194 015354          ;                               BR   50307$
3195 015354 000771          ;                               50310$:
3196 015356          ;                               IF DEVTBL(R5) EQ #END THEN ;IF ALL UNITS HAVE BEEN DROPPED THEN:
3197 015356          ;                               ;                               CMP  DEVTBL(R5),#END
3198 015356 026527 002532 177777 ;                               ;                               BNE  50311$
3199 015364 001001          ;                               ;                               DOCLN          ;DO CLEAN CODE AND TERMINATE PASS.
3200 015366          ;                               ;                               TRAP  C$DCLN
3201 015366 104444          ;                               ;                               50311$:
3202 015370          ;                               ;                               LET L$LUN := DEVTBL(R5) ;SET UNIT # IN 'HEADER' FOR ERROR REPORT
3203 015370          ;                               ;                               MOV  DEVTBL(R5),L$LUN
3204 015370          ;                               ;                               RTS    PC          ;RETURN WITH 1ST DEVICE IN R5.
3205 015370 016537 002532 002074
3206 015376 000207
3207
3208
3209
3210
3211
3212      :      SUBROUTINE TO FIND THE NEXT UNIT IN THE TEST CYCLE.
3213      :      INPUTS:
3214      :      OUTPUTS:
3215      :      REGISTERS:
3216      :      CALLS:
3217
3218 015400 NEXTU:: LET DROPED :B= #0          ;CLR UNIT DROPPED FLAG
3219 015400 105037 003446          ;REPEAT UNTIL THE NEXT DEVICE IS FOUND.      CLRB  DROPED
3220 015404          REPEAT          ;REPEAT UNTIL THE NEXT DEVICE IS FOUND.      50312$:
3221 015404          ;                               LET R5 := R5 + #2          ;UPDATE DEVICE TABLE POINTER.
3222 015404          ;                               ;                               ADD  #2,R5
3223 015404 062705 000002          UNTIL DEVTBL(R5) NE #NINUSE
3224 015410          ;                               ;                               CMP  DEVTBL(R5),#NINU
3225 015410 026527 002532 177774 ;                               ;                               BEQ  50312$
3226 015416 001772          LET L$LUN := DEVTBL(R5)          ;SET UNIT # IN 'HEADER' FOR ERROR REPORT
3227 015420          ;                               ;                               MOV  DEVTBL(R5),L$LUN
3228 015420 016537 002532 002074
3229 015426 000207          ;RETURN.
3230
3231
3232

```



```

3233 : SUBROUTINE TO DROP A DEVICE FROM THE TEST SEQUENCE.
3234 : INPUTS:
3235 : OUTPUTS:
3236 : REGISTERS:
3237 : CALLS:          MOVMSG, PRXST, LOG
3238
3239 015430          DROPU:: LET FTLCNT(R5) := FTLCNT(R5) + #1 ;INCREMENT THE FATAL ERROR COUNT.
3240 015430 005265 003314          INC          FTLCNT(R5)
3241 015434          LET R4 := MSGPKT+MS.XS3 CLR.BY #377 ;GET UDIAG ERROR CODE FROM XSTAT3.
3242 015434 013704 002350          MOV          MSGPKT+MS.XS3,R4
3243 015440 042704 000377          BIC          #377,R4
3244 015444          LET R3 := MSGPKA(R5)          ;ADR OF THIS UNIT'S MSG PACKET.
3245 015444 016503 002502          MOV          MSGPKA(R5),R3
3246 015450          LET R2 := #0          ;CLR COUNTER.
3247 015450 005002          WHILE R2 NE #MSGCNT DO          ;WHILE THERE ARE MORE LOCATIONS:
3248 015452          50313$:
3249 015452          CMP          R2,#MSGCNT
3250 015452 020227 000016          BEQ          50314$
3251 015456 001405          LET (R3)+ := #-1          ;INIT THE MSG PACKET WITH ALL 1'S
3252 015460          LET R2 := R2 + #2          ;UPDATE COUNTER.
3253 015460 012723 177777          MOV          #-1,(R3)+
3254 015464 062702 000002          ADD          #2,R2
3255 015470          ENDDO
3256 015470 000770          BR          50313$
3257 015472          50314$:
3258 015472          LET @TSDB(R5) := #GSCPCK          ;INITIATE A GET STATUS COMMAND.
3259 015472          JSR PC,WSSR          ;WAIT A WHILE FOR SSR=1
3260 015472 012775 002320 002452          JSR PC,MOVMSG          ;MOVE MSG PACKET TO COMMON AREA.
3261 015500 004737 011152          IF R4 EQ #X3.RNY THEN          ;IF WE HAVE A CAPSTAN RUNAWAY THEN:
3262 015504 004737 011206          ERRDF #16,RNYM,STAERM          ;REPORT CAPSTAN RUNAWAY WITH TACH CNT.
3263 015510          TRAP          C$ERDF
3264 015510 020427 157400          .WORD          16
3265 015514 001005          .WORD          RNYM
3266 015516          .WORD          STAERM
3267 015516 104455          ELSE
3268 015520 000020          ;ELSE-IF NOT A RUNAWAY:
3269 015522 004504          BR          50316$
3270 015524 005372          50315$:
3271 015526 000402          JSR PC,PRXST          ;PRINT EXTENDED STATUS REGISTERS.
3272 015526 000402          ENDIF
3273 015530          IFB RECLOG NE #0 THEN          ;IF THE RECORD HAS BEEN LOGGED THEN:
3274 015530 004737 015642          TSTB          RECLOG
3275 015534          BEQ          50317$
3276 015534          LET DROPED :B= DROPED + #1          ;SET UNIT DROPPED FLAG.
3277 015534          JSR PC,LOG          ;LOG DATA BYTES + RD/WR ERRORS.
3278 015534 105737 003411          ENDDO
3279 015540 001404          DORPT          ;PRINT PERFORMANCE REPORT
3280 015542          TRAP          C$DRPT
3281 015542 105237 003446          IF PASCNT(R5) NE #0 THEN          TST          PASCNT(R5)
3282 015546 004737 014016
3283 015552
3284 015552
3285 015552
3286 015552 104424
3287 015554
3288 015554 005765 003254

```

```

3289 015560 001402
3290 015562
3291 015562 005365 003254
3292 015566
3293 015566
3294 015566
3295 015566 016537 002532 015640
3296 015574
3297 015574 010500
3298 015576 006200
3299 015600
3300 015600 104451
3301 015602
3302 015602 026527 002532 177774
3303 015610 001410
3304 015612
3305 015612 105737 002211
3306 015616 001005
3307 015620 000240
3308 015622 000240
3309 015624 000240
3310 015626
3311 015626 105237 003452
3312 015632
3313 015632
3314 015632
3315 015632
3316 015632
3317 015632 105237 003446
3318 015636 000207
3319
3320 015640 000000

      LET PASCNT(R5) := PASCNT(R5) - #1
      ENDIF
      LET DROPN := DEVTBL(R5)
      LET R0 := R5 SHIFT -1
      DODU R0
      IF DEVTBL(R5) NE #NINUSE THEN
      IFB IREC EQ #0 THEN
      NOP
      NOP
      NOP
      LET STAFLG :B= STAFLG + #1
      ENDIF
      ENDIF
      DRORTN: LET DROPED :B= DROPED + #1
      RTS PC
      DROPN: .WORD 0

      BEQ 50320$
      DEC PASCNT(R5)
      50320$:
      MOV DEVTBL(R5),DROPN
      MOV R5,R0
      ASR R0
      ;DROP THE UNIT: EXEC BGNDU-ENDDU CODE IF IDU = 0
      TRAP C$DODU
      CMP DEVTBL(R5),#NINU
      BEQ 50321$
      ;IF RECOVERY IS ENABLED THEN:
      TSTB IREC
      BNE 50322$
      ;SET START FLAG TO ENABLE REWIND,
      INCB STAFLG
      50322$:
      50321$:
      INCB DROPED
      ;RETURN.
      ;# OF UNIT TO BE DROPPED
    
```



```

3321      ;      SUBROUTINE TO PRINT EXTENDED STATUS REGISTERS.
3322      ;      INPUTS:
3323      ;      OUTPUTS:
3324      ;      REGISTERS:
3325      ;      CALLS:
3326
3327      PRXST:: PRINTX #GETSTM
3328      015642 012746 005157      MOV      #GETSTM,-(SP)
3329      015646 012746 000001      MOV      #1,-(SP)
3330      015652 010600      MOV      SP,R0
3331      015654 104415      TRAP    C$PNTX
3332      015656 062706 000004      ADD      #4,SP
3333      015662      PRINTX #STAERS,MSGPKT+MS.XS0,MSGPKT+MS.XS1,MSGPKT+MS.XS2,MSGPKT+MS.XS3
3334      015662 013746 002350      MOV      MSGPKT+MS.XS3,-(
3335      015666 013746 002346      MOV      MSGPKT+MS.XS2,-(
3336      015672 013746 002344      MOV      MSGPKT+MS.XS1,-(
3337      015676 013746 002342      MOV      MSGPKT+MS.XS0,-(
3338      015702 012746 006217      MOV      #STAERS,-(SP)
3339      015706 012746 000005      MOV      #5,-(SP)
3340      015712 010600      MOV      SP,R0
3341      015714 104415      TRAP    C$PNTX
3342      015716 062706 000014      ADD      #14,SP
3343      015722 000207      RTS PC
3344
3345
3346
3347
3348      ;      SUBROUTINE TO HALT AFTER EACH COMMAND.
3349      ;      INPUTS:
3350      ;      OUTPUTS:
3351      ;      REGISTERS:      R3, R4
3352      ;      CALLS:
3353
3354      CKHAE:: IFB HAE NE #0 THEN      ;IF HALT FLAG IS SET:
3355      015724 105737 002204      TSTB    HAE
3356      015730 001430      BEQ     50323$
3357      015732      IFB MISCFG EQ #0 THEN      ;
3358      015732 105737 003455      TSTB    MISCFG
3359      015736 001023      BNE     50324$
3360      015740      MANUAL      ;IS MANUAL INTERVENTION ALLOWED?
3361      015740 104450      TRAP    C$MANI
3362      015742      BNCOMPLETE CKHRTN      ;BR IF NOT.
3363      015742 103023      BCC     CKHRTN
3364      015744      LET R4 := CMDWRD      ;COMMAND WORD.
3365      015744 013704 003346      MOV     CMDWRD,R4
3366      015750 004737 007416      JSR PC,GCMDA      ;FETCH ADR OF CMD ASCII.
3367      015754      LET HALTM :B= (R3)+      ;MOVE CMD ASCII
3368      015754 112337 004042      MOV     (R3)+,HALTM
3369      015760      LET HALTM+1 :B= (R3)+
3370      015760 112337 004043      MOV     (R3)+,HALTM+1
3371      015764      LET HALTM+2 :B= (R3)      ;INTO MESSAGE.
3372      015764 111337 004044      MOV     (R3),HALTM+2
3373      015770      GMANIL HALTM,TIME1,1,YES      ;HALT - WAIT FOR AN OEPRATOR INPUT.
3374      015770 104443      TRAP    C$GMAN
3375      015772 000404      BR     10000$
3376      015774 003364      .WORD  TIME1

```

3377	015776	000130							
3378	016000	004042						.WORD	T\$CODE
3379	016002	000001						.WORD	HALTM
3380	016004		10000\$:					.WORD	1
3381	016004			ELSE					
3382	016004	000402							
3383	016006								
3384	016006			LET MISCFCG :B= #0			50324\$:	BR	50325\$
3385	016006	105037 003455							
3386	016012			ENDIF				CLRB	MISCFCG
3387	016012						50325\$:		
3388	016012			ENDIF					
3389	016012						50323\$:		
3390	016012	000207	CKHRTN:	RTS	PC				
3391				.EVEN					
3392									
3393	016014			ENDMOD					



```

3394
3395          .TITLE MISCELLANEOUS SECTIONS
3396          .SBTTL  REPORT CODING SECTION
3397
3398 016014          BGNMOD
3399
3400          :++
3401          : THE REPORT CODING SECTION CONTAINS THE
3402          : 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
3403          :--
3404
3405 016014          BGNRPT
3406 016014          L$RPT::
3407
3408
3409 016014          LET      R5SAVE := R5          ;SAVE CURRENT DEVICE POINTER.
3410 016014 010537 003400          ;MOV      R5,R5SAVE
3411 016020 004737 015332          JSR      PC,FIRSTU          ;FIND THE FIRST UNIT.
3412 016024          WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
3413 016024          ;50326$:
3414 016024 026527 002532 177777          CMP      DEVTBL(R5),#END
3415 016032 001562          BEQ      50327$
3416 016034          PRINTS          #RPT1A,DEVTBL(R5),PASCNT(R5),RECCNT(R5)
3417 016034 016546 003324          MOV      RECCNT(R5),-(SP)
3418 016040 016546 003254          MOV      PASCNT(R5),-(SP)
3419 016044 016546 002532          MOV      DEVTBL(R5),-(SP)
3420 016050 012746 016656          MOV      #RPT1A,-(SP)
3421 016054 012746 000004          MOV      #4,-(SP)
3422 016060 010600          MOV      SP,R0
3423 016062 104416          TRAP    C$PNTS
3424 016064 062706 000012          ADD      #12,SP
3425 016070          PRINTS          #RPT1B,WRBC+30(R5),WRBC+20(R5),WRBC+10(R5),WRBC(R5)
3426 016070 016546 002554          MOV      WRBC(R5),-(SP)
3427 016074 016546 002564          MOV      WRBC+10(R5),-(SP)
3428 016100 016546 002574          MOV      WRBC+20(R5),-(SP)
3429 016104 016546 002604          MOV      WRBC+30(R5),-(SP)
3430 016110 012746 016733          MOV      #RPT1B,-(SP)
3431 016114 012746 000005          MOV      #5,-(SP)
3432 016120 010600          MOV      SP,R0
3433 016122 104416          TRAP    C$PNTS
3434 016124 062706 000014          ADD      #14,SP
3435 016130          PRINTS          #RPT1C,RRBC+30(R5),RRBC+20(R5),RRBC+10(R5),RRBC(R5)
3436 016130 016546 002614          MOV      RRBC(R5),-(SP)
3437 016134 016546 002624          MOV      RRBC+10(R5),-(SP)
3438 016140 016546 002634          MOV      RRBC+20(R5),-(SP)
3439 016144 016546 002644          MOV      RRBC+30(R5),-(SP)
3440 016150 012746 017004          MOV      #RPT1C,-(SP)
3441 016154 012746 000005          MOV      #5,-(SP)
3442 016160 010600          MOV      SP,R0
3443 016162 104416          TRAP    C$PNTS
3444 016164 062706 000014          ADD      #14,SP
3445 016170          PRINTS          #RPT1D,RFBC+30(R5),RFBC+20(R5),RFBC+10(R5),RFBC(R5)
3446 016170 016546 002654          MOV      RFBC(R5),-(SP)
3447 016174 016546 002664          MOV      RFBC+10(R5),-(SP)
3448 016200 016546 002674          MOV      RFBC+20(R5),-(SP)
3449 016204 016546 002704          MOV      RFBC+30(R5),-(SP)

```

