

TM78, TU78

TM78 PERF EXER  
CZTMJAO

AH-E646A-MC  
FICHE 1 OF 1

NOV 1980  
COPYRIGHT © 1980  
MADE IN USA



Microfiche grid containing multiple frames of data, including text and graphics.



.REM 8

IDENTIFICATION  
-----

PRODUCT CODE: AC-E645A-MC  
PRODUCT NAME: CZTMJAO TM78 PERFORMANCE EXERCISER  
MAINTAINER: DIAGNOSTIC ENGINEERING  
DATE: JUNE 1,1980  
AUTHOR: JEFF MEYER

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) 1980 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL  
DEC

PDP  
DECUS

UNIBUS  
DECTAPE

MASSBUS

8

.REM 8

TABLE OF CONTENTS  
-----

1.0 GENERAL INFORMATION  
1.1 PROGRAM ABSTRACT  
1.2 PROGRAM STRUCTURE  
1.3 SYSTEM REQUIREMENTS  
1.3.1 MINIMUM SYSTEM CONFIGURATIONS (HARDWARE)  
1.3.2 MINIMUM SYSTEM CONFIGURATIONS (SOFTWARE)  
1.4 RELATED DOCUMENTS AND STANDARDS  
1.5 DIAGNOSTIC HIERARCHY PREREQUISITES  
1.6 ASSUMPTIONS  
  
2.0 OPERATING INSTRUCTIONS  
2.1 BASIC INFORMATION TO VERIFY DEVICE STATUS  
2.2 USER OPTIONS AVAILABLE IN DIAGNOSTIC  
  
3.0 USER INFORMATION  
3.1 COMMANDS  
3.2 DATA PATTERNS  
3.3 RETRY SEQUENCES  
3.3.1 WRITE RETRY SEQUENCE  
3.3.2 READ RETRY SEQUENCE  
  
4.0 ERROR INFORMATION  
4.1 ERRORS REPORTED  
4.2 SAMPLE ERROR PRINTOUTS  
  
5.0 STATISTICAL REPORT  
5.1 SAMPLE STATISTICAL PRINTOUT  
5.2 STATISTICAL UPDATES  
1.0 GENERAL INFORMATION  
-----

1.1 PROGRAM ABSTRACT  
-----

THE FUNCTION OF THIS PROGRAM IS THREE FOLD.

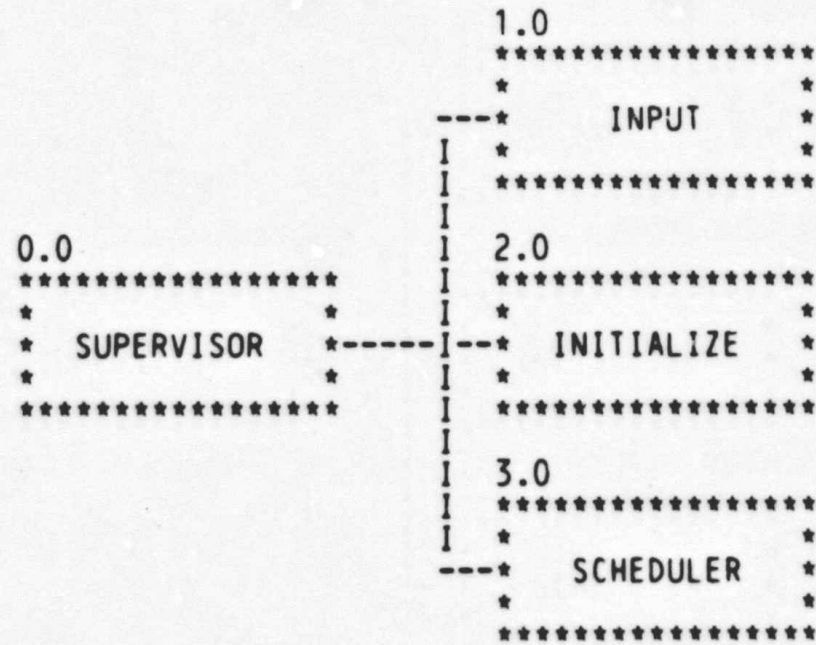
1. DATA RELIABILITY PERFORMS A PREDEFINED SEQUENCE OF COMMANDS(\*), ON A MAXIMUM OF 8 TU78 TYPE DRIVES IN SEQUENCE, WITH EXTENSIVE ERROR REPORTING FACILITIES(\*). IT'S PURPOSE IS TO RECORD THE PERFORMANCE QUALITY OF EACH TRANSPORT THROUGH THE ACCUMULATION OF STATISTICS(\*).
2. BRUTUS IS INTENDED AS A BRUTE FORCE ROUTINE TO EXECUTE A COMMAND OR SERIES OF UP TO SIX COMMANDS(\*) CONTINUOUSLY, TO THE SAME SLAVE TRANSPORT REGARDLESS OF THE RESULTS OF THE OPERATION.
3. BRUTUS WITH ERRORS IS A COMBINATION OF THE ABOVE TWO. IT

ADDS TO BRUTUS ERROR REPORTING FACILITIES AND ACCUMULATION  
OF STATISTICS SUPPLIED IN THE DATA RELIABILITY OPTION.

ALL THREE ARE CONVENIENTLY ACCESSED THROUGH THE TELETYPE USER  
INTERFACE.

-----  
(\* ) DEFINED IN DETAIL IN SECTIONS 3 AND 4

1.2 PROGRAM STRUCTURE  
-----



```
2.1
*****
* BUILD H-P-T *
---* STAT TABLES *
I * SETUP FLAGS *
I *****
I
I 2.2
I *****
I *CHECK TU EXIST.*
I--* + TM78 TYPE *
I * DRIVES *
I *****
I
I 2.3
I *****
I * ASK *
I--* BTUTUS *
I * QUESTIONS *
I *****
I
I 2.4
I *****
I * CLEAR *
I--* STATISTICS *
I * *
I *****
I
I 2.5
I *****
I * *
I--* ERROR REPORTS *
I * *
I *****

2.0
*****
* INITIALIZE *
* CLEAR *-----
* FLAGS *
*****

3.1
*****
* SEQUENCE *
---* DEFINITION *
```

```

I  *
I  *****
I  3.2
I  *****
I  *
I  --* ISSUE COMMAND *
I  *
I  *****
3.0 *****
*
* SCHEDULER *-----
*
* *****
I  3.3
I  *****
I  *
I  --* COMMAND
I  * TERMINATION *
I  *
I  *****
I  3.4
I  *****
I  *
I  --* O/P STATS *
I  *
I  *****

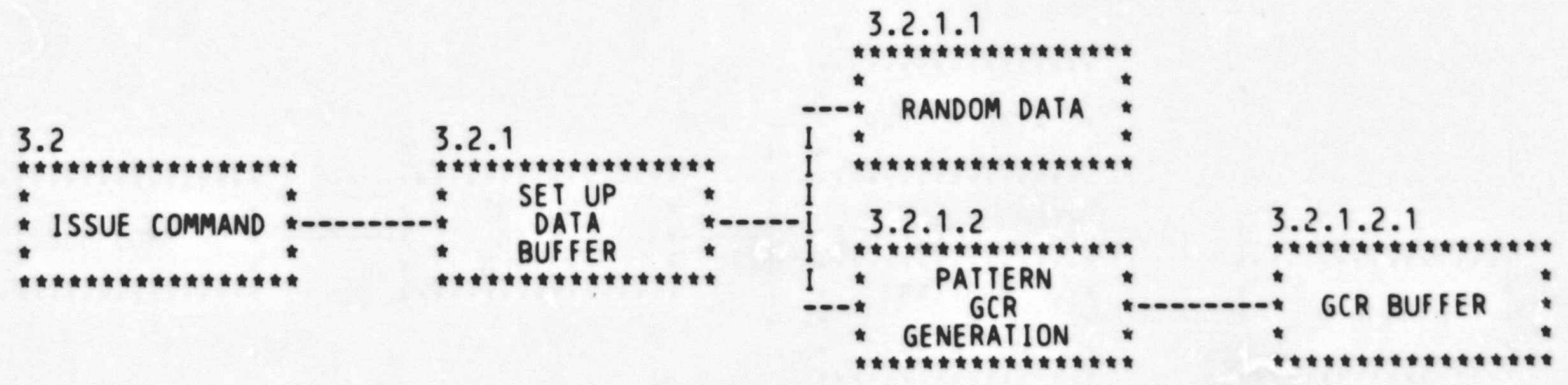
```

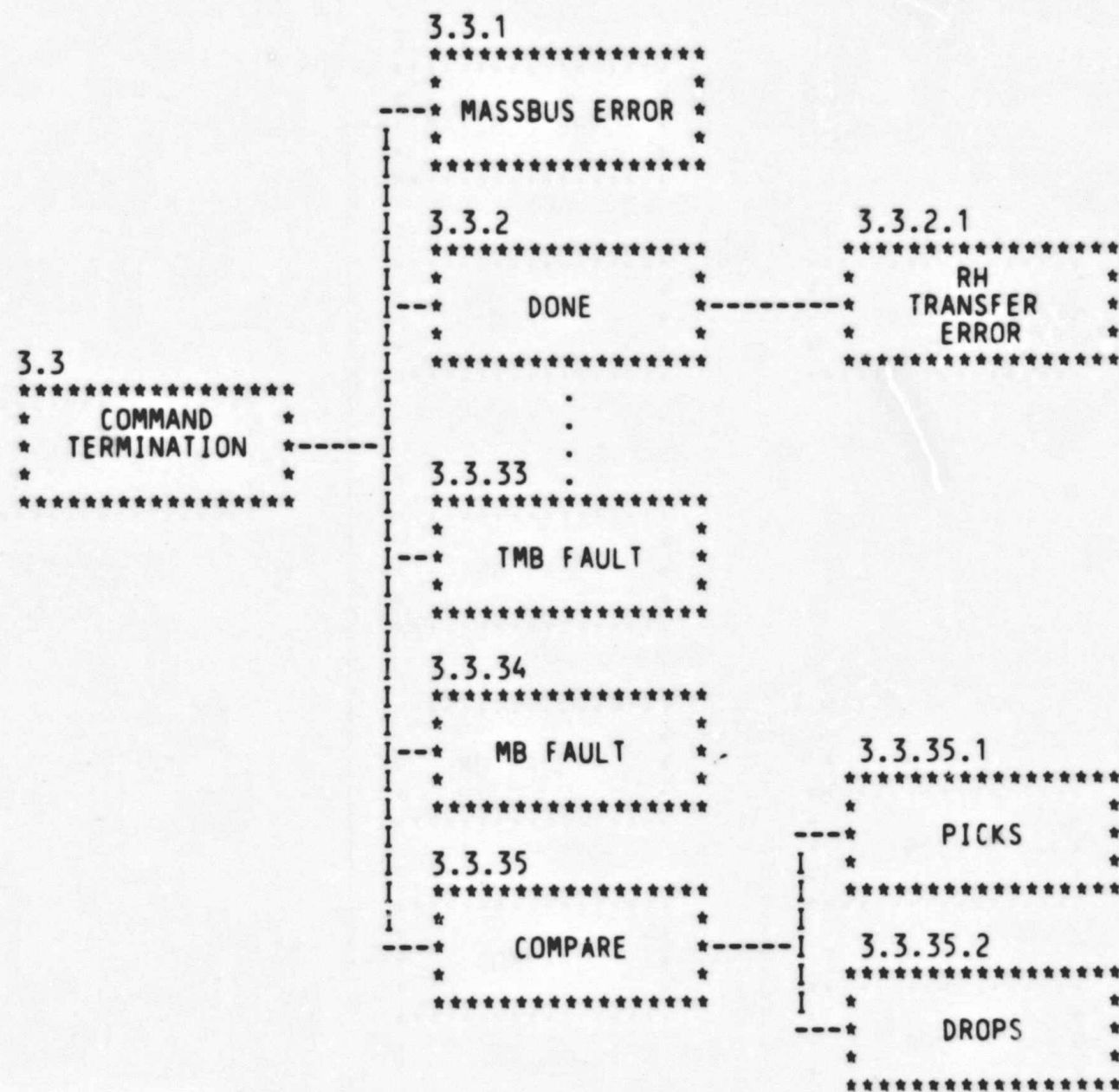
```

I  3.1.1
I  *****
I  *
I  ---*BRUTUS SEQUENCE*
I  *
I  *****
I  3.1.2
I  *****
I  *
I  --* CANNED
I  * SEQUENCE *
I  * #1 *
I  *****
3.1 *****
* SEQUENCE *
* DEFINITION *-----
*
* *****
I  3.1.3
I  *****
I  *
I  --* CANNED
I  * SEQUENCE *
I  * #2 *
I  *****

```

```
I 3.1.4
I *****
I * CANNED *
I--* SEQUENCE *
I * #3 *
I *****
I
I 3.1.5
I *****
I *
I--*RETRY SEQUENCE *
I *
I *****
```





1.3 SYSTEM REQUIREMENTS

1.3.1 MINIMUM SYSTEM CONFIGURATIONS (HARDWARE)

1. PDP-11/05
2. 24K OF MEMORY.



3. CONSOLE
4. AT LEAST ONE TM78 (MAXIMUM 8)
5. AT LEAST ONE TU78 (NO MORE THAN 4 PER TM78) AND NO MORE THAN EIGHT
6. SOME PERIPHERAL LOAD DEVICE (POSSIBLE TM78 ITSELF)

#### 1.3.2 MINIMUM SYSTEM CONFIGURATIONS (SOFTWARE)

1. DIAGNOSTIC SUPERVISOR
2. XXDP+ MONITOR AND DRIVER

#### 1.4 RELATED DOCUMENTS AND STANDARDS

1. XXDP+ USERS MANUAL (CHQUSA.SEQ)
2. MASSBUS SPEC (DEC STD 159)

#### 1.5 DIAGNOSTIC HIERARCY PREREQUISITES

1. DIAGNOSTIC SUPERVISOR
2. XXDP+ (LOAD DEVICE)

#### 1.6 ASSUMPTIONS

THIS PERFORMANCE EXERCISER WILL NOT ISOLATE MALFUNCTIONS TO THE MODULE LEVEL.

