

TSV05

TSV05 DATA REL
CVTSEA0

AH-T179A-MC
FICHE 1 OF 1

SEP 1982
COPYRIGHT© 1982
MADE IN USA



Table with multiple columns and rows of data, likely a technical specification or data log. The content is too faint to transcribe accurately but appears to be organized in a grid format.



.REM_

IDENTIFICATION

PRODUCT CODE: AC-T178A-MC

PRODUCT NAME: CVTSEAO TSV05 DATA RELIABILITY

PRODUCT DATE: 08-MAR-82

MAINTAINER: SCOTT SNOWDON

AUTHOR: DICK GORDON

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

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

COPYRIGHT (C) 1982,1982 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DECPDP
DECUSUNIBUS
DECTAPE

MASSBUS

48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104

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

2.0 OPERATING INSTRUCTIONS

2.1 HARDWARE PARAMETERS

2.2 SOFTWARE PARAMETERS

- 2.2.1 TS05 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

105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

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 BOUNDARY
3.1.2 ERROR #2 - TS05 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 - R/F 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 - RANDOM/OPERATOR SELECTED COMMAND SEQUENCE

6.0 DEVICE INFORMATION

6.1 GENERAL
6.2 Q-BUS INTERFACE SPECIFICATIONS
6.3 BIT DEFINITIONS FOR TSV05/TS05 REGISTERS
6.3.1 TSV05/TS05 REGISTER SUMMARY
6.3.2 TSV05 STATUS REGISTER (TSSR)
6.3.2.1 TSV05 EXTENDED DATA BUFFER REGISTER (TSDBX)
6.3.3 EXTENDED STATUS REGISTER 0 (XSTAT0)
6.3.4 EXTENDED STATUS REGISTER 1 (XSTAT1)
6.3.5 EXTENDED STATUS REGISTER 2 (XSTAT2)

SEQ 0004

162
163
164
165
166
167
168

6.3.6 EXTENDED STATUS REGISTER 3 (XSTAT3)
6.3.7 EXTENDED STATUS REGISTER 4 (XSTAT4)

7.0 DIAGNOSTIC HISTORY

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

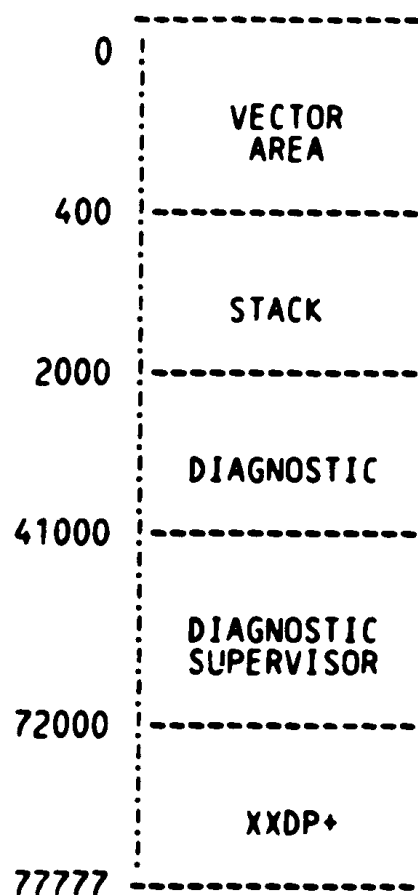
1.1.1 FUNCTIONAL DESCRIPTION

THIS PROGRAM CAN BE USED AS A BASIC FUNCTION TEST, A DATA RELIABILITY TEST, OR A COMPATABILITY TEST.

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 MEMORY 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 ONE CONTROLLER AND UP TO 2 DRIVES. THE 2 DRIVES
ARE ASSIGNED LOGICAL DRIVE NUMBERS 0 - 1 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 - RANDOM/OPERATOR SELECTED SEQUENCE UTILITY.

1.1.4.2 ERROR RECOVERY

ERROR RECOVERY IS PERFORMED ON READ, WRITE AND WRITE TAPE MARK FUNCTIONS
UNLESS ERROR RECOVERY IS INHIBITED BY THE OPERATOR AT START UP TIME.
THE READ FORWARD/READ REVERSE RETRY LIMIT IS 16 (8 IN THE SAME DIRECTION
AND 8 IN THE OPPOSITE DIRECTION). FOR MORE DETAILED INFORMATION ON ERROR RECOVERY
PROCEDURES, REFER TO SECTION 3.0 (ERROR REPORTING) OF THIS LISTING.

1.1.4.3 WRITE ERROR RECOVERY

THERE ARE 2 , SELECTABLE WRITE-ERROR-RECOVERY ALGORITHMS USED BY THIS DIAGNOSTIC:

1. MEDIA/OPERATIONAL SELECTIVE ALGORITHM
2. OPERATIONAL ALGORITHM

BY DEFAULT THE DIAGNOSTIC SELECTS THE FIRST ALGORITHM TO IDENTIFY
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 ?

IF 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

THIS ALGORITHM IDENTIFIES MEDIA RELATED WRITE ERRORS FROM
OPERATIONAL ONES.

ALGORITHM

A WRITE RETRY SUBROUTINE IS CALLED BY THE RECOVERABLE ERROR SUBROUTINE WHICH IS
ENTERED UPON DETECTION OF A WRITE RECOVERABLE ERROR.
THE WRITE RETRY SUBROUTINE ATTEMPTS TO REWRITE THE RECORD IN SAME SPOT ON TAPE
4 TIMES.

284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340

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

IF ANY OF THE 4 REWRITE ATTEMPTS FAIL, THE ROUTINE WILL ERASE THE BAD RECORD, AND LOG BAD SPOT AT THAT RECORD NUMBER, THE ROUTINE WILL THEN ATTEMPT TO WRITE THE RECORD AGAIN 3 INCHES FURTHER DOWN TAPE AND RETRY THIS SEQUENCE 4 TIMES, FOR UP TO 4 REPEATS EACH.

IF A RECORD CANNOT BE WRITTEN WITHOUT RECOVERABLE ERRORS AFTER 4 RETRIES, THEN THE ROUTINE WILL ERASE THE RECORD AND 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 BOT TO EOT PASS OF TAPE. WHEN 20 BAD SPOTS HAVE BEEN LOGGED, WHETHER ON THE SAME RECORD NUMBER OR NOT, TAPE IS CONSIDERED DEFECTIVE: A BAD TAPE OVERFLOW MESSAGE IS PRINTED AND THE UNIT IS REWOUND, THEN DROPPED.

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

ALL BADLY WRITTEN RECORDS LOGGED WITH RECOVERABLE ERRORS ARE ERASED UNTIL RECOVERED, INCLUDING THE RECORD AT THE 20TH BAD SPOT, SO THAT ALL RECORDS LEFT ON TAPE ARE KNOWN 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. THAT LONG STRETCH OF BAD TAPE WOULD THEN BE LOGGED WITH 20 BAD SPOTS AT SAME RECORD NUMBER AND THE TAPE CONSIDERED DEFECTIVE.

BAD SPOTS REPORTS

IF THE PRINTING 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 SPOT ACTUALLY PRECEDES THE RECORD NUMBER THAT IDENTIFIES IT. 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 PASS OF TAPE. IN THAT REPORT, ALL COUNTS ARE CUMULATIVE FROM PASS TO PASS, EXCEPT FOR THE NUMBER OF BAD SPOTS: IT RELATES TO A 'BOT TO EOT 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. DON'T CONFUSE THIS WITH A PASS BY THE SUPERVISOR WHICH IS DEFINED AS A RUN THROUGH A 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.

TO CLEAR CUMULATIVE COUNTS, ANSWER 'Y' TO: CLEAR COUNTERS (L) Y ?.
THE BAD TAPE SPOTS COUNT IS THEN CLEARED WHEN WRITING THE TAPE 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 FROM TAPE PASS TO TAPE PASS
FOR THE BAD SPOTS ID TO REMAIN CONSISTENT IN THOSE TAPE PASSES.

EXAMPLE OF A PRINT OUT FOR A BAD SPOT ON TAPE:

CVTSE 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

CMDPKT	TSBA	RFC	TSSR	TCC
100205	002406	000000	100210	4
026600				
000000				
003107				
XST0	XST1	XST2	XST3	XST4
000350	000002	100400	000000	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				

CVTSE 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

CMDPKT	TSBA	RFC	TSSR	TCC
100205	002406	000000	100210	4
026600				
000000				
004000				
XST0	XST1	XST2	XST3	XST4
000350	000002	100010	000000	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

398 RECOVERABLE ERRORS 1 0 0
399 UNRECOVERABLE ERRORS 0 0 0
400 WRITE RETRIES 3

401 2 BAD SPOTS THIS TAPE PASS PRECEDING RECORD #:

402 SPEC COND 5 6
403 HARD 0 FATAL 0 COMPARE 0
404 DR>

405 THIS EXAMPLE SHOWS:

406 RECORD 6 RECOVERED ON 2ND RETRY GROUP
407 THE 2 BAD SPOTS RESIDE IN A 18 INCH ERASE GAP BETWEEN RECORDS 5 AND 6
408 RECORD 10210 RECOVERED ON 1ST RETRY OF 4 GOOD REPEATS
409 3 WRITE GROUP RETRIES ATTEMPTED, RESULTING IN:
410 1 RECOVERABLE WRT ERR FROM RECORD 10210
411 2 BAD SPOTS BETWEEN RECORDS 5 AND 6

412 1.1.4.3.2 OPERATIONAL WRITE-ERROR-RECOVERY ALGORITHM

413 WHEN THIS ALGORITHM IS SELECTED, THE TSV05 WRITE RETRY COMMAND
414 IS ISSUED UP TO 16 TIMES OR UNTIL RECORD IS RECOVERED, ON
415 A WRITE RECOVERABLE ERROR. THE WRITE RETRY COMMAND CONSISTS
416 OF A SPACE REVERSE OVER THE BAD RECORD, THEN AN ERASE OF 3 INCHES
417 OF TAPE AND REWRITE OF THE RECORD. THAT COMPOSITE COMMAND
418 DOES NOT ALLOW THE DETECTION OF BAD SPOTS ON TAPE.
419 THEREFORE NO BAD TAPE SPOTS STATUS IS PRINTED.

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

422 1.1.4.4 DIAGNOSTIC TIMING ADJUSTMENT

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

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

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

438 1.2 SYSTEM REQUIREMENTS

439 -----

1.2.1 HARDWARE REQUIREMENTS

PDP-11/23 PROCESSOR WITH 32K OR MORE OF MEMORY
CONSOLE DEVICE (VTS2,LA36,ETC.)
PROGRAM LOAD DEVICE
TSV05/TS05

1.2.2 SOFTWARE REQUIREMENTS

DIAGNOSTIC SUPERVISOR

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. CIOPMA0 XXDP+ PROGRAMMER'S MANUAL; DOCUMENT NUMBER AC-S296A-AC
DATE: 14 JULY 1980.
2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE; DOCUMENT NUMBER EK-TSV05-UG-001
DATE: AUGUST 1982
3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSV05-TM-001
DATE: AUGUST 1982
4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSV05-IN-001
DATE: AUGUST 1982

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

ORDER OF HOST CPU DIAGNOSTIC USAGE:

- 1) CONTROL LOGIC PROGRAM - ALL TESTS.
(VTS A,VTS B,VTSC,VTSD)
- 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.

SEQ 0011

512
513
514

VTSA,VTSB,VTSC, AND VTSD HAVE ALL SUCESSFULLY RUN WITHOUT ERRORS.

2.0 OPERATING INSTRUCTIONS

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

COMMANDS

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

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

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

OPERATOR COMMANDS

THE TSV05 DIAGNOSTIC IS A PDP-11/23 DIAGNOSTIC SUPERVISOR COMPATIBLE
PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE
PDP-11 PROGRAMMER'S MANUAL "CIOPMAO XXDP+ PROGRAMMERS MANUAL, NUMBER
AC-S296A-AC. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC XXDP MEDIA

(HMDLBO XXDP+ DL MONITOR 28K
BOOTED VIA UNIT 0
ENTER DATE (DD-MMM-YR): " ENTER DATE OR JUST <CR> "
RESTART ADDRESS: 153726
50 HZ? N " <CR> "
LSI? N " Y<CR> "
THIS IS XXDP+. TYPE 'H' OR 'H/L' FOR DETAILS
R VTSEAO
VTSEA0BINDRS LOADED
DIAG. RUN-TIME SERVICES REV D. APR 79

CVTSE-A-0
TSV05 DATA RELIABILITY
UNIT IS TSV05

SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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

630 FLAGS
631 -----
632

633 FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS
634 LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN
635 CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS
636 ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE
637 FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR
638 ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS,
639 NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR
640 CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.
641

642 FLAG	EFFECT
643 ----	-----
644 HOE	HALT ON ERROR - CONTROL IS RETURNED TO 645 RUNTIME SERVICES COMMAND MODE
646 LOE	LOOP ON ERROR
647 IER*	INHIBIT ALL ERROR REPORTS
648 IBR*	INHIBIT ALL ERROR REPORTS EXCEPT 649 FIRST LEVEL (FIRST LEVEL CONTAINS 650 ERROR TYPE, NUMBER, PC, TEST AND UNIT)
651 IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE 652 CALLED BY PRINTX MACRO'S)
653 PRI	DIRECT MESSAGES TO LINE PRINTER
654 PNT	PRINT TEST NUMBER AS TEST EXECUTES
655 BOE	'BELL' ON ERROR
656 UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
657 ISR	INHIBIT STATISTICAL REPORTS (DOES NOT 658 APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT 659 STATISTICAL REPORTING)
660 IDR	INHIBIT PROGRAM DROPPING OF UNITS
661 ADR	EXECUTE AUTODROP CODE
662 LOT	LOOP ON TEST

663 *ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1
664

665 SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY
666 SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE,
667 TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS
668 AND TYPE A 'BELL' ON ERROR, YOU MAY USE THE FOLLOWING STRING:
669

670 /FLAGS:LOE:IER:BOE
671

672 2.1 HARDWARE PARAMETERS
673 -----
674

675 ON A 'N' RESPONSE TO 'CHANGE HW?', THE DIAG SHALL RUN ASSUMING
676 ONE UNIT AT TSDB = 172520 WITH A VECTOR = 224 AND DRIVE=0.
677

678 ON A 'Y' RESPONSE TO 'CHANGE HW?' QUESTION, THEN
679 THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE
680 VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT
681 VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.
682
683
684
685
686

TSDB ADDRESS (0) 172520 ?

VECTOR (0) 224 ?

SELECT DRIVE 0-1 (0) ?

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 TSDB(S). IF NO RESPONSE, IT DROPS THE UNIT(S) IMMEDIATELY WITH THE FOLLOWING MESSAGE:

BUS TRAP AT XXXXXX (XXXXXX = TSDB AD)
INTERFACE BAD OR NOT SET TO ABOVE ADDRESS.

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 DRIVE IS CHECKED. A MESSAGE SHALL BE PRINTED EVERY SO OFTEN TO WARN THE OPERATOR OF DRIVES BEING NOT READY OR OFF-LINE. THESE DRIVES SHALL BE DROPPED AFTER A REASONABLE AMOUNT OF TIME.

2.2 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED WHEN ONE ANSWERS YES TO THE CHANGE SOFTWARE QUESTION 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 ? (SEE NOTE1:)

DEFAULT SWITCH SETTINGS (L) Y ?

100IPS (L) N ?

WRITE BUFFERING (L) N ?

744 READ BUFFERING (L) N ?

745
746
747 ANSWERING NO TO THE DEFAULT SWITCH QUESTION WILL CAUSE THE
748 100 IPS QUESTION TO BE ASKED.

749
750 ANSWERING YES TO THE 100 IPS QUESTION WILL INHIBIT THE LAST
751 TWO QUESTIONS.

752
753 ANSWERING NO TO THE 100 IPS QUESTION WILL CAUSE THE WRITE
754 BUFFERING QUESTION TO BE ASKED.

755
756 ANSWERING YES TO THE WRITE BUFFERING QUESTION WILL INHIBIT THE
757 LAST QUESTION.

758
759 ANSWERING NO TO THE WRITE BUFFERING QUESTION WILL CAUSE THE
760 READ BUFFERING QUESTION TO BE ASKED.

761
762 NOTE1: THIS QUESTION SHOULD BE ANSWERED (N) UNLESS AN
763 OPERATOR SELECTED SEQUENCE IS TO BE EXECUTED.
764 IF THIS QUESTION WAS ANSWERED Y, THE FOLLOWING
765 QUESTIONS MUST BE ANSWERED OR DEFAULTED WITH A <CR> ONLY:

766 CHARACTERISTICS CODE (D) 40 ?	(0,20,40,200)	(OCTAL)
767 CMD/2 (D) 13 ?	(1-27)	(DECIMAL)
768 BRF COUNT (D) 1 ?	(1-2K)	(DECIMAL)
769 # OF OPERATIONS (D) 1 ?	(1-32K)	(DECIMAL)
770 PATTERN (D) 7 ?	(0-8)	(DECIMAL)
771 CMD/3 (D) 4 ?	(1-27)	(DECIMAL)
772 BRF COUNT (D) 2048 ?	(1-2K)	(DECIMAL)
773 # OF OPERATIONS (D) 32000 ?	(1-32K)	(DECIMAL)
774 PATTERN (D) 7 ?	(0-8)	(DECIMAL)
775 CMD/4 (D) 3 ?	(1-27)	(DECIMAL)
776 BRF COUNT (D) 2048 ?	(1-2K)	(DECIMAL)
777 # OF OPERATIONS (D) 32000 ?	(1-32K)	(DECIMAL)
778 PATTERN (D) 7 ?	(0-8)	(DECIMAL)
779 CMD/5 (D) 2 ?	(1-27)	(DECIMAL)
780 BRF COUNT (D) 2048 ?	(1-2K)	(DECIMAL)
781 # OF OPERATIONS (D) 32000 ?	(1-32K)	(DECIMAL)
782 PATTERN (D) 7 ?	(0-8)	(DECIMAL)
783 CMD/6 (D) 13 ?	(1-27)	(DECIMAL)
784 BRF COUNT (D) 1 ?	(1-2K)	(DECIMAL)
785 # OF OPERATIONS (D) 1 ?	(1-32K)	(DECIMAL)
786 PATTERN (D) 7 ?	(0-8)	(DECIMAL)
787 CMD/7 (D) 27 ?	(1-27)	(DECIMAL)
788 BRF COUNT (D) 2048 ?	(1-2K)	(DECIMAL)
789 # OF OPERATIONS (D) 32000 ?	(1-32K)	(DECIMAL)
790 PATTERN (D) 7 ?	(0-8)	(DECIMAL)
791 CMD/8 (D) 27 ?	(1-27)	(DECIMAL)
792 BRF COUNT (D) 2048 ?	(1-2K)	(DECIMAL)
793 # OF OPERATIONS (D) 32000 ?	(1-32K)	(DECIMAL)
794 PATTERN (D) 7 ?	(0-8)	(DECIMAL)

795
796 NOTE: THE PROGRAM AUTOMATICALLY INSERTS A CHARACTERISTIC CODE OF 40
797 AS THE FIRST COMMAND IN THE SEQUENCE TABLE. IF A
800 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 TSV05/TS05 PROGRAMMING SPECIFICATION FOR DESCRIPTION.
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.
6	ALTERNATING BYTES OF 000 AND 377.

SEQ 0018

858
859

7
8

RANDOM DATA PATTERN.
NO PATTERN GENERATION.

2.3 EXAMPLES OF SOFTWARE DIALOGUE

CHANGE HW (L) ?

#UNITS (D) ?

TSDB ADDRESS (O) 172520 ?

VECTOR (O) 224 ?

SELECT DRIVE 0-1 (O) ?

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

CHANGE SW (L) ?

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 ? <CR>

PRINT RECOVERABLE ERRORS (L) N ? Y<CR>

HALT AFTER EACH CMD (L) N ? <CR>

INHIBIT RECOVERY (L) N ? <CR>

BAD TAPE SPOT DETECTION (L) Y ? <CR>

DISABLE INTERRUPTS (L) N ? <CR>

INHIBIT RFC ERROR REPORT (L) N ? <CR>

CHANGE CMD SEQUENCE (L) N ? <CR>

DEFAULT SWITCH SETTINGS (L) Y ? <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>

918 INHIBIT RFC ERROR REPORT (L) N ? Y<CR>
919 CHANGE CMD SEQUENCE (L) N ? N<CR>
920 DEFAULT SWITCH SETTINGS (L) Y ? <CR>
921
922

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

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

927 - CHANGE SW (L) ? Y<CR>
928 CLEAR COUNTERS (L) N ? Y<CR>
929 RESET RANDOM VARIABLES (L) N ? N<CR>
930 PRINT RECOVERABLE ERRORS (L) N ? N<CR>
931 HALT AFTER EACH CMD (L) N ? N<CR>
932 INHIBIT RECOVERY (L) N ? N<CR>
933 BAD TAPE SPOT DETECTION (L) Y ? N<CR>
934 DISABLE INTERRUPTS (L) N ? Y<CR>
935 INHIBIT RFC ERROR REPORT (L) N ? Y<CR>
936 CHANGE CMD SEQUENCE (L) N ? Y<CR>
937 CHARACTERISTICS CODE (O) 40 ? 40<CR>
938 CMD/2 (D) 5 ? 13<CR> (REWIND)
939 BRF COUNT (D) 2048 ? 1<CR>
940 # OF OPERATIONS (D) 10 ? 1<CR>
941 PATTERN (D) 7 ? 1<CR>
942 CMD/3 (D) 5 ? 4<CR> (WRITE)
943 BRF (D) 2048 ? 1000<CR>
944 # OF OPERATIONS (D) 10 ? 10000<CR>
945 PATTERN (D) 7 ? 1<CR>
946 CMD/4 (D) 5 ? 27<CR> (END)
947 BRF (D) 2048 ? <^Z>

2.4 EXECUTION TIMES

2.4.1 SYSTEM CONFIGURATION

952 -----
953
954 PDP11/23
955 MOS MEMORY
956 LA36
957 TSV05/TS05
958

2.4.2 TEST EXECUTION TIMES (2400 FT. TAPE)

959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
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 - RANDOM/OPERATOR SELECTED SEQUENCE -20 MINUTES PER PASS.

NOTE: ALL EXECUTION TIMES ARE SHOWN FOR ONE DRIVE DEPRATION.

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:

```
CVTSE 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 XST4
XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX
```

* CAUTION *

INTERPRET THAT 'RECORD READ' COUNT WITH CAUTION.
IF VERY DIFFERENT FROM RECORD COUNT TRACKED BY THE DIAGNOSTIC,
TAPE 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 THE TAPE IS NOT REWOUND. THIS IS
NECESSARY BECAUSE THERE IS NO ACCURATE WAY TO DETERMINE
ON WHAT RECORD COUNT OF WHICH 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:

```
CVTSE 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
```

1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088

000371

XST0	XST1	XST2	XST3	XST4
000350	000002	100004	000000	040055

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 - TSOS NOT READY:

BEFORE ANY COMMAND IS ISSUED TO THE TSOS, THE SUBSYSTEM READY BIT IN THE TSSR 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 TSOS 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

1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145

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.

1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202

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

1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259

	WRT	RDR	RDF
RECOVERABLE ERRORS	XXXXX	XXXXX	XXXXX
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 TS05 FUNCTIONS.

SUBTEST 1 - SET CHAR, DRIVE INIT, GET STATUS.

- + SET CHARACTERISTIC 200.
- + DRIVE INITIATE.
- + SET CHARACTERISTIC 20.
- + GET STATUS
- + SET CHARACTERISTIC 40.
- + PRINT TS05 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.

1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299

+ 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 64 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 - RANDOM/OPERATOR SELECTED COMMAND SEQUENCE.

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. OPERATOR CAN ENTER SEQUENCE OF COMMANDS UP TO SEVEN IF THEY DON'T WANT DEFAULT SEQUENCE.

6.0 DEVICE INFORMATION TABLES

6.1 GENERAL

THE TS05 TAPE SUBSYSTEM CONSISTS OF A TSV05 Q-BUS CONTROLLER CONNECTED TO A TS05 DRIVE. FROM A SOFTWARE VIEWPOINT THIS CONFIGURATION IS UNIQUE (FOR A Q-BUS 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 TS05 SUBSYSTEM TO FETCH THE WORD(S) WITHIN THE COMMAND PACKET. THE WORDS WITHIN THE COMMAND PACKET ARE:

1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389

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 TS05 TO TRANSFER EXTENDED STATUS INFORMATION TO THE MEMORY AREA POINTED TO BY THE BUFFER ADDRESS OF THE "GET STATUS" COMMAND. THERE ARE FIVE EXTENDED STATUS REGISTERS. SEE .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) .

SEQ 0029

1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401

6.2 Q-BUS INTERFACE SPECIFICATIONS

TSV05/ TS05 -----	INT. VECTOR -----	UNIBUS ADDRESS -----	REGISTER -----
FIRST	224	772520 772522	TSBA/TSDB TSSR

6.3 BIT DEFINITIONS FOR TSV05/TS05 REGISTERS

6.3.1 TSV05/TS05 REGISTER SUMMARY

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R/O) TSBA	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
(W/O) TSDB	P15	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P17	P16
(R/O) TSSR	SC	0	SCE	RMR	NXM	NBA	A17	A16	SSR	OFL	FC1	FC0	TC2	TC1	TC0	0
(W/O) TSDBX	BT	0	0	0	P21	P20	P19	P18	(TSDBX EXISTS ONLY WHEN ENABLED BY THE EXTENDED FEATURES SWITCH ON THE M7196)							
XST0	TMK	RLS	LET	RLI	WLE	NEF	ILC	ILA	MOT	ONL	IE	VCK	PED	WLK	BJT	EOT
XST1	DLT	0	COR	0	0	0	0	RBP	0	0	0	0	0	0	UNC	0
XST2	OPM	RCE	0	0	0	WCF	0	0	RL7	RL6	RL5	RL4	RL3	RL2	RL1	RL0
XST3	MICRO DIAGNOSTIC ERROR CODE								0	OPI	REV	TRF	DCK	0	0	RIB
XST4	HSP	RCE	0	0	0	0	0	0	WRITE RETRY COUNT							

TERMINATION CLASS CODES (TSSR TC0-TC2):

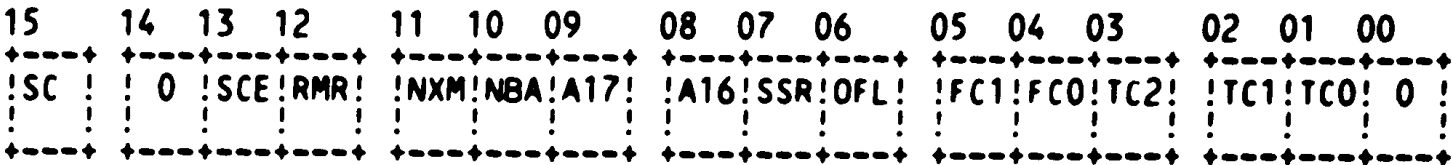
- 0 = NORMAL TERMINATION
- 1 = ATTENTION CONDITION
- 2 = TAPE STATUS ALERT
- 3 = FUNCTION REJECT
- 4 = RECOVERABLE ERROR - TAPE POSITION = ONE RECORD
DOWN TAPE FROM START OF FUNCTION
- 5 = RECOVERABLE ERROR - TAPE NOT MOVED
- 6 = UNRECOVERABLE ERROR - TAPE POSITION LOST
- 7 = FATAL CONTROLLER ERROR

FATAL CLASS CODES (TSSR FC0-FC1):

- 0 = MICRO DIAGNOSTIC FAILURE. SEE ERROR CODE BYTE (XST3) FOR FAILED FUNCTION.
- 1 = RESERVED
- 2 = NOT USED
- 3 = RESERVED FOR FUTURE USE ALWAYS READ AS A 0

6.3.2 TSV05 STATUS REGISTER (TSSR)

Q-BUS ADDRESS + 2 - READ ONLY



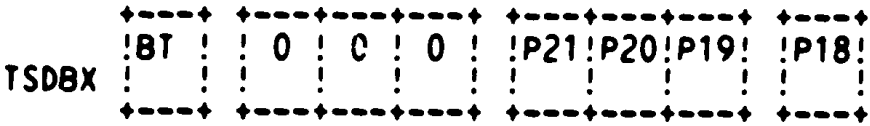
BIT	NAME	TCC	DEFINITION
15	SC	S	SPECIAL CONDITION. WHEN SET, INDICATES THAT THE LAST COMMAND DID NOT COMPLETE WITHOUT INCIDENT. SPECIFICALLY, EITHER AN ERROR WAS DETECTED OR AN EXCEPTION CONDITION OCCURRED. EXCEPTION CONDITIONS CAN BE TAPE MARKS ON READ COMMANDS, REVERSE MOTION AND AT BOT, EOT WHILE WRITING, ETC. MAY ALSO BE SET BY THE ERROR BITS CONTAINED IN THE TSSR REGISTER: SCE, RMR, AND NXM. THE TERMINATION CLASS BITS ARE SOMETHING OTHER THAN 0 (UNLESS RMR IS THE ONLY ERROR - SEE RMR).
14	-	-	RESERVED (ALWAYS A 0)
13	SCE	FC0	SANITY CHECK ERROR-SETS WHEN THE CONTROLLER DETECTS AN ABNORMAL CONDITION WITHIN ITSELF DURING EXECUTION OF IT'S FUNCTIONS AND THE PROBLEM IS SERIOUS ENOUGH THAT A MESSAGE PACKET IS NOT STORED.
12	RMR	S	REGISTER MODIFICATION REFUSED. SET BY THE TSV05 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 TS05 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 TSV05 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 TS05 NEEDS A MESSAGE BUFFER ADDRESS. THIS BIT IS CLEARED DURING THE SET CHARACTERISTICS

1518				COMMAND (IF A GOOD ADDRESS WAS GIVEN).
1519				
1520	09	A17	S	BUS ADDRESS BIT 17. A17 AND A16 (BIT 08) TRACK
1521				THE VALUES OF BITS 17 AND 16 OF THE TSBA
1522				REGISTER. LOADED FROM TSDB BITS 01-00 WHEN TSDB
1523				IS WRITTEN.
1524				
1525	08	A16	S	BUS ADDRESS BIT 16. SEE A17 (BIT 09).
1526				
1527	07	SSR	S	SUB-SYSTEM READY. WHEN SET, INDICATES THAT THE
1528				TSV05/TS05 SUBSYSTEM IS NOT BUSY AND IS READY TO
1529				ACCEPT A NEW COMMAND POINTER.
1530				
1531	06	OFL	S,1,3	OFF-LINE. WHEN SET, INDICATES THAT THE TS05 IS
1532				OFF-LINE AND UNAVAILABLE FOR ANY TAPE MOTION
1533				COMMANDS. THIS BIT CAN CAUSE A TERMINATION CLASS
1534				OF 1 (ON ATTN INTERRUPT) OR 3 (RESULTS IN NEF).
1535				
1536	05	FC1	7	FATAL TERMINATION CLASS 01. FC1 AND FC0 (BIT
1537				04) ARE USED TO INDICATE THE TYPE OF FATAL
1538				ERROR WHICH HAS OCCURRED ON THE TS05. THESE
1539				BITS ARE VALID ONLY WHEN SC IS SET AND THE
1540				TERMINATION CLASS CODE BITS ARE ALL SET (111).
1541				
1542	04	FC0	7	FATAL TERMINATION CLASS 00. SEE FC1 (BIT 05).
1543				
1544	03	TC2	S	TERMINATION CLASS BIT 02. THIS BIT, ALONG WITH
1545				THE TC1 AND TC0 BITS, ACT AS AN OFFSET VALUE
1546				WHENEVER AN ERROR OR EXCEPTION CONDITION OCCURS
1547				ON A COMMAND. EACH OF THE EIGHT POSSIBLE
1548				VALUES OF THIS FIELD REPRESENT A PARTICULAR
1549				CLASS OF ERRORS OR EXCEPTIONS. THE CONDITIONS
1550				IN EACH CLASS HAVE SIMILAR SIGNIFICANCE AND, AS
1551				APPLICABLE, RECOVERY PROCEDURES. THE CODE
1552				PROVIDED IN THIS FIELD IS EXPECTED TO BE
1553				UTILIZED AS AN OFFSET INTO A DISPATCH TABLE FOR
1554				HANDLING OF THE CONDITION.
1555				
1556	02	TC1	S	TERMINATION CLASS BIT 01. SEE TC2 (BIT 03).
1557				
1558	01	TC0	S	TERMINATION CLASS BIT 00. SEE TC2 (BIT 03).
1559				
1560	00	-	-	NOT USED. (ALWAYS A 0)

Q-BUS ADDRESS + 2 - WRITE ONLY

SUBSYSTEM INITIALIZE

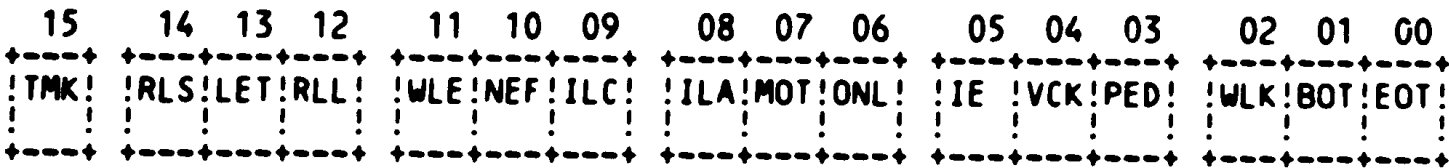
6.3.2.1 TSV05 EXTENDED DATA BUFFER REGISTER (TSDBX)



(TSDBX EXISTS ONLY WHEN
ENABLED BY THE EXTENDED
FEATURES SWITCH ON THE M7196)

BIT	NAME	TCC	DEFINITION
15	BT	-	BOOT COMMAND BIT. WHEN WRITTEN TO A 1, WITH SSR=1, CAUSES THE TAPE TO BE REWOUND TO BOT, THE FIRST TAPE RECORD TO BE SKIPPED, AND THE SECOND RECORD TO BE LOADED INTO CPU MEMORY SPACE STARTING AT LOCATION 0.
14-12			RESERVED (ALWAYS A 0)
11-08	P<21:18>		COMMAND POINTER BITS 21-18. WHEN THE TSDBX IS WRITTEN AND SSR=1, THE DATA IS LOADED INTO BITS 21-18 OF THE INTERNAL TSBA REGISTER.
07-00			RESERVED (ALWAYS A 0)

6.3.3 EXTENDED STATUS REGISTER 0 (XSTAT0)



BIT	NAME	TCC	DEFINITION
15	TMK	S,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.

1626				
1627	13	LET	2	LOGICAL END OF TAPE. SET ONLY ON THE SKIP TAPE
1628				MARKS COMMAND WHEN EITHER TWO CONTIGUOUS TAPE
1629				MARKS ARE DETECTED OR WHEN MOVING OFF OF BOT
1630				AND THE FIRST RECORD ENCOUNTERED IS A TAPE
1631				MARK. THE SETTING OF THIS BIT WILL NOT OCCUR
1632				UNLESS THIS MODE OF TERMINATION IS ENABLED
1633				THROUGH USE OF THE SET CHARACTERISTICS COMMAND.
1634				
1635	12	RLL	2	RECORD LENGTH LONG. WHEN SET, THIS BIT
1636				INDICATES THAT THE RECORD READ WAS LONGER THAN
1637				THE BYTE COUNT SPECIFIED.
1638				
1639	11	WLE	3.6	WRITE LOCK ERROR. WHEN SET, INDICATES THAT A
1640				WRITE OPERATION WAS ISSUED BUT THE MOUNTED TAPE
1641				DID NOT CONTAIN A WRITE ENABLE RING OR THE WRT
1642				LOCK SWITCH ACTIVATED DURING THE OPERATION.
1643				
1644	10	NEF	3	NON-EXECUTABLE FUNCTION. WHEN SET, INDICATES
1645				THAT THE COMMAND COULD NOT BE EXECUTED DUE TO
1646				ONE OF THE FOLLOWING CONDITIONS:
1647				
1648				- THE COMMAND SPECIFIED REVERSE TAPE
1649				DIRECTION BUT THE TAPE WAS ALREADY
1650				POSITIONED AT BOT.
1651				- THE ISSUING OF ANY MOTION COMMAND EXCEPT

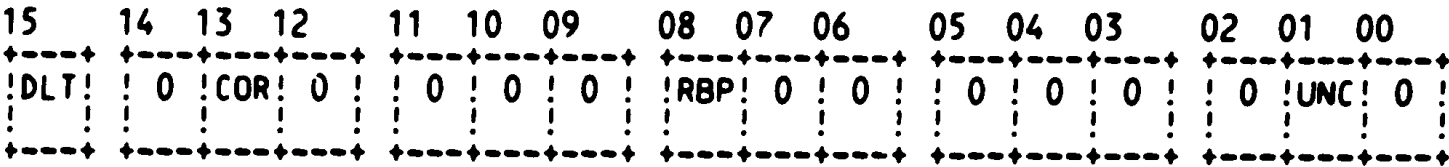
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701

WHEN THE VOLUME CHECK BIT IS SET.
- ANY COMMAND, EXCEPT GET STATUS OR DRIVE INITIALIZE, WHEN THE TS05 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 TS05.
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 TS05 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. ALWAYS SET, INDICATES THAT THE TS05 IS CAPABLE OF READING AND WRITING ONLY 1600 BPI PHASE ENCODED DATA.
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.

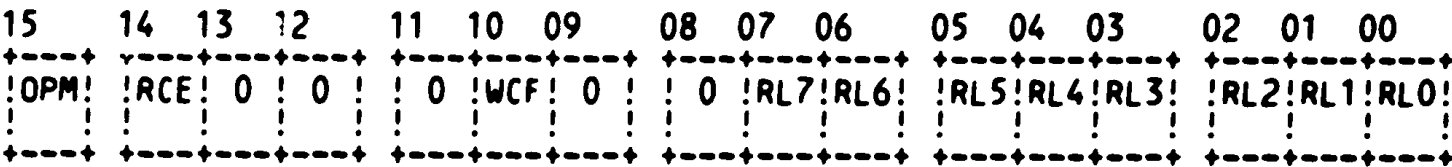
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738

6.3.4 EXTENDED STATUS REGISTER 1 (XSTAT1)



BIT	NAME	TCC	DEFINITION
---	----	---	-----
15	DLT	4	DATA LATE. SET WHEN THE FIFO IS FULL ON A READ OR EMPTY ON A WRITE. THESE CONDITIONS OCCUR WHENEVER THE Q-BUS LATENCY EXCEEDS THE DATA TRANSFER RATE OF THE TS05.
14	-	-	NOT USED. (ALWAYS A 0)
13	COR	S	CORRECTABLE DATA. CORRECTABLE DATA ERROR HAS BEEN ENCOUNTERED.
12-09			RESERVED (ALWAYS A 0)
08	RPB	4	READ BUS PARITY ERROR. SET WHEN CONTROLLER DETECTS A PARITY ERROR ON THE READ DATA LINES OF THE TRANSPORT BUS.
07-02 & 00			RESERVED (ALWAYS A 0)
01	UNC	4	UNCORRECTABLE DATA ERROR.

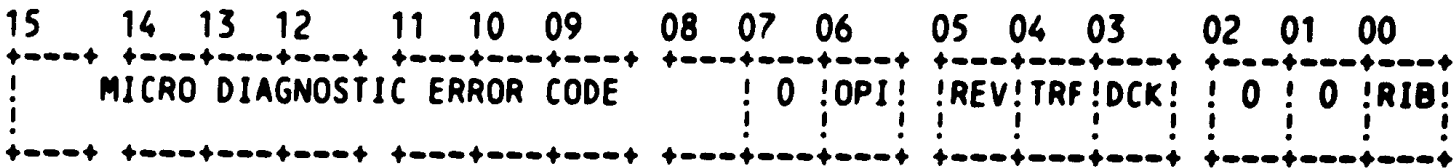
6.3.5 EXTENDED STATUS REGISTER 2 (XSTAT2)



BIT	NAME	TCC	DEFINITION
15	OPM	S	OPERATION IN PROGRESS. (TAPE MOVING)
14	RCE	7,F2	RAM CHECKSUM ERROR. CAUSES FATAL CLASS 2 BECAUSE THE ERROR MIGHT HAVE OCCURRED DURING THE TRANSMISSION OF THE MESSAGE PACKET.
13-11			RESERVED (ALWAYS A 0)
10	WCF	7	WRITE CLOCK FAILURE. SET DURING A WRITE TO INDICATE THAT THE FIFO IS NOT BEING EMPTIED BY THE TRANSPORT.
09-08			RESERVED (ALWAYS A 0)
07-00	RL	-	REVISION LEVEL.
	7-0		

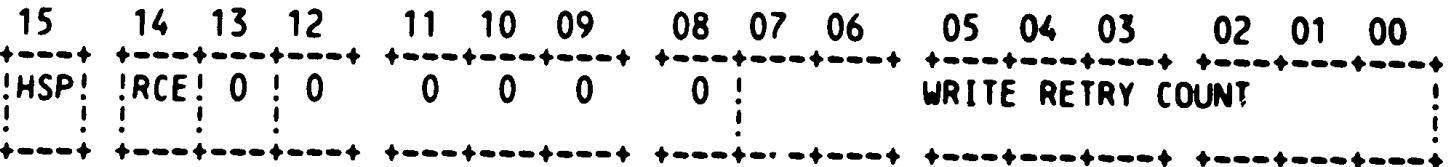
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828

6.3.6 EXTENDED STATUS REGISTER 3 (XSTAT3)



BIT	NAME	TCC	DEFINITION
15 TO 08			MICRO DIAGNOSTIC ERROR CODE. (SEE LIST OF CODES BELOW).
07			RESERVED (ALWAYS A 0)
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	-	-	RESERVED (ALWAYS A 0)
03	DCK	S,6	DENSITY CHECK. SET WHEN A PE IDENTIFICATION BURST (IDB) WAS NOT DETECTED WHEN MOVING OFF OF BOT.
02-01			RESERVED (ALWAYS A 0)
00	RIB	2	REVERSE INTO BOT. A READ, SPACE, OR SKIP COMMAND ALREADY IN PROGRESS HAS ENCOUNTERED THE BOT MARKER WHEN MOVING TAPE IN THE REVERSE DIRECTION. TAPE MOTION WILL BE HALTED AT BOT.