```

3450 016210 012746 017055      MOV      #RPT1D,-(SP)
3451 016214 012746 000005      MOV      #5,-(SP)
3452 016220 010600                MOV      SP,R0
3453 016222 104416                TRAP     C$PNTS
3454 016224 062706 000014      ADD      #14,SP
3455 016230                PRINTS   #RPT1F,WRREC(R5),RRREC(R5),RFREC(R5)
3456 016230 016546 002754      MOV      RFREC(R5),-(SP)
3457 016234 016546 002734      MOV      RRREC(R5),-(SP)
3458 016240 016546 002714      MOV      WRREC(R5),-(SP)
3459 016244 012746 017161      MOV      #RPT1F,-(SP)
3460 016250 012746 000004      MOV      #4,-(SP)
3461 016254 010600                MOV      SP,R0
3462 016256 104416                TRAP     C$PNTS
3463 016260 062706 000012      ADD      #12,SP
3464 016264                PRINTS   #RPT1G,WRUNR(R5),RRUNR(R5),RFUNR(R5)
3465 016264 016546 002764      MOV      RFUNR(R5),-(SP)
3466 016270 016546 002744      MOV      RRUNR(R5),-(SP)
3467 016274 016546 002724      MOV      WRUNR(R5),-(SP)
3468 016300 012746 017232      MOV      #RPT1G,-(SP)
3469 016304 012746 000004      MOV      #4,-(SP)
3470 016310 010600                MOV      SP,R0
3471 016312 104416                TRAP     C$PNTS
3472 016314 062706 000012      ADD      #12,SP
3473 016320                IFB     BADTSW NE #0 THEN ;
3474 016320 105737 002206                TSTB    BADTSW
3475 016324 001402                BEQ     50330$
3476 016326 004737 016410      JSR     PC,BTRPT ;GO PRINT BAD TAPE SPOTS WHEN ENABLED
3477 016332                ENDIF
3478 016332                50330$:
3479 016332                PRINTS   #RPT1I,SCCNT(R5),HRDCNT(R5),FTLCNT(R5),VFYCNT(R5)
3480 016332 016546 003274      MOV      VFYCNT(R5),-(SP)
3481 016336 016546 003314      MOV      FTLCNT(R5),-(SP)
3482 016342 016546 003304      MOV      HRDCNT(R5),-(SP)
3483 016346 016546 003264      MOV      SCCNT(R5),-(SP)
3484 016352 012746 017427      MOV      #RPT1I,-(SP)
3485 016356 012746 000005      MOV      #5,-(SP)
3486 016362 010600                MOV      SP,R0
3487 016364 104416                TRAP     C$PNTS
3488 016366 062706 000014      ADD      #14,SP
3489 016372 004737 015400      JSR     PC,NEXTU ;FIND THE NEXT UNIT.
3490 016376                ENDDO
3491 016376 000612                BR      50326$
3492 016400                50327$:
3493 016400                LET     R5 := R5SAVE ;RESTORE CURRENT DEVICE POINTER.
3494 016400 013705 003400      MOV      R5SAVE,R5
3495 016404                EXIT    RPT
3496 016404 000167                .WORD   J$JMP
3497 016406 001130                .WORD   L10010-2-.
3498
3499
3500
3501
3502 ; SUBR TO PRINT BAD TAPES SPOTS DURING THE REPORT PRINTS
3503 ; WRITE RETRIES: CUMULATIVE COUNT
3504 ; BAD TAPE SPOTS: COUNT PER TAPE PASS ONLY, NOT CUMULATIVE.
3505 ; COUNT OF RECOVERABLE WRITE ERRORS EXCLUDES BAD TAPE SPOTS.

```



3506							
3507							
3508	016410				BTRPT: PRINTS #RPT1E,WRTYCT(R5)		;PRINT GLOBAL WRITE RETRY COUNT
3509	016410	016546	003244				MOV WRTYCT(R5),-(SP)
3510	016414	012746	017303				MOV #RPT1E, -(SP)
3511	016420	012746	000002				MOV #2, -(SP)
3512	016424	010600					MOV SP,R0
3513	016426	104416					TRAP C\$PNTS
3514	016430	062706	000006				ADD #6,SP
3515	016434				LET BTPT := BTADDR(R5)		;BTPT IS BOTH THE BAD TAPE SPOT COUNTER
3516	016434	016537	002544	003436			MOV BTADDR(R5),BTPT
3517	016442				LET R3 := @BTPT SHIFT -1		;AND THE LOGGING INDEX
3518	016442	017703	164770				MOV @BTPT,R3
3519	016446	006203					ASR R3
3520	016450				PRINTS #RPT1J,R3		;PRINT # OF BAD TAPE SPOTS
3521	016450	010346					MOV R3, -(SP)
3522	016452	012746	017333				MOV #RPT1J, -(SP)
3523	016456	012746	000002				MOV #2, -(SP)
3524	016462	010600					MOV SP,R0
3525	016464	104416					TRAP C\$PNTS
3526	016466	062706	000006				ADD #6,SP
3527	016472				IF R3 NE #0 THEN		;PRINT RECORD # IF BAD SPOTS DETECTED
3528	016472	005703					TST R3
3529	016474	001457					BEQ 50331\$
3530	016476				IF R3 HI #20. THEN		:
3531	016476	020327	000024				CMP R3,#20.
3532	016502	101402					BLOS 50332\$
3533	016504				LET R3 := #20.		;20 BAD SPOTS IS THE LIMIT
3534	016504	012703	000024				MOV #20.,R3
3535	016510				ENDIF		
3536	016510						50332\$:
3537	016510				PRINTS #CRLFSP		:
3538	016510	012746	005216				MOV #CRLFSP, -(SP)
3539	016514	012746	000001				MOV #1, -(SP)
3540	016520	010600					MOV SP,R0
3541	016522	104416					TRAP C\$PNTS
3542	016524	062706	000004				ADD #4,SP
3543	016530				LET R4 := BTPT + #2		;FETCH A BAD SPOT ID
3544	016530	013704	003436				MOV BTPT,R4
3545	016534	062704	000002				ADD #2,R4
3546	016540				LET R2 := #0		;R2 = PRINT COUNT PER LINE: 10 MAX
3547	016540	005002					CLR R2
3548	016542				REPEAT		:
3549	016542						50333\$:
3550	016542				PRINTS #RPT1K,(R4)		;PRINT A BAD SPOT ID
3551	016542	011446					MOV (R4), -(SP)
3552	016544	012746	017420				MOV #RPT1K, -(SP)
3553	016550	012746	000002				MOV #2, -(SP)
3554	016554	010600					MOV SP,R0
3555	016556	104416					TRAP C\$PNTS
3556	016560	062706	000006				ADD #6,SP
3557	016564				LET R2 := R2 + #1		;COUNT PRINTS
3558	016564	005202					INC R2
3559	016566				LET R4 := R4 + #2		;NEXT
3560	016566	062704	000002				ADD #2,R4
3561	016572				IF R2 EQ #10. THEN		:

```

3562 016572 020227 000012
3563 016576 001014
3564 016600 PRINTS #CRLFSP ;GO TO NEXT PRINT LINE PAST 10 PRINTS
3565 016600 012746 005216 MOV #CRLFSP,-(SP)
3566 016604 012746 000001 MOV #1,-(SP)
3567 016610 010600 MOV SP,R0
3568 016612 104416 TRAP C$PNTS
3569 016614 062706 000004 ADD #4,SP
3570 016620 LET R3 := R3 - #10. ;ADJUST BAD SPOT COUNT
3571 016620 162703 000012 SUB #10.,R3
3572 016624 LET R2 := R2 - #10. ;ADJUST PRINT COUNT
3573 016624 162702 000012 SUB #10.,R2
3574 016630 ENDIF ;
3575 016630 UNTIL R2 EQ R3 ;LIMIT: # OF BAD SPOTS
3576 016630 50334$:
3577 016630 020203 CMP R2,R3
3578 016632 001343 BNE 50333$
3579 016634 ENDIF ;
3580 016634 PRINTS #CRLF ; 50331$:
3581 016634
3582 016634 012746 005213 MOV #CRLF,-(SP)
3583 016640 012746 000001 MOV #1,-(SP)
3584 016644 010600 MOV SP,R0
3585 016646 104416 TRAP C$PNTS
3586 016650 062706 000004 ADD #4,SP
3587 016654 000207 RTS PC
3588
3589
3590

```

```

016656 047045 047045 040445 RPT1A: .ASCII BEX
016733 045 041101 052131 RPT1B: .ASCIZ /%N%N%AUNIT %D1%S3%APASS:%D5%S3%ARECORD:%D5%N/
017004 040445 054502 042524 RPT1C: .ASCIZ /%ABYTES WRITTEN %D3%A,%Z3%A,%Z3%A,%Z3%N/
017055 045 041101 052131 RPT1D: .ASCIZ /%ABYTES READ REV %D3%A,%Z3%A,%Z3%A,%Z3%N/
017125 045 031123 022463 RPT1E: .ASCIZ /%ABYTES READ FWD %D3%A,%Z3%A,%Z3%A,%Z3%N/
017161 045 051101 041505 RPT1F: .ASCIZ /%S23%AWRT%S4%ARDR%S4%ARDF%N/
017232 040445 047125 042522 RPT1G: .ASCIZ /%ARECOVERABLE ERRORS %D5%S2%D5%S2%D5%N/
017303 045 053501 044522 RPT1H: .ASCIZ /%AUNRECOVERABLE ERRORS %D5%S2%D5%S2%D5%N/
017333 045 022516 031104 RPT1I: .ASCIZ /%AWRITE RETRIES%S8%D5%N/
017420 042045 022465 030523 RPT1J: .ASCIZ /%N%D2%A BAD SPOTS THIS TAPE PASS PRECEDING RECORD #:/
017427 045 051501 042520 RPT1K: .ASCIZ /%D5%S1/
017503 045 031523 042045 RPT1L: .ASCII '%ASPEC COND%S3%AHARD%S3%AFATAL%S3%ACOMPARE%N''
      .ASCIZ /%S3%D5%S3%D5%S3%D5%S3%D5%N%N/
      .LIST BEX
      .EVEN

```

```

3591
3592
3593 017540 ENDRPT
3594 017540 L10010:
3595 017540 104425 TRAP C$RPT
3596
3597 .SBTTL LOAD DEVICE PROTECTION TABLE
3598
3599 ;++
3600 ;TABLE FOR SUPERVISOR TO IDENTIFY THE P-TBL FOR THE LOAD DEV
3601 ;THE SUPERVISOR USES THE TBL TO WARN THE OPERATOR WHEN HE TRIES TO TEST THE LOAD DEV
3602 ;--
3603
3604 017542 BGNPROT

```



3605 017542  
3606 017542 000000  
3607 017544 177777  
3608 017546 177777  
3609 017550

L\$PROT::

.WORD 0  
.WORD -1  
.WORD -1  
ENDPRCT

:P-TBL OFFSET OF TSSR, THE TS11 CSR  
:P-TBL OFFSET OF MASS BUS UNIT #: -1 = NOT A MASS BUS DE  
:P-TBL OFFSET OF DRIVE #: -1 = NONE, ONE DRIVE PER UNIBU

```

3610 .SBTTL INITIALIZE SECTION
3611
3612
3613 :++
3614 : THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3615 : AT THE BEGINNING OF EACH PASS.
3616 :--
3617 017550 BGNINIT
3618 017550 L$INIT::
3619
3620 017550 INIT10: IF #BIT0!BIT1 SET IN #CMDPKT THEN ;IF CMD PACKET IS NOT ON MODULO 4 BOUNDRY:
3621 017550 032727 000003 002310 BIT #BIT0!BIT1,#CMDP
3622 017556 001421 BEQ 50335$
3623 017560 ERRSF #1,CMDPKM ;PRINT ERROR MSG,
3624 017560 104454 TRAP C$ERSF
3625 017562 000001 .WORD 1
3626 017564 004102 .WORD CMDPKM
3627 017566 000000 .WORD 0
3628 017570 DELAY 2000. ;GO TO SUPERVISOR, WAIT 2 SECONDS.
3629 017570 012727 003720 MOV #2000.,(PC)+
3630 017574 000000 .WORD 0
3631 017576 013727 002116 MOV L$DLY,(PC)+
3632 017602 000000 .WORD 0
3633 017604 005367 177772 DEC -6(PC)
3634 017610 001375 BNE -4
3635 017612 005367 177756 DEC -22(PC)
3636 017616 001367 BNE -20
3637 017620 000753 BR INIT10 ;
3638 017622 ENDIF
3639 017622 50335$:
3640
3641 017622 IFB CLRFLG NE #0 THEN ;IF CLR COUNTERS FLAG SET:
3642 017622 105737 002202 TSTB CLRFLG
3643 017626 001413 BEQ 50336$
3644 017630 105037 002202 CLR B CLRFLG
3645 017634 LET R2 := #0 ;INIT CLR FLAG.
3646 017634 005002 CLR R2
3647 017636 WHILE R2 NE #CNTLEN DO
3648 017636 50337$:
3649 017636 020227 000550 CMP R2,#CNTLEN
3650 017642 001405 BEQ 50340$
3651 017644 LET WRBC(R2) := #0 ;CLR ALL STATISTICAL COUNTERS.
3652 017644 005062 002554 CLR WRBC(R2)
3653 017650 LET R2 := R2 + #2
3654 017650 062702 000002 ADD #2,R2
3655 017654 ENDDO BR 50337$
3656 017654 000770 50340$:
3657 017656 ENDIF 50336$:
3658 017656
3659 017656
3660
3661 017656 IFB RRANV NE #0 THEN ;IF RESET RANDOM VARIABLE FLAG IS SET THEN:
3662 017656 105737 002203 TSTB RRANV
3663 017662 001406 BEQ 50341$
3664 017664 LET RANB := #RANBC ;RESET RANDOM BASE #.
3665 017664 012737 153624 003360 MOV #RANBC,RANB

```



3666	017672				LET RANS := #RANSC		;RESET RANDOM SAVE LOCATION.
3667	017672	012737	032561	003362			MOV #RANSC,RANS
3668	017700				ENDIF		
3669	017700						50341\$:
3670	017700				READEF #EF.START		;READ START COMMAND EVENT FLAG.
3671	017700	012700	000040				MOV #EF.START,R0
3672	017704	104447					TRAP C\$REFG
3673	017706				BNCOMPLETE INIT15		;BRANCH IF NOT STARTING.
3674	017706	103026					BCC INIT15
3675	017710				LET STAFLG :B= STAFLG + #1		;SET START COMMAND FLAG.
3676	017710	105237	003452				INCB STAFLG
3677	017714				LET R5 := #6		
3678	017714	012705	000006				MOV #6,R5
3679	017720				REPEAT		;INITIATE UNIT NUMBER TABLE
3680	017720						50342\$:
3681	017720				LET DEVTBL(R5) := #NINUSE		;BY STORING NOT IN USE IN EACH LOCATION.
3682	017720	012765	177774	002532			MOV #NINUSE,DEVTBL(R
3683	017726				LET R5 := R5 - #2		
3684	017726	162705	000002				SUB #2,R5
3685	017732				UNTIL R5 EQ #0		
3686	017732	005705					TST R5
3687	017734	001371					BNE 50342\$
3688	017736				LET R5 := L\$UNIT SHIFT 1		
3689	017736	013705	002012				MOV L\$UNIT,R5
3690	017742	006305					ASL R5
3691	017744				REPEAT		;STORE ALL UNIT
3692	017744						50343\$:
3693	017744				LET R5 := R5 - #2		;NUMBERS IN DEVTBL.
3694	017744	162705	000002				SUB #2,R5
3695	017750				LET DEVTBL(R5) := R5 SHIFT -1		
3696	017750	010565	002532				MOV R5,DEVTBL(R5)
3697	017754	006265	002532				ASR DEVTBL(R5)
3698	017760				UNTIL R5 EQ #0		
3699	017760	005705					TST R5
3700	017762	001370					BNE 50343\$
3701							
3702	017764				INIT15: READEF #EF.PWR		;HAS THERE BE A POWER FAILURE?
3703	017764	012700	000034				MOV #EF.PWR,R0
3704	017770	104447					TRAP C\$REFG
3705	017772				BNCOMPLETE INIT16		;BRANCH IF NOT.
3706	017772	103004					BCC INIT16
3707	017774				LET STAFLG :B= STAFLG + #1		;IF SO - SET THE START FLAG.
3708	017774	105237	003452				INCB STAFLG
3709	020000				LET PWRFLG :B= PWRFLG + #1		;IF SO - SET THE POWER FAIL FLAG.
3710	020000	105237	003453				INCB PWRFLG
3711							
3712	020004				INIT16: RFLAGS OPFLAG		;READ AND STORE FLAGS SET BY OPERATOR
3713	020004	104421					TRAP C\$RFLA
3714	020006	010037	003456				MOV R0,OPFLAG
3715	020012				LET R3 := #0		;CLEAR EVENT FLAG
3716	020012	005003					CLR R3
3717	020014				IFB PWRFLG EQ #0 THEN		;IF POWER FAIL HAS NOT OCCURRED THEN:
3718	020014	105737	003453				TSTB PWRFLG
3719	020020	001020					BNE 50344\$
3720	020022				READEF #EF.NEW		;UPDATE PASS COUNT WHEN
3721	020022	012700	000035				MOV #EF.NEW,R0

3722	020026	104447				TRAP	C\$REFG
3723	020030			IFCOND CS THEN	;SUPERVISOR IS IN NEW PASS	BCC	50345\$
3724	020030	103014				TSTB	STAF LG
3725	020032			IFB STAF LG EQ #0 THEN	;AND DIAG WAS NEITHER STARTED	BNE	50346\$
3726	020032	105737	003452			MOV	#EF .RES ,R0
3727	020036	001010				TRAP	C\$REFG
3728	020040			READEF #EF .RES	;NOR		
3729	020040	012700	000037			MOV	#EF .RES ,R0
3730	020044	104447				TRAP	C\$REFG
3731	020046			IFCOND CC THEN	;RESTARTED		
3732	020046	103402				BCS	50347\$
3733	020050			LET R3 := COMP R3	;DO IT		
3734	020050	005103				COM	R3
3735	020052			ELSE			
3736	020052	000401				BR	50350\$
3737	020054						
3738	020054			LET R3 := R3 + #1	;SET 1ST PASS IF NEW PASS AND		
3739	020054	005203				INC	R3
3740	020056			ENDIF	;RESTARTING		
3741	020056						
3742	020056			ELSE			
3743	020056	000401				BR	50351\$
3744	020060						
3745	020060			LET R3 := R3 + #1	;SET 1ST PASS IF NEW PASS AND		
3746	020060	005203				INC	R3
3747	020062			ENDIF	;STARTING		
3748	020062						
3749	020062			ENDIF	;DO NOT UPDATE IT ON CONTINUE		
3750	020062						
3751	020062			ENDIF	;OR ON POWER FAIL		
3752	020062						
3753	020062	004737	015332	JSR PC ,FIRSTU	;INIT DEVICE POINTER.		
3754	020066			LET R2 := #0	;INIT DEVICE COUNTER.		
3755	020066	005002				CLR	R2
3756	020070			WHILE DEVTBL(R5) NE #END DO			
3757	020070						
3758	020070	026527	002532				
3759	020076	001450	177777			CMP	DEVTBL(R5) ,#END
3760	020100			LET R2 := R2 + #1		BEQ	50353\$
3761	020100	005202				INC	R2
3762	020102			LET R0 := R5 SHIFT -1			
3763	020102	010500				MOV	R5 ,R0
3764	020104	006200				ASR	R0
3765	020106			GPHARD R0 ,R0	;GET HARDWARE P TABLE FROM SUPER.		
3766	020106	104442				TRAP	C\$GPHRD
3767	020110			IFCOND CS THEN			
3768	020110	103036				BCC	50354\$
3769	020112			LET TSSR(R5) := (R0)	;SAVE TSSR ADDRESS.		
3770	020112	011065	002462			MOV	(R0) ,TSSR(R5)
3771	020116			LET TSDB(R5) := (R0)+ - #2	;SAVE TSDB ADDRESS.		
3772	020116	012065	002452			MOV	(R0)+ ,TSDB(R5)
3773	020122	162765	000002			SUB	#2 ,TSDB(R5)
3774	020130		002452				
3775	020130	011065	002472	LET TSVCT(R5) := (R0)	;SAVE INTERRUPT VECTOR ADDRESS.		
3776	020134			SETVEC TSVCT(R5) ,TS4INT(R5) ,#INTPRI	;SET UP INTERUPT PROCESSING CONDITIONS.		
3777	020134	012746	000340			MOV	#INTPRI ,-(SP)



3778	020140	016546	002512				MOV	TS4INT(R5),-(SP)
3779	020144	016546	002472				MOV	TSVCT(R5),-(SP)
3780	020150	012746	000003				MOV	#3, -(SP)
3781	020154	104437					TRAP	C\$SVEC
3782	020156	062706	000010				ADD	#10, SP
3783	020162							
3784	020162	005703			IF R3 NE #0 THEN			;ACTUAL PASSCOUNT UPDATE PER R3
3785	020164	001410					TST	R3
3786	020166				IF R3 LT #0 THEN		BEQ	50355\$
3787	020166	005703					TST	R3
3788	020170	002003					BGE	50356\$
3789	020172				LET PASCNT(R5) := PASCNT(R5) + #1			
3790	020172	005265	003254				INC	PASCNT(R5)
3791	020176				ELSE			
3792	020176	000403					BR	50357\$
3793	020200							50356\$:
3794	020200				LET PASCNT(R5) := #1			
3795	020200	012765	000001	003254			MOV	#1, PASCNT(R5)
3796	020206				ENDIF			
3797	020206							50357\$:
3798	020206				ENDIF			
3799	020206							50355\$:
3800	020206				ENDIF			
3801	020206							50354\$:
3802	020206				LET RECCNT(R5) := #0			;CLEAR RECORD COUNT
3803	020206	005065	003324				CLR	RECCNT(R5)
3804	020212	004737	015400		JSR PC, NEXTU			;DO IT FOR ALL DEVICES.
3805	020216				ENDDO			
3806	020216	000724					BR	50352\$
3807	020220							50353\$:
3808								
3809	020220				IF R2 EQ #0 THEN			;IF THERE ARE NO UNITS:
3810	020220	005702					TST	R2
3811	020222	001026					BNE	50360\$
3812	020224				PRINTF #AUDRPM			;PRINT ALL UNITS DROPPED,
3813	020224	012746	004645				MOV	#AUDRPM, -(SP)
3814	020230	012746	000001				MOV	#1, -(SP)
3815	020234	010600					MOV	SP, R0
3816	020236	104417					TRAP	C\$PNTF
3817	020240	062706	000004				ADD	#4, SP
3818	020244				DELAY 2000.			;GO TO SUPERVISOR, WAIT 2 SECONDS.
3819	020244	012727	003720				MOV	#2000., (PC)+
3820	020250	000000					.WORD	0
3821	020252	013727	002116				MOV	L\$DLY, (PC)+
3822	020256	000000					.WORD	0
3823	020260	005367	177772				DEC	-6(PC)
3824	020264	001375					BNE	-.4
3825	020266	005367	177756				DEC	-22(PC)
3826	020272	001367					BNE	.-20
3827	020274				BREAK			;GO TO SUPERVISOR, CHECK TTY.
3828	020274	104422					TRAP	C\$BRK
3829	020276				DOCLN			;DO CLEAN CODE + ABORT PASS.
3830	020276	104444					TRAP	C\$DCLN
3831	020300				ENDIF			
3832	020300							50360\$:
3833								