#### 2.0 OPERATING INSTRUCTIONS

#### 2.1 BASIC INFORMATION TO VERIFY DEVICE STATUS

THINGS IN " " USER MUST TYPE IN.

1. BOOT XXDP+ MEDIA AND ANSWER QUESTIONS
2. .'R ZTMJA0 <CR>''
3. DRS-C>'START/PASS:1 <CR>''
4. CHANGE HW(L) ? 'Y<CR>''

5. # UNITS (D) ? "X<CR>" X = NUMBER OF TU78'S  
TO BE TESTED (MAXIMUM EIGHT).
6. UNIT 0  
RH ADDRESS (0) 172400 ? "<CR>" IF DEFAULT ADDRESS IS CORRECT.  
"AAAAAA<CR>" AAAAAA=RH ADDRESS OF  
UNIT IF NO DEFAULT.  
VECTOR ADDRESS (0) 224 ? "<CR>" IF DEFAULT ADDRESS IS CORRECT.  
"AAA<CR>" AAA = ADDRESS OF INTERRUPT  
VECTOR FOR ABOVE  
RH ADDRESS  
TM78 # (D) 0 ? "<CR>" IF DEFAULT NUMBER IS CORRECT.  
"X" X=TM78 NUMBER (0-7 ARE ALL POSSIBILITIES)  
TU78 # (D) 0 ? "<CR>" IF DEFAULT NUMBER IS CORRECT.  
"X" X=TU78 NUMBER (0-3 ARE ALL POSSIBILITIES)

THE ABOVE SEQUENCE WILL BE REPEATED FOR "X" TIMES, WHERE "X"  
IS THE ANSWER IN STEP 5.

7. CHANGE SW (L) ? "N<CR>"

PROGRAM SHOULD NOW PRINT OUT A SERIAL # FOR EACH TU78 AND  
STATISTICS AT THE END OF VERIFICATION FOR EACH TU78.

## 2.2 USER OPTIONS AVAILABLE IN DIAGNOSTIC

-----

1. STEPS 1 + 2 OF SECTION 2.1 STAY THE SAME
2. DRS-C> (ENTER A COMMAND WITH APPROPRIATE FLAGS)\*
3. CHANGE HW (L) ? THIS QUESTION ONLY APPEARS IF A START  
COMMAND WAS ISSUED. IF YOU ANSWER YES THEN YOU CAN CHANGE  
THE # OF UNITS TESTED AND THE INFORMATION PERTAINING TO EACH  
UNIT. (SEE 6 IN SECTION 2.1) IF YOU ANSWER NO THEN THE #  
OF UNITS AND THEIR LOCATION STAYS THE SAME AS INPUTTED THE  
FIRST TIME.
4. CHANGE SW (L) ? IF YOU ANSWER NO TO THIS QUESTION THE PROGRAM  
TAKES THE MOST RECENT ANSWERS TO THE QUESTION AND USES THEM  
AGAIN. IF YOU ANSWER YES THEN THE FOLLOWING QUESTIONS FOLLOW:
  1. BRUTUS OPTION (L) N ? <CR> IF YOU SAY NO THEN GOES TO  
NEXT QUESTION. IF YES THEN YOU HAVE TWO OPTIONS.
    - A. BRUTUS WITH NO ERRORS OR STATISTICS
    - B. BRUTUS WITH ERRORS AND STATISTICS

THESE BRUTUS OPTIONS ALLOW THE USER TO INPUT A STRING OF  
UP TO SIX COMMANDS.\*\* THESE COMMANDS CAN BE ANY AVAILABLE  
COMMAND IN ANY ORDER. THE UNITS WILL NOT BE DROPPED  
REGARDLESS OF THE ERRORS THAT OCCUR. THE PATTERN, RECORD  
LENGTH, AND NUMBER OF CONSECUTIVE TIMES A COMMAND SHOULD BE  
ISSUED ARE USER OPTIONS. BRUTUS WILL ALSO ALLOW YOU TO

RUN PAST EOT AND OFF TAPE. BRUTUS WITH ERRORS ALLOWS THE USER TO SELECT DATA COMPARES, RETRIES, STATISTICS AT END OF PASS, AND STATISTICS AT END OF TAPE.

- \* REFER TO CHQUSA.SEG MANUAL  
\*\* REFER TO SECTION 3.1 FOR COMMANDS OR TM78 PROGRAMMING MANUAL

2. CANNED SEQUENCE (1-4) 4 ? THIS ALLOWS FOR FOUR OPTIONS. ALL OPTIONS DO THE FOLLOWING:

- A. RECORD #
  - B. FILE #
  - C. REPORTS ERRORS
  - D. UPDATES STATISTICS
  - E. DROPS UNITS WHEN FATAL ERRORS OCCUR AND TRYS TO PICK UP DROPPED UNITS ON EVERY REWIND.
1. CANNED SEQUENCE 1 REWINDS TAPE AND THEN READS THE TAPE UNTIL EOT, LEOT, OR FATAL ERROR IS FOUND.
  2. CANNED SEQUENCE 2 REWINDS TAPE AND THEN STARTING WITH PE WRITES, READ REVERSE, AND READS THE LENGTH OF TAPE. IT THEN REWINDS AND DOES THE SAME PROCEDURE IN GCR. THIS SEQUENCE DOES RETRIES, DATA COMPARES, RANDOM RECORD LENGTHS, (MAXIMUM 4K), AND RANDOM DATA PATTERNS.\*
  3. CANNED SEQUENCE 3 REWINDS TAPE AND THEN STARTING WITH PE WRITES 24 RECORDS, WRITES TM, SPACE REVERSE FILE, READ REVERSE 24 RECORDS, READ FORWARD 24 RECORDS, AND SPACE FORWARD FILE. THIS SEQUENC EIS DONE THE LENGTH OF TAPE REWOUND AND REPEATED FOR GCR. DATA PATTERNS ARE CHANGED EVERY NEW FILE. RECORD LENGTH IS 2K. THIS SEQUENCE DOES RETRIES, AND DATA COMPARES.
  4. CANNED SEQUENCE 4 -- FUNCTIONAL SEQUENCE

A.NOOP TEST:

THIS TEST WILL ASSURE THAT THE NOOP FUNCTION WILL EXECUTE AND THAT A DONE INTERUPT CODE WILL BE RECIEVED.

B.REWIND TEST:

THIS TEST WILL ASSURE THAT THE REWIND FUNCTION WILL POSITION THE TAPE TO BOT WITH NO ERRORS.

- 1.ISSUE A REWIND COMMAND
- 2.CHECK FOR 'BOT'
- 3.ERASE 200' OF TAPE TO GET OFF 'BOT'
- 4.REWIND TAPE AGAIN
- 5.CHECK FOR 'BOT'

C.WRITE/READ TEST:

THIS TEST WILL CHECK BOTH READ AND WRITE  
FUNCTIONS IN ALL DENSITIES (PE + GCR)

1. REWIND TAPE TO BOT
2. WRITE 200 RECORDS
  - A. ALL ONES DATA
  - B. 1000 FRAMES
  - C. 1600 BPI
3. CHECK FOR ERRORS
4. READ REVERSE THEN FORWARD ALL 200 RECORDS
5. REPEAT STEP 1-4 FOR GCR

D. WRITE CHECK TEST:

THIS TEST WILL CHECK FOR CORRECT OPERATION  
OF THE WRITE CHECK REVERSE AND WRITE CHECK  
FORWARD FUNCTIONS IN BOTH PE AND GCR.

1. REWIND
2. WRITE 1 RECORD
  - A. 1000 FRAMES
  - B. PE MODE FIRST (1600 BPI)
3. ISSUE A WRITE CHECK REVERSE
4. ASSURE NO "WCE"
5. ISSUE A WRITE CHECK FORWARD
6. ASSURE NO "WCE"
7. REPEAT FOR GCR MODE.

E. SPACE TEST:

THIS TEST WILL ASSURE THAT PROPER POSITIONING  
IS MAINTAINED BY BOTH SPACE FORWARD RECORD  
AND SPACE REVERSE RECORD.

1. REWIND TAPE TO BOT
2. WRITE 200 RECORDS PE
  - A. EACH RECORD IS ONE FRAME LARGER THAN  
THE LAST, STARTING WITH 100 FRAMES.  
THIS WILL ALLOW FOR POSITION CHECKING  
BY RECORD SIZE.
3. EACH RECORD IS ERROR CHECKED
4. DATA IS NOT CHECKED
5. SPACE REVERSE 177 RECORDS AND READ REVERSE  
1, THE FRAME COUNT SHOULD BE 100. THIS IS  
THE SIZE OF THE FIRST RECORD.
6. NOW, SPACE FORWARD 176 RECORDS AND READ  
FORWARD 1 RECORD. THE FRAME COUNT SHOULD  
BE 177. THIS IS THE SIZE OF THE NEXT TO LAST  
RECORD.
7. CONTINUE SPACING REVERSE THEN FORWARD WHILE  
DECREMENTING THE RECORD COUNT UNTIL ALL  
POSSIBILITIES HAVE BEEN CHECKED.
8. REPEAT 1-7 FOR GCR (6250 BPI)

F. TAPE MARK WRITE/READ TEST:

THIS TEST WILL ASSURE THAT A TAPE MARK

CAN BE WRITTEN AND READ IN BOTH PE AND GCR.

- 1.REWIND TAPE TO BOT
- 2.ISSUE A WRITE TAPE MARK SET PE
- 3.CHECK FOR ERRORS
- 4.SPACE REVERSE FILE (TO TAPE MARK)
- 5.SPACE FORWARD FILE (TO TAPE MARK)
- 6.BOT SHOULD NOT BE REACHED
- 7.REPEAT STEPS 1-6 FOR GCR.

G.TAPE MARK SPACE TEST:

THIS TEST WILL ASSURE THAT SPACING WILL BE TERMINATED BY THE RECOGNITION OF A TAPE MARK BOTH IN PE AND GCR MODES.

- 1.REWIND TAPE
- 2.WRITE THE FOLLOWING PATTERN ON TAPE  
:TM:20 REC:TM:40 REC:TM:60 REC:TM:100 REC:TM:  
3.ASSURE NO ERRORS
- 4.SPACE REVERSE 200 RECORDS
- 5.TAPE SHOULD STOP ON EACH TAPE MARK
- 6.BOT SHOULD NEVER BE REACHED
- 7.CHECK FOR UNEXPECTED TAPE MARK INT CODE
- 8.REPEAT FOR GCR MODE

H.WORST CASE VACUUM COLUMN TEST

THIS TEST WILL ATTEMPT TO STRESS THE VACUUM COLUMNS TO VERIFY THAT THEY WILL NOT RELEASE THE TAPE OR DROP VACUUM.

- 1.REWIND THE TAPE
- 2.WRITE A PATTERN ON TAPE CONSISTING OF 25 20 BYTE RECORDS.
- 3.REWIND THE TAPE
- 4.SPACE FORWARD 1 RECORD
- 5.REWIND
- 6.SPACE FORWARD 2 RECORDS
- 7.REWIND
- 8.CONTINUE INCREMENTING THE RECORD COUNT UNTIL ALL RECORDS HAVE BEEN SPACED OVER.
- 9.THEN BEGIN DECREMENTING THE RECORD COUNT UNTIL ZERO
- 10.REPEAT FOR GCR

I.ERASE TEST:

THIS TEST WILL ASSURE THAT THE ERASE FUNCTION WILL INDEED ERASE TAPES.

- 1.REWIND TO BOT
- 2.ISSUE 200 ERASE 3" COMMANDS

3. ASSURE NO ERRORS
4. REWIND TAPE TO BOT
5. ISSUE A READ COMMAND
6. EXPECT NOT CAPABLE INTERRUPT CODE
7. REPEAT FOR GCR

#### J. SHORT AND LONG RECORD TEST

THIS TEST WILL CHECK TO SEE THAT THE TM78/TU78 CAN WRITE AND READ SHORT AND LONG RECORDS IN BOTH PE AND GCR DENSITIES.