6.3.7 EXTENDED STATUS REGISTER 4 (XSTAT4)



BIT	NAME	TCC	DEFINITION
-----	------	-----	------------

1829	15	HSP	S	HIGH SPEED. WHEN SET, INDICATES THAT THE TRANSPORT IS
1830				OPERATING IN HIGH SPEED MODE.(100IPS) WHEN CLEAR, THE
1831				TRANSPORT IS OPERATING IN LOW SPEED MODE.(25IPS)
1832				
1833	14	RCE	6	RETRY COUNT EXCEEDED. WHEN SET, INDICATES THAT THE CONTROLLER
1834				WAS BUFFERING WRITE DATA AND COULD NOT SUCESSFULLY OUTPUT
1835				THE BUFFERED RECORD WITHIN THE SPECIFIED NUMBER OF RETRIES.
1836				CAUSES TAPE POSITION LOST TERMINATION.
1837				
1838	13-8		-	RESERVED (ALWAYS A 0)
1839				
1840	7-0	WRC	S	WRITE RECOUNT COUNT STATISTIC. THIS FIELD INDICATES, WHEN
1841				THE CONTROLLER IS BUFFERING WRITE DATA RECORDS, THE TOTAL
1842				NUMBER OF CONTROLLER INITIATED RETRIES PERFORMED IN ORDER
1843				TO WRITE THE PREVIOUS BUFFERED RECORD. THIS COUNT IS CLEARED
1844				AFTER IT IS DISPLAYED.
1845				
1846				
1847				

7.0 DIAGNOSTIC HISTORY

REVISION A - MAR 1982

- MODIFIED CZTSHC FROM TS11 FOR TSV05

1855

```

12      .TITLE PROGRAM HEADER AND TABLES
13      .SBTTL PROGRAM HEADER
42
44      .ENABL ABS,AMA
45      =      2000
47 002000 BGNMOD
48
49      :++
50      : THE PROGRAM HEADER IS THE INTERFACE BETWEEN
51      : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
52      :--
53
54 002000      POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU
55
63
64 002000      HEADER CVTSE,A,0,5000,1
002000      L$NAME::      ;DIAGNOSTIC NAME
002000      103
002001      126      .ASCII /C/
002002      124      .ASCII /V/
002003      123      .ASCII /T/
002004      105      .ASCII /S/
002005      000      .ASCII /E/
002006      000      .BYTE 0
002007      000      .BYTE 0
002010      L$REV::      ;REVISION LEVEL
002010      101
002011      L$DEPO::      ;0
002011      060      .ASCII /A/
002012      L$UNIT::      ;NUMBER OF UNITS
002012      000000      .ASCII /O/
002014      L$TIML::      ;LONGEST TEST TIME
002014      005000      .WORD 0
002016      L$HPCP::      ;POINTER TO H.W. QUES.
002016      027732      .WORD 5000
002020      L$SPCP::      ;POINTER TO S.W. QUES.
002020      030040      .WORD L$HARD
002022      L$HPTP::      ;PTR. TO DEF. H.W. PTABLE
002022      002174      .WORD L$SOFT
002024      L$SPTP::      ;PTR. TO S.W. PTABLE
002024      002204      .WORD L$HW
002026      L$LADP::      ;DIAG. END ADDRESS
002026      032004      .WORD L$SW
002030      L$STA::      ;RESERVED FOR APT STATS
002030      000000      .WORD L$LAST
002032      L$CO::      .WORD 0
002032      000000      .WORD 0
002034      L$DTYP::      ;DIAGNOSTIC TYPE
002034      000001      .WORD 0
002036      L$APT::      ;APT EXPANSION
002036      000000      .WORD 1
002040      L$DTP::      ;PTR. TO DISPATCH TABLE
002040      002124      .WORD 0
002042      L$PRIO::      ;DIAGNOSTIC RUN PRIORITY
002042      000000      .WORD L$DISPATCH
002044      L$ENVI::      ;FLAGS DESCRIBE HOW IT WAS SETUP
002044      000000      .WORD 0
002044      000000      .WORD 0

```

002046		L\$EXP1::	;EXPANSION WORD		
002046	000000			.WORD	0
002050		L\$MREV::	;SVC REV AND EDIT #		
002050	003			.BYTE	C\$REVISION
002051	003			.BYTE	C\$EDIT
002052		L\$EF::	;DIAG. EVENT FLAGS		
002052	000000			.WORD	0
002054	000000			.WORD	0
002056		L\$SPC::			
002056	000000			.WORD	0
002060		L\$DEVP::	; POINTER TO DEVICE TYPE LIST		
002060	002164			.WORD	L\$DVTYP
002062		L\$REPP::	;PTR. TO REPORT CODE		
002062	017546			.WORD	L\$RPT
002064		L\$EXP4::			
002064	000000			.WORD	0
002066		L\$EXP5::			
002066	000000			.WORD	0
002070		L\$AUT::	;PTR. TO ADD UNIT CODE		
002070	024030			.WORD	L\$AU
002072		L\$DUT::	;PTR. TO DROP UNIT CODE		
002072	023756			.WORD	L\$DU
002074		L\$LUN::	;LUN FOR EXERCISERS TO FILL		
002074	000000			.WORD	0
002076		L\$DESP::	;POINTER TO DIAG. DESCRIPTION		
002076	002136			.WORD	L\$DESC
002100		L\$LOAD::	;GENERATE SPECIAL AUTOLOAD EMT		
002100	104035			EMT	E\$LOAD
002102		L\$ETP::	;POINTER TO ERR_TBL		
002102	000000			.WORD	0
002104		L\$ICP::	;PTR. TO INIT CODE		
002104	021302			.WORD	L\$INIT
002106		L\$CCP::	;PTR. TO CLEAN-UP CODE		
002106	023714			.WORD	L\$CLEAN
002110		L\$ACP::	;PTR. TO AUTO CODE		
002110	023272			.WORD	L\$AUTO
002112		L\$PRT::	;PTR. TO PROTECT TABLE		
002112	021274			.WORD	L\$PROT
002114		L\$TEST::	;TEST NUMBER		
002114	000000			.WORD	0
002116		L\$DLY::	;DELAY COUNT		
002116	000000			.WORD	0
002120		L\$HIME::	;PTR. TO HIGH MEM		
002120	000000			.WORD	0

```

72      .SBTTL DISPATCH TABLE
73
74      :++
75      : THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
76      : IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
77      :--
78
79      DISPATCH 5
      002122      000005
      002122      000005
      002124      024134
      002126      025622
      002130      026456
      002132      026652
      002134      027032
      .WORD      5
      L$DISPATCH::
      .WORD      T1
      .WORD      T2
      .WORD      T3
      .WORD      T4
      .WORD      T5

80
87
88      .SBTTL DESCRIPTIVE TEXT
89
90      :++
91      : 2 LINES OF TEXT PRINTED TO THE OPERATOR TO IDENTIFY THE DIAGNOSTIC AND THE DEVICE UNDER TES
92      :--
93
94      DESCRIPT      <DATA RELIABILITY TEST>
      002136
      002136      104      101      124
      002136      101      040      122
      002141      101      040      122
      002144      105      114      111
      002147      101      102      111
      002152      114      111      124
      002155      131      040      124
      002160      105      123      124
      002163      000
      .ASCIZ      /DATA RELIABILITY TE

95      DEV TYP      <TSV05>
      002164
      002164      124      123      126
      002164      124      123      126
      002167      060      065      000
      L$DVTYP::
      .ASCIZ      /TSV05/
      .EVEN

```

PROGRAM HEADER AND TABLES
DEFAULT HARDWARE P-TABLE

MACRO M1113 25-MAY-82 09:51 PAGE 18

SEQ 0043

97
98
99
100
101
102
103
104
105 002172
002172 000003
002174
002174
106
107
108 002174 172520
109 002176 000224
110 002200 000000
111
112 002202
002202

.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::

.WORD L10000-L\$HW/2

.WORD 172520 ;TSDB ADDRESS.
.WORD 224 ;VECTOR ADDRESS.
.WORD 0 ;DRIVE #0 FOR DEFAULT

ENDHW
L10000:


```
114 .SBTTL SOFTWARE P-TABLE
115
116
117 :++
118 : THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
119 : PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
120 :--
121 002202          BGNSW  SFPTBL
      002202 000051
      002204
      002204
122
129 002204 001
130 002205 000
131 002206 000
132 002207 000
133 002210 001
134 002211 000
135 002212 000
136 002213 000
137 002214 000
138 002215 000
139 002216 000
140 002217 000
141 002220 000040
142 002222 000015
143 002224 000001
144 002226 000001
145 002230 000007
146 002232 000004
147 002234 004000
148 002236 076400
149 002240 000007
150 002242 000003
151 002244 004000
152 002246 076400
153 002250 000007
154 002252 000002
155 002254 004000
156 002256 076400
157 002260 000007
158 002262 000015
159 002264 000001
160 002266 000001
161 002270 000007
162 002272 000033
163 002274 004000
164 002276 076400
165 002300 000007
166 002302 000033
167 002304 004000
168 002306 076400
169 002310 000007
170 002312 000001
171 002314 000000
172 002316 000000
173 002320 000000

      L$SW::
      SFPTBL::

      CLRFLG:: .BYTE 1
      RRANV:: .BYTE 0
      HAE:: .BYTE 0
      ERCVER:: .BYTE 0
      BADTSW:: .BYTE 1
      DINT:: .BYTE 0
      IREC:: .BYTE 0
      CHGFLG:: .BYTE 0
      PIRE:: .BYTE 0
      CHAR:: CH.EAI
      CMDD:: .WORD 13.
      .WORD 1
      .WORD 1
      .WORD RANP
      .WORD 4
      .WORD DATCNT
      .WORD 32000.
      .WORD RANP
      .WORD 3
      .WORD DATCNT
      .WORD 32000.
      .WORD RANP
      .WORD 2
      .WORD DATCNT
      .WORD 32000.
      .WORD RANP
      .WORD 13.
      .WORD 1
      .WORD 1
      .WORD RANP
      .WORD 27.
      .WORD DATCNT
      .WORD 32000.
      .WORD RANP
      .WORD 27.
      .WORD DATCNT
      .WORD 32000.
      .WORD RANP
      TS1MD:: .WORD 1
      RDBUF:: .WORD 0
      WTBUF:: .WORD 0
      HSSW:: .WORD 0

      .WORD L10001-L$SW/2

      ;CLEAR COUNTERS FLAG.
      ;RESET RANDOM VARIABLES EACH PASS FLAG.
      ;HALT AFTER EACH COMMAND FLAG.
      ;ENABLE RECOVERABLE ERROR PRINTS FLAG.
      ;BAD TAPE SWITCH TO REWRITE ON SAME SPOT & DETECT BAD TAPE
      ;SPARE
      ;DISABLE INTERRUPTS FLAG.
      ;INHIBIT ERROR RECOVERY FLAG.
      ;CHANGE CMD SEQ TABLE FLAG.
      ;SPARE.
      ;INHIBIT RESIDUAL FRAMECOUNT ERROR REPORT FLAG.
      ;SPARE.
      ;CHARACTERISTICS CODE (DEFAULT = 40).
      ;COMMAND 2 (DEFAULT = REWIND).
      ;BYTE COUNT
      ;NUMBER OF OPERATIONS
      ;PATTERN
      ;COMMAND 3 (DEFAULT = WRITE)
      ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
      ;NUMBER OF OPERATIONS (DEFAULT = 32000).
      ;PATTERN (DEFAULT = RANDOM).
      ;COMMAND 4 (DEFAULT = READ REV).
      ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
      ;NUMBER OF OPERATIONS (DEFAULT = 32,000).
      ;PATTERN (DEFAULT = RANDOM).
      ;COMMAND 5 (DEFAULT = READ FWD).
      ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
      ;NUMBER OF OPERATIONS (DEFAULT = 32,000).
      ;PATTERN (DEFAULT = RANDOM).
      ;COMMAND 6 (DEFAULT = REWIND).
      ;BYTE COUNT
      ;NUMBER OF OPERATIONS
      ;PATTERN
      ;END OF CMD SEQ TABLE CODE (DEF) OR CMD 7
      ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
      ;NUMBER OF OPERATIONS (DEFAULT = 32000).
      ;PATTERN (DEFAULT = RANDOM).
      ;END OF CMD SEQ TABLE CODE (DEF) OR CMD 8
      ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
      ;NUMBER OF OPERATIONS (DEFAULT = 32000).
      ;PATTERN (DEFAULT = RANDOM).
      ;DEFAULT SWITCH SETTING
      ;ENABLE READ BUFFERING
      ;ENABLE WRITE BUFFERING
      ;RUN AT 100IPS SWITCH
```

PROGRAM HEADER AND TABLES
SOFTWARE P-TABLE

MACRO M1113 25-MAY-82 09:51 PAGE 19-1

SEQ 0045

174 002322 000000
175 002324 000000
176
177 002326
002326
178
179 002326

EXTFEA::WORD 0
BENBSW::WORD 0

;EXTENDED FEATURES SOFTWARE SW 0=OFF;1=ON
;BUFFER ENABLE SOFTWARE SW 0=OFF;1=ON

ENDSW
L10001:

ENDMOD

192
193
194
203
204 002326
205
206
207
208
209
210
211 002326.TITLE GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

BGNMOD

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

EQUALS

: BIT DIFINITIONS

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

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

: EVENT FLAG DEFINITIONS

: EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

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

: PRIORITY LEVEL DEFINITIONS

000340	PRI07== 340
000300	PRI06== 300

```

000240      PRI05== 240
000200      PRI04== 200
000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0
;
; OPERATOR FLAG BITS
;
000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      HOE== 100000

```

; REGISTER USAGE.

```

;
; R0 - PASSES PARAMETERS TO/FROM DIAGNOSTIC SUPERVISOR.
; R1 - COMMAND SEQUENCE TABLE POINTER.
; R2 - GENERAL PURPOSE REGISTER.
; R3 - GENERAL PURPOSE REGISTER.
; R4 - GENERAL PURPOSE REGISTER.
; R5 - CURRENT LOGICAL DEVICE NUMBER X 2.
; R6 - STACK POINTER.
; R7 - PROGRAM COUNTER.

```

; THE FOLLOWING ARE BIT DEFINITIONS FOR THE TSSR REGISTERS.

```

100000      TS.SC==100000      ;SPECIAL CONDITION BIT.
040000      TS.UPE==40000      ;UNIBUS PARITY ERROR
020000      TS.SPE==20000      ;SERIAL BUS PARITY ERROR.
010000      TS.RMR==10000      ;REGISTER MODIFICATION REFUSED.
004000      TS.NXM==4000       ;NON-EXISTENT MEMORY.
002000      TS.NBA==2000       ;NEED BUFFER ADDRESS.
001000      TS.A17==1000       ;BUS ADDRESS BIT 17.
000400      TS.A16==400        ;BUS ADDRESS BIT 16.
000200      TS.SSR==200        ;UNIT READY BIT.
000100      TS.OFL==100        ;OFF LINE.
177717      TSC.FCC==177717    ;FATAL CLASS CODE MASK.
177761      TSC.TCC==177761    ;TERMINATION CLASS CODE MASK.

```

; THE FOLLOWING ARE BIT DEFINITIONS FOR THE COMMAND WORD

```

100000      ACK.C==100000      ;ACKNOWLEDGE BIT
040000      CVC.C==40000      ;CLEAR VOLUME CHECK.
020000      OPP.C==20000      ;OPPOSITE BIT
010000      SWB.C==10000      ;SWAP BYTE BIT

```

212
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252

253	0G4000	MOD.C3==4000	:MODE BIT 3
254	004000	BRF.C==4000	:BYTE/RECORD/FILE COUNT FLAG BIT. NOT USED
255			:BY TS05 BUT USED INTERNALLY BY THIS PROGRAM ONLY.
256	002000	MOD.C2==2000	:MODE BIT 2
257	001000	MOD.C1==1000	:MODE BIT 1
258	000400	MOD.C0==400	:MODE BIT 0
259	000200	IE.C==200	:INTERRUPT ENABLE
260	000100	FMT.C1==100	:FORMAT BIT 1
261	000100	VFY.C==100	:WRITE VERIFY FLAG BIT. INTERNAL USE ONLY.
262			:NOT USED BY TS05.
263	000040	FMT.C0==40	:FORMAT BIT 0.
264	000040	JMP.C==40	:JUMP BIT-TO DIRECT THIS PROGRAM TO JUMP TO
265			:A CERTAIN LOCATION IN THE COMMAND SEQUENCE
266			:TABLE. INTERNAL USE ONLY.
267	000020	CMD.C4==20	:COMMAND BIT 4
268	000020	DLY.C==20	:INSERT DELAY. INTERNAL USE ONLY.
269	000010	CMD.C3==10	:COMMAND BIT 3
270	000004	CMD.C2==4	:COMMAND BIT 2
271	000002	CMD.C1==2	:COMMAND BIT 1
272	000001	CMD.C0==1	:COMMAND BIT 0
273			
274			:BIT DEFINITIONS FOR DEVICE CHARACTERISTICS.
275			
276	000200	CH.ESS==200	:ENABLE SKIP TAPE MARKS STOP (STOP AT LOGICAL EOT).
277	000040	CH.EAI==40	:ENABLE ATTENTION INTERRUPTS.
278	000020	CH.ERI==20	:ENABLE MESSAGE BUFFER RELEASE INTERRUPTS.
279	000040	DFTSCH==CH.EAI	:DEFAULT CHARACTERISTICS CODE.
280			
281			:BIT DEFINITIONS FOR EXTENDED CONTROL WORD
282			
283	000040	EF.HSS==40	:ENABLE HIGH SPEED SELECT
284	000030	EF.RWB==30	:ENABLE BOTH READ & WRITE BUFFERING
285	000020	EF.RBO==20	:ENABLE READ BUFFERING ONLY
286			
287			:THE FOLLOWING INDICATES THE RELATIVE POSITIONS OF THE STATUS WORDS
288			:IN THE MESSAGE BUFFER.
289			
290	000004	MS.RFC==4	:RESIDUAL FRAME COUNT.
291	000006	MS.XS0==6	:EXT STATUS REG 0
292	000010	MS.XS1==10	:EXT STATUS REG 1
293	000012	MS.XS2==12	:EXT STATUS REG 2
294	000014	MS.XS3==14	:EXT STATUS REG 3
295	000016	MS.XS4==16	:EXT STATUS REG 4
296			
297			:THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0.
298			
299	100000	X0.TMK==100000	:TAPE MARK.
300	040000	X0.RLS==40000	:RECORD LENGTH SHORT.
301	020000	X0.LET==20000	:LOGICAL EOT.
302	010000	X0.RLL==10000	:RECORD LENGTH LONG.
303	000100	X0.ONL==100	:ON LINE BIT.
304	000004	X0.WLK==4	:WRITE LOCK BIT
305	000002	X0.BOT==2	:BOT BIT.
306	000001	X0.EOT==1	:EOT BIT.
307			
308			:THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2.
309			

```

310      100000      X2.OPM==100000      ;OPERATION IN PROGRESS, TAPE MOVING
311      000200      X2.EFE==200        ;EXTENDED FEATURES ENABLED
312      000100      X2.BFE==100        ;BUFFERING ENABLED
313
314      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3.
315
316      000010      X3.DCK==10          ;DENSITY CHECK.
317      157400      X3.RNY==157400      ;CAPSTAN RUNAWAY UDIAG ERROR CODE.
318
319      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4.
320
321      100000      X4.HSS==100000      ;HIGH SPEED SWITCH INDICATING 100IPS
322      040000      X4.RCE== 40000      ;RETRY COUNT EXCEEDED
323
324
325      ;THE FOLLOWING DEFINITIONS SHOW THE RELATIVE POSITIONS OF THE COMMAND
326      ;PACKET ENTRIES.
327
328      000000      CP.CMD==0            ;CMDPKT+0==TS05 COMMAND.
329      000002      CP.ADL==2           ;CMDPKT+2==BUFFER ADDRESS LOW.
330      000004      CP.ADH==4           ;CMDPKT+4==BUFFER ADDRESS HIGH.
331      000006      CP.CNT==6           ;CKDPKT+6==BYTE/FILE/RECORD COUNT
332
333      ;MISCELLANEOUS DEFINITIONS.
334
335      000340      INTPRI==PRI07        ;PRIORITY TO BE USED IN INTERRUPT STATE.
336      000012      SCHCNT==12          ;ARBITRARY BYTE LENGTH FOR CHARACTERISTIC
337                                     ;BUFFER LENGTH. (EVEN #)
338      000020      MSGCNT==20          ;MESSAGE BUFFER LENGTH IN BYTES. (EVEN #)
339      000020      DIACNT==20          ;DIAGNOSTIC COMMAND BUFFER EXTENT.
340      004000      DATCNT==2048.       ;MAXIMUM RECORD LENGTH IN BYTES.
341                                     ;THIS COUNT SHOULD BE A MULTIPLE OF 256 TO INSURE
342                                     ;PROPER READ/WRITE BUFFER ALLOCATION BY THE SUPER.
343      177740      RNOPSC==177740      ;RANDOM # OF OPERATIONS MASK.
344      000007      RANP==7             ;CODE TO SELECT RANDOM PATTERN.
345      000020      RRECL==16.          ;READ RECOVERY ATTEMPT LIMIT.
346      000020      WRECL==16.          ;WRITE RECOVERY ATTEMPT LIMIT.
347      153624      RANBC==153624      ;CONSTANT USED TO RESET RANDOM # GENERATOR BASE.
348      032561      RANSC==32561        ;CONSTANT USED TO RESET RANDOM # SAVE LOCATION.
349      177774      NINUSE==177774     ;NOT IN USE CODE FOR DEVICE STATE TABLE.
350      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
351                                     ;NOT "COMMAND" BITS.
352
353      ;THE FOLLOWING DEFINES THE COMMAND WORD FOR EACH TS05 COMMAND.
354
355      100013      DRI==  ACK.C!CMD.C3!CMD.C1!CMD.CO      ;DRIVE INIT.
356
357      104001      RDF==  ACK.C!BRF.C!CMD.CO              ;READ FORWARD
358
359      104401      RDR==  ACK.C!BRF.C!MOD.CO!CMD.CO        ;READ REVERSE
360
361      104005      WRT==  ACK.C!BRF.C!CMD.CO!CMD.C2        ;WRITE COMMAND
362
363
364
365
366

```

367	104105	WTV==	ACK.C!BRF.C!VFY.C!CMD.CO!CMD.C2	
368				;WRITE VERIFY
369				
370	104010	SRF==	ACK.C!BRF.C!CMD.C3	
371				;SPACE RECORD FORWARD
372				
373	104410	SRR==	ACK.C!BRF.C!MOD.CO!CMD.C3	
374				;SPACE RECORD REVERSE
375				
376	105401	RNR==	ACK.C!BRF.C!MOD.C1!MOD.CO!CMD.CO	
377				;READ REV RETRY1 - REREAD NEXT REVERSE, IE. SPACE FWD, READ REVERSE
378				
379	125401	RNF==	ACK.C!BRF.C!OPP.C!MOD.C1!MOD.CO!CMD.CO	
380				;READ REV RETRY2 - REREAD NEXT FORWARD, IE. READ FORWARD, SPACE REVERSE
381				
382	105001	RPF==	ACK.C!BRF.C!MOD.C1!CMD.CO	
383				;READ FWD RETRY1 - REREAD PREVIOUS FORWARD, IE. SPACE REVERSE, READ FORWARD
384				
385	125001	RPR==	ACK.C!BRF.C!OPP.C!MOD.C1!CMD.CO	
386				;READ FWD RETRY2 - REREAD PREVIOUS REVERSE, IE. READ REVERSE, SPACE FORWARD
387				
388	105005	WRR==	ACK.C!MOD.C1!BRF.C!CMD.C2!CMD.CO	
389				;WRITE RETRY
390				
391	102010	RWD==	ACK.C!MOD.C2!CMD.C3	
392				;REWIND COMMAND
393				
394	100012	MBR==	ACK.C!CMD.C3!CMD.C1	
395				;MESSAGE BUFFER RELEASE
396				
397	100011	WTM==	ACK.C!CMD.C3!CMD.CO	
398				;WRITE TAPE MARK.
399				
400	101011	WTR==	ACK.C!MOD.C1!CMD.C3!CMD.CO	
401				;WRITE TAPE MARK RETRY.
402				
403	105010	SFF==	ACK.C!BRF.C!MOD.C1!CMD.C3	
404				;SPACE FILE FORWARD
405				
406	105410	SFR==	ACK.C!BRF.C!MOD.CO!MOD.C1!CMD.C3	
407				;SPACE FILE REVERSE
408				
409	100017	GES==	ACK.C!CMD.CO!CMD.C1!CMD.C2!CMD.C3	
410				;GET EXTENDED STATUS
411				
412	100411	ERS==	ACK.C!MOD.CO!CMD.C3!CMD.CO	
413				;ERASE 3 INCHES OF TAPE
414				
415	100412	UNL==	ACK.C!MOD.CO!CMD.C3!CMD.C1	
416				;UNLOAD COMMAND
417				
418	101012	CLN==	ACK.C!MOD.C1!CMD.C3!CMD.C1	
419				;ERASE TAPE.
420				
421	140004	SCH==	ACK.C!CVC.C!CMD.C2	;SET DEVICE CHARACTERISTICS.
422				
423	140006	WSM==	ACK.C!CVC.C!CMD.C2!CMD.C1	;WRITE SUB-SYS MEM

```

424
425      100006      DIA==  ACK.C!CMD.C2!CMD.C1      ;DIAGNOSTICS.
426
427      000040      JMP==  JMP.C                      ;JUMP TO 'N'TH COMMAND
428
429      000020      DLY==  DLY.C                      ;DELAY 'N' MS.
430
431      177777      END==  177777                    ;END OF COMMAND SEQUENCES
432
433      .SBTTL  GLOBAL DATA SECTION
434      :++
435      : THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
436      : IN MORE THAN ONE TEST.
437      :--
438
439
440
441      :          COMMAND PACKET.
442
443      :          =          <.+3>&177774      ;MUST BE ON MOD 4 BOUNDRY.
444      CNDPKT:: 0          ;1ST WORD IS TS05 COMMAND.
445      0          ;2ND WORD IS THE BUFFER LOW ADDRESS.
446      0          ;3RD WORD IS THE BUFFER HIGH ADDRESS.
447      0          ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
448
449
450      :          GET STATUS COMMAND PACKET.
451
452      :          =          <.+3>&177774      ;MUST BE ON MOD 4 BOUNDRY.
453      GSCPK:: .WORD  GES
454
455
456      :          MESSAGE BUFFER RELEASE COMMAND PACKET.
457
458      :          =          <.+3>&177774      ;MUST BE ON MOD 4 BOUNDRY.
459      BRCPK:: .WORD  MBR
460
461
462
463
464      :          REWIND COMMAND PACKET (USED IN ERROR RECOVERY ONLY)
465
466      :          =          <.+3>&177774      ;MUST BE ON MOD 4 BOUNDRY.
467      RWCpk:: .WORD  RWD
468      .WORD  1
469
470
471
472      :          WORK AREA FOR ANALYSIS OF MESSAGE PACKET CONTENTS.
473
474      MSGPKT:: .BLKW  8.      ;1ST WORD:: MESSAGE TYPE.
475      ;2ND WORD:: DATA FIELD LENGTH.
476      ;3RD WORD:: RESIDUAL FRAME COUNT.
477      ;4TH WORD:: XSTAT0
478      ;5TH WORD:: XSTAT1
479      ;6TH WORD:: XSTAT2
480      ;7TH WORD:: XSTAT3

```



```
GLOBAL DATA SECTION

481
482
483
484 002374 MSGPK0:: .BLKW 8. ;MESSAGE PACKET FOR DEVICE #0
485 002414 MSGPK1:: .BLKW 8. ;MESSAGE PACKET FOR DEVICE #1
486 002434 MSGPK2:: .BLKW 8. ;MESSAGE PACKET FOR DEVICE #2
487 002454 MSGPK3:: .BLKW 8. ;MESSAGE PACKET FOR DEVICE #3
488
489
490 ; SET CHARACTERISTIC BLOCK.
491 002474 002374 SCHBK:: MSGPK0 ;1ST WORD:: MSGPKT ADDR LO(SET UP BY EXECUTE ROUTINE).
492 002476 000000 0 ;2ND WORD:: MSGPKT ADDR HI.
493 002500 000020 MSGCNT ;3RD WORD:: MSG BUFFER LENGTH (BYTES)
494 002502 000040 CH.EAI ;4TH WORD:: CHARACTERISTICS WORD(SET BY SETUP ROUTINE).
495 002504 000000 0 ;5TH WORD:: HSP & BUFFER CONTROL ON EXT'D FEATURES
496
497
498 ; WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
499
500 002506 000000 WSMBK:: 0 ;1ST WORD:: SEL 0
501 002510 000000 0 ;2ND WORD:: SEL 2
502 002512 000000 0 ;3RD WORD:: SEL 4
503
504 ; TS05 REGISTER ADDRESSES.
505
506 002514 TSDB:: .BLKW 4 ;TS05 DATA BUFFER ADDRESSES.
507 002524 TSSR:: .BLKW 4 ;TS05 STATUS REGISTER ADDRESSES.
508 002534 TSVCT:: .BLKW 4 ;TS05 VECTOR ADDRESSES.
509 002514 TSBA==TSDB ;DATA BUFFER ADDRESS REGISTER.
510
511
512 ; ADDRESSES OF MESSAGE PACKETS.
513
514 002544 002374 MSGPKA:: MSGPK0 ;DEVICE 0.
515 002546 002414 MSGPK1 ;DEVICE 1.
516 002550 002434 MSGPK2 ;DEVICE 2.
517 002552 002454 MSGPK3 ;DEVICE 3.
518
519 ; ADDRESSES OF INTERRUPT HANDLING ROUTINES.
520
521 002554 010034 TSSINT:: TSSIN0 ;DEVICE 0.
522 002556 010042 TSSIN1 ;DEVICE 1.
523 002560 010050 TSSIN2 ;DEVICE 2.
524 002562 010056 TSSIN3 ;DEVICE 3.
525
526 ; TS05 CODE LEVELS, WILL BE STORED AFTER SCH CMD IN BASIC FUNCTION TEST
527
528 002564 000000 TSSCL:: 0 ;DEVICE 0
529 002566 000000 0 ;DEVICE 1
530 002570 000000 0 ;DEVICE 2
531 002572 000000 0 ;DEVICE 3
532
533 ; TS05 EXT. FEA & BUF. ENA SW'S, WILL BE STORED AFTER SCH CMD IN BASIC FUNCTION TEST
534
535 002574 000000 TSSSW:: 0 ;DEVICE 0
536 002576 000000 0 ;DEVICE 1
537 002600 000000 0 ;DEVICE 2
```

```
GLOBAL DATA SECTION

538 002602 000000          0          ;DEVICE 3
539
540      :      UNIT NUMBERS OF ALL DEVICES BEING TESTED(1-4).
541      :      WHEN DEVICE IS NOT IN USE, IT'S LOCATION WILL = -3.
542      :      R5 WILL ALWAYS CONTAIN THE PRESENT LOGICAL UNIT NUMBER X 2.
543
544 002604 177774      DEVTBL:: .WORD  NINUSE
545 002606 177774          .WORD  NINUSE
546 002610 177774          .WORD  NINUSE
547 002612 177774          .WORD  NINUSE
548 002614 177777          .WORD  END
549
550
551      :      BAD TAPE TABLE POINTER: USED BY WRITE RETRY ROUTINE
552      :      'WRTY' TO LOG BAD TAPE SPOTS ON UNITS UNDER TEST
553
554 002616 003046      BTADDR:: BT0
555 002620 003120          BT1
556 002622 003172          BT2
557 002624 003244          BT3
558      :      COUNT .R AREA.
559
560      002626      CNTBGN=.
561 002626      WRBC:: .BLKW  20          ;BYTES WRITTEN.
562 002666      RRBC:: .BLKW  20          ;BYTES READ REV.
563 002726      RFBC:: .BLKW  20          ;BYTES READ FWD.
564 002766      WRREC:: .BLKW  4          ;RECOVERABLE WRITE ERRORS.
565 002776      WRUNR:: .BLKW  4          ;UNRECOVERABLE WRITE ERRORS.
566 003006      RRREC:: .BLKW  4          ;RECOVERABLE READ REV ERRORS.
567 003016      RRUNR:: .BLKW  4          ;UNRECOVERABLE READ REV ERRORS.
568 003026      RFREC:: .BLKW  4          ;RECOVERABLE READ FWD ERRORS.
569 003036      RFUNR:: .BLKW  4          ;UNRECOVERABLE READ FWD ERRORS.
570 003046      BT0:: .BLKW  21.          ;UNIT 0 BAT TAPE SPOTS LOG
571 003120      BT1:: .BLKW  21.          ;UNIT 1 BAT TAPE SPOTS LOG
572 003172      BT2:: .BLKW  21.          ;UNIT 2 BAT TAPE SPOTS LOG
573 003244      BT3:: .BLKW  21.          ;UNIT 3 BAT TAPE SPOTS LOG
574 003316      WRTYCT:: .BLKW  4          ;WRITE RETRY COUNTER
575 003326      PASCNT:: .BLKW  4          ;PASS COUNT.
576 003336      SCCNT:: .BLKW  4          ;SPECIAL CONDITION COUNT.
577 003346      VFYCNT:: .BLKW  4          ;COUNT OF TS05 DATA COMPARE ERRORS.
578 003356      HRDCNT:: .BLKW  4          ;COUNT OF HARD ERRORS.
579 003366      FTLCNT:: .BLKW  4          ;COUNT OF FATAL ERRORS.
580      003376      CNTEND=.          ;END OF STATISTICAL COUNTERS.
581 003376      RECCNT:: .BLKW  4          ;NUMBER OF RECORDS FROM BOT: CLEARED ON REWIND
582      000550          ;AND WHEN RESTARTING OR CONTINUING TEST 2.
583      CNTLEN==CNTEND-CNTBGN          ;LENGTH OF STATISTICAL COUNTER AREA.
584
585
586      :      THE FOLLOWING ARE THE DEFINITIONS OF VARIABLES
587      :      USED BY THE PROGRAM.
588
589 003406 000000      DATAWT:: .WORD  0          ;WRITE BUFFER ADDRESS.
590      003406      DIABLK==DATAWT          ;WRITE BUFFER ALSO USED FOR DIAG CMD.
591 003410 000000      DATARD:: .WORD  0          ;READ BUFFER ADDRESS.
592 003412 000000      NCNT:: .WORD  0          ;STORAGE FOR VALUE OF N.
593 003414 000000      NCNT1:: .WORD  0          ;TEMP STORAGE FOR VALUE OF N.
594 003416 000000      BRFCNT:: .WORD  0          ;STORAGE FOR BPCR VALUE.
```

```
GLOBAL DATA SECTION

595 003420 177777 CMDWRD:: .WORD END ;CONTAINS COMMAND WORD BEING EXECUTED PRESENTLY.
596 003422 177777 CMDSAV:: .WORD END ;SAVE LOCATION FOR CMD WORD DURING ERROR RECOVERY
597 003424 177777 PCMDWD:: .WORD END ;CONTAINS PREVIOUS COMMAND WORD.
598 003426 000000 CMDLG:: .WORD 0 ;CURRENT COMMAND LOGGING CODE.
599 003430 000000 LENMSK:: .WORD 0 ;RANDOM WRITE LENGTH MASK, TO BE SET UP BY TESTS
600 003432 153624 RANB:: .WORD 153624 ;RANDOM # GENERATOR BASE.
601 003434 032561 RANS:: .WORD 32561 ;RANDOM # SAVE LOCATION.
602 003436 000000 TIME1:: .WORD 0 ;TIME COUNT 1.
603 003440 000000 TIME2:: .WORD 0 ;TIME COUNT 2.
604 003442 000000 JLOOP:: .WORD 0 ;JMP COMMAND LOOP COUNT.
605 003444 000000 JLOC:: .WORD 0 ;JMP COMMAND LOCATION COUNT.
606 003446 000000 PATERN:: .WORD 0 ;PATTERN SELECT CODE.
607 003450 000000 CTCC:: .WORD 0 ;CURRENT TERMINATION CLASS CODE.
608 003452 000000 RSSAVE:: .WORD 0 ;LOCATION FOR SAVING CURRENT DEVICE POINTER.
609 003454 000000 TSSREG:: .WORD 0 ;CURRENT STATUS REGISTER.
610 003456 000000 WTMFLG:: .WORD 0 ;WRITE TAPE MARK FLAG
611
612 ;
613 ; ERROR FLAG AREA, THESE FLAGS ARE CLEARED DURING INITIALIZATION AND
614 ; AFTER EACH COMMAND IS COMPLETED.
615
616 003460 000000 BGNFLG=.
617 003462 000 RETRYC:: .WORD 0 ;# OF RECOVERY ATTEMPTS EXECUTED.
618 003463 000 RPTCNT:: .BYTE 0 ;WRITE REPEAT ON SAME SPOT CNTR: 4 PER WRITE RETRY
619 003464 000 WRTYFG:: .BYTE 0 ;WRITE RETRY ON SAME SPOT IN PROGRESS FLAG
620 003465 000 WRTYER:: .BYTE 0 ;WRITE RETRY ON SAME SPOT ERROR FLAG
621 003466 000 RECLOG:: .BYTE 0 ;RECORD COUNT HAS BEEN UPDATED FOR THIS RECORD.
622 003467 000 ERLOG:: .BYTE 0 ;DATA BYTES AND ERRORS HAVE BEEN LOGGED FOR THIS RECORD.
623 003470 000 RWERR:: .BYTE 0 ;READ/WRITE ERROR HAS OCCURED.
624 003471 000 UNREC:: .BYTE 0 ;UNRECOVERABLE ERROR HAS OCCURED.
625 .EVEN ERRREC:: .BYTE 0 ;ERROR RECOVERY MODE.
626 003472 ENDERF=.
627
628 ;
629 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED DURING INITIALIZATION.
630
631 003472 INTFLG:: .BLKW 4 ;INTERRUPT OCCURRED FLAGS FOR EACH DEVICE.
632 003502 EOTFLG:: .BLKW 4 ;EOT/BOT FLAGS FOR EACH DEVICE (XSTAT0).
633 003512 000000 BTPT:: .WORD 0 ;BAD TAPE SPOT POINTER TO BT0-BT3 VIA BTADDR
634 003514 000 EXPBOT:: .BYTE 0 ;BOT IS EXPECTED, DO NOT ABORT ON BOT/FUNC RTI.
635 003515 000 RANDOM:: .BYTE 0 ;RANDOM EVERYTHING FLAG.
636 003516 000 VFYFLG:: .BYTE 0 ;SET DURING WRITE/VERIFY COMMAND.
637 003517 000 RPTFLG:: .BYTE 0 ;PERFORMANCE REPORT HAS BEEN REQUESTED.
638 003520 000 SWBFLG:: .BYTE 0 ;ENABLES SWAP BYTE FUNCTION WHEN NOT EQUAL TO ZERO.
639 003521 000 IRE:: .BYTE 0 ;INHIBIT RESIDUAL FRAME COUNT ERROR REPORT.
640 003522 000 DROPED:: .BYTE 0 ;CURRENT UNIT HAS BEEN DROPPED
641 003523 000 T1SWB:: .BYTE 0 ;TEST1 SWAP BYTES FLAG
642 003524 000 ALLEOT:: .BYTE 0 ;ALL UNITS @ EOT FLAG
643 003525 000 ERSFLG:: .BYTE 0 ;ERASE FLAG: DO ERASE AFTER A SPACE REV TO DELETE
644 ;BADLY WRITTEN RECORD. 1 TO 4 ERASES LEAVING
645 ;A 3 TO 12 INCH GAP MAY RESULT.
646 .EVEN
647 003526 ENDFLG=.
648
649 ;
650 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED ONLY AFTER BEING CHECKED.
651 003526 000 STAFLG:: .BYTE 0 ;START FLAG - SET BY INIT CODE IF STARTING.
```

```
652 003527      000      PWRFLG:: .BYTE 0      ;POWER FAILURE FLAG - SET ONLY DURING INIT.
653 003530      000      TRAPD4:: .BYTE 0      ;TRAPED AT 4 FLAG
654 003531      000      MISCFG:: .BYTE 0      ;MISCELLANEOUS FLAG
655 003532      000000    TSUNT:: .WORD 0      ;NUMBER OF THE UNIT UNDER TEST PLUS HSSP&BUF
656 003534      000000    TSNP:: .WORD 0      ;FOR PRINT OUT UNIT # ONLY
657
658      ;      OPERATOR FLAG SETTINGS PASSED BY DIAG. SUPERVISOR IN A 16 BIT WORD
659      ;      SEE GLOBAL EQUATES SECTION FOR FLAG BIT LIST
660
661 003536      000000    OPFLAG:: .WORD 0      ;READ ONLY OPERATOR FLAG WORD
662      .EVEN
663
664      ;THE FOLLOWING IS THE COMMAND SEQUENCE TABLE. THE TABLE
665      ;HAS DEFAULT VALUES AT PROGRAM LOAD AS SHOWN. THESE VALUES
666      ;CAN BE UPDATED BY A TEST OR BY OPERATOR INPUT.
667
668 003540      140004    CMDSEQ:: .WORD SCH      ;SET CHARACTERISTICS.
669 003542      000040    .WORD CH.EAI
670 003544      000001    .WORD 1
671 003546      000000    .WORD 0
672 003550      102010    CMDSE2:: .WORD RWD      ;REWIND.
673 003552      000001    .WORD 1      ;BYTE COUNT.
674 003554      000001    .WORD 1      ;ONCE.
675 003556      000007    .WORD RANP      ;PATTERN.
676 003560      104005    .WORD WRT      ;WRITE.
677 003562      004000    .WORD DATCNT     ;MAX BUFFER LENGTH.
678 003564      076400    .WORD 32000.    ;32,000 RECORDS.
679 003566      000007    .WORD RANP      ;RANDOM PATTERN.
680 003570      104401    .WORD RDR      ;READ REV.
681 003572      004000    .WORD DATCNT     ;MAX BUFFER LENGTH.
682 003574      076400    .WORD 32000.    ;32,000 RECORDS
683 003576      000007    .WORD RANP      ;RANDOM PATTERN.
684 003600      104001    .WORD RDF      ;READ FWD.
685 003602      004000    .WORD DATCNT     ;MAX BUFFER LENGTH.
686 003604      076400    .WORD 32000.    ;32,000 RECORDS.
687 003606      000007    .WORD RANP      ;RANDOM PATTERN.
688 003610      102010    .WORD RWD      ;REWIND.
689 003612      000001    .WORD 1      ;BYTE COUNT.
690 003614      000001    .WORD 1      ;ONCE.
691 003616      000007    .WORD RANP      ;PATTERN.
692 003620      177777    .BLKW 40.      ;EXTENSION TO DOUBLE BUFFER SIZE
693 003740      177777    SEQEND:: .WORD END    ;SOFT END OF SEQUENCE TABLE.
694 003742      177777    .WORD END
695 003744      177777    .WORD END
696 003746      177777    .WORD END
697 003750      177777    .WORD END      ;HARD END OF SEQUENCE TABLE.
698      ;THE FOLLOWING IS THE TS05 COMMAND TABLE
699
700 003752      100013    CMDTBL:: .WORD DRI     ;DRIVE INIT.
701 003754      104001    .WORD RDF      ;READ FORWARD.
702 003756      104401    .WORD RDR      ;READ REVERSE.
703 003760      104005    .WORD WRT      ;WRITE
704 003762      104105    .WORD WTV      ;WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND
705      ;CHECK DATA ON ALL RECORDS, RDF AND
706      ;CHECK DATA ON ALL RECORDS.)
707 003764      104010    .WORD SRF      ;SPACE 'N' RECORDS FORWARD.
708 003766      104410    .WORD SRR      ;SPACE 'N' RECORDS REVERSE.
```

709	003770	105401	.WORD	RNR	:READ NEXT REVERSE. I.E., SPACE FWD, READ REVERSE.
710	003772	125401	.WORD	RNF	:READ NEXT FORWARD, I.E., READ FORWARD, SPACE REVERSE.
711	003774	105001	.WORD	RPF	:READ PREVIOUS FORWARD. I.E., SPACE REVERSE, READ FORWARD
712	003776	125001	.WORD	RPR	:READ PREVIOUS REVERSE. I.E., READ REVERSE, SPACE FORWARD
713	004000	105005	.WORD	WRR	:WRITE RETRY.
714	004002	102010	.WORD	RWD	:REWIND.
715	004004	100012	.WORD	MBR	:MESSAGE BUFFER RELEASE
716	004006	100011	.WORD	WTM	:WRITE TAPE MARK
717	004010	101011	.WORD	WTR	:WRITE TAPE MARK RETRY.
718	004012	105010	.WORD	SFF	:SPACE 'N' FILES FORWARD.
719	004014	105410	.WORD	SFR	:SPACE 'N' FILES REVERSE.
720	004016	100017	.WORD	GES	:GET EXTENDED STATUS.
721	004020	100411	.WORD	ERS	:ERASE 3 INCHES OF TAPE.
722	004022	100412	.WORD	UNL	:REWIND AND UNLOAD.
723	004024	101012	.WORD	CLN	:CLEAR TAPE.
724	004026	140004	.WORD	SCH	:SET CHARACTERISTICS.
725	004030	100006	.WORD	DIA	:DIAGNOSTIC COMMAND.
726	004032	000040	.WORD	JMP	:JUMP TO THE NTH COMMAND IN THE SEQUENCE.
727	004034	000020	.WORD	DLY	:DELAY 'N' MS.
728	004036	177777	.WORD	END	:END OF COMMAND TABLE

: THE FOLLOWING TABLE CONTAINS THE ASCII FOR EACH COMMAND.

732	004040	104	122	111	CMDASC:: .ASCII /DRI/	:DRIVE INIT.
733	004043	122	104	106	.ASCII /RDF/	:READ FORWARD.
734	004046	122	104	122	.ASCII /RDR/	:READ REVERSE.
735	004051	127	122	124	.ASCII /WRT/	:WRITE
736	004054	127	124	126	.ASCII /WTV/	:WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND CHECK DATA
737						:ON ALL RECORDS, RDF AND CHECK DATA ON ALL RECORDS.)
738	004057	123	122	106	.ASCII /SRF/	:SPACE 'N' RECORDS FORWARD.
739	004062	123	122	122	.ASCII /SRR/	:SPACE 'N' RECORDS REVERSE.
740	004065	122	116	122	.ASCII /RNR/	:READ NEXT REVERSE. I.E., SPACE FWD READ REVERSE.
741	004070	127	116	106	.ASCII /RNF/	:READ NEXT FORWARD, I.E., READ FORWARD, SPACE REVERSE.
742	004073	122	120	106	.ASCII /RPF/	:READ PREVIOUS FORWARD. I.E., SPACE REVERSE, READ FORWARD
743	004076	122	120	122	.ASCII /RPR/	:READ PREVIOUS REVERSE. I.E., READ REVERSE, SPACE FORWARD
744	004101	127	122	122	.ASCII /WRR/	:WRITE RETRY.
745	004104	122	127	104	.ASCII /RWD/	:REWIND.
746	004107	115	102	122	.ASCII /MBR/	:MESSAGE BUFFER RELEASE
747	004112	127	124	115	.ASCII /WTM/	:WRITE TAPE MARK
748	004115	127	124	122	.ASCII /WTR/	:WRITE TAPE MARK RETRY.
749	004120	123	106	106	.ASCII /SFF/	:SPACE 'N' FILES FORWARD.
750	004123	123	106	122	.ASCII /SFR/	:SPACE 'N' FILES REVERSE.
751	004126	107	105	123	.ASCII /GES/	:GET EXTENDED STATUS.
752	004131	105	122	123	.ASCII /ERS/	:ERASE 3 INCHES OF TAPE.
753	004134	125	116	114	.ASCII /UNL/	:REWIND AND UNLOAD.
754	004137	103	114	116	.ASCII /CLN/	:CLEAN TAPE.
755	004142	123	103	110	.ASCII /SCH/	:SET CHARACTERISTICS. WHERE BRF=200, 40, 20, 0.
756						:SEE TSV05/TS05 PROGRAMMING SPECIFICATION FOR DESCRIPTION.
757	004145	104	111	101	.ASCII /DIA/	:DIAGNOSTICS. SEE TSV05/TS05 PROGRAMMING SPECIFICATION
758						:FOR DESCRIPTION. ODT MUST BE USED TO LOAD DIAGNOSTIC DATA
759						:INTO THE WRITE BUFFER BEFORE THIS CMD IS ISSUED.
760	004150	112	115	120	.ASCII /JMP/	:JUMP TO THE NTH COMMAND IN THE COMMAND
761						:SEQUENCE TABLE, WHERE N IS DEFINED IN
762						:THE # OF OPERATIONS.
763	004153	104	114	131	.ASCII /DLY/	:DELAY 'N' MS, WHERE N IS DEFINED IN
764						:THE # OF OPERATIONS.
765	004156	105	116	104	.ASCII /END/	:END OF COMMAND SEQUENCE.

```

766                                     .EVEN
767
768
769
770                                     .SBTTL GLOBAL TEXT SECTION
771
772                                     :++
773                                     : THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
774                                     : MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
775                                     : MORE THAN ONE TEST.
776                                     :--
777
778
785                                     :
786                                     : FORMAT STATEMENTS USED IN PRINT CALLS
787                                     :
788
789                                     .NLIST BEX
790
791 004162      045      116      045 CODELM:: .ASCII /%AUNIT %D1% TS05 CODE LEVEL %O3%N%N/
792 004231      045      116      045 SWSET:: .ASCII /%AUNIT %D1% TS05 SWITCH SETTINGS %O3%N%N/
793                                     .EVEN
794 004306      130      130      130 HALTM:: .ASCII /XXX CMD - TYPE <CR> TO CONTINUE/
795 004346      103      115      104 CMDPKM:: .ASCII /CMD PACKET ADR NOT ON MODULO 4 BOUNDARY: RELOAD!/
796                                     .EVEN
797 004430      104      101      124 WTVRM:: .ASCII /DATA COMPARE ERROR/
798 004453      116      117      040 TOERM:: .ASCII /NO TSV05 RESPONSE/
799 004475      125      116      104 SCERM:: .ASCII /UNDEFINED SPEC COND/
800 004521      122      106      103 RFCERM:: .ASCII /RFC NON ZERO/
801 004536      124      123      126 NSSRM:: .ASCII /TSV05 NOT READY/
802 004556      122      105      124 RLEXM:: .ASCII /RETRY LIMIT EXCEEDED/
803 004603      104      122      111 ATTNM:: .ASCII /DRIVE OFF LINE/
804 004622      106      125      116 FUNRM:: .ASCII /FUNCTION REJECT/
805 004642      106      101      124 FATSM:: .ASCII /FATAL SUBSYSTEM ERROR/
806 004670      116      117      040 NOINTM:: .ASCII /NO INTERRUPT/
807 004705      124      101      120 TSAM:: .ASCII /TAPE STATUS ALERT/
808 004727      124      117      117 TOOMM:: .ASCII /TOO MANY INTERRUPTS/
809 004753      103      101      120 RNYM:: .ASCII /CAPSTAN RUNAWAY-GET STATUS RESULTS:/
810 005017      122      105      103 RERM:: .ASCII /RECOVERABLE ERROR/
811 005041      125      116      122 URERM:: .ASCII /UNRECOVERABLE ERROR/
812 005065      045      116      045 DROPDM:: .ASCII /%A DROPPED UNIT %D1%N/
813 005114      045      116      045 AUDRPM:: .ASCII /%A ALL UNITS DROPPED%N%N/
814 005146      045      116      045 AUDRUN:: .ASCII /%A DIAGNOSTIC ONLY SUPPORTS ONE CONTROLLER%N%N/
815 005226      045      116      045 DTAER2:: .ASCII '%A BYTE: %D4% S2% A WAS: %B8% S2% AS/ B: %B8%N'
816 005275      045      104      064 DTAER3:: .ASCII '%D4% A BYTES IN ERROR OUT OF %D4%N'
817 005337      045      101      116 DTAER4:: .ASCII /%A NO DATA READ%N/
818 005360      045      101      122 DTAER5:: .ASCII /%A RECORD TOO LONG: >%D4% A BYTES%N/
819 005422      045      101      122 NURTY1:: .ASCII /%A RECOVERED ON RETRY # %D2%N/
820 005456      045      101      104 OFLINM:: .ASCII /%A DRIVE %D1% A OFF LINE%N/
821 005507      045      101      107 GETSTM:: .ASCII /%A GET STATUS CMD RESULTS: %N/
822 005543      045      116      045 NODEV:: .ASCII /%A BUS TRAP AT %O6%N/
823 005570      045      101      111 .ASCII /%A INTERFACE BAD OR TSDB NOT SET TO ABOVE ADDRESS%N/
824 005653      040      052      052 UNIWLK:: .ASCII / *****TAPE IS WRITE-LOCKED AND WILL CAUSE ERRORS*****/
825 005741      045      116      000 CRLF:: .ASCII /%N/
826 005744      045      116      045 CRLFSP:: .ASCII /%N%S7/
827                                     .LIST BEX
828                                     .EVEN