```

3834 020300          SETPRI #PRI00          ;LOWER CPU PRIORITY TO 0
3835 020300 012700 000000          MOV #PRI00,R0
3836 020304 104441          TRAP C$SPRI
3837 020306          IFB IREC EQ #0 AND #ADR NOTSETIN OPFLAG THEN ;IF ERROR RECOVERY IS ENABLED
3838 020306 105737 002211          TSTB IREC
3839 020312 001145          BNE 50361$
3840 020314 032737 000020 003456          BIT #ADR,OPFLAG
3841 020322 001141          BNE 50361$
3842 020324 004737 015332          JSR PC,FIRSTU          ;AND AUTO-DROP NOT CALLED, THEN SET UP FOR FIRST
3843 020330          WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
3844 020330          50362$:
3845 020330 026527 002532 177777          CMP DEVTBL(R5),#END
3846 020336 001533          BEQ 50363$
3847 020340          BEGIN COUNTER          ;START 3.5 MINUTE COUNTER
3848 020340          INCR TIME1 FROM #1 TO #25 BY #1
3849 020340 012737 000001 003364          MOV #1,TIME1
3850 020346 000402          BR 50365$
3851 020350          50366$:
3852 020350 005237 003364          INC TIME1
3853 020354          50365$:
3854 020354 023727 003364 000025          CMP TIME1,#25
3855 020362 003106          BGT 50367$
3856 020364          LEI ;DB(R5) := #GSCPK ;AND GET UNITS STATUS
3857 020364 012775 002320 002452          MOV #GSCPK,@TSDB(R5)
3858 020372          DEL 1          ;WAIT
3859 020372 012727 000001          MOV #1,(PC)+
3860 020376 000000          .WORD 0
3861 020400 013727 002116          MOV LSDLY,(PC)+
3862 020404 000000          .WORD 0
3863 020406 005367 177772          DEC -6(PC)
3864 020412 001375          BNE -4
3865 020414 005367 177756          DEC -22(PC)
3866 020420 001367          BNE -20
3867 020422          IF #TS.SSR SETIN @TSSR(R5) THEN
3868 020422 032775 000200 002462          BIT #TS.SSR,@TSSR(R5)
3869 020430 001420          BEQ 50370$
3870 020432          IF #TS.OFL NOTSETIN @TSSR(R5) THEN
3871 020432 032775 000100 002462          BIT #TS.OFL,@TSSR(R5)
3872 020440 001001          BNE 50371$
3873 020442          LEAVE COUNTER          ;EXIT COUNTER WHEN UNIT ON LINE
3874 020442 000456          BR 50364$
3875 020444          ELSE
3876 020444          50371$:
3877 020444          PRINTF #OFLINM,DEVTBL(R5) ;PRINT UNIT OFF LINE EVERY 10 SEC
3878 020444 016546 002532          MOV DEVTBL(R5),-(SP)
3879 020450 012746 005127          MOV #OFLINM, -(SP)
3880 020454 012746 000002          MOV #2, -(SP)
3881 020460 010600          MOV SP,R0
3882 020462 104417          TRAP C$PNTF
3883 020464 062706 000006          ADD #6,SP
3884 020470          ENDIF
3885 020470          50372$:
3886 020470          ELSE
3887 020470 000412          BR 50373$
3888 020472          50370$:
3889 020472          PRINTF #NRDYM,DEVTBL(R5)

```



3890	020472	016546	002532				MOV	DEVTBL (R5), -(SP)
3891	020476	012746	021424				MOV	#NRDYM, -(SP)
3892	020502	012746	000002				MOV	#2, -(SP)
3893	020506	010600					MOV	SP, R0
3894	020510	104417					TRAP	C\$PNTF
3895	020512	062706	000006				ADD	#6, SP
3896	020516				ENDIF			
3897	020516							50373\$:
3898	020516				INCR TIME2 FROM #1 TO #13 BY #1			
3899	020516	012737	000001	003366			MOV	#1, TIME2
3900	020524	000402					BR	50374\$
3901	020526							50375\$:
3902	020526	005237	003366				INC	TIME2
3903	020532							50374\$:
3904	020532	023727	003366	000013			CMP	TIME2, #13
3905	020540	003016					BGT	50376\$
3906	020542				DELAY 1000.			
3907	020542	012727	001750					;WAIT FOR UNIT TO BE SET ON-LINE
3908	020546	000000					MOV	#1000., (PC)+
3909	020550	013727	002116				.WORD	0
3910	020554	000000					MOV	L\$DLY, (PC)+
3911	020556	005367	177772				.WORD	0
3912	020562	001375					DEC	-6(PC)
3913	020564	005367	177756				BNE	-.4
3914	020570	001367					DEC	-22(PC)
3915	020572				BREAK		BNE	.-20
3916	020572	104422						;ALLOW TERMINAL INTERRUPT
3917	020574				ENDINC		TRAP	C\$BRK
3918	020574	000754					BR	50375\$
3919	020576							50376\$:
3920	020576				ENDINC		BR	50366\$
3921	020576	000664						50367\$:
3922	020600							
3923	020600				END JUNTER			50364\$:
3924	020600							
3925	020600				IF TIME1 GT #25 THEN			;IF OFF LINE FOR 3.5 MINUTES
3926	020600	023727	003364	000025			CMP	TIME1, #25
3927	020606	003404					BLE	50377\$
3928	020610	004737	011206					;GET MESSAGE PACKET
3929	020614	004737	011702		JSR PC, MOVMSG			;PRINT ERROR AND DROP OFF LINE UNIT
3930	020620				JSR PC, TCC1			
3931	020620				ENDIF			50377\$:
3932								;REPEAT UNTIL ON LINE OR TIMED OUT.
3933	020620	004737	015400		JSR PC, NEXTU			;SET UP FOR NEXT UNIT.
3934	020624				ENDDO			
3935	020624	000641					BR	50362\$
3936	020626							50363\$:
3937	020626				ENDIF			
3938	020626							50361\$:
3939	020626				IFB PWRFLG EQ #0 THEN			
3940	020626	105737	003453				TSTB	PWRFLG
3941	020632	001026					BNE	50400\$
3942	020634				MEMORY DATAWT			;REQUEST MEMORY FROM SUPER FOR RD/WR BUFFERS.
3943	020634	104431					TRAP	C\$MEM
3944	020636	010037	003334				MOV	R0, DATAWT
3945	020642				LET DATARD := DATAWT + #DATCNT			;SET RD BFR AD

```

3946 020642 013737 003334 003336
3947 020650 062737 004000 003336
3948 020656
3949 020656 027727 162452 004000
3950 020664 002011
3951 020666
3952 020666 012746 020734
3953 020672 012746 000001
3954 020676 010600
3955 020700 104417
3956 020702 062706 000004
3957 020706
3958 020706 104444
3959 020710
3960 020710
3961 020710
3962 020710
3963
3964 020710
3965 020710 105037 002212
3966 020714
3967 020714 012703 003452
3968 020720 004737 011136
3969 020724
3970 020724 105037 003453
3971
3972 020730
3973 020730 104432
3974 020732 000104
3975 020734 040445 051106 042505 MEMOM: .ASCII /%AFREE MEMO TOO SMALL FOR RD-WR BFRS%/
3976 020742 046440 046505 020117
3977 020750 047524 020117 046523
3978 020756 046101 020114 047506
3979 020764 020122 042122 053455
3980 020772 020122 043102 051522
3981 021000 047045
3982 021002 040445 042522 046055 .ASCIIZ /%ARE-LOAD IN LARGER MEMO%/
3983 021010 040517 020104 047111
3984 021016 046040 051101 042507
3985 021024 020122 042515 047515
3986 021032 047045 000
3987 021036
3988
3989 021036
3990 021036
3991 021036 104411 L10012: TRAP C$INIT
  
```

```

MOV DATAW, DATARD
ADD #DATCNT, DATARD
; WHEN NOT ENOUGH FREE MEMO AVAILABLE
CMP @DATAW, #DATCNT
BGE 50401$
PRINTF #MEMOM ; WARN OPERATOR
MOV #MEMOM, -(SP)
MOV #1, -(SP)
MOV SP, R0
TRAP C$PNTF
ADD #4, SP
DOCLN ; AND ABORT PASS
TRAP C$DCLN
ENDIF ; DIAG MUST BE RE-LOADED IN A CPU WITH LARGER MEMO
50401$:
50400$:
LET CHGFLG :B= #0 ; CLR CHANGE CMD SEQ TBL FLAG.
CLR CHGFLG
LET R3 := #ENDFLG
MOV #ENDFLG, R3
JSR PC, CLRERR ; CLEAR ALL FLAGS.
LET PWRFLG :B= #0 ; CLEAR THE POWER FAIL FLAG.
CLR PWRFLG
EXIT INIT
TRAP C$EXIT
.word L10012-.

```





4048	021206	032775	000200	002462		BIT	#TS.SSR,@TSSR(R5
4049	021214	001423				BEQ	50406\$
4050	021216				IF #TS.OFL SETIN @TSSR(R5) THEN		
4051	021216	032775	000100	002462		BIT	#TS.OFL,@TSSR(R5
4052	021224	001416				BEQ	50407\$
4053	021226				LET FTLCNT(R5) := FTLCNT(R5) + #1		
4054	021226	005265	003314		PRINTF #OFLINM,DEVTBL(R5)	INC	FTLCNT(R5)
4055	021232						
4056	021232	016546	002532			MOV	DEVTBL(R5),-(SP)
4057	021236	012746	005127			MOV	#OFLINM, -(SP)
4058	021242	012746	000002			MOV	#2, -(SP)
4059	021246	010600				MOV	SP,R0
4060	021250	104417				TRAP	C\$PNTF
4061	021252	062706	000006			ADD	#6,SP
4062	021256	004737	015554		JSR PC,DROPUA		
4063	021262				ENDIF		
4064	021262						50407\$:
4065	021262				ELSE		
4066	021262	000416				BR	50410\$
4067	021264						50406\$:
4068	021264				LET FTLCNT(R5) := FTLCNT(R5) + #1		
4069	021264	005265	003314		PRINTF #NRDYM,DEVTBL(R5)	INC	FTLCNT(R5)
4070	021270						
4071	021270	016546	002532			MOV	DEVTBL(R5),-(SP)
4072	021274	012746	021424			MOV	#NRDYM, -(SP)
4073	021300	012746	000002			MOV	#2, -(SP)
4074	021304	010600				MOV	SP,R0
4075	021306	104417				TRAP	C\$PNTF
4076	021310	062706	000006			ADD	#6,SP
4077	021314	004737	015554		JSR PC,DROPUA		
4078	021320				ENDIF		
4079	021320						50410\$:
4080	021320				ENDIF		
4081	021320						50405\$:
4082	021320	004737	015400		JSR PC,NEXTU		
4083	021324				ENDDO		
4084	021324	000647					50403\$:
4085	021326					BR	50402\$
4086							
4087	021326				ENDAUTO		
4088	021326						
4089	021326	104461			L10013:	TRAP	C\$AUTO
4090							
4091	021330	040445	052502	020123	AUTODM: .ASCII /%ABUS TRAP AT %06%N/		
4092	021336	051124	050101	040440			
4093	021344	020124	047445	022466			
4094	021352	116					
4095	021353	045	044501	052116	.ASCIZ /%AINTERFACE BAD OR NOT SET TO ABOVE AD%N/		
4096	021360	051105	040506	042503			
4097	021366	041040	042101	047440			
4098	021374	020122	047516	020124			
4099	021402	042523	020124	047524			
4100	021410	040440	047502	042526			
4101	021416	040440	022504	000116			
4102	021424	040445	0471	052111	NRDYM: .ASCIZ /%AUNIT %D1%A NOT RDY%N/		
4103	021432	022440	030504	040445			



4104 021440 047040 052117 051040  
4105 021446 054504 047045 000  
4106 021454  
4107  
4108  
4109  
4110  
4111  
4112 021454  
4113 021454 105237 003454  
4114 021460 000002  
4115  
4116  
4117

.EVEN  
:  
: DEVICE BUS TRAP HANDLER  
: OUTPUT: TRAPD4 BYTE 1: TRAPED AT 4  
: 0: NO TRAP  
TRAP4:: LET TRAPD4 :B= TRAPD4 + #1  
RTI

INCB TRAPD4

4118  
 4119  
 4120  
 4121  
 4122  
 4123  
 4124  
 4125  
 4126  
 4127  
 4128  
 4129  
 4130  
 4131  
 4132  
 4133  
 4134  
 4135  
 4136  
 4137  
 4138  
 4139  
 4140  
 4141  
 4142  
 4143  
 4144  
 4145  
 4146  
 4147  
 4148  
 4149  
 4150

021462  
 021462  
 021462 004737 015332  
 021466  
 021466  
 021466 026527 002532 177777  
 021474 001410  
 021476 004737 011152  
 021502  
 021502 016500 002472  
 021506 104436  
 021510 004737 015400  
 021514  
 021516 000764  
 021516  
 021516 104432  
 021520 000002  
 021522  
 021522  
 021522 104412

```

.SBTTL CLEANUP CODING SECTION
:++
: THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
: AT THE END OF EACH PASS.
:--

      BGNCLN
L$CLEAN::

      JSR PC,FIRSTU          ;FIND FIRST UNIT.
      WHILE DEVTBL(R5) NE #END DO
                                50411$:
                                CMP DEVTBL(R5),#END
                                BEQ 50412$
      JSR PC,WSSR             ;WAIT FOR UNIT READY OR TIMEOUT,
      CLRVEC TSVCT(R5)       ;RELEASE INTERRUPT VECTORS FOR ALL DEV.
                                MOV TSVCT(R5),R0
                                TRAP C$CVEC
      JSR PC,NEXTU           ;FIND NEXT UNIT.
      ENDDO

                                BR 50411$
                                50412$:
      EXIT CLN
                                TRAP C$EXIT
                                .WORD L10014-.