1. REWIND TAPE
2. SET BYTE COUNT TO ZERO
3. WRITE 20 RECORDS IN PE
4. READ REVERSE 20 RECORDS
5. READ FORWARD 20 RECORDS
6. INCREMENT BYTE COUNT AND REPEAT STEPS 3-5 UNTIL THE BYTE COUNT IS GREATER THAN 20
7. REWIND TAPE
8. SET BYTE COUNT TO 177777
9. WRITE 2 RECORDS WITH DATA PATTERN OF 000377
10. READ REVERSE 2 RECORDS
11. READ FORWARD 2 RECORDS
12. DECREMENT BYTE COUNT
13. REPEAT STEPS 9-12 UNTIL BYTE COUNT = 177776
14. REPEAT STEPS 1 --> 13 USING GCR MODE

#### K. CAPSTAN WORST CASE TEST:

THIS TEST WILL ASSURE THAT A CAPSTAN WRAP WILL NOT RESULT AFTER A HIGH SPEED REWIND FROM THE CENTER OF TAPE HAS BEEN ISSUED.

1. REWIND
2. ERASE 400' OF TAPE
3. WAIT FOR TAPE TO SETTLE (1 MIN.)
4. REWIND TAPE AT HIGH SPEED

-----  
\* REFER TO SECTION 3.2 FOR DATA PATTERNS.

### 3.0 USER INFORMATION

-----

#### 3.1 COMMANDS

-----

FUNCT  
CODE\*GO

NAME

DESCRIPTION

03	NO-OP	GENERATE A UNIQUE NO-OP INT CODE..
05	UNLOAD	UNLOAD TAPE AND INTERRUPT IMMEDIATELY.
07	REWIND	REWIND TAPE AND INTERRUPT WHEN DONE.
11	SENSE	PUT STATUS INFO INTO CAS 6,7 AND 10.
13	DSE	ERASE THE REMAINDER OF TAPE AND REWIND.
15	WTM PE	WRITE A PE TAPE MARK.
17	WTM GCR	WRITE A GCR TAPE MARK.
21	SP FWD REC	SPACE FORWARD RECORD, STOP IF TAPE MARK.
23	SP REV REC	SPACE REVERSE RECORD, STOP IF TAPE MARK.
25	SP FWD FILE	SPACE FORWARD FILE (TO TAPE MARK).
27	SP REV FILE	SPACE REVERSE FILE (TO TAPE MARK).
31	SP FWD EITHER	SPACE FWD EITHER RECORD OR FILE
33	SP REV EITHER	SPACE REV EITHER RECORD OR FILE
35	ERG PE	ERASE 3 INCHES OF TAPE, SET PE.
37	ETG GCR	ERASE 3 INCHES OF TAPE, SET GCR.
41	CLOSE FILE PE	WRITE 2 TAPE MARKS, SPACE REV 1, SET PE.
43	CLOSE FILE GCR	WRITE 2 TAPE MARKS, SPACE REV 1, SET GCR.
45	SPACE LEOT	SPACE FWD UNTIL 2 TAPE MARKS, SPACE REV 1.
47	SPACE FWD FILE/LEOT	SPACE FORWARD TO TAPE MARK, STOP IF 2 SUCCESSIVE TAPE MARKS (LOGICAL END OF TAPE), THEN SPACE REVERSE ONE TAPE MARK.
61	WRITE PE	WRITE PHASE ENCODED RECORDS.
63	WRITE GCR	WRITE GROUP CODED RECORDS.
71	READ FWD	READ RECORDS FORWARD.
77	READ REV	READ RECORDS REVERSE.

3.2 DATA PATTERNS

<u>PATTERN #</u>	<u>DESCRIPTION</u>
0	ALL ZEROES
1	ALL ONES
2	ALTERNATING BYTE OF ZEROES AND ONES
3	REPEATING 2 BYTES ONES 2 BYTES ZEROES
4	REPEATING 4 BYTES ONES 4 BYTES ZEROES
5	REPEATING 3 BYTES ZEROES 1 BYTE ONES
6	USER INPUTS 14 WORDS OF ANY PATTERN
7	INCREMENTING PATTERN 0-255
8	RANDOM PATTERN
9	WORSE CASE GCR PATTERN

3.3 RETRY SEQUENCES

3.3.1 WRITE RETRY SEQUENCE

A WRITE RETRY COMMAND WILL FOLLOW THE SEQUENCE BELOW.

```
-----  
! WRITE RECORD BAD !...  
-----  
.....  
.  
-----  
!WRITE RECORD AGAIN!...      FAILED  
-----  
.....      GOOD  
.  
-----  
!WRITE RECORD AGAIN!...  
-----  
.....      GOOD  
.  
-----  
!WRITE RECORD AGAIN!...  
-----  
.....      GOOD  
.  
-----  
!WRITE RECORD AGAIN!...  
-----  
.....      GOOD  
.  
-----  
!   RETRY DONE   !  
-----  
.....  
.  
-----  
! ERASE 3 INCHES !...  
-----  
.....  
.  
-----  
!   WRITE RECORD   !  
-----
```

IF WRITE GOOD FOUR CONSECUTIVE TIMES THEN SOFT WRITE ERROR.  
IF AFTER ERASE 3 INCHES WRITE IS GOOD FOUR CONSECUTIVE TIMES  
THEN BAD TAPE SPOT.  
IF AFTER ERASING 4 TIMES AND STILL CAN'T WRITE THE SAME RECORD  
GOOD FOUR CONSECUTIVE TIMES THEN HARD WRITE ERROR.

3.3.2 READ RETRY SEQUENCE



THE RETRY OF A FAILING READ OPERATION IS ACCOMPLISHED IN CONJUNCTION WITH THE TM78 OPERATIONAL MICRO CODE WHEN A READ COMMAND IS ISSUED, THE SUPPRESS ERROR REPOSITIONING (SER) IS DISABLED. IF THE TM78 DETECTS AN ERROR IN PERFORMING THE READ OPERATION WHICH REQUIRES RETRYING, IT REPOSITIONS THE TAPE AND RESPONDS TO THE HOST PROCESSOR WITH AN INTERRUPT CODE OF 22 (RETRY). AT THIS POINT THE HOST PROCESSOR REISSUES THE READ COMMAND. THIS SEQUENCE WILL BE REPEATED UP TO TWELVE TIMES IF SUBSEQUENT RETRIES ARE UNSUCCESSFUL. IF AFTER THE TM78 HAS ISSUED THE 22 (RETRY) INTERRUPT THE TWELFTH TIME THE READ IS STILL UNSUCCESSFUL, THE TM78 WILL ISSUE AN INTERRUPT OF 23 (READ OPPOSITE) AND THE HOST PROCESSOR WILL RESPOND BY ISSUING A READ COMMAND IN THE OPPOSITE DIRECTION. THIS SEQUENCE CAN CONTINUE UNTIL THE TM78 HAS RESPONDED WITH INTERRUPT CODE 23 (READ OPPOSITE) EIGHT TIMES. AT THIS POINT THE TM78 WILL RESPOND WITH ONE INTERRUPT CODE 22 (RETRY), AND FINALLY INTERRUPT CODE 24 (UNREADABLE). THIS IS A HARD READ ERROR.

4.0

ERROR INFORMATION  
-----

4.1

ERRORS REPORTED  
-----

INITIAL ERRORS REPORTED IN THE INIT CODE FOR ALL OPTIONS OF THE PROGRAM.

1. INSUFFICIENT MEMORY AVAILABLE DIAGNOSTIC CANNOT RUN
2. THIS (TU #) ON THIS (RH ADDRESS) AND (TM78 #) IS NOT A TM78 TYPE DRIVE
3. TOO MANY UNITS TO BE TESTED UNDER BRUTUS, ONLY ONE ALLOWED
4. SUPERVISOR UAM FLAG IS SET, MUST BE CLEARED IN ORDER TO RUN BRUTUS OPTION
5. THIS (TM78 #) ON THIS RH ADDRESS WITH THIS (TU #) HAS NOT RESPONDED TO SENSE COMMAND
6. TOO MANY UNITS TO TEST UNKNOWN TAPE, ONLY ONE ALLOWED.
7. THIS TU # ON THIS TM78 # ON THIS RH ADDRESS HAS GENERATED AN INTERRUPT.
8. THIS TU # ON THIS TM78 # ON THIS RH ADDRESS IS NON EXISTENT.

ERRORS REPORTED EXCEPT IN BRUTUS:

1. TM = UNEXPECTED TAPE MARK FOUND

2. BOT = UNEXPECTED BEGINNING OF TAPE
3. EOT = TAPE IS POSITIONED BEYOND END OF TAPE MARKER.
4. LEOT = UNEXPECTED LOGICAL END OF TAPE.
5. FPT = A WRITE WAS ATTEMPTED ON A FILE PROTECTED TAPE.
6. NOT RDY = TU IS ON LINE BUT TAPE IS REWINDING OR LOADING.
7. NOT AVL = TU IS NOT SWITCHED TO THIS PORT, BUT IS ON LINE.
8. OFF LINE = TU IS NOT SWITCHED ON LINE WITH A TAPE LOADED.
9. NON EX = TU DOES NOT EXIST OR POWER IS OFF.
10. LONG REC = THE LAST RECORD WAS LONGER THAN THE BYTE COUNT.
11. SHORT REC = WAS SHORTER THEN THE BYTE COUNT.
12. RETRY = ERROR, THE INITIAL OPERATION SHOULD BE REPEATED.
13. READ OPP = READ ERROR, THE INITIAL READ SHOULD BE PERFORMED IN OPPOSITE DIRECTION.
14. UNREAD = READ RETRIES HAVE FAILED TO READ THE RECORD.
15. ERROR = AN ERROR HAS OCCURRED WHICH REQUIRED A RETRY BUT SUPPRESS ERROR REPOSITIONING IS SET.
16. EOT ERROR = WRITE ERROR HAS OCCURRED BEYOND THE EOT MARKER, AND SER IS SET.
17. BAD TAPE = TAPE POSITION HAS BEEN LOST, OR WRITE RETRIES HAVE FAILED TO WRITE THE RECORD.
18. TM FAULT A = THE HARDWARE HAS FAILED OR A SOFTWARE BUG HAS BEEN DETECTED.
19. TU FAULT A = TAPE UNIT HAS FAILED.
20. TIMEOUT = IF TU DOES NOT RESPOND IN A GIVEN AMOUNT OF TIME.
21. HARD WRITE ERROR = UNABLE TO CORRECTLY WRITE A RECORD.

22. MEDIA ERROR = A CORRECTABLE WRITE ERROR HAS OCCURRED.

23. DATA DOES NOT COMPARE CORRECTLY

24. MASS BUS STATUS ERRORS

4.2 SAMPLE ERROR PRINTOUT

-----  
CZTMJA DVC FTL ERR 00023 ON UNIT 00 TST 001 SUB 000 PC: 024134  
STATUS ERROR  
RH ADDRESS 172400 TM78# 0 TU78# 0  
ELAPSED TIME 0:05  
CS1 144260 CS2 000100  
ERROR OCCURRED WHILE EXECUTING COMMAND 61  
RECORD # 18 FILE # 1  
INTERRUPT CODE 25 FAILURE CODE 03  
PATTERN # 7 RECORD LENGTH 2048  
COMMAND CODE 061 INTERRUPT CODE 025 FAILURE CODE 003  
CH0TIE 000 RPFAL 002 RAMT 377 CASSTA 100 MTA0 250 WMCSTA 300  
CH1TIE 010 RPATH 046 RDON 000 CBUS 003 MTA1 002 WRDAT 000  
CH2TIE 000 RSTAT 000 RILL 002 DBUS 040 MTA2 126 DDRA 000  
CH3TIE 000 RCMLP 001 RMK2 000 TUSELO 000 MTA3 000 DDRB 000  
CH4TIE 000 CRCWRD 306 REND 000 TUSEL1 200 MTA4 001 WMCERR 004  
CH5TIE 000 ECCOR 377 RPSTA 201 TAMT 000 MTA5 240 INSTA 044  
CH6TIE 000 ECCSTA 221 RPOSTN 000 PSTAT 024 EXTRA 000 EXTRA 000  
CH7TIE 000 RTIER 010 RDATA 000 PRDD 000 TACH 231031030222  
CHPTIE 000 BYTCNT 377371 PADCNT 000000 ERRCNT 021351

NOTE: DESCRIPTIONS OF THE EXTENDED SENSE INFORMATION CAN  
BE FOUND IN THE 'TM78 TAPE CONTROLLER/FORMATTER'  
ENGINEERING SPECIFICATION.