```

```

829      .SBTTL  GLOBAL ERROR REPORT SECTION
830
831      ;++
832      ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
833      ; THAT ARE USED IN MORE THAN ONE TEST.  IT ALSO INCLUDES THE ASCII MESSAGES
834      ; THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
835      ;--
836
837
838      005752      BGNMSG  DTAERM
839      005752      DTAERM::
840      005752      DATERM::PRINTB  #STAER1,TSNP,PASCNT(R5),RECCNT(R5)
841      005752      016546      003376
842      005756      016546      003326
843      005762      013746      003534
844      005766      012746      006436
845      005772      012746      000004
846      005776      010600
847      006000      104414
848      006002      062706      000012
849      006006      PRINTB  #STAER7
850      006006      012746      006530
851      006012      012746      000001
852      006016      010600
853      006020      104414
854      006022      062706      000004
855      006026      010237      007066
856      006032      010337      003436
857      006036      010437      003440
858      006042      004737      010064
859      006046      013702      007066
860      006052      010337      007066
861      006056      013703      003436
862      006062      013704      003440
863      006066      MOV      R2,RECRED      ;SAVE R2
864      006066      013746      007066      MOV      R3,TIME1      ;SAVE R3
865      006072      012746      006562      MOV      R4,TIME2      ;SAVE R4
866      006076      012746      000002      JSR      PC,RECTAP      ;RETRIEVE RECORD READ
867      006102      010600
868      006104      104414
869      006106      062706      000006      MOV      RECRED,R2      ;RESTORE R2
870      006112      000167      MOV      R3,RECRED      ;SAVE RECORD READ
871      006114      000000      MOV      TIME1,R3      ;RESTORE R3
872      006116      000000      MOV      TIME2,R4      ;RESTORE R4
873      006118      000000      PRINTB  #STAER6,RECRED      ;PRINT RECORD READ
874      006120      013746      007066      MOV      RECRED,-(SP)
875      006122      012746      006562      MOV      #STAER6,-(SP)
876      006124      012746      000002      MOV      #2,-(SP)
877      006126      010600
878      006128      104414
879      006130      062706      000006      MOV      SP,R0
880      006132      000167      EXIT      MSG      TRAP      C$PNTB
881      006134      000000      .WORD      JSJMP      #6,SP
882      006136      000000      .WORD      L10002-2-.
883      006138      000000
884      006140      .EVEN
885      006142      .EVEN
886      006144      .EVEN
887      006146      .EVEN
888      006148      .EVEN
889      006150      .EVEN
890      006152      .EVEN
891      006154      .EVEN
892      006156      .EVEN
893      006158      .EVEN
894      006160      .EVEN
895      006162      .EVEN
896      006164      .EVEN
897      006166      .EVEN
898      006168      .EVEN
899      006170      .EVEN
900      006172      .EVEN
901      006174      .EVEN
902      006176      .EVEN
903      006178      .EVEN
904      006180      .EVEN
905      006182      .EVEN
906      006184      .EVEN
907      006186      .EVEN
908      006188      .EVEN
909      006190      .EVEN
910      006192      .EVEN
911      006194      .EVEN
912      006196      .EVEN
913      006198      .EVEN
914      006200      .EVEN
915      006202      .EVEN
916      006204      .EVEN
917      006206      .EVEN
918      006208      .EVEN
919      006210      .EVEN
920      006212      .EVEN
921      006214      .EVEN
922      006216      .EVEN
923      006218      .EVEN
924      006220      .EVEN
925      006222      .EVEN
926      006224      .EVEN
927      006226      .EVEN
928      006228      .EVEN
929      006230      .EVEN
930      006232      .EVEN
931      006234      .EVEN
932      006236      .EVEN
933      006238      .EVEN
934      006240      .EVEN
935      006242      .EVEN
936      006244      .EVEN
937      006246      .EVEN
938      006248      .EVEN
939      006250      .EVEN
940      006252      .EVEN
941      006254      .EVEN
942      006256      .EVEN
943      006258      .EVEN
944      006260      .EVEN
945      006262      .EVEN
946      006264      .EVEN
947      006266      .EVEN
948      006268      .EVEN
949      006270      .EVEN
950      006272      .EVEN
951      006274      .EVEN
952      006276      .EVEN
953      006278      .EVEN
954      006280      .EVEN
955      006282      .EVEN
956      006284      .EVEN
957      006286      .EVEN
958      006288      .EVEN
959      006290      .EVEN
960      006292      .EVEN
961      006294      .EVEN
962      006296      .EVEN
963      006298      .EVEN
964      006300      .EVEN
965      006302      .EVEN
966      006304      .EVEN
967      006306      .EVEN
968      006308      .EVEN
969      006310      .EVEN
970      006312      .EVEN
971      006314      .EVEN
972      006316      .EVEN
973      006318      .EVEN
974      006320      .EVEN
975      006322      .EVEN
976      006324      .EVEN
977      006326      .EVEN
978      006328      .EVEN
979      006330      .EVEN
980      006332      .EVEN
981      006334      .EVEN
982      006336      .EVEN
983      006338      .EVEN
984      006340      .EVEN
985      006342      .EVEN
986      006344      .EVEN
987      006346      .EVEN
988      006348      .EVEN
989      006350      .EVEN
990      006352      .EVEN
991      006354      .EVEN
992      006356      .EVEN
993      006358      .EVEN
994      006360      .EVEN
995      006362      .EVEN
996      006364      .EVEN
997      006366      .EVEN
998      006368      .EVEN
999      006370      .EVEN
1000     006372      .EVEN
1001     006374      .EVEN
1002     006376      .EVEN
1003     006378      .EVEN
1004     006380      .EVEN
1005     006382      .EVEN
1006     006384      .EVEN
1007     006386      .EVEN
1008     006388      .EVEN
1009     006390      .EVEN
1010     006392      .EVEN
1011     006394      .EVEN
1012     006396      .EVEN
1013     006398      .EVEN
1014     006400      .EVEN
1015     006402      .EVEN
1016     006404      .EVEN
1017     006406      .EVEN
1018     006408      .EVEN
1019     006410      .EVEN
1020     006412      .EVEN
1021     006414      .EVEN
1022     006416      .EVEN
1023     006418      .EVEN
1024     006420      .EVEN
1025     006422      .EVEN
1026     006424      .EVEN
1027     006426      .EVEN
1028     006428      .EVEN
1029     006430      .EVEN
1030     006432      .EVEN
1031     006434      .EVEN
1032     006436      .EVEN
1033     006438      .EVEN
1034     006440      .EVEN
1035     006442      .EVEN
1036     006444      .EVEN
1037     006446      .EVEN
1038     006448      .EVEN
1039     006450      .EVEN
1040     006452      .EVEN
1041     006454      .EVEN
1042     006456      .EVEN
1043     006458      .EVEN
1044     006460      .EVEN
1045     006462      .EVEN
1046     006464      .EVEN
1047     00646
```

006140	012746	000004			MOV	#4,-(SP)
006144	010600				MOV	SP,R0
006146	104414				TRAP	C\$PNTB
006150	062706	000012			ADD	#12,SP
862 006154			PRINTB	#STAER7		
006154	012746	006530			MOV	#STAER7,-(SP)
006160	012746	000001			MOV	#1,-(SP)
006164	010600				MOV	SP,R0
006166	104414				TRAP	C\$PNTB
006170	062706	000004			ADD	#4,SP
863 006174	013702	002330	MOV	CMDPKT,R2		
864 006200	042702	177740	BIC	#177740,R2		
865 006204	005302		DEC	R2		
866 006206	005702		TST	R2		:IF CMD IS A READ
867 006210	001016		BNE	50000\$		
868 006212	004737	010064	JSR	PC,RECTAP		:THEN RETRIEVE
869 006216	010337	010064	MOV	R3,RECTAP		:AND
870 006222			PRINTB	#STAER6,RECRED		:TYPE RECORD READ
006222	013746	007066			MOV	RECRED,-(SP)
006226	012746	006562			MOV	#STAER6,-(SP)
006232	012746	000002			MOV	#2,-(SP)
006236	010600				MOV	SP,R0
006240	104414				TRAP	C\$PNTB
006242	062706	000006			ADD	#6,SP
871 006246			50000\$: PRINTX	#STAER2		
006246	012746	006616			MOV	#STAER2,-(SP)
006252	012746	000001			MOV	#1,-(SP)
006256	010600				MOV	SP,R0
006260	104415				TRAP	C\$PNTX
006262	062706	000004			ADD	#4,SP
872 006266			PRINTX	#STAER3,CMDPKT,@TSDB(R5),MSGPKT+MS.RFC,TSSREG,CTCC		
006266	013746	003450			MOV	CTCC,-(SP)
006272	013746	003454			MOV	TSSREG,-(SP)
006276	013746	002360			MOV	MSGPKT+MS.RFC,-(SP)
006302	017546	002514			MOV	@TSDB(R5),-(SP)
006306	013746	002330			MOV	CMDPKT,-(SP)
006312	012746	006675			MOV	#STAER3,-(SP)
006316	012746	000006			MOV	#6,-(SP)
006322	010600				MOV	SP,R0
006324	104415				TRAP	C\$PNTX
006326	062706	000016			ADD	#16,SP
873 006332			PRINTX	#STAER4,CMDPKT+2,CMDPKT+4,CMDPKT+6		
006332	013746	002336			MOV	CMDPKT+6,-(SP)
006336	013746	002334			MOV	CMDPKT+4,-(SP)
006342	013746	002332			MOV	CMDPKT+2,-(SP)
006346	012746	006733			MOV	#STAER4,-(SP)
006352	012746	000004			MOV	#4,-(SP)
006356	010600				MOV	SP,R0
006360	104415				TRAP	C\$PNTX
006362	062706	000012			ADD	#12,SP
874 006366			PRINTX	#STAER5,MSGPKT+MS.XS0,MSGPKT+MS.XS1,MSGPKT+MS.XS2,MSGPKT+MS.XS3,MSGPKT+MS.XS		
006366	013746	002372			MOV	MSGPKT+MS.XS4,-(SP)
006372	013746	002370			MOV	MSGPKT+MS.XS3,-(SP)
006376	013746	002366			MOV	MSGPKT+MS.XS2,-(SP)
006402	013746	002364			MOV	MSGPKT+MS.XS1,-(SP)
006406	013746	002362			MOV	MSGPKT+MS.XS0,-(SP)
006412	012746	006753			MOV	#STAER5,-(SP)


```

924      JSR      PC,SOFINIT
925      BCS      CONTINUE
926      ERDF
927      ;REPORT FATAL ERROR
928      :-
929
930 007072  SOFINIT::
931
932 007072 012775 000000 002524      MOV      #0,@TSSR(R5)      ; (SAVREG) SAVE THE REGISTERS
933 007100 004737 012700      JSR      PC,WSSR      ; DO THE INIT.
934 007104 012703 000550      MOV      #360.,R3      ;WAIT FOR UNIT TO BE READY
935 007110 004737 007204 2$:      JSR      PC,WAITF      ; WAIT FOR SSR
936 007114 103416      BCS      3$
937 007116      DELAY      250
                                MOV      #250,(PC)+
                                .WORD    0
                                MOV      L$DLY,(PC)+
                                .WORD    0
                                DEC      -6(PC)
                                BNE      -4
                                DEC      -22(PC)
                                BNE      -20
938 007146 005303      DEC      R3
939 007150 001357      BNE      2$
940 007152 017500 002524 3$:      MOV      @TSSR(R5),R0      ;GET THE TSSR REGISTER
941 007156 010004      MOV      R0,R4      ;TSSR CONTENTS
942 007160 042704 176277      BIC      #^C<TS.A17!TS.A16!TS.OFL>,R4
943 007164 052704 002200      BIS      #TS.SSR!TS.NBA,R4      ;R4 HAS EXPECTED CONTENTS
944 007170 020400      CMP      R4,R0      ;ONLY EXPECTED BITS SET ?
945 007172 001402      BEQ      5$      ;BRANCH IF OKAY
946 007174 000241      CLC      ;CLEAR THE CARRY FOR ERROR
947 007176 000401      BR      10$      ;GO TO EXIT
948 007200 000261 5$:      SEC      ;SET THE CARRY BIT
949 007202 000207 10$:      RTS      PC      ;RETURN TO CALLER
950
951      ;
952      SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
953
954      INPUTS:
955
956      RS      CURRENT UNIT NUMBER
957
958      OUTPUTS:
959
960      R0      CONTENTS OF LAST TSSR READ
961      CARRY    SET - READY BIT SET
962      CLR      CLR - TIMEOUT WAITING FOR READY
963
964
965 007204 104422 005670 002524 WAITF:: BREAK      ; DO A SUPVSR BREAK FIRST.
966 007206 012746      MOV      #3000.,-(SP)      ; 300 MSEC TIMER.
967 007212 017500 002524 2$:      MOV      @TSSR(R5),R0      ;READ THE TSSR REGISTER
968 007216 105700      TSTB      R0      ;TEST FOR READY BIT SET
969 007220 100420      BMI      3$      ; EXIT ON STOP FLAG.
970 007222      DELAY      25      ; WAIT
                                MOV      #25,(PC)+

```

```

007226 000000
007230 013727 002116
007234 000000
007236 005367 177772
007242 001375
007244 005367 177756
007250 001367
971 007252 005316
972 007254 001356
973 007256 000241
974 007260 000401
975 007262 000261
976 007264 005326
977 007266 000207
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009 007270
1010 007270 010475 002514
1011 007274 004737 007204
1012 007300 103401
1013 007302 000421
1014 007304 005724
1015 007306 011402
1016 007310 011203
1017 007312 032763 000200 000012
1018 007320 001402
1019 007322 005237 002322
1020 007326

                                .WORD 0
                                MOV    L$DLY,(PC)+
                                .WORD 0
                                DEC    -6(PC)
                                BNE    -4
                                DEC    -22(PC)
                                BNE    -20

                                DEC    (SP)      ;REDUCE DELAY COUNT
                                BNE    2$        ;RETRY UNTIL TIMER EXPIRES
                                CLC
                                BR     4$        ; C = 0, CONTROLLER STILL RUNNING...
                                SEC            ;...OR HUNG-UP AFTER 300 MSEC.
                                DEC    (SP)+      ; C = 1, CONTROLLER IS STOPPED.
                                RTS     PC        ;RESTORE STACK WITHOUT CHANGING CARRY BIT

;+
;ROUTINE TO ISSUE A WRITE CHARACTERISTICS COMMAND AND CHECK FEATURES
;INPUT:
;      R4      ADDRESS OF COMMAND PACKET
;      R5      CURRENT UNIT NUMBER
;      REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
;OUTPUT:
;      R0      TSSR CONTENTS
;      CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
;              CLR - WRITE CHARACTERISTICS FAILED
;IMPLICIT OUTPUT:
;      SOFTWARE SWITCHES SET AS FOLLOWS:
;      EXTFEA = EXTENDED FEATURES PRESENT
;      BENBSW = BUFFER ENABLE SWITCH ON OR OFF
;SIDE EFFECTS:
;-

WRTCHK::
10$:  MOV    R4,@TSDB(R5)      ;SEND OUT COMMAND
      JSR    PC,WAITF        ;WAIT FOR SSR
      BCS    40$             ;BR, IF SSR IS SET AND OK
      BR     60$             ;BR IF TROUBLE CARRY = CLEAR
40$:  TST    (R4)+            ;STEP IT
      MOV    (R4),R2          ;POINT TO WRT CHARA DATA PACKET
      MOV    (R2),R3          ;GET ADDRESS OF MESSAGE BUFFER
      BIT    #X2.EFE,MS.XS2(R3) ;EXTENDED FEATURES BIT SET?
      BEQ    45$             ;BR IF NO
      INC    EXTFEA          ;SET EXTENDED FEATURES SW SWITCH
45$:

```

```

1021 007326 032763 000100 000012      BIT    #X2.BFE,MS.XS2(R3)      ;BUFFER ENABLE SWITCH SET
1022 007334 001402                      BEQ     50$              ;BR, IF SWITCH NOT SET
1023 007336 005237 002324              INC     BENBSW          ;SET SOFTWARE SWITCH FOR ENABLED
1024 007342                      50$:
1025 007342 000261                      55$:      SEC              ;SET CARRY NO TROUBLE
1026 007344 000401                      BR      70$              ;EXIT
1027 007346 000241                      60$:      CLC              ;CARRY CLEAR = ERROR
1028 007350 017500 002524              70$:      MOV     @TSSR(R5),R0    ;RETURN TSSR CONTENTS
1029 007354 000207                      RTS      PC              ;RETURN
1030
1031      ;+
1032      ;ROUTINE TO CHECK WRITE LOCK CONDITION
1033      ;INPUT:
1034      ;
1035      ;R4      ADDRESS OF COMMAND PACKET
1036      ;R5      CURRENT UNIT NUMBER
1037      ;
1038      ;-
1039      ;
1040      ;
1041 007356      WLKCHK::
1042 007356 010475 002514              10$:      MOV     R4,@TSDB(R5)      ;SEND OUT COMMAND
1043 007362 004737 007204              JSR     PC,WAITF          ;WAIT FOR SSR
1044 007366 103401                      BCS     40$              ;BR, IF SSR IS SET AND OK
1045 007370 000420                      BR      60$              ;BR IF TROUBLE CARRY = CLEAR
1046 007372 005724                      40$:      TST     (R4)+          ;STEP IT
1047 007374 011402                      MOV     (R4),R2          ;POINT TO WRT CHARA DATA PACKET
1048 007376 011203                      MOV     (R2),R3          ;GET ADDRESS OF MESSAGE BUFFER
1049 007400 032763 000004 000006      BIT     #X0.WLK,MS.XS0(R3)    ;IS UNIT WRITE LOCKED?
1050 007406 001407                      BEQ     55$              ;NO,PROCEED WITH TESTING
1051 007410                      ERRHRD 1,UNIWLK          ;TAPE IS WRITE LOCKED
1052 007410 104456                      TRAP    .WORD 1      CSERHRD
1053 007412 000001                      .WORD 1
1054 007414 005653                      .WORD 1      UNIWLK
1055 007416 000000                      .WORD 0
1056 007420 004737 017156              JSR     PC,DROPU          ;DROP IT
1057 007424 000402                      BR      60$              ;EXIT WITH CARRY=0
1058 007426 000261                      55$:      SEC              ;SET CARRY NO TROUBLE
1059 007430 000401                      BR      70$              ;EXIT
1060 007432 000241                      60$:      CLC              ;CARRY CLEAR = ERROR
1061 007434                      70$:      RTS      PC              ;RETURN
1062
1063      ;+
1064      ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS COMMAND
1065      ;INPUT:
1066      ;
1067      ;R4      ADDRESS OF COMMAND PACKET
1068      ;R5      CURRENT UNIT NUMBER
1069      ;REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1070      ;
1071      ;OUTPUT:
1072      ;
1073      ;R0      TSSR CONTENTS

```

```

1074      CARRY SET - WRITE CHARACTERISTICS COMMAND OK
1075      CLR - WRITE CHARACTERISTICS FAILED
1076
1077      IMPLICIT OUTPUT:
1078
1079
1080
1081      SIDE EFFECTS:
1082
1083
1084      -
1085
1086 007436 WRTCHR::
1087 007436 010475 002514 10$: MOV R4,@TSDB(R5) ;SEND OUT COMMAND
1088 007442 004737 007204 JSR PC,WAITF ;WAIT FOR SSR
1089 007446 103401 BCS 50$ ;BR, IF SSR IS SET AND OK
1090 007450 000402 BR 60$ ;BR IF TROUBLE CARRY = CLEAR
1091 007452
1092 007452 000261 50$: SEC ;SET CARRY NO TROUBLE
1093 007454 000401 BR 70$ ;EXIT
1094 007456 000241 60$: CLC ;CARRY CLEAR = ERROR
1095 007460 017500 002524 70$: MOV @TSSR(R5),R0 ;RETURN TSSR CONTENTS
1096 007464 000207 RTS PC ;RETURN
1097
1098
1099
1100      +
1101      ROUTINE TO DO SET UP OF RUNNING CONDITIONS
1102
1103      INPUTS:
1104
1105      R5 CURRENT UNIT NUMBER
1106
1107
1108      OUTPUTS:
1109
1110
1111      CALLING SEQUENCE:
1112      JSR PC,FIRSTU
1113      JSR PC,SOFINIT
1114      BCS CONTINUE
1115      ERRDF ;REPORT FATAL ERROR
1116      JSR PC,MDSET
1117
1118      -
1119
1120 007466 MDSET:: BREAK ; DO A SUPVSR BREAK FIRST.
1121 007466 104422 JSR PC,SETDEF TRAP CSBRK
1122 007474 004737 007766 JSR PC,WLKCHK ;RESTORE DEFAULT
1123 007500 103416 BCS 1$ ;CHECK WRITE LOCK
1124 007502 DELAY 1 ;C=1 IS O.K.
1125 007502 012727 000001 ;WAIT
1126 007506 000000 MOV #1,(PC)+
1127 007510 013727 002116 .WORD 0
1128 007514 000000 MOV L$DLY,(C)+
1129 007516 005367 177772 .WORD 0
1130 DEC -6(PC)

```

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

```

1177 007732 012737 002506 002332      MOV      #WSMBK,CMDPKT+CP.ADL      ;MSG BUF ADDR
1178 007740 012737 000006 002336      MOV      #6,CMDPKT+CP.CNT      ;BYTE COUNT
1179 007746 012737 100010 002506      MOV      #100010,WSMBK      ;INVERT THE SWITCH
1180 007754 012704 002330              MOV      #CMDPKT,R4
1181 007760 004737 007436              JSR      PC,WRTCHR
1182 007764 000207              RTS      PC      ;DO IT
1183                                     ;RETURN
1184
1185                                     ; SUBROUTINE TO SETUP DEFAULT SET CHAR CMD
1186                                     ; INPUTS:
1187                                     ;
1188                                     ;
1189                                     ;
1190                                     ;
1191                                     ; OUTPUTS:
1192                                     ;
1193                                     ;      R4      ADDRESS OF COMMAND PACKET
1194
1195 007766      SETDEF::
1196 007766 012701 140004      MOV      #SCH,R1      ;WRITE CHAR CMD
1197 007772 010137 002330      MOV      R1,CMDPKT+CP.CMD      ;SET UP COMMAND
1198 007776 012737 002474 002332      MOV      #SCHBK,CMDPKT+CP.ADL      ;SET UP ADR LO TO POINT TO MSG BUF(MSGPKO)
1199 010004 012737 000012 002336      MOV      #SCHCNT,CMDPKT+CP.CNT      ;SET BUFFER EXTENT
1200 010012 012737 000040 002502      MOV      #DFTSCH,SCHBK+6      ;STORE CHARACTERISTIC CODE IN SCH BLOCK.
1201 010020 013737 003532 002504      MOV      TSUNT,SCHBK+10      ;UNIT #
1202 010026 012704 002330      MOV      #CMDPKT,R4      ;ADDRESS OF CMD PACKET
1203 010032 000207      RTS      PC      ;RETURN
1204
1205
1206      ;      MODULES TO HANDLE TS05 INTERRUPTS.
1207
1208
1209 010034      BGNSRV      TS5IN0
1210 010034 005237 003472      TS5IN0::      INC      INTFLG      ;SET INTERRUPT OCCURRED FLAG.
1211 010040      ENDSRV
1212 010040 000002      L10004:
1213                                     RTI
1214 010042      BGNSRV      TS5IN1
1215 010042 005237 003474      TS5IN1::      INC      INTFLG+2      ;SET INTERRUPT OCCURRED FLAG.
1216 010046      ENDSRV
1217 010046 000002      L10005:
1218                                     RTI
1219 010050      BGNSRV      TS5IN2
1220 010050 005237 003476      TS5IN2::      INC      INTFLG+4      ;SET INTERRUPT OCCURRED FLAG.
1221 010054      ENDSRV
1222 010054 000002      L10006:
1223                                     RTI
1224 010056      BGNSRV      TS5IN3
1225 010056 005237 003500      TS5IN3::      INC      INTFLG+6      ;SET INTERRUPT OCCURRED FLAG.
1226 010062      ENDSRV

```

```

010062      000002      L10007:
1224      010062      RTI
1225      :
1226      :      SUBROUTINE TO RETRIEVE RECORD COUNT READ FROM TAPE FOR ERROR
1227      :      PRINTS.
1228      :      INPUTS:
1229      :      OUTPUTS: R3 = RECORD COUNT READ
1230      :      REGISTERS: R2, R3, R4
1231      :      CALLS:
1232      010064      032737      000400      003420      RECTAP: BIT      #MOD.CO,CMDWRD      ;READ REV FETCH
1233      010072      001430      BEQ      50001$
1234      010074      013702      002360      MOV      MSGPKT+MS.RFC,R2      ;FIND LAST READ AD.
1235      010100      063702      003410      ADD      DATARD,R2
1236      010104      032702      000001      BIT      #BIT00,R2      ;ODD AD., REASSEMBLE
1237      010110      001417      BEQ      50002$
1238      010112      005202      INC      R2      ;REC COUNT STARTING
1239      010114      111203      MOVB      (R2),R3      ;WITH UPPER BYTE FETCH
1240      010116      142703      177400      BICB      #177400,R3
1241      010122      000303      SWAB      R3
1242      010124      005302      DEC      R2      ;LET R2 := R2 - #1      ;LOWER BYTE AD.
1243      010126      105737      003520      TSTB      SWBFLG      ;IFB SWBFLG NE #0 THEN
1244      010132      001401      BEQ      50003$
1245      010134      005302      DEC      R2      ;LET R2 := R2 - #1      ;LOWER BYTE AD. ON SWAP
1246      :
1247      010136      111204      50003$:      MOVB      (R2),R4      ;FETCH LOWER BYTE
1248      010136      142704      177400      BICB      #177400,R4
1249      010140      050403      BIS      R4,R3
1250      010144      000401      BR      50004$
1251      010146      011203      50002$:      MOV      (R2),R3      ;LET R3 := (R2)      ;EVEN AD. FETCH
1252      010150      000402      50004$:      BR      50005$
1253      010152      017703      173230      50001$:      MOV      @DATARD,R3      ;LET R3 := @DATARD      ;READ FWD FETCH
1254      010154      000207      50005$:      RTS      PC
1255      010160      :
1256      010160      :      SUBROUTINE TO STORE A SET CHARACTERISTIC COMMAND AS
1257      010160      :      THE FIRST ENTRY IN THE SEQUENCE TABLE.
1258      010160      :      INPUTS:
1259      010160      :      OUTPUTS:
1260      010160      :      REGISTERS:
1261      010160      :      CALLS:
1262      010162      :
1263      010162      :      SETCH::
1264      010162      012701      003540      MOV      #CMDSEQ,R1      ;INIT CMD SEQUENCE TABLE POINTER.
1265      010166      012721      140004      MOV      #SCH,(R1)+      ;THIS CODE SETS UP A SET CHARACTERISTIC
1266      010172      012721      000040      MOV      #DFTSCH,(R1)+      ;COMMAND AS THE FIRST COMMAND IN THE
1267      010176      012721      000001      MOV      #1,(R1)+      ;SEQUENCE TABLE.
1268      010202      005721      TST      (R1)+      ;SKIP PATTERN LOCATION.
1269      010204      000207      RTS      PC
1270      :
1271      :      SUBROUTINE TO STORE A REWIND COMMAND IN THE SEQUENCE TABLE
1272      :      INPUTS:
1273      :
1274      :
1275      :
1276      :
1277      :
1278      :

```



```

1279      :      OUTPUTS:
1280      :      REGISTERS:
1281      :      CALLS:
1282
1283 010206 012721 102010      SETRW:: MOV      #RWD,(R1)+      ;CMD = REWIND.
1284 010212 012721 000001      MOV      #1,(R1)+      ;BRF.
1285 010216 012721 000001      MOV      #1,(R1)+      ;# OF OPERATIONS.
1286 010222 005721      TST      (R1)+      ;SKIP PATTERN.
1287 010224 000207      RTS      PC      ;RETURN
1288
1289      :      SUBROUTINE TO EXECUTE ALL COMMANDS IN THE SEQUENCE TABLE ON ALL
1290      :      DEVICES.
1291      :      INPUTS:
1292      :      OUTPUTS:      R2 = TERMINATION INDICATOR (0=END OF TABLE,1=EOT)
1293      :      REGISTERS:
1294      :      CALLS:      CMDAC,SETUP,EXSUB,CKHAE,NEXTU,FIRSTU,VFYDAT.
1295
1296 010226 012701 003540      EXALL:: MOV      #CMDSEQ,R1      ;INIT SEQUENCE TABLE POINTER.
1297 010232 50006$      50006$:      CMP      (R1),#END      ;WHILE THERE ARE CMDS IN THE SEQUENCE TABLE.
1298 010232 021127 177777      BEQ      50007$
1299 010236 001530      JSR      PC,SETUP      ;GO SETUP THE COMMAND BLOCK.
1300 010240 004737 011172      50010$: BREAK      ; DO A SUPVSR BREAK FIRST.
1301 010244
1302 010244 104422      50010$:      CMP      NCNT,NCNT1      ;WHILE THERE ARE RECORDS REMAINING:      TRAP      C$BRK
1303 010254 002116      BGE      50011$
1304 010256 004737 011064      JSR      PC,CMDAC      ;STORE CMD ASCII IN ERROR MESSAGE.
1305 010262 105737 003515      TSTB     RANDOM      ;IF IN RANDOM MODE:
1306 010266 001435      BEQ      50012$
1307 010270 023727 003420 104005      CMP      CMDWRD,#WRT      ;IF CMD IS A WRITE THEN:
1308 010276 001031      BNE      50013$
1309 010300 105737 003516      TSTB     VFYFLG      ;IF DATA IS NOT TO BE VERIFIED THEN:
1310 010304 001026      BNE      50014$
1311 010306 063737 003434 003432      ADD      RANS,RANB      ;LET RANB := RANB + RANS ;GENERATE
1312 010314 063737 003432 003434      ADD      RANB,RANS      ;LET RANS := RANS + RANB ;RANDOM
1313 010322 013737 003434 003416      MOV      RANS,BRFCNT      ;LET BRFCNT := RANS ;LENGTH
1314 010330 043737 003430 003416      BIC      LENMSK,BRFCNT      ;MASK RANDOM LENGTH.
1315 010336 023727 003416 000022      CMP      BRFCNT,#18.      ;DO NOT ALLOW BYTE COUNT OF LESS THAN 18
1316 010344 002003      BGE      50015$
1317 010346 012737 000022 003416      MOV      #18.,BRFCNT      ;CHANGE COUNT OF 0-17 TO 18.
1318
1319 010354      50015$:
1320 010354 013737 003416 002336      MOV      BRFCNT,CMDPKT+CP.CNT      ;MOVE BRF TO CMD PACKET.
1321
1322 010362      50014$:
1323
1324 010362      50013$:
1325
1326 010362      50012$:
1327 010362 004737 010524      JSR      PC,EXSUB      ;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATUS.
1328 010366 004737 017456      JSR      PC,CKHAE      ;CHECK HALT AFTER EACH CMD FLAG.
1329 010372 012702 000001      MOV      #1,R2      ;LET R2 := #1 ;SET ALL UNITS AT BOT/EOT.
1330 010376 004737 017060      JSR      PC,FIRSTU      ;FIND FIRST UNIT.
1331
1332 010402      50016$:
1333 010402 026527 002604 177777      CMP      DEVTBL(R5),#END      ;WHILE THERE ARE MORE UNITS:
1334 010410 001426      BEQ      50017$

```

```

1335 010412 032737 000400 003420      BIT    #MOD.CO,CMDWRD      ;IF CMD IS REVERSE THEN:
1336 010420 001406                      BEQ    50020$
1337 010422 032765 000002 003502      BIT    #X0.BOT,EOTFLG(R5)    ;IF NOT AT BOT THEN:
1338 010430 001001                      BNE    50021$
1339 010432 005002                      CLR    R2                ;LET R2 := #0 ;CLEAR EOT/BOT FLAG.
1340
1341 010434                      50021$:
1342 010434 000411                      BR     50022$                ;ELSE IF CMD IS NOT REVERSE:
1343 010436                      50020$:
1344 010436 032765 000001 003502      BIT    #X0.EOT,EOTFLG(R5)
1345 010444 001404                      BEQ    50023$
1346 010446 032737 000001 003420      BIT    #CMD.CO,CMDWRD
1347 010454 001001                      BNE    50024$
1348 010456                      50023$:
1349
1350 010456 005002                      CLR    R2                ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
1351
1352 010460                      50024$:
1353
1354 010460                      50022$:
1355 010460 004737 017126      JSR    PC,NEXTU          ;FIND NEXT UNIT
1356 010464 000746                      BR     50016$
1357 010466                      50017$:
1358 010466 020227 000001      CMP    R2,#1                ;IF ALL UNIT ARE AT EOT/BOT THEN:
1359 010472 001001                      BNE    50025$
1360 010474 000412                      BR     EXARTN          ;RETURN WITH R2 = #1.
1361
1362 010476                      50025$:
1363 010476 005237 003412      INC    NCNT                ;LET NCNT := NCNT + #1 ;UPDATE RECORD COUNT.
1364 010502 013737 003420 003424      MOV    CMDWRD,PCMDWD      ;SAVE PREVIOUS COMMAND WORD.
1365
1366 010510 000655                      BR     50010$
1367 010512                      50011$:
1368 010512 004737 016044      JSR    PC,VFYDAT          ;IF LAST CMD WAS A WRITE VERIFY, THEN GO
1369
1370
1371 010516 000645                      BR     50006$
1372 010520                      50007$:
1373 010520 005002                      CLR    R2                ;LET R2 := #0 ;SET NORMAL RETURN INDICATOR.
1374 010522 000207      EXARTN: RTS PC          ;RETURN.
1375
1376
1377
1378      :      SUBROUTINE TO ISSUE COMMAND TO ALL DEVICES, WAIT FOR
1379      :      ALL INTERRUPTS, AND CHECK ALL STATUS.
1380      :      INPUTS:
1381      :      OUTPUTS:
1382      :      REGISTERS:
1383      :      CALLS:      EXCUTE,GOWAIT,NEXTU,FIRSTU.
1384
1385 010524 004737 017060      EXSUB:: JSR    PC,FIRSTU          ;SET UP FOR FIRST UNIT.
1386 010530                      50026$:
1387 010530 026527 002604 177777      CMP    DEVTBL(R5),#END      ;WHILE THERE ARE MORE DEVICES:
1388 010536 001465                      BEQ    50027$
1389 010540 032737 000400 003420      BIT    #MOD.CO,CMDWRD      ;IF CMD IS REVERSE THEN:
1390 010546 001421                      BEQ    50030$
1391 010550 032765 000002 003502      BIT    #X0.BOT,EOTFLG(R5)    ;IF NOT AT BOT

```

1392	010556	001014			BNE	50031\$	
1393	010560	032765	000001	003502	BIT	#X0.EOT,EOTFLG(R5)	;BUT IF AT EOT
1394	010566	001406			BEQ	50032\$	
1395	010570	105737	003524		TSTB	ALLEOT	;AND ALL OTHERS AT EOT
1396	010574	001402			BEQ	50033\$	
1397	010576	004737	012054		JSR	PC,EXECUTE	;THEN EXECUTE REV CMD
1398							;IF NOT ALL AT EOT, FREEZE UNIT(S) AT EOT
1399	010602				50033\$:		
1400	010602	000402			BR	50034\$;IF NOT AT BOT AND
1401	010604				50032\$:		
1402	010604	004737	012054		JSR	PC,EXECUTE	;NOT AT EOT, EXEC REV CMD
1403							
1404	010610				50034\$:		
1405							
1406	010610				50031\$:		
1407	010610	000435			BR	50035\$;ELSE IF CMD IS NOT REVERSE:
1408	010612				50030\$:		
1409	010612	023727	003426	000002	CMP	CMDLG,#2	
1410	010620	001011			BNE	50036\$	
1411	010622	032765	000002	003502	BIT	#X0.BOT,EOTFLG(R5)	
1412	010630	001405			BEQ	50036\$	
1413							;CLEAR BAD SPOT COUNTS WHEN WRITING FROM BOT
1414	010632	016537	002616	003512	MOV	BTADDR(R5),BTPT	;LET BTPT := BTADDR(R5)
1415	010640	005077	172646		CLR	@BTPT	;LET @BTPT := #0
1416							
1417	010644				50036\$:		
1418	010644	032765	000001	003502	BIT	#X0.EOT,EOTFLG(R5)	
1419	010652	001404			BEQ	50037\$	
1420	010654	032737	000001	003420	BIT	#CMD.CO,CMDWRD	
1421	010662	001003			BNE	50040\$	
1422	010664				50037\$:		
1423							;IF NOT AT EOT OR NOT A MOTION CMD THEN:
1424	010664	004737	012054		JSR	PC,EXECUTE	;ISSUE CMD TO TS05
1425							
1426	010670	000405			BR	50041\$	
1427	010672				50040\$:		
1428	010672	105737	003524		TSTB	ALLEOT	;IFB ALLEOT NE #0 THEN
1429	010676	001402			BEQ	50042\$	
1430	010700	004737	012054		JSR	PC,EXECUTE	
1431							
1432	010704				50042\$:		
1433							
1434	010704				50041\$:		
1435							
1436	010704				50035\$:		
1437	010704	004737	017126		JSR	PC,NEXTU	;FIND NEXT UNIT IN TEST CYCLE.
1438							
1439	010710	000707			BR	50026\$	
1440	010712				50027\$:		
1441	010712	105737	003517		TSTB	RPTFLG	;IF REPORT HAS BEEN REQUESTED THEN:
1442	010716	001403			BEQ	50043\$	
1443	010720	105037	003517		CLRB	RPTFLG	;CLR THE FLAG,
1444	010724				DORPT		;PRINT THE PERFORMANCE REPORT.
1445	010726						TRAP C\$DRPT
1446	010726	004737	017060		50043\$:		
1447	010732				JSR	PC,FIRSTU	;SET UP FOR FIRST UNIT.
					50044\$:		

```

1448 010732 026527 002604 177777      CMP      DEVTBL(R5),#END      ;WHILE THERE ARE MORE DEVICES:
1449 010740 001450                      BEQ      50045$
1450 010742 032737 000400 003420      BIT      #MOD.CO,CMDWRD      ;IF CMD IS REVERSE THEN:
1451 010750 001421                      BEQ      50046$
1452 010752 032765 000002 003502      BIT      #X0.BOT,EOTFLG(R5) ;IF NOT AT BOT
1453 010760 001014                      BNE      50047$
1454 010762 032765 000001 003502      BIT      #X0.EOT,EOTFLG(R5) ;BUT IF AT EOT
1455 010770 001406                      BEQ      50050$
1456 010772 105737 003524      TSTB     ALLEOT                      ;AND ALL OTHERS AT EOT
1457 010776 001402                      BEQ      50051$
1458 011000 004737 012364      JSR      PC,GOWAIT                      ;THEN WAIT FOR CMD END
1459                                     ;IF NOT ALL AT EOT, DO NOT WAIT
1460 011004                                     50051$:
1461                                     ;NOT AT BOT, AND NOT AT EOT
1462 011004 000402                                     BR      50052$
1463 011006                                     50050$:
1464 011006 004737 012364      JSR      PC,GOWAIT                      ;WAIT FOR INT,CHECK STAT
1465                                     50052$:
1466 011012
1467
1468
1469 011012                                     50047$:
1470 011012 000420                                     BR      50053$                      ;ELSE IF CMD IS FORWARD:
1471 011014                                     50046$:
1472 011014 032765 000001 003502      BIT      #X0.EOT,EOTFLG(R5)
1473 011022 001404                      BEQ      50054$
1474 011024 032737 000001 003420      BIT      #CMD.CO,CMDWRD
1475 011032 001003                      BNE      50055$
1476 011034                                     50054$:
1477                                     ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
1478 011034 004737 012364      JSR      PC,GOWAIT                      ;WAIT FOR INT,CHECK STATUS.
1479
1480 011040 000405                                     BR      50056$
1481 011042                                     50055$:
1482 011042 105737 003524      TSTB     ALLEOT                      ;IFB ALLEOT NE #0 THEN
1483 011046 001402                      BEQ      50057$
1484 011050 004737 012364      JSR      PC,GOWAIT
1485
1486 011054                                     50057$:
1487
1488 011054                                     50056$:
1489
1490 011054                                     50053$:
1491 011054 004737 017126      JSR      PC,NEXTU                      ;FIND NEXT UNIT IN TEST CYCLE.
1492
1493 011060 000724                                     BR      50044$
1494 011062                                     50045$:
1495 011062 000207      RTS      PC                      ;RETURN.
1496
1497 ;
1498 ;
1499 ;
1500 ;
1501 ;
1502 ;
1503 ;
1504 ;

```

THIS SUBROUTINE STORES THE ASCII FOR THE CURRENT COMMAND AND PREVIOUS
COMMAND IN THE STANDARD ERROR MESSAGE. ON ENTRY LOCATION CMDWRD
CONTAINS CURRENT CMD AND LOCATION PCMDWD CONTAINS PREVIOUS CMD.
INPUTS:
OUTPUTS:
REGISTERS: R3, R4.
CALLS: GCMDA

```

1505 011064 013704 003420 CMDAC:: MOV CMDWRD,R4;LET R4 := CMDWRD ;R4 = CMD BINARY.
1506 011070 004737 011136 JSR PC,GCMDA ;GET CMD ASCII.
1507 011074 112337 006440 MOVB (R3)+,STAER1+2 ;MOVE CMD ASCII
1508 011100 112337 006441 MOVB (R3)+,STAER1+3 ;
1509 011104 111337 006442 MOVB (R3),STAER1+4 ;INTO MSG.
1510 011110 013704 003424 MOV PCMDWD,R4 ;R4 = PREVIOUS CMD BINARY.
1511 011114 004737 011136 JSR PC,GCMDA ;GET CMD ASCII.
1512 011120 112337 006554 MOVB (R3)+,STAER7+24 ;MOVE CMD ASCII
1513 011124 112337 006555 MOVB (R3)+,STAER7+25 ;
1514 011130 111337 006556 MOVB (R3),STAER7+26 ;INTO MSG.
1515 011134 000207 RTS PC ;RETURN. GO EXECUTE NEXT FUNCTION.
1516
1517
1518 : SUBROUTINE TO FIND THE ASCII EQUIVALENT OF THE COMMAND IN R4.
1519 : ADDRESS OF ASCII 1ST WORD IS RETURNED IN R3.
1520 : INPUTS: R4 = PRESENT COMMAND WORD.
1521 : OUTPUTS: R3 = ADDRESS OF PRESENT COMMAND ASCII.
1522 : REGISTERS:
1523 : CALLS:
1524
1525 011136 005003 GCMDA:: CLR R3;LET R3 := #0 ;INIT CMD TBL POINTER.
1526 011140 50060$ BEQ 50061$ ;UNTIL CURRENT CMD IS FOUND:
1527 011140 026304 003752 ADD #2,R3 ;LET R3 := R3 + #2 ;SEARCH CMD TABLE.
1528 011144 001403 BR 50060$
1529 011146 062703 000002
1530 011152 000772
1531 011154 50061$: MOV R3,R4 ;LET R4 := R3
1532 011154 010304 ASR R3 ;POINT TO ASCII FOR THAT COMMAND
1533 011156 006203 NOP
1534 011160 000240 ADD R4,R3
1535 011162 060403 ADD #CMDASC,R3
1536 011164 062703 004040 RTS PC ;RETURN.
1537 011170 000207
1538
1539 : THIS SUBROUTINE LOADS THE TS05 COMMAND PACKET FROM ONE
1540 : ENTRY IN THE SEQUENCE TABLE.
1541 : INPUTS:
1542 : OUTPUTS:
1543 : REGISTERS: R2, R3.
1544 : CALLS: GENPAT.
1545
1546 011172 005037 003426 SETUP:: CLR CMDLG ;CLR CMD LOGGING CODE(DISABLES LOGGING)
1547 011176 012137 002330 MOV (R1)+,CMDPKT ;LOAD THE COMMAND WORD.
1548 011202 011137 002336 MOV (R1),CMDPKT+CP.CNT ;LOAD THE BYTE/RECORD/FILE COUNT.
1549 011206 011137 003416 MOV (R1),BRFCNT ;SAVE BRFCNT FOR THIS COMMAND.
1550 011212 013702 002330 MOV CMDPKT,R2 ;GET CMD.
1551 011216 042702 177740 BIC #NCMD.C,R2 ;CLR ALL BUT CMD BITS.
1552 011222 010203 MOV R2,R3 ;SAVE IT TWICE.
1553 011224 162703 000010 SUB #CMD.C3,R3 ;POSITION COMMAND?
1554 011230 001003 BNE 2$ ;BR IF NOT.
1555 011232 011137 002332 MOV (R1),CMDPKT+2 ;MOVE BPCR IN 2ND PKT WORD FOR POSITION CMD.
1556 011236 000464 BR 3$
1557 011240 023727 002330 100011 2$: CMP CMDPKT,#WTM ;IF CMD IS A WRITE TAPE MARK THEN:
1558 011246 001003 BNE 50062$
1559 011250 012737 000002 003426 MOV #2,CMDLG ;WTM LOGGING CODE IS 2.
1560
1561 011256 50062$:

```

1562	011256	010203			MOV	R2,R3		
1563	011260	162703	000001		SUB	#CMD.CO,R3		:IS IT A READ?
1564	011264	001017			BNE	1\$:BR IF NOT.
1565	011266	013737	003410	002332	MOV	DATARD,CMDPKT+CP.ADL		:IF SO, LOAD THE BUFFER ADDR.
1566	011274	032737	000400	002330	BIT	#MOD.CO,CMDPKT		:IF CMD IS A READ REV THEN:
1567	011302	001404			BEQ	50063\$		
1568	011304	012737	000004	003426	MOV	#4,CMDLG		:LOGGING CODE IS 4.
1569								:ELSE - IF CMD IS A READ FWD:
1570	011312	000403			BR	50064\$		
1571	011314							
1572	011314	012737	000006	003426	50063\$: MOV	#6,CMDLG		:LOGGING CODE IS 6.
1573								
1574	011322				50064\$:			
1575	011322	000432						:CONTINUE.
1576	011324	010203			1\$: MOV	R2,R3		:IS IT
1577	011326	162703	000004		SUB	#CMD.C2,R3		:A SET CHARACTERISTICS CMD?
1578	011332	001014			BNE	4\$:BR IF NOT.
1579	011334	012737	002474	002332	MOV	#SCHBK,CMDPKT+CP.ADL		:SET UP ADR LO FOR SET CHAR.
1580	011342	012737	000012	002336	MOV	#SCHCNT,CMDPKT+CP.CNT		:SET BUFFER EXTENT
1581	011350	011137	002502		MOV	(R1),SCHBK+6		:STORE CHARACTERISTIC CODE IN SCH BLOCK.
1582	011354	013737	003532	002504	MOV	TSUNT,SCHBK+10		:UNIT #
1583	011362	000412			BR	3\$:CONTINUE.
1584	011364	010203			4\$: MOV	R2,R3		:IS IT
1585	011366	162703	000006		SUB	#CMD.C1!CMD.C2,R3		:A DIAGNOSTIC (DIA) CMD?
1586	011372	001006			BNE	3\$:BR IF NOT.
1587	011374	012737	000020	002336	MOV	#DIACNT,CMDPKT+CP.CNT		:LOAD BUFFER EXTENT.
1588	011402	012737	003406	002332	MOV	#DIABLK,CMDPKT+CP.ADL		:LOAD BUFFER ADR LOW.
1589	011410	005721			3\$: TST	(R1)+		:POINT TO N (NUMBER OF TIMES TO EXECUTE THIS INS
1590	011412	012137	003414		MOV	(R1)+,NCNT1		:SAVE NUMBER OF OPERATIONS
1591	011416	005037	003412		CLR	NCNT		:CLEAR OPERATION COUNTER.
1592	011422	012137	003446		MOV	(R1)+,PATERN		:SAVE PATTERN CODE FOR CURRENT CMD.
1593	011426	010203			MOV	R2,R3		:IS IT
1594	011430	162703	000005		SUB	#CMD.CO!CMD.C2,R3		:A WRITE?
1595	011434	001010			BNE	5\$:BR IF NOT.
1596	011436	013737	003406	002332	MOV	DATARD,CMDPKT+CP.ADL		:LOAD WRITE BUFFER LO ORDER.
1597	011444	004737	011556		JSR	PC,GENPAT		:GO GENERATE THE WRITE PATTERN.
1598	011450	012737	000002	003426	MOV	#2,CMDLG		:WRITE LOGGING CODE IS 2.
1599	011456	032737	000100	002330	5\$: BIT	#VFY.C,CMDPKT		:IF DATA VERIFICATION IS REQUIRED:
1600	011464	001407			BEQ	50065\$		
1601	011466	112737	000001	003516	MOVB	#1,VFYFLG		:SET VERIFY FLAG.
1602	011474	042737	000100	002330	BIC	#VFY.C,CMDPKT		:CLEAR VERIFY BIT(NOT USED BY HARDWARE).
1603								:IF DATA VERIFICATION IS NOT REQUIRED:
1604	011502	000402			BR	50066\$		
1605	011504				50065\$:			
1606	011504	105037	003516		CLRB	VFYFLG		:CLR VERIFY FLAG.
1607								
1608	011510				50066\$:			
1609	011510	013737	003420	003424	MOV	CMDWRD,PCMDWD		:SAVE PREVIOUS CMD WORD.
1610	011516	013737	002330	003420	MOV	CMDPKT,CMDWRD		:SAVE PRESENT CMD WORD.
1611	011524	105737	003520		TSTB	SWBFLG		:IF SWAP BYTES IS ENABLED:
1612	011530	001403			BEQ	50067\$		
1613	011532	052737	010000	002330	BIS	#SWB.C,CMDPKT		:SET SWAP BIT IN COMMAND.
1614								
1615	011540				50067\$:			
1616	011540	042737	004000	002330	BIC	#BRF.C,CMDPKT		:CLR BRF BIT (INTERNAL ONLY).
1617	011546	013737	002330	003422	MOV	CMDPKT,CMSAV		:SAVE 1ST WORD OF COMMAND PACKET.
1618	011554	000207			RTS	PC		:RETURN.

```

1619
1620
1621
1622
1623
1624
1625
1626
1627 011556 013703 003446
1628 011562 006303
1629 011564 013704 003416
1630 011570 005204
1631 011572 042704 000001
1632 011576 162704 000002
1633 011602 013702 003406
1634 011606 062702 000002
1635 011612 004773 011620
1636 011616 000207
1637
1638
1639
1640
1641 011620 011642
1642 011622 011700
1643 011624 011720
1644 011626 011730
1645 011630 011754
1646 011632 011766
1647 011634 012000
1648 011636 012020
1649 011640 012052
1650
1651
1652
1653 011642 012703 000400
1654 011646 162704 000002
1655 011652 100411
1656 011654 010322
1657 011656 062703 001002
1658 011662 020327 001000
1659 011666 001002
1660 011670 012703 000400
1661
1662 011674
1663 011674 000764
1664
1665 011676 000207
1666
1667
1668
1669 011700 012703 177777
1670 011704 162704 000002
1671 011710 100402
1672 011712 010322
1673 011714 000773
1674
1675 011716 000207

```

```

:      THIS SUBROUTINE SETS UP AND CALLS THE APPROPRIATE SUBROUTINE TO GENERATE
:      THE DESIRED PATTERN FOR THE WRITE AND WRITE/VERIFY COMMANDS.
:      INPUTS:
:      OUTPUTS:
:      REGISTERS:      R2, R3, R4.
:      CALLS:          PATR0 - PATR7

GENPAT:  MOV     PATERN,R3      ;SETUP PATTERN ROUTINE POINTER
        ASL     R3
        MOV     BRFCNT,R4      ;SET LENGTH OF WRITE BFR
        INC     R4
        BIC     #1,R4          ;ROUNDED UP TO NEXT WORD
        SUB     #2,R4          ;WITH FIRST WORD RESERVED
        MOV     DATAW,R2      ;FOR RECORD COUNT
        ADD     #2,R2
        JSR     PC,@PATBL(R3)  ;GO GENERATE THE APPROPRIATE PATTERN.
        RTS     PC             ;RETURN TO SETUP SUBROUTINE.

;TS05 WRITE PATTERN LOOKUP TABLE. USED TO JSR TO THE
;CORRECT DATA PATTERN GENERATING ROUTINE.

PATBL:  PATR0
        PATR1
        PATR2
        PATR3
        PATR4
        PATR5
        PATR6
        PATR7
        PATR8

;INCREMENTING PATTERN. 0 - 377.

PATR0:: MOV     #400,R3;LET R3 := #400
1$:      SUB     #2,R4;LET R4 := R4 - #2 ;DECREMENT WORD COUNT.
        BMI     2$
        MOV     R3,(R2)+      ;STORE DATA WORD.
        ADD     #1002,R3      ;UPDATE PATTERN.
        CMP     R3,#1000      ;IF PATTERN HAS WRAPPED AROUND THEN:
        BNE     50070$
        MOV     #400,R3      ;INIT THE PATTERN AGAIN.

50070$:  BR      1$           ;DO IT AGAIN.

2$:      RTS     PC          ;RETURN.

;ALL ONE'S PATTERN.

PATR1:: MOV     #-1,R3
ZROPAT: SUB     #2,R4
        BMI     1$
        MOV     R3,(R2)+
        BR      ZROPAT
        ;ALL ONES PATTERN;.
        ;DECREMENT BYTE COUNT.
        ;DONE?,BR IF YES.
        ;IF NOT LOAD NEXT BYTE WITH PATTERN.
        ;DO IT AGAIN.

1$:      RTS     PC          ;RETURN.