      .EVEN
      ENDCLN
L10014:
                                TRAP C$CLEAN
  
```



```

4151 .SBTTL DROP UNIT SECTION
4152
4153 ;++
4154 ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4155 ; TO NO LONGER BE TESTED. THAT CODE SHALL BE EXECUTED WHEN DODU
4156 ;MACRO IS CALLED WHILE IDU FLAG IS NOT SET BY OPERATOR
4157 ;--
4158
4159 021524 BGNDU
4160 021524 L$DU::
4161
4162 021524 LET R5 := R0 SHIFT 1 ;R5 = LOGICAL DEVICE NUMBER X 2.
4163 021524 010005 MOV R0,R5
4164 021526 006305 ASL R5
4165 021530 LET DEVTBL(R5) := #NINUSE ;SET NOT IN USE FLAG FOR THE DEVICE.
4166 021530 012765 177774 002532 MOV #NINUSE,DEVTBL(R
4167 021536 CLRVEC TSVCT(R5) ;RELEASE THE INTERRUPT VECTOR.
4168 021536 016500 002472 MOV TSVCT(R5),R0
4169 021542 104436 TRAP C$CVEC
4170 021544 PRINTF #DROPDM,DROPN ;PRINT DROP DEVICE MESSAGE
4171 021544 013746 015640 MOV DROPN,-(SP)
4172 021550 012746 004616 MOV #DROPDM,-(SP)
4173 021554 012746 000002 MOV #2,-(SP)
4174 021560 010600 MOV SP,R0
4175 021562 104417 TRAP C$PNTF
4176 021564 062706 000006 ADD #6,SP
4177
4178 021570 EXIT DU
4179 021570 000167 .WORD J$JMP
4180 021572 000000 .WORD L10015-2-.
4181 .EVEN
4182
4183 021574 ENDDU
4184 021574 L10015:
4185 021574 104453 TRAP C$DU
  
```

```

4186      .SBTTL  ADD UNIT SECTION
4187
4188
4189      :++
4190      : THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4191      : TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
4192      : 'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
4193      :--
4194      021576      BGNUAU
4195      021576      L$AU::
4196
4197
4198      021576      LET R5 := R0 SHIFT 1          ;R5 = LOGICAL DEVICE NUMBER X 2.
4199      021576      010005      MOV          R0,R5
4200      021600      006305      ASL          R5
4201      021602      LET DEVTBL(R5) := R0      ;STORE UNIT # IN DEVICE TABLE.
4202      021602      010065      002532      MOV          R0,DEVTBL(R5)
4203      021606      GPHARD R0,R0              ;GET HARDWARE P TABLE FROM SUPER.
4204      021606      104442      TRAP        C$GPHRD
4205      021610      LET TSSR(R5) := (R0)      ;SAVE TSSR ADDRESS.
4206      021610      011065      002462      MOV          (R0),TSSR(R5)
4207      021614      LET TSDB(R5) := (R0)+ - #2 ;SAVE TSDB ADDRESS.
4208      021614      012065      002452      MOV          (R0)+,TSDB(R5)
4209      021620      162765      000002      002452      SUB          #2,TSDB(R5)
4210      021626      LET TSVCT(R5) := (R0)      ;SAVE INTERRUPT VECTOR ADDRESS.
4211      021626      011065      002472      MOV          (R0),TSVCT(R5)
4212      021632      SETVEC TSVCT(R5),TS4INT(R5),#INTPRI ;SET UP INTERUPT PROCESSING CONDITIONS.
4213      021632      012746      000340      MOV          #INTPRI,-(SP)
4214      021636      016546      002512      MOV          TS4INT(R5),-(SP)
4215      021642      016546      002472      MOV          TSVCT(R5),-(SP)
4216      021646      012746      000003      MOV          #3,-(SP)
4217      021652      104437      TRAP        C$SVEC
4218      021654      062706      000010      ADD          #10,SP
4219      021660      LET INTFLG(R5) := #0      ;CLEAR INTERRUPT FLAGS.
4220      021660      005065      003416      CLR          INTFLG(R5)
4221
4222      021664      EXIT      AU
4223      021664      000167      .WORD      J$JMP
4224      021666      000000      .WORD      L10016-2-.
4225
4226      .EVEN
4227
4228      021670      ENDAU
4229      021670      L10016:
4230      021670      104452      TRAP        C$AU
4231
4232      021672      ENDMOD
4233

```



```

4234
4235      .TITLE HARDWARE TESTS
4236
4237      .SBTTL TEST 1: BASIC FUNCTIONS.
4238
4239      :++
4240      : TEST TO EXECUTE ALL TS04 FUNCTIONS.
4241      :--
4242
4243 021672      BGNMOD
4244
4245 021672      BGNTST
4246 021672      T1::
4247
4248 021672      LET RANDOM :B= #0      ;CLR THE RANDOM OPERATIONS FLAG.
4249 021672 105037 003441      ;CLR EXPECT BOT FLAG.      CLRB      RANDOM
4250 021676      LET EXPBOT :B= #0      ;CLR EXPECT BOT FLAG.      CLRB      EXPBOT
4251 021676 105037 003440
4252
4253 021702      BGNSUB      ;SUBTEST 1 - SET CHAR, DRIVE INIT, GET STATUS.
4254 021702      T1.1:
4255 021702 104402      TRAP      C$BSUB
4256
4257 021704      LET R2 := #BFSEQO      ;ADR,OF CMD SEQ.
4258 021704 012702 022530      ;SET UP CMD SEQ.      MOV      #BFSEQO,R2
4259 021710 004737 022504      JSR      PC,BFSEQ
4260 021714 004737 006510      JSR      PC,EXALL      ;EXECUTE CMD SEQ ON ALL DEVICES.
4261 021720 004737 015332      JSR      PC,FIRSTU      ;FIND THE FIRST UNIT.
4262 021724      WHILE DEVTBL(R5) NE #END DO      ;WHILE THERE ARE MORE DEVICES:
4263 021724
4264 021724 026527 002532 177777      50413$:
4265 021732 001434      CMP      DEVTBL(R5),#END
4266 021734      LET R2 := MSGPKA(R5)      ;GET MSG PACKET ADR,      BEQ      50414$
4267 021734 016502 002502      MOV      MSGPKA(R5),R2
4268 021740      LET R2 := R2 + #12      ;GET XSTAT2 ADR,
4269 021740 062702 000012      ADD      #12,R2
4270 021744      LET TS4CL(R5) := (R2) CLR.BY #177400 ;STORE CODE LEVEL FROM DTR BYTE,
4271 021744 011265 002522      MOV      (R2),TS4CL(R5)
4272 021750 042765 177400 002522      BIC      #177400,TS4CL(R5)
4273 021756      IF PASCNT(R5) EQ #1 THEN      ;IF THIS IS PASS 1 THEN:
4274 021756 026527 003254 000001      CMP      PASCNT(R5),#1
4275 021764 001014      BNE      50415$
4276 021766      PRINTF #CODELM,DEVTBL(R5),TS4CL(R5) ;PRINT THE TS04 MICROCODE LEVEL.
4277 021766 016546 002522      MOV      TS4CL(R5),-(SP)
4278 021772 016546 002532      MOV      DEVTBL(R5),-(SP)
4279 021776 012746 003772      MOV      #CODELM,-(SP)
4280 022002 012746 000003      MOV      #3,-(SP)
4281 022006 010600      MOV      SP,R0
4282 022010 104417      TRAP      C$PNTF
4283 022012 062706 000010      ADD      #10,SP
4284 022016      ENDIF
4285 022016
4286 022016 004737 015400      JSR      PC,NEXTU      ;FIND NEXT UNIT.      50415$:
4287 022022      ENDDO
4288 022022 000740      BR      50413$
4289 022024      50414$:

```

4290	022024			ENDSUB			
4291	022024			L10020:			
4292	022024	104403				TRAP	C\$ESUB
4293							
4294	022026			BGNSUB		;SUBTEST 2 - REWIND.	
4295	022026			T1.2:			
4296	022026	104402				TRAP	C\$BSUB
4297							
4298	022030			LET R2 := #BFSEQ1		;ADR OF CMD SEQ.	
4299	022030	012702	022602			MOV	#BFSEQ1,R2
4300	022034	004737	022504	JSR PC,BFSEQ		;SET UP CMD SEQ.	
4301	022040	004737	006510	JSR PC,EXALL		;EXECUTE CMD SEQ ON ALL DEVICES.	
4302	022044			LET STAFLG :B= #0		;CLEAR START FLAG	
4303	022044	105037	003452			CLRB	STAFLG
4304	022050			ENDSUB			
4305	022050			L10021:			
4306	022050	104403				TRAP	C\$ESUB
4307							
4308	022052			BGNSUB		;SUBTEST 3 - WRITE/VERIFY.	
4309	022052			T1.3:			
4310	022052	104402				TRAP	C\$BSUB
4311							
4312	022054			LET R2 := #BFSEQ2		;ADR OF CMD SEQ.	
4313	022054	012702	022614			MOV	#BFSEQ2,R2
4314	022060	004737	022504	JSR PC,BFSEQ		;SET UP CMD SEQ.	
4315	022064	004737	006510	JSR PC,EXALL		;EXECUTE CMD SEQ ON ALL DEVICES.	
4316	022070			ENDSUB			
4317	022070			L10022:			
4318	022070	104403				TRAP	C\$ESUB
4319							
4320	022072			BGNSUB		;SUBTEST 4 - WRITE TAPE MARK, ERASE.	
4321	022072			T1.4:			
4322	022072	104402				TRAP	C\$BSUB
4323							
4324	022074			LET R2 := #BFSEQ3		;ADR OF CMD SEQ.	
4325	022074	012702	022706			MOV	#BFSEQ3,R2
4326	022100	004737	022504	JSR PC,BFSEQ		;SET UP CMD SEQ.	
4327	022104	004737	006510	JSR PC,EXALL		;EXECUTE CMD SEQ ON ALL DEVICES.	
4328	022110			ENDSUB			
4329	022110			L10023:			
4330	022110	104403				TRAP	C\$ESUB
4331							
4332	022112			BGNSUB		;SUBTEST 5 - SPACE FILES.	
4333	022112			T1.5:			
4334	022112	104402				TRAP	C\$BSUB
4335							
4336	022114			LET R2 := #BFSEQ4		;ADR OF CMD SEQ.	
4337	022114	012702	022760			MOV	#BFSEQ4,R2
4338	022120	004737	022504	JSR PC,BFSEQ		;SET UP CMD SEQ.	
4339	022124	004737	006510	JSR PC,EXALL		;EXECUTE CMD SEQ ON ALL DEVICES.	
4340	022130			ENDSUB			
4341	022130			L10024:			
4342	022130	104403				TRAP	C\$ESUB
4343							
4344	022132			BGNSUB		;SUBTEST 6 - SPACE RECORDS.	
4345	022132			T1.6:			



```

4346 022132 104402                                TRAP  C$BSUB
4347
4348 022134                                LET R2 := #BFSEQ5                ;ADR OF CMD SEQ.
4349 022134 012702 023022                                ;SET UP CMD SEQ.                MOV  #BFSEQ5,R2
4350 022140 004737 022504                                ;EXECUTE CMD SEQ ON ALL DEVICES.
4351 022144 004737 006510                                JSR  PC,BFSEQ
4352 022150                                JSR  PC,EXALL
4353 022150                                ENDSUB
4354 022150 104403                                L10025:                                TRAP  C$ESUB
4355
4356 022152                                BGNSUB                                ;SUBTEST 7 - WRITE RETRY.
4357 022152                                T1.7:
4358 022152 104402                                TRAP  C$BSUB
4359
4360 022154                                LET R2 := #BFSEQ6                ;ADR OF CMD SEQ.
4361 022154 012702 023074                                ;SET UP CMD SEQ.                MOV  #BFSEQ6,R2
4362 022160 004737 022504                                ;EXECUTE CMD SEQ ON ALL DEVICES.
4363 022164 004737 006510                                JSR  PC,BFSEQ
4364 022170                                JSR  PC,EXALL
4365 022170                                ENDSUB
4366 022170 104403                                L10026:                                TRAP  C$ESUB
4367
4368 022172                                BGNSUB                                ;SUBTEST 8 - READ REV RETRY.
4369 022172                                T1.8:
4370 022172 104402                                TRAP  C$BSUB
4371
4372 022174                                LET R2 := #BFSEQ7                ;ADR OF CMD SEQ.
4373 022174 012702 023126                                ;SET UP CMD SEQ.                MOV  #BFSEQ7,R2
4374 022200 004737 022504                                ;EXECUTE CMD SEQ ON ALL DEVICES.
4375 022204 004737 006510                                JSR  PC,BFSEQ
4376 022210                                JSR  PC,EXALL
4377 022210                                ENDSUB
4378 022210 104403                                L10027:                                TRAP  C$ESUB
4379
4380 022212                                BGNSUB                                ;SUBTEST 9 - READ FWD RETRY.
4381 022212                                T1.9:
4382 022212 104402                                TRAP  C$BSUB
4383
4384 022214                                LET R2 := #BFSEQ8                ;ADR OF CMD SEQ.
4385 022214 012702 023160                                ;SET UP CMD SEQ.                MOV  #BFSEQ8,R2
4386 022220 004737 022504                                ;EXECUTE CMD SEQ ON ALL DEVICES.
4387 022224 004737 006510                                JSR  PC,BFSEQ
4388 022230                                JSR  PC,EXALL
4389 022230                                ENDSUB
4390 022230 104403                                L10030:                                TRAP  C$ESUB
4391
4392 022232                                BGNSUB                                ;SUBTEST 10- CLEAN.
4393 022232                                T1.10:
4394 022232 104402                                TRAP  C$BSUB
4395
4396 022234                                LET R2 := #BFSEQ9                ;ADR OF CMD SEQ.
4397 022234 012702 023212                                ;SET UP CMD SEQ.                MOV  #BFSEQ9,R2
4398 022240 004737 022504                                ;EXECUTE CMD SEQ ON ALL DEVICES.
4399 022244 004737 006510                                JSR  PC,BFSEQ
4400 022250                                JSR  PC,EXALL
4401 022250                                ENDSUB
L10031:

```







```

4479      ;      SUBROUTINE TO MOVE A COMMAND SEQUENCE TO THE SEQUENCE TABLE.
4480      ;      INPUTS:          R2 = FWA OF COMMAND SEQUENCE.
4481      ;      OUTPUTS:
4482      ;      REGISTERS:
4483      ;      CALLS:
4484
4485 022504      BFSEQ: LET R1 := #CMDSEQ          ;INIT SEQ TABLE ADDRESS.
4486 022504 012701 003460      WHILE (R2) NE #END DO      ;WHILE THERE ARE MORE COMMANDS:
4487 022510      ;                               50420$:
4488 022510      ;                               CMP      (R2),#END
4489 022510 021227 177777      ;                               BEQ      50421$
4490 022514 001402      ;                               LET (R1)+ := (R2)+
4491 022516      ;                               ;MOVE COMMANDS TO SEQ TABLE.
4492 022516 012221      ;                               MOV      (R2)+,(R1)+
4493 022520      ENDDO
4494 022520 000773      ;                               BR      50420$
4495 022522      ;                               50421$:
4496 022522      LET (R1) := #END          ;STORE END OF SEQUENCE CODE.
4497 022522 012711 177777      ;                               MOV      #END,(R1)
4498 022526 000207      RTS      PC          ;RETURN.
4499
4500
4501
4502      ;      BASIC FUNCTION COMMAND SEQUENCE
4503
4504 022530 140004      BFSEQ0: .WORD      SCH          ;SET CHAR. 200.      (1)
4505 022532 000200      ;                               200
4506 022534 000001      ;                               1
4507 022536 000000      ;                               0
4508 022540 100013      ;DRIVE INIT.      (2)
4509 022542 000001      ;                               1
4510 022544 000001      ;                               1
4511 022546 000000      ;                               0
4512 022550 140004      ;SET CHAR. 20      (3)
4513 022552 000020      ;                               20
4514 022554 000001      ;                               1
4515 022556 000000      ;                               0
4516 022560 100017      ;GET STATUS.      (4)
4517 022562 000001      ;                               1
4518 022564 000001      ;                               1
4519 022566 000000      ;                               0
4520 022570 140004      ;SET CHAR. 40.      (5)
4521 022572 000040      ;                               40
4522 022574 000001      ;                               1
4523 022576 000000      ;                               0
4524 022600 177777      .WORD      END
4525
4526 022602 102010      BFSEQ1:      RWD          ;REWIND TWICE.      (6)
4527 022604 000001      ;                               1
4528 022606 000002      ;                               2
4529 022610 000000      ;                               0
4530 022612 177777      .WORD      END
4531
4532 022614 104105      BFSEQ2:      WTV          ;WRITE/VERIFY PAT 1.      (7)
4533 022616 004000      ;                               DATCNT
4534 022620 000001      ;                               1
    
```



HARDWARE TESTS MACY11 30(1046) 11-OCT-79 14:02 PAGE 108  
 CZTSHC.P11 11-OCT-79 13:59 TEST 1: BASIC FUNCTIONS.

4535	022622	000001		1		
4536	022624	104105		WTV	;WTV PAT 2.	(8)
4537	022626	004000		DATCNT		
4538	022630	000001		1		
4539	022632	000002		2		
4540	022634	104105		WTV	;WTV PAT 3.	(9)
4541	022636	004000		DATCNT		
4542	022640	000001		1		
4543	022642	000003		3		
4544	022644	104105		WTV	;WTV PAT 4.	(10)
4545	022646	004000		DATCNT		
4546	022650	000001		1		
4547	022652	000004		4		
4548	022654	104105		WTV	;WTV PAT 5.	(11)
4549	022656	004000		DATCNT		
4550	022660	000001		1		
4551	022662	000005		5		
4552	022664	104105		WTV	;WTV PAT 6.	(12)
4553	022666	004000		DATCNT		
4554	022670	000001		1		
4555	022672	000006		6		
4556	022674	104105		WTV	;WTV PAT 0.	(13)
4557	022676	004000		DATCNT		
4558	022700	000001		1		
4559	022702	000000		0		
4560	022704	177777	.WORD	END		
4561						
4562	022706	100011	BFSEQ3:	WTM	;WRITE TAPE MARK.	(14)
4563	022710	000001		1		
4564	022712	000001		1		
4565	022714	000000		0		
4566	022716	104005		WRT	;WRITE 10 RECORDS.	(15)
4567	022720	004000		DATCNT		
4568	022722	000010		10		
4569	022724	000001		1		
4570	022726	100411		ERS	;ERASE 10 TIMES.	(16)
4571	022730	000001		1		
4572	022732	000010		10		
4573	022734	000000		0		
4574	022736	100011		WTM	;WRITE TAPE MARK.	(17)
4575	022740	000001		1		
4576	022742	000001		1		
4577	022744	000000		0		
4578	022746	101011		WTR	;WTR RETRY	(18)
4579	022750	000001		1		
4580	022752	000001		1		
4581	022754	000000		0		
4582	022756	177777	.WORD	END		
4583						
4584	022760	105410	BFSEQ4:	SFR	;SPACE 2 FILES REV.	(19)
4585	022762	000002		2		
4586	022764	000001		1		
4587	022766	000000		0		
4588	022770	105010		SFF	;SPACE 2 FILES FWD.	(20)
4589	022772	000002		2		
4590	022774	000001		1		

4591	022776	000000		0		
4592	023000	105410		SFR	;SPACE 2 FILES REV.	(21)
4593	023002	000001		1		
4594	023004	000002		2		
4595	023006	000000		0		
4596	023010	105010		SFF	;SPACE 2 FILES FWD.	(22)
4597	023012	000001		1		
4598	023014	000002		2		
4599	023016	000000		0		
4600	023020	177777	.WORD	END		
4601						
4602	023022	102010	BFSEQ5:	RWD	;REWIND.	(23)
4603	023024	000001		1		
4604	023026	000001		1		
4605	023030	000000		0		
4606	023032	104010		SRF	;SPACE 7 RECORDS FWD.	(24)
4607	023034	000007		7		
4608	023036	000001		1		
4609	023040	000000		0		
4610	023042	104410		SRR	;SPACE 7 RECORDS REV.	(25)
4611	023044	000007		7		
4612	023046	000001		1		
4613	023050	000000		0		
4614	023052	104010		SRF	;SPACE 7 RECORDS FWD.	(26)
4615	023054	000001		1		
4616	023056	000007		7		
4617	023060	000000		0		
4618	023062	104410		SRR	;SPACE 7 RECORDS REV.	(27)
4619	023064	000001		1		
4620	023066	000007		7		
4621	023070	000000		0		
4622	023072	177777	.WORD	END		
4623						
4624	023074	102010	BFSEQ6:	RWD	;REWIND.	(28)
4625	023076	000001		1		
4626	023100	000001		1		
4627	023102	000000		0		
4628	023104	104005		WRT	;WRITE.	(29)
4629	023106	004000		DATCNT		
4630	023110	000001		1		
4631	023112	000001		1		
4632	023114	105005		WRR	;WRITE RETRY.	(30)
4633	023116	004000		DATCNT		
4634	023120	000001		1		
4635	023122	000001		1		
4636	023124	177777	.WORD	END		
4637						
4638	023126	104401	BFSEQ7:	RDR	;READ REV.	(31)
4639	023130	004000		DATCNT		
4640	023132	000001		1		
4641	023134	000001		1		
4642	023136	105401		RNR	;READ NEXT REV.	(32)
4643	023140	004000		DATCNT		
4644	023142	000001		1		
4645	023144	000001		1		
4646	023146	125401		RNF	;READ NEXT FWD.	(33)



4647	023150	004000		DATCNT		
4648	023152	000001		1		
4649	023154	000001		1		
4650	023156	177777	.WORD	END		
4651						
4652	023160	104001	BFSEQ8:	RDF		
4653	023162	004000		DATCNT		
4654	023164	000001		1		
4655	023166	000001		1		
4656	023170	105001		RPF		
4657	023172	004000		DATCNT		
4658	023174	000001		1		
4659	023176	000001		1		
4660	023200	125001		RPR		
4661	023202	004000		DATCNT		
4662	023204	000001		1		
4663	023206	000001		1		
4664	023210	177777	.WORD	END		
4665						
4666	023212	101012	BFSEQ9: .WORD	CLN		
4667	023214	000001		1		
4668	023216	000001		1		
4669	023220	000000		0		
4670	023222	102010		RWD		
4671	023224	000001		1		
4672	023226	000001		1		
4673	023230	000000		0		
4674	023232	177777	.WORD	END		
4675						
4676	023234	104105	BFSEQ10:	WTV		
4677	023236	000012		12		
4678	023240	000001		1		
4679	023242	000000		0		
4680	023244	104105		WTV		
4681	023246	000011		11		
4682	023250	000001		1		
4683	023252	000000		0		
4684	023254	177777	.WORD	END		
4685			.EVEN			
4686						
4687	023254		ENDTST			
4688	023256		L10^17:			
4689	023256	104401				

TRAP C\$ETST