SAMPLE DATA COMPARE ERROR PRINTOUT

CZTMJA DVC FTL ERR 00034 ON UNIT 00 TST 001 SUB 000 PC: 026650  
DATA COMPARE ERROR  
RH ADDRESS 172400 TM78# 0 TU78# 0  
ELAPSED TIME 0:21  
CS1 004276 CS2 000100  
ERROR OCCURRED WHILE EXECUTING COMMAND 77  
PATTERN # 8 RECORD LENGTH 2048  
RECORD # 29 FILE # 0  
BAD BYTE # 0 DATA EXPECTED 10000101 ACTUAL 10100110  
BAD BYTE # 1 DATA EXPECTED 01001001 ACTUAL 10011000  
BAD BYTE # 2 DATA EXPECTED 10100110 ACTUAL 01101101  
BAD BYTE # 3 DATA EXPECTED 10011000 ACTUAL 10000000  
BAD BYTE # 4 DATA EXPECTED 01101101 ACTUAL 10100001  
BAD BYTE # 5 DATA EXPECTED 10000000 ACTUAL 11101000  
BAD BYTE # 6 DATA EXPECTED 10100001 ACTUAL 00000000  
BAD BYTE # 7 DATA EXPECTED 11101000 ACTUAL 00000000  
NUMBER OF BYTES BAD IN A RECORD 1896

5.0 STATISTICAL REPORT

5.1 SAMPLE STATISTICAL PRINTOUT

STATISTICAL REPORT FOR UNIT # 0

RH ADDRESS 172400 TM78# 0 TU78# 0

TAPE DENSITY -- 1600 BPI (PE) OR -- 6250 BPI (GCR)

	WRITE	READ	READ REVERSE
STATUS ERRORS	0	0	0
NON-RECOVERABLE	0	0	0
BYTES TRANSFERRED	000004915	000004915	000004915
MEDIA ERRORS	--	--	--
DOUBLE TRACK CORRECTIONS	--	--	--
SINGLE TRACK CORRECTIONS	--	--	--
DATA COMPARE ERRORS	--	0	0

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	(TRACK7)	(TRACK6)	(TRACK5)	(TRACK3)	(TRACK9)	(TRACK1)	(TRACK8)	(TRACK2)
DROPS	0	0	0	0	0	0	0	0
PICKS	0	0	0	0	0	0	0	0

UNIT DROPPED 0

5.2 STATISTICAL UPDATES

WRITE

STATUS ERRORS           UPDATED ON EVERY WRITE THAT IS NOT PERFECT THE FIRST TRY AT WRITING BUT DOES NOT WRITE PERFECT ON THE SAME SPOT OF TAPE WHEN RETRIED OR WHEN RETRIES SUPRESSED.

MEDIA ERRORS            UPDATED ON ANY WRITE THAT IS RETRIED AFTER AN ERASE AND IS SUCCESSFUL.

NON-RECOVERABLE        UPDATE IF RETRIES FAIL TO WRITE THE RECORD PERFECTLY ON TAPE.

BYTES WRITTEN           UPDATED AFTER EVERY SUCCESSFUL WRITE OPERATION

READ + READ REVERSE

STATUS ERRORS           UPDATE ONCE FOR EVERY READ THAT IS RETRIED AND SUCCESSFUL. UPDATED ON ANY INTERRUPT

	BUT A DONE 0 OR 1.
NON-RECOVERABLE	UPDATED ON ANY READ UNABLE TO DO SO.
DOUBLE TRACK	UPDATED WHEN DOUBLE TRACK BIT HAS BEEN SET
SINGLE TRACK	UPDATED WHEN DOUBLE TRACK IS NOT SET AND SINGLE TRACK IS.
DATA COMPARE ERRORS	UPDATED WHEN DATA READ DOES NOT EXACTLY MATCH DATA WRITTEN
BYTES READ FORWARD AND REVERSE	UPDATED ON EVERY SUCCESSFUL READ OPERATION
DROPS AND PICKS	UPDATED ON DATA COMPARE ERRORS FOR BITS DROPPED OR PICKED IN EACH CHANNEL.
UNIT DROPPED	UPDATED EVERY TIME THIS UNIT WAS DROPPED REGARDLESS OF OTHER STATISTICS BEING UPDATED ON THE SAME ERROR.

```
1110 .TITLE PROGRAM HEADER AND TABLES
1111 .SBTTL PROGRAM HEADER
1145
1147 .ENABL ABS,AMA
1148 = 2000
1150
1151 002000 BGNMOD
1152
1153 :++
1154 : THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1155 : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1156 :--
1157
1158 002000 POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU
1159
1167
1168 002000 HEADER CZTMJA,0,0,600.,1,340
1169
1170 002122 DEVTYP <TM78,TU78>
1171
1172 002134 DESCRIPT <PERFORMANCE EXERCISER>
1173
1179 .SBTTL DISPATCH TABLE
1180
1181 :++
1182 : THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1183 : IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1184 :--
1185
1186 002162 DISPATCH 1
1187
1194 .SBTTL DEFAULT HARDWARE P-TABLE
1195
1196 :++
1197 : THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1198 : THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
1199 : IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
1200 :--
1201
1202 002166 BGNHW DFPTBL
1203
1204 002170 172400 .WORD 172400
1205 002172 000224 .WORD 224
1206 002174 000000 .WORD 0
1207 002176 000000 .WORD 0
1208 002200 000000 .WORD 0
1209
1215
1216 002202 ENDPHW
1217 .SBTTL SOFTWARE P-TABLE
1218
1219 :++
1220 : THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
1221 : PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
1222 :--
1223
```

			BGNSW	SFPTBL	
1224	002202				
1225					
1226	002204	000000	BRUTUS: .WORD	0	:BRUTUS OPTION
1227	002206	000002	CANSEQ: .WORD	2	:CANNED SEQUENCE NUMBER
1228	002210	000000	BRUERR: .WORD	0	:BRUTUS WITH ERRORS
1229	002212	000000	STAEOP: .WORD	0	:STATISTIC AT END OF PASS
1230	002214	000000	STAEOT: .WORD	0	:STATISTICS ON END OF TAPE
1231	002216	000000	RETRY: .WORD	0	:AUTOREPOSITION ON OR OFF
1232	002220	004000	CHAREC: .WORD	4000	:BYTES IN A RECORD
1233	002222	000000	PATTRN: .WORD	0	:PATTERN NUMBER
1234	002224	000000	RALNTH: .WORD	0	:RANDOM RECORD LENGTH
1235	002226	000000	CATCHK: .WORD	0	:CHECK THE DATA
1236	002230	000000	CMDSTR: .WORD	0	:LENGTH OF COMMAND STRING
1237	002232	000000	COMMD1: .WORD	0	:FIRST COMMAND
1238	002234	000000	REP1: .WORD	0	:REPITION OF ABOVE COMMAND
1239	002236	000000	COMMD2: .WORD	7	:SECOND COMMAND
1240	002240	000000	REP2: .WORD	0	:REPITION
1241	002242	000000	COMMD3: .WORD	0	:THIRD COMMAND
1242	002244	000000	REP3: .WORD	0	:REPITION
1243	002246	000000	COMMD4: .WORD	0	:FOURTH COMMAND
1244	002250	000000	REP4: .WORD	0	:REPITION
1245	002252	000000	COMMD5: .WORD	0	:FIFTH COMMAND
1246	002254	000000	REP5: .WORD	0	:REPITION
1247	002256	000000	COMMD6: .WORD	0	:SIXTH COMMAND
1248	002260	000000	REP6: .WORD	0	:REPITION
1249					
1256					
1257	002262		ENDSW		
1258					
1259	002262		ENDMOD		

1272  
1273  
1310  
1320  
1321 002262  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330 002262

.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

(1)		BIT15== 100000
(1)		BIT14== 40000
(1)		BIT13== 20000
(1)		BIT12== 10000
(1)		BIT11== 4000
(1)		BIT10== 2000
(1)		BIT09== 1000
(1)		BIT08== 400
(1)		BIT07== 200
(1)		BIT06== 100
(1)		BIT05== 40
(1)		BIT04== 20
(1)		BIT03== 10
(1)		BIT02== 4
(1)		BIT01== 2
(1)		BIT00== 1

(1)	100000	
(1)	040000	
(1)	020000	
(1)	010000	
(1)	004000	
(1)	002000	
(1)	001000	
(1)	000400	
(1)	000200	
(1)	000100	
(1)	000040	
(1)	000020	
(1)	000010	
(1)	000004	
(1)	000002	
(1)	000001	

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

(1)	000040	EF.START== 32.	: START COMMAND WAS ISSUED
(1)	000037	EF.RESTART== 31.	: RESTART COMMAND WAS ISSUED
(1)	000036	EF.CONTINUE== 30.	: CONTINUE COMMAND WAS ISSUED
(1)	000035	EF.NEW== 29.	: A NEW PASS HAS BEEN STARTED
(1)	000034	EF.PWR== 28.	: A POWER-FAIL/POWER-UP OCCURRED

:  
: PRIORITY LEVEL DEFINITIONS



```

(1)
(1)      000340      PRI07== 340
(1)      000300      PRI06== 300
(1)      000240      PRI05== 240
(1)      000200      PRI04== 200
(1)      000140      PRI03== 140
(1)      000100      PRI02== 100
(1)      000040      PRI01==  40
(1)      000000      PRI00==  0
(1)
(1)      ;OPERATOR FLAG BITS
(1)
(1)      000004      EVL==      4
(1)      000010      LOT==      10
(1)      000020      ADR==      20
(1)      000040      IDU==      40
(1)      000100      ISR==     100
(1)      000200      UAM==     200
(1)      000400      BOE==     400
(1)      001000      PNT==    1000
(1)      002000      PRI==    2000
(1)      004000      IXE==    4000
(1)      010000      IBE==   10000
(1)      020000      IER==   20000
(1)      040000      LOE==   40000
(1)      100000      HOE==  100000

```

```

1331
1332
1333      ;HARDWARE INFORMATION AND STATISTIC OFFSETS
1334
1335
1336
1337

```

```

1338      -000000      RHADD==0      ;RH ADDRESS
1339      000002      RHVEC==2      ;RH VECTOR ADDRESS
1340      000004      TMNUM==4      ;TM NUMBER
1341      000006      TUNUM==6      ;TU NUMBER
1342      000010      SPUNIT==10     ;SUPERVISOR UNIT #
1343      000012      SERWRT==12    ;STATUS WRITE ERRORS
1344      000014      SERRFO==14    ;STATUS READ ERRORS
1345      000016      SERRRV==16    ;STATUS READ REVERSE ERRORS
1346      000020      NRECWR==20    ;NON-RECOVERABLE WRITES
1347      000022      NRECRE==22    ;NON-RECOVERABLE READS
1348      000024      NRECRR==24    ;NON-RECOVERABLE READ REVERSES
1349      000026      BYWRHI==26    ;
1350      000030      BYWRMD==30    ;BYTES WRITTEN
1351      000032      BYWRLO==32    ;
1352      000034      BYRFHI==34    ;
1353      000036      BYRFMD==36    ;BYTES READ FORWARD
1354      000040      BYRFLO==40    ;
1355      000042      BYRRHI==42    ;
1356      000044      BYRRMD==44    ;BYTES READ REVERSE
1357      000046      BYRRLO==46    ;
1358      000050      MEDIAE==50     ;MEDIA ERRORS
1359      000052      DBLTRF==52    ;DOUBLE TRACK ERROR READ FORWARD
1360      000054      DBLTRR==54    ;DOUBLE TRACK ERROR READ REVERSE

```

1361	000056	SNGTRF==56	:SINGLE TRACK READ FORWARD
1362	000060	SNGTRR==60	:SINGLE TRACK READ REVERSE
1363	000062	DERREF==62	:DATA COMPARE ERROR FORWARD
1364	000064	DERRRV==64	:DATA COMPARE ERROR REVERSE
1365	000066	DROP7==66	:
1366	000070	DROP6==70	:
1367	000072	DROP5==72	:
1368	000074	DROP4==74	:BITS DROP PER TRACK
1369	000076	DROP3==76	:
1370	000100	DROP2==100	:
1371	000102	DROP1==102	:
1372	000104	DROP0==104	:
1373	000106	PICK7==106	:
1374	000110	PICK6==110	:
1375	000112	PICK5==112	:
1376	000114	PICK4==114	:
1377	000116	PICK3==116	:BITS PICK PER TRACK
1378	000120	PICK2==120	:
1379	000122	PICK1==122	:
1380	000124	PICK0==124	:
1381	000126	DRPUNT==126	:UNIT DROPPED
1382	000130	OTHWRT==130	:OTHER ERRORS WRITE
1383	000132	OTHRFO==132	:OTHER ERRORS READ FORWARD
1384	000134	OTHRRV==134	:OTHER ERRORS READ REVERSE

:EQUATES FOR COMMON ADDRESS SPACE (CAS)

1385			
1386			
1387			
1388			
1389			
1390			
1391			
1392			
1393			
1394			
1395			
1396			
1397	000000	BASE==0	
1398	000002	WRDCNT==2	
1399	000004	BSADDR==4	
1400	000006	BYTCNT==6	
1401	000010	UNSLCT==10	
1402	000012	INTCDE==12	
1403	000014	CMDADR==14	
1404	000016	ATTBIT==16	
1405	000020	DSEQ==20	
1406	000022	DATBUF==22	
1407	000024	DIAGTS==24	
1408	000026	DRVTYP==26	
1409	000030	SERIAL==30	
1410	000032	DIAGRQ==32	
1411	000034	ACTDGD==34	
1412	000036	NDTICD==36	
1413	000040	NDTFC0==40	
1414	000042	NDTFC1==42	
1415	000044	NDTFC2==44	
1416	000046	NDTFC3==46	