```

```

1676
1677
1678
1679 011720 005003
1680 011722 004737 011704
1681 011726 000207
1682
1683
1684
1685 011730 012703 000401
1686 011734 162704 000002
1687 011740 100404
1688 011742 010322
1689 011744 006303
1690 011746 005503
1691 011750 000771
1692 011752 000207
1693
1694
1695
1696 011754 012703 177376
1697 011760 004737 011734
1698 011764 000207
1699
1700
1701
1702
1703 011766 012703 125125
1704 011772 004737 011704
1705 011776 000207
1706
1707
1708
1709 012000 012703 177400
1710 012004 162704 000002
1711 012010 100402
1712 012012 010322
1713 012014 000773
1714 012016 000207
1715
1716
1717
1718 012020 162704 000002
1719 012024 100411
1720 012026 063737 003434 003432
1721 012034 063737 003432 003434
1722 012042 013722 003434
1723 012046 000764
1724 012050 000207
1725
1726
1727
1728 012052 000207
1729
1730
1731
1732

;ALL ZEROES PATTERN.
PATR2:: CLR R3 ;CLR PATTERN REGISTER.
        JSR PC,ZROPAT ;GO GENERATE IT.
        RTS PC ;RETURN.

;ONE BIT WALKING FROM R TO L IN A FIELD OF ZEROES.
PATR3:: MOV #401,R3 ;INIT PATTERN REGISTER.
WLKZRO: SUB #2,R4;LET R4 := R4 - #2 ;DECREMENT WORD COUNT.
        BMI 1$ ;BR IF DONE.
        MOV R3,(R2)+ ;LOAD DATA.
        ASL R3 ;SHIFT PATTERN.
        ADC R3 ;ADD CARRY BACK INTO PATTERN.
        BR WLKZRO ;DO IT AGAIN.
1$: RTS PC ;RETURN.

;ZERO BIT WALKING FROM R TO L IN A FIELD OF 1'S.
PATR4:: MOV #177376,R3 ;INIT PATTERN REGISTER.
        JSR PC,WLKZRO ;GO GENERATE IT.
        RTS PC ;RETURN.

;ALTERNATING ONE AND ZERO BITS WITH ALTERNATE BYTES
;COMPLEMENTED.
PATR5:: MOV #125125,R3 ;INIT PATTERN REGISTER.
        JSR PC,ZROPAT ;GO GENERATE IT.
        RTS PC ;RETURN.

;ALTERNATING BYTES OF 000 AND 377.
PATR6:: MOV #177400,R3 ;INIT PATTERN REGISTER.
1$: SUB #2,R4 ;DECREMENT WORD COUNT.
        BMI 2$ ;BR IF DONE.
        MOV R3,(R2)+ ;LOAD DATA.
        BR 1$ ;DO IT AGAIN.
2$: RTS PC ;RETURN.

;RANDOM PATTERN GENERATOR
PATR7:: SUB #2,R4 ;DECREMENT WORD COUNT
        BMI GIT ;BR IF DONE.
        ADD RANS,RANB ;GET NEW #.
        ADD RANB,RANS ;SAVE #.
        MOV RANS,(R2)+ ;CONTINUE.
        BR PATR7 ;RETURN
GIT: RTS PC ;RETURN

; NO PATTERN GENERATION.
PATR8:: RTS PC ;RETURN.

; THIS SUBROUTINE INITIATES TS05 COMMAND EXECUTION
; AND CHECKS FOR TS05 RESPONSE.
; INPUTS:

```



```

1733      :      OUTPUTS:
1734      :      REGISTERS:      R2, R3.
1735      :      CALLS:      DROP, MOVMSG, FIRSTU, NEXTU, WSSR.
1736
1737 012054 012737 177777 003436 EXCUTE:: MOV # -1, TIME1      ;INIT TIMEOUT COUNTER.
1738 012062 50071$: ;REPEAT      ;WAIT -
1739 012062 005337 003436 DEC TIME1      ;UPDATE TIMEOUT COUNTER.
1740 012066 005737 003436 TST TIME1      ;IF TIMED OUT:
1741 012072 001011 BNE 50072$      ;MOVE CURRENT PACKET MSG.
1742 012074 004737 012734 JSR PC, MOVMSG      ;REPORT TS05 NOT READY
1743 012100 ERRDF 2, NSSRM, STAERM
      012100 104455
      012102 000002
      012104 004536
      012106 006120
1744 012110 004737 017156 JSR PC, DROP
1745 012114 000522 BR EXCRN      ;DROP THE UNIT.
      :RETURN.
1746
1747 012116 50072$:
1748 012116 032775 000200 002524 BIT #TS, SSR, @TSSR(R5)      ;WAIT UNTIL DEVICE IS READY.
1749 012124 001756 BEQ 50071$
1750 012126 023727 003420 140004 CMP CMDWRD, #SCH      ;IF WE ARE DOING A SET CHAR CMD THEN:
1751 012134 001022 BNE 50073$
1752 012136 010537 003452 MOV R5, RSSAVE      ;SAVE CURRENT DEVICE POINTER.
1753 012142 004737 017060 JSR PC, FIRSTU      ;FIND FIRST UNIT.
1754 012146 50074$:
1755 012146 026527 002604 177777 CMP DEVTBL(R5), #END      ;WHILE DEVTBL(R5) NE #END DO
1756 012154 001405 BEQ 50075$
1757 012156 004737 012700 JSR PC, WSSR      ;WAIT FOR UNIT READY OR TIME OUT,
1758 012162 004737 017126 JSR PC, NEXTU      ;FIND NEXT UNIT.
1759
1760 012166 000767 BR 50074$
1761 012170 50075$:
1762 012170 013705 003452 MOV RSSAVE, R5      ;RESTORE CURRENT DEVICE POINTER.
1763 012174 016537 002544 002474 MOV MSGPKA(R5), SCHBK      ;SET UP ADR OF MSG PKT IN SCH BLOCK.
1764
1765 012202 50073$:
1766 012202 016503 002544 MOV MSGPKA(R5), R3      ;ADR OF THIS UNIT'S MSG PACKET.
1767 012206 005002 CLR R2      ;CLR COUNTER.
1768 012210 50076$:
1769 012210 020227 000020 CMP R2, #MSGCNT      ;WHILE THERE ARE MORE LOCATIONS:
1770 012214 001405 BEQ 50077$
1771 012216 012723 177777 MOV # -1, (R3)+      ;INIT THE MSG PACKET WITH ALL 1'S
1772 012222 062702 000002 ADD #2, R2      ;UPDATE COUNTER.
1773
1774 012226 000770 BR 50076$
1775 012230 50077$:
1776 012230 105737 002212 TSTB DINT      ;ARE INTERRUPTS DISABLED.
1777 012234 001023 BNE 1$      ;BR IF YES.
1778 012236 126527 003472 000001 CMPB INTFLG(R5), #1      ;IF MORE THAN ONE INTERRUPT HAS OCCURED:
1779 012244 003412 BLE 50100$
1780 012246 017537 002524 003454 MOV @TSSR(R5), TSSREG      ;FREEZE THE CURRENT STATUS REG FOR PRINT
1781 012254 ERRDF 15, TOOMM, STAERM      ;REPORT TOO MANY INTERRUPTS.
      012254 104455
      012256 000017
      012260 004727
      012262 006120
      TRAP C$ERDF
      .WORD 15
      .WORD TOOMM
      .WORD STAERM

```

```

1782 012264 004737 017156      JSR      PC,DROPU      ;DROP THE UNIT
1783 012270 000434              BR          EXCRTN      ;RETURN - UNIT HAS BEEN DROPPED.
1784
1785 012272                    50100$:
1786 012272 005065 003472      CLR      INTFLG(R5)      ;CLR INTERRUPT FLAG FOR THIS DEV.
1787 012276 052737 000200 002330  BIS      #IE.C,CMDPKT      ;SET INT ENABLE BIT.
1788 012304 105737 003471      TSTB     ERRREC;IFB ERRREC EQ #0 THEN ;IF NOT RETRYING
1789 012310 001005              BNE      50101$
1790 012312 005265 003376      INC      RECCNT(R5)      ;LET RECCNT(R5) := RECCNT(R5) + #1
1791 012316 016577 003376 171062  MOV      RECCNT(R5),@DATAWT ;THEN UPDATE REC COUNT TO WRITE IT ON TAPE
1792
1793 012324                    50101$:
1794 012324 012775 002330 002514  MOV      #CMDPKT,@TSDB(R5)      ;LOAD TSDB WITH CMDPKT ADDRESS
1795                                ;THIS INITIATES COMMAND EXECUTION.
1796 012332 032775 000200 002524  BIT      #TS.SSR,@TSSR(R5)      ;IF READY DID NOT DROP THEN:
1797 012340 001410              BEQ      50102$
1798 012342 004737 012734      JSR      PC,MOVMSG      ;MOVE CURRENT MESSAGE PACKET TO COMMON.
1799 012346              ERRDF  3,TOERM,STAERM      ;REPORT NO TS05 RESPONSE.
                                TRAP      C$ERDF
                                .WORD     3
                                .WORD     TOERM
                                .WORD     STAERM
1800 012356 004737 017156      JSR      PC,DROPU      ;DROP THE UNIT
1801
1802 012362                    50102$:
1803 012362 000207      EXCRTN: RTS      PC      ;RETURN.
1804
1805      ;      THIS SUBROUTINE WAITS FOR THE TS05 INERRUPT OR DONE BIT TO SET AND ALLOWS THE
1806      ;      OPERATOR TO TRANSFER CONROL TO THE SUPERVISOR.
1807      ;      UPON APPEARANCE OF THE INTERRUPT OR DONE, CHECK TSSR FOR STATUS ERRORS,
1808      ;      LOG BYTES AND ERRORS AND PERFORM ERROR RECOVERY IF NESSASARY.
1809      ;      INPUTS:
1810      ;      OUTPUTS:
1811      ;      REGISTERS:      R2, R3.
1812      ;      CALLS:      DROPU, MOVMSG, RECUD, CHKERR, LOG, CLRERR.
1813
1814 012364 012737 177777 003436  GOWAIT:: MOV      #-1,TIME1      ;INIT TIME OUT COUNTER.
1815 012372                    50103$: ;REPEAT
1816 012372              BREAK      ;REPEAT UNTIL INTERRUPT OCCURES:
                                ;GO TO THE SUPER TO ALLOW TTY INPUT.
                                TRAP      C$BRK
1817 012374 023727 003420 102010  CMP      CMDWRD,#RWD      ;IF COMMAND WAS REWIND THEN:
1818 012402 001014              BNE      50104$
1819 012404              DELAY     10.      ;WAIT EXTRA MSECS EACH LOOP.
                                MOV      #10.,(PC)+
                                .WORD     0
                                MOV      L$DLY,(PC)+
                                .WORD     0
                                DEC      -6(PC)
                                BNE      -4
                                DEC      -22(PC)
                                BNE      -20
                                .WORD     0
012404 012727 000012
012410 000000
012412 013727 002116
012416 000000
012420 005367 177772
012424 001375
012426 005367 177756
012432 001367
1820 012434                    50104$:
1821 012434 023727 003420 105010  CMP      CMDWRD,#SFF      ;IF CMDWRD EQ #SFF OR CMDWRD EQ #SFR THEN
1822 012442 001404              BEQ      50105$
1823 012444 023727 003420 105410  CMP      CMDWRD,#SFR
1824 012452 001014              BNE      50106$
1825 012454                    50105$:

```

Address	Offset	Instruction	Comments
1826	012454	DELAY 12.	;ADD DELAY FOR SPACE TAPE MARK COMMANDS
	012454	012727 000014	MOV #12.,(PC)+
	012460	000000	.WORD 0
	012462	013727 002116	MOV L\$DLY,(PC)+
	012466	000000	.WORD 0
	012470	005367 177772	DEC -6(PC)
	012474	001375	BNE -4
	012476	005367 177756	DEC -22(PC)
	012502	001367	BNE -20
1827	012504	50106\$:	
1828	012504	105737 002212	TSTB DINT ;IF INTERRUPTS ARE ENABLED.
1829	012510	001003	BNE 50107\$
1830	012512	016502 003472	MOV INTFLG(R5),R2 ;FETCH INTERRUPT OCCURRED FLAG.
1831			
1832	012516	000406	BR 50110\$
1833	012520	50107\$:	
1834	012520	012703 000200	MOV #TS.SSR,R3 ;SET UP A MASK FOR THE DONE BIT.
1835	012524	005103	COM R3
1836	012526	017502 002524	MOV @TSSR(R5),R2 ;FETCH DONE BIT.
1837	012532	040302	BIC R3,R2
1838			
1839	012534	50110\$:	
1840	012534	005337 003436	DEC TIME1 ;UPDATE TIMEOUT COUNTER.
1841	012540	005702	TST R2 ;REPEAT UNTIL INTERRUPT OR READY OCCURES.
1842	012542	001003	BNE 50111\$
1843	012544	005737 003436	TST TIME1
1844	012550	001310	BNE 50103\$
1845	012552	50111\$:	
1846	012552	005737 003436	TST TIME1 ;IF TIME OUT HAS OCCURRED:
1847	012556	001022	BNE 50112\$
1848	012560	016577 003376 170620	MOV RECCNT(R5),@DATAWT
1849	012566	005377 170614	DEC @DATAWT
1850	012572	004737 012734	JSR PC,MOVMSG ;MOVE CURRENT MSG PACKET TO COMMON AREA.
1851	012576		ERRDF 4,NOINTM,STAERM ;REPORT NO INTERRUPT.
	012576	104455	
	012600	000004	TRAP C\$ERDF
	012602	004670	.WORD 4
	012604	006120	.WORD NOINTM
1852	012606	004737 017156	.WORD STAERM
1853	012612	012703 003472	JSR PC,DROPU ;DROP THE UNIT.
1854	012616	004737 012664	MOV #ENDERF,R3 ;LET R3 := #ENDERF
1855			JSR PC,CLRERR ;CLEAR ALL ERROR FLAGS
1856	012622	000417	BR 50113\$
1857	012624	50112\$:	
1858	012624	004737 012734	JSR PC,MOVMSG ;MOVE CURRENT MSG. PACKET TO COMMON AREA.
1859	012630	004737 013020	JSR PC,RECU D ;UPDATE THE RECORD COUNT.
1860	012634	004737 013166	JSR PC,CHKERR ;CHECK FOR STATUS ERRORS.
1861	012640	105737 003463	TSTB WRTYFG ;IFB WRTYFG EQ #0 THEN
1862	012644	001006	BNE 50114\$
1863	012646	004737 015544	JSR PC,LOG ;LOG BYTES AND ERRORS.
1864	012652	012703 003472	MOV #ENDERF,R3 ;LET R3 := #ENDERF
1865	012656	004737 012664	JSR PC,CLRERR ;CLEAR ALL ERROR FLAGS
1866			
1867	012662	50114\$:	
1868			
1869	012662	50113\$:	
1870	012662	000207	RTS PC ;RETURN IF DONE.

```

1871
1872
1873      :      SUBROUTINE TO CLEAR FLAGS.
1874      :      INPUTS:      R3 = LWA TO BE CLEARED + 2.
1875      :      OUTPUTS:
1876      :      REGISTERS:      R2
1877      :      CALLS:
1878 012664 012702 003460 CLRERR: MOV #BGNFLG,R2      ;LET R2 := #BGNFLG
1879 012670 50115$: REPEAT
1880 012670 005022 CLR (R2)+      ;LET (R2)+ := #0
1881 012672 020203 CMP R2,R3      ;UNTIL R2 EQ R3
1882 012674 001375 BNE 50115$
1883 012676 000207 RTS PC
1884
1885
1886      :      SUBROUTINE TO WAIT UNTIL CURRENT UNIT IS READY OR UNTIL TIME OUT.
1887      :      INPUTS:
1888      :      OUTPUTS:
1889      :      REGISTERS:
1890      :      CALLS:
1891
1892 012700 WSSR:
1893 012700 012737 177777 003436 MOV #-1,TIME1      ;INIT TIMEOUT COUNTER.
1894 012706 50116$: REPEAT UNTIL DEV READY OR TIMEOUT:
1895 012706 BREAK      ;BREAK TO THE SUPERVISOR.
1896 012706 104422 TRAP CSBRK
1897 012710 005337 003436 DEC TIME1      ;UPDATE TIMEOUT COUNTER.
1898 012714 032775 000200 002524 BIT #TS.SSR,@TSSR(R5) ;UNTIL #TS.SSR SET IN @TSSR(R5) OR TIME1 EQ #0
1899 012722 001003 BNE 50117$
1900 012724 005737 003436 TST TIME1
1901 012730 001366 BNE 50116$
1902 012732 50117$: RTS PC      ;RETURN.
1903
1904
1905
1906      :      SUBROUTINE TO MOVE THE CURRENT MESSAGE PACKET TO THE COMMON AREA AND
1907      :      TO UPDATE THE CURRENT TERMINATION CLASS CODE.
1908      :      INPUTS:
1909      :      OUTPUTS:
1910      :      REGISTERS:      R2, R3.
1911      :      CALLS:
1912
1913 012734 017537 002524 003454 MOVMSG: MOV @TSSR(R5),TSSREG      ;FREEZE THE STATUS REG CONTENTS
1914 012742 013702 003454 MOV TSSREG,R2      ;EXTRACT THE TERMINATION CLASS CODE.
1915 012746 042702 177761 BIC #TSC.TCC,R2
1916 012752 010237 003450 MOV R2,CTCC      ;AND SAVE IT
1917 012756 006237 003450 ASR CTCC
1918 012762 016503 002544 MOV MSGPKA(R5),R3      ;ADR OF THIS DEVICE'S MSG.
1919 012766 005002 CLR R2      ;CLR COUNTER.
1920 012770 50120$:
1921 012770 020227 000020 CMP R2,#MSGCNT      ;WHILE THERE ARE MORE LOCATIONS:
1922 012774 001405 BEQ 50121$
1923 012776 012362 002354 MOV (R3)+,MSGPKT(R2)      ;MOVE MSG TO COMMON AREA.
1924 013002 062702 000002 ADD #2,R2      ;UPDATE COUNTER.
1925
1926 013006 000770 BR 50120$

```

```

1927 013010
1928 013010 013737 002362 003502 50121$: MOV MSGPKT+MS.XS0,EOTFLG ;MOVE XSTATO TO EOT FLAG.
1929 013016 000207 RTS PC
1930
1931 : SUBROUTINE TO ADJUST THE RECORD COUNT.
1932 : INPUTS:
1933 : OUTPUTS:
1934 : REGISTERS:
1935 : CALLS:
1936
1937 013020 105737 003465 RECUD:: TSTB RECLOG ;IF RECORD HAS NOT BEEN LOGGED:
1938 013024 001057 BNE 50122$
1939 013026 005365 003376 DEC RECCNT(R5) ;LET RECCNT(R5) := RECCNT(R5) - #1
1940 013032 032737 000001 003450 BIT #BIT0,CTCC ;IF TAPE MOVED
1941 013040 001046 BNE 50123$
1942 013042 032737 100000 002366 BIT #X2.OPM,MSGPKT+MS.XS2
1943 013050 001442 BEQ 50123$
1944 013052 105237 003465 INCB RECLOG ;SET RECORD LOGGED.
1945 013056 023727 003420 102010 CMP CMDWRD,#RWD ;IF THIS IS A REWIND CMD:
1946 013064 001003 BNE 50124$
1947 013066 005065 003376 CLR RECCNT(R5) ;CLEAR RECORD COUNT.
1948
1949 013072 000431 BR 50125$
1950 013074
1951 013074 032737 004000 003420 50124$: BIT #BRF.C,CMDWRD ;IF BRF USED, UPDATE RECORD COUNT.
1952 013102 001425 BEQ 50126$
1953 013104 032737 000400 003420 BIT #MOD.CO,CMDWRD ;IF A FORWARD CMD:
1954 013112 001007 BNE 50127$
1955 013114 032737 000400 003424 BIT #MOD.CO,PCMDWD ;IF PREV CMD WAS A FWD ALSO:
1956 013122 001002 BNE 50130$
1957 013124 005265 003376 INC RECCNT(R5) ;INCREMENT RECORD COUNT.
1958
1959 013130 50130$:
1960
1961 013130 000412 BR 50131$ ;IF REVERSE CMD:
1962 013132
1963 013132 032737 000400 003424 50127$: BIT #MOD.CO,PCMDWD ;IF PREVIOUS CMD WAS A REV ALSO:
1964 013140 001406 BEQ 50132$
1965 013142 032765 000002 003502 BIT #X0.BOT,EOTFLG(R5) ;WHEN NOT AT BOT THEN
1966 013150 001002 BNE 50133$
1967 013152 005365 003376 DEC RECCNT(R5) ;DECREMENT RECORD COUNT.
1968
1969 013156 50133$:
1970
1971 013156 50132$:
1972
1973 013156 50131$:
1974
1975
1976 013156 50126$:
1977
1978 013156 50125$:
1979
1980 013156 50123$:
1981 013156 016577 003376 170222 MOV RECCNT(R5),@DATAWT ;LET @DATAWT := RECCNT(R5)
1982
1983 013164 50122$:

```

```

1984 013164 000207          RTS      PC              ;RETURN.
1985
1986
1987      ;      THIS IS THE ERROR CHECK SUBROUTINE.  AFTER INTERRUPT THIS
1988      ;      SUBROUTINE IS CALLED TO CHECK THE TS05 STATUS.
1989      ;      IF SPECIAL COND IS SET THEN THE TCC HANDLING SUBROUTINE IS ENTERED.
1990      ;      IF THE RFC IS NON ZERO FOR A COMMAND REQUIRING A BPCR,
1991      ;      THEN AN ERROR RFC IS REPORTED,
1992      ;      INPUTS:
1993      ;      OUTPUTS:
1994      ;      REGISTERS:      R2, R4.
1995      ;      CALLS:      TCC0-TCC7.
1996 013166 032737 100000 003454 CHKERR: BIT      #TS.SC,TSSREG          ;IF SPECIAL COND STATUS IS SET THEN:
1997 013174 001441          BEQ      50134$
1998 013176 023727 003450 000002      CMP      CTCC,#2              ;IF TCC IS NOT 2 THEN:
1999 013204 001405          BEQ      50135$
2000 013206 105737 003471          TSTB     ERRREC              ;IF NOT IN ERROR RECOVERY:
2001 013212 001002          BNE      50136$
2002 013214 005265 003336          INC      SCCNT(R5)              ;INC SC COUNTER.
2003
2004 013220          50136$:
2005
2006 013220          50135$:
2007 013220 032737 004000 003454      BIT      #TS.NXM,TSSREG          ;WHEN NON-EXISTANT MEMO
2008 013226 001004          BNE      50137$
2009 013230 032737 040000 003454      BIT      #TS.UPE,TSSREG
2010 013236 001412          BEQ      50140$
2011 013240          50137$:
2012 013240 032737 100000 002366      BIT      #X2.OPM,MSGPKT+MS.XS2      ;AND TAPE NOT MOVED
2013 013246 001003          BNE      50141$
2014 013250 012702 000005          MOV      #5,R2              ;SET TCC5 INDEX
2015
2016 013254 000402          BR      50142$
2017 013256          50141$:
2018 013256 012702 000004          MOV      #4,R2              ;TAPE MOVED, SET TCC4 INDEX
2019
2020 013262          50142$:
2021
2022 013262 000402          BR      50143$
2023 013264          50140$:
2024 013264 013702 003450          MOV      CTCC,R2              ;SET DETECTED TCC INDEX
2025
2026 013270          50143$:
2027 013270 006302          ASL      R2              ;CURRENT TCC X 2.
2028 013272 004772 013372          JSR      PC,@TCCRA(R2)      ;GO TO THE TCC HANDLING SUBROUTINE.
2029
2030 013276 000426          BR      50144$
2031 013300          50134$:
2032 013300 032737 004000 003420      BIT      #BRF.C,CMDWRD          ;IF BRF IS USED IN THIS CMD THEN:
2033 013306 001422          BEQ      50145$
2034 013310 005737 002360          TST      MSGPKT+MS.RFC          ;IF THERE IS AN RFC THEN:
2035 013314 001417          BEQ      50146$
2036 013316 105737 003515          TSTB     RANDOM              ;IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN
2037 013322 001403          BEQ      50147$
2038 013324 105737 003516          TSTB     VFYFLG
2039 013330 001411          BEQ      50150$
2040 013332          50147$:

```

```
2041
2042 013332 105737 003521          TSTB   IRE          ;IF NOT IN RANDOM OR IF CMD IS WTV:
2043 013336 001006                BNE     50151$        ;IF RFC ERROR REPORTS ARE ALLOWED:
2044 013340 005265 003356          INC     HRDCNT(R5)
2045 013344                ERRHRD  13,RFCERM,STAERM      ;UPDATE HARD ERROR COUNT
                                                ;REPORT RFC ERROR
                                                TRAP   C$ERHRD
                                                .WORD  13
                                                .WORD  RFCERM
                                                .WORD  STAERM
2046
2047 013354                50151$:
2048
2049 013354                50150$:
2050
2051 013354                50146$:
2052
2053 013354                50145$:
2054
2055 013354                50144$:
2056 013354 105737 003467          TSTB   RWERR          ;IF A READ/WRITE ERROR HAS OCCURRED THEN:
2057 013360 001403                BEQ     50152$
2058 013362 013737 003422 002330  MOV     CMDSAV,CMDPKT      ;RESTORE CMD PACKET AFTER ERROR RECOV.
2059
2060 013370                50152$:
2061 013370 000207          RTS     PC          ;RETURN.
2062
2063          :      ADDRESSES OF TCC HANDLING ROUTINES FOR TERMINATION CLASS CODES 0 - 7.
2064
2065 013372 013412          TCCRA:  TCC0
2066 013374 013430          TCC1
2067 013376 013446          TCC2
2068 013400 013556          TCC3
2069 013402 013574          TCC4
2070 013404 014210          TCC5
2071 013406 014306          TCC6
2072 013410 014450          TCC7
2073
2074          :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 0, UNDEFINED SPECIAL
2075          :      CONDITION ERROR.
2076          :      INPUTS:
2077          :      OUTPUTS:
2078          :      REGISTERS:
2079          :      CALLS:
2080
2081 013412 005265 003356          TCC0:: INC     HRDCNT(R5)          ;UPDATE HARD ERROR COUNT.
2082 013416                ERRHRD  5,SCERM,STAERM      ;REPORT SPECIAL CONDITION ERROR.
                                                TRAP   C$ERHRD
                                                .WORD  5
                                                .WORD  SCERM
                                                .WORD  STAERM
2083 013426 000207          RTS PC          ;RETURN.
2084
2085
2086          :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 1, ATTENTION CONDITION.
2087          :      THIS TCC INDICATES THAT THE DRIVE HAS UNDERGONE A STATUS CHANGE
2088          :      SUCH AS GOING OFFLINE OR COMING ONLINE.
2089          :      INPUTS:
```

```

2090      :      OUTPUTS:
2091      :      REGISTERS:      R2,R4
2092      :      CALLS:      DROPU
2093
2094      013430      104455      TCC1::  ERRDF      6,ATTNM,STAERM      ;REPORT ATTENTION-UNIT OFF LINE.
      013430      000006      TRAP      C$ERDF
      013432      004603      .WORD      6
      013434      006120      .WORD      ATTNM
      013436      004737      .WORD      STAERM
2095      013440      000207      017156      JSR      PC,DROPU      ;DROP THE UNIT.
2096      013444      000207      RTS      PC      ;RETURN.
2097
2098      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 2, TAPE STATUS ALERT.
2099      :      A STATUS CONDITION HAS BEEN ENCOUNTERED THAT MAY HAVE SIGNIFICANCE
2100      :      TO THE PROGRAM. BITS OF INTEREST INCLUDE TMK, RLS, LET, RLL, BOT, EOT.
2101      :      INPUTS:
2102      :      OUTPUTS:
2103      :      REGISTERS:
2104      :      CALLS:
2105
2106      013446      032737      000002      002362      TCC2::  BIT      #X0.BOT,MSGPKT+MS.XS0
2107      013454      001404      BEQ      50153$
2108      013456      105737      003514      TSTB      EXPBOT
2109      013462      001401      BEQ      50153$
2110
2111      013464      000433      BR      TC2RTN      ;IF AT BOT AND BOT IS EXPECTED:
2112      ;RETURN-TCC2 CAUSED BY EXPECTED BOT.
2113      013466      032737      170002      002362      50153$:  BIT      #X0.RLS!X0.RLL!X0.TMK!X0.LET!X0.BOT,MSGPKT+MS.XS0
2114      013466      001427      ;IF #X0.RLS!X0.RLL!X0.TMK!X0.LET!X0.BOT SETIN MSGPKT+MS.XS0 THEN
2115
2116      013474      001427      BEQ      50154$
2117
2118      013476      105737      003515      TSTB      RANDOM      ;IF TCC2 CAUSED BY ANYTHING BUT EOT:
2119      013502      001403      BEQ      50155$      ;IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN
2120      013504      105737      003516      TSTB      VFYFLG
2121      013510      001421      BEQ      50156$
2122
2123      013512      50155$:
2124
2125      013512      105737      003521      TSTB      IRE      ;IF NOT IN RANDOM OR IF CMD IS WTV:
2126      013516      001016      BNE      50157$      ;IF RFC ERROR REPORTS ARE ALLOWED:
2127      013520      105737      003471      TSTB      ERRREC      ;IF WE ARE IN ERROR RECOVERY THEN:
2128      013524      001403      BEQ      50160$
2129      013526      105237      003470      INCB      UNREC      ;SET UNRECOVERABLE FLAG FOR LOG.
2130      ;ELSE - IF NOT IN ERROR RECOVERY:
2131      013532      000402      BR      50161$
2132      013534      005265      003336      50160$:  INC      SCCNT(R5)      ;INCREMENT THE SPEC COND COUNTER.
2133
2134      013540      005265      003356      50161$:  INC      HRDCNT(R5)      ;UPDATE HARD ERROR COUNT.
2135      013540      004705      ERRHRD      7,TSAM,STAERM      ;REPORT TAPE STATUS ALERT.
2136      013544      104456      TRAP      C$ERHRD
2137      013546      000007      .WORD      7
2138      013550      006120      .WORD      TSAM
      013552      .WORD      STAERM

```



```

2139 013554      50157$:
2140
2141 013554      50156$:
2142
2143 013554      50154$:
2144
2145 013554 000207 TC2RTN:  RTS  PC                      ;RETURN.
2146
2147
2148      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 3, FUNCTION REJECT.
2149      :      THE SPECIFIED FUNCTION WAS NOT INITIATED. BITS OF INTEREST ARE
2150      :      RMR, OFL, VCK, BOT, ILC, WLE, ILA, AND NBA.
2151      :      INPUTS:
2152      :      OUTPUTS:
2153      :      REGISTERS:      R2,R4
2154      :      CALLS:          DROPUP
2155
2156 013556      TCC3:: ERRDF  8,FUNRM,STAERM          ;REPORT FUNCTION REJECT.
      013556 104455
      013560 000010
      013562 004622
      013564 006120
2157 013566 004737 017156 JSR  PC,DROPUP          ;DROP THE UNIT.
2158 013572 000207 RTS  PC                      ;RETURN.
2159
2160      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 4, RECOVERABLE ERROR.
2161      :      TAPE POSITION IS ONE RECORD BEYOND WHAT ITS POSITION WAS WHEN
2162      :      THE FUNCTION WAS INITIATED. RECOVERY PROCEDURE IS TO LOG THE
2163      :      ERROR AND ISSUE THE APPROPRIATE RETRY COMMAND.
2164      :      2 WRITE-ERROR-RECOVERY ALGORITHMS CAN BE SELECTED:
2165      :      THE FIRST ONE, VIA BADTSW SWITCH, DOES DETECT BAD SPOTS ON TAPE.
2166      :      IT CALLS A WRITE RETRY SUBR UNTIL THE RECORD IS RECOVERED
2167      :      OR 20 BAD SPOTS HAVE BEEN LOGGED. ON REACHING 20 BAD
2168      :      SPOTS LOGGED, A BAD TAPE OVERFLOW MSG IS PRINTED AND THE
2169      :      UNIT DROPPED.
2170      :      THE SECOND ALGORITHM ISSUES THE TS05 WRITE RETRY COMMAND
2171      :      UP TO 16 TIMES BEFORE DROPPING THE UNIT OR PROCEEDING
2172      :      WITH THE NEXT RECORD ON RECOVERY.
2173      :      INPUTS:
2174      :      OUTPUTS:
2175      :      REGISTERS:      R2,R4.
2176      :      CALLS:          RTLE, EXCUTE, GOWAIT, DROPUP, WRTY
2177
2178 013574 023727 003426 000002 TCC4:: CMP  CMDLG,#2      ;IF CMDLG EQ #2 ANDB BADTSW NE #0 THEN
2179 013602 001125 BNE  50162$
2180 013604 105737 002210 TSTB  BADTSW
2181 013610 001522 BEQ  50162$
2182 013612 105737 003471 TSTB  ERRREC      ;IFB ERRREC EQ #0 ANDB ERCVER NE #0 THEN
2183 013616 001007 BNE  50163$
2184 013620 105737 002207 TSTB  ERCVER
2185 013624 001404 BEQ  50163$
2186 013626 ERRSOFT 9,RERM,STAERM ;
      013626 104457
      013630 000011
      013632 005017
      013634 006120
2187

```

```

TRAP  CSERDF
.WORD 8
.WORD FUNRM
.WORD STAERM

```

```

TRAP  CSERSOFT
.WORD 9
.WORD RERM
.WORD STAERM

```

```
2188 013636
2189 013636 105737 002213
2190 013642 001102
2191 013644 105237 003471
2192 013650 105237 003464
2193 013654 105737 003463
2194 013660 001072
2195
2196 013662 013737 003420 015064
2197 013670 013737 002330 015062
2198 013676 013737 002336 015066
2199 013704 105237 003467
2200 013710 105237 003463
2201
2202 013714
2203 013714 005265 003316
2204 013720 005037 003460
2205 013724 105037 003462
2206 013730 004737 014614
2207 013734 105737 003464
2208 013740 001404
2209 013742 027727 167544 000050
2210 013750 103761
2211 013752
2212
2213 013752 027727 167534 000050
2214 013760 103423
2215 013762
    013762 012746 015155
    013766 012746 000001
    013772 010600
    013774 104414
    013776 062706 000004
2216 014002 004737 015274
2217 014006 005365 003376
2218 014012 004737 017156
2219 014016 005065 003376
2220 014022 012775 002350 002514
2221
2222 014030
2223 014030 105037 003463
2224 014034 105237 003531
2225 014040 013737 015064 003424
2226
2227 014046
2228
2229 014046 000402
2230 014050
2231 014050 105237 003470
2232
2233 014054
2234
2235 014054 000454
2236 014056
2237 014056 004737 014466
2238 014062 023727 003426 000002
2239 014070 003411

50163$:
    TSTB IREC ;IFB IREC EQ #0 THEN
    BNE 50164$
    INCB ERRREC ;RETRY FLAG FOR EXECUTE SUBR: DON'T UPDATE REC CN
    INCB WRTYER ;REWRITE ERROR FLAG FOR WRTY SUBR
    TSTB WRTYFG ;FIRST RETRY ON THIS RECORD: SUBSEQUENT
    BNE 50165$
                ;RETRIES WITH TCC4 ERRORS BY-PASS THIS SECTION
    MOV CMDWRD,WTYWRD ;SAVE WRITE COMMAND PACKET
    MOV CMDPKT,WTYCMD
    MOV CMDPKT+CP.CNT,WTYBRF
    INCB RWERR ;LOG SUBR FLAG: COUNT WRT ERRORS
    INCB WRTYFG ;RETRY IN PROGRESS FLAG

50166$:
    INC ;REPEAT
    WRTYCT(R5) ;COUNT GLOBAL WRITE RETRIES
    CLR RETRYC ;CLEAR # OF RETRIES PER RECORD
    CLRB RPTCNT ;CLEAR # OF REPEATS
    JSR PC,WRTY ;CALL WRITE RETRY
    TSTB WRTYER ;REPEAT RETRIES ON SAME RECORD
    BEQ 50167$
    CMP @BTPT,#40.
    BLO 50166$

50167$:
                ;UNTIL RECOVERED OR 20 BAD SPOTS
                ;WHEN 20 BAD SPOTS LOGGED
    CMP @BTPT,#40.
    BLO 50170$
    PRINTB #BTMSG2 ;PRINT BAD TAPE OVERFLOW MSG
                MOV #BTMSG2,-(SP)
                MOV #1,-(SP)
                MOV SP,R0
                TRAP C$PNTB
                ADD #4,SP

    JSR PC,BORERS ;ERASE BAD RECORD
    DEC RECCNT(R5)
    JSR PC,DROPU ;DROP UNIT
    CLR RECCNT(R5)
    MOV #RWCPK,@TSDB(R5) ;REWIND UNIT

50170$:
    CLRB WRTYFG ;RETRY COMPLETE FLAG
    INCB MISCFG ;DO NOT HALT ON THIS CMD FLG
    MOV WTYWRD,PCMDWD ;RESTORE ORIGINAL WRT CMD AFTER RECOVERY

50165$:
    BR 50171$

50164$:
    INCB UNREC ;LET UNREC :B= UNREC + #1 ;

50171$:
    BR 50172$

50162$:
    JSR PC,RTLE ;CHECK FOR RETRY LIMIT EXCEEDED.
    CMP CMDLG,#2 ;IF READ CMD THEN:
    BLE 50173$
```

```

2240 014072 012702 000020      MOV      #RRECL,R2      ;R2=READ RETRY COUNT LIMIT / 2
2241 014076 006202              ASR      R2
2242 014100 023702 003460      CMP      RETRYC,R2      ;IF RETRY COUNT IS MORE THAN HALF LIMIT:
2243 014104 002403              BLT      50174$
2244 014106 052737 020000 002330  BIS      #OPP.C,CMDPKT      ;SET OPPOSITE BIT FOR RETRY2.
2245
2246 014114              50174$:
2247
2248 014114              50173$:
2249 014114 005737 003460      TST      RETRYC      ;IF THIS IS THE ORIGINAL ERROR THEN:
2250 014120 001007              BNE      50175$
2251 014122 105737 002207      TSTB     ERVER
2252 014126 001404              BEQ      50175$
2253 014130              ERRSOFT 9,RERM,STAERM ;REPORT RECOVERABLE ERROR
      014130 104457
      014132 000011
      014134 005017
      014136 006120
2254
2255 014140              50175$:
2256 014140 005237 003460      INC      RETRYC      ;UPDATE RETRY COUNT.
2257 014144 052737 001000 002330  BIS      #MOD.C1,CMDPKT ;SET RETRY BIT IN CMD PACKET.
2258 014152 105737 002213      TSTB     IREC      ;IF ERROR RECOVERY ENABLED:
2259 014156 001011              BNE      50176$
2260 014160 105237 003471      INCB     ERRREC      ;SET ERROR RECOVERY FLAG.
2261 014164 012602              MOV      (SP)+,R2      ;POP 2 RTN ADRS FROM STACK.
2262 014166 012602              MOV      (SP)+,R2
2263 014170 004737 012054      JSR      PC,EXCUTE
2264 014174 000137 012364      JMP      GOWAIT
2265
2266 014200 000402              BR       50177$
2267 014202
2268 014202 105237 003470      50176$:
2269
2270 014206              50177$:
2271
2272 014206              50172$:
2273 014206 000207      RTS PC      ;RETURN
2274
2275      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 5, RECOVERABLE ERROR.
2276      :      TAPE POSITION HAS NOT CHANGED. RECOVERY PROCEDURE IS TO LOG THE
2277      :      ERROR AND RE-ISSUE THE ORIGINAL COMMAND.
2278      :      INPUTS:
2279      :      OUTPUTS:
2280      :      REGISTERS:      R2,R4.
2281      :      CALLS:      RTLE, EXCUTE, GOWAIT, DROP.
2282
2283 014210 004737 014466      TCC5:: JSR      PC,RTLE      ;CHECK FOR RETRY LIMIT EXCEEDED
2284 014214 005737 003460      TST      RETRYC      ;IF THIS IS THE ORIGINAL ERROR THEN:
2285 014220 001004              BNE      50200$
2286 014222              ERRSOFT 10,RERM,STAERM ;REPORT RECOVERABLE ERROR.
      014222 104457
      014224 000012
      014226 005017
      014230 006120
2287 014232
2288 014232 005237 003460      50200$:
      INC      RETRYC      ;UPDATE RETRY COUNTER.