```

4690          .SBTTL TEST 2: DATA RELIABILITY.
4691
4692          :++
4693          : TEST TO CHECK THE DATA RELIABILITY OF THE TS04.
4694          :--
4695 023260      BGNTST
4696 023260      T2::
4697
4698 023260      LET RANDOM :B= #1          ;SET THE RANDOM OPERATIONS FLAG.
4699 023260 112737 000001 003441          ;CLEAR EXPECT BOT FLAG.      MOVB #1,RANDOM
4700 023266      LET EXPBOT :B= #0          ;SET UP THE RECORD LENGTH MASK,
4701 023266 105037 003440          ;ALLOW MAXIMUM BUFFER.      CLRB EXPBOT
4702 023272      LET R2 := #DATCNT - #1      ;CMD 1 = SET CHARACTERISTIC.
4703 023272 012702 004000          ;IF STARTING THEN:      MOV #DATCNT,R2
4704 023276 005302          ;CLR START FLAG.      DEC R2
4705 023300      LET LENMSK := COMP R2
4706 023300 010237 003356          ;31 OPERATIONS.      MOV R2,LENMSK
4707 023304 005137 003356          ;RANDOM PATTERN.      COM LENMSK
4708 023310 004737 006444          ;REPEAT TO EOT:      JSR PC,SETCH
4709 023314      IFB STAFLG NE #0 THEN      ;FILL SEQ TBL WITH RANDOM CMDS.
4710 023314 105737 003452          ;REPEAT TO EOT:      IFB STAFLG NE #0 THEN
4711 023320 001404          ;REPEAT TO EOT:      JSR PC,SETRW
4712 023322 004737 006470          ;REPEAT TO EOT:      LET STAFLG :B= #0
4713 023326      ENDIF
4714 023326 105037 003452          ;REPEAT TO EOT:      CLRB STAFLG
4715 023332
4716 023332          ;REPEAT TO EOT:      50422$:
4717 023332      LET (R1)+ := #WTV          ;REPEAT TO EOT:      ;CMD3 = WRITE/ VERIFY.
4718 023332 012721 104105          ;REPEAT TO EOT:      MOV #WTV,(R1)+
4719 023336      LET (R1)+ := #DATCNT      ;REPEAT TO EOT:      ;SET BRF TO MAX FOR PATTERN GENERATION.
4720 023336 012721 004000          ;REPEAT TO EOT:      MOV #DATCNT,(R1)+
4721 023342      LET R2 := COMP #RNOPSC
4722 023342 012702 177740          ;REPEAT TO EOT:      MOV #RNOPSC,R2
4723 023346 005102          ;REPEAT TO EOT:      COM R2
4724 023350      LET (R1)+ := R2
4725 023350 010221          ;REPEAT TO EOT:      ;31 OPERATIONS.
4726 023352      LET (R1)+ := #RANP
4727 023352 012721 000007          ;REPEAT TO EOT:      MOV R2,(R1)+
4728 023356      REPEAT
4729 023356          ;REPEAT TO EOT:      MOV #RANP,(R1)+
4730 023356          ;REPEAT TO EOT:      50423$:
4731 023356          ;REPEAT TO EOT:      ;FILL SEQ TBL WITH RANDOM CMDS.
4732 023356 020127 003550          ;REPEAT TO EOT:      50424$:
4733 023362 002012          ;REPEAT TO EOT:      CMP R1,#SEQEND
4734 023364          ;REPEAT TO EOT:      BGE 50425$
4735 023364 063737 003360 003362      LET RANS := RANS + RANB
4736 023372          ;REPEAT TO EOT:      ADD RANB,RANS
4737 023372 013702 003362          ;REPEAT TO EOT:      LET R2 := RANS CLR.BY #177741 ;R2 = RANDOM # (0 - 36).
4738 023376 042702 177741          ;REPEAT TO EOT:      MOV RANS,R2
4739 023402 004772 023540          ;REPEAT TO EOT:      BIC #177741,R2
4740 023406      JSR PC,@RANCMD(R2)          ;SET UP A RANDOM CMD + BR.
4741 023406 000763          ;REPEAT TO EOT:      ENDDO
4742 023410          ;REPEAT TO EOT:      BR 50424$
4743 023410          ;REPEAT TO EOT:      50425$:
4744 023410 012711 177777          ;REPEAT TO EOT:      ;STORE END OF SEQUENCE CODE IN TABLE.
4745 023414 004737 006510          ;REPEAT TO EOT:      MOV #END,(R1)
          ;GO EXECUTE ALL CMDS IN SEQUENCE TABLE.
          JSR PC,EXALL

```



4746	023420			LET R1 := #CMDSEQ		;INIT CMD SEQ TBL POINTER,	
4747	023420	012701	003460			MOV #CMDSEQ,R1	
4748	023424			UNTIL R2 NE #0		;REPEAT UNTIL EOT IS REACHED	
4749	023424	005702				TST R2	
4750	023426	001753				BEQ 50423\$	
4751	023430			LET ALLEOT :B= ALLEOT + #1		;FLAG ALL UNITS @ EOT	
4752	023430	105237	003450			INCB ALLEOT	
4753	023434	000240		NOP			
4754	023436	000240		NOP			
4755	023440	000240		NOP			
4756	023442	004737	024764	JSR PC,T5WEOT		;WRITE ONE RECORD BEYOND EOT ON ALL UNITS	
4757						;SO THAT SHORTER READ STOP DISTANCE	
4758						;SHALL POSITION HEAD IN CLEAN IRG GAP	
4759						;READ REV THAT EXTRA REC TO RE-POSITION THE TAPE	
4760	023446	004737	023600	JSR PC,RANRD		;SET UP READ REV/FWD CMDS,	
4761	023452			LET CMDSEQ+4 := COMP #RNOPSC		;# OF RECORDS FOR READ REV.	
4762	023452	012737	177740			MOV #RNOPSC,CMDSEQ+4	
4763	023460	005137	003464			COM CMDSEQ+4	
4764	023464			LET CMDSEQ+14 := CMDSEQ+4		;# OF RECORDS FOR READ FORWARD.	
4765	023464	013737	003464			MOV CMDSEQ+4,CMDSEQ+	
4766	023472			LET (R1) := #END		;STORE END OF SEQUENCE CODE IN SEQ TABLE.	
4767	023472	012711	177777			MOV #END,(R1)	
4768	023476	004737	006510	JSR PC,EXALL		;GO EXECUTE READ REV/FWD OF LAST N RECORDS.	
4769	023502			LET ALLEOT :B= #0		;CLEAR ALL UNITS @ EOT FLAG	
4770	023502	105037	003450			CLRB ALLEOT	
4771	023506			LET RPTFLG :B= #1		;REQUEST PERFORMANCE REPORT DURING REWIND.	
4772	023506	112737	000001			MOVB #1,RPTFLG	
4773	023514			LET R1 := #CMDSEQ		;INIT SEQ TBL POINTER,	
4774	023514	012701	003460			MOV #CMDSEQ,R1	
4775	023520	004737	006470	JSR PC,SETRW		;STORE REWIND IN SEQ TBL,	
4776	023524			LET (R1) := #END		;STORE END IN SEQ TBL,	
4777	023524	012711	177777			MOV #END,(R1)	
4778	023530	004737	006510	JSR PC,EXALL		;EXECUTE REWIND CMD ON ALL UNITS	
4779							
4780	023534			EXIT TST			
4781	023534	104432				TRAP C\$EXIT	
4782	023536	000174				.WORD L10034-	
4783							

```

4784      ;      ADDRESSES OF SUBROUTINES USED TO SET UP RANDOM OPERATIONS IN
4785      ;      THE DATA RELIABILITY TEST.
4786
4787 023540 023666      RANCMD: RANWV      ;WRITE/VERIFY.
4788 023542 023654      RANWR      ;WRITE.
4789 023544 023654      RANWR      ;WRITE.
4790 023546 023654      RANWR      ;WRITE.
4791 023550 023654      RANWR      ;WRITE.
4792 023552 023654      RANWR      ;WRITE.
4793 023554 023654      RANWR      ;WRITE.
4794 023556 023654      RANWR      ;WRITE.
4795 023560 023600      RANRD      ;READ.
4796 023562 023600      RANRD      ;READ.
4797 023564 023600      RANRD      ;READ.
4798 023566 023600      RANRD      ;READ.
4799 023570 023600      RANRD      ;READ.
4800 023572 023600      RANRD      ;READ.
4801 023574 023600      RANRD      ;READ.
4802 023576 023600      RANRD      ;READ.
4803
4804
4805
4806
4807
4808      ;      SUBROUTINE TO SET UP READ COMMANDS IN SEQUENCE TABLE.
4809      ;      INPUTS:
4810      ;      OUTPUTS:
4811      ;      REGISTERS:      R2
4812      ;      CALLS:
4813
4814 023600      RANRD: LET (R1)+ := #RDR      ;STORE READ REV CMD.
4815 023600 012721 104401      ;MOV      #RDR,(R1)+
4816 023604      LET (R1)+ := #DATCNT      ;SET BRF TO MAX FOR READ RANDOM LENGTHS.
4817 023604 012721 004000      ;MOV      #DATCNT,(R1)+
4818 023610      LET RANB := RANB + RANS
4819 023610 063737 003362 003360      ;ADD      RANS,RANB
4820 023616      LET R2 := RANB CLR.BY #RNOPSC
4821 023616 013702 003360      ;MOV      RANB,R2
4822 023622 042702 177740      ;BIC      #RNOPSC,R2
4823 023626      LET (R1)+ := R2      ;SET RANDOM # OF OPERATIONS.
4824 023626 010221      ;MOV      R2,(R1)+
4825 023630      LET (R1)+ := #RANP      ;RANDOM PATTERN.
4826 023630 012721 000007      ;MOV      #RANP,(R1)+
4827 023634      LET (R1)+ := #RDF      ;STORE READ FWD CMD.
4828 023634 012721 104001      ;MOV      #RDF,(R1)+
4829 023640      LET (R1)+ := #DATCNT      ;SET BRF TO MAX TO READ RANDOM LENGTHS.
4830 023640 012721 004000      ;MOV      #DATCNT,(R1)+
4831 023644      LET (R1)+ := R2      ;SET RANDOM # OF OPERATIONS.
4832 023644 010221      ;MOV      R2,(R1)+
4833 023646      LET (R1)+ := #RANP      ;RANDOM PATTERN.
4834 023646 012721 000007      ;MOV      #RANP,(R1)+
4835 023652 000207      RTS PC
  
```



```

4836      ;      SUBROUTINE TO SET UP A WRITE COMMAND IN THE SEQUENCE TABLE.
4837      ;      INPUTS:
4838      ;      OUTPUTS:
4839      ;      REGISTERS:
4840      ;      CALLS:
4841
4842      023654      RANWR: LET (R1)+ := #WRT      ;STORE WRITE CMD.
4843      023654      012721      104005      MOV      #WRT,(R1)+
4844      023660      004737      023700      JSR PC,RANW      ;STORE BRF, # OF OPERATIONS, PATTERN.
4845      023664      000207      RTS PC
4846
4847
4848
4849
4850
4851      ;      SUBROUTINE TO SET UP A WRITE/VERIFY COMMAND IN THE SEQUENCE TABLE.
4852      ;      INPUTS:
4853      ;      OUTPUTS:
4854      ;      REGISTERS:
4855      ;      CALLS:
4856
4857      023666      RANWV: LET (R1)+ := #WTV      ;STORE WRITE/VERIFY CMD.
4858      023666      012721      104105      MOV      #WTV,(R1)+
4859      023672      004737      023700      JSR PC,RANW      ;STORE BRF, # OF OPERATIONS, PATTERN.
4860      023676      000207      RTS      PC
4861
4862
4863
4864
4865
4866      ;      SUBROUTINE TO STORE BRF, # OF OPERATIONS, PATTERN IN COMMAND
4867      ;      SEQUENCE TABLE FOR WRITE AND WRITE/VERIFY COMMANDS.
4868      ;      INPUTS:
4869      ;      OUTPUTS:
4870      ;      REGISTERS:      R2
4871      ;      CALLS:
4872
4873      023700      RANW: LET (R1)+ := #DATCNT      ;SET BRF TO MAX FOR PATTERN GENERATION.
4874      023700      012721      004000      MOV      #DATCNT,(R1)+
4875      ;RANDOM BRF WILL BE GENERATED FOR EACH RECORD.
4876      023704      LET RANB := RANB + RANS
4877      023704      063737      003362      003360      ADD      RANS,RANB
4878      023712      LET R2 := RANB CLR.BY #RNOPSC
4879      023712      013702      003360      MOV      RANB,R2
4880      023716      042702      177740      BIC      #RNOPSC,R2
4881      023722      LET (R1)+ := R2      ;SET RANDOM # OF OPERATIONS.
4882      023722      010221      MOV      R2,(R1)+
4883      023724      LET (R1)+ := #RANP      ;RANDOM PATTERN.
4884      023724      012721      000007      MOV      #RANP,(R1)+
4885      023730      000207      RTS PC      ;RETURN.
4886
4887      .EVEN
4888
4889      023732      L10034:      ENDTST
4890      023732
4891      023732      104401      TRAP      C$ETST

```

4892



.SBTTL TEST 3: WRITE COMPATABILITY/WRITE UTILITY.

;++  
: TEST TO WRITE RECORDS FROM BOT TO EOT.  
:--

4893  
4894  
4895  
4896  
4897  
4898  
4899 023734  
4900 023734  
4901  
4902 023734  
4903 023734 112737 000001 003441  
4904 023742  
4905 023742 105037 003440  
4906 023746  
4907 023746 012702 004000  
4908 023752 005302  
4909 023754  
4910 023754 010237 003356  
4911 023760 005137 003356  
4912 023764 004737 006444  
4913 023770 004737 006470  
4914 023774  
4915 023774 105037 003452  
4916 024000  
4917 024000  
4918 024000  
4919 024000  
4920 024000 020127 003550  
4921 024004 002003  
4922 024006 004737 023654  
4923 024012  
4924 024012 000772  
4925 024014  
4926 024014  
4927 024014 012711 177777  
4928 024020 004737 006510  
4929 024024  
4930 024024 012701 003460  
4931 024030  
4932 024030 005702  
4933 024032 001762  
4934 024034  
4935 024034 105237 003450  
4936 024040 000240  
4937 024042 000240  
4938 024044 000240  
4939 024046 004737 024764  
4940  
4941  
4942  
4943 024052  
4944 024052 105037 003450  
4945 024056 004737 006470  
4946 024062  
4947 024062 012711 177777  
4948 024066 004737 006510

T3::

BGNTST

LET RANDOM :B= #1

;SET THE RANDOM OPERATIONS FLAG.

MOVB #1,RANDOM

LET EXPBOT :B= #0

;CLEAR EXPECT BOT FLAG.

CLRB EXPBOT

LET R2 := #DATCNT - #1

;SET UP THE RECORD LENGTH MASK.

MOV #DATCNT,R2  
DEC R2

LET LENMSK := COMP R2

;ALLOW MAXIMUM BUFFER.

MOV R2,LENMSK  
COM LENMSK

JSR PC,SETCH

;CMD 1 = SET CHARACTERISTIC.

JSR PC,SETRW

;CMD2=REWIND

LET STAFLG :B= #0

;CLEAR START FLAG

CLRB STAFLG

REPEAT

;REPEAT TO EOT.

WHILE R1 LT #SEQEND DO

50426\$:  
;WHILE THERE IS MORE ROOM IN SEQ TABLE:  
50427\$:

CMP R1,#SEQEND  
BGE 50430\$

JSR PC,RANWR

;STORE A WRITE CMD IN SEQUENCE TABLE.

ENDDO

BR 50427\$

LET (R1) := #END

50430\$:  
;STORE END OF SEQUENCE CODE IN TABLE.

MOV #END,(R1)

JSR PC,EXALL

;EXECUTE ALL CMDS IN SEQ TBL ON UNITS.

LET R1 := #CMDSEQ

;INIT SEQ TBL POINTER,

MOV #CMDSEQ,R1

UNTIL R2 NE #0

;REPEAT UNTIL EOT IS REACHED

TST R2  
BEQ 50426\$

LET ALLEOT :B= ALLEOT + #1

;SET ALL UNITS @ EOT FLAG

INCB ALLEOT

NOP

NOP

NOP

JSR PC,T5WEOT

;WRITE ONE RECORD BEYOND EOT ON ALL UNITS

;SO THAT SHORTER READ STOP DISTANCE

;SHALL POSITION HEAD IN CLEAN IRG GAP

;READ REV THAT EXTRA REC TO RE-POSITION TAPE

;CLEAR ALL UNITS @ EOT FLAG

CLRB ALLEOT

LET ALLEOT :B= #0

JSR PC,SETRW

;STORE REWIND IN SEQ TBL,

LET (R1) := #END

;STORE END IN SEQ TBL,

MOV #END,(R1)

JSR PC,EXALL

;EXECUTE REWIND CMD ON ALL UNITS

4949							
4950	024072		EXIT	TST			
4951	024072	104432				TRAP	C\$EXIT
4952	024074	000002				.WORD	L10035-
4953							
4954			.EVEN				
4955							
4956	024076		ENDTST				
4957	024076		L10035:				
4958	024076	104401				TRAP	C\$ETST
4959							