1417 000050  
1418 000052  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429 000003  
1430 000005  
1431 000007  
1432 000011  
1433 000013  
1434 000015  
1435 000017  
1436 000021  
1437 000023  
1438 000025  
1439 000027  
1440 000031  
1441 000033  
1442 000035  
1443 000037  
1444 000041  
1445 000043  
1446 000045  
1447 000047  
1448 000051  
1449 000057  
1450 000061  
1451 000063  
1452 000071  
1453 000073  
1454 000077  
1455  
1456

·  
·EQUATES FOR COMMANDS  
·

INTRAD==50  
INTRDA==52

NOOP==03  
UNLOAD==05  
REWIND==07  
SENSE==11  
DSE==13  
WTMPE==15  
WTMGCR==17  
SPFWRC==21  
SPRVRC==23  
SPFWFL==25  
SPRVFL==27  
SPFWEI==31  
SPRVEI==33  
ERGPE==35  
ERGGCR==37  
CLFLPE==41  
CLFGCR==43  
SPLEOT==45  
SFFLET==47  
WCKFWD==51  
WCKREV=57  
WRTPE==61  
WRTGCR==63  
REDFWD==71  
REDEXT==73  
REDREV==77

1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524

.SBTTL GLOBAL DATA SECTION

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

:\*\*\*\*\*  
: DATA USED ONLY FOR INITIALIZE MODULE  
:\*\*\*\*\*

MINSIZ:: .WORD 2000 ;SMALLEST AMOUNT OF FREE MEMORY NEEDED  
FETBL:: .WORD EMSG2 ;MESSAGES FOR INIT CODE  
          .WORD EMSG5  
          .WORD EMSG7  
          .WORD EMSG10  
          .WORD EMSG11

START1:: .WORD 0  
SERNUM:: .WORD 0 ;ASCII STORAGE OF SERIAL NUMBER  
          .WORD 0  
          .WORD 0

USEPAT:: .WORD 0 ;USER INPUT PATTERN FLAG  
REPCMD:: .WORD 0 ;NUMBER OF TIMES COMMAND WILL BE EXECUTED

:\*\*\*\*\*  
: DATA FLAGS USED THROUGHOUT ALL MODULES  
:\*\*\*\*\*

BUFSIZ:: .WORD 0 ;SIZE OF FREE MEMORY BUFFER IN BYTES  
BRTEST:: .WORD 0 ;BRUTUS AND BRUTUS W/ERRORS COMBINED  
DISPLC:: .WORD 0 ;CONTAINS ADDRESS OF SPECIFIED DISPLACEMENT IN TABLE  
ERRFLG:: .WORD 0 ;SPECIFIES AN ERROR AND TYPE OF ERROR  
HOURS:: .WORD 0 ;ELAPSED HOURS  
MINUTE:: .WORD 0 ;ELAPSED MINUTES  
INTLZE:: .WORD 0 ;INITIALZE FLAG TELLS IF WENT THRU INIT CODE  
PTBLAD:: .WORD 0 ;ADDRESS OF HARD P-TABLE  
REDBUF:: .WORD 0 ;ADDRESS OF READ BUFFER  
STRST:: .WORD 0 ;START OR RESTART FLAG  
STRTBL:: .WORD 0 ;VARIABLE NEED IN DISPLACEMENT SUBROUTINES  
CONTER: .WORD 0 ;USER INPUT COMMAND COUNTER  
STTBLH:: .WORD 124 ;LENGTH OF STAT TABLE

1525	002344	000000	TBLCNT:: .WORD	0	:NUMBER OF UNITS BEING TESTED
1526	002346	000136	TBLNTH:: .WORD	136	:LENGTH OF UNIT TABLE
1527	002350	000004	PATLOP:: .WORD	4	:NORMAL LENGTH OF PATTERN LOOP
1528	002352	000000	TBLNUM:: .WORD	0	:DUMMY TABLE COUNT
1529	002354	000000	VRMADD:: .WORD	0	:START OF FREE MEMORY
1530	002356	000000	RHADDR:: .WORD	0	:RH ADDRESS
1531	002360	000000	CSRCLK:: .WORD	0	:CSR CLOCK ADDRESS
1532	002362	000000	LCLOCK:: .WORD	0	:TICK COUNTER FOR CLOCK
1533	002364	000000	HERTZ:: .WORD	0	:HERTZ 60 OR 50
1534	002366	000000	PRIRTY:: .WORD	0	:PRIORITY OF CLOCK MINUS 1
1535	002370	000000	TM78N:: .WORD	0	:TM78 NUMBER
1536	002372	000000	TU78N:: .WORD	0	:TU78 NUMBER
1537	002374	000000	INTRPT:: .WORD	0	:INTERRUPT FLAG
1538	002376	000000	INTVEC:: .WORD	0	:INTERRUPT VECTOR
1539	002400	000000	RHADD1: .WORD	0	:TABLE STORES ALL THE RH ADDRESS
1540	002402	000000	RHADD2: .WORD	0	
1541	002404	000000	RHADD3: .WORD	0	
1542	002406	000000	RHADD4: .WORD	0	
1543	002410	000000	RHADD5: .WORD	0	
1544	002412	000000	RHADD6: .WORD	0	
1545	002414	000000	RHADD7: .WORD	0	
1546	002416	000000	RHADD8: .WORD	0	
1547	002420	000000	R8: .WORD	0	
1548	002422	000000	R9: .WORD	0	:PSEUDO REGISTERS
1549	002424	000000	R10: .WORD	0	
1550	002426	177777	GENFLG: .WORD	-1	:GENERATION FLAG
1551	002430	000000	BNRYNB: .WORD	0	:BINARY NUMBER
1552	002432	000000	BITPST: .WORD	0	:BIT POSITION NUMBER
1553	002434	000000	CHKBOT: .WORD	0	:CHECK BOT STATUS AFTER REWIND
1554	002436	000000	CHKUTM: .WORD	0	:CHECK FOR UNEXPECTED TAPE MARK
1555	002440	000000	ERASED: .WORD	0	:NOT CAPABLE FLAG FOR FUNCTIONAL SEQ.
1556	002442	000000	MESNUM: .WORD	0	:FUNCTIONAL TEST MESSAGE NUMBER
1557	002444	000000	DENFLG: .WORD	0	:STATISTIC TABLE DENSITY SELECT FLAG
1558	002446	000377	LNGDAT: .WORD	377	:DATA PATTERN FOR LONG RECORDS
1559	002450	000000	BIGREC: .WORD	0	:LARGE RECORD FLAG
1560					
1561					
1562					
1563					
1564					
1565					
1566					
1567					
1568					
1569					
1570					
1571					
1572					
1573	002452	000000	CMMAND:: .WORD	0	:HOLDS COMMAND TO BE EXECUTED
1574	002454	000000	EICMMD:: .WORD	0	:COMMAND W/ INTERRUPT ENABLE SET
1575	002456	000000	BUFFER:: .WORD	0	:BUFFER LENGTH
1576	002460	000000	BUFLST:: .WORD	0	:HOLD LAST BUFFER ADDRESS
1577	002462	000000	RETRCD:: .WORD	0	:RETRY COMMAND TO BE EXECUTED
1578	002464	000000	STAFLG:: .WORD	0	:STATISTIC PRINTOUT FLAG
1579	002466	000000	AUTORW:: .WORD	0	:AUTO REWIND FLAG
1580	002470	000000	VISIT:: .WORD	0	:RETRY VISIT FLAG

\*\*\*\*\*  
:DATA USED FOR TEST MODULES ONLY  
\*\*\*\*\*

1581	002472	000000	EXPECT:: .WORD	0	: EXPECTED DATA
1582	002474	000000	ACTUAL:: .WORD	0	: ACTUAL DATA
1583	002476	000000	HRDCNT:: .WORD	0	: HARD READ ERROR COUNT
1584	002500	000000	WTUFLG:: .WORD	0	: WRONG TU FLAG
1585	002502	000000	WTMFLG:: .WORD	0	: WRONG TM FLAG
1586	002504	000000	INTROK:: .WORD	0	: INTERRUPT OK
1587	002506	000000	TAPMON:: .WORD	0	: TM78 NUMBER
1588	002510	000000	DENSWT:: .WORD	0	: DENSITY SWITCH FLAG
1589	002512	000000	DTCOMP:: .WORD	0	: DATA COMPARE FLAG
1590	002514	000000	ERAFLG:: .WORD	0	: ERASE FLAG FOR MEDIA ERROR
1591	002516	000000	DATRAN:: .WORD	0	: DATA TRANSFER FLAG
1592	002520	000000	STORE:: .WORD	0	:
1593	002522	000000	EXTERR:: .WORD	0	: EXTENDED ERROR CODE
1594	002524	000000	EXTDSN:: .WORD	0	: EXTENDED SENSE FLAG
1595	002526	000000	ICCODE:: .WORD	0	: INTERRUPT CODE
1596	002530	000000	UNITFL:: .WORD	0	: UNIT FLAG
1597	002532	000000	MASBUS:: .WORD	0	: RH ADDRESS
1598	002534	000000	DRPFLG:: .WORD	0	: DROP UNITS (1-8) FLAG
1599	002536	000000	PATCNT:: .WORD	0	: PATTERN COUNTER
1600	002540	000000	EOTFLG:: .WORD	0	: ALL UNITS AT END OF TAPE
1601	002542	000000	TIMOUT:: .WORD	0	: TIME OUT
1602	002544	000000	TIMER:: .WORD	0	: TIME OUT
1603	002546	000000	BADCNT:: .WORD	0	: BAD WRITE RETRY COUNT
1604	002550	000000	GCRFLG:: .WORD	0	: VISIT FLAG TO GCR ROUTINE
1605	002552	000000	GCRBIT:: .WORD	0	: ROTATION BIT STORAGE
1606	002554	000000	OKAY:: .WORD	0	: COMPARE FLAG
1607	002556	000000	WRTCNT:: .WORD	0	: WRITE RETRY COUNT
1608	002560	000000	WRTYCT:: .WORD	0	: WRITE RETRY COUNT
1609	002562	000000	ALLINT:: .WORD	0	: UNIT INTERRUPT FLAGS
1610	002564	000000	RECTBL:: .WORD	0	: MORE THAN ONE UNIT FLAG
1611	002566	000000	RECORD:: .WORD	0	:
1612	002570	000000	FILENM:: .WORD	0	:
1613	002572	000000	FIRBYT:: .WORD	0	:
1614	002574	000000	GCCRPT:: .WORD	0	: GCR LOOP COUNTER
1615	002576	000000	GRCRCNT:: .WORD	0	: BIT PATTERN COUNTER
1616	002600	000000	ALLEOT:: .WORD	0	: ALL UNITS AT EOT
1617	002602	000000	REPITN:: .WORD	0	: # OF TIMES COMMAND IS REPEATED
1618	002604	000000	RTYFLG:: .WORD	0	: RETRY FLAG
1619	002606	000000	SENFLG:: .WORD	0	: SENSE READ FLAG
1620	002610	000000	RWDCNT:: .WORD	0	: REWIND COMMAND COUNTER
1621	002612	000000	RWDFLG:: .WORD	0	: REWIND FLAG
1622	002614	000020	PARMAX:: .WORD	20	: MAXIMUM PARITY ERROR ALLOWED PER UNIT
1623	002616	000000	CMDSTG:: .WORD	0	: STORAGE FOR COMMAND
1624	002620	000000	COUNT:: .WORD	0	: COUNTER FOR NUMBER OF COMMANDS
1625	002622	000377	RPTFLG:: .WORD	377	: STATISTICAL REPORT FLAG
1626	002624	000252	BYTEA:: .WORD	252	: GCR BIT PATTERN A
1627	002626	000125	BYTEB:: .WORD	125	: GCR BIT PATTERN B
1628	002630	000011	GCRLOP:: .WORD	11	: GCR LOOP COUNTER MAXIMUM
1629	002632	153624	RANB:: .WORD	153624	: RANDOM NUMBER GENERATOR BASE
1630	002634	032561	RANS:: .WORD	32561	: RANDOM # SAVE LOCTION
1631	002636	153624	RANB2:: .WORD	153624	:
1632	002640	032561	RANS2:: .WORD	32561	:
1633					:
1634					:
1635					:
1636					:

1637  
1638  
1639  
1640 002642 000000  
1641 002644 000000  
1642 002646 000000  
1643 002650 000000  
1644 002652 000000  
1645 002654 000000  
1646 002656 000000  
1647 002660 000000

:PARITY ERROR COUNTERS

```

TRECNT:: .WORD 0
          .WORD 0
          .WORD 0
          .WORD 0
          .WORD 0
          .WORD 0
          .WORD 0
          .WORD 0
  