```

```

TRAP      CSERSOFT
.WORD     9
.WORD     RERM
.WORD     STAERM

```

```

2289 014236 105737 002213      TSTB      IREC      ;IF ERROR RECOVERY IS ENABLED:
2290 014242 001016      BNE      50201$
2291 014244 105237 003471      INCB      ERRREC      ;SET ERROR RECOVERY FLAG.
2292 014250 005265 003376      INC      RECCNT(R5) ;UPDATE REC COUNT
2293 014254 016577 003376 167124 MOV      RECCNT(R5),@DATAWT ;AND INSERT IT INTO WRT BFR
2294 014262 012602      MOV      (SP)+,R2 ;POP 2 RTN ADRS FROM STACK.
2295 014264 012602      MOV      (SP)+,R2
2296 014266 004737 012054      JSR      PC,EXCUTE ;GO RE-ISSUE THE COMMAND.
2297 014272 000137 012364      JMP      GOWAIT ;GO WAIT FOR INTERRUPT + CHECK STATUS.
2298                                     ;ELSE IF ERROR RECOVERY IS NOT ENABLED:
2299 014276 000402      BR      50202$
2300 014300      50201$:      INCB      UNREC      ;SET UNRECOVERABLE ERROR FLAG.
2301 014300 105237 003470
2302                                     50202$:
2303 014304      RTS      PC      ;RETURN.
2304 014304 000207
2305
2306                                     :
2307                                     SUBROUTINE TO HANDLE TERMINATION CLASS CODE 6, UNRECOVERABLE ERROR.
2308                                     TAPE POSITION HAS BEEN LOST. THE ONLY VALID RECOVERY PROCEDURE
2309                                     IS TO REWIND AND START OVER AT BOT UNLESS THE TAPE HAS LABELS OR
2310                                     SEQUENCE NUMBERS. THIS DIAGNOSTIC WILL REWIND AND RETRY THE
2311                                     COMMAND ONLY IF DENSITY CHECK IS SET, OTHERWISE THE UNIT WILL BE
2312                                     DROPPED FROM THE TEST SEQUENCE.
2313                                     INPUTS:
2314                                     OUTPUTS:
2315                                     REGISTERS:      R2, R4
2316                                     CALLS:      RTLE, WSSR, EXCUTE, GOWAIT, DROPU
2317
2318 014306 033737 000010 002370 TCC6:: BIT      X3.DCK,MSGPKT+MS.XS3;IF X3.DCK NOTSETIN MSGPKT+MS.XS3 THEN
2319 014314 001016      BNE      50203$
2320                                     ;IF THERE IS NO DENSITY CHECK THEN:
2321 014316 005737 003426      TST      CMDLG      ;IF CMD IS A READ OR WRITE THEN:
2322 014322 001404      BEQ      50204$
2323 014324 105237 003467      INCB      RWERR
2324 014330 105237 003470      INCB      UNREC      ;SET RD/WR ERROR FLAG,
2325                                     ;SET UNRECOVERABLE ERROR FLAG.
2326 014334      50204$:
2327 014334      ERRDF      11,URERM,STAERM ;REPORT UNRECOVERABLE ERROR.
2328 014334 104455      TRAP      C$ERDF
2329 014336 000013      .WORD      11
2330 014340 005041      .WORD      URERM
2331 014342 006120      .WORD      STAERM
2332 014344 004737 017156      JSR      PC,DROPU ;REPORT ERROR + DROP UNIT.
2333                                     ;ELSE-IF THERE IS DENSITY CHECK:
2334 014350 000436      BR      50205$
2335 014352      50203$:
2336 014352 004737 014466      JSR      PC,RTLE ;CHECK FOR RETRY LIMIT EXCEEDED.
2337 014356 005737 003460      TST      RETRYC ;IF THIS IS THE ORIGINAL ERROR THEN:
2338 014362 001004      BNE      50206$
2339 014364      ERRSOFT 11,URERM,STAERM ;REPORT DENSITY CHECK ERROR
2340 014364 104457      TRAP      C$ERSOFT
2341 014366 000013      .WORD      11
2342 014370 005041      .WORD      URERM
2343 014372 006120      .WORD      STAERM
2344 014374      50206$:

```

```
2338 014374 005237 003460      INC      RETRYC      ;UPDATE RETRY COUNT.
2339 014400 105737 003521      TSTB     IRE        ;IF ERROR RECOVERY IS ENABLED THEN:
2340 014404 001016              BNE      50207$
2341 014406 105237 003471      INCB     ERRREC      ;SET ERROR RECOVERY FLAG,
2342 014412 012775 002350      MOV      #RWCPK,@TSDB(R5) ;ISSUE A REWIND COMMAND,
2343 014420 004737 012700      JSR      PC,WSSR    ;WAIT FOR SUBSYSTEM READY,
2344 014424 012602              MOV      (SP)+,R2    ;POP 2 RTN ADRS FROM STACK.
2345 014426 012602              MOV      (SP)+,R2
2346 014430 004737 012054      JSR      PC,EXCUTE
2347 014434 000137 012364      JMP      GOWAIT    ;REISSUE THE COMMAND,
2348                                ;WAIT FOR INTERRUPT
2349 014440 000402              BR       50210$      ;ELSE-IF ERR REC DISABLED:
2350 014442              50207$:
2351 014442 105237 003470      INCB     UNREC      ;SET UNRECOVERABLE ERROR FLAG.
2352
2353 014446              50210$:
2354
2355 014446              50205$:
2356 014446 000207      RTS      PC                ;RETURN
2357
2358      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 7, FATAL SUBSYSTEM
2359      :      ERROR. THE SUBSYSTEM IS INCAPABLE OF PROPERLY PERFORMING
2360      :      COMMANDS OR AT LEAST ITS INTEGRITY IS SERIOUSLY QUESTIONABLE.
2361      :      REFER TO THE FATAL CLASS CODE FIELD IN THE TSSR REGISTER FOR
2362      :      ADDITIONAL INFORMATION ON THE TYPE OF FATAL ERROR.
2363      :      INPUTS:
2364      :      OUTPUTS:
2365      :      REGISTERS:      R2, R4
2366      :      CALLS:
2367
2368 014450      TCC7:: ERRDF 12,FATSM,STAERM      ;REPORT FATAL SUBSYSTEM ERROR.
2369 014450 104455              TRAP      C$ERDF
2370 014452 000014              .WORD    12
2371 014454 004642              .WORD    FATSM
2372 014456 006120              .WORD    STAERM
2373
2374 014460 004737 017156      JSR      PC,DROPU    ;DROP THE UNIT.
2375 014464 000207      RTS      PC                ;RETURN.
2376
2377      :      SUBROUTINE TO CHECK FOR RETRY LIMIT EXCEEDED. PRINTS ERROR MESSAGE
2378      :      IF EXCEEDED AND DROP UNIT UNLESS COMMAND IS A READ.
2379      :      INPUTS:
2380      :      OUTPUTS:
2381      :      REGISTERS:      R2, R4.
2382      :      CALLS:      DROPU
2383
2384 014466 005737 003426      RTLE:: TST      CMDLG      ;IF CMD IS NOT A READ OR WRITE THEN:
2385 014472 001010      BNE      50211$
2386 014474 104455      ERRDF 11,URERM,STAERM ;REPORT UNRECOVERABLE ERROR.
2387 014476 000013              TRAP      C$ERDF
2388 014500 005041              .WORD    11
2389 014502 006120              .WORD    URERM
2390 014504 004737 017156      JSR      PC,DROPU    ;DROP THE UNIT.
2391 014510 012602      MOV      (SP)+,R2    ;POP RTN ADRS FROM STACK.
2392 014512 000437      BR       RTLRTN      ;AND RETURN.
2393
```

```

2387 014514      50211$:
2388 014514      105237 003467      INCB  RWERR      ;SET READ/WRITE ERROR FLAG.
2389 014520      023727 003426 000002      CMP  CMDLG,#2      ;IF CMD IS A WRT OR WTM:
2390 014526      001016      BNE  50212$
2391 014530      023727 003460 000020      CMP  RETRYC,#WRECL ;IF RETRY COUNT HAS REACHED LIMIT:
2392 014536      001011      BNE  50213$
2393 014540      105237 003470      INCB  UNREC      ;SET UNRECOVERABLE FLAG
2394 014544      104455      ERRDF 14,RLEXM,STAERM ;REPORT RETRY LIMIT EXCEEDED.
                014546      000016      TRAP  C$ERDF
                014550      004556      .WORD 14
                014552      006120      .WORD RLEXM
2395 014554      004737 017156      .WORD STAERM
2396 014560      012602      JSR  PC,DROPU      ;DROP THE UNIT.
2397 014562      50213$:      MOV  (SP)+,R2      ;POP 2 RTN ADRS FROM STACK.
2398 014562      000413      BR  50214$      ;ELSE - CMD IS A READ:
2400 014564      50212$:      CMP  RETRYC,#RRECL ;IF RETRY COUNT HAS REACHED LIMIT:
2401 014564      023727 003460 000020      BNE  50215$
2402 014572      001007      INCB  UNREC      ;SET UNRECOVERABLE FLAG
2403 014574      105237 003470      ERRHRD 14,RLEXM,STAERM ;REPORT RECOVERABLE ERROR.
2404 014600      104456      TRAP  C$ERHRD
                014602      000016      .WORD 14
                014604      004556      .WORD RLEXM
                014606      006120      .WORD STAERM
2405 014610      012602      MOV  (SP)+,R2      ;POP 2 RTN ADRS FROM STACK.
2406 014612      50215$:
2407 014612      50214$:
2408 014612      000207      PTLRTN: RTS  PC      ;RETURN
2409 014612
2410
2411      :      SUBR TO REWRITE A BAD, BUT RECOVERABLE WRITTEN RECORD.
2412      :      REWRITE RECORD ON SAME SPOT: REPEAT 4 TIMES.
2413      :      IF ALL 4 REPEATS GOOD, RECORD IS RECOVERED
2414      :      AND A RECOVERABLE WRITE ERROR IS LOGGED.
2415      :      IF ANY OF 4 REPEATS BAD, ERASE BAD RECORD, LOG SUSPECTED
2416      :      BAD SPOT, RETRY AGAIN. RETRY 4 TIMES, UP TO 4 REPEATS EACH.
2417      :      IF RECORD NOT GOOD AFTER 4 RETRIES, ERASE IT, EXIT WITH
2418      :      ERROR FLAG WRTYER SET, PRINTING RETRY FAILED.
2419      :      THIS ALL SCHEME IS REENTERED 20 TIMES MAX, IE 20 BAD
2420      :      SPOTS MAX ARE ALLOWED.
2421      :
2422      :      INPUTS:
2423      :      OUTPUTS:
2424      :      REGISTERS:      R3,R4
2425      :      CALLS:      BORERS, REWRT
2426
2427 014614      WRTY:: ;BEGIN RETRY      ;REPEAT
2428
2429 014614      50217$:
2430      :      ;BEGIN REPEAT      ;REPEAT
2431
2432 014614      50221$:
2433 014614      004737 015274      JSR  PC,BORERS      ;BACKSPACE/ERASE ONE RECORD
2434 014620      105037 003464      CLR8  WRTYER      ;CLEAR WRITE RETRY ERROR
2435 014624      004737 015450      JSR  PC,REWRT      ;REWRITE RECORD ON SAME SPOT

```

```

2436 014630 105237 003462          INCB      RPTCNT          ;COUNT REPEATS
2437 014634 123727 003462 000004    CMPB      RPTCNT,#4      ;LIMIT: 4 REPEATS OR RECOVERED
2438 014642 001403                    BEQ       50222$
2439 014644 105737 003464          TSTB      WRTYER
2440 014650 001761                    BEQ       50221$
2441 014652          50222$:
2442          ;END REPEAT
2443 014652          50220$:
2444 014652 005237 003460          INC        RETRYC          ;COUNT RETRIES
2445 014656 105737 003464          TSTB      WRTYER
2446 014662 001001                    BNE       50223$
2447 014664 000457                    BR        50216$          ;EXIT RETRY LOOP IF RECOVERED
2448
2449 014666          50223$:
2450 014666 105737 002207          TSTB      ERCVER          ;IFB ERCVER NE #0 THEN
2451 014672 001415                    BEQ       50225$
2452 014674          PRINTB      #BTMSG1,RETRYC,<B,RPTCNT>      ;PRINT SUSPECTED BAD SPOT
2453 014674 005046                    CLR        -(SP)
2454 014676 153716 003462          BISB      RPTCNT,(SP)
2455 014702 013746 003460          MOV       RETRYC, -(SP)
2456 014706 012746 015070          MOV       #BTMSG1, -(SP)
2457 014712 012746 000003          MOV       #3, -(SP)
2458 014716 010600          MOV       SP,R0
2459 014720 104414          MOV       SP,R0
2460 014722 062706 000010          TRAP      C$PNTB
2461 014726          ADD        #10,SP
2462 014726 023727 003460 000001    50225$:
2463 014734 001021                    CMP        RETRYC,#1          ;ON FIRST RETRY, LOGG BAD SPOT
2464 014736 016537 002616 003512    BNE       50226$
2465 014744 017704 166542          MOV       BTADDR(R5),BTPT          ;BTPT IS BOTH THE BAD SPOT COUNTER
2466 014750 062704 000002          MOV       @BTPT,R4          ;AND THE LOGGING INDEX
2467 014754 010477 166532          ADD        #2,R4
2468 014760 020427 000050          MOV       R4,@BTPT
2469 014764 101005                    CMP        R4,#40          ;IF R4 LOS #40. THEN
2470 014766 013703 003512          BHI       50227$
2471 014772 060304          MOV       BTPT,R3          ;STORE FIRST 20 BAD SPOTS
2472 014774 016514 003376          ADD        R3,R4          ;LET R4 := R4 + R3
2473 015000          MOV       RECCNT(R5),(R4)          ;LET (R4) := RECCNT(R5)
2474 015000          50227$:
2475 015000          50226$:
2476 015000 105237 003525          INCB      ERSFLG          ;ERASE FLAG TO ERASE BAD RECORD
2477 015004 105037 003467          CLRB      RWERR          ;CANCELL 'LOG' ERROR FLAG ON FAILING RET
2478 015010 105037 003462          CLRB      RPTCNT          ;CLEAR REPEAT COUNT FOR NEXT RETRY
2479
2480 015014          50224$:
2481 015014 023727 003460 000004    CMP        RETRYC,#4          ;LIMIT: 4 RETRIES
2482 015022 001274                    BNE       50217$
2483          ;END RETRY
2484 015024          50216$:
2485 015024 105737 003464          TSTB      WRTYER          ;IFB WRTYER NE #0 THEN
2486 015030 001413                    BEQ       50230$
2487 015032 105737 002207          TSTB      ERCVER          ;IFB ERCVER NE #0 THEN
2488 015036 001410                    BEQ       50231$
2489 015040          PRINTB      #BTMSG3          ;PRINT RETRY FAILED
2490 015044 012746 015225          MOV       #BTMSG3, -(SP)
2491          015044 012746 000001    MOV       #1, -(SP)

```

	015050	010600				
	015052	104414				
	015054	062706	000004			MOV SP,RO TRAP C\$PNTB ADD #4,SP
2483						
2484	015060			50231\$:		
2485						
2486	015060			50230\$:		
2487	015060	000207		RTS PC		
2488						
2489	015062	000000		WTYCMD: .WORD 0		:STORAGE FOR WRITE CMD WHILE RETRYING
2490	015064	000000		WTYWRD: .WORD 0		:STORAGE FOR WRITE CMD WORD WHILE RETRYING
2491	015066	000000		WTYBRF: .WORD 0		:STORAGE FOR WRITE BPCR WHILE RETRYING
2492						
2493	015070	045	101	123	BTMSG1: .ASCIZ	/XASUSPECT BAD SPOT AFTER %D1XA RETRY, %D1XA REPEATXN/
	015073	125	123	120		
	015076	105	103	124		
	015101	040	102	101		
	015104	104	040	123		
	015107	120	117	124		
	015112	040	101	106		
	015115	124	105	122		
	015120	040	045	104		
	015123	061	045	101		
	015126	040	122	105		
	015131	124	122	131		
	015134	054	040	045		
	015137	104	061	045		
	015142	101	040	122		
	015145	105	120	105		
	015150	101	124	045		
	015153	116	000			
2494	015155	045	116	045	BTMSG2: .ASCIZ	/XN%ABAD TAPE OVERFLOW: CHANGE TAPE!XN%N/
	015160	101	102	101		
	015163	104	040	124		
	015166	101	120	105		
	015171	040	117	126		
	015174	105	122	106		
	015177	114	117	127		
	015202	072	040	103		
	015205	110	101	116		
	015210	107	105	040		
	015213	124	101	120		
	015216	105	041	045		
	015221	116	045	116		
	015224	000				
2495	015225	045	101	122	BTMSG3: .ASCIZ	/XARETRY FAILED ON BAD SPOT...ERASED!XN/
	015230	105	124	122		
	015233	131	040	106		
	015236	101	111	114		
	015241	105	104	040		
	015244	117	116	040		
	015247	102	101	104		
	015252	040	123	120		
	015255	117	124	056		
	015260	056	056	105		
	015263	122	101	123		
	015266	105	104	041		


```

015271 045 116 000
2496 .EVEN
2497
2498 : SUBR TO BACKSPACE ONE RECORD
2499 : IF THE ERASE FLAG IS SET, THEN ERASE THAT RECORD.
2500 : INPUTS: ERSFLG 1 = DO ERASE
2501 : OUTPUTS:
2502 : REGISTERS:
2503 : CALLS: EXCUTE, GOWAIT, CKHAE
2504
2505 015274 013737 003420 003424 BORERS::MOV CMDWRD,PCMDWD ;SET COMMAND TO SPACE REV
2506 015302 012737 104410 003420 MOV #SRR,CMDWRD ;LET CMDWRD := #SRR
2507 015310 013737 003420 002330 MOV CMDWRD,CMDPKT ;LET CMDPKT := CMDWRD CLR.BY #BRF.C ;
2508 015316 042737 004000 002330 BIC #BRF.C,CMDPKT
2509 015324 013737 002330 003422 MOV CMDPKT,CMSAV ;LET CMSAV := CMDPKT
2510 015332 012737 000001 002332 MOV #1,CMDPKT+CP.ADL ;LET CMDPKT+CP.ADL := #1 ;
2511 015340 005037 003426 CLR CMDLG ;LET CMDLG := #0 ;
2512 015344 004737 011064 JSR PC,CMDAC
2513 015350 004737 012054 JSR PC,EXCUTE
2514 015354 004737 012364 JSR PC,GOWAIT
2515 015360 004737 017456 JSR PC,CKHAE
2516 015364 105737 003525 TSTB ERSFLG ;WHEN ERASE FLAG IS SET, DO ERASE
2517 015370 001426 BEQ 50232$
2518 015372 013737 003420 003424 MOV CMDWRD,PCMDWD ;LET PCMDWD := CMDWRD ;
2519 015400 012737 100411 003420 MOV #ERS,CMDWRD ;LET CMDWRD := #ERS ;
2520 015406 013737 003420 002330 MOV CMDWRD,CMDPKT ;LET CMDPKT := CMDWRD ;
2521 015414 013737 002330 003422 MOV CMDPKT,CMSAV ;LET CMSAV := CMDPKT ;
2522 015422 004737 011064 JSR PC,CMDAC
2523 015426 004737 012054 JSR PC,EXCUTE
2524 015432 004737 012364 JSR PC,GOWAIT
2525 015436 004737 017456 JSR PC,CKHAE
2526 015442 105037 003525 CLRB ERSFLG ;LET ERSFLG := #0
2527
2528 015446 50232$: RTS PC
2529 015446 000207
2530
2531 : SUBR TO REWRITE A BADLY WRITTEN RECORD
2532
2533 015450 013737 003420 003424 REWRT::MOV CMDWRD,PCMDWD ;RESTORE WRITE COMMAND PACKET
2534 015456 013737 015064 003420 MOV WTYWRD,CMDWRD ;LET CMDWRD := WTYWRD ;
2535 015464 013737 015062 002330 MOV WTYCMD,CMDPKT ;LET CMDPKT := WTYCMD ;
2536 015472 013737 002330 003422 MOV CMDPKT,CMSAV ;LET CMSAV := CMDPKT ;
2537 015500 013737 003406 002332 MOV DATAWT,CMDPKT+CP.ADL ;LET CMDPKT+CP.ADL := DATAWT ;
2538 015506 013737 015066 002336 MOV WTYBRF,CMDPKT+CP.CNT ;LET CMDPKT+CP.CNT := WTYBRF ;
2539 015514 012737 000002 003426 MOV #2,CMDLG ;LET CMDLG := #2 ;
2540 015522 004737 011064 JSR PC,CMDAC
2541 015526 004737 012054 JSR PC,EXCUTE ;RE-WRITE RECORD
2542 015532 004737 012364 JSR PC,GOWAIT
2543 015536 004737 017456 JSR PC,CKHAE
2544 015542 000207 RTS PC
2545
2546 : SUBROUTINE TO LOG BYTES READ/WRITTEN.
2547 : ALSO UPDATES READ/WRITE ERROR COUNTERS.
2548 : INPUTS:
2549 : OUTPUTS:
2550 : REGISTERS: R2, R3, R4.
2551 : CALLS:

```

```

2552
2553 015544 105737 003466      LOG::  TSTB  ERLOG      ;IF DATA AND ERRORS HAVE NOT BEEN LOGGED THEN:
2554 015550 001126              BNE    50233$
2555 015552 105237 003466      INCB   ERLOG      ;SET LOG DONE FLAG.
2556 015556 013704 003426      MOV    CMDLG,R4    ;GET CURRENT CMD LOGGING CODE.
2557 015562 005704              RST    R4          ;IF THERE IS A CODE THEN:
2558 015564 001520              BEQ    50234$
2559 015566 162704 000002      SUB     #2,R4      ;ADJUST THE CODE FOR TABLE INDEX.
2560 015572 010502              MOV     R5,R2      ;R2 = ADR OF BYTE COUNT LSW.
2561 015574 066402 016030      ADD     BINC(R4),R2
2562 015600 062702 002626      ADD     #CNTBGN,R2
2563 015604 063712 003416      ADD     BRFCNT,(R2)      ;ADD BRFCNT TO LSW.
2564 015610 023737 002360 003416  CMP    MSGPKT+MS.RFC,BRFCNT ;IF THE RFC IS LOWER OR THE SAME AS BRFC THEN
2565 015616 101002              BHI    50235$
2566 015620 163712 002360      SUB     MSGPKT+MS.RFC,(R2) ;SUBTRACT RFC FROM EXPECTED BRFC.
2567
2568 015624              50235$:
2569 015624 010203              MOV     R2,R3      ;R3 = ADR OF 2ND WORD.
2570 015626 062703 000010      ADD     #10,R3
2571
2572 015632              50236$:
2573 015632 021227 001747      :WHILE (R2) GT #999. DO
2574 015636 003404              CMP     (R2),#999.
2575 015640 162712 001750      BLE     50237$
2576 015644 005213              SUB     #1000.,(R2) ;UPDATE BYTE COUNT
2577                                INC     (R3)      ;LET (R3) := (R3) + #1      ;2ND WORD.
2578 015646 000771              BR      50236$
2579 015650              50237$:
2580 015650 010302              MOV     R3,R2      ;LET R2 := R3 + #10      ;R2 = ADR OF 3RD WORD.
2581 015652 062702 000010      ADD     #10,R2
2582 015656              50240$:
2583 015656 021327 001747      :WHILE (R3) GT #999. DO
2584 015662 003404              CMP     (R3),#999.
2585 015664 162713 001750      BLE     50241$
2586 015670 005212              SUB     #1000.,(R3) ;UPDATE BYTE COUNT
2587                                INC     (R2)      ;LET (R2) := (R2) + #1      ;3RD WORD.
2588 015672 000771              BR      50240$
2589 015674              50241$:
2590 015674 010203              MOV     R2,R3      ;LET R3 := R2 + #10      ;R3 = ADR OF 4TH WORD.
2591 015676 062703 000010      ADD     #10,R3
2592 015702              50242$:
2593 015702 021227 001747      :WHILE (R2) GT #999. DO
2594 015706 003404              CMP     (R2),#999.
2595 015710 162712 001750      BLE     50243$
2596 015714 005213              SUB     #1000.,(R2) ;UPDATE BYTE COUNT
2597                                INC     (R3)      ;LET (R3) := (R3) + #1      ;4TH WORD.
2598 015716 000771              BR      50242$
2599 015720              50243$:
2600 015720 105737 003467      TSTB   RWERR      ;IF R/W ERROR, UPDATE ERROR COUNT.
2601 015724 001440              BEQ     50244$
2602 015726 010502              MOV     R5,R2      ;R2 = ADR OF COUNTER.
2603 015730 066402 016036      ADD     EINC(R4),R2
2604 015734 062702 002766      ADD     #WRREC,R2
2605 015740 105737 003470      TSTB   UNREC      ;IS THE ERROR UNRECOVERABLE?
2606 015744 001404              BEQ     50245$
2607 015746 062702 000010      ADD     #10,R2      ;YES, POINT TO NEXT COUNTER.
2608 015752 005212              INC     (R2)      ;UPDATE THE ERROR COUNTER

```

```

2609
2610 015754 000424
2611 015756
2612 015756 005212
2613 015760 105737 002213
2614 015764 001020
2615 015766 105737 003522
2616 015772 001015
2617 015774 105737 002207
2618 016000 001412
2619 016002
    016002 013746 003460
    016006 012746 005422
    016012 012746 000002
    016016 010600
    016020 104414
    016022 062706 000006

2620
2621 016026
2622
2623 016026
2624
2625
2626 016026
2627
2628 016026
2629
2630 016026
2631
2632 016026
2633 016026 000207
2634
2635
2636 016030 000000
2637 016032 000040
2638 016034 000100
2639
2640 016036 000000
2641 016040 000020
2642 016042 000040
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652 016044 105737 003516
2653 016050 001426
2654 016052 013737 003420 003424
2655 016060 012737 104401 003420
2656 016066 012737 000004 003426
2657 016074 004737 016130
2658 016100 013737 003420 003424
2659 016106 012737 104001 003420

;ELSE - IF ERROR IS RECOVERABLE:
50245$: BR 50246$
    INC (R2)
    TSTB IREC
    BNE 50247$
    TSTB DROPED
    BNE 50250$
    TSTB ERCVER
    BEQ 50250$
    PRINTB #NURTY1,RETRYC
;UPDATE THE ERROR COUNTER
;IF ERROR RECOVERY IS ENABLED:
;IF UNIT HAS NOT BEEN DROPPED:
;PRINT # OF RETRIES TO RECOVER
    MOV RETRYC,-(SP)
    MOV #NURTY1,-(SP)
    MOV #2,-(SP)
    MOV SP,R0
    TRAP C$PNTB
    ADD #6,SP
;PROVIDED PRINT HAS BEEN ENABLED

50250$:
50247$:
50246$:
50244$:
50234$:
50233$:
    RTS PC
;
;INDEXES TO BYTE COUNTERS.
BINC: 0
      40
      100
;INDEXES TO READ/WRITE ERROR COUNTERS.
EINC: 0
      20
      40
;WRITE.
;READ REV.
;READ FWD.
;WRITE.
;READ REV.
;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::TSTB VFYFLG
        BEQ 50251$
        MOV CMDWRD,PCMDWD
        MOV #RDR,CMDWRD
        MOV #4,CMDLG
        JSR PC,VFEXC
        MOV CMDWRD,PCMDWD
        MOV #RDF,CMDWRD
;IF DATA IS TO BE VERIFIED:
;SAVE THE PREVIOUS COMMAND WORD.
;COMMAND IS READ REV.
;SET UP CMD LOGGING INDEX.
;GO READ ALL THE RECORDS REV.
;SAVE THE PREVIOUS COMMAND WORD.
;COMMAND IS READ FWD.

```

```

2660 016114 012737 000006 003426      MOV    #6,CMDLG      ;SET UP CMD LOGGING INDEX.
2661 016122 004737 016130      JSR     PC,VFEXC      ;GO READ ALL RECORDS FWD.
2662
2663 016126      50251$:      RTS     PC      ;RETURN.
2664 016126 000207
2665
2666
2667
2668
2669      :      SUBROUTINE TO EXECUTE THE READ AND VERIFY, FORWARD OR REVERSE.
2670      :      INPUTS:
2671      :      OUTPUTS:
2672      :      REGISTERS:      R2
2673      :      CALLS:      CMDAC, FIRSTU, VFISU, NEXTU, CKHAE.
2674
2675 016130 013737 003420 002330 VFEXC:: MOV    CMDWRD,CMDPKT      ;COMMAND PACKET = READ REV OR FWD.
2676 016136 042737 004000 002330      BIC     #BRF.C,CMDPKT
2677 016144 105737 003520      TSTB    SWBFLG      ;IF BYTES ARE TO BE SWAPPED:
2678 016150 001403      BEQ     50252$
2679 016152 052737 010000 002330      BIS     #SWB.C,CMDPKT      ;SET SWAB BIT IN CMD PACKET.
2680
2681 016160      50252$:      MOV     CMDPKT,CMDSAV      ;SAVE COMMAND PACKET 1ST WORD.
2682 016160 013737 002330 003422      MOV     DATARD,CMDPKT+CP.ADL      ;SAVE BUFFER START ADDRESS.
2683 016166 013737 003410 002332      CLR     NCNT      ;CLEAR NUMBER OF OPERATIONS.
2684 016174 005037 003412
2685
2686 016200      50253$:      ;WHILE NCNT LT NCNT1 DO      ;WHILE THERE ARE RECORDS REMAINING:
2687 016200 023737 003412 003414      CMP     NCNT,NCNT1
2688 016206 002062      BGE     50254$
2689 016210 004737 011064      JSR     PC,CMDAC      ;STORE CMD ASCII IN ERROR MSG.
2690 016214 004737 017060      JSR     PC,FIRSTU      ;SET UP FOR FIRST UNIT.
2691 016220      50255$:      ;WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE DEVICES REMAINING:
2692 016220 026527 002604 177777      CMP     DEVTBL(R5),#END
2693 016226 001442      BEQ     50256$
2694 016230 032737 000400 003420      BIT     #MOD.CO,CMDWRD      ;IF CMD IS REVERSE THEN:
2695 016236 001421      BEQ     50257$
2696 016240 032765 000002 003502      BIT     #X0.BOT,EOTFLG(R5)      ;IF NOT AT BOT
2697 016246 001014      BNE     50260$
2698 016250 032765 000001 003502      BIT     #X0.EOT,EOTFLG(R5)      ;BUT IF AT EOT
2699 016256 001406      BEQ     50261$
2700 016260 105737 003524      TSTB    ALLEOT      ;AND ALL OTHERS AT EOT
2701 016264 001402      BEQ     50262$
2702 016266 004737 016356      JSR     PC,VFISU      ;THEN READ VERIFY
2703
2704 016272      50262$:      ;IF NOT ALL AT EOT, FREEZE UNIT(S)
2705
2706 016272 000402      BR     50263$      ;IF NOT AT BOT AND
2707 016274      50261$:      JSR     PC,VFISU      ;NOT AT EOT, READ VFY
2708 016274 004737 016356
2709
2710 016300      50263$:
2711
2712 016300      50260$:
2713 016300 000412      BR     50264$      ;ELSE IF CMD IS NOT REVERSE:
2714 016302      50257$:
2715 016302 032765 000001 003502      BIT     #X0.EOT,EOTFLG(R5)
2716 016310 001404      BEQ     50265$

```

```

2717 016312 032737 000001 003420      BIT      #CMD.CO,CMDWRD
2718 016320 001002                      BNE      50266$
2719 016322                      50265$:
2720                                ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
2721 016322 004737 016356      JSR      PC,VFISU      ;ISSUE CMD, CHECK STATUS AND DATA.
2722
2723 016326                      50266$:
2724
2725 016326                      50264$:
2726 016326 004737 017126      JSR      PC,NEXTU      ;GO FIND THE NEXT UNIT.
2727
2728 016332 000732                      BR      50255$
2729 016334                      50256$:
2730 016334 004737 017456      JSR      PC,CKHAE      ;CHECK FOR HALT AFTER EACH CMD.
2731 016340 005237 003412      INC      NCNT      ;UPDATE THE RECORD COUNT.
2732 016344 013737 003420 003424  MOV     CMDWRD,PCMDWD ;SAVE PREVIOUS COMMAND WORD.
2733
2734 016352 000712                      BR      50253$
2735 016354                      50254$:
2736 016354 000207      RTS      PC      ;RETURN.
2737
2738      :      SUBROUTINE TO ISSUE COMMAND, AWAIT INTERRUPT,
2739      :      CHECK STATUS, CHECK DATA.
2740      :      INPUTS:
2741      :      OUTPUTS:
2742      :      REGISTERS:      R2
2743      :      CALLS:      EXCUTE, GOWAIT, CKDATA.
2744
2745 016356 013702 003410  VFISU:: MOV     DATARD,R2      ;INIT READ BUFFER POINTER.
2746 016362 062702 000010      ADD     #8.,R2
2747 016366                      50267$:      ;WHILE R2 NE DATARD DO      ;UNTIL 8 BYTES HAVE BEEN SET,
2748 016366 020237 003410      CMP     R2,DATARD
2749 016372 001403      BEQ     50270$
2750 016374 012742 177777      MOV     #-1,-(R2)      ;INIT READ BUFFER.
2751
2752 016400 000772                      BR      50267$
2753 016402                      50270$:
2754 016402 004737 012054      JSR      PC,EXCUTE      ;GO EXECUTE THE COMMAND.
2755 016406 105737 003522      TSTB    DROPED      ;IF UNIT HAS NOT BEEN DROPPED THEN:
2756 016412 001002      BNE      50271$
2757 016414 004737 012364      JSR      PC,GOWAIT      ;GO WAIT FOR DONE BIT.
2758
2759 016420                      50271$:
2760 016420 105737 003522      TSTB    DROPED      ;IF UNIT HAS NOT BEEN DROPPED THEN:
2761 016424 001006      BNE      50272$
2762 016426 032765 000002 003502  BIT     #X0.BOT,EOTFLG(R5) ;WHEN NOT REVERSED INTO BOT, THEN
2763 016434 001002      BNE      50273$
2764 016436 004737 016444      JSR      PC,CKDATA      ;GO VERIFY DATA.
2765
2766 016442                      50273$:
2767
2768 016442                      50272$:
2769 016442 000207      RTS      PC
2770
2771
2772      :      SUBROUTINE TO COMPARE DATA BETWEEN READ AND WRITE BUFFERS
2773      :      AND PRINT ERROR MESSAGE ON MISCOMPARE.

```

```

2774      :      INPUTS:
2775      :      OUTPUTS:
2776      :      REGISTERS:      R2, R3, R4.
2777      :      CALLS:      GCMDB
2778
2779 016444 013703 003416      CKDATA: :MOV      BRFCNT,R3      ;COMPUTE REC LENGTH READ
2780 016450 163703 002360      SUB      MSGPKT+MS.RFC,R3
2781 016454 005703      TST      R3      ;WHEN NO DATA RECEIVED
2782 016456 001015      BNE      50274$
2783 016460      ERRHRD 17,WTVERM,DTAERM      ;PRINT ERROR AND EXIT
2784      016460 104456      TRAP      C$ERHRD
2785      016462 000021      .WORD    17
2786      016464 004430      .WORD    WTVERM
2787      016466 005752      .WORD    DTAERM
2788      016470      PRINTB #DTAER4      ;COMPARE ROUTINE
2789      016470 012746 005337      MOV      #DTAER4,-(SP)
2790      016474 012746 000001      MOV      #1,-(SP)
2791      016500 010600      MOV      SP,R0
2792      016502 104414      TRAP      C$PNTB
2793      016504 062706 000004      ADD      #4,SP
2794      016510 000560      BR      50275$
2795      016512      50274$: CMP      R3,BRFCNT      ;WHEN REC READ IS LONGER
2796      016512 020337 003416      BLOS    50276$
2797      016516 101417      ERRHRD 17,WTVERM,DTAERM      ;THAN EXPECTED, PRINT
2798      016520      TRAP      C$ERHRD
2799      016520 104456      .WORD    17
2800      016522 000021      .WORD    WTVERM
2801      016524 004430      .WORD    DTAERM
2802      016526 005752      PRINTB #DTAER5,CMDPKT+CP.CNT      ;AN ERROR MESSAGE
2803      016530      MOV      CMDPKT+CP.CNT,-(SP)
2804      016530 013746 002336      MOV      #DTAER5,-(SP)
2805      016534 012746 005360      MOV      #2,-(SP)
2806      016540 012746 000002      MOV      SP,R0
2807      016544 010600      TRAP      C$PNTB
2808      016546 104414      ADD      #6,SP
2809      016550 062706 000006      ;AND EXIT ROUTINE
2810      016554 000536      BR      50277$
2811      016556      50276$: MOV      R3,CKDCNT      ;SAVE VERIFICATION LENGTH - 1.
2812      016556 010337 017054      DEC      CKDCNT
2813      016556 005337 017054      CLR      CKDFF
2814      016556 005037 017056      CLR      R2
2815      016556 005002      MOV      DATAWT,R3
2816      016556 013703 003406      MOV      DATARD,R4
2817      016556 013704 003410      TSTB    T1SWB
2818      016556 105737 003523      BEQ     50300$
2819      016556 001401      SWAB     (R3)
2820      016556 000313      ;SWAP FIRST WORD OF WRT BFR
2821      016614      50300$: ;WHICH CONTAINS THE RECORD COUNT
2822      016614      50301$: ;REPEAT
2823      016614 020237 017054      ;REPEAT UNTIL ALL DATA IS COMPARED:
2824      016620 001011      CMP      R2,CKDCNT
2825      016622 105737 003520      BNE      50302$
2826      016626 001406      TSTB    SWBFLG
2827      016630 032737 000001 017054      BEQ     50303$
2828      016630      BIT      #BIT00,CKDCNT
2829      016630      ;IF THIS IS THE LAST BYTE THEN:
2830      016630      ;IF BYTE SWAPPING IS ENABLED THEN:
2831      016630      ;IF RECORD LENGTH IS ODD THEN:

```

```
2812 016636 001002      BNE      50304$
2813 016640 105723      TSTB     (R3)+
2814 016642 105724      TSTB     (R4)+      ;LAST BYTE WILL BE IN
2815                                ;THE UPPER BYTE.
2816 016644      50304$:
2817
2818 016644      50303$:
2819
2820 016644      50302$:
2821 016644 121314      CMPB     (R3),(R4)      ;ARE THEY EQUAL.
2822 016646 001452      BEQ      3$          ;BR IF SO.
2823 016650 005737 017056 TST      CKDFF      ;1 ST TIME THRU?
2824 016654 001010      BNE      2$          ;BR IF NOT.
2825 016656 005265 003346 INC      VFYCNTR(R5)      ;INC THE VERIFY ERROR COUNTER.
2826 016662 005265 003356 INC      HRDCNTR(R5)      ;INC THE HARD ERROR COUNT.
2827 016666      ERRHRD 17,WTVERM,DTAERM      ;REPORT WRITE/VERIFY ERROR.
2828 016676 005237 017056      2$:      INC      CKDFF;LET CKDFF := CKDFF + #1 ;INCREMENT # OF BYTES IN ERROR.
2829 016702 111437 003436      MOV     (R4),TIME1      ;SAVE WAS DATA FOR TYPOT.
2830 016706 042737 177400 003436      BIC     #177400,TIME1      ;CLEAR GARBAGE.
2831 016714 111337 003440      MOV     (R3),TIME2      ;SAVE SHOULD BE DATA FOR TYPOT.
2832 016720 042737 177400 003440      BIC     #177400,TIME2      ;CLEAR GARBAGE.
2833 016726 023727 017056 000013      CMP     CKDFF,#11.      ;IF ERROR BYTE COUNT IS LESS THAN 11:
2834 016734 002017      BGE      50305$
2835 016736      PRINTX  #DTAER2,R2,<B,TIME1>,<B,TIME2>;PRINT ACTUAL & EXPECTED DATA
2836 016736 005046      CLR      -(SP)
2837 016740 153716 003440      BISB     TIME2,(SP)
2838 016744 005046      CLR      -(SP)
2839 016746 153716 003436      BISB     TIME1,(SP)
2840 016752 010246      MOV     R2,-(SP)
2841 016754 012746 005226      MOV     #DTAER2,-(SP)
2842 016760 012746 000004      MOV     #4,-(SP)
2843 016764 010600      MOV     SP,R0
2844 016766 104415      TRAP     C$PNTX
2845 016770 062706 000012      ADD     #12,SP
2846 016774      50305$:
2847 016776 105723      3$:      TSTB     (R3)+      ;UPDATE WRITE BUFFER ADDRESS.
2848 016776 105724      TSTB     (R4)+      ;UPDATE READ BUFFER ADDRESS.
2849 017000 105722      TSTB     (R2)+      ;UPDATE BYTE COUNTER.
2850 017002 020237 017054      CMP      R2,CKDCNT      ;END OF DATA COMPARE REPEAT LOOP.
2851 017006 003702      BLE      50301$
2852 017010 005237 017054      INC      CKDCNT
2853 017014 005737 017056      TST      CKDFF
2854 017020 001414      BEQ      50306$
2855 017022      PRINTB  #DTAER3,CKDFF,CKDCNT      ;PRINT # OF BYTES IN ERROR.
2856 017022 013746 017054      MOV     CKDCNT,-(SP)
2857 017026 013746 017056      MOV     CKDFF,-(SP)
2858 017032 012746 005275      MOV     #DTAER3,-(SP)
2859 017036 012746 000003      MOV     #3,-(SP)
2860 017042 010600      MOV     SP,R0
2861 017044 104414      TRAP     C$PNTB
2862 017046 062706 000010      ADD     #10,SP
```

```

2848 017052          50306$:
2849
2850 017052          50277$:
2851
2852 017052          50275$:
2853 017052 000207      RTS      PC              ;OTHERWISE, RETURN.
2854
2855 017054 000000      CKDCNT: .WORD 0          ;# OF BYTES TO BE VERIFIED -1.
2856 017056 000000      CKDFF:  .WORD 0          ;# OF BYTES IN ERROR COUNTER.
2857
2858                :      SUBROUTINE TO FIND THE FIRST DEVICE IN THE TEST SEQUENCE.
2859                :      INPUTS:
2860                :      OUTPUTS:
2861                :      REGISTERS:
2862                :      CALLS:
2863
2864 017060 105037 003522 FIRSTU:: CLRB DROPED          ;CLR UNIT DROPPED FLAG
2865 017064 005005          CLR      R5              ;CLR DEVICE POINTER.
2866 017066 026527 002604 177774 50307$: CMP      DEVTBL(R5),#NINUSE      ;WHILE DEVICES ARE NOT IN USE:
2867 017074 001003          BNE      50310$
2868 017076 062705 000002          ADD      #2,R5          ;LET R5 := R5 + #2          ;POINT TO NEXT DEVICE.
2869 017102 000771          BR       50307$
2870 017104          50310$:
2871 017104 026527 002604 177777      CMP      DEVTBL(R5),#END          ;IF ALL UNITS HAVE BEEN DROPPED THEN:
2872 017112 001001          BNE      50311$
2873 017114          DOCLN              ;DO CLEAN CODE AND TERMINATE PASS.
2874 017114 104444          TRAP      C$DCLN
2875 017116
2876 017116 016537 002604 002074 50311$: MOV      DEVTBL(R5),L$LUN      ;SET UNIT # IN 'HEADER' FOR ERROR REPORT
2877 017124 000207          RTS      PC              ;RETURN WITH 1ST DEVICE IN R5.
2878
2879
2880                :      SUBROUTINE TO FIND THE NEXT UNIT IN THE TEST CYCLE.
2881                :      INPUTS:
2882                :      OL "US:
2883                :      REGISTERS:
2884                :      CALLS:
2885
2886 017126 105037 003522 NEXTU:: CLRB DROPED          ;CLR UNIT DROPPED FLAG
2887          :REPEAT          ;REPEAT UNTIL THE NEXT DEVICE IS FOUND.
2888 017132          50312$:
2889 017132 062705 000002          ADD      #2,R5          ;UPDATE DEVICE TABLE POINTER.
2890 017136 026527 002604 177774      CMP      DEVTBL(R5),#NINUSE      ;UNTIL DEVTBL(R5) NE #NINUSE
2891 017144 001772          BEQ      50312$
2892 017146 016537 002604 002074      MOV      DEVTBL(R5),L$LUN      ;SET UNIT # IN 'HEADER' FOR ERROR REPORT
2893 017154 000207          RTS      PC              ;RETURN.
2894
2895
2896                :      SUBROUTINE TO DROP A DEVICE FROM THE TEST SEQUENCE.
2897                :      INPUTS:
2898                :      OUTPUTS:
2899                :      REGISTERS:
2900                :      CALLS:          MOVMSG, PRXST, LOG
2901
2902 017156 005265 003366 DROPU:: INC      FTLCNT(R5)      ;INCREMENT THE FATAL ERROR COUNT.
2903 017162 013704 002370          MOV      MSGPKT+MS.XS3,R4      ;GET UDIAG ERROR CODE FROM XSTAT3.

```



```
2904 017166 042704 000377      BIC      #377,R4
2905 017172 016503 002544      MOV      MSGPKA(R5),R3 ;ADR OF THIS UNIT'S MSG PACKET.
2906 017176 005002              CLR      R2 ;LET R2 := #0 ;CLR COUNTER.
2907 017200              50313$: ;WHILE R2 NE #MSGCNT DO ;WHILE THERE ARE MORE LOCATIONS:
2908 017200 020227 000020      CMP      R2,#MSGCNT
2909 017204 001405              BEQ      50314$
2910 017206 012723 177777      MOV      #-1,(R3)+ ;INIT THE MSG PACKET WITH ALL 1'S
2911 017212 062702 000002      ADD      #2,R2 ;LET R2 := R2 + #2 ;UPDATE COUNTER.
2912
2913 017216 000770              BR       50313$
2914 017220
2915 017220 012775 002340 002514 50314$: MOV      #GSCPK,@TSDB(R5) ;INITIATE A GET STATUS COMMAND.
2916 017226 004737 012700      JSR      PC,WSSR ;WAIT A WHILE FOR SSR=1
2917 017232 004737 012734      JSR      PC,MOVMSG ;MOVE MSG PACKET TO COMMON AREA.
2918 017236 020427 157400      CMP      R4,#X3.RNY ;IF WE HAVE A CAPSTAN RUNAWAY THEN:
2919 017242 001005              BNE      50315$
2920 017244              ERRDF 16,RNYM,STAERM ;REPORT CAPSTAN RUNAWAY WITH TACH CNT.
2921              TRAP C$ERDF
2922              .WORD 16
2923              .WORD RNYM
2924              .WORD STAERM
2925
2926 017262              ;ELSE-IF NOT A RUNAWAY:
2927 017262 105737 003465      BR       50316$
2928 017266 001404
2929 017270 105237 005522      50315$: JSR      PC,PRXST ;PRINT EXTENDED STATUS REGISTERS.
2930 017274 004737 015544
2931
2932 017300              50316$: TSTB     RECLOG ;IF THE RECORD HAS BEEN LOGGED THEN:
2933 017300 104424              BEQ      50317$
2934 017302 005765 003326      INCB     DROPE ;SET UNIT DROPPED FLAG.
2935 017306 001402              JSR      PC,LOG ;LOG DATA BYTES + RD/WR ERRORS.
2936 017310 005365 003326
2937
2938 017314              50317$: DORPT ;PRINT PERFORMANCE REPORT
2939 017314 013737 003534 017366 50320$: TRAP C$DRPT
2940 017322 013700 003534      DROPUA: TST     PASCNT(R5) ;IF PASCNT(R5) NE #0 THEN
2941 017326 104451              BEQ      50320$
2942              DEC     PASCNT(R5) ;LET PASCNT(R5) := PASCNT(R5) - #1
2943
2944 017330 026527 002604 177774      MOV      TSNP,DROPN ;SAVE # OF UNIT TO BE DROPPED.
2945 017336 001410              MOV      TSNP,R0 ;R0=LOGICAL DEVICE NUMBER
2946 017340 105737 002213              DODU     R0 ;DROP THE UNIT
2947 017344 001005              TRAP C$DODU
2948 017346 000240              ;EXEC BGNDU-ENDDU CODE IF IDU = 0
2949 017350 000240
2950 017352 000240
2951 017354 105237 003526      CMP      DEVTBL(R5),#NINUSE ;IF UNIT NOT DROPPED
2952              BEQ      50321$
2953 017360              TSTB     IREC ;IF RECOVERY IS ENABLED THEN:
2954              BNE      50322$
2955              NOP
2956              NOP
2957              INCB     STAFLG ;SET START FLAG TO ENABLE REWIND,
2958              50322$:
2959
```

```

2955 017360      50321$:
2956 017360      DRORTN: INCB      DROPE
2957 017364      105237 003522      RTS      PC      ;SET UNIT DROPPED FLAG.
2958              000207              ;RETURN.
2959 017366      000000      DROPN: .WORD 0      ;# OF UNIT TO BE DROPPED
2960
2961      :      SUBROUTINE TO PRINT EXTENDED STATUS REGISTERS.
2962      :      INPUTS:
2963      :      OUTPUTS:
2964      :      REGISTERS:
2965      :      CALLS:
2966
2967 017370      PRXST:: PRINTX #GETSTM
      017370      012746 005507
      017374      012746 000001
      017400      010600
      017402      104415
      017404      062706 000004
2968 017410      PRINTX #STAERS,MSGPKT+MS.XS0,MSGPKT+MS.XS1,MSGPKT+MS.XS2,MSGPKT+MS.XS3,MSGPKT+MS.XS
      017410      013746 002372
      017414      013746 002370
      017420      013746 002366
      017424      013746 002364
      017430      013746 002362
      017434      012746 006753
      017440      012746 000006
      017444      010600
      017446      104415
      017450      062706 000016
      017454      000207
2969              RTS PC
2970
2971      :      SUBROUTINE TO HALT AFTER EACH COMMAND.
2972      :      INPUTS:
2973      :      OUTPUTS:
2974      :      REGISTERS:      R3, R4
2975      :      CALLS:
2976
2977 017456      105737 002206      CKHAE:: TSTB      HAE;IFB HAE NE #0 THEN      ;IF HALT FLAG IS SET:
2978 017462      001430      BEQ      50323$
2979 017464      105737 003531      TSTB      MISCFG      ;IFB MISCFG EQ #0 THEN      :
2980 017470      001023      BNE      50324$
2981 017472      104450      MANUAL      ;IS MANUAL INTERVENTION ALLOWED?
2982 017474      103023      BNCOMPLETE CKHRTN      ;BR IF NOT.
2983 017476      013704 003420      MOV      CMDWRD,R4      ;LET R4 := CMDWRD
2984 017502      004737 011136      JSR      PC,GCMDA      ;FETCH ADR OF CMD ASCII.
2985 017506      112337 004306      MOV      (R3)+,HALTM      ;MOVE CMD ASCII
2986 017512      112337 004307      MOV      (R3)+,HALTM+1      ;LET HALTM+1 := (R3)+
2987 017516      111337 004310      MOV      (R3),HALTM+2      ;INTO MESSAGE.
2988 017522      104443      GMANIL HALTM,TIME1,1,YES      ;HALT - WAIT FOR AN OEPRTOR INPUT.
      017522      000404      TRAP      C$GMAN
      017526      003436      BR      10000$
      017530      000130      .WORD      TIME1
      017532      004306      .WORD      T$CODE
      017534      000001      .WORD      HALTM
      .WORD      1