```

4960
4961
4962
4963
4964
4965
4966
4967 024100
4968 024100
4969
4970 024100
4971 024100 112737 000001 003441
4972 024106
4973 024106 112737 000001 003440
4974 024114 004737 006444
4975 024120 004737 006470
4976 024124
4977 024124 105037 003452
4978 024130
4979 024130 012721 104001
4980 024134
4981 024134 012721 004000
4982 024140
4983 024140 012721 077777
4984 024144
4985 024144 012721 000007
4986 024150
4987 024150 012711 177777
4988 024154 004737 006510
4989 024160
4990 024160 105237 003450
4991 024164
4992 024164 012701 003460
4993 024170
4994 024170 012721 104401
4995 024174
4996 024174 012721 004000
4997 024200
4998 024200 012721 077777
4999 024204
5000 024204 012721 000007
5001 024210
5002 024210 012711 177777
5003 024214 004737 006510
5004 024220
5005 024220 105037 003450
5006
5007 024224
5008 024224 104432
5009 024226 000002
5010
5011
5012
5013 024230
5014 024230
5015 024230 104401

.SBTTL TEST 4: READ COMPATABILITY/READ UTILITY.
:++
: TEST TO READ ENTIRE TAPE FORWARD AND REVERSE.
:--

T4::
      BGNTST
      LET RANDOM :B= #1           ;SET THE RANDOM OPERATIONS FLAG.
      LET EXPBOT :B= #1           ;SET EXPECT BOT FLAG.
      JSR PC,SETCH                 ;CMD 1 = SET CHARACTERISTIC.
      JSR PC,SETRW                 ;CMD2=REWIND.
      LET STAF LG :B= #0          ;CLEAR START FLAG
      LET (R1)+ := #RDF           ;CMD3 = READ FORWARD.
      LET (R1)+ := #DATCNT        ;SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
      LET (R1)+ := #77777        ;SET RECORD COUNT TO MAX FOR WHOLE TAPE.
      LET (R1)+ := #RANP         ;PATTERN = RANDOM.
      LET (R1) := #END            ;STORE END OF SEQUENCE CODE IN TABLE.
      JSR PC,EXALL                 ;EXECUTE ALL CMDS IN SEQ TBL ON ALL UNITS.
      LET ALLEOT :B= ALLEOT + #1 ;FLAG TO ALLOW ALL UNITS AT EOT TO READ REV
      LET R1 := #CMDSEQ           ;INIT CMD SEQ TBL POINTER.
      LET (R1)+ := #RDR           ;CMD1 = READ REVERSE.
      LET (R1)+ := #DATCNT        ;SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
      LET (R1)+ := #77777        ;RECORD COUNT = MAX FOR WHOLE TAPE.
      LET (R1)+ := #RANP         ;PATTERN = RANDOM.
      LET (R1) := #END            ;STORE END OF SEQUENCE CODE IN TABLE.
      JSR PC,EXALL                 ;GO EXECUTE READ REV. OF ENTIRE TAPE.
      LET ALLEOT :B= #0          ;CLEAR ALL UNITS @ EOT FLAG
      EXIT TST
      TRAP C$EXIT
      .WORD L10036-.

      .EVEN
      L10036:
      ENDTST
      TRAP C$ETST
  
```



TEST 5: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

```

5016 .SBTTL TEST 5: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
5017
5018 :++
5019 : TEST TO EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
5020 :--
5021
5022 024232          BGNTST
5023 024232          T5::
5024
5025 024232          LET RANDOM :B= #0          ;CLEAR RANDOM MODE FLAG.
5026 024232 105037 003441          ;CLRB RANDOM
5027 024236          LET EXPBOT :B= #1          ;SET EXPECT BOT FLAG.
5028 024236 112737 000001 003440          ;MOVE INHIBIT RFC ERROR REPORT FLAG.
5029 024244          LET IRE :B= PIRE          ;MOVE PIRE,IRE
5030 024244 113737 002214 003445          ;CMD 1 = SET CHARACTERISTIC.
5031 024252 004737 006444          ;MOVE CHAR CODE FROM P TBL TO SEQ TBL.
5032 024256          JSR PC,SETCH          ;MOV CHAR,CMDSEQ+2
5033 024256 013737 002216 003462          ;R2 POINTS TO CMD2 IN SOFT P TABLE.
5034 024264          LET R2 := #CMDD          ;MOV #CMDD,R2
5035 024264 012702 002220          ;MOVE CMD 2 FROM P TBL TO SEQ TBL.
5036 024270 004737 024742          JSR PC,PTCMDS          ;MOVE CMD 3 FROM P TBL TO SEQ TBL.
5037 024274 004737 024742          JSR PC,PTCMDS          ;MOVE CMD 4 FROM P TBL TO SEQ TBL.
5038 024300 004737 024742          JSR PC,PTCMDS          ;MOVE CMD 5 FROM P TBL TO SEQ TBL.
5039 024304 004737 024742          JSR PC,PTCMDS          ;MOVE CMD 6 FROM P TBL TO SEQ TBL.
5040 024310 004737 024742          JSR PC,PTCMDS          ;MOVE CMD 7 FROM P TBL TO SEQ TBL.
5041 024314 004737 024742          JSR PC,PTCMDS          ;MOVE END CMD FROM P TBL TO SEQ TBL.
5042 024320 004737 024742          JSR PC,PTCMDS          ;CLEAR JMP CMD LOOP COUNT.
5043 024324          LET JLOOP := #0          ;CLR JLOOP
5044 024324 005037 003370          ;CLEAR START FLAG
5045 024330          LET STAFLG :B= #0          ;CLRB STAFLG
5046 024330 105037 003452          ;INIT SEQUENCE TABLE POINTER.
5047 024334          LET R1 := #CMDSEQ          ;MOV #CMDSEQ,R1
5048 024334 012701 003460          ;WHILE THERE ARE CMDS LEFT IN SEQUENCE TBL:
5049 024340          3$: WHILE (R1) NE #END DO          50431$:
5050 024340          ;CMP (R1),#END
5051 024340 021127 177777          ;BEQ 50432$
5052 024344 001574          ;IS THIS A JUMP CMD?
5053 024346 022711 000040          ;BR IF NOT.
5054 024352 001024          ;POINT TO BRF.
5055 024354          ;ADD #2,R1
5056 024354 062701 000002          ;SAVE BR (LOCATION).
5057 024360 012137 003372          ;MOV (R1)+,JLOC          ;HAS LOOP COUNT BE SATISFIED?
5058 024364 022137 003370          ;CMP (R1)+,JLOOP          ;IF NOT, JMP AGAIN.
5059 024370 001003          ;BNE 1$          ;IF SO, ADJUST SEQ POINTER
5060 024372          LET R1 := R1 + #2          ;ADD #2,R1
5061 024372 062701 000002          ;AND GO TO NEXT COMMAND.
5062 024376 000760          1$: LET JLOOP := JLOOP + #1          ;UPDATE THE LOOP COUNT.
5063 024400          ;INC JLOOP
5064 024400 005237 003370          ;LET R1 := #CMDSEQ          ;INIT CMD SEQ TABLE POINTER.
5065 024404          ;MOV #CMDSEQ,R1
5066 024404 012701 003460          2$: DEC JLOC          ;DECR LOCATION COUNTER.
5067 024410 005337 003372          ;BEQ 3$          ;IF THIS IS THE RIGHT LOCATION TO JMP TO, GO SET
5068 024414 001751          ;LET R1 := R1 + #10          ;IF NOT, UPDATE SEQ POINTER TO NEXT CMD.
5069 024416          ;ADD #10,R1
5070 024416 062701 000010          ;BR 2$          ;DO IT AGAIN.
5071 024422 000772

```



HARDWARE TESTS MACY11 30(1046)  
CZTSHC.P11 11-OCT-79 13:59

11-OCT-79 14:02 PAGE 120

TEST 5: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

```

5072 024424 022711 000020      6$:      CMP  #DLY.C,(R1)          ;DELAY?
5073 024430 001026              BNE  4$                ;BR IF NOT.
5074 024432                    LET R1 := R1 + #4      ;R1 = LOCATION OF N COUNT.
5075 024432 062701 000004              ADD  #4,R1
5076 024436                    LET TIME2 := (R1)        ;SAVE N COUNT.
5077 024436 011137 003366              MOV  (R1),TIME2
5078 024442                    7$:      DELAY 1                ;GO TO SUPER-WAIT 1 MSEC.
5079 024442 012727 000001              MOV  #1,(PC)+
5080 024446 000000              .WORD 0
5081 024450 013727 002116              MOV  LSDLY,(PC)+
5082 024454 000000              .WORD 0
5083 024456 005367 177772              DEC  -6(PC)
5084 024462 001375              BNE  -4
5085 024464 005367 177756              DEC  -22(PC)
5086 024470 001367              BNE  -20
5087 024472 005337 003366              DEC  TIME2
5088 024476 001361              BNE  7$
5089 024500                    LET R1 := R1 + #4      ;POINT TO NEXT CMD.
5090 024500 062701 000004              ADD  #4,R1
5091 024504 000715              BR   3$                ;GO CHECK NEXT CMD.
5092 024506 004737 007452      4$:      JSR  PC,SETUP          ;GO SETUP THE COMMAND BLOCK.
5093 024512                    WHILE NCNT LT NCNT1 DO ;WHILE THERE ARE RECORDS REMAINING:
5094 024512                    50433$:
5095 024512 023737 003340 003342              CMP  NCNT,NCNT1
5096 024520 002103              BGE  50434$
5097 024522 004737 007344              JSR  PC,CMDAC          ;STORE CMD ASCII IN ERROR MSG.
5098 024526 004737 007004              JSR  PC,EXSUB         ;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATUS.
5099 024532                    IF CMDWRD EQ #GES THEN ;IF CMD IS GET STATUS THEN:
5100 024532 023727 003346 100017              CMP  CMDWRD,#GES
5101 024540 001002              BNE  50435$
5102 024542 004737 015642              JSR  PC,PRXST         ;PRINT EXTENDED STATUS REGISTERS.
5103 024546                    ENDIF
5104 024546                    50435$:
5105 024546 004737 015724              JSR  PC,CKHAE         ;CHECK HALT AFTER EACH CMD FLAG.
5106 024552                    LET R2 := #1                ;SET ALL UNITS AT BOT/EOT.
5107 024552 012702 000001              MOV  #1,R2
5108 024556 004737 015332              JSR  PC,FIRSTU        ;FIND FIRST UNIT.
5109 024562                    WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE UNITS:
5110 024562                    50436$:
5111 024562 026527 002532 177777              CMP  DEVTBL(R5),#END
5112 024570 001426              BEQ  50437$
5113 024572                    IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
5114 024572 032737 000400 003346              BIT  #MOD.CO,CMDWRD
5115 024600 001406              BEQ  50440$
5116 024600                    IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT THEN:
5117 024602 032765 00 2 003426              BIT  #X0.BOT,EOTFLG(R)
5118 024610 001001              BNE  50441$
5119 024612                    LET R2 := #0                ;CLEAR EOT/BOT FLAG.
5120 024612 005002              CLR  R2
5121 024614                    ENDIF
5122 024614                    50441$:
5123 024614                    ELSE                ;ELSE IF CMD IS NOT REVERSE:
5124 024614 000411              BR   50442$
5125 024616                    50440$:
5126 024616                    IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
5127 024616 032765 000001 003426              BIT  #X0.EOT,EOTFLG(R)

```





```

5184
5185
5186      :      SUBROUTINE TO MOVE A COMMAND FROM THE SOFTWARE P TABLE TO
5187      :      THE COMMAND SEQUENCE TABLE.
5188      :      INPUTS:          R2 = POINTER TO SOFT 'P' TABLE
5189      :      OUTPUTS:
5190      :      REGISTERS:      R3.
5191      :      CALLS:
5192
5193      PTCMDS: LET R3 := (R2)+ - #1 SHIFT +1 ;R3 = COMMAND TABLE INDEX.
5194      024742 012203      MOV      (R2)+,R3
5195      024744 005303      DEC      R3
5196      024746 006303      ASL      R3
5197      024750      LET (R1)+ := CMDTBL(R3) ;MOVE COMMAND WORD.
5198      024750 016321 003562      MOV      CMDTBL(R3), (R1)+
5199      024754      LET (R1)+ := (R2)+ ;MOVE # OF BYTES.
5200      024754 012221      MOV      (R2)+, (R1)+
5201      024756      LET (R1)+ := (R2)+ ;MOVE # OF OPERATIONS.
5202      024756 012221      MOV      (R2)+, (R1)+
5203      024760      LET (R1)+ := (R2)+ ;MOVE PATTERN CODE.
5204      024760 012221      MOV      (R2)+, (R1)+
5205      024762 000207      RTS PC
5206
5207      :      SUBROUTINE TO WRITE THEN READ REVERSE ONE RECORD BEYOND EOT
5208      :      INPUTS:
5209      :      OUTPUTS:
5210      :      REGISTERS:
5211      :      CALLS:          CMDAC,EXSUB,CKHAE
5212
5213      T5WEOT: NOP
5214      024764 000240      NOP
5215      024770 004737 007004      JSR PC,EXSUB ;WRITE ONE RECORD BEYOND EOT
5216      024774 004737 015724      JSR PC,CKHAE ;SO THAT READ SHORTER STOP DISTANCE
5217      : ;SHALL POSITION HEAD IN CLEAN IRG GAP
5218      025000      LET PCMDWD := CMDWRD ;REPOSITION TAPE
5219      025000 013737 003346 003352      MOV      CMDWRD,PCMDWD
5220      025006      LET CMDWRD := #RDR ;BEFORE EXTRA RECORD
5221      025006 012737 104401 003346      MOV      #RDR,CMDWRD
5222      025014      LET CMDLG := #4 ;BY READING REVERSE
5223      025014 012737 000004 003354      MOV      #4,CMDLG
5224      025022      LET CNDPKT := CMDWRD CLR.BY #BRF.C
5225      025022 013737 003346 002310      MOV      CMDWRD,CNDPKT
5226      025030 042737 004000 002310      BIC      #BRF.C,CNDPKT
5227      025036      LET CMDSAV := CNDPKT ;THAT RECORD TO ALLOW
5228      025036 013737 002310 003350      MOV      CNDPKT,CMDSAV
5229      025044      LET CNDPKT+CP.ADL := DATARD ;NEXT COMMAND IN THE
5230      025044 013737 003336 002312      MOV      DATARD,CNDPKT+CP
5231      025052 004737 007344      JSR PC,CMDAC ;TABLE TO BE EXECUTED
5232      025056 004737 007004      JSR PC,EXSUB
5233      025062 004737 015724      JSR PC,CKHAE
5234      025066 000207      RTS PC
5235
5236
5237      .EVEN
5238
5239      025070      ENDTST

```

HARDWARE TESTS MACY11 30(1046) 11-OCT-79 14:02 PAGE 123  
CZTSHC.P11 11-OCT-79 13:59 TEST 5: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

5240 025070  
5241 025070 104401  
5242  
5243 025072

L10037:

TRAP CSETST

ENDMOD



```

5244      .TITLE PARAMETER CODING
5245
5246      .SBTTL  HARDWARE PARAMETER CODING SECTION
5247
5248 025072      BGNMOD
5249
5250
5251      ;++
5252      ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
5253      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
5254      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
5255      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
5256      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
5257      ; WITH THE OPERATOR.
5258      ;--
5259      BGNHRD
5260 025072 000024      L$HARD::      .WORD L10040-L$HARD/2
5261 025074
5262
5263 025074      GPRMA  TS4ADR,0,0,160002,177564,YES
5264 025074 000031      .WORD  T$CODE
5265 025076 025120      .WORD  TS4ADR
5266 025100 160002      .WORD  T$LLOLIM
5267 025102 177564      .WORD  T$HILIM
5268 025104      GPRMD  TS4VCT,2,0,777,60,776,YES
5269 025104 001032      .WORD  T$CODE
5270 025106 025135      .WORD  TS4VCT
5271 025110 000777      .WORD  777
5272 025112 000060      .WORD  T$LLOLIM
5273 025114 000776      .WORD  T$HILIM
5274
5275 025116      EXIT HRD
5276 025116 013004      .WORD  T$CODE
5277
5278      .NLIST  BEX
5279      025120 051524 051123 040440 TS4ADR: .ASCIZ /TSSR ADDRESS/
5280      025135      126 041505 047524 TS4VCT: .ASCIZ /VECTOR/
5281      .LIST  BEX
5282      .EVEN
5283 025144      ENDHRD
                    L10040:      .EVEN

```

5284  
 5285  
 5286  
 5287  
 5288  
 5289  
 5290  
 5291  
 5292  
 5293  
 5294  
 5295 025144  
 5296 025144 000501  
 5297 025146  
 5298  
 5299  
 5300 025146  
 5301 025146 000130  
 5302 025150 025710  
 5303 025152 000001  
 5304 025154  
 5305 025154 000130  
 5306 025156 025727  
 5307 025160 000400  
 5308 025162  
 5309 025162 001130  
 5310 025164 026032  
 5311 025166 000400  
 5312 025170  
 5313 025170 001130  
 5314 025172 025756  
 5315 025174 000001  
 5316 025176  
 5317 025176 003130  
 5318 025200 026106  
 5319 025202 000400  
 5320 025204  
 5321 025204 004024  
 5322 025206  
 5323 025206 002130  
 5324 025210 026002  
 5325 025212 000001  
 5326 025214  
 5327 025214 003130  
 5328 025216 026063  
 5329 025220 000001  
 5330 025222  
 5331 025222 005130  
 5332 025224 026153  
 5333 025226 000001  
 5334 025230  
 5335 025230 004130  
 5336 025232 026127  
 5337 025234 000001  
 5338 025236  
 5339 025236 127044