```

1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656

: INTERRUPT CODE RESPONSE LOCATIONS

1657 002662 030602  
1658 002664 030754  
1659 002666 032320  
1660 002670 032460  
1661 002672 032540  
1662 002674 033000  
1663 002676 033122  
1664 002700 033052  
1665 002702 033054  
1666 002704 033124  
1667 002706 033152  
1668 002710 033204  
1669 002712 033232  
1670 002714 033260  
1671 002716 033324  
1672 002720 033352  
1673 002722 033374  
1674 002724 033374  
1675 002726 033670  
1676 002730 033732  
1677 002732 033774  
1678 002734 034050  
1679 002736 034262  
1680 002740 034350  
1681 002742 034416  
1682 002744 034464  
1683 002746 034540  
1684 002750 034516  
1685 002752 034540

```

INTTBL: .WORD MBTERM
         .WORD DONE
         .WORD TM
         .WORD BOT
         .WORD EOT
         .WORD LEOT
         .WORD INOOP
         .WORD IREWND
         .WORD FPT
         .WORD NOTRDY
         .WORD NOTAVL
         .WORD OFFLNE
         .WORD NONEX
         .WORD BLKTAP
         .WORD ILLINT
         .WORD ONLINE
         .WORD LONREC
         .WORD SHTREC
         .WORD IRETRY
         .WORD READOP
         .WORD UNRDAB
         .WORD ERROR
         .WORD EOTERR
         .WORD BADTAP
         .WORD TMFALT
         .WORD TUFALT
         .WORD TMFALB
         .WORD TUFALB
         .WORD MBFALT
  
```

```

:MASS BUS TERMINATION INTERRUPT
:COMMAND FINISHED WITHOUT ERROR
:UNEXPECTED TAPE MARK
:UNEXPECTED BEGINNING OF TAPE
:SUCCESSFUL WRITE AFTER EOT
:UNEXPECTED LOGICAL EOT
:NO-OP COMPLETED
:FIRST INTERRUPT FROM REWIND
:WRITE ATTEMPTED ON FILE PROTECT TAPE
:TU IS NOT READY
:TU IS NOT SWITCHED TO THIS PORT
:TU IS NOT ON LINE
:TU DOES NOT EXIST
:BLANK TAPE OR CANNOT FIND RECORD
:ILLEGAL INTERRUPT
:A TAPE UNIT HAS COME ON LINE
:RECORD WRITTEN BUT TOO LONG
:RECORD WRITTEN BUT TOO SHORT
:OPERATION SHOULD BE REPEATED
:READ SHOULD BE PERFORMED IN OPPOSITE DIRECTION
:RETRIES HAVE FAILED TO READ THE RECORD
:ERROR HAS OCCURRED BUT SER FLAG SET
:ERROR OCCURRED BEYOND EOT
:TAPE POSITION HAS BEEN LOST
:TM HARDWARE FAILURE - COMMAND INITIATED
:TU HARDWARE FAILURE - COMMAND INITIATED
:TM HARDWARE FAILURE - TM INITIATED
:TU HARDWARE FAILURE
:MASS BUS ERROR - TM INITIATED
  
```

1686  
1687  
1688  
1689  
1690  
1691  
1692

: SEQUENCE SELECT TABLE

```

1693
1694
1695 002754 020174      SEQSEL:: .WORD  BRTSEQ
1696 002756 020302      .WORD  UNTAPE
1697 002760 020452      .WORD  FIELD1
1698 002762 020770      .WORD  FIELD2
1699 002764 021454      .WORD  FUNCT
1700
1701
1702      ;
1703      ; COMMANDS FOR FIELD1
1704      ;
1705
1706 002766 000061      CANA1: .WORD  WRTPE
1707 002770 000077      .WORD  REDREV
1708 002772 000071      .WORD  REDFWD
1709 002774 000000      .WORD  0
1710
1711 002776 000063      CANB1: .WORD  WRTGCR
1712 003000 000077      .WORD  REDREV
1713 003002 000071      .WORD  REDFWD
1714
1715
1716      ;
1717      ; COMMANDS FOR FIELD2 AND FIELD3
1718      ;
1719
1720
1721 003004 000061      CANA2: .WORD  WRTPE
1722 003006 000015      .WORD  WTMPE
1723 003010 000027      .WORD  SPRVFL
1724 003012 000077      .WORD  REDREV
1725 003014 000071      .WORD  REDFWD
1726 003016 000025      .WORD  SPFWFL
1727 003020 000000      .WORD  0
1728
1729 003022 000063      CANB2: .WORD  WRTGCR
1730 003024 000017      .WORD  WTMGCR
1731 003026 000027      .WORD  SPRVFL
1732 003030 000077      .WORD  REDREV
1733 003032 000071      .WORD  REDFWD
1734 003034 000025      .WORD  SPFWFL
1735
1736
1737      ;
1738      ; COMMANDS FOR FUNCTIONAL SEQUENCE
1739      ;
1740
1741 003036 021556      FCNTBL: F.NOP      ;NOOP TEST
1742 003040 021606      F.REW      ;REWIND TEST
1743 003042 021632      F.SNS      ;SENSE TO CHECK FOR BOT
1744 003044 023532      F.ER1      ;ERASE TO MOVE TAPE
1745 003046 021606      F.REW      ;REWIND
1746 003050 021632      F.SNS
1747
1748      ; WRITE READ TEST

```



1749				
1750	003052	021672	F.WPE	:WRITE PE
1751	003054	022010	F.RDR1	:READ REVERSE
1752	003056	022020	F.RDF1	:READ FORWARD
1753	003060	021606	F.REW	:REWIND
1754	003062	021730	F.WGCR	:WRITE GCR
1755	003064	022052	F.RDR2	:READ REVERSE
1756	003066	022062	F.RDF2	:READ FORWARD
1757				
1758			:	WRITE CHECK FUNCTIONAL TESTS
1759				
1760	003070	021606	F.WC:	F.REW :REWIND
1761	003072	022114		F.WC1 :WRITE A RECORD
1762	003074	022202		F.WC2 :WRITE CHECK REV
1763	003076	022226		F.WC3 :WRITE CHECK FWD
1764				
1765			:	SPACING FUNCTIONAL TESTS
1766				
1767	003100	022306	F.SP:	F.SP1 :REWIND
1768	003102	022330		F.SP2 :FORMAT TAPE
1769	003104	022450	F.SPRV:	F.SP3 :SPACE REVERSE
1770	003106	022474		F.SP4 :READ REVERSE
1771	003110	022526		F.SP5 :SPACE FORWARD
1772	003112	022560	F.SPND:	F.SP6 :READ FORWARD
1773				
1774			:	TAPE MARK WRITE/READ FUNCTIONAL TEST
1775				
1776	003114	021606	F.REW	:REWIND
1777	003116	022674	F.TM1	:WRITE A TM
1778	003120	022770	F.TM2	:SPACE REV
1779	003122	023014	F.TM3	:SPACE FORWARD
1780				
1781			:	TAPE MARK SPACE TEST
1782				
1783	003124	023066	F.TMS:	F.TMS1 :REWIND TAPE
1784	003126	023102		F.TMS2 :WRITE A TM
1785	003130	023110		F.TMS3 :WRITE RECORDS
1786	003132	023204		F.TMS4 :SPACE
1787				
1788			:	VACUUM COLUMNS WORST CASE TEST
1789				
1790	003134	021606	F.VC:	F.REW :REWIND
1791	003136	023276		F.VC1 :FORMAT TAPE
1792	003140	021606		F.REW :REWIND TAPE
1793	003142	023370		F.VC2 :SPACE WITH INCREMENTING RECORD COUNT
1794	003144	021606		F.REW :REWIND
1795	003146	023440		F.VC3 :SPACE WITH DECREMENTING RECORD COUNT
1796				
1797			:	ERASE FUNCTIONAL TEST
1798				
1799	003150	021606	F.ER:	F.REW :REWIND TAPE
1800	003152	023532		F.ER1 :ERASE 40'
1801	003154	021606		F.REW :REWIND TAPE
1802	003156	023612		F.ER2 :ATTEMPT TO READ
1803	003160	023642		F.ER3 :CHECK FOR NOT CAPABLE
1804				

```
1805 ; SHORT AND LONG RECORD TEST
1806
1807 003162 023722 F.SL: F.SH1 ;REWIND
1808 003164 023752 F.SHRT: F.SH2 ;WRITE
1809 003166 024046 F.SH3 ;READ REV
1810 003170 024100 F.SH4 ;READ FORWARD
1811
1812 ; LONG RECORDS
1813
1814 003172 024156 F.LNG1 ;REWIND
1815 003174 024202 F.LNG2 ;WRITE
1816 003176 024276 F.LNG3 ;READ REV
1817 003200 024330 F.LNG4 ;READ FWD
1818
1819 ; CAPSTAN WORST CASE TEST
1820
1821 003202 021606 F.REW ;REWIND TAPE
1822 003204 024434 F.CP1 ;MOVE TAPE 400 '
1823 003206 024472 F.CP2 ;DELAY
1824
1825 003210 024550 F.END ;END OF FUNCTIONAL TEST
1826
1827
1828 ;
1829 ;RETRY SELECT TABLE
1830 ;
1831
1832
1833 003212 000000 RTYSEQ: .WORD 0
1834 003214 024564 .WORD RTRY01
1835 003216 024642 .WORD RTRY02
1836 003220 024652 .WORD RTRY03
1837 003222 024702 .WORD RTRY04
1838 003224 025030 .WORD RTRY05
1839 003226 025064 .WORD RTRY06
1840
1841
1842 ;
1843 ;EXTENDED SENSE TABLE
1844 ;
1845
1846
1847 003230 000 COMCOD: .BYTE 0
1848 003231 000 INTCOD: .BYTE 0
1849 003232 000 FAILCD: .BYTE 0
1850 003233 000 RPFAIL: .BYTE 0
1851 003234 000 RPATH: .BYTE 0
1852 003235 000 RSTAT: .BYTE 0
1853 003236 000 RCMLP: .BYTE 0
1854 003237 000 RAMT: .BYTE 0
1855 003240 000 RDON: .BYTE 0
1856 003241 000 RILL: .BYTE 0
1857 003242 000 RMK2: .BYTE 0
1858 003243 000 REND: .BYTE 0
1859 003244 000 RPSTA: .BYTE 0
1860 003245 000 RPOSTN: .BYTE 0
```

1861	003246	000	RDATA:	.BYTE	0
1862	003247	000	CRCWRD:	.BYTE	0
1863	003250	000	ECCCOR:	.BYTE	0
1864	003251	000	ECCSTA:	.BYTE	0
1865	003252	000	CHOTIE:	.BYTE	0
1866	003253	000	CH1TIE:	.BYTE	0
1867	003254	000	CH2TIE:	.BYTE	0
1868	003255	000	CH3TIE:	.BYTE	0
1869	003256	000	CH4TIE:	.BYTE	0
1870	003257	000	CH5TIE:	.BYTE	0
1871	003260	000	CH6TIE:	.BYTE	0
1872	003261	000	CH7TIE:	.BYTE	0
1873	003262	000	CHPTIE:	.BYTE	0
1874	003263	000	RTIER:	.BYTE	0
1875	003264	000	TAMT:	.BYTE	0
1876	003265	000	PSTAT:	.BYTE	0
1877	003266	000	PRDD:	.BYTE	0
1878	003267	000	CASSTA:	.BYTE	0
1879	003270	000	CBUS:	.BYTE	0
1880	003271	000	DBUS:	.BYTE	0
1881	003272	000	WMCSTA:	.BYTE	0
1882	003273	000	TUSELO:	.BYTE	0
1883	003274	000	TUSEL1:	.BYTE	0
1884	003275	000	WRDAT:	.BYTE	0
1885	003276	000	BCNTLO:	.BYTE	0
1886	003277	000	BCNTHI:	.BYTE	0
1887	003300	000	PDCNLO:	.BYTE	0
1888	003301	000	PDCNHI:	.BYTE	0
1889	003302	000	ERCNLO:	.BYTE	0
1890	003303	000	ERCNHI:	.BYTE	0
1891	003304	000	DDRA:	.BYTE	0
1892	003305	000	DDRB:	.BYTE	0
1893	003306	000	WMCERR:	.BYTE	0
1894	003307	000	INSTA:	.BYTE	0
1895	003310	000	MTA0:	.BYTE	0
1896	003311	000	MTA1:	.BYTE	0
1897	003312	000	MTA2:	.BYTE	0
1898	003313	000	MTA3:	.BYTE	0
1899	003314	000	MTA4:	.BYTE	0
1900	003315	000	MTA5:	.BYTE	0
1901	003316	000	RETCT:	.BYTE	0
1902	003317	000	RETCT1:	.BYTE	0
1903	003320	000	TUX:	.BYTE	0
1904	003321	000	XFRCTL:	.BYTE	0
1905	003322	000	XRETRY:	.BYTE	0
1906	003323	000	ENAON:	.BYTE	0
1907				.EVEN	
1908					
1909					
1910					
1911					
1912	003324	003350			
1913	003326	003360			
1914	003330	003370			
1915	003332	003400			
1916	003334	003410			

;  
;PATTERN GENERATION ROUTINES  
;  
PATGEN:: .WORD PATZER  
          .WORD PATONE  
          .WORD PATTWO  
          .WORD PATTHE  
          .WORD PATFOR