```

```

2989 017536 10000$:
2990 017536 10000$:
2991 017536 000402 BR 50325$
2992 017540 50324$: CLRB MISCFG ;LET MISCFG :B= #0 ;
2993 017540 105037 003531
2994
2995 017544 50325$:
2996
2997 017544 50323$:
2998 017544 000207 CKHRTN: RTS PC ;RETURN
2999 .EVEN
3000
3001 017546 ENDMGD
3002
3003
3004 .TITLE MISCELLANEOUS SECTIONS
3005 .SBTTL REPORT CODING SECTION
3006
3007
3008 ;++
3009 ; THE REPORT CODING SECTION CONTAINS THE
3010 ; 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
3011 ;--
3012
3013 017546 BGNRPT
3014 017546 LSRPT::
3015 017552 010537 003452 MOV R5,R5SAVE ;SAVE CURRENT DEVICE POINTER.
3016 017556 004737 017060 JSR PC,FIRSTU ;FIND THE FIRST UNIT.
3017 017556 026527 002604 177777 50326$: ;WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
3018 017564 001562 CMP DEVTBL(R5),#END
3019 017566 BEQ 50327$
3020 017566 PRINTS #RPT1A,DEVTBL(R5),PASCNT(R5),RECCNT(R5)
3021 017566 016546 003376 MOV RECCNT(R5),-(SP)
3022 017572 016546 003326 MOV PASCNT(R5),-(SP)
3023 017576 016546 002604 MOV DEVTBL(R5),-(SP)
3024 017602 012746 020410 MOV #RPT1A, -(SP)
3025 017606 012746 000004 MOV #4, -(SP)
3026 017612 010600 MOV SP,R0
3027 017614 104416 TRAP C$PNTS
3028 017616 062706 000012 ADD #12, SP
3029 017622 PRINTS #RPT1B,WRBC+30(R5),WRBC+20(R5),WRBC+10(R5),WRBC(R5)
3030 017622 016546 002626 MOV WRBC(R5),-(SP)
3031 017626 016546 002636 MOV WRBC+10(R5),-(SP)
3032 017632 016546 002646 MOV WRBC+20(R5),-(SP)
3033 017636 016546 002656 MOV WRBC+30(R5),-(SP)
3034 017642 012746 020465 MOV #RPT1B, -(SP)
3035 017646 012746 000005 MOV #5, -(SP)
3036 017652 010600 MOV SP,R0
3037 017654 104416 TRAP C$PNTS
3038 017656 062706 000014 ADD #14, SP
3039 017662 PRINTS #RPT1C,RRBC+30(R5),RRBC+20(R5),RRBC+10(R5),RRBC(R5)
3040 017662 016546 002666 MOV RRBC(R5),-(SP)
3041 017666 016546 002676 MOV RRBC+10(R5),-(SP)
3042 017672 016546 002706 MOV RRBC+20(R5),-(SP)
3043 017676 016546 002716 MOV RRBC+30(R5),-(SP)
3044 017702 012746 020536 MOV #RPT1C, -(SP)

```

MISCELLANEOUS SECTIONS
REPORT CODING SECTION

MACRO M1113 25-MAY-82 09:51 PAGE 20-57

M 8

SEQ 0103

	017706	012746	000005				MOV	#5,-(SP)
	017712	010600					MOV	SP,R0
	017714	104416					TRAP	C\$PNTS
	017716	062706	000014				ADD	#14,SP
3022	017722			PRINTS	#RPT1D,RFBC+30(R5),RFBC+20(R5),RFBC+10(R5),RFBC(R5)			
	017722	016546	002726				MOV	RFBC(R5),-(SP)
	017726	016546	002736				MOV	RFBC+10(R5),-(SP)
	017732	016546	002746				MOV	RFBC+20(R5),-(SP)
	017736	016546	002756				MOV	RFBC+30(R5),-(SP)
	017742	012746	020607				MOV	#RPT1D,-(SP)
	017746	012746	000005				MOV	#5,-(SP)
	017752	010600					MOV	SP,R0
	017754	104416					TRAP	C\$PNTS
	017756	062706	000014				ADD	#14,SP
3023	017762			PRINTS	#RPT1F,WRREC(R5),RRREC(R5),RFREC(R5)			
	017762	016546	003026				MOV	RFREC(R5),-(SP)
	017766	016546	003006				MOV	RRREC(R5),-(SP)
	017772	016546	002766				MOV	WRREC(R5),-(SP)
	017776	012746	020713				MOV	#RPT1F,-(SP)
	020002	012746	000004				MOV	#4,-(SP)
	020006	010600					MOV	SP,R0
	020010	104416					TRAP	C\$PNTS
	020012	062706	000012				ADD	#12,SP
3024	020016			PRINTS	#RPT1G,WRUNR(R5),RRUNR(R5),RFUNR(R5)			
	020016	016546	003036				MOV	RFUNR(R5),-(SP)
	020022	016546	003016				MOV	RRUNR(R5),-(SP)
	020026	016546	002776				MOV	WRUNR(R5),-(SP)
	020032	012746	020764				MOV	#RPT1G,-(SP)
	020036	012746	000004				MOV	#4,-(SP)
	020042	010600					MOV	SP,R0
	020044	104416					TRAP	C\$PNTS
	020046	062706	000012				ADD	#12,SP
3025	020052	105737	002210	TSTB	BADTSW ;IFB BADTSW NE #0 THEN			
3026	020056	001402		BEQ	50330\$			
3027	020060	004737	020142	JSR	PC,BTRPT ;GO PRINT BAD TAPE SPOTS WHEN ENABLED			
3028								
3029	020064			50330\$:				
3030	020064			PRINTS	#RPT1I,SCCNT(R5),HRDCNT(R5),FTLCNT(R5),VFYCNT(R5)			
	020064	016546	003346				MOV	VFYCNT(R5),-(SP)
	020070	016546	003366				MOV	FTLCNT(R5),-(SP)
	020074	016546	003356				MOV	HRDCNT(R5),-(SP)
	020100	016546	003336				MOV	SCCNT(R5),-(SP)
	020104	012746	021161				MOV	#RPT1I,-(SP)
	020110	012746	000005				MOV	#5,-(SP)
	020114	010600					MOV	SP,R0
	020116	104416					TRAP	C\$PNTS
	020120	062706	000014				ADD	#14,SP
3031	020124	004737	017126	JSR	PC,NEXTU ;FIND THE NEXT UNIT.			
3032								
3033	020130	000612		BR	50326\$			
3034	020132			50327\$:				
3035	020132	013705	003452	MOV	R\$SAVE,R\$;RESTORE CURRENT DEVICE POINTER.			
3036	020136			EXIT	RPT			
	020136	000167						
	020140	001130					.WORD	J\$JMP
3037							.WORD	L10010-2-
3038				:	SUBR TO PRINT BAD TAPES SPOTS DURING THE REPORT PRINTS			

```
3039      :      WRITE RETRIES: CUMULATIVE COUNT
3040      :      BAD TAPE SPOTS: COUNT PER TAPE PASS ONLY, NOT CUMULATIVE.
3041      :      COUNT OF RECOVERABLE WRITE ERRORS EXCLUDES BAD TAPE SPOTS.
3042
3043      020142      BTRPT:: PRINTS  #RPT1E,WRTYCT(R5)      ;PRINT GLOBAL WRITE RETRY COUNT
3044      020142      016546      003316      MOV      WRTYCT(R5),-(SP)
3045      020146      012746      021035      MOV      #RPT1E, -(SP)
3046      020152      012746      000002      MOV      #2, -(SP)
3047      020156      010600      MOV      SP,R0
3048      020160      104416      TRAP     C$PNTS
3049      020162      062706      000006      ADD      #6,SP
3050      020166      016537      002616      003512      MOV      BTADDR(R5),BTPT      ;BTPT IS BOTH THE BAD TAPE SPOT COUNTER
3051      020174      017703      163312      MOV      @BTPT,R3      ;AND THE LOGGING INDEX
3052      020200      006203      ASR      R3
3053      020202      PRINTS  #RPT1J,R3      ;PRINT # OF BAD TAPE SPOTS
3054      020202      010346      MOV      R3, -(SP)
3055      020204      012746      021065      MOV      #RPT1J, -(SP)
3056      020210      012746      000002      MOV      #2, -(SP)
3057      020214      010600      MOV      SP,R0
3058      020216      104416      TRAP     C$PNTS
3059      020220      062706      000006      ADD      #6,SP
3060      020224      005703      TST      R3      ;PRINT RECORD # IF BAD SPC S DETECTED
3061      020226      001457      BEQ      50331$
3062      020230      020327      000024      CMP      R3,#20.      ;IF R3 HI #20. THEN
3063      020234      101402      BLOS     50332$
3064      020236      012703      000024      MOV      #20.,R3      ;20 BAD SPOTS IS THE LIMIT
3065
3066      50332$:
3067      020242      PRINTS  #CRLFSP
3068      020242      012746      005744      MOV      #CRLFSP, -(SP)
3069      020246      012746      000001      MOV      #1, -(SP)
3070      020252      010600      MOV      SP,R0
3071      020254      104416      TRAP     C$PNTS
3072      020256      062706      000004      ADD      #4,SP
3073      020262      013704      003512      MOV      BTPT,R4      ;LET R4 := BTPT + #2 ;FETCH A BAD SPOT ID
3074      020266      062704      000002      ADD      #2,R4
3075      020272      005002      CLR      R2      ;R2 = PRINT COUNT PER LINE: 10 MAX
3076      020274      50333$:      ;REPEAT
3077      020274      PRINTS  #RPT1K,(R4)      ;PRINT A BAD SPOT ID
3078      020274      011446      MOV      (R4), -(SP)
3079      020276      012746      021152      MOV      #RPT1K, -(SP)
3080      020302      012746      000002      MOV      #2, -(SP)
3081      020306      010600      MOV      SP,R0
3082      020310      104416      TRAP     C$PNTS
3083      020312      062706      000006      ADD      #6,SP
3084      020316      005202      INC      R2      ;LET R2 := R2 + #1 ;COUNT PRINTS
3085      020320      062704      000002      ADD      #2,R4      ;LET R4 := R4 + #2 ;NEXT
3086      020324      020227      000012      CMP      R2,#10.      ;IF R2 EQ #10. THEN
3087      020330      001014      BNE      50334$
3088      020332      PRINTS  #CRLFSP      ;GO TO NEXT PRINT LINE PAST 10 PRINTS
3089      020332      012746      005744      MOV      #CRLFSP, -(SP)
3090      020336      012746      000001      MOV      #1, -(SP)
3091      020342      010600      MOV      SP,R0
3092      020344      104416      TRAP     C$PNTS
3093      020346      062706      000004      ADD      #4,SP
3094      020352      162703      000012      SUB      #10.,R3      ;LET R3 := R3 - #10. ;ADJUST BAD SPOT COUNT
3095      020356      162702      000012      SUB      #10.,R2      ;LET R2 := R2 - #10. ;ADJUST PRINT COUNT
```

```
3068
3069 020362
3070 020362 020203
3071 020364 001343
3072
3073 020366
3074 020366
      020366 012746 005741
      020372 012746 000001
      020376 010600
      020400 104416
      020402 062706 000004
3075 020406 000207
3076
3077
3078 020410 045 116 045 RPT1A: .NLIST BEX
3079 020465 045 101 102 RPT1B: .ASCIIZ /%N%N%AUNIT %D1%S3%APASS:%D5%S3%ARECORD:%D5%N/
3080 020536 045 101 102 RPT1C: .ASCIIZ /%ABYTES WRITTEN %D3%A,%Z3%A,%Z3%A,%Z3%N/
3081 020607 045 101 102 RPT1D: .ASCIIZ /%ABYTES READ REV %D3%A,%Z3%A,%Z3%A,%Z3%N/
3082 020657 045 123 062 RPT1E: .ASCIIZ /%ABYTES READ FWD %D3%A,%Z3%A,%Z3%A,%Z3%N/
3083 020713 045 101 122 RPT1F: .ASCIIZ /%S23%AWRT%S4%ARDR%S4%ARDF%N/
3084 020764 045 101 125 RPT1G: .ASCIIZ /%ARECOVERABLE ERRORS %D5%S2%D5%S2%D5%N/
3085 021035 045 101 127 RPT1H: .ASCIIZ /%AUNRECOVERABLE ERRORS %D5%S2%D5%S2%D5%N/
3086 021065 045 116 045 RPT1I: .ASCIIZ /%AWRITE RETRIES%S8%D5%N/
3087 021152 045 104 065 RPT1J: .ASCIIZ /%N%D2% BAD SPOTS THIS TAPE PASS PRECEDING RECORD #:/
3088 021161 045 101 123 RPT1K: .ASCIIZ /%D5%S1/
3089 021235 045 123 063 RPT1L: .ASCIIZ /%ASPEC COND%S3%AHARD%S3%AFATAL%S3%ACOMPARE%N'
3090
3091
3092
3093 021272
      021272
      021272 104425
3094
3095
3096
3097
3098
3099
3100
3101
3102 021274
      021274
3103
3104 021274 000000
3105 021276 177777
3106 021300 177777
3107 021302
3108
3109
3110
3111
3112
3113
3114
3115
3116 021302
```

50334\$: ;
CMP R2,R3 ;UNTIL R2 EQ R3 ;LIMIT: # OF BAD SPOTS
BNE 50333\$;
50331\$: ;
PRINTS #CRLF ;
RTS PC

MOV #CRLF,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTS
ADD #4,SP

ENDRPT

L10010: TRAP C\$RPT

.SBTTL LOAD DEVICE PROTECTION TABLE

;++
;TABLE FOR SUPERVISOR TO IDENTIFY THE P-TBL FOR THE LOAD DEV
;THE SUPERVISOR USES THE TBL TO WARN THE OPERATOR WHEN HE TRIES TO TEST THE LOAD DEV
;--

BGNPROT

L\$PROT::

.WORD 0 ;P-TBL OFFSET OF TSDB
.WORD -1 ;P-TBL OFFSET OF MASS BUS UNIT #: -1 = NOT A MASS BUS DE
.WORD -1 ;P-TBL OFFSET OF DRIVE #: -1 = NONE, THREE DRIVES PER CONTRO
ENDPROT

.SBTTL INITIALIZE SECTION

;++
; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
; AT THE BEGINNING OF EACH PASS.
;--

BGNINIT

MISCELLANEOUS SECTIONS
INITIALIZE SECTION

MACRO M1113 25-MAY-82 09:51 PAGE 20-60

C 9

SEQ 0106

```

021302          LS$INIT::
3117
3118 021302 032727 000003 002330 INIT10: BIT    #BIT0!BIT1,#CMDPKT ;IF CMD PACKET IS NOT ON MODULO 4 BOUNDARY:
3119 021310 001421          BEQ      50335$
3120 021312          ERRSF 1,CMDPKM          ;PRINT ERROR MSG,
          TRAP      C$ERSF
          .WORD     1
          .WORD     CMDPKM
          .WORD     0
3121 021322          DELAY 200.          ;GO TO SUPERVISOR, WAIT 2 SECONDS.
          MOV      #200.,(PC)+
          .WORD     0
          MOV      L$DLY,(PC)+
          .WORD     0
          DEC      -6(PC)
          BNE      -4
          DEC      -22(PC)
          BNE      -20
          BR      INIT10
3122 021352 000753
3123
3124 021354          50335$:
3125
3126 021354 105737 002204          TSTB    CLRFLG          ;IF CLR COUNTERS FLAG SET:
3127 021360 001413          BEQ      50336$
          CLRB    CLRFLG          ;INIT CLR FLAG.
3128 021362 105037 002204          CLR      R2          ;LET R2 := #0
3129 021366 005002          50337$: ;WHILE R2 NE #CNTLEN DO
          CMP      R2,#CNTLEN
          BEQ      50340$
          CLR      WRBC(R2)
          ADD      #2,R2          ;CLR ALL STATISTICAL COUNTERS.
          ;LET R2 := R2 + #2
3130 021370
3131 021370 020227 000550          BR      50337$
3132 021374 001405
3133 021376 005062 002626
3134 021402 062702 000002
3135
3136 021406 000770          50340$:
3137 021410          50336$:
3138
3139 021410
3140
3141 021410 105737 002205          TSTB    RRANV          ;IF RESET RANDOM VARIABLE FLAG IS SET THEN:
3142 021414 001406          BEQ      50341$
3143 021416 012737 153624 003432          MOV      #RANBC,RANB          ;RESET RANDOM BASE #.
3144
3145 021424 012737 032561 003434          MOV      #RANSC,RANS          ;RESET RANDOM SAVE LOCATION.
3146
3147 021432          50341$:
3148 021432          READEF #EF.START          ;READ START COMMAND EVENT FLAG.
          MOV      #EF.START,RO
          TRAP      C$REFG
          BNCOMPLETE          INIT15          ;BRANCH IF NOT STARTING.
          BCC      INIT15
          INCB     STAFLG          ;SET START COMMAND FLAG.
3150 021442 105237 003526          MOV      #6,R5          ;LET R5 := #6
3151 021446 012705 000006          50342$: ;REPEAT
          MOV      #NINUSE,DEVTBL(R5)          ;INITIATE UNIT NUMBER TABLE
          SUB      #2,R5          ;BY STORING NOT IN USE IN EACH LOCATION.
          TST      R5          ;LET R5 := R5 - #2
          BNE      50342$          ;UNTIL R5 EQ #0
          CMP      #1,L$UNIT          ;ONLY ONE UNIT ALLOWED
3152 021452
3153 021452 012765 177774 002604
3154 021460 162705 000002
3155 021464 005705
3156 021466 001371
3157 021470 022737 000001 002012

```

MISCELLANEOUS SECTIONS
INITIALIZE SECTION

MACRO M1113 25-MAY-82 09:51 PAGE 20-61

D 9

SEQ 0107

```

3158 021476 001425      BEQ      5034$      ;OK
3159 021500      PRINTF   #AUDRUN      ;TELL THE MAN
      021500 012746 005146      MOV      #AUDRUN,-(SP)
      021504 012746 000001      MOV      #1,-(SP)
      021510 010600      MOV      SP,R0
      021512 104417      TRAP     C$PNTF
      021514 062706 000004      ADD      #4,SP
3160 021520      DELAY    25          ;WAIT
      021520 012727 000025      MOV      #25,(PC)+
      021524 000000      .WORD    0
      021526 013727 002116      MOV      L$DLY,(PC)+
      021532 000000      .WORD    0
      021534 005367 177772      DEC      -6(PC)
      021540 001375      BNE      -4
      021542 005367 177756      DEC      -22(PC)
      021546 001367      BNE      -20
3161 021550      DOCLN          ;ABORT
      021550 104444      TRAP     C$DCLN
3162 021552 013705 002012      5034$: MOV      L$UNIT,R5      ;LET R5 := L$UNIT SHIFT 1
3163 021556 006305      ASL      R5
3164 021560      50343$: ;REPEAT      ;STORE ALL UNIT
3165 021560 162705 000002      SUB      #2,R5      ;LET R5 := R5 - #2
3166 021564 010565 002604      MOV      R5,DEVTBL(R5) ;LET DEVTBL(R5) := R5 SHIFT -1
3167 021570 006265 002604      ASR      DEVTBL(R5)
3168 021574 005705      TST      R5      ;UNTIL R5 EQ #0
3169 021576 001370      BNE      50343$
3170
3171 021600      INIT15: READEF #EF.PWR      ;HAS THERE BE A POWER FAILURE?
      021600 012700 000034      MOV      #EF.PWR,R0
      021604 104447      TRAP     C$REFG
3172 021606      BNCOMPLETE INIT16      ;BRANCH IF NOT.
      021606 103004      BCC      INIT16
3173 021610 105237 003526      INCB     STAFLE
3174 021614 105237 003527      INCB     PWRFLG      ;IF SO - SET THE START FLAG.
3175      ;IF SO - SET THE POWER FAIL FLAG.
3176 021620      INIT16: RFLAGS OPFLAG      ;READ AND STORE FLAGS SET BY OPERATOR
      021620 104421      TRAP     C$RFLA
      021622 010037 003536      MOV      R0,OPFLAG
3177 021626 005003      CLR      R3      ;LET R3 := #0
3178 021630 105737 003527      TSTB    PWRFLG      ;CLEAR EVENT FLAG
3179 021634 001020      BNE      50344$      ;IF POWER FAIL HAS NOT OCCURRED THEN:
3180 021636      RFADEF #EF.NEW      ;UPDATE PASS COUNT WHEN
      021636 012700 000035      MOV      #EF.NEW,R0
      021642 104447      TRAP     C$REFG
3181 021644 103014      BCC      50345$      ;SUPERVISOR IS IN NEW PASS
3182 021646 105737 003526      TSTB    STAFLE      ;AND DIAG WAS NEITHER STARTED
3183 021652 001010      BNE      50346$
3184 021654      READEF #EF.RES      ;NOR
      021654 012700 000037      MOV      #EF.RES,R0
      021660 104447      TRAP     C$REFG
3185 021662 103402      BCS      50347$      ;IFCOND CC THEN ;RESTARTED
3186 021664 005103      COM      R3      ;LET R3 := COMP R3 ;DO IT
3187
3188 021666 C00401      50347$: BR      50350$
3189 021670
3190 021670 005203      INC      R3      ;SET 1ST PASS IF NEW PASS AND
3191      ;RESTARTING

```


MISCELLANEOUS SECTIONS
INITIALIZE SECTION

MACRO M1113 25-MAY-82 09:51 PAGE 20-62

E 9

SEQ 0108

```

3192 021672          50350$:
3193
3194 021672 000401          BR      50351$
3195 021674          50346$:
3196 021674 005203          INC      R3
3197                                     ;SET 1ST PASS IF NEW PASS AND
3198 021676          50351$:                                     ;STARTING
3199                                     ;DO NOT UPDATE IT ON CONTINUE
3200 021676          50345$:                                     ;OR ON POWER FAIL
3201
3202 021676          50344$:
3203 021676 004737 017060      JSR      PC,FIRSTU          ;INIT DEVICE POINTER.
3204 021702 005002          CLR      R2          ;LET R2 := #0          ;INIT DEVICE COUNTER.
3205 021704          50352$: ;WHILE DEVTBL(R5) NE #END DO
3206 021704 026527 002604 177777  CMP      DEVTBL(R5),#END
3207 021712 001456          BEQ      50353$
3208 021714 005202          INC      R2          ;LET R2 := R2 + #1
3209 021716 010500          MOV      R5,R0          ;LET R0 := R5 SHIFT -1
3210 021720 006200          ASR      R0
3211 021722          GPHARD  R0,R0          ;GET HARDWARE P TABLE FROM SUPER.
3212 021722 104442          BCC      50354$          ;IFCOND CS THEN          TRAP      C$GPHRD
3213 021724 103044          MOV      (R0),TSDB(R5)          ;SAVE TSDB ADDRESS.
3214 021732 011065 002514      MOV      (R0)+,TSSR(R5)          ;SAVE TSSR ADDRESS.
3215 021732 012065 002524      MOV      (R0)+,TSVCT(R5)          ;SAVE INTERRUPT VECTOR ADDRESS.
3216 021736 062765 000002 002524      MOV      (R0),TSUNT(R5)          ;SAVE NUMBER OF DRIVE
3217 021744 012065 002534      MOV      (R0),TSNP          ;SAVE FOR PRINT OUT'S
3218 021750 011065 003532      SETVEC  TSVCT(R5),TS5INT(R5),#INTPRI
3219 021754 011037 003534
3219 021760          MOV      #INTPRI,-(SP)
3219 021760 012746 000340          MOV      TS5INT(R5),-(SP)
3219 021764 016546 002554          MOV      TSVCT(R5),-(SP)
3219 021770 016546 002534          MOV      #3,-(SP)
3219 021774 012746 000003          TRAP      C$SVEC
3219 022000 104437          ADD      #10,SP
3219 022002 062706 000010          ;SET UP INTERUPT PROCESSING CONDITIONS.
3220
3221 022006 005065 003472      CLR      INTFLG(R5)          ;CLEAR INTERRUPT FLAGS.
3222 022012 005703          R3          ;ACTUAL PASSCOUNT UPDATE PER R3
3223 022014 001410          BEQ      50355$
3224 022016 005703          TST      R3          ;IF R3 LT #0 THEN
3225 022020 002003          BGE      50356$
3226 022022 005265 003326      INC      PASCNT(R5)          ;LET PASCNT(R5) := PASCNT(R5) + #1
3227
3228 022026 000403          BR      50357$
3229 022030          50356$:
3230 022030 012765 000001 003326      MOV      #1,PASCNT(R5)          ;LET PASCNT(R5) := #1
3231
3232 022036          50357$:
3233
3234 022036          50355$:
3235
3236 022036          50354$:
3237 022036 005065 003376      CLR      RECCNT(R5)          ;CLEAR RECORD COUNT
3238 022042 004737 017126      JSR      PC,NEXTU          ;DO IT FOR ALL DEVICES.
3239
3240 022046 000716          BR      50352$
3241 022050          50353$:

```

MISCELLANEOUS SECTIONS
INITIALIZE SECTION

MACRO M1113 25-MAY-82 09:51 PAGE 20-63

F 9

SEQ 0109

```

3242
3243 022050 005702
3244 022052 001026
3245 022054
      022054 012746 005114
      022060 012746 000001
      022064 010600
      022066 104417
      022070 062706 000004
3246 022074
      022074 012727 000310
      022100 000000
      022102 013727 002116
      022106 000000
      022110 005367 177772
      022114 001375
      022116 005367 177756
      022122 001367
3247 022124
      022124 104422
3248 022126
      022126 104444
3249
3250 022130
3251
3252
3253 022130
      022130 012700 000000
      022134 104441
3254 022136 105737 002213
3255 022142 001033
3256 022144 032737 000020 003536
3257 022152 001027
3258 022154 004737 017060
3259 022160
3260 022160 026527 002604 177777
3261 022166 001421
3262 022170 105037 003530
3263 022174
      022174 012746 000340
      022200 012746 023706
      022204 012746 000004
      022210 012746 000003
      022214 104437
      022216 062706 000010
3264
3265 022222 012737 000001 003436
3266 022230 000404
3267 022232 000137 023060
3268
3269 022236
3270 022236 005237 003436
3271 022242
3272 022242 023727 003436 000025
3273 022250 003134
3274 022252 012775 002340 002514
3275 022260

```

```

TST R2
BNE 50360$
PRINTF #AUDRPM

```

DELAY 200.

BREAK

DOCLN

50360\$:

SETPRI #PRI00

```

TSTB IREC
BNE 1$
BIT #ADR,OPFLAG
BNE 1$
JSR PC,FIRSTU

```

50362\$:

```

CMP DEVTBL(R5),#END
BEQ 1$
CLRB TRAPD4
SETVEC #4,#TRAP4,#INTPRI

```

1\$:

```

MOV #1,TIME1
BR 50365$
JMP 50363$

```

50366\$:

INC TIME1

50365\$:

```

CMP TIME1,#25
BGT 4$
MOV #GSCPK,@TSDB(R5)
DELAY 25

```

;IF THERE ARE NO UNITS:

;PRINT ALL UNITS DROPPED,

```

MOV #AUDRPM,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #4,SP

```

;GO TO SUPERVISOR, WAIT 2 SECONDS.

```

MOV #200.,(PC)+
WORD 0
MOV L$DLY,(PC)+
WORD 0
DEC -6(PC)
BNE -4
DEC -22(PC)
BNE -20

```

;GO TO SUPERVISOR, CHECK TTY.

TRAP C\$BRK

;DO CLEAN CODE + ABORT PASS.

TRAP C\$DCLN

;LOWER CPU PRIORITY TO 0

```

MOV #PRI00,R0
TRAP C$SPRI

```

;IF ERROR RECOVERY IS ENABLED

;AND AUTO-DROP NOT CALLED, THEN SET UP FOR FIRST
;WHILE THERE ARE MORE DEVICES:

;CLEAR TRAP FLAG
;SET VECTOR 4,PRIORITY @6

```

MOV #INTPRI,-(SP)
MOV #TRAP4,-(SP)
MOV #4,-(SP)
MOV #3,-(SP)
TRAP C$SVEC
ADD #10,SP

```

;START 3.5 MINUTE COUNTER
;INCR TIME1 FROM #1 TO #25 BY #1

;AND GET UNITS STATUS
;WAIT

```

CLRVEC    #4

TSTB      TRAPD4
BEQ        2$
INC        FTLCNT(R5)
PRINTF     #NODEV,TSSR(R5)

MOV        DEVTBL(R5),DROPN
MOV        R5,R0
ASR        R0
DOCU       R0

DOCLN

CLRB      TRAPD4
SETVEC    #4,#TRAP4,#INTPRI

TST       @TSSR(R5)
DELAY     25

CLRVEC    #4

TSTB      TRAPD4
BEQ        3$
INC        FTLCNT(R5)
PRINTF     #NODEV,TSSR(R5)

```

```

MOV      #25,(PC)+
.WORD    0
MOV      LSDLY,(PC)+
.WORD    0
DEC      -6(PC)
BNE      -.4
DEC      -22(PC)
BNE      .-20

;CLEAR VECTOR AT 4

MOV      #4,R0
TRAP     C$CVEC

;IFB TRAPD4 NE #0 THEN

;LET FTLCNT(R5) := FTLCNT(R5) + #1
;PRINT ERROR

MOV      TSSR(R5),-(SP)
MOV      #NODEV, -(SP)
MOV      #2, -(SP)
MOV      SP,R0
TRAP     C$PNTF
ADD      #6,SP

;SAVE # OF UNIT TO BE DROPPED.
;R0=LOGICAL DEVICE NUMBER

;DROP THE UNIT

TRAP     C$DODU
; EXEC BGNDU-ENDDU CODE IF IDU = 0
;DO CLEAN &ABORT

TRAP     C$DCLN

;CLEAR TRAP FLAG
;SET VECTOR 4,PRIORITY @6

MOV      #INTPRI, -(SP)
MOV      #TRAP4, -(SP)
MOV      #4, -(SP)
MOV      #3, -(SP)
TRAP     C$SVEC
ADD      #10,SP

;CHECK FOR ADDRESS
;WAIT

MOV      #25,(PC)+
.WORD    0
MOV      LSDLY,(PC)+
.WORD    0
DEC      -6(PC)
BNE      -.4
DEC      -22(PC)
BNE      .-20

;CLEAR VECTOR AT 4

MOV      #4,R0
TRAP     C$CVEC

;IFB TRAPD4 NE #0 THEN

;LET FTLCNT(R5) := FTLCNT(R5) + #1
;PRINT ERROR

MOV      TSSR(R5), -(SP)
MOV      #NODEV, -(SP)

```

MISCELLANEOUS SECTIONS
INITIALIZE SECTION

MACRO M1113 25-MAY-82 09:51 PAGE 20-65 H 9

SEQ 0111

022510	012746	000002							
022514	010600							MOV	#2,-(SP)
022516	104417							MOV	SP,R0
022520	062706	000006						TRAP	C\$PNTF
3297 022524	016537	002604	017366					ADD	#6,SP
3298 022532	010500			MOV	DEVTBL(R5),DROPN				
3299 022534	006200			MOV	R5,R0				;SAVE # OF UNIT TO BE DROPPED.
3300 022536				ASR	R0				;R0=LOGICAL DEVICE NUMBER
022536	104451			DODU	R0				;DROP THE UNIT
3301								TRAP	C\$DODU
3302 022540				DOCLN					; EXEC BGNDU-ENDDU CODE IF IDU = 0
022540	104444								;DO CLEAN &ABORT
3303								TRAP	C\$DOCLN
3304 022542	003127			4\$: BGT	50367\$				
3305									
3306 022544	004737	007766		3\$: JSR	PC,SETDEF				;SET UNIT NUMBER
3307 022550	010475	002514		MOV	R4,@TSDB(R5)				
3308 022554				DELAY	25				
022554	012727	000025						MOV	#25,(PC)+
022560	000000							.WORD	0
022562	013727	002116						MOV	L\$DLY,(PC)+
022566	000000							.WORD	0
022570	005367	177772						DEC	-6(PC)
022574	001375							BNE	.-4
022576	005367	177756						DEC	-22(PC)
022602	001367							BNE	.-20
3309 022604	012775	002340	002514	MOV	#GSCP,K,@TSDB(R5)				;AND GET UNITS STATUS
3310 022612				DELAY	25				;WAIT
022612	012727	000025						MOV	#25,(PC)+
022616	000000							.WORD	0
022620	013727	002116						MOV	L\$DLY,(PC)+
022624	000000							.WORD	0
022626	005367	177772						DEC	-6(PC)
022632	001375							BNE	.-4
022634	005367	177756						DEC	-22(PC)
022640	001367							BNE	.-20
3311 022642	032775	000200	002524	BIT	#TS.SSR,@TSSR(R5)				;IF #TS.SSR SETIN @TSSR(R5) THEN
3312 022650	001420			BEQ	50370\$				
3313 022652	032775	000100	002524	BIT	#TS.OFL,@TSSR(R5)				;IF #TS.OFL NOTSETIN @TSSR(R5) THEN
3314 022660	001001			BNE	50371\$				
3315 022662	000457			BR	50364\$;EXIT COUNTER WHEN UNIT ON LINE
3316									
3317 022664				50371\$:					
3318 022664					PRINTF #OFLINM,TSNP				;PRINT UNIT OFF LINE EVERY 10 SEC
022664	013746	003534						MOV	TSNP,-(SP)
022670	012746	005456						MOV	#OFLINM,-(SP)
022674	012746	000002						MOV	#2,-(SP)
022700	010600							MOV	SP,R0
022702	104417							TRAP	C\$PNTF
022704	062706	000006						ADD	#6,SP
3319									
3320 022710				50372\$:					
3321									
3322 022710	000412			BR	50373\$				
3323 022712				50370\$:					
3324 022712					PRINTF #NRDYM,DEVTBL(R5)				
022712	016546	002604						MOV	DEVTBL(R5),-(SP)

Address	Instruction	Comments	Address	Instruction	Comments
022716	012746	023656		MOV	#NRDYM, -(SP)
022722	012746	000002		MOV	#2, -(SP)
022726	010600			MOV	SP, R0
022730	104417			TRAP	C\$PNTF
022732	062706	000006		ADD	#6, SP
3325					
3326	022736		50373\$:		
3327	022736	012737 000001 003440	MOV	#1, TIME2	; INCR TIME2 FROM #1 TO #13 BY #1
3328	022744	000402	BR	50374\$	
3329	022746		50375\$:		
3330	022746	005237 003440	INC	TIME2	
3331	022752		50374\$:		
3332	022752	023727 003440 000013	CMP	TIME2, #13	
3333	022760	003016	BGT	50376\$	
3334	022762		DELAY	100.	; WAIT FOR UNIT TO BE SET ON-LINE
	022762	012727 000144			
	022766	000000			MOV #100., (PC)+
	022770	013727 002116			.WORD 0
	022774	000000			MOV L\$DLY, (PC)+
	022776	005367 177772			.WORD 0
	023002	001375			DEC -6(PC)
	023004	005367 177756			BNE .-4
	023010	001367			DEC -22(PC)
3335	023012		BREAK		BNE .-20
	023012	104422			; ALLOW TERMINAL INTERRUPT
3336	023014	000754	BR	50375\$	TRAP C\$BRK
3337	023016		50376\$:		
3338	023016	000137 022236	JMP	50366\$	
3339	023022		50367\$:		
3340	023022		50364\$:		
3341	023022		CLRVEC	#4	; CLEAR VECTOR AT 4
	023022	012700 000004			
	023026	104436			MOV #4, R0
3342	023030	023727 003436 000025	CMP	TIME1, #25	TRAP C\$CVEC
3343	023036	003404	BLE	50377\$; IF OFF LINE FOR 3.5 MINUTES
3344	023040	004737 012734			
3345	023044	004737 013430	JSR	PC, MOVMSG	; GET MESSAGE PACKET
3346			JSR	PC, TCC1	; PRINT ERROR AND DROP OFF LINE UNIT
3347	023050		50377\$:		
3348					
3349	023050	004737 017126	JSR	PC, NEXTU	; REPEAT UNTIL ON LINE OR TIMED OUT.
3350					; SET UP FOR NEXT UNIT.
3351	023054	000137 022160	JMP	50362\$	
3352					
3353	023060		50363\$:		
3354	023060		50361\$:		
3355	023060	105737 003527	TSTB	PWRFLG	; IFB PWRFLG EQ #0 THEN
3356	023064	001026	BNE	50400\$	
3357	023066				
	023066	104431			MEMORY DATAWT ; REQUEST MEMORY FROM SUPER FOR RD/WR BUFFERS.
	023070	010037 003406			TRAP C\$MEM
3358	023074	013737 003406 003410	MOV	DATAWT, DATARD	MOV R0, DATAWT
3359	023102	062737 004000 003410	ADD	#DATCNT, DATARD	; SET RD BFR ADDRESS
3360	023110	027727 160272 004000	CMP	@DATAWT, #DATCNT	; WHEN NOT ENOUGH FREE MEMO AVAILABLE
3361	023116	002011	BGE	50401\$	
3362	023120				
	023120	012746 023166	PRINTF	#MEMOM	; WARN OPERATOR
					MOV #MEMOM, -(SP)

```

INITIALIZE SECTION

023124 012746 000001
023130 010600
023132 104417
023134 062706 000004
3363 023140 DOCLN ;AND ABORT PASS
023140 104444 TRAP C$DCLN
3364 ;DIAG MUST BE RE-LOADED IN A CPU WITH LARGER MEMO
3365 023142 50401$:
3366 50400$:
3367 023142
3368
3369 023142 105037 002214 CLR B CHGFLG ;CLR CHANGE CMD SEQ TBL FLAG.
3370 023146 012703 003526 MOV #ENDFLG,R3 ;LET R3 := #ENDFLG
3371 023152 004737 012664 JSR PC,CLRERR ;CLEAR ALL FLAGS.
3372 023156 105037 003527 CLR B PWRFLG ;CLEAR THE POWER FAIL FLAG.
3373
3374 023162 EXIT INIT
023162 104432
023164 000104 TRAP C$EXIT
3375 023166 045 101 106 MEMOM: .ASCII /%AFREE MEMO TOO SMALL FOR RD-WR BFRS%/
023171 122 105 105
023174 040 115 105
023177 115 117 040
023202 124 117 117
023205 040 123 115
023210 101 114 114
023213 040 106 117
023216 122 040 122
023221 104 055 127
023224 122 040 102
023227 106 122 123
023232 045 116
3376 023234 045 101 122 .ASCIIZ /%ARE-LOAD IN LARGER MEMO%/
023237 105 055 114
023242 117 101 104
023245 040 111 116
023250 040 114 101
023253 122 107 105
023256 122 040 115
023261 105 115 117
023264 045 116 000
3377 .EVEN
3378
3379 023270 ENDINIT
023270 L10012:
023270 104411 TRAP C$INIT
3380
3381 .SBTTL AUTO DROP SECTION
3382
3383 ;++
3384 ;SECTION EXECUTED AFTER THE INIT CODE WHEN 'ADR' FLAG IS SET BY OPERATOR
3385 ;SECTION CHEKS FOR A VALID INTERFACE LOCATION. DROPS UNIT IF NO RESPONSE
3386 ;FROM INTERFACE
3387 ;--
3388
3389 023272 BGNAUTO
023272 L$AUTO::

```

MISCELLANEOUS SECTIONS
AUTO DROP SECTION

MACRO M1113 25-MAY-82 09:51 PAGE 20-68

K 9

SEQ 0114

```

3390
3391 023272 004737 017060
3392 023276
3393 023276 026527 002604 177777
3394 023304 001525
3395 023306 105037 7530
3396 023312
      023312 012746 000340
      023316 012746 023706
      023322 012746 000004
      023326 012746 000003
      023332 104437
      023334 062706 000010
3397 023340 017502 002514
3398 023344
      023344 012700 000004
      023350 104436
3399 023352 105737 003530
3400 023356 001423
3401 023360 005265 003366
3402 023364
      023364 016546 002514
      023370 012746 023562
      023374 012746 000002
      023400 010600
      023402 104417
      023404 062706 000006
3403 023410 016537 002604 017366
3404 023416 010500
3405 023420 006200
3406 023422
      023422 104451
3407
3408 023424 000452
3409 023426
3410 023426 012775 002340 002514
3411 023434 004737 012700
3412 023440 032775 000200 002524
3413 023446 001423
3414 023450 032775 000100 002524
3415 023456 001416
3416 023460 005265 003366
3417 023464
      023464 013746 003534
      023470 012746 005456
      023474 012746 000002
      023500 010600
      023502 104417
      023504 062706 000006
3418 023510 004737 017302
3419
3420 023514
3421
3422 023514 000416
3423 023516
3424 023516 005265 003366
3425 023522

50402$: JSR PC,FIRSTU ;FIND FIRST UNIT
        ;WHILE DEVTBL(R5) NE #END DO
        CMP DEVTBL(R5),#END
        BEQ 50403$
        CLRB TRAPD4 ;LET TRAPD4 :B= #0
        SETVEC #4,#TRAP4,#INTPRI ;SET VECTOR 4
        MOV #INTPRI,-(SP)
        MOV #TRAP4,-(SP)
        MOV #4,-(SP)
        MOV #3,-(SP)
        TRAP C$SVEC
        ADD #10,SP
        MOV #4,R0
        TRAP C$CVEC

        MOV @TSDB(R5),R2 ;ADDRESS TS05 INTERFACE
        CLRVEC #4 ;CLEAR VECTOR AT 4

        TSTB TRAPD4 ;IFB TRAPD4 NE #0 THEN
        BEQ 50404$
        INC FTLCNT(R5) ;LET FTLCNT(R5) := FTLCNT(R5) + #1
        PRINTF #AUTODM,TSDB(R5) ;PRINT ERROR
        MOV TSDB(R5),-(SP)
        MOV #AUTODM,-(SP)
        MOV #2,-(SP)
        MOV SP,R0
        TRAP C$PNTF
        ADD #6,SP

        MOV DEVTBL(R5),DROPN ;SAVE # OF UNIT TO BE DROPPED.
        MOV R5,R0 ;R0=LOGICAL DEVICE NUMBER
        ASR R0
        DODU R0 ;DROP THE UNIT: EXEC BGNDU-ENDDU CODE IF IDU = 0
        TRAP C$DODU

50404$: BR 50405$
        MOV #GSCPK,@TSDB(R5) ;SEND GET STATUS COMMAND
        JSR PC,WSSR ;WAIT
        BIT #TS.SSR,@TSSR(R5) ;IF #TS.SSR SETIN @TSSR(R5) THEN
        BEQ 50406$
        BIT #TS.OFL,@TSSR(R5) ;IF #TS.OFL SETIN @TSSR(R5) THEN
        BEQ 50407$
        INC FTLCNT(R5) ;LET FTLCNT(R5) := FTLCNT(R5) + #1
        PRINTF #OFLINM,TSNP
        MOV TSNP,-(SP)
        MOV #OFLINM,-(SP)
        MOV #2,-(SP)
        MOV SP,R0
        TRAP C$PNTF
        ADD #6,SP

        JSR PC,DROPUA

50407$:
50406$: BR 50410$
        INC FTLCNT(R5) ;LET FTLCNT(R5) := FTLCNT(R5) + #1
        PRINTF #NRDYM,DEVTBL(R5)

```

SEQ 0115

023522 016546 002604
 023526 012746 023656
 023532 012746 000002
 023536 010600
 023540 104417
 023542 062706 000006
 3426 023546 004737 017302

MOV DEVTBL(R5),-(SP)
 MOV #NRDYM, -(SP)
 MOV #2, -(SP)
 MOV SP, R0
 TRAP C\$PNTF
 ADD #6, SP

JSR PC,DROPUA

3427
 3428 023552
 3429
 3430 023552
 3431 023552 004737 017126
 3432
 3433 023556 000647
 3434 023560
 3435
 3436 023560
 023560
 023560 104461

50410\$:

50405\$:

JSR PC,NEXTU

BR 50402\$

50403\$:

ENDAUTO

L10013:

TRAP C\$AUTO

3437
 3438 023562 045 101 102
 023565 125 123 040
 023570 124 122 101
 023573 120 040 101
 023576 124 040 045
 023601 117 066 045
 023604 116
 3439 023605 045 101 111
 023610 116 124 105
 023613 122 106 101
 023616 103 105 040
 023621 102 101 104
 023624 040 117 122
 023627 040 116 117
 023632 124 040 123
 023635 105 124 040
 023640 124 117 040
 023643 101 102 117
 023646 126 105 040
 023651 101 104 045
 023654 116 000
 3440 023656 045 101 125
 023661 116 111 124
 023664 040 045 104
 023667 061 045 101
 023672 040 116 117
 023675 124 040 122
 023700 104 131 045
 023703 116 000

AUTODM: .ASCII /%ABUS TRAP AT %06%N/

.ASCIIZ /%AINTERFACE BAD OR NOT SET TO ABOVE AD%N/

NRDYM: .ASCIIZ /%AUNIT %D1%A NOT RDY%N/

.EVEN

: DEVICE BUS TRAP HANDLER
 : OUTPUT: TRAPD4 BYTE 1: TRAPED AT 4
 : 0: NO TRAP

3441
 3442
 3443
 3444
 3445
 3446
 3447 023706 105237 003530
 3448 023712 000002

TRAP4:: INCB TRAPD4;LET TRAPD4 :B= TRAPD4 + #1
 RTI

SEQ 0116

```

3449
3450
3451      .SBTTL  CLEANUP CODING SECTION
3452
3453      ;++
3454      ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
3455      ; AT THE END OF EACH PASS.
3456      ;--
3457
3458 023714      BGNCLN
3459 023714      L$CLEAN::
3460 023714 004737 017060      JSR      PC,FIRSTU      ;FIND FIRST UNIT.
3461 023720      50411$: ;WHILE DEVTBL(R5) NE #END DO
3462 023720 026527 002604 177777      CMP      DEVTBL(R5),#END
3463 023726 001410      BEQ      50412$
3464 023730 004737 012700      JSR      PC,WSSR      ;WAIT FOR UNIT READY OR TIMEOUT,
3465 023734      CLRVEC      TSVCT(R5)      ;RELEASE INTERRUPT VECTORS FOR ALL DEV.
3466 023734 016500 002534      MOV      TSVCT(R5),R0
3467 023740 104436      TRAP      C$CVEC
3468 023742 004737 017126      JSR      PC,NEXTU      ;FIND NEXT UNIT.
3469 023746 000764      BR      50411$
3470 023750      50412$:
3471 023750      EXIT      CLN
3472 023752 104432      TRAP      C$EXIT
3473 023752 000002      .WORD      L10014-.
3474 023754      .EVEN
3475 023754      ENDCLN
3476 023754 104412      L10014:
3477      TRAP      C$CLEAN
3478
3479      .SBTTL  DROP UNIT SECTION
3480
3481      ;++
3482      ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3483      ; TO NO LONGER BE TESTED. THAT CODE SHALL BE EXECUTED WHEN DODU
3484      ; MACRO IS CALLED WHILE IDU FLAG IS NOT SET BY OPERATOR
3485      ;--
3486
3487 023756      BGNDU
3488 023756      L$DU::
3489 023756 010005      MOV      R0,R5      ;R5 = LOGICAL DEVICE NUMBER X 2.
3490 023760 006305      ASL      R5
3491 023762 012765 177774 002604      MOV      #NINUSE,DEVTBL(R5)      ;SET NOT IN USE FLAG FOR THE DEVICE.
3492 023770      CLRVEC      TSVCT(R5)      ;RELEASE THE INTERRUPT VECTOR.
3493 023770 016500 002534      MOV      TSVCT(R5),R0
3494 023774 104436      TRAP      C$CVEC
3495 023776      PRINTF      #DROPDM,DROPN      ;PRINT DROP DEVICE MESSAGE
3496 023776 013746 017366      MOV      DROPN,-(SP)
3497 024002 012746 005065      MOV      #DROPDM,-(SP)
3498 024006 012746 000002      MOV      #2,-(SP)
3499 024012 010600      MOV      SP,R0
3500 024014 104417      TRAP      C$PNTF
  
```

```

3491 024016 062706 000006                                ADD    #6,SP
      024022                                EXIT    DU
      024022 000167                                .WORD  JSJMP
      024024 000000                                .WORD  L10015-2-.
3492                                .EVEN
3493
3494 024026                                ENDDU
      024026                                L10015:
      024026 104453                                TRAP    C$DU
3495
3496                                .SBTTL  ADD UNIT SECTION
3497
3498                                ;++
3499                                ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3500                                ; TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
3501                                ; 'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
3502                                ;--
3503
3504 024030                                BGNAU
      024030                                L$AU::
3505 024030 010005                                MOV     R0,R5                                ;R5 = LOGICAL DEVICE NUMBER X 2.
3506 024032 006305                                ASL     R5
3507 024034 010065 002604                                MOV     R0,DEVTBL(R5)                        ;STORE UNIT # IN DEVICE TABLE.
3508 024040                                GPWARD  R0,R0                                ;GET HARDWARE P TABLE FROM SUPER.
      024040 104442                                TRAP    C$GPHRD
3509 024042 011065 002514                                MOV     (R0),TSDB(R5)                        ;SAVE TSDB ADDRESS.
3510 024046 012065 002524                                MOV     (R0)+,TSSR(R5)                      ;SAVE TSSR ADDRESS.
3511 024052 062765 000002 002524                                ADD     #2,TSSR(R5)
3512 024060 011065 002534                                MOV     (R0),TSVCT(R5)                      ;SAVE INTERRUPT VECTOR ADDRESS.
3513 024064 011065 003532                                MOV     (R0),TSUNT(R5)                      ;SAVE NUMBER OF DRIVE
3514 024070 011037 003534                                MOV     (R0),TSNP                           ;SAVE FOR PRINT OUT'S
3515 024074                                SETVEC  TSVCT(R5),TSSINT(R5),#INTPRI
      024074 012746 000340                                MOV     #INTPRI,-(SP)
      024100 016546 002554                                MOV     TSSINT(R5),-(SP)
      024104 016546 002534                                MOV     TSVCT(R5),-(SP)
      024110 012746 000003                                MOV     #3,-(SP)
      024114 104437                                TRAP    C$SVEC
      024116 062706 000010                                ADD     #10,SP
3516                                ;SET UP INTERRUPT PROCESSING CONDITIONS.
3517 024122 005065 003472                                CLR     INTFLG(R5)                          ;CLEAR INTERRUPT FLAGS.
3518
3519 024126                                EXIT    AU
      024126 000167                                .WORD  JSJMP
      024130 000000                                .WORD  L10016-2-.
3520
3521                                .EVEN
3522
3523 024132                                ENDAU
      024132                                L10016:
      024132 104452                                TRAP    C$AU
3524
3525
3526
3527                                .TITLE  HARDWARE TESTS
3528
3529                                .SBTTL  TEST 1:  BASIC FUNCTIONS.
3530

```

```

3531
3532      :++
3533      : TEST TO EXECUTE ALL TS05 FUNCTIONS.
3534      :--
3535 024134      BGNMOD
3536
3537 024134      BGNTST
3538 024134      T1::
3539 024140 105037 003515      CLRB      RANDOM      ;CLR THE RANDOM OPERATIONS FLAG.
3540 024140 105037 003514      CLRB      EXPBOT      ;CLR EXPECT BOT FLAG.
3541
3542 024144      BGNSUB      ;SUBTEST 1 - SET CHAR, DRIVE INIT, GET STATUS.
3543 024144 104402      T1.1:
3544 024146 004737 017060      JSR      PC,FIRSTU      ;FIND THE FIRST UNIT.
3545 024152 004737 007072      JSR      PC,SOFINIT      ;INIT DEVICE
3546 024156 103404      BCS      11$
3547 024160      ERRDF      2,NSSRM,STAERM      ;REPORT TS05 NOT READY
3548 024160 104455      TRAP      C$SERDF
3549 024162 000002      .WORD      2
3550 024164 004536      .WORD      NSSRM
3551 024166 006120      .WORD      STAERM
3552
3553 024170 004737 007466      11$: JSR      PC,MDSET      ;GO DO SETUP'S
3554 024174 012702 025052      MOV      #BFSEQ0,R2      ;ADR OF CMD SEQ.
3555 024200 004737 025026      JSR      PC,BFSEQ      ;SET UP CMD SEQ.
3556 024204 004737 010226      JSR      PC,EXALL      ;EXECUTE CMD SEQ ON ALL DEVICES.
3557 024210 004737 017060      JSR      PC,FIRSTU      ;FIND THE FIRST UNIT.
3558 024214      50413$: ;WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
3559 024214 026527 002604 177777      CMP      DEVTBL(R5),#END
3560 024222 001451      BEQ      50414$
3561 024224 016502 002544      MOV      MSGPKA(R5),R2      ;GET MSG PACKET ADR.
3562 024230 062702 000012      ADD      #12,R2      ;LET R2 := R2 + #12 ;GET XSTAT2 ADR.
3563 024234 011265 002564      MOV      (R2),TS5CL(R5)      ;STORE CODE LEVEL FROM DTR BYTE.
3564 024240 042765 177700 002564      BIC      #177700,TS5CL(R5)
3565 024246 011265 002574      MOV      (R2),TS5SW(R5)      ;STORE SWITCH SETTINGS
3566 024252 042765 177477 002574      BIC      #177477,TS5SW(R5)
3567 024260      PRINTF      #CODELM,DEVTBL(R5),TS5CL(R5)
3568 024260 016546 002564      MOV      TS5CL(R5),-(SP)
3569 024264 016546 002604      MOV      DEVTBL(R5),-(SP)
3570 024270 012746 004162      MOV      #CODELM,-(SP)
3571 024274 012746 000003      MOV      #3,-(SP)
3572 024300 010600      MOV      SP,R0
3573 024302 104417      TRAP      C$PNTF
3574 024304 062706 000010      ADD      #10,SP
3575
3576 024310      PRINTF      #SWSET,DEVTBL(R5),TS5SW(R5) ;PRINT THE TS05 MICROCODE LEVEL.
3577 024310 016546 002574      MOV      TS5SW(R5),-(SP)
3578 024314 016546 002604      MOV      DEVTBL(R5),-(SP)
3579 024320 012746 004231      MOV      #SWSET,-(SP)
3580 024324 012746 000003      MOV      #3,-(SP)
3581 024330 010600      MOV      SP,R0
3582 024332 104417      TRAP      C$PNTF
3583 024334 062706 000010      ADD      #10,SP
3584
3585      ;PRINT THE TS05 SWITCH SETTINGS.