```
.SBTTL SOFTWARE PARAMETER CODING SECTION
:++
: THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
: THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES THE
: MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
: INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
: MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
: WITH THE OPERATOR.
:--

          BCNSFT
L$SOFT::                                     .WORD L10041-L$SOFT/2

          GPRML  CLRM,0,1,YES                .WORD  T$CODE
                                           .WORD  CLRM
                                           .WORD  1
          GPRML  RRVM,0,400,YES              .WORD  T$CODE
                                           .WORD  RRVM
                                           .WORD  400
          GPRML  RCVERM,2,400,YES           .WORD  T$CODE
                                           .WORD  RCVERM
                                           .WORD  400
          GPRML  HAEM,2,1,YES               .WORD  T$CODE
                                           .WORD  HAEM
                                           .WORD  1
          GPRML  IRECM,6,400,YES           .WORD  T$CODE
                                           .WORD  IRECM
                                           .WORD  400
          XFERT  NEXTSP
          GPRML  BADTM,4,1,YES              .WORD  T$CODE
                                           .WORD  BADTM
                                           .WORD  1
NEXTSP: GPRML  DINTM,6,1,YES               .WORD  T$CODE
                                           .WORD  DINTM
                                           .WORD  1
          GPRML  IREM,12,1,YES             .WORD  T$CODE
                                           .WORD  IREM
                                           .WORD  1
          GPRML  CHGM,10,1,YES             .WORD  T$CODE
                                           .WORD  CHGM
                                           .WORD  1
          XFERF  ENDSP1                     .WORD  T$CODE
```



5340	025240		GPRMD	CHARM,14,0,377,0,777,YES		
5341	025240	006032			.WORD	T\$CODE
5342	025242	026204			.WORD	CHARM
5343	025244	000377			.WORD	377
5344	025246	000000			.WORD	T\$LOLIM
5345	025250	000777			.WORD	T\$HILIM
5346	025252		GPRMD	CMD2M,16,D,37,1,33,YES		
5347	025252	007052			.WORD	T\$CODE
5348	025254	026231			.WORD	CMD2M
5349	025256	000037			.WORD	37
5350	025260	000001			.WORD	T\$LOLIM
5351	025262	000033			.WORD	T\$HILIM
5352	025264		GPRMD	BPCRM,20,D,-1,1,DATCNT,YES		
5353	025264	010052			.WORD	T\$CODE
5354	025266	026237			.WORD	BPCRM
5355	025270	177777			.WORD	-1
5356	025272	000001			.WORD	T\$LOLIM
5357	025274	004000			.WORD	T\$HILIM
5358	025276		GPRMD	NUMBM,22,D,-1,1,77777,YES		
5359	025276	011052			.WORD	T\$CODE
5360	025300	026251			.WORD	NUMBM
5361	025302	177777			.WORD	-1
5362	025304	000001			.WORD	T\$LOLIM
5363	025306	077777			.WORD	T\$HILIM
5364	025310		GPRMD	PATTM,24,D,17,0,10,YES		
5365	025310	012052			.WORD	T\$CODE
5366	025312	026271			.WORD	PATTM
5367	025314	000017			.WORD	17
5368	025316	000000			.WORD	T\$LOLIM
5369	025320	000010			.WORD	T\$HILIM
5370	025322		GPRMD	CMD3M,26,D,37,1,33,YES		
5371	025322	013052			.WORD	T\$CODE
5372	025324	026304			.WORD	CMD3M
5373	025326	000037			.WORD	37
5374	025330	000001			.WORD	T\$LOLIM
5375	025332	000033			.WORD	T\$HILIM
5376	025334		GPRMD	BPCRM,30,D,-1,1,DATCNT,YES		
5377	025334	014052			.WORD	T\$CODE
5378	025336	026237			.WORD	BPCRM
5379	025340	177777			.WORD	-1
5380	025342	000001			.WORD	T\$LOLIM
5381	025344	004000			.WORD	T\$HILIM
5382	025346		GPRMD	NUMBM,32,D,-1,1,77777,YES		
5383	025346	015052			.WORD	T\$CODE
5384	025350	026251			.WORD	NUMBM
5385	025352	177777			.WORD	-1
5386	025354	000001			.WORD	T\$LOLIM
5387	025356	077777			.WORD	T\$HILIM
5388	025360		GPRMD	PATTM,34,D,17,0,10,YES		
5389	025360	016052			.WORD	T\$CODE
5390	025362	026271			.WORD	PATTM
5391	025364	000017			.WORD	17
5392	025366	000000			.WORD	T\$LOLIM
5393	025370	000010			.WORD	T\$HILIM
5394	025372		GPRMD	CMD4M,36,D,37,1,33,YES		
5395	025372	017052			.WORD	T\$CODE

5396	025374	026312			.WORD	CMD4M
5397	025376	000037			.WORD	37
5398	025400	000001			.WORD	T\$LOLIM
5399	025402	000033			.WORD	T\$HILIM
5400	025404		GPRMD	BPCRM,40,D,-1,1,DATCNT,YES		
5401	025404	020052			.WORD	T\$CODE
5402	025406	026237			.WORD	BPCRM
5403	025410	177777			.WORD	-1
5404	025412	000001			.WORD	T\$LOLIM
5405	025414	004000			.WORD	T\$HILIM
5406	025416		GPRMD	NUMBM,42,D,-1,1,77777,YES		
5407	025416	021052			.WORD	T\$CODE
5408	025420	026251			.WORD	NUMBM
5409	025422	177777			.WORD	-1
5410	025424	000001			.WORD	T\$LOLIM
5411	025426	077777			.WORD	T\$HILIM
5412	025430		GPRMD	PATTM,44,D,17,0,10,YES		
5413	025430	022052			.WORD	T\$CODE
5414	025432	026271			.WORD	PATTM
5415	025434	000017			.WORD	17
5416	025436	000000			.WORD	T\$LOLIM
5417	025440	000010			.WORD	T\$HILIM
5418	025442		GPRMD	CMD5M,46,D,37,1,33,YES		
5419	025442	023052			.WORD	T\$CODE
5420	025444	026320			.WORD	CMD5M
5421	025446	000037			.WORD	37
5422	025450	000001			.WORD	T\$LOLIM
5423	025452	000033			.WORD	T\$HILIM
5424	025454		GPRMD	BPCRM,50,D,-1,1,DATCNT,YES		
5425	025454	024052			.WORD	T\$CODE
5426	025456	026237			.WORD	BPCRM
5427	025460	177777			.WORD	-1
5428	025462	000001			.WORD	T\$LOLIM
5429	025464	004000			.WORD	T\$HILIM
5430	025466		GPRMD	NUMBM,52,D,-1,1,77777,YES		
5431	025466	025052			.WORD	T\$CODE
5432	025470	026251			.WORD	NUMBM
5433	025472	177777			.WORD	-1
5434	025474	000001			.WORD	T\$LOLIM
5435	025476	077777			.WORD	T\$HILIM
5436	025500		GPRMD	PATTM,54,D,17,0,10,YES		
5437	025500	026052			.WORD	T\$CODE
5438	025502	026271			.WORD	PATTM
5439	025504	000017			.WORD	17
5440	025506	000000			.WORD	T\$LOLIM
5441	025510	000010			.WORD	T\$HILIM
5442	025512		XFER	ENDSP2		
5443	025512	002004			.WORD	T\$CODE
5444	025514		ENDSP1: XFER	ENDSP		
5445	025514	075004			.WORD	T\$CODE
5446	025516		ENDSP2: GPRMD	CMD6M,56,D,37,1,33,YES		
5447	025516	027052			.WORD	T\$CODE
5448	025520	026326			.WORD	CMD6M
5449	025522	000037			.WORD	37
5450	025524	000001			.WORD	T\$LOLIM
5451	025526	000033			.WORD	T\$HILIM



5452	025530		GPRMD	BPCRM,60,D,-1,1,DATCNT,YES			
5453	025530	030052			.WORD	T\$CODE	
5454	025532	026237			.WORD	BPCRM	
5455	025534	177777			.WORD	-1	
5456	025536	000001			.WORD	T\$LOLIM	
5457	025540	004000			.WORD	T\$HILIM	
5458	025542		GPRMD	NUMBM,62,D,-1,1,77777,YES			
5459	025542	031052			.WORD	T\$CODE	
5460	025544	026251			.WORD	NUMBM	
5461	025546	177777			.WORD	-1	
5462	025550	000001			.WORD	T\$LOLIM	
5463	025552	077777			.WORD	T\$HILIM	
5464	025554		GPRMD	PATTM,64,D,17,0,10,YES			
5465	025554	032052			.WORD	T\$CODE	
5466	025556	026271			.WORD	PATTM	
5467	025560	000017			.WORD	17	
5468	025562	000000			.WORD	T\$LOLIM	
5469	025564	000010			.WORD	T\$HILIM	
5470	025566		GPRMD	CMD7M,66,D,37,1,33,YES			
5471	025566	033052			.WORD	T\$CODE	
5472	025570	026334			.WORD	CMD7M	
5473	025572	000037			.WORD	37	
5474	025574	000001			.WORD	T\$LOLIM	
5475	025576	000033			.WORD	T\$HILIM	
5476	025600		GPRMD	BPCRM,70,D,-1,1,DATCNT,YES			
5477	025600	034052			.WORD	T\$CODE	
5478	025602	026237			.WORD	BPCRM	
5479	025604	177777			.WORD	-1	
5480	025606	000001			.WORD	T\$LOLIM	
5481	025610	004000			.WORD	T\$HILIM	
5482	025612		GPRMD	NUMBM,72,D,-1,1,77777,YES			
5483	025612	035052			.WORD	T\$CODE	
5484	025614	026251			.WORD	NUMBM	
5485	025616	177777			.WORD	-1	
5486	025620	000001			.WORD	T\$LOLIM	
5487	025622	077777			.WORD	T\$HILIM	
5488	025624		GPRMD	PATTM,74,D,17,0,10,YES			
5489	025624	036052			.WORD	T\$CODE	
5490	025626	026271			.WORD	PATTM	
5491	025630	000017			.WORD	17	
5492	025632	000000			.WORD	T\$LOLIM	
5493	025634	000010			.WORD	T\$HILIM	
5494	025636		GPRMD	CMD8M,76,D,37,1,33,YES			
5495	025636	037052			.WORD	T\$CODE	
5496	025640	026342			.WORD	CMD8M	
5497	025642	000037			.WORD	37	
5498	025644	000001			.WORD	T\$LOLIM	
5499	025646	000033			.WORD	T\$HILIM	
5500	025650		GPRMD	BPCRM,100,D,-1,1,DATCNT,YES			
5501	025650	040052			.WORD	T\$CODE	
5502	025652	026237			.WORD	BPCRM	
5503	025654	177777			.WORD	-1	
5504	025656	000001			.WORD	T\$LOLIM	
5505	025660	004000			.WORD	T\$HILIM	
5506	025662		GPRMD	NUMBM,102,D,-1,1,77777,YES			
5507	025662	041052			.WORD	T\$CODE	

5508 025664 026251  
5509 025666 177777  
5510 025670 000001  
5511 025672 077777  
5512 025674  
5513 025674 042052  
5514 025676 026271  
5515 025700 000017  
5516 025702 000000  
5517 025704 000010  
5518 025706  
5519 025706  
5520 025706 176004

GPRMD PATTM,104,D,17,0,10,YES

ENDSP: XFER JMPMSG

.WORD NUMBM  
.WORD -1  
.WORD T\$LOLIM  
.WORD T\$HILIM  
  
.WORD T\$CODE  
.WORD PATTM  
.WORD 17  
.WORD T\$LOLIM  
.WORD T\$HILIM  
  
.WORD T\$CODE



5521  
5522

```

025710 046103 040505 020122 CLRM: .NLIST BEX
025727 122 051505 052105 .ASCIZ /CLEAR COUNTERS/
025756 040510 052114 040440 RRVN: .ASCIZ /RESET RANDOM VARIABLES/
026002 040502 020104 040524 HAEM: .ASCIZ /HALT AFTER EACH CMD/
026032 051120 047111 020124 BADTM: .ASCIZ /BAD TAPE SPOT DETECTION/
026063 104 051511 041101 RCVERM: .ASCIZ /PRINT RECOVERABLE ERRORS/
026106 047111 044510 044502 DINTM: .ASCIZ /DISABLE INTERRUPTS/
026127 103 040510 043516 IRECM: .ASCIZ /INHIBIT RECOVERY/
026153 111 044116 041111 CHGM: .ASCIZ /CHANGE CMD SEQUENCE/
026204 044103 051101 041501 IREM: .ASCIZ /INHIBIT RFC ERROR REPORT/
026231 103 042115 031057 CHARM: .ASCIZ /CHARACTERISTICS CODE/
026237 102 043122 041440 CMD2M: .ASCIZ 'CMD/2'
026251 043 047440 020106 BPCRM: .ASCIZ /BRF COUNT/
026271 120 052101 042524 NUMBM: .ASCIZ /# OF OPERATIONS/
          PATTM: .ASCIZ /PATTERN/
  
```

5523  
5524  
5525  
5526  
5527  
5528  
5529

```

          026302 .LIST BEX
          .EVEN
          JMPMSG:
          EXIT SFT
          .WORD T$CODE
  
```

```

026304 046503 027504 000063 CMD3M: .NLIST BEX
026312 046503 027504 000064 .ASCIZ 'CMD/3'
026320 046503 027504 000065 CMD4M: .ASCIZ 'CMD/4'
026326 046503 027504 000066 .ASCIZ 'CMD/5'
026334 046503 027504 000067 .ASCIZ 'CMD/6'
026342 046503 027504 000070 .ASCIZ 'CMD/7'
          .ASCIZ 'CMD/8'
          .LIST BEX
          .EVEN
  
```

5530  
5531  
5532  
5533  
5534  
5535  
5536  
5537  
5538  
5539  
5540  
5541  
5542  
5543  
5544  
5545  
5546  
5547  
5548  
5549

```

          026350 ENDSFT
          .EVEN
          026350 L10041:
          ;*****
          ;*****
          ; PATCH AREA
          026350 000100 PATCH:: .BLKW 64.
          ;*****
          ;*****
          026550 LASTAD
          .EVEN
          026550 026572 .WORD T$FREE
          026552 000007 .WORD T$SIZE
          L$LAST::
          ENDMOD
  
```

PARAMETER CODING  
CZTSHC.P11

11-OCT-79 13:59

MACY11 30(1046)

11-OCT-79 14:02 PAGE 131  
HARD CODED P-TBL

N 13

5550  
5551  
5552  
5553  
5554  
5555  
5556 026554  
5557 026554  
5558 026554 000000  
5559 026556 000000  
5560 026560 000000  
5561 026562 000000  
5562 026564 000002  
5563 026566  
5564 026566 172522  
5565 026570 000224  
5566 026572  
5567 026572  
5568 026572  
5569  
5570 000001

.SBTTL HARD CODED P-TBL  
:++  
:DIAG IS PRE-PARAMETERIZED PER TBL  
:--  
BGNSETUP 1  
BGNPTAB  
L10042:  
172522  
224  
ENDPTAB  
L10044:  
ENDSETUP  
.END

.WORD 0  
.WORD 0  
.WORD 0  
.WORD 0  
.WORD L10044-./2-1



ACK.C = 100000 G	BRFCNT 003344 G	CNTEND= 003324	CSPNTX= 000015	ERRREC 003415 G
ADR = 000020 G	BRF.C = 004000 G	CNTLEN= 000550 G	CSQIO = 000377	ERS = 100411 G
ALLEOT 003450 G	BTADDR 002544 G	CODELM 003772 G	CSRDBU= 000007	ERSFLG 003451 G
ASSEMB= 000010	BTMSG1 013342	COUNTE= 050364	CSREFG= 000047	EVL = 000004 G
ATTNM 004335 G	BTMSG2 013427	CP.ADH= 000004 G	CSRESE= 000033	EXALL 006510 G
AUDRPM 004645 G	BTMSG3 013477	CP.ADL= 000002 G	CSREVI= 000003	EXARTN 007002
AUTODM 021330	BTPT 003436 G	CP.CMD= 000000 G	CSRFLA= 000021	EXCRTN 010634
BADTM 026002	BTRPT 016410	CP.CNT= 000006 G	CSRPT = 000025	EXCUTE 010326 G
BADTSW 002206 G	BT0 002774 G	CRLF 005213 G	CSSEFG= 000046	EXPBOT 003440 G
BFSEQ 022504	BT1 003046 G	CRLFSP 005216 G	CS\$PRI= 000041	EXSUB 007004 G
BFSEQ0 022530	BT2 003120 G	CTCC 003376 G	CS\$VEC= 000037	ESEND = 002100
BFSEQ1 022602	BT3 003172 G	CVC.C = 040000 G	C\$TPRI= 000013	ESLOAD= 000035
BFSEQ2 022614	CHAR 002216 G	CSAU = 000052	DATARD 003336 G	FATSM 004373 G
BFSEQ3 022706	CHARM 026204	CSAUTO= 000061	DATAWT 003334 G	FIRSTU 015332 G
BFSEQ4 022760	CHGFLG 002212 G	CSBRK = 000022	DATCNT= 004000 G	FMT.CO= 000040 G
BFSEQ5 023022	CHGM 026127	CSBSEG= 000004	DEVTBL 002532 G	FMT.C1= 000100 G
BFSEQ6 023074	CHKERR 011440 G	CSBSLS= 000002	DFPTBL 002174 G	FTLCNT 003314 G
BFSEQ7 023126	CH.EAI= 000040 G	CSCEFG= 000045	DFTSCH= 000040 G	FUNRM 004353 G
BFSEQ8 023160	CH.ERI= 000020 G	CSCLCK= 000062	DIA = 100006 G	FSAU = 000015
BFSEQ9 023212	CH.ESS= 000200 G	CSCLEA= 000012	DIABLK= 003334 G	FSAUTO= 000020
BFSE10 023234	CKDATA 014716 G	CS\$CLOS= 000035	DIACNT= 000020 G	F\$BGN = 000040
BGNFLG= 003404	CKDCNT 015326	CSCLP1= 000006	DIAGMC= 000000	F\$CLEA= 000007
BINC 014302	CKDFF 015330	CS\$VEC= 000036	DINT 002210 G	F\$DU = 000016
BIT0 = 000001 G	CKHAE 015724 G	CS\$DCLN= 000044	DINTM 026063	F\$END = 000041
BIT00 = 000001 G	CKHRTN 016012	CS\$DODU= 000051	DLY = 000020 G	F\$HARD= 000004
BIT01 = 000002 G	CLN = 101012 G	CS\$DRPT= 000024	DLY.C = 000020 G	F\$HW = 000013
BIT02 = 000004 G	CLRERR 011136 G	CS\$DU = 000053	DRI = 100013 G	F\$INIT= 000006
BIT03 = 000010 G	CLRFLG 002202 G	CS\$EDIT= 000002	DROPDN 004616 G	F\$JMP = 000050
BIT04 = 000020 G	CLRM 025710	CS\$ERDF= 000055	DROPED 003446 G	F\$MOD = 000000
BIT05 = 000040 G	CMDAC 007344 G	CS\$ERHR= 000056	DROPN 015640	F\$MSG = 000011
BIT06 = 000100 G	CMDASC 003650 G	CS\$ERRO= 000060	DROPU 015430 G	F\$PROT= 000021
BIT07 = 000200 G	CMDDD 002220 G	CS\$ERSF= 000054	DROPUA 015554	F\$PWR = 000017
BIT08 = 000400 G	CMDLG 003354 G	CS\$ERSO= 000057	DRORTN 015632	F\$RPT = 000012
BIT09 = 001000 G	CMDPKM 004102 G	CS\$ESCA= 000010	DTAERM 005224 G	F\$SEG = 000003
BIT1 = 000002 G	CMDPKT 002310 G	CS\$ESEG= 000005	DTAER2 004677 G	F\$SOFT= 000005
BIT10 = 002000 G	CMDSAV 003350 G	CS\$ESUB= 000003	DTAER3 004746 G	F\$SRV = 000010
BIT11 = 004000 G	CMDSEQ 003460 G	CS\$ETST= 000001	DTAER4 005010 G	F\$SUB = 000002
BIT12 = 010000 G	CMDSE2 003470 G	CS\$EXIT= 000032	DTAER5 005031 G	F\$SW = 000014
BIT13 = 020000 G	CMDTBL 003562 G	CS\$GTB= 000026	EF.CON= 000036 G	F\$TEST= 000001
BIT14 = 040000 G	CMDWRD 003346 G	CS\$GETW= 000027	EF.NEW= 000035 G	GCMDA 007416 G
BIT15 = 100000 G	CMD.CO= 000001 G	CS\$GPHR= 000043	EF.PWR= 000034 G	GENPAT 010030 G
BIT2 = 000004 G	CMD.C1= 000002 G	CS\$GPLO= 000030	EF.RES= 000037 G	GES = 100017 G
BIT3 = 000010 G	CMD.C2= 000004 G	CS\$GPRI= 000040	EF.STA= 000040 G	GETSTM 005157 G
BIT4 = 000020 G	CMD.C3= 000010 G	CS\$INIT= 000011	EINC 014310	GIT 010322
BIT5 = 000040 G	CMD.C4= 000020 G	CS\$INLP= 000020	END = 177777 G	GOWAIT 010636 G
BIT6 = 000100 G	CMD2M 026231	CS\$MANI= 000050	ENDERF= 003416	G\$CPK 002320 G
BIT7 = 000200 G	CMD3M 026304	CS\$MEM = 000031	ENDFLG= 003452	G\$CNTO= 000200
BIT8 = 000400 G	CMD4M 026312	CS\$MSG = 000023	ENDSP 025706	G\$DISP= 000003
BIT9 = 001000 G	CMD5M 026320	CS\$OPEN= 000034	ENDSP1 025514	G\$EXCP= 000400
BOE = 000400 G	CMD6M 026326	CS\$PNTB= 000014	ENDSP2 025516	G\$HILI= 000002
BORERS 013546 G	CMD7M 026334	CS\$PNTF= 000017	EOTFLG 003426 G	G\$LOLI= 000001
BPCRM 026237	CMD8M 026342	C\$PNTS= 000016	ERCVR 002205 G	G\$NO = 000000
BRCPK 002324 G	CNTBGN= 002554		ERLOG 003412 G	G\$OFFS= 000400



GSOFSI= 000376	JMPMSG 026302	L\$SPCP 002020 G	MSGPK0 002352 G	PRI02 = 000100 G
GSPRMA= 000001	JMP.C = 000040 G	L\$SPTP 002024 G	MSGPK1 002370 G	PRI03 = 000140 G
GSPRMD= 000002	JSJMP = 000167	L\$STA 002030 G	MSGPK2 002406 G	PRI04 = 000200 G
GSPRML= 000000	LENMSK 003356 G	L\$SW 002202 G	MSGPK3 002424 G	PRI05 = 000240 G
GSRADA= 000147	LOE = 040000 G	L\$TEST 002114 G	MS.RFC= 000004 G	PRI06 = 000300 G
GSRADB= 000000	LOG 014016 G	L\$TIML 002014 G	MS.XS0= 000006 G	PRI07 = 000340 G
GSRADD= 000040	LOT = 000010 G	L\$UNIT 002012 G	MS.XS1= 000010 G	PRXST 015642 G
GSRADL= 000120	LSACP 002110 G	L10000 002200	MS.XS2= 000012 G	PTCMDS 024742
GSRADO= 000020	LSAPT 002036 G	L10001 002310	MS.XS3= 000014 G	PWRFLG 003453 G
GSXFER= 000004	LSAU 021576 G	L10002 005370	NCMD.C= 177740 G	RANB 003360 G
G\$YES = 000010	LSAUT 002070 G	L10003 006314	NCNT 003340 G	RANBC = 153624 G
HAE 002204 G	LSAUTO 021040 G	L10004 006322	NCNT1 003342 G	RANCMD 023540
HAEM 025756	L\$CCP 002106 G	L10005 006330	NEXTSP 025214	RANDOM 003441 G
HALTM 004042 G	L\$CLEA 021462 G	L10006 006336	NEXTU 015400 G	RANP = 000007 G
HELP = 000000	L\$CO 002032 G	L10007 006344	NINUSE= 177774 G	RANRD 023600
HOE = 100000 G	L\$DEPO 002011 G	L10010 017540	NOINTM 004421 G	RANS 003362 G
HRDCNT 003304 G	L\$DESC 002136 G	L10012 021036	NRDYM 021424	RANSC = 032561 G
IBE = 010000 G	L\$DESC 002076 G	L10013 021326	NSSRM 004271 G	RANW 023700
IDU = 000040 G	L\$DEVP 002060 G	L10014 021522	NUMBM 026251	RANWR 023654
IER = 020000 G	L\$DISP 002124 G	L10015 021574	NURTY1 005073 G	RANWV 023666
IE.C = 000200 G	L\$DLY 002116 G	L10016 021670	GFLINM 005127 G	RCVERM 026032
INIT10 017550	L\$DTP 002040 G	L10017 023256	ONEFIL= 000001	RDF = 104001 G
INIT15 017764	L\$DTP 002034 G	L10020 022024	OPFLAG 003456 G	RDR = 104401 G
INIT16 020004	L\$DU 021524 G	L10021 022050	OPP.C = 020000 G	RECCNT 003324 G
INTFLG 003416 G	L\$DUT 002072 G	L10022 022070	OSAPTS= 000000	RECLOG 003411 G
INTPRI= 000340 G	L\$DVTY 002164 G	L10023 022110	OSAU = 000001	RECRED 006312
IRE 003445 G	L\$EF 002052 G	L10024 022130	OSBGNR= 000001	RECTAP 006346 G
IREC 002211 G	L\$ETP 002102 G	L10025 022150	OSBNS= 000001	RECU 011272 G
IRECM 026106	L\$EXP1 002044 G	L10026 022170	OSDU = 000001	REPEAT= 050220
IREM 026153	L\$EXP2 002046 G	L10027 022210	OSERT= 000000	RERM 004550 G
ISR = 000100 G	L\$EXP4 002064 G	L10030 022230	OSGNSW= 000001	RETRY = 050216
IXE = 004000 G	L\$EXP5 002066 G	L10031 022250	OSPOIN= 000001	RETRYC 003404 G
ISAU = 000041	L\$SHARD 025074 G	L10032 022306	OSSETU= 000001	REWRT 013722
ISAUTO= 000041	L\$HIME 002120 G	L10033 022472	PASCNT 003254 G	RFBC 002654 G
ISCLN = 000041	L\$HPCP 002016 G	L10034 023732	PATCH 026350 G	RFCEM 004254 G
ISDU = 000041	L\$HPTP 002022 G	L10035 024076	PATERN 003374 G	RFREC 002754 G
ISHRD = 000041	L\$HW 002174 G	L10036 024230	PATRO 010114 G	RFUNR 002764 G
ISINIT= 000041	L\$ICP 002104 G	L10037 025070	PATR1 010152 G	RLEXM 004310 G
ISMOD = 000041	L\$INIT 017550 G	L10040 025144	PATR2 010172 G	RNF = 125401 G
ISMSG = 000041	L\$LADP 002026 G	L10041 026350	PATR3 010202 G	RNOPSC= 177740 G
ISPROT= 000040	L\$LAST 026554 G	L10042 026566	PATR4 010226 G	RNR = 105401 G
ISPTAB= 000041	L\$LOAD 002100 G	L10044 026572	PATR5 010240 G	RNYM 004504 G
ISPR = 000041	L\$LUN 002074 G	MBR = 100012 G	PATR6 010252 G	RPF = 105001 G
ISRPT = 000041	L\$MREV 002050 G	MEMOM 020734	PATR7 010272 G	RPR = 125001 G
ISSEG = 000041	L\$NAME 002000 G	MISCFG 003455 G	PATR8 010324 G	RPTCNT 003406 G
ISSETU= 000041	L\$PRIO 002042 G	MOD.C0= 000400 G	PATTBL 010072	RPTFLG 003443 G
ISSFT = 000041	L\$PROT 017542 G	MOD.C1= 001000 G	PATM 026271	RPT1A 016656
ISSRV = 000041	L\$PRT 002112 G	MOD.C2= 002000 G	PCMDWD 003352 G	RPT1B 016733
ISSUB = 000041	L\$REPP 002062 G	MOD.C3= 004000 G	PIRE 002214 G	RPT1C 017004
IS1ST = 000041	L\$REV 002010 G	MOVMSG 011206 G	PNT = 001000 G	RPT1D 017055
JLOC 003372 G	L\$RPT 016014 G	MSGCNT= 000016 G	PRI = 002000 G	RPT1E 017303
JLOOP 003370 G	L\$SOFT 025146 G	MSGPKA 002502 G	PRI00 = 000000 G	RPT1F 017161
JMP = 000040 G	L\$SPC 002056 G	MSGPKT 002334 G	PRI01 = 000040 G	RPT1G 017232



RPT1I	017427	TCC6	012560	G	TSPTHV=	000001	VFEXC	014402	G	SFSNAM=	000160
RPT1J	017333	TCC7	012722	G	TSPTNV=	000001	VFISU	014630	G	SFSNO =	000403
RPT1K	017420	TC2RTN	012026		TSSAVL=	177777	VFYCNT	003274	G	SFSOR =	000320
RRANV	002203	TIME1	003364	G	TSSSEGL=	177777	VFYDAT	014316	G	SFSRTN=	000300
RRBC	002614	TIME2	003366	G	TSSIZE=	000007	VFYFLG	003442	G	SFSSEL=	000140
RRCL =	000020	TOERM	004207	G	TSSUBN=	000000	VFY.C =	007100	G	SFSTHE=	000330
RRREC	002734	TOOMM	004460	G	TSTAGL=	177777	WLKZRO	010206		SFSTRU=	000404
RRUNR	002744	TRAPD4	003454	G	TSTAGN=	010045	WRBC	002554	G	SFSUNT=	000130
RRVM	025727	TRAP4	021454	G	TSTEMP=	000000	WRECL =	000020	G	SFSWHI=	000120
RTLE	012740	TSAM	004436	G	TSTEST=	000005	WRR =	105005	G	SFSYES=	000402
RTLRTN	013064	TSBA =	002452	G	TSTSTM=	177777	WRREC	002714	G	SIFLEV=	177777
RWCPK	002330	TSC.FC=	177717	G	TSTSTS=	000001	WRT =	104005	G	SISK0 =	000001
RWD =	102010	TSC.TC=	177761	G	TSSAU =	010016	WRTY	013066	G	SISK1 =	000001
RWERR	003413	TSDB	002452	G	TSSAUT=	010013	WRTYCT	003244	G	SISK2 =	000001
RSSAVE	003400	TSSR	002462	G	TSSCLE=	010014	WRTYER	003410	G	SISK3 =	000001
SCCNT	003264	TSSREG	003402	G	TSSDAT=	010044	WRTYFG	003407	G	SISK4 =	000001
SCERM	004230	TSVCT	002472	G	TSSDU =	010015	WRUNR	002724	G	SISK5 =	000001
SCH =	140004	TS.A16=	000400	G	TSSHAR=	010040	WSSR	011152	G	SISK6 =	000001
SCHBK	002442	TS.A17=	001000	G	TSSHW =	010000	WTM =	100011	G	SLOCTA=	177777
SCHCNT=	000010	TS.NBA=	002000	G	TSSINI=	010012	WTR =	101011	G	SLSTCN=	177777
SEQEND	003550	TS.NXM=	004000	G	TSSMSG=	010003	WTV =	104105	G	SLSTIN=	000001
SETCH	006444	TS.OFL=	000100	G	TSSPC =	000001	WTVERM	004164	G	SLSTST=	177777
SETRW	006470	TS.RMR=	010000	G	TSSPRO=	010011	WTYBRF	013340		SLSTTA=	000001
SETUP	007452	TS.SC =	100000	G	TSSPTA=	010043	WTYCMD	013334		SMCALL=	000000
SFF =	105010	TS.SPE=	020000	G	TSSRPT=	010010	WTYWRD	013336		SNESTL=	177777
SFPTBL	002202	TS.SSR=	000200	G	TSSSOE=	010041	X\$ALWA=	000000		SNSK0 =	000120
SFR =	105410	TS.UPE=	040000	G	TSSSRV=	010007	X\$FALS=	000040		SNSK1 =	000120
SRF =	104010	TS4ADR	025120		TSSSUB=	010033	X\$OFFS=	000400		SNSK2 =	000110
SRR =	104410	TS4CL	002522	G	TSSSW =	010001	X\$TRUE=	000020		SNSK3 =	000110
STAERM	005372	TS4INT	002512	G	TSSTES=	010037	X0.BOT=	000002	G	SNSK4 =	000110
STAER1	005704	TS4INO	006316	G	T1	021672	X0.EOT=	000001	G	SNSK5 =	000110
STAER2	006062	TS4IN1	006324	G	T1SWB	003447	X0.LET=	020000	G	SNSK6 =	000110
STAER3	006141	TS4IN2	006332	G	T1.1	021702	X0.ONL=	000100	G	\$SAVLE=	177777
STAER4	006177	TS4IN3	006340	G	T1.10	022232	X0.RLL=	010000	G	\$SSK0 =	050432
STAER5	006217	TS4VCT	025135		T1.11	022252	X0.RLS=	040000	G	\$TAGLE=	177777
STAER6	006026	TSARGC=	000003		T1.12	022336	X0.TMK=	100000	G	\$TAGNU=	050450
STAER7	005776	TS\$CODE=	023004		T1.2	022026	X2.OPM=	100000	G	\$TEMP =	000402
STAF LG	003452	TS\$ERRN=	000001		T1.3	022052	X3.DCK=	000010	G	\$TSK0 =	050431
SVCGBL=	000000	TS\$EXCP=	000000		T1.4	022072	X3.RNY=	157100	G	\$TSK1 =	050432
SVCINS=	000001	TS\$FLAG=	000041		T1.5	022112	ZROPAT	010156		\$TSK2 =	050433
SVC\$SUB=	000000	TS\$FREE=	026572		T1.6	022132	\$BGNLE=	177777		\$TSK3 =	050434
SVCTAG=	000000	TS\$GMAN=	000000		T1.7	022152	\$ERFLG=	000400		\$TSK4 =	050447
SVCTST=	000000	TS\$HILI=	000010		T1.8	022172	\$F\$AND=	000310		\$TSK5 =	050446
SWBFLG	003444	TS\$LAST=	000001		T1.9	022212	\$F\$BAD=	000401		\$TSK6 =	050442
SWB.C =	010000	TS\$LOLI=	000000		T2	023260	\$F\$BLA=	000170		\$TSK7 =	050444
SLSYM=	010000	TS\$LSYM=	010000		T3	023734	\$F\$CAS=	000150		\$SARGC=	000000
TCCRA	011644	TS\$LTNO=	000005		T4	024100	\$F\$DEC=	000220		\$S\$BYTE=	000403
TCC0	011664	TS\$NEST=	177777		T5	024232	\$F\$DO =	000340		\$S\$CASE=	000000
TCC1	011702	TS\$NS0 =	000000		TSWEOT	024764	\$F\$FAL=	000405		\$SDST =	000000
TCC2	011720	TS\$NS1 =	000005		UAM =	000200	\$F\$G00=	000400		\$SELOC=	000402
TCC3	012030	TS\$NS2 =	000002		UNL =	100412	\$F\$IF =	000110		\$SERFL=	000000
TCC4	012046	TS\$PCNT=	000000		UNREC	003414	\$F\$INC=	000210		\$SFLAG=	000001
TCC5	012462	TS\$PTAB=	010043		URERM	004572	\$F\$LOO=	000200		\$S\$FROM=	000000

\$\$LOC = 024700	\$\$RETN= 000000	\$\$SRC = 000000	\$\$TGS2= 000000	. = 026572
\$\$LOCN= 000000	\$\$RTN1= 000000	\$\$TGSV= 000000	\$\$TO = 000000	
\$\$REG = 177777	\$\$RTN2= 000000	\$\$TGS1= 000000	\$\$TAG= 050000	

. ABS. 026572 000

ERRORS DETECTED: 0

CZTSHC,CZTSHC/SOL/EQ:ONEFILE=NLSTF.P11,SVC.SML,SPMAC.SML,LISTF.P11,CZTSHC.P11  
RUN-TIME: 149 156 .9 SECONDS  
RUN-TIME RATIO: 530/307=1.7  
CORE USED: 28K (56 PAGES)