```

3567	024340			50415\$:			
3568	024340	004737	017126		JSR	PC,NEXTU	:FIND NEXT UNIT.
3569							
3570	024344	000723			BR	50413\$	
3571	024346			50414\$:			
3572							
3573	024346				ENDSUB		
	024346			L10020:			
	024346	104403					TRAP C\$ESUB
3574							
3575	024350				BGNSUB		:SUBTEST 2 - REWIND.
	024350			T1.2:			
	024350	104402					TRAP C\$BSUB
3576							
3577	024352	012702	025124		MOV	#BFSEQ1,R2	:ADR OF CMD SEQ.
3578	024356	004737	025026		JSR	PC,BFSEQ	:SET UP CMD SEQ.
3579	024362	004737	010226		JSR	PC,EXALL	:EXECUTE CMD SEQ ON ALL DEVICES.
3580	024366	105037	003526		CLRB	STAFLG	:CLEAR START FLAG
3581	024372				ENDSUB		
	024372			L10021:			
	024372	104403					TRAP C\$ESUB
3582							
3583	024374				BGNSUB		:SUBTEST 3 - WRITE/VERIFY.
	024374			T1.3:			
	024374	104402					TRAP C\$BSUB
3584							
3585	024376	012702	025136		MOV	#BFSEQ2,R2	:ADR OF CMD SEQ.
3586	024402	004737	025026		JSR	PC,BFSEQ	:SET UP CMD SEQ.
3587	024406	004737	010226		JSR	PC,EXALL	:EXECUTE CMD SEQ ON ALL DEVICES.
3588	024412				ENDSUB		
	024412			L10022:			
	024412	104403					TRAP C\$ESUB
3589							
3590	024414				BGNSUB		:SUBTEST 4 - WRITE TAPE MARK, ERASE.
	024414			T1.4:			
	024414	104402					TRAP C\$BSUB
3591							
3592	024416	012702	025230		MOV	#BFSEQ3,R2	:ADR OF CMD SEQ.
3593	024422	004737	025026		JSR	PC,BFSEQ	:SET UP CMD SEQ.
3594	024426	004737	010226		JSR	PC,EXALL	:EXECUTE CMD SEQ ON ALL DEVICES.
3595	024432				ENDSUB		
	024432			L10023:			
	024432	104403					TRAP C\$ESUB
3596							
3597	024434				BGNSUB		:SUBTEST 5 - SPACE FILES.
	024434			T1.5:			
	024434	104402					TRAP C\$BSUB
3598							
3599	024436	012702	025302		MOV	#BFSEQ4,R2	:ADR OF CMD SEQ.
3600	024442	004737	025026		JSR	PC,BFSEQ	:SET UP CMD SEQ.
3601	024446	004737	010226		JSR	PC,EXALL	:EXECUTE CMD SEQ ON ALL DEVICES.
3602	024452				ENDSUB		
	024452			L10024:			
	024452	104403					TRAP C\$ESUB
3603							
3604	024454				BGNSUB		:SUBTEST 6 - SPACE RECORDS.
	024454			T1.6:			

3605	024454	104402				TRAP	C\$BSUB
3606	024456	012702	025344	MOV	#BFSEQ5,R2		
3607	024462	004737	025026	JSR	PC,BFSEQ		:ADR OF CMD SEQ.
3608	024466	004737	010226	JSR	PC,EXALL		:SET UP CMD SEQ.
3609	024472			ENDSUB			:EXECUTE CMD SEQ ON ALL DEVICES.
	024472			L10025:			
	024472	104403				TRAP	C\$ESUB
3610	024474						
3611	024474			T1.7:	BGNSUB		:SUBTEST 7 - WRITE RETRY.
	024474	104402				TRAP	C\$BSUB
3612	024476	012702	025416	MOV	#BFSEQ6,R2		
3613	024502	004737	025026	JSR	PC,BFSEQ		:ADR OF CMD SEQ.
3614	024506	004737	010226	JSR	PC,EXALL		:SET UP CMD SEQ.
3615	024512			ENDSUB			:EXECUTE CMD SEQ ON ALL DEVICES.
3616	024512			L10026:			
	024512	104403				TRAP	C\$ESUB
3617	024514						
3618	024514			T1.8:	BGNSUB		:SUBTEST 8 - READ REV RETRY.
	024514	104402				TRAP	C\$BSUB
3619	024516	012702	025470	MOV	#BFSEQ7,R2		
3620	024522	004737	025026	JSR	PC,BFSEQ		:ADR OF CMD SEQ.
3621	024526	004737	010226	JSR	PC,EXALL		:SET UP CMD SEQ.
3622	024532			ENDSUB			:EXECUTE CMD SEQ ON ALL DEVICES.
3623	024532			L10027:			
	024532	104403				TRAP	C\$ESUB
3624	024534						
3625	024534			T1.9:	BGNSUB		:SUBTEST 9 - READ FWD RETRY.
	024534	104402				TRAP	C\$BSUB
3626	024536	012702	025522	MOV	#BFSEQ8,R2		
3627	024542	004737	025026	JSR	PC,BFSEQ		:ADR OF CMD SEQ.
3628	024546	004737	010226	JSR	PC,EXALL		:SET UP CMD SEQ.
3629	024552			ENDSUB			:EXECUTE CMD SEQ ON ALL DEVICES.
3630	024552			L10030:			
	024552	104403				TRAP	C\$ESUB
3631	024554						
3632	024554			T1.10:	BGNSUB		:SUBTEST 10- CLEAN.
	024554	104402				TRAP	C\$BSUB
3633	024556	012702	025554	MOV	#BFSEQ9,R2		
3634	024562	004737	025026	JSR	PC,BFSEQ		:ADR OF CMD SEQ.
3635	024566	004737	010226	JSR	PC,EXALL		:SET UP CMD SEQ.
3636	024572			ENDSUB			:EXECUTE CMD SEQ ON ALL DEVICES.
3637	024572			L10031:			
	024572	104403				TRAP	C\$ESUB
3638	024574						
3639	024574			T1.11:	BGNSUB		:SUBTEST 11 - WTV SWAPPED DATA BYTES.
	024574	104402				TRAP	C\$BSUB
3640	024576	012702	025576	MOV	#BFSEQ10,R2		:ADR OF CMD SEQ.

```

3641 024602 004737 025026      JSR      PC,BFSEQ      ;SET UP CMD SEQ.
3642 024606 004737 010226      JSR      PC,EXALL      ;WRITE/VERIFY RECORDS 1 AND 2.
3643 024612 112737 000001 003520  MOVB     #1,SWBFLG      ;ENABLE BYTE SWAPPING.
3644 024620 004737 010226      JSR      PC,EXALL      ;WRITE/VERIFY RECORDS 3 AND 4.
3645 024624 105037 003520      CLRB     SWBFLG      ;DISABLE BYTE SWAPPING.
3646 024630      ENDSUB
                                L10032:
                                024630      104403
3647 024632 013702 003406      MOV      DATAW,R2      ;INIT WRITE BUFFER POINTER. TRAP C$ESUB
3648 024636 062702 000012      ADD      #10.,R2
3649 024642      50416$: ;WHILE R2 NE DATAW DO ;UNTIL 10 BYTES HAVE BEEN SWAPPED.
3650 024642 020237 003406      CMP      R2,DATAW
3651 024646 001402      BEQ      50417$
3652 024650 000342      SWAB     -(R2)      ;SWAP DATA BYTES IN WRITE BUFFER.
3653
3654 024652 000773      BR      50416$
3655 024654      50417$:
3656 024654 105237 003523      INCB     T1SWB      ;SET T1 SWAP BYTES FLAG FOR 'CKDATA' SUBR
3657
3658 024660      BGNSUB      ;SUBTEST 12 - READ SWAPPED DATA BYTES.
                                024660      104402
                                024660      012737 104401 003420      MOV      #RDR,CMDWRD      TRAP C$BSUB
3659 024662 012737 104401 003420      JSR      PC,VFEXC      ;VERIFY ODD LENGTH SWAP (RECORD 4).
3660 024670 004737 016130      MOV      #12,CMDPKT+CP.CNT      ;CHANGE BYTE COUNT TO 10.
3661 024674 012737 000012 002336      JSR      PC,VFEXC      ;VERIFY EVEN LENGTH SWAP (RECORD 3).
3662 024702 004737 016130      MOVB     #1,SWBFLG      ;ENABLE BYTE SWAPPING.
3663 024706 112737 000001 003520      MOV      #11,CMDPKT+CP.CNT      ;CHANGE BYTE COUNT TO 9.
3664 024714 012737 000011 002336      JSR      PC,VFEXC      ;VERIFY ODD LENGTH SWAP (RECORD 2).
3665 024722 004737 016130      MOV      #12,CMDPKT+CP.CNT      ;CHANGE BYTE COUNT TO 10.
3666 024726 012737 000012 002336      JSR      PC,VFEXC      ;VERIFY EVEN LENGTH SWAP (RECORD 1).
3667 024734 004737 016130      MOV      #11,CMDPKT+CP.CNT      ;CHANGE BYTE COUNT TO 9.
3668 024740 012737 104001 003420      JSR      PC,VFEXC      ;VERIFY ODD LENGTH SWAP (RECORD 2).
3669 024746 004737 016130      CLRB     SWBFLG      ;DISABLE BYTE SWAPPING.
3670 024752 012737 000011 002336      MOV      #12,CMDPKT+CP.CNT      ;CHANGE BYTE COUNT TO 10.
3671 024760 004737 016130      JSR      PC,VFEXC      ;VERIFY EVEN LENGTH SWAP (RECORD 3).
3672 024764 105037 003520      MOV      #11,CMDPKT+CP.CNT      ;CHANGE BYTE COUNT TO 9.
3673 024770 012737 000012 002336      JSR      PC,VFEXC      ;VERIFY ODD LENGTH SWAP (RECORD 4).
3674 024776 004737 016130      ENDSUB
3675 025002 012737 000011 002336      L10033:
3676 025010 004737 016130      025014      104403
3677      025014
3678      025014      TRAP C$ESUB
3679
3680 025016 105037 003523      CLRB     T1SWB      ;CLEAR T1 SWAP BYTES FLAG
3681
3682 025022      EXIT     TST
                                025022      104432
                                025024      000574      TRAP C$EXIT
                                                .WORD L10017-.
3683
3684      ; SUBROUTINE TO MOVE A COMMAND SEQUENCE TO THE SEQUENCE TABLE.
3685      ; INPUTS:      R2 = FWA OF COMMAND SEQUENCE.
3686      ; OUTPUTS:
3687      ; REGISTERS:
3688      ; CALLS:
3689

```

3690	025026	012701	003540	BFSEQ:: MOV	#CMDSEQ,R1	;INIT SEQ TABLE ADDRESS.
3691	025032			50420\$: ;WHILE	(R2) NE #END DO	;WHILE THERE ARE MORE COMMANDS:
3692	025032	021227	177777		CMP (R2),#END	
3693	025036	001402			BEQ 50421\$	
3694	025040	012221			MOV (R2)+,(R1)+	;MOVE COMMANDS TO SEQ TABLE.
3695						
3696	025042	000773			BR 50420\$	
3697	025044			50421\$:		
3698	025044	012711	177777		MOV #END,(R1)	;STORE END OF SEQUENCE CODE.
3699	025050	000207			RTS PC	;RETURN.
3700						
3701						
3702				:	BASIC FUNCTION COMMAND SEQUENCE	
3703						
3704	025052	140004		BFSEQ0: .WORD	SCH	;SET CHAR. 200. (1)
3705	025054	000200			200	
3706	025056	000001			1	
3707	025060	000000			0	
3708	025062	100013			DRI	;DRIVE INIT. (2)
3709	025064	000001			1	
3710	025066	000001			1	
3711	025070	000000			0	
3712	025072	140004			SCH	;SET CHAR. 20 (3)
3713	025074	000020			20	
3714	025076	000001			1	
3715	025100	000000			0	
3716	025102	100017			GES	;GET STATUS. (4)
3717	025104	000001			1	
3718	025106	000001			1	
3719	025110	000000			0	
3720	025112	140004			SCH	;SET CHAR. 40. (5)
3721	025114	000040			40	
3722	025116	000001			1	
3723	025120	000000			0	
3724	025122	177777		.WORD	END	
3725						
3726	025124	102010		BFSEQ1:	RWD	;REWIND TWICE. (6)
3727	025126	000001			1	
3728	025130	000002			2	
3729	025132	000000			0	
3730	025134	177777		.WORD	END	
3731						
3732	025136	104105		BFSEQ2:	WTV	;WRITE/VERIFY PAT 1. (7)
3733	025140	004000			DATCNT	
3734	025142	000001			1	
3735	025144	000001			1	
3736	025146	104105			WTV	;WTV PAT 2. (8)
3737	025150	004000			DATCNT	
3738	025152	000001			1	
3739	025154	000002			2	
3740	025156	104105			WTV	;WTV PAT 3. (9)
3741	025160	004000			DATCNT	
3742	025162	000001			1	
3743	025164	000003			3	
3744	025166	104105			WTV	;WTV PAT 4. (10)
3745	025170	004000			DATCNT	
3746	025172	000001			1	

3747	025174	000004	4			
3748	025176	104105	WTV		;WTV PAT 5.	(11)
3749	025200	004000	DATCNT			
3750	025202	000001	1			
3751	025204	000005	5			
3752	025206	104105	WTV		;WTV PAT 6.	(12)
3753	025210	004000	DATCNT			
3754	025212	000001	1			
3755	025214	000006	6			
3756	025216	104105	WTV		;WTV PAT 0.	(13)
3757	025220	004000	DATCNT			
3758	025222	000001	1			
3759	025224	000000	0			
3760	025226	177777	.WORD	END		
3761						
3762	025230	100011	BFSEQ3:	WTM	;WRITE TAPE MARK.	(14)
3763	025232	000001	1			
3764	025234	000001	1			
3765	025236	000000	0			
3766	025240	104005	WRT		;WRITE 10 RECORDS.	(15)
3767	025242	004000	DATCNT			
3768	025244	000010	10			
3769	025246	000001	1			
3770	025250	100411	ERS		;ERASE 10 TIMES.	(16)
3771	025252	000001	1			
3772	025254	000010	10			
3773	025256	000000	0			
3774	025260	100011	WTM		;WRITE TAPE MARK.	(17)
3775	025262	000001	1			
3776	025264	000001	1			
3777	025266	000000	0			
3778	025270	101011	WTR		;WTM RETRY	(18)
3779	025272	000001	1			
3780	025274	000001	1			
3781	025276	000000	0			
3782	025300	177777	.WORD	END		
3783						
3784	025302	105410	BFSEQ4:	SFR	;SPACE 2 FILES REV.	(19)
3785	025304	000002	2			
3786	025306	000001	1			
3787	025310	000000	0			
3788	025312	105010	SFF		;SPACE 2 FILES FWD.	(20)
3789	025314	000002	2			
3790	025316	000001	1			
3791	025320	000000	0			
3792	025322	105410	SFR		;SPACE 2 FILES REV.	(21)
3793	025324	000001	1			
3794	025326	000002	2			
3795	025330	000000	0			
3796	025332	105010	SFF		;SPACE 2 FILES FWD.	(22)
3797	025334	000001	1			
3798	025336	000002	2			
3799	025340	000000	0			
3800	025342	177777	.WORD	END		
3801						
3802	025344	102010	BFSEQ5:	RWD	;REWIND.	(23)
3803	025346	000001	1			

3804	025350	000001	1		
3805	025352	000000	0		
3806	025354	104010	SRF		;SPACE 7 RECORDS FWD. (24)
3807	025356	000007	7		
3808	025360	000001	1		
3809	025362	000000	0		
3810	025364	104410	SRR		;SPACE 7 RECORDS REV. (25)
3811	025366	000007	7		
3812	025370	000001	1		
3813	025372	000000	0		
3814	025374	104010	SRF		;SPACE 7 RECORDS FWD. (26)
3815	025376	000001	1		
3816	025400	000007	7		
3817	025402	000000	0		
3818	025404	104410	SRR		;SPACE 7 RECORDS REV. (27)
3819	025406	000001	1		
3820	025410	000007	7		
3821	025412	000000	0		
3822	025414	177777	END	.WORD	
3823					
3824	025416	102010	RWD	BFSEQ6:	;REWIND. (28)
3825	025420	000001	1		
3826	025422	000001	1		
3827	025424	000000	0		
3828	025426	104005	WRT		;WRITE. (29)
3829	025430	004000	DATCNT		
3830	025432	000001	1		
3831	025434	000001	1		
3832	025436	105005	WRR		;WRITE RETRY. (30)
3833	025440	004000	DATCNT		
3834	025442	000001	1		
3835	025444	000001	1		
3836	025446	100011	WTM		;WRITE TAPE MARK.
3837	025450	000001	1		
3838	025452	000001	1		
3839	025454	000000	0		
3840	025456	105410	SFR		;SPACE 1 FILE REV.
3841	025460	000001	1		
3842	025462	000001	1		
3843	025464	000000	0		
3844	025466	177777	END	.WORD	
3845					
3846	025470	104401	RDR	BFSEQ7:	;READ REV. (31)
3847	025472	004000	DATCNT		
3848	025474	000001	1		
3849	025476	000001	1		
3850	025500	105401	RNR		;READ NEXT REV. (32)
3851	025502	004000	DATCNT		
3852	025504	000001	1		
3853	025506	000001	1		
3854	025510	125401	RNF		;READ NEXT FWD. (33)
3855	025512	004000	DATCNT		
3856	025514	000001	1		
3857	025516	000001	1		
3858	025520	177777	END	.WORD	
3859					
3860	025522	104001	RDF	BFSEQ8:	;READ FWD. (34)

3861	025524	004000		DATENT		
3862	025526	000001		1		
3863	025530	000001		1		
3864	025532	105001		RPF	;READ PREVIOUS FWD.	(35)
3865	025534	004000		DATENT		
3866	025536	000001		1		
3867	025540	000001		1		
3868	025542	125001		RPR	;READ PREVIOUS REV.	(36)
3869	025544	004000		DATENT		
3870	025546	000001		1		
3871	025550	000001		1		
3872	025552	177777	.WORD	END		
3873						
3874	025554	101012	BFSEQ9: .WORD	CLN	;CLEAN.	(37)
3875	025556	000001		1		
3876	025560	000001		1		
3877	025562	000000		0		
3878	025564	102010		RWD	;REWIND	(38)
3879	025566	000001		1		
3880	025570	000001		1		
3881	025572	000000		0		
3882	025574	177777	.WORD	END	;END OF SEQUENCE.	
3883						
3884	025576	104105	BFSE10:	WTV	;WRITE/VERIFY EVEN LENGTH.	(39)
3885	025600	000012		12		
3886	025602	000001		1		
3887	025604	000000		0		
3888	025606	104105		WTV	;WRITE/VERIFY ODD LENGTH.	(40)
3889	025610	000011		11		
3890	025612	000001		1		
3891	025614	000000		0		
3892	025616	177777	.WORD	END		
3893			.EVEN			
3894						
3895	025620		ENDTST			
	025620		L10017:			
	025620	104401			TRAP	C\$ETST
3896						
3897			.SBTTL	TEST 2: DATA RELIABILITY.		
3898						
3899			;++			
3900			; TEST TO CHECK THE DATA RELIABILITY OF THE TS05.			
3901			;-			
3902	025622		BGNTST			
	025622		T2::			
3903						
3904	025622	112737	000001	003515	MOV B #1,RANDOM	;SET THE RANDOM OPERATIONS FLAG.
3905	025630	105037	003514		CLRB EXPBOT	;CLEAR EXPECT BOT FLAG.
3906	025634	005037	003456		CLR WTMFLG	;CLEAR WRITE TAPE MARK FLAG
3907	025640	004737	017060		JSR PC,FIRSTU	;FIND THE FIRST UNIT.
3908	025644	004737	007072		JSR PC,SOFINIT	;INIT DEVICE
3909	025650	103404			BCS 11\$	
3910	025652				ERRDF 2,NSSRM,STAERM	;REPORT TS05 NOT READY
	025652	104455				TRAP
	025654	000002				.WORD
	025656	004536				.WORD
	025660	006120				.WORD
						2
						NSSRM
						STAERM

3911					
3912	025662	004737	007466	11\$:	JSR PC,MDSET ;GO DO SETUP'S
3913	025666	012702	004000		MOV #DATCNT,R2 ;SET UP THE RECORD LENGTH MASK,
3914	025672	005302			DEC R2
3915	025674	010237	003430		MOV R2,LENMSK ;ALLOW MAXIMUM BUFFER.
3916	025700	005137	003430		COM LENMSK
3917	025704	004737	010162		JSR PC,SETCH ;CMD 1 = SET CHARACTERISTIC.
3918	025710	105737	003526		TSTB STAFLG ;IFB STAFLG NE #0 THEN ;IF STARTING THEN:
3919	025714	001417			BEQ 50424\$
3920	025716	004737	010206		JSR PC,SETRW ;CMD2=REWIND
3921	025722	105037	003526		CLRB STAFLG ;LET STAFLG :B= #0 ;CLR START FLAG.
3922					
3923	025726			50422\$:	
3924	025726	012721	104105		MOV #WTV,(R1)+
3925	025732	012721	004000		MOV #DATCNT,(R1)+
3926	025736	012702	177740		MOV #RNOPSC,R2
3927	025742	005102			COM R2
3928	025744	010221			MOV R2,(R1)+
3929	025746	012721	000007		MOV #RANP,(R1)+
3930					
3931	025752			50423\$:	BREAK ; DO A SUPVSR BREAK FIRST.
	025752	104422			TRAP C\$BRK
3932					
3933	025754			50424\$:	
3934	025754	020127	003740		CMP R1,#SEQEND ;FILL SEQ TBL WITH RANDOM CMDS.
3935	025760	002012			BGE 50425\$
3936	025762	063737	003432	003434	ADD RANB,RANS ;LET RANS := RANS + RANB
3937	025770	013702	003434		MOV RANS,R2
3938	025774	042702	177741		BIC #177741,R2
3939	026000	004772	026136		JSR PC,@RANCMD(R2) ;SET UP A RANDOM CMD + BRK.
3940					
3941	026004	000763			BR 50424\$
3942	026006			50425\$:	
3943	026006	012711	177777		MOV #END,(R1) ;STORE END OF SEQUENCE CODE IN TABLE.
3944	026012	004737	010226		JSR PC,EXALL ;GO EXECUTE ALL CMDS IN SEQUENCE TABLE.
3945					
3946	026016	012701	003540		MOV #CMDSEQ,R1 ;INIT CMD SEQ TBL POINTER,
3947	026022	005702			TST R2 ;REPEAT UNTIL EOT IS REACHED
3948	026024	001752			BEQ 50423\$
3949	026026	105237	003524		INCB ALLEOT ;FLAG ALL UNITS @ EOT
3950	026032	000240			NOP
3951	026034	000240			NOP
3952	026036	000240			NOP
3953	026040	004737	027612		JSR PC,T\$WEOT ;WRITE ONE RECORD BEYOND EOT ON ALL UNITS
3954					
3955					
3956					
3957	026044	004737	026176		JSR PC,RANRD ;READ REV THAT EXTRA REC TO RE-POSITION THE TAPE
3958	026050	012737	177740	003544	MOV #RNOPSC,CMDSEQ+4 ;SET UP READ REV/FWD CMDS,
3959	026056	005137	003544		COM CMDSEQ+4 ;# OF RECORDS FOR READ REV.
3960	026062	013737	003544	003554	MOV CMDSEQ+4,CMDSEQ+14 ;# OF RECORDS FOR READ FORWARD.
3961	026070	012711	177777		MOV #END,(R1) ;STORE END OF SEQUENCE CODE IN SEQ TABLE.
3962	026074	004737	010226		JSR PC,EXALL ;GO EXECUTE READ REV/FWD OF LAST N RECORDS.
3963	026100	105037	003524		CLRB ALLEOT ;CLEAR ALL UNITS @ EOT FLAG
3964	026104	112737	000001	003517	MOVB #1,RPTFLG ;REQUEST PERFORMANCE REPORT DURING REWIND.
3965	026112	012701	003540		MOV #CMDSEQ,R1 ;INIT SEQ TBL POINTER,
3966	026116	004737	010206		JSR PC,SETRW ;STORE REWIND IN SEQ TBL.

```

3967 026122 012711 177777      MOV    #END,(R1)          ;STORE END IN SEQ TBL,
3968 026126 004737 010226      JSR PC,EXALL          ;EXECUTE REWIND CMD ON ALL UNITS
3969
3970 026132      EXIT    TST
      026132 104432
      026134 000320
                                     TRAP    C$EXIT
                                     .WORD    L10034-.

3971
3972      :      ADDRESSES OF SUBROUTINES USED TO SET UP RANDOM OPERATIONS IN
3973      :      THE DATA RELIABILITY TEST.
3974
3975 026136 026334      RANCMD: RANWV          ;WRITE/VERIFY.
3976 026140 026310      RANWR          ;WRITE.
3977 026142 026310      RANWR          ;WRITE.
3978 026144 026310      RANWR          ;WRITE.
3979 026146 026310      RANWR          ;WRITE.
3980 026150 026310      RANWR          ;WRITE.
3981 026152 026310      RANWR          ;WRITE.
3982 026154 026310      RANWR          ;WRITE.
3983 026156 026176      RANRD          ;READ.
3984 026160 026176      RANRD          ;READ.
3985 026162 026176      RANRD          ;READ.
3986 026164 026176      RANRD          ;READ.
3987 026166 026176      RANRD          ;READ.
3988 026170 026176      RANRD          ;READ.
3989 026172 026176      RANRD          ;READ.
3990 026174 026176      RANRD          ;READ.
3991
3992
3993      :      SUBROUTINE TO SET UP READ COMMANDS IN SEQUENCE TABLE.
3994      :      INPUTS:
3995      :      OUTPUTS:
3996      :      REGISTERS:      R2
3997      :      CALLS:
3998
3999 026176 005737 003456      RANRD:: TST    WTMFLG          ;WAS LAST CMD A WRITE?
4000 026202 001406      BEQ    1$          ;NO,GO AHEAD
4001 026204 004737 026346      JSR    PC,RAWTM          ;YES PUT DOWN TAPE MARK
4002 026210 004737 026374      JSR    PC,RASFR          ;AND SPACE FILE REV
4003 026214 005037 003456      CLR    WTMFLG          ;THEN CLEAR THE FLAG
4004 026220 020127 003740      1$:  CMP    R1,#SEQEND
4005 026224 002030      BGE    2$
4006 026226 012721 104401      MOV    #RDR,(R1)+          ;STORE READ REV CMD.
4007 026232 012721 004000      MOV    #DATCNT,(R1)+          ;SET BRJ TO MAX FOR READ RANDOM LENGTHS.
4008 026236 063737 003434      ADD    RANS,RANB          ;LET RANB := RANB + RANS
4009 026244 013702 003432      MOV    RANB,R2          ;LET R2 := RANB CLR.BY #RNOPSC
4010 026250 042702 177740      BIC    #RNOPSC,R2
4011 026254 010221      MOV    R2,(R1)+          ;SET RANDOM # OF OPERATIONS.
4012 026256 012721 000007      MOV    #RANP,(R1)+          ;RANDOM PATTERN.
4013 026262 020127 003740      CMP    R1,#SEQEND
4014 026266 002007      BGE    2$
4015 026270 012721 104001      MOV    #RDF,(R1)+          ;STORE READ FWD CMD.
4016 026274 012721 004000      MOV    #DATCNT,(R1)+          ;SET BRJ TO MAX TO READ RANDOM LENGTHS.
4017 026300 010221      MOV    R2,(R1)+          ;SET RANDOM # OF OPERATIONS.
4018 026302 012721 000007      MOV    #RANP,(R1)+          ;RANDOM PATTERN.
4019 026306 000207      2$:  RTS PC
4020
4021      :      SUBROUTINE TO SET UP A WRITE COMMAND IN THE SEQUENCE TABLE.

```

```

4022      ;      THEN A WRITE TAPE MARK AND SPACE FILE REVERSE.
4023      ;
4024      ;      INPUTS:
4025      ;      OUTPUTS:
4026      ;      REGISTERS:
4027      ;      CALLS:
4028
4029 026310 012721 104005  RANWR:: MOV      #WRT,(R1)+      ;STORE WRITE CMD.
4030 026314 004737 026422      JSR PC,RANW      ;STORE BR# , # OF OPERATIONS, PATTERN.
4031 026320 005737 003456      TST      WTMFLG      ;LAST CMD A WRT?
4032 026324 001002      BNE      1$      ;YES,RETURN
4033 026326 005237 003456      INC      WTMFLG      ;NO,SET THE FLAG
4034 026332 000207 1$:      RTS PC
4035
4036
4037      ;      SUBROUTINE TO SET UP A WRITE/VERIFY COMMAND IN THE SEQUENCE TABLE.
4038      ;      INPUTS:
4039      ;      OUTPUTS:
4040      ;      REGISTERS:
4041      ;      CALLS:
4042
4043 026334 012721 104105  RANWV:: MOV      #WTV,(R1)+      ;STORE WRITE/VERIFY CMD.
4044 026340 004737 026422      JSR PC,RANW      ;STORE BR# , # OF OPERATIONS, PATTERN.
4045 026344 000207      RTS      PC
4046
4047
4048      ;      SUBROUTINE TO SET UP A WRITE TAPE MARK IN THE SEQUENCE TABLE.
4049      ;      INPUTS:
4050      ;      OUTPUTS:
4051      ;      REGISTERS:
4052      ;      CALLS:
4053
4054 026346 020127 003740  RAWTM:: CMP      R1,#SEQEND
4055 026352 002007      BGE      1$
4056 026354 012721 100011      MOV      #WTM,(R1)+      ;STORE WRITE TAPE MARK CMD.
4057 026360 012721 000001      MOV      #1,(R1)+      ;BR#
4058 026364 012721 000001      MOV      #1,(R1)+      ;# OF OPERATIONS
4059 026370 005721      TST      (R1)+      ;SKIP PATTERNS
4060 026372 000207 1$:      RTS PC
4061
4062      ;      SUBROUTINE TO SET UP A SPACE FILE REVERSE IN THE SEQUENCE TABLE.
4063      ;      INPUTS:
4064      ;      OUTPUTS:
4065      ;      REGISTERS:
4066      ;      CALLS:
4067
4068 026374 020127 003740  RASFR:: CMP      R1,#SEQEND
4069 026400 002007      BGE      1$
4070 026402 012721 105410      MOV      #SFR,(R1)+      ;STORE SPACE FILE REVERSE
4071 026406 012721 000001      MOV      #1,(R1)+      ;BR#
4072 026412 012721 000001      MOV      #1,(R1)+      ;# OF OPERATIONS
4073 026416 005721      TST      (R1)+      ;SKIP PATTERNS
4074 026420 000207 1$:      RTS PC
4075
4076
4077      ;      SUBROUTINE TO STORE BR# , # OF OPERATIONS, PATTERN IN COMMAND
4078      ;      SEQUENCE TABLE FOR WRITE AND WRITE/VERIFY COMMANDS.

```

```

4079      :      INPUTS:
4080      :      OUTPUTS:
4081      :      REGISTERS:      R2
4082      :      CALLS:
4083
4084 026422 012721 004000      RANW:: MOV      #DATCNT,(R1)+      ;SET BRJ TO MAX FOR PATTERN GENERATION.
4085      :      :RANDOM BRJ WILL BE GENERATED FOR EACH RECORD.
4086 026426 063737 003434 003432      ADD      RANS,RANB      ;LET RANB := RANB + RANS
4087 026434 013702 003432      MOV      RANB,R2      ;LET R2 := RANB CLR.BY #RNOPSC
4088 026440 042702 177740      BIC      #RNOPSC,R2
4089 026444 010221      MOV      R2,(R1)+      ;SET RANDOM # OF OPERATIONS.
4090 026446 012721 000007      MOV      #RANP,(R1)+      ;RANDOM PATTERN.
4091 026452 000207      RTS PC      ;RETURN.
4092
4093      .EVEN
4094
4095 026454      ENDTST
4096 026454      L10034:
4097 026454 104401      TRAP      C$ETST
4098
4099      .SBTTL TEST 3: WRITE COMPATABILITY/WRITE UTILITY.
4100      :++
4101      : TEST TO WRITE RECORDS FROM BOT TO EOT.
4102      :--
4103 026456      BGNTST
4104 026456      T3::
4105 026456 112737 000001 003515      MOVVB   #1,RANDOM      ;SET THE RANDOM OPERATIONS FLAG.
4106 026464 105037 003514      CLRB     EXPBOT ;LET EXPBOT :B= #0      ;CLEAR EXPECT BOT FLAG.
4107
4108 026470 004737 017060      JSR      PC,FIRSTU      ;FIND THE FIRST UNIT.
4109 026474 004737 007072      JSR      PC,SOFINIT      ;INIT DEVICE
4110 026500 103404      BCS      11$
4111 026502      ERRDF   2,NSSRM,STAERM      ;REPORT TS05 NOT READY
4112      :      TRAP      C$ERDF
4113      :      .WORD     2
4114      :      .WORD     NSSRM
4115      :      .WORD     STAERM
4116
4117 026512 004737 007466      11$: JSR      PC,MDSET      ;GO DO SETUP'S
4118 026516 012702 004000      MOV      #DATCNT,R2      ;SET UP THE RECORD LENGTH MASK.
4119 026522 005302      DEC      R2
4120 026524 010237 003430      MOV      R2,LENMSK      ;ALLOW MAXIMUM BUFFER.
4121 026530 005137 003430      COM      LENMSK
4122 026534 004737 010162      JSR PC,SETCH      ;CMD 1 = SET CHARACTERISTIC.
4123 026540 004737 010206      JSR PC,SETRW      ;CMD2=REWIND
4124 026544 105037 003526      CLRB     STAFLG ;LET STAFLG :B= #0      ;CLEAR START FLAG
4125 026550      50426$: BREAK      ; DO A SUPVSR BREAK FIRST.
4126 026550 104422      TRAP      C$BRK
4127
4128 026552      50427$:
4129 026552 020127 003740      CMP      R1,#SEQEND      ;WHILE THERE IS MORE ROOM IN SEQ TABLE:
4130 026556 002003      BGE      50430$
4131 026560 004737 026310      JSR      PC,RANWR      ;STORE A WRITE CMD IN SEQUENCE TABLE.
4132 026564 000772      BR       50427$

```

```

4128 026566          50430$:
4129 026566 012711 177777      MOV    #END,(R1)          ;STORE END OF SEQUENCE CODE IN TABLE.
4130 026572 004737 010226      JSR    PC,EXALL          ;EXECUTE ALL CMDS IN SEQ TBL ON UNITS.
4131 026576 012701 003540      MOV    #CMDSEQ,R1          ;INIT SEQ TBL POINTER,
4132 026602 005702              TST    R2              ;REPEAT UNTIL EOT IS REACHED
4133 026604 001761              BEQ    50426$
4134 026606 105237 003524      INCB   ALLEOT          ;SET ALL UNITS @ EOT FLAG
4135 026612 000240              NOP
4136 026614 000240              NOP
4137 026616 000240              NOP
4138 026620 004737 027612      JSR    PC,T5WEOT      ;WRITE ONE RECORD BEYOND EOT ON ALL UNITS
4139                                ;SO THAT SHORTER READ STOP DISTANCE
4140                                ;SHALL POSITION HEAD IN CLEAN IRG GAP
4141                                ;READ REV THAT EXTRA REC TO RE-POSITION TAPE
4142 026624 105037 003524      CLRB   ALLEOT          ;CLEAR ALL UNITS @ EOT FLAG
4143 026630 004737 010206      JSR    PC,SETRW          ;STORE REWIND IN SEQ TBL,
4144 026634 012711 177777      MOV    #END,(R1)          ;STORE END IN SEQ TBL,
4145 026640 004737 010226      JSR    PC,EXALL          ;EXECUTE REWIND CMD ON ALL UNITS
4146
4147
4148 026644          EXIT    TST
4149 026644 104432
4150 026646 000002          TRAP    C$EXIT
4151                                .WORD    L10035-.
4152
4153                                .EVEN
4154
4155                                ENDTST
4156                                L10035:
4157                                TRAP    C$SETST
4158                                .WORD    C$SETST
4159
4160                                .SBTTL TEST 4: READ COMPATABILITY/READ UTILITY.
4161                                ;++
4162                                ;TEST TO READ ENTIRE TAPE FORWARD AND REVERSE.
4163                                ;--
4164
4165                                BGNTST
4166                                T4::
4167 026652 112737 000001 003515      MOVB   #1,RANDOM          ;SET THE RANDOM OPERATIONS FLAG.
4168 026660 112737 000001 003514      MOVB   #1,EXPBOT          ;SET EXPECT BOT FLAG.
4169 026666 004737 017060              JSR    PC,FIRSTU          ;FIND THE FIRST UNIT.
4170 026672 004737 007072              JSR    PC,SOFINIT          ;INIT DEVICE
4171 026676 103404              BCS    11$
4172 026700              ERDIF   2,NSSRM,STAERM          ;REPORT TS05 NOT READY
4173 026702 104455
4174 026704 000002          TRAP    C$ERDF
4175 026706 004536          .WORD    2
4176 026706 006120          .WORD    NSSRM
4177                                .WORD    STAERM
4178
4179 11$: JSR    PC,MDSET          ;GO DO SETUP'S
4180      JSR    PC,SETCH          ;CMD 1 = SET CHARACTERISTIC.
4181      JSR    PC,SETRW          ;CMD2=REWIND.
4182      CLRB   STAFLG          ;LET STAFLG :B= #0          ;CLEAR START FLAG
4183      MOV    #RDF,(R1)+      ;CMD3 = READ FORWARD.

```

4176	026734	012721	004000		MOV	#DATCNT,(R1)+	:SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
4177	026740	012721	077777		MOV	#77777,(R1)+	:SET RECORD COUNT TO MAX FOR WHOLE TAPE.
4178	026744	012721	000007		MOV	#RANP,(R1)+	:PATTERN = RANDOM.
4179	026750	012711	177777		MOV	#END,(R1)	:STORE END OF SEQUENCE CODE IN TABLE.
4180	026754	004737	010226		JSR	PC,EXALL	:EXECUTE ALL CMDS IN SEQ TBL ON ALL UNITS.
4181	026760	105237	003524		INCB	ALLEOT	:FLAG TO ALLOW ALL UNITS AT EOT TO READ REV
4182	026764	012701	003540		MOV	#CMDSEQ,R1	:INIT CMD SEQ TBL POINTER.
4183	026770	012721	104401		MOV	#RDR,(R1)+	:CMD1 = READ REVERSE.
4184	026774	012721	004000		MOV	#DATCNT,(R1)+	:SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
4185	027000	012721	077777		MOV	#77777,(R1)+	:RECORD COUNT = MAX FOR WHOLE TAPE.
4186	027004	012721	000007		MOV	#RANP,(R1)+	:PATTERN = RANDOM.
4187	027010	012711	177777		MOV	#END,(R1)	:STORE END OF SEQUENCE CODE IN TABLE.
4188	027014	004737	010226		JSR	PC,EXALL	:GO EXECUTE READ REV. OF ENTIRE TAPE.
4189	027020	105037	003524		CLRB	ALLEOT	:CLEAR ALL UNITS @ EOT FLAG
4190							
4191	027024				EXIT	TST	
	027024	104432					TRAP C\$EXIT
	027026	000002					.WORD L10036-
4192							
4193							
4194							
4195	027030						
	027030						
	027030	104401					TRAP C\$ETST
4196							
4197							
4198							
4199							
4200							
4201							
4202							
4203	027032						
	027032						
4204							
4205	027032	105037	003515		CLRB	RANDOM	:CLEAR RAMDOM MODE FLAG.
4206	027036	112737	000001	003514	MOVB	#1,EXPBOT	:SET EXPECT BOT FLAG.
4207							
4208	027044	004737	017060		JSR	PC,FIRSTU	:FIND THE FIRST UNIT.
4209	027050	004737	007072		JSR	PC,SOFINIT	:INIT DEVICE
4210	027054	103404			BCS	11\$	
4211	027056				ERRDF	2,NSSRM,STAERM	:REPORT TS05 NOT READY
	027056	104455					TRAP C\$ERDF
	027060	000002					.WORD 2
	027062	004536					.WORD NSSRM
	027064	006120					.WORD STAERM
4212							
4213	027066	004737	007466				
4214	027072	113737	002216	003521	11\$: JSR	PC,MDSET	:GO DO SETUP'S
4215	027100	004737	010162		MOVB	PIRE,IRE	:MOVE INHIBIT RFC ERROR REPORT FLAG.
4216	027104	013737	002220	003542	JSR	PC,SETCH	:CMD 1 = SET CHARACTERISTIC.
4217	027112	012702	002222		MOV	CHAR,CMDSEQ+2	:MOVE CHAR CODE FROM P TBL TO SEQ TBL.
4218	027116	004737	027570		MOV	#CMD2,R2	:R2 POINTS TO CMD2 IN SOFT P TABLE.
4219	027122	004737	027570		JSR	PC,PTCMDS	:MOVE CMD 2 FROM P TBL TO SEQ TBL.
4220	027126	004737	027570		JSR	PC,PTCMDS	:MOVE CMD 3 FROM P TBL TO SEQ TBL.
4221	027132	004737	027570		JSR	PC,PTCMDS	:MOVE CMD 4 FROM P TBL TO SEQ TBL.
4222	027136	004737	027570		JSR	PC,PTCMDS	:MOVE CMD 5 FROM P TBL TO SEQ TBL.
4223	027142	004737	027570		JSR	PC,PTCMDS	:MOVE CMD 6 FROM P TBL TO SEQ TBL.
					JSR	PC,PTCMDS	:MOVE CMD 7 FROM P TBL TO SEQ TBL.

4224	027146	004737	027570	JSR	PC,PTCMDS		:MOVE END CMD FROM P TBL TO SEQ TBL.
4225	027152	005037	003442	CLR	JLOOP		:CLEAR JMP CMD LOOP COUNT.
4226	027156	105037	003526	CLRB	STAFLG		:CLEAR START FLAG
4227	027162	012701	003540	MOV	#CMDSEQ,R1		:INIT SEQUENCE TABLE POINTER.
4228	027166			3\$:	:WHILE (R1) NE #END DO		:WHILE THERE ARE CMDS LEFT IN SEQUENCE TBL:
4229	027166			50431\$:			
4230	027166	021127	177777	CMP	(R1),#END		
4231	027172	001574		BEQ	50432\$		
4232	027174	022711	000040	CMP	#JMP.C,(R1)		:IS THIS A JUMP CMD?
4233	027200	001024		BNE	6\$:BR IF NOT.
4234	027202	062701	000002	ADD	#2,R1	:LET R1 := R1 + #2	:POINT TO BR.
4235	027206	012137	003444	MOV	(R1)+,JLOC		:SAVE BR (LOCATION).
4236	027212	022137	003442	CMP	(R1)+,JLOOP		:HAS LOOP COUNT BE SATISFIED?
4237	027216	001003		BNE	1\$:IF NOT, JMP AGAIN.
4238	027220	062701	000002	ADD	#2,R1		:IF SO, ADJUST SEQ POUNTER
4239	027224	000700		BR	3\$:AND GO TO NEXT COMMAND.
4240	027226	005237	003442	1\$:	INC	JLOOP	:UPDATE THE LOOP COUNT.
4241	027232	012701	003540	MOV	#CMDSEQ,R1		:INIT CMD SEQ TABLE POINTER.
4242	027236	005337	003444	2\$:	DEC	JLOC	:DECR LOCATION COUNTER.
4243	027242	001751		BEQ	3\$:IF THIS IS THE RIGHT LOCATION TO JMP TO, GO SET
4244	027244	062701	000010	ADD	#10,R1		:IF NOT, UPDATE SEQ POINTER TO NEXT CMD.
4245	027250	000772		BR	2\$:DO IT AGAIN.
4246							
4247	027252	022711	000020	6\$:	CMP	#DLY.C,(R1)	:DELAY?
4248	027256	001026		BNE	4\$:BR IF NOT.
4249	027260	062701	000004	ADD	#4,R1		:R1 = LOCATION OF N COUNT.
4250	027264	011137	003440	MOV	(R1),TIME2		:SAVE N COUNT.
4251	027270			7\$:	DELAY	1	:GO TO SUPER-WAIT 1 MSEC.
	027270	012727	000001				
	027274	000000					MOV #1,(PC)+
	027276	013727	002116				.WORD 0
	027302	000000					MOV L\$DLY,(PC)+
	027304	005367	177772				.WORD 0
	027310	001375					DEC -6(PC)
	027312	005367	177756				BNE -4
	027316	001367					DEC -22(PC)
							BNE -20
4252	027320	005337	003440	DEC	TIME2		
4253	027324	001361		BNE	7\$		
4254	027326	062701	000004	ADD	#4,R1	:LET R1 := R1 + #4	:POINT TO NEXT CMD.
4255	027332	000715		BR	3\$:GO CHECK NEXT CMD.
4256	027334	004737	011172	4\$:	JSR	PC,SETUP	:GO SETUP THE COMMAND BLOCK.
4257	027340			50433\$:	:WHILE NCNT LT NCNT1 DO		:WHILE THERE ARE RECORDS REMAINING:
4258	027340	023737	003412 003414	CMP	NCNT,NCNT1		
4259	027346	002103		BGE	50434\$		
4260	027350	004737	011064	JSR	PC,CMDAC		:STORE CMD ASCII IN ERROR MSG.
4261	027354	004737	010524	JSR	PC,EXSUB		:ISSUE CMD TO ALL,AWAIT INTS,CHECK STATUS.
4262	027360	023727	003420 100017	CMP	CMDWRD,#GES		:IF CMD IS GET STATUS THEN:
4263	027366	001002		BNE	50435\$		
4264	027370	004737	017370	JSR	PC,PRXST		:PRINT EXTENDED STATUS REGISTERS.
4265							
4266	027374			50435\$:			
4267	027374	004737	017456	JSR	PC,CKHAE		:CHECK HALT AFTER EACH CMD FLAG.
4268	027400	012702	000001	MOV	#1,R2		:SET ALL UNITS AT BOT/EOT.
4269	027404	004737	017060	JSR	PC,FIRSTU		:FIND FIRST UNIT.
4270	027410			50436\$:	:WHILE DEVTBL(R5) NE #END DO		:WHILE THERE ARE MORE UNITS:
4271	027410	026527	002604 177777	CMP	DEVTBL(R5),#END		
4272	027416	001426		BEQ	50437\$		

HARDWARE TESTS MACRO M1113 25-MAY-82 09:51 PAGE 20-87
TEST 5: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

SEQ 0133

```

4273 027420 032737 000400 003420      BIT      #MOD.CO,CMDWRD      ;IF CMD IS REVERSE THEN:
4274 027426 001406                      BEQ      50440$
4275 027430 032765 000002 003502      BIT      #X0.BOT,EOTFLG(R5)      ;IF NOT AT BOT THEN:
4276 027436 001001                      BNE      50441$
4277 027440 005002                      CLR      R2      ;CLEAR EOT/BOT FLAG.
4278
4279 027442                      50441$:
4280 027442 000411                      BR      50442$      ;ELSE IF CMD IS NOT REVERSE:
4281 027444                      50440$:
4282 027444 032765 000001 003502      BIT      #X0.EOT,EOTFLG(R5)
4283 027452 001404                      BEQ      50443$
4284 027454 032737 000001 003420      BIT      #CMD.CO,CMDWRD
4285 027462 001001                      BNE      50444$
4286 027464                      50443$:
4287
4288 027464 005002                      CLR      R2      ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
4289
4290                      50444$:      ;LET R2 := #0      ;CLEAR EOT/BOT FLAG.
4291
4292                      50442$:
4293 027466 004737 017126                      JSR PC,NEXTU      ;FIND NEXT UN'T
4294
4295 027472 000746                      BR      50436$      ;
4296 027474                      50437$:
4297 027474 020227 000001      CMP      R2,#1      ;IF ALL UNIT ARE AT EOT/BOT THEN:
4298 027500 001016                      BNE      50445$
4299 027502 013737 003412 003414      MOV      NCNT,NCNT1      ;FORCE TERMINATION OF COMMAND.
4300 027510 005237 003414                      INC      NCNT1
4301 027514 105237 003524                      INCB     ALLEOT      ;FLAG ALL UNITS AT EOT/BOT TO ALLOW VERIFY OF D
4302 027520 023727 003426 000002      CMP      CMDLG,#2      ;WHEN WRITING IS CURRENT COMMAND
4303 027526 001002                      BNE      50446$
4304 027530 004737 027612                      JSR PC,T5WEOT      ;GO WRITE/READ REV ONE RECORD BEYOND EOT
4305
4306 027534                      50446$:
4307
4308 027534 000402                      BR      50447$
4309 027536                      50445$:
4310 027536 105037 003524      CLRB     ALLEOT      ;WHEN NOT ALL @EOT, CLEAR FLAG
4311
4312                      50447$:
4313 027542 005237 003412                      INC      NCNT      ;UPDATE RECORD COUNT.
4314 027546 013737 003420 003424      MOV      CMDWRD,PCMDWD      ;SAVE PREVIOUS COMMAND WORD.
4315
4316 027554 000671                      BR      50433$
4317 027556                      50434$:
4318 027556 004737 016044                      JSR PC,VFYDAT      ;IF LAST CMD WAS A WRITE VERIFY, THEN GO
4319
4320                      ;VERIFY THE LAST N RECORDS OF DATA.
4321 027562 000601                      BR      50431$
4322 027564                      50432$:
4323
4324 027564                      EXIT     TST
4325 027564 104432                      TRAP     C$EXIT
4326 027566 000140                      .WORD     L10037-.
4327
;      SUBROUTINE TO MOVE A COMMAND FROM THE SOFTWARE P TABLE TO
;      THE COMMAND SEQUENCE TABLE.

```

```

4328      :      INPUTS:      R2 = POINTER TO SOFT 'P' TABLE
4329      :      OUTPUTS:
4330      :      REGISTERS:    R3.
4331      :      CALLS:
4332
4333 027570 012203      PTCMDS: MOV      (R2)+,R3      ;R3 = COMMAND TABLE INDEX.
4334 027572 005303      DEC      R3
4335 027574 006303      ASL      R3
4336 027576 016321 003752  MOV      CMDTBL(R3),(R1)+      ;MOVE COMMAND WORD.
4337 027602 012221      MOV      (R2)+,(R1)+      ;MOVE # OF BYTES.
4338 027604 012221      MOV      (R2)+,(R1)+      ;MOVE # OF OPERATIONS.
4339 027606 012221      MOV      (R2)+,(R1)+      ;MOVE PATTERN CODE.
4340 027610 000207      RTS PC
4341
4342      :      SUBROUTINE TO WRITE THEN READ REVERSE ONE RECORD BEYOND EOT
4343      :      INPUTS:
4344      :      OUTPUTS:
4345      :      REGISTERS:
4346      :      CALLS:      CMDAC,EXSUB,CKHAE
4347
4348 027612 000240      TSWEOT: NOP
4349 027614 000240      NOP
4350 027616 004737 010524  JSR PC,EXSUB      ;WRITE ONE RECORD BEYOND EOT
4351 027622 004737 017456  JSR PC,CKHAE      ;SO THAT READ SHORTER STOP DISTANCE
4352
4353 027626 012700 000002      MOV      #2,R0      ;SHALL POSITION HEAD IN CLEAN IRG GAP
4354 027632 013737 003420 003424 1$: MOV      CMDWRD,PCMDWD      ;SET UP COUNTER FOR EOT
4355 027640 012737 104401 003420  MOV      #RDR,CMDWRD      ;LET PCMDWD := CMDWRD ;REPOSITION TAPE
4356 027646 012737 000004 003426  MOV      #4,CMDLG      ;LET CMDWRD := #RDR ;BEFORE EXTRA RECORD
4357 027654 013737 003420 002330  MOV      CMDWRD,CMDPKT      ;BY READING REVERSE
4358 027662 042737 004000 002330  BIC      #BRF.C,CMDPKT      ;LET CMDPKT := CMDWRD CLR.BY #BRF.C
4359 027670 013737 002330 003422  MOV      CMDPKT,CMDSAV      ;LET CMDSAV := CMDPKT ;THAT RECORD TO ALLOW
4360 027676 013737 003410 002332  MOV      DATARD,CMDPKT+CP.ADL      ;NEXT COMMAND IN iHE
4361 027704 004737 011064      JSR PC,CMDAC      ;TABLE TO BE EXECUTED
4362 027710 004737 010524      JSR PC,EXSUB
4363 027714 004737 017456      JSR PC,CKHAE
4364 027720 005300      DEC      R0      ;FOUND EOT YET?
4365 027722 001343      BNE      1$      ;NO,KEEP GOING
4366 027724 000207      RTS PC      ;YES,RETURN
4367
4368      .EVEN
4369
4370 027726      ENDTST
4371 027726 104401      L10037:
4372 027726
4373
4374      TRAP      C$ETST
4375
4376      ENDMOD
4377
4378 027730      .TITLE PARAMETER CODING
4379
4380      .SBTTL HARDWARE PARAMETER CODING SECTION
4381
4382      BGNMOD
4383
4384      ;++
4385      ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
4386      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE

```

```

4383      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
4384      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
4385      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
4386      ; WITH THE OPERATOR.
4387      ;--
4388
4389 027730      BGNHRD
4389 027730      000042
4389 027732      L$HARD::
4390
4391 027732      GPRMA    TS5ADR,0,0,160010,177564,YES
4391 027732      000031
4391 027734      027770
4391 027736      160010
4391 027740      177564
4392 027742      GPRMD    TS5VCT,2,0,777,60,776,YES
4392 027742      001032
4392 027744      030005
4392 027746      000777
4392 027750      000060
4392 027752      000776
4393 027754      GPRMD    TS5UNT,4,0,1,0,1,NO
4393 027754      002022
4393 027756      030014
4393 027760      000001
4393 027762      000000
4393 027764      000001
4394 027766      EXIT HRD
4394 027766      024004
4395
4396
4397 027770      124      123      104      TS5ADR: .NLIST  BEX
4398 030005      126      105      103      TS5VCT: .ASCIZ  /TSDB ADDRESS/
4399 030014      123      105      114      TS5UNT: .ASCIZ  /VECTOR/
4400
4401
4402
4403 030036      .LIST  BEX
4403
4403 030036      .EVEN
4403
4403      ENDHRD
4404
4404      L10040:
4405
4405      .SBTTL  SOFTWARE PARAMETER CODING SECTION
4406
4407      ;++
4408      ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
4409      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES  THE
4410      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
4411      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
4412      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
4413      ; WITH THE OPERATOR.
4414      ;--
4415
4416 030036      BGNSFT
4416 030036      000302
4416 030040      L$SOFT::
4417 030040      GPRML    CLRM,0,1,YES
4417 030040      000130

```

.WORD L10040-L\$HARD/2

.WORD T\$CODE
 .WORD TS5ADR
 .WORD T\$LOLIM
 .WORD T\$HILIM

.WORD T\$CODE
 .WORD TS5VCT
 .WORD 777
 .WORD T\$LOLIM
 .WORD T\$HILIM

.WORD T\$CODE
 .WORD TS5UNT
 .WORD 1
 .WORD T\$LOLIM
 .WORD T\$HILIM

.WORD T\$CODE

.EVEN

.WORD L10041-L\$SOFT/2

.WORD T\$CODE

	030042	030644			.WORD	CLRM
	030044	000001			.WORD	1
4418	030046		GPRML	RRVM,0,400,YES		
	030046	000130			.WORD	T\$CODE
	030050	030663			.WORD	RRVM
	030052	000400			.WORD	400
4419	030054		GPRML	RCVERM,2,400,YES		
	030054	001130			.WORD	T\$CODE
	030056	030766			.WORD	RCVERM
	030060	000400			.WORD	400
4420	030062		GPRML	HAEM,2,1,YES		
	030062	001130			.WORD	T\$CODE
	030064	030712			.WORD	HAEM
	030066	000001			.WORD	1
4421	030070		GPRML	IREFM,6,400,YES		
	030070	003130			.WORD	T\$CODE
	030072	031042			.WORD	IREFM
	030074	000400			.WORD	400
4422	030076		XFERT	NEXTSP		
	030076	004024			.WORD	T\$CODE
4423	030100		GPRML	BADTM,4,1,YES		
	030100	002130			.WORD	T\$CODE
	030102	030736			.WORD	BADTM
	030104	000001			.WORD	1
4424	030106		NEXTSP: GPRML	DINTM,6,1,YES		
	030106	003130			.WORD	T\$CODE
	030110	031017			.WORD	DINTM
	030112	000001			.WORD	1
4425	030114		GPRML	IREFM,12,1,YES		
	030114	005130			.WORD	T\$CODE
	030116	031107			.WORD	IREFM
	030120	000001			.WORD	1
4426	030122		GPRML	CHGM,10,1,YES		
	030122	004130			.WORD	T\$CODE
	030124	031063			.WORD	CHGM
	030126	000001			.WORD	1
4427	030130		XFERF	ENDSP1		
	030130	127044			.WORD	T\$CODE
4428	030132		GPRMD	CHARM,14,0,377,0,777,YES		
	030132	006032			.WORD	T\$CODE
	030134	031140			.WORD	CHARM
	030136	000377			.WORD	377
	030140	000000			.WORD	T\$LOLIM
	030142	000777			.WORD	T\$HILIM
4429	030144		GPRMD	CMD2M,16,D,37,1,33,YES		
	030144	007052			.WORD	T\$CODE
	030146	031165			.WORD	CMD2M
	030150	000037			.WORD	37
	030152	000001			.WORD	T\$LOLIM
	030154	000033			.WORD	T\$HILIM
4430	030156		GPRMD	BPCRM,20,D,-1,1,DATCNT,YES		
	030156	010052			.WORD	T\$CODE
	030160	031173			.WORD	BPCRM
	030162	177777			.WORD	-1
	030164	000001			.WORD	T\$LOLIM
	030166	004000			.WORD	T\$HILIM
4431	030170		GPRMD	NUMBM,22,D,-1,1,77777,YES		

	030170	011052				.WORD	T\$CODE
	030172	031205				.WORD	NUMBM
	030174	177777				.WORD	-1
	030176	000001				.WORD	T\$LOLIM
	030200	077777				.WORD	T\$HILIM
4432	030202		GPRMD	PATTM,24,D,17,0,10,YES			
	030202	012052				.WORD	T\$CODE
	030204	031225				.WORD	PATTM
	030206	000017				.WORD	17
	030210	000000				.WORD	T\$LOLIM
	030212	000010				.WORD	T\$HILIM
4433	030214		GPRMD	CMD3M,26,D,37,1,33,YES			
	030214	013052				.WORD	T\$CODE
	030216	031334				.WORD	CMD3M
	030220	000037				.WORD	37
	030222	000001				.WORD	T\$LOLIM
	030224	000033				.WORD	T\$HILIM
4434	030226		GPRMD	BPCRM,30,D,-1,1,DATCNT,YES			
	030226	014052				.WORD	T\$CODE
	030230	031173				.WORD	BPCRM
	030232	177777				.WORD	-1
	030234	000001				.WORD	T\$LOLIM
	030236	004000				.WORD	T\$HILIM
4435	030240		GPRMD	NUMBM,32,D,-1,1,77777,YES			
	030240	015052				.WORD	T\$CODE
	030242	031205				.WORD	NUMBM
	030244	177777				.WORD	-1
	030246	000001				.WORD	T\$LOLIM
	030250	077777				.WORD	T\$HILIM
4436	030252		GPRMD	PATTM,34,D,17,0,10,YES			
	030252	016052				.WORD	T\$CODE
	030254	031225				.WORD	PATTM
	030256	000017				.WORD	17
	030260	000000				.WORD	T\$LOLIM
	030262	000010				.WORD	T\$HILIM
4437	030264		GPRMD	CMD4M,36,D,37,1,33,YES			
	030264	017052				.WORD	T\$CODE
	030266	031342				.WORD	CMD4M
	030270	000037				.WORD	37
	030272	000001				.WORD	T\$LOLIM
	030274	000033				.WORD	T\$HILIM
4438	030276		GPRMD	BPCRM,40,D,-1,1,DATCNT,YES			
	030276	020052				.WORD	T\$CODE
	030300	031173				.WORD	BPCRM
	030302	177777				.WORD	-1
	030304	000001				.WORD	T\$LOLIM
	030306	004000				.WORD	T\$HILIM
4439	030310		GPRMD	NUMBM,42,D,-1,1,77777,YES			
	030310	021052				.WORD	T\$CODE
	030312	031205				.WORD	NUMBM
	030314	177777				.WORD	-1
	030316	000001				.WORD	T\$LOLIM
	030320	077777				.WORD	T\$HILIM
4440	030322		GPRMD	PATTM,44,D,17,0,10,YES			
	030322	022052				.WORD	T\$CODE
	030324	031225				.WORD	PATTM
	030326	000017				.WORD	17

	030330	000000				.WORD	T\$LOLIM
	030332	000010				.WORD	T\$HILIM
4441	030334		GPRMD	CMD5M,46,D,37,1,33,YES			
	030334	023052				.WORD	T\$CODE
	030336	031350				.WORD	CMD5M
	030340	000037				.WORD	37
	030342	000001				.WORD	T\$LOLIM
	030344	000033				.WORD	T\$HILIM
4442	030346		GPRMD	BPCRM,50,D,-1,1,DATCNT,YES			
	030346	024052				.WORD	T\$CODE
	030350	031173				.WORD	BPCRM
	030352	177777				.WORD	-1
	030354	000001				.WORD	T\$LOLIM
	030356	004000				.WORD	T\$HILIM
4443	030360		GPRMD	NUMBM,52,D,-1,1,77777,YES			
	030360	025052				.WORD	T\$CODE
	030362	031205				.WORD	NUMBM
	030364	177777				.WORD	-1
	030366	000001				.WORD	T\$LOLIM
	030370	077777				.WORD	T\$HILIM
4444	030372		GPRMD	PATTM,54,D,17,0,10,YES			
	030372	026052				.WORD	T\$CODE
	030374	031225				.WORD	PATTM
	030376	000017				.WORD	17
	030400	000000				.WORD	T\$LOLIM
	030402	000010				.WORD	T\$HILIM
4445	030404		XFER	ENDSP2			
	030404	002004				.WORD	T\$CODE
4446	030406		ENDSP1: XFER	ENDSP3			
	030406	076004				.WORD	T\$CODE
4447	030410		ENDSP2: GPRMD	CMD6M,56,D,37,1,33,YES			
	030410	027052				.WORD	T\$CODE
	030412	031356				.WORD	CMD6M
	030414	000037				.WORD	37
	030416	000001				.WORD	T\$LOLIM
	030420	000033				.WORD	T\$HILIM
4448	030422		GPRMD	BPCRM,60,D,-1,1,DATCNT,YES			
	030422	030052				.WORD	T\$CODE
	030424	031173				.WORD	BPCRM
	030426	177777				.WORD	-1
	030430	000001				.WORD	T\$LOLIM
	030432	004000				.WORD	T\$HILIM
4449	030434		GPRMD	NUMBM,62,D,-1,1,77777,YES			
	030434	031052				.WORD	T\$CODE
	030436	031205				.WORD	NUMBM
	030440	177777				.WORD	-1
	030442	000001				.WORD	T\$LOLIM
	030444	077777				.WORD	T\$HILIM
4450	030446		GPRMD	PATTM,64,D,17,0,10,YES			
	030446	032052				.WORD	T\$CODE
	030450	031225				.WORD	PATTM
	030452	000017				.WORD	17
	030454	000000				.WORD	T\$LOLIM
	030456	000010				.WORD	T\$HILIM
4451	030460		GPRMD	CMD7M,66,D,37,1,33,YES			
	030460	033052				.WORD	T\$CODE
	030462	031364				.WORD	CMD7M

	030464	000037			.WORD	37
	030466	000001			.WORD	TSLOLIM
	030470	000033			.WORD	TSHILIM
4452	030472		GPRMD	BPCRM,70,D,-1,1,DATCNT,YES		
	030472	034052			.WORD	TSCODE
	030474	031173			.WORD	BPCRM
	030476	177777			.WORD	-1
	030500	000001			.WORD	TSLOLIM
	030502	004000			.WORD	TSHILIM
4453	030504		GPRMD	NUMBM,72,D,-1,1,77777,YES		
	030504	035052			.WORD	TSCODE
	030506	031205			.WORD	NUMBM
	030510	177777			.WORD	-1
	030512	000001			.WORD	TSLOLIM
	030514	077777			.WORD	TSHILIM
4454	030516		GPRMD	PATTM,74,D,17,0,10,YES		
	030516	036052			.WORD	TSCODE
	030520	031225			.WORD	PATTM
	030522	000017			.WORD	17
	030524	000000			.WORD	TSLOLIM
	030526	000010			.WORD	TSHILIM
4455	030530		GPRMD	CMD8M,76,D,37,1,33,YES		
	030530	037052			.WORD	TSCODE
	030532	031372			.WORD	CMD8M
	030534	000037			.WORD	37
	030536	000001			.WORD	TSLOLIM
	030540	000033			.WORD	TSHILIM
4456	030542		GPRMD	BPCRM,100,D,-1,1,DATCNT,YES		
	030542	040052			.WORD	TSCODE
	030544	031173			.WORD	BPCRM
	030546	177777			.WORD	-1
	030550	000001			.WORD	TSLOLIM
	030552	004000			.WORD	TSHILIM
4457	030554		GPRMD	NUMBM,102,D,-1,1,77777,YES		
	030554	041052			.WORD	TSCODE
	030556	031205			.WORD	NUMBM
	030560	177777			.WORD	-1
	030562	000001			.WORD	TSLOLIM
	030564	077777			.WORD	TSHILIM
4458	030566		GPRMD	PATTM,104,D,17,0,10,YES		
	030566	042052			.WORD	TSCODE
	030570	031225			.WORD	PATTM
	030572	000017			.WORD	17
	030574	000000			.WORD	TSLOLIM
	030576	000010			.WORD	TSHILIM
4459	030600		XFER	ENDSP		
	030600	022004			.WORD	TSCODE
4460	030602		ENDSP3: GPRML	TSMD,106,1,YES		
	030602	043130			.WORD	TSCODE
	030604	031235			.WORD	TSMD
	030606	000001			.WORD	1
4461	030610		XFERT	ENDSP		
	030610	016024			.WORD	TSCODE
4462	030612		GPRML	FAST,114,1,YES		
	030612	046130			.WORD	TSCODE
	030614	031325			.WORD	FAST
	030616	000001			.WORD	1

4463	030620				XFERT	ENDSP4			
	030620	011024						.WORD	T\$CODE
4464	030622				GPRML	WTBF,112,1,YES			
	030622	045130						.WORD	T\$CODE
	030624	031305						.WORD	WTBF
	030626	000001						.WORD	1
4465	030630				XFERT	ENDSP			
	030630	006024						.WORD	T\$CODE
4466	030632				GPRML	RDBF,110,1,YES			
	030632	044130						.WORD	T\$CODE
	030634	031266						.WORD	RDBF
	030636	000001						.WORD	1
4467	030640				ENDSP5: XFER	ENDSP			
	030640	002004						.WORD	T\$CODE
4468	030642				ENDSP4: XFER	ENDSP			
	030642	001004						.WORD	T\$CODE
4469	030644				ENDSP:				
4470	030644				ENDSFT				
	030644							.EVEN	
4471					L10041:				
4472					.EVEN				
4473					.NLIST	BEX			
4474	030644	103	114	105	CLRM:	.ASCIZ /CLEAR COUNTERS/			
4475	030663	122	105	123	RRVM:	.ASCIZ /RESET RANDOM VARIABLES/			
4476	030712	110	101	114	HAEM:	.ASCIZ /HALT AFTER EACH CMD/			
4477	030736	102	101	104	BADTM:	.ASCIZ /BAD TAPE SPOT DETECTION/			
4478	030766	120	122	111	RCVERM:	.ASCIZ /PRINT RECOVERABLE ERRORS/			
4479	031017	104	111	123	DINTM:	.ASCIZ /DISABLE INTERRUPTS/			
4480	031042	111	116	110	IREFM:	.ASCIZ /INHIBIT RECOVERY/			
4481	031063	103	110	101	CHGM:	.ASCIZ /CHANGE CMD SEQUENCE/			
4482	031107	111	116	110	IREFM:	.ASCIZ /INHIBIT RFC ERROR REPORT/			
4483	031140	103	110	101	CHARM:	.ASCIZ /CHARACTERISTICS CODE/			
4484	031165	103	115	104	CMD2M:	.ASCIZ "CMD/2"			
4485	031173	102	122	106	BPCRM:	.ASCIZ /BRF COUNT/			
4486	031205	043	040	117	NUMBM:	.ASCIZ /# OF OPERATIONS/			
4487	031225	120	101	124	PATM:	.ASCIZ /PATTERN/			
4488	031235	104	105	106	TSMD:	.ASCIZ /DEFAULT SWITCH SETTINGS?/			
4489	031266	122	105	101	RDBF:	.ASCIZ /READ BUFFERING/			
4490	031305	127	122	111	WTBF:	.ASCIZ /WRITE BUFFERING/			
4491	031325	061	060	060	FAST:	.ASCIZ /100IPS/			
4492					.LIST	BEX			
4493					.EVEN				
4494									
4495					.NLIST	BEX			
4496	031334	103	115	104	CMD3M:	.ASCIZ "CMD/3"			
4497	031342	103	115	104	CMD4M:	.ASCIZ "CMD/4"			
4498	031350	103	115	104	CMD5M:	.ASCIZ "CMD/5"			
4499	031356	103	115	104	CMD6M:	.ASCIZ "CMD/6"			
4500	031364	103	115	104	CMD7M:	.ASCIZ "CMD/7"			
4501	031372	103	115	104	CMD8M:	.ASCIZ "CMD/8"			
4502					.LIST	BEX			
4503					.EVEN				
4504									
4505									
4506									
4507									

```

4508
4509
4510
4511
4512
4513
4514 031400
4515
4517      032000
4519
4520 032000
      032000 000000
      032002 000000
      032004
4521 032004
4522
4523
4524
4525
4526
4527
4528
4529 032004
4530 032004
      032004 000000
      032006 000003
      032010
4531 032010 172522
4532 032012 000224
4533 032014 000000
4534 032016
      032016
4535 032016
4536
4537      000001
  
```

```

*****
:      PATCH AREA
:      AND AN ADJUSTMENT TO ACCOUNT FOR THE 'LASTAD BIT7' HACK
:      DESCRIBED IN 'SUPPRG.MEM' (FOR REV C).
:
PATCH:: .BLKW   64.
      .=.!377+1
      LASTAD
      .EVEN
      .WORD      0
      .WORD      0
L$LAST::
      ENDMOD
.SBTTL  HARD CODED P-TBL
:++
:DIAG IS PRE-PARAMETERIZED PER TBL
:--
      BGNSETUP 1
      BGNPTAB
      .WORD      0
      .WORD      L10044-./2-1
L10042:
      172522
      224
      0
      ENDPTAB
L10044:
      ENDSETUP
.END
  
```

PARAMETER CODING
SYMBOL TABLE

MACRO M1113 25-MAY-82 09:51 PAGE 20-96

M 1

SEQ 0142

ACK.C = 100000 G	BTADDR 002616 G	CP.CNT= 000006 G	CSTPRI= 000013	EXPBOT 003514 G
ADR = 000020 G	BTMSG1 015070	CRLF 005741 G	DATARD 003410 G	EXSUB 010524 G
ALLEOT 003524 G	BTMSG2 015155	CRLFSP 005744 G	DATAWT 003406 G	EXTFEA 002322 G
ASSEMB= 000010	BTMSG3 015225	CTCC 003450 G	DATCNT= 004000 G	ESEND = 002100
ATTNM 004603 G	BTPT 003512 G	CVC.C = 040000 G	DATERM 005752 G	ESLOAD= 003035
AUDRPM 005114 G	BTRPT 020142 G	CSAU = 000052	DEVTBL 002604 G	FAST 031325
AUDRUN 005146 G	BT0 003046 G	CSAUTO= 000061	DFPTBL 002174 G	FATSM 004642 G
AUTODM 023562	BT1 003120 G	CSBRK = 000022	DFTSCH= 000040 G	FIRSTU 017060 G
BADTM 030736	BT2 003172 G	CSBSEG= 000004	DIA = 100006 G	FMT.CO= 000040 G
BADTSW 002210 G	BT3 003244 G	CSBSUB= 000002	DIABLK= 003406 G	FMT.C1= 000100 G
BENBSW 002324 G	CHAR 002220 G	CSCEFG= 000045	DIACNT= 000020 G	FTLCNT 003366 G
BFSEQ 025026 G	CHARM 031140	CSCLCK= 000062	DIAGMC= 000000	FUNRM 004622 G
BFSEQ0 025052	CHGFLG 002214 G	CSCLEA= 000012	DINT 002212 G	FSAU = 000015
BFSEQ1 025124	CHGM 031063	CSCLOSE= 000035	DINTM 031017	FSAUTO= 000020
BFSEQ2 025136	CHKERR 013166 G	CSCLP1= 000006	DLY = 000020 G	FSBGN = 000040
BFSEQ3 025230	CH.EAI= 000040 G	CSCEVC= 000036	DLY.C = 000020 G	FSCLEA= 000007
BFSEQ4 025302	CH.ERI= 000020 G	CSDECLN= 000044	DRI = 100013 G	FSDU = 000016
BFSEQ5 025344	CH.ESS= 000200 G	CSDDODU= 000051	DROPDN 005065 G	FSEND = 000041
BFSEQ6 025416	CKDATA 016444 G	CSDRPT= 000024	DROPEP 003522 G	FSHARD= 000004
BFSEQ7 025470	CKDCNT 017054	CSDU = 000053	DROPN 017366	FSHW = 000013
BFSEQ8 025522	CKDFF 017056	CSEDIT= 000003	DROPU 017156 G	FSINIT= 000006
BFSEQ9 025554	CKHAE 017456 G	CSERDF= 000055	DROPUA 017302	FSJMP = 000050
BFSE10 025576	CKHRTN 017544	CSERHR= 000056	DRORTN 017360	FSMOD = 000000
BGNFLG= 003460	CLN = 101012 G	CSERRO= 000060	DTAERM 005752 G	FSMSG = 000011
BINC 016030	CLRERR 012664 G	CSERSF= 000054	DTAER2 005226 G	FSPROT= 000021
BIT0 = 000001 G	CLRFLG 002204 G	CSERSO= 000057	DTAER3 005275 G	FSPWR = 000017
BIT00 = 000001 G	CLRM 030644	CSESCA= 000010	DTAER4 005337 G	FSRPT = 000012
BIT01 = 000002 G	CMDAC 011064 G	CSSEEG= 000005	DTAER5 005360 G	FSSEG = 000003
BIT02 = 000004 G	CMDASC 004040 G	CSesub= 000003	EF.CON= 000036 G	FSOFT= 000005
BIT03 = 000010 G	CMDD 002222 G	CSETST= 000001	EF.HSS= 000040 G	FSRV = 000010
BIT04 = 000020 G	CMDLG 003426 G	CSEXIT= 000032	EF.NEW= 000035 G	FSUB = 000002
BIT05 = 000040 G	CMDPKM 004346 G	C\$GETB= 000026	EF.PWR= 000034 G	FSW = 000014
BIT06 = 000100 G	CMDPKT 002330 G	C\$GETW= 000027	EF.RBO= 000020 G	FSTESi= 000001
BIT07 = 000200 G	CMDSAV 003422 G	C\$GMAN= 000043	EF.RES= 000037 G	GCMDA 011136 G
BIT08 = 000400 G	CMDSEQ 003540 G	C\$GPHR= 000042	EF.RWB= 000030 G	GENPAT 011556 G
BIT09 = 001000 G	CMDSE2 003550 G	C\$GPLO= 000030	EF.STA= 000040 G	GES = 100017 G
BIT1 = 000002 G	CMDTBL 003752 G	C\$GPRI= 000040	EINC 016036	GETSTM 005507 G
BIT10 = 002000 G	CMDWRD 003420 G	C\$INIT= 000011	END = 177777 G	GIT 012050
BIT11 = 004000 G	CMD.CO= 000001 G	C\$INLP= 000020	ENDERF= 003472	GOWAIT 012364 G
BIT12 = 010000 G	CMD.C1= 000002 G	C\$MANI= 000050	ENDFLG= 003526	GSCPCK 002340 G
BIT13 = 020000 G	CMD.C2= 000004 G	C\$MEM = 000031	ENDSP 030644	G\$CNT0= 000200
BIT14 = 040000 G	CMD.C3= 000010 G	C\$MSG = 000023	ENDSP1 030406	G\$DELM= 000372
BIT15 = 100000 G	CMD.C4= 000020 G	C\$OPEN= 000034	ENDSP2 030410	G\$DISP= 000003
BIT2 = 000004 G	CMD2M 031165	C\$PNTB= 000014	ENDSP3 030602	G\$EXCP= 000400
BIT3 = 000010 G	CMD3M 031334	C\$PNTF= 000017	ENDSP4 030642	G\$HILI= 000002
BIT4 = 000020 G	CMD4M 031342	C\$PNTS= 000016	ENDSP5 030640	G\$LOLI= 000001
BIT5 = 000040 G	CMD5M 031350	C\$PNTX= 000015	EOTFLG 003502 G	G\$NO = 000000
BIT6 = 000100 G	CMD6M 031356	C\$QIO = 000377	ERCVER 002207 G	G\$OFFS= 000400
BIT7 = 000200 G	CMD7M 031364	C\$RDBU= 000007	ERLOG 003466 G	G\$OF SI= 000376
BIT8 = 000400 G	CMD8M 031372	C\$REFG= 000047	ERRREC 003471 G	G\$PRMA= 000001
BIT9 = 001000 G	CNTBGN= 002626	C\$RESE= 000033	ERS = 100411 G	G\$PRMD= 000002
BOE = 000400 G	CNTEND= 003376	C\$REVI= 000003	ERSFLG 003525 G	G\$PRML= 000000
BORERS 015274 G	CNTLEN= 000550 G	C\$RFLA= 000021	EVL = 000004 G	G\$RADA= 000140
BPCRM 031173	CODELM 004162 G	C\$RPT = 000025	EXALL 010226 G	G\$RADB= 000000
BRCPK 002344 G	CP.ADH= 000004 G	C\$SEFG= 000046	EXARTN 010522	G\$RADD= 000040
BRFCNT 003416 G	CP.ADL= 000002 G	C\$SPRI= 000041	EXCRTN 012362	G\$RADL= 000120
BRF.C = 004000 G	CP.CMD= 000000 G	C\$SVEC= 000037	EXCUTE 012054 G	G\$RADO= 000020

PARAMETER CODING
SYMBOL TABLE

MACRO M1113 25-MAY-82 09:51 PAGE 20-97

N 11

SEQ 0143

GSXFER= 000004	L\$CCP 002106 G	L10012 023270	NSSRM 004536 G	RAWTM 026346 G
GSYES = 000010	L\$CLEA 023714 G	L10013 023560	NUMBM 031205	RCVERM 030766
HAEM 002206 G	L\$CO 002032 G	L10014 023754	NURTY1 005422 G	RDBF 031266
HALTM 004306 G	L\$DEPO 002011 G	L10015 024026	OFLINM 005456 G	RDBUF 002314 G
HELP = 000000	L\$DESC 002136 G	L10016 024132	ONEFIL= 000001	RDF = 104001 G
HOE = 100000 G	L\$DESP 002076 G	L10017 025620	OPFLAG 003536 G	RDR = 104401 G
HRDCNT 003356 G	L\$DEVP 002060 G	L10020 024346	OPP.C = 020000 G	RECCNT 003376 G
HSSW 002320 G	L\$DISP 002124 G	L10021 024372	OSAPTS= 000000	RECLOG 003465 G
IBE = 010000 G	L\$DLY 002116 G	L10022 024412	OSAU = 000001	RECRED 007066
IDU = 000040 G	L\$DTP 002040 G	L10023 024432	OSBGNR= 000001	RECTAP 010064 G
IER = 020000 G	L\$DTYP 002034 G	L10024 024452	OSBGNS= 000001	RECUD 013020 G
IE.C = 000200 G	L\$DU 023756 G	L10025 024472	OSDU = 000001	RERM 005017 G
INIT10 021302	L\$DUT 002072 G	L10026 024512	OSERRT= 000000	RETRYC 003460 G
INIT15 021600	L\$DVTY 002164 G	L10027 024532	OSGNSW= 000001	REWRT 015450 G
INIT16 021620	L\$EF 002052 G	L10030 024552	OSPOIN= 000001	RFBC 002726 G
INTFLG 003472 G	L\$ENVI 002044 G	L10031 024572	OSSETU= 000000	RFCERM 004521 G
INTPRI= 000340 G	L\$ETP 002102 G	L10032 024630	PASCNT 003326 G	RFREC 003026 G
INVRT 007724 G	L\$EXP1 002046 G	L10033 025014	PATCH 031400 G	RFUNR 003036 G
IRE 003521 G	L\$EXP4 002064 G	L10034 026454	PATERN 003446 G	RLEXM 004556 G
IREC 002213 G	L\$EXP5 002066 G	L10035 026650	PATRO 011642 G	RNF = 125401 G
IRECM 031042	L\$HARD 027732 G	L10036 027030	PATR1 011700 G	RNOPSC= 177740 G
IREM 031107	L\$HIME 002120 G	L10037 027726	PATR2 011720 G	RNR = 105401 G
ISR = 000100 G	L\$HPCP 002016 G	L10040 030036	PATR3 011730 G	RNYM 004753 G
IXE = 004000 G	L\$HPTP 002022 G	L10041 030644	PATR4 011754 G	RPF = 105001 G
ISAU = 000041	L\$HW 002174 G	L10042 032010	PATR5 011766 G	RPR = 125001 G
ISAUTO= 000041	L\$ICP 002104 G	L10044 032016	PATR6 012000 G	RPTCNT 003462 G
ISCLN = 000041	L\$INIT 021302 G	MBR = 100012 G	PATR7 012020 G	RPTFLG 003517 G
ISDU = 000041	L\$LADP 002026 G	MDSET 007466 G	PATR8 012052 G	RPT1A 020410
ISHRD = 000041	L\$LAST 032004 G	MEMOM 023166	PATTLB 011620	RPT1B 020465
ISINIT= 000041	L\$LOAD 002100 G	MISCFG 003531 G	PATTM 031225	RPT1C 020536
ISMOD = 000041	L\$LUN 002074 G	MOD.CO= 000400 G	PCMDWD 003424 G	RPT1D 020607
ISMSG = 000041	L\$MREV 002050 G	MOD.C1= 001000 G	PIRE 002216 G	RPT1E 021035
ISPROT= 000040	L\$NAME 002000 G	MOD.C2= 002000 G	PNT = 001000 G	RPT1F 020713
ISPTAB= 000041	L\$PRIO 002042 G	MOD.C3= 004000 G	PRI = 002000 G	RPT1G 020764
ISPR = 000041	L\$PROT 021274 G	MOVMSG 012734 G	PRI00 = 000000 G	RPT1I 021161
ISRPT = 000041	L\$PR1 002112 G	MSGCNT= 000020 G	PRI01 = 000040 G	RPT1J 021065
ISSEG = 000041	L\$REPP 002062 G	MSGPKA 002544 G	PRI02 = 000100 G	RPT1K 021152
ISSETU= 000041	L\$REV 002010 G	MSGPKT 002354 G	PRI03 = 000140 G	RRANV 002205 G
ISSFT = 000041	L\$RPT 017546 G	MSGPK0 002374 G	PRI04 = 000200 G	RRBC 002666 G
ISSRV = 000041	L\$SOFT 030040 G	MSGPK1 002414 G	PRI05 = 000240 G	RRECL = 000020 G
ISSUB = 000041	L\$SPC 002056 G	MSGPK2 002434 G	PRI06 = 000300 G	RRREC 003006 G
ISTST = 000041	L\$SPCP 002020 G	MSGPK3 002454 G	PRI07 = 000340 G	RRUNR 003016 G
JLOC 003444 G	L\$SPTP 002024 G	MS.RFC= 000004 G	PRXST 017370 G	RRVM 030663
JLOOP 003442 G	L\$STA 002030 G	MS.XS0= 000006 G	PTCMDS 027570 G	RTLE 014466 G
JMP = 000040 G	L\$SW 002204 G	MS.XS1= 000010 G	PWRFLG 003527 G	RTLRTN 014612
JMP.C = 000040 G	L\$TEST 002114 G	MS.XS2= 000012 G	RANB 003432 G	RWCPK 002350 G
JSJMP = 000167	L\$TIML 002014 G	MS.XS3= 000014 G	RANBC = 153624 G	RWD = 102010 G
LENMSK 003430 G	L\$UNIT 002012 G	MS.XS4= 000016 G	RANCMD 026136	RWERR 003467 G
LOE = 040000 G	L10000 002202	NCMD.C= 177740 G	RANDOM 003515 G	RSSAVE 003452 G
LOG 015544 G	L10001 002326	NCNT 003412 G	RANP = 000007 G	SCCNT 003336 G
LOT = 000010 G	L10002 006116	NCNT1 003414 G	RANRD 026176 G	SCERM 004475 G
L\$ACP 002110 G	L10003 007070	NEXTSP 030106	RANS 003434 G	SCH = 140004 G
L\$APT 002036 G	L10004 010040	NEXTU 017126 G	RANSC = 032561 G	SCHBK 002474 G
L\$AU 024030 G	L10005 010046	NINUSE= 177774 G	RANW 026422 G	SCHCNT= 000012 G
L\$AUT 002070 G	L10006 010054	NODEV 005543 G	RANWR 026310 G	SEQEND 003740 G
L\$AUTO 023272 G	L10007 010062	NOINTM 004670 G	RANWV 026334 G	SETCH 010162 G
	L10010 021272	NRDYM 023656	RASFR 026374 G	SETDEF 007766 G

PARAMETER CODING
SYMBOL TABLE

MACRO M1113 25-MAY-82 09:51 PAGE 20-98

B 12

SEQ 0144

SETRW 010206 G	TRAP4 023706 G	T\$LAST= 000001	T1 024134 G	WRTY 014614 G
SETUP 011172 G	TSAM 004705 G	T\$LOLI= 000000	T1SWB 003523 G	WRTYCT 003316 G
SFF = 105010 G	TSBA = 002514 G	T\$LSYM= 010000	T1.1 024144	WRTYER 003464 G
SFPTBL 002204 G	TSC.FC= 177717 G	T\$LTNO= 000005	T1.10 024554	WRTYFG 003463 G
SFR = 105410 G	TSC.TC= 177761 G	T\$NEST= 177777	T1.11 024574	WRUNR 002776 G
SOFINI 007072 G	TSDB 002514 G	T\$NS0 = 000000	T1.12 024660	WSM = 140006 G
SRF = 104010 G	TSMD 031235	T\$NS1 = 000005	T1.2 024350	WSMBK 002506 G
SRR = 104410 G	TSNP 003534 G	T\$NS2 = 000002	T1.3 024374	WSSR 012700 G
STAERM 006120 G	TSSR 002524 G	T\$PCNT= 000C00	T1.4 024414	WTBF 031305
STAER1 006436	TSSREG 003454 G	T\$PTAB= 010043	T1.5 024434	WTBUF 002316 G
STAER2 006616	TSUNT 003532 G	T\$PTHV= 000001	T1.6 024454	WTM = 100011 G
STAER3 006675	TSVCT 002534 G	T\$PTNU= 000001	T1.7 024474	WTMFLG 003456 G
STAER4 006733	TS.A16= 000400 G	T\$SAVL= 177777	T1.8 024514	WTR = 101011 G
STAER5 006753	TS.A17= 001000 G	T\$SEGL= 177777	T1.9 024534	WTV = 104105 G
STAER6 006562	TS.NBA= 002000 G	T\$SIZE= 000005	T2 025622 G	WTVERM 004430 G
STAER7 006530	TS.NXM= 004000 G	T\$SUBN= 000000	T3 026456 G	WTYBRF 015066
STAFLG 003526 G	TS.OFL= 000100 G	T\$TAGL= 177777	T4 026652 G	WTYCMD 015062
SVCGBL= 000000	TS.RMR= 010000 G	T\$TAGN= 010045	T5 027032 G	WTYWRD 015064
SVCINS= 000001	TS.SC = 100000 G	T\$TEMP= 000000	TSWEOT 027612 G	X\$ALWA= 000000
SVCSUB= 000000	TS.SPE= 020000 G	T\$TEST= 000005	UAM = 000200 G	X\$FALS= 000040
SVCTAG= 000000	TS.SSR= 000200 G	T\$TSTM= 177777	UNIWLK 005653	X\$OFFS= 000400
SVCTST= 000000	TS.UPE= 040000 G	T\$TSTS= 000001	UNL = 100412 G	X\$TRUE= 000020
SWBFLG 003520 G	TS1MD 002312 G	T\$SAU = 010016	UNREC 003470 G	X0.BOT= 000002 G
SWB.C = 010000 G	TS5ADR 027770	T\$SAUT= 010013	URERM 005041 G	X0.EOT= 000001 G
SWSET 004231 G	TS5CL 002564 G	T\$SCLE= 010014	VFEXC 016130 G	X0.LET= 020000 G
S\$LSYM= 010000	TS5INT 002554 G	T\$SDAT= 010044	VFISU 016356 G	X0.ONL= 000100 G
TCCRA 013372	TS5INO 010034 G	T\$SDU = 010015	VFYCNT 003346 G	X0.RLL= 010000 G
TCC0 013412 G	TS5IN1 010042 G	T\$SHAR= 010040	VFYDAT 01604 G	X0.RLS= 040000 G
TCC1 013430 G	TS5IN2 010050 G	T\$SHW = 010000	VFYFLG 003516 G	X0.TMK= 100000 G
TCC2 013446 G	TS5IN3 010056 G	T\$SINI= 010012	VFY.C = 000100 G	X0.WLK= 000004 G
TCC3 013556 G	TS5SW 002574 G	T\$MSG= 010003	WAITF 007204 G	X2.BFE= 000100 G
TCC4 013574 G	TS5UNT 030014	T\$SPC = 000001	WLKCHK 007356 G	X2.EFE= 000200 G
TCC5 014210 G	TS5VCT 030005	T\$SPRO= 010011	WLKZRO 011734	X2.OPM= 100000 G
TCC6 014306 G	T\$ARGC= 000003	T\$SPTA= 010043	WRBC 002626 G	X3.DCK= 000010 G
TCC7 014450 G	T\$CODE= 001004	T\$SRP1= 010010	WRECL = 000020 G	X3.RNY= 157400 G
TC2RTN 013554	T\$ERRN= 000002	T\$SSOF= 010041	WRR = 105005 G	X4.HSS= 100000 G
TIME1 003436 G	T\$EXCP= 000000	T\$SSRV= 010007	WRREC 002766 G	X4.RCE= 040000 G
TIME2 003440 G	T\$FLAG= 000041	T\$SSUB= 010033	WRT = 104005 G	ZROPAT 011704
TOERM 004453 G	T\$FREE= 032016	T\$SSW = 010001	WRTCHK 007270 G	\$LSTIN= 000001
TOOMM 004727 G	T\$GMAN= 000000	T\$STES= 010037	WRTCHR 007436 G	\$LSTTA= 000001
TRAPD4 003530 G	T\$HILI= 000010			

. ABS. 032016 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 26197 WORDS (103 PAGES)
DYNAMIC MEMORY: 20346 WORDS (78 PAGES)
ELAPSED TIME: 00:17:56
CVTSEA0,CVTSEA0/-SP=SV34R/ML,TSV1E,CVTSEA.SRC/EN:AMA:ABS/DS:GBL