1917	003336	003420	.WORD	PATFIV
1918	003340	003430	.WORD	PATSIX
1919	003342	003464	.WORD	PATSVN
1920	003344	003466	.WORD	PATEGT
1921	003346	003476	.WORD	PATNIN

1922  
1923  
1924  
1925  
1926  
1927

:  
: PATTERNS TO BE GENERATED  
:

1929	003350	000000	PATZER: .WORD	0
1930	003352	000000	.WORD	0
1931	003354	000000	.WORD	0
1932	003356	000000	.WORD	0

1934	003360	177777	PATONE: .WORD	177777
1935	003362	177777	.WORD	177777
1936	003364	177777	.WORD	177777
1937	003366	177777	.WORD	177777

1939	003370	000377	PATTWO: .WORD	377
1940	003372	000377	.WORD	377
1941	003374	000377	.WORD	377
1942	003376	000377	.WORD	377

1944	003400	177777	PATTHE: .WORD	177777
1945	003402	000000	.WORD	0
1946	003404	177777	.WORD	177777
1947	003406	000000	.WORD	0

1949	003410	177777	PATFOR: .WORD	177777
1950	003412	177777	.WORD	177777
1951	003414	000000	.WORD	0
1952	003416	000000	.WORD	0

1954	003420	000000	PATFIV: .WORD	0
1955	003422	177400	.WORD	177400
1956	003424	000000	.WORD	0
1957	003426	177400	.WORD	177400

1959	003430	000000	PATSIX: .WORD	0
1960	003432	000000	.WORD	0
1961	003434	000000	.WORD	0
1962	003436	000000	.WORD	0

1963	003440	000000	.WORD	0
1964	003442	000000	.WORD	0
1965	003444	000000	.WORD	0
1966	003446	000000	.WORD	0

1967	003450	000000	.WORD	0
1968	003452	000000	.WORD	0
1969	003454	000000	.WORD	0
1970	003456	000000	.WORD	0
1971	003460	000000	.WORD	0
1972	003462	000000	.WORD	0

1973				
1974	003464	000400	PATSVN: .WORD	400
1975				
1976				
1977	003466	000000	PATEGT: .WORD	0
1978	003470	000000	.WORD	0
1979	003472	000000	.WORD	0
1980	003474	000000	.WORD	0
1981				
1982				
1983	003476	000000	PATNIN: .WORD	0
1984	003500	000000	.WORD	0
1985	003502	000000	.WORD	0
1986	003504	000000	.WORD	0
1987				
1988				
1989				
1999				
2000				

2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064

.SBTTL GLOBAL TEXT SECTION

.NLIST BEX

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

: THE FOLLOWING MESSAGES ARE FOR THE HARDWARE P-TABLE  
:

003506	044122	040440	042104	MSG1::	.ASCIZ	/RH ADDRESS/
003521	126	041505	047524	MSG2::	.ASCIZ	/VECTOR ADDRESS/
003540	046524	034067	021440	MSG3::	.ASCIZ	/TM78 #/
003547	124	033525	020070	MSG4::	.ASCIZ	/TU78 #/

: THE FOLLOWING MESSAGES ARE FOR THE SOFTWARE P-TABLE  
:

003556	051102	052125	051525	MSG5::	.ASCIZ	/BRUTUS OPTION/
003574	040503	047116	042105	MSG6::	.ASCIZ	/CANNED SEQUENCE # (1-4)/

: THE FOLLOWING MESSAGES ARE FOR THE SOFTWARE P-TABLE AND BRUTUS  
:

003624	052123	052101	020123	MSG7::	.ASCIZ	/STATS PRINTED AT EOT/
003651	102	052522	052524	MSG9::	.ASCIZ	/BRUTUS TO REPORT ERRORS/
003701	123	040524	051524	MSG10::	.ASCIZ	/STATS PRINTED AT EOP/

: THE FOLLOWING MESSAGES ARE FOR THE BRUTUS WITH ERRORS OPTION  
:

003726	040522	042116	046517	MSG11::	.ASCIZ	/RANDOM RECORD LENGTH/
003753	122	052105	044522	MSG12::	.ASCIZ	/RETRIES/

GLOBAL AREAS  
ZTMJA3.P11

MACY11 30(1046)  
02-JUL-80 11:53

02-JUL-80 15:09 PAGE 3-15  
GLOBAL TEXT SECTION

M 3

SEQ 0038

2065	003763	043	047440	020106	MSG13:: .ASCIZ	/# OF BYTES IN A RECORD (1,65535)/
2066	004024	040520	052124	051105	MSG14:: .ASCIZ	/PATTERN NUMBER (0-9)/
2067	004051	043	047440	020106	MSG15:: .ASCIZ	/# OF RECORDS IN A FILE (1,65535)/
2068	004112	040504	040524	041440	MSG17:: .ASCIZ	/DATA COMPARE/
2069	004127	110	053517	046440	MSG19:: .ASCIZ	/HOW MANY CMDS IN THE STRING (1-6) /
2070	004173	122	050105	052111	MSG20:: .ASCIZ	/REPITITION OF CMD (1,65535)/
2071	004227	105	052116	051105	MSG21:: .ASCIZ	/ENTER COMMAND/
2072	004245	045	022516	020101	INPATT: .ASCIZ	/%N% INPUT 14 WORDS OF YOUR OWN PATTERN/
2073	004315	077	000		MSG8:: .ASCIZ	/?/
2074						
2075	004320				.EVEN	
2076						
2077						
2078						
2084						
2091						
2092						

.SBTTL GLOBAL ERROR REPORT SECTION

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

: ERROR MESSAGES USED THROUGH OUT TEST MODULES  
:

2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2161

004320  
004320 022737 000004 002206  
004326 001016  
004330  
004332 013702 002442  
004336  
004362  
004364  
004420  
004450  
004504  
004530 004737 011404  
004534  
004536  
004536 022737 000004 002206  
004544 001016  
004546  
004550 013702 002442  
004554  
004600  
004602 022737 000032 002526  
004610 101015  
004612  
004642 000416  
004644  
004700  
004730 004737 011404  
004734  
004736  
004736 022737 000004 002206  
004744 001016  
004746  
004750 013702 014670  
004754  
005000  
005002 004737 011640  
005006

BGNMSG ERRMS1  
CMP #4,CANSEQ ;FUNCTIONAL SEQUENCE ?  
BNE 1\$ ;NO  
PUSH R2  
MOV MESNUM,R2  
PRINTB #FMTFCT,MESADR(R2)  
POP R2  
1\$: PRINTB #FORM01,RHADDR, TM78N, TU78N  
PRINTB #FORM23,HOURS, MINUTE  
PRINTB #FORM02,RHADD(R1),UNSLCT(R1),CMDADR(R1)  
PRINTB #FORM03,CMMAND  
JSR PC,HEDER  
ENDMSG  
BGNMSG ERRMS2  
CMP #4,CANSEQ ;FUNCTIONAL SEQ ?  
BNE 2\$ ;NO  
PUSH R2  
MOV MESNUM,R2 ;GET WHICH TEST  
PRINTB #FMTFCT,MESADR(R2) ;PRINT FAILING TEST MESSAGE  
POP R2  
2\$: CMP #32,ICCODE ;SEE IF FAULT B  
BHI 5\$ ;BRANCH IF NOT  
PRINTB #FORM27,RHADDR, TM78N  
BR 6\$  
5\$: PRINTB #FORM04,RHADDR, TM78N, TU78N  
6\$: PRINTB #FORM23,HOURS, MINUTE  
JSR PC,HEDER  
ENDMSG  
BGNMSG ERRMS3  
CMP #4,CANSEQ ;FUNCTIONAL SEQUENCE  
BNE 10\$ ;NO  
PUSH R2 ;SAVE R2  
MOV MESADR,R2  
PRINTB #FMTFCT,MESADR(R2) ;PRINT FAILING TEST  
POP R2  
10\$: JSR PC,HEDER3  
ENDMSG



2167  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228

:  
: FORMAT STATEMENTS USED IN PRINTF CALLS IN THE INIT CODE  
:

005010	047045	040445	047503	RANDOM:	.ASCIZ	/XN%ACONSECUTIVE WRITES FOLLOWED BY CONSECUTIVE READS WILL FAIL DATA COM
005126	047045	040445	052040	SERLNB:	.ASCII	/XN% THE SERIAL # FOR TU# %D1% ON TM78# %D1% ON RHADD %06%/
005221	045	022516	020101		.ASCIZ	/XN% IS %T/
005234	047045	040445	052502	ERRBUF:	.ASCIZ	/XN%ABUFFER REQUESTED TOO LARGE SUBSTITUTED LARGEST AVAILABLE %D5/
005335	045	022516	044501	EMSG1:	.ASCIZ	/XN%AINSUFFICIENT MEMORY AVAILABLE DIAGNOSTIC CAN NOT RUN/
005426	047045	040445	044124	EMSG2:	.ASCII	/XN%ATHIS TU# %D1% ON TM78# %D1% ON RHADD %06%/
005507	045	022516	044501		.ASCIZ	/XN%AIS NOT A TM78 TYPE DRIVE./
005545	045	022516	052101	EMSG3:	.ASCIZ	/XN%ATOO MANY UNITS TO BE TESTED UNDER BRUTUS, ONLY ONE ALLOWED/
005644	047045	040445	040525	EMSG4:	.ASCIZ	/XN%AUAM FLAG IS SET. MUST BE CLEARED IN ORDER TO RUN BRUTUS OPTION/
005747	045	022516	052101	EMSG5:	.ASCII	/XN%ATHIS TU# %D1% ON TM78# %D1% ON RHADD %06%/
006027	045	022516	042101		.ASCIZ	/XN%ADID NOT RESPOND TO SENSE CMD./
006071	045	022516	052101	EMSG6:	.ASCIZ	/XN%ATOO MANY UNITS TO TEST UNKNOWN TAPE, ONLY ALLOWED ONE/
006163	045	022516	052101	EMSG7:	.ASCII	/XN%ATHIS TU# %D1% ON TM78# %D1% ON RHADD %06%/
006243	045	022516	044101		.ASCIZ	/XN%AHAS GENERATED %06% INTRPT./
006304	047045	040445	044124	EMSG10:	.ASCIZ	/XN%ATHIS TU# %D1% ON TM78# %D1% ON RHADD %06% IS NON EXISTENT./
006412	047045	040445	044124	EMSG11:	.ASCII	/XN%ATHIS TM78# %D1% ON RHADD %06% IS NOT READY/
006472	047045	040445	042522		.ASCIZ	/XN%AREG #52 = %06%/
					.EVEN	

:  
: FORMAT STATEMENTS USED IN ERROR MESSAGES  
:

006516	040445	044122	040440	FORM01:	.ASCIZ	/XARH ADDRESS %06% TM78# %D1% TU78# %D1%/
006576	040445	051503	020061	FORM02:	.ASCIZ	/XACS1 %06% CS2 %06% CMDADR %06%/
006653	045	042501	051122	FORM03:	.ASCIZ	/XAERROR OCCURRED WHILE EXECUTING COMMAND %02%/
006733	045	051101	020110	FORM04:	.ASCIZ	/XARH ADDRESS %06% TM78# %D1% TU78# %D1%/
007013	045	044501	052116	FORM05:	.ASCIZ	/XAINTERRUPT CODE %02% FAILURE CODE %02%/
007066	040445	040502	020104	FORM06:	.ASCIZ	/XABAD BYTE # %D5% DATA EXPECTED %B8% ACTUAL %B8%/
007157	045	022516	052501	FORM08:	.ASCIZ	/XN%UNIT DROPPED%/
007202	040445	042522	047503	FORM07:	.ASCIZ	/XARECORD # %D5% FILE # %D5%/
007243	045	050101	052101	FORM09:	.ASCIZ	/XAPATTERN # %D2% RECORD LENGTH %D5%/
007314	040445	052516	041115	FORM10:	.ASCIZ	/XANUMBER OF BYTES BAD IN A RECORD %D5%/
007365	045	020101	047503	FORM11:	.ASCIZ	/XA COMMAND CODE %03% INTERRUPT CODE %03% FAILURE CODE %03%/
007463	045	041501	030110	FORM12:	.ASCIZ	/XACH0TIE %03% RPFAL %03% RAMT %03% CASSTA %03% MTA0 %03% WMCS
007603	045	041501	030510	FORM13:	.ASCIZ	/XACH1TIE %03% RPATH %03% RDON %03% CBUS %03% MTA1 %03% WRTD
007723	045	041501	031110	FORM14:	.ASCIZ	/XACH2TIE %03% RSTAT %03% RILL %03% DBUS %03% MTA2 %03% DDRB
010043	045	041501	031510	FORM15:	.ASCIZ	/XACH3TIE %03% RCMLP %03% RMK2 %03% TUSELO %03% MTA3 %03% DDRB
010163	045	041501	032110	FORM16:	.ASCIZ	/XACH4TIE %03% CRCWRD %03% REND %03% TUSEL1 %03% MTA4 %03% WMCE
010303	045	041501	032510	FORM17:	.ASCIZ	/XACH5TIE %03% ECCCOR %03% RPSTA %03% TAMT %03% MTA5 %03% INTS
010423	045	041501	033110	FORM18:	.ASCIZ	/XACH6TIE %03% ECCSTA %03% RPOSTN %03% PSTAT %03% RETCT %03% RETC
010543	045	041501	033510	FORM19:	.ASCIZ	/XACH7TIE %03% RTIER %03% RDATA %03% PRDD %03% TUX %03% XFRC
010663	045	041501	050110	FORM20:	.ASCIZ	/XACHPTIE %03% XRETRY %03% ENAON %03%/
010734	040445	042522	042101	FORM21:	.ASCIZ	/XAREAD EXTENDED SENSE FAILED%/
010773	045	022516	051101	FORM22:	.ASCIZ	/XN%ARETRY SUCCESSFUL%/
011022	040445	046105	051501	FORM23:	.ASCIZ	/XAELASPED TIME %D3%: %Z2%/
011055	045	053501	051117	FORM24:	.ASCIZ	/XAWORD COUNT %06%/

```

2229 011101 045 047101 020117 FORM25: .ASCIZ /%ANO L CLOCK AVAILABLE%/
2230 011132 040445 047520 042527 FORM26: .ASCIZ /%APOWER FAILURE%/
2231 011154 040445 044122 040440 FORM27: .ASCIZ /%ARH ADDRESS %06% TM78# %D1%/
2232 011213 045 022516 052101 FORM28: .ASCIZ /%N%ATAPE IS POSITIONED BEYOND EOT MARKER%/
2233 011266 040445 054502 041524 FORM29: .ASCIZ /%ABYTCNT %06% PADCNT %06% ERRCNT %06%/
2234 011344 040445 043133 047125 FMTFCT: .ASCIZ /%A[FUNCTIONAL SEQUENCE -- %T%/
2235 011404 011404 .EVEN
2236
2237 011404 005037 002606 HEDER: CLR SENFLG ;SET UP EXTENDED SENSE FLAG
2238 011410 005737 002542 TST TIMEOUT ;SEE IF UNIT TIMED OUT
2239 011414 001110 BNE 1$ ;BRANCH IF SO
2240 011416 005737 002206 TST CANSEQ ;SEE IF CANNED SEQUENCE
2241 011422 001420 BEQ 4$ ;BRANCH IF NOT
2242 011424 022737 000007 002452 CMP #REWIND,CMMAND
2243 011432 001414 BEQ 4$
2244 011434 PRINTB #FORM07,RECORD,FILENM
2245 011464 4$: PRINTB #FORM05,ICCODE,EXTERR
2246 011514 022737 000001 002206 CMP #1,CANSEQ ;UNKNOWN TAPE ?
2247 011522 001420 BEQ 2$ ;DON'T PRINT PATTERN OR REC LENGTH
2248 011524 022737 000060 002452 CMP #60,CMMAND
2249 011532 101014 BHI 2$
2250 011534 PRINTB #FORM09,PATRN,CHAREC
2251 011564 005737 002524 2$: TST EXTDSN ;SEE IF EXTENDED SENSE UPDATED
2252 011570 001422 BEQ 1$ ;BRANCH IF NO
2253 011572 005037 002524 CLR EXTDSN ;DO EXTEND READ
2254 011576 004737 035704 JSR PC,EXREAD
2255 011602 005737 002606 TST SENFLG ;SEE IF EXTEND SENSE COMPLETE
2256 011606 001411 BEQ 3$ ;BRANCH IF YES
2257 011610 PRINTB #FORM21
2258 011630 000402 BR 1$ ;SKIP EXTEND SENSE PRINTOUT
2259 011632 004737 012164 3$: JSR PC,EXTSEN
2260 011636 000207 1$: RTS PC
2261
2262 011640 HEDER3: PRINTB #FORM01,RHADDR, TM78N, TU78N
2263 011674 PRINTB #FORM23,HOURS,MINUTE
2264 011724 PRINTB #FORM02,RHADD(R1),UNSLCT(R1),CMDADR(R1)
2265 011760 PRINTB #FORM24,WRDCNT(R1)
2266 012004 PRINTB #FORM03,CMMAND
2267 012030 022737 000001 002206 CMP #1,CANSEQ ;UNKNOWN TAPE ?
2268 012036 001414 BEQ 2$ ;DON'T PRINT PATTERN OR REC LENGTH
2269 012040 PRINTB #FORM09,PATRN,CHAREC
2270 012070 005737 002206 2$: TST CANSEQ
2271 012074 001414 BEQ 1$
2272 012076 PRINTB #FORM07,RECORD,FILENM
2273 012126 1$: PRINTB #FORM06,FIRBYT,EXPECT,ACTUAL
2274 012162 000207 RTS PC
2275
2276 012164 EXTSEN: PRINTX #FORM11,<B,COMCOD>,<B,INTCOD>,<B,FAILCD>
2277 012226 PRINTX #FORM12,<B,CH0TIE>,<B,RPFAIL>,<B,RAMT>,<B,CASSTA>,<B,MTA0>,<B,WMCSTA>
2278 012312 PRINTX #FORM13,<B,CH1TIE>,<B,RPATH>,<B,RDON>,<B,CBUS>,<B,MTA1>,<B,WRTDAT>
2279 012376 PRINTX #FORM14,<B,CH2TIE>,<B,RSTAT>,<B,RILL>,<B,DBUS>,<B,MTA2>,<B,DDRA>
2280 012462 PRINTX #FORM15,<B,CH3TIE>,<B,RCMLP>,<B,RMK2>,<B,TUSEL0>,<B,MTA3>,<B,DDR8>
2281 012546 PRINTX #FORM16,<B,CH4TIE>,<B,CRCWRD>,<B,REND>,<B,TUSEL1>,<B,MTA4>,<B,WMCERR>
2282 012632 PRINTX #FORM17,<B,CH5TIE>,<B,ECCCOR>,<B,RPSTA>,<B,TAMT>,<B,MTA5>,<B,INSTA>
2283 012716 PRINTX #FORM18,<B,CH6TIE>,<B,ECCSTA>,<B,RPOSTN>,<B,PSTAT>,<B,RETCT>,<B,RETCT1>
2284 013002 PRINTX #FORM19,<B,CH7TIE>,<B,RTIER>,<B,RDATA>,<B,PRDD>,<B,TUX>,<B,XFRCTL>

```

2285 013066  
2286 013130  
2287 013164 000207

PRINTX #FORM20,<B,CHPTIE>,<B,XRETRY>,<B,ENAON>  
PRINTX #FORM29,BCNTLO,PDCNLO,ERCNLO  
RTS PC

2288  
2289  
2290  
2291  
2292  
2293  
2294  
2295

:  
: ERROR MESSAGES FOR TEST MODULES  
:

2296	013166	050117	051105	052101	ERM001::	.ASCIZ /OPERATION TIMED OUT/
2297	013212	047111	042524	051122	ERM002::	.ASCIZ /INTERRUPT FROM NON-EXISTENT UNIT/
2298	013253	127	047522	042524	ERM003::	.ASCIZ /WROTE ON TAPE BUT NOT PERFECTLY, RETRY/
2299	013322	047125	042504	044506	ERM004::	.ASCIZ /UNDEFINED INTERRUPT CODE RECEIVED/
2300	013364	050117	051105	052101	ERM005::	.ASCIZ /OPERATION COMPLETED BUT EXTENDED SENSE HAS STATUS OF INTEREST/
2301	013462	047125	054105	042520	ERM006::	.ASCIZ /UNEXPECTED TAPE MARK FOUND/
2302	013515	125	042516	050130	ERM007::	.ASCIZ /UNEXPECTED BEGINNING OF TAPE/
2303	013552	047125	054105	042520	ERM009::	.ASCIZ /UNEXPECTED LOGICAL EOT/
2304	013601	124	050101	020105	ERM010::	.ASCIZ /TAPE PROTECTED CAN'T WRITE/
2305	013634	052524	047440	020116	ERM011::	.ASCIZ /TU ON LINE, BUT NOT READY/
2306	013666	052524	047440	020116	ERM012::	.ASCIZ /TU ON LINE, WRONG PORT/
2307	013715	124	020125	051511	ERM013::	.ASCIZ /TU IS OFFLINE/
2308	013733	124	020125	047504	ERM014::	.ASCIZ /TU DOES NOT EXIST OR POWER OFF/
2309	013772	046102	047101	020113	ERM015::	.ASCIZ /BLANK TAPE/
2310	014005	125	044516	020124	ERM016::	.ASCIZ /UNIT JUST CAME ON LINE/
2311	014034	047514	043516	051040	ERM017::	.ASCIZ /LONG RECORD/
2312	014050	044123	051117	020124	ERM018::	.ASCIZ /SHORT RECORD/
2313	014065	122	050105	040505	ERM019::	.ASCIZ /REPEAT OPERATION/
2314	014106	042522	042522	042101	ERM020::	.ASCIZ /REREAD IN OPPOSITE DIRECTION/
2315	014143	125	051116	040505	ERM021::	.ASCIZ /UNREADABLE RECORD/
2316	014165	123	040524	052524	ERM022::	.ASCIZ /STATUS ERROR/
2317	014202	051127	052111	020105	ERM023::	.ASCIZ /WRITE ERROR BEYOND EOT/
2318	014231	102	042101	052040	ERM024::	.ASCIZ /BAD TAPE, POSITIONED LOST/
2319	014263	124	020115	040506	ERM025::	.ASCIZ /TM FAULT A/
2320	014276	052524	043040	052501	ERM026::	.ASCIZ /TU FAULT A/
2321	014311	124	020115	040506	ERM027::	.ASCIZ /TM FAULT B/
2322	014324	046524	051040	040505	ERM028::	.ASCIZ /TM READY DOESN'T SET/
2323	014351	124	020125	040506	ERM029::	.ASCIZ /TU FAULT B/
2324	014364	041115	043040	052501	ERM030::	.ASCIZ /MB FAULT/
2325	014375	115	051501	041123	ERM031::	.ASCIZ /MASSBUS STATUS ERROR/
2326	014422	040515	051523	052502	ERM032::	.ASCIZ /MASSBUS TERMINATION OF DATA TRANSFER/
2327	014467	104	052101	020101	ERM033::	.ASCIZ /DATA COMPARE ERROR/
2328	014512	040510	042122	053440	ERM034::	.ASCIZ /HARD WRITE ERROR/
2329	014533	115	051501	041123	ERM035::	.ASCIZ /MASSBUS TERMINATION OF NON DATA TRANSFER/
2330	014604	052524	034067	047040	ERM036::	.ASCIZ /TU78 NOT AT BOT AFTER REWIND/
2331	014641	105	040522	042523	ERM037::	.ASCIZ /ERASE FUNCTION FAILED/

2332 014670  
2333  
2334 014670 014716  
2335 014672 014731  
2336 014674 014746  
2337 014676 014767  
2338 014700 015011  
2339 014702 015025  
2340 014704 015060

:  
: .EVEN  
: FUNCTIONAL SEQUENCE ERROR HEADER MESSAGES  
: MESADR: FM.NOP  
: FM.REW  
: FM.WR  
: FM.WCK  
: FM.SP  
: FM.TM  
: FM.TS

2341	014706	015106				FM.WV	
2342	014710	015145				FM.E	
2343	014712	015172				FM.SL	
2344	014714	015226				FM.WC	
2345							
2346	014716	047516	050117	052040	FM.NOP:	.ASCIZ/NOOP TEST]/	
2347	014731	122	053505	047111	FM.REW:	.ASCIZ/REWIND TEST]/	
2348	014746	051127	052111	026505	FM.WR:	.ASCIZ/WRITE-READ TEST]/	
2349	014767	127	044522	042524	FM.WCK:	.ASCIZ/WRITE CHECK TEST]/	
2350	015011	123	040520	042503	FM.SP:	.ASCIZ/SPACE TEST]/	
2351	015025	124	050101	020105	FM.TM:	.ASCIZ/TAPE MARK WRITE-READ TEST]/	
2352	015060	040524	042520	046440	FM.TS:	.ASCIZ/TAPE MARK SPACE TEST]/	
2353	015106	047527	051522	020124	FM.WV:	.ASCIZ/WORST CASE VACUUM COLUMN TEST]/	
2354	015145	105	040522	042523	FM.E:	.ASCIZ/ERASE FUNCTION TEST]/	
2355	015172	044123	051117	020124	FM.SL:	.ASCIZ/SHORT AND LONG RECORD TEST]/	
2356	015226	047527	051522	020124	FM.WC:	.ASCIZ/WORST CASE CAPSTAN TEST]/	
2357		015260				.EVEN	
2358						.LIST	BEX
2359							

2361  
2362  
2363  
2364  
2365  
2366  
2367  
2368  
2369  
2370  
2371  
2372  
2373  
2374  
2375  
2376  
2377  
2378  
2379  
2380  
2381  
2382  
2383  
2384  
2385  
2386  
2387  
2388  
2389  
2390  
2391  
2392  
2393  
2394  
2395  
2396  
2397  
2398  
2447  
2448  
2449  
2450  
2451  
2452  
2453  
2454  
2455  
2456  
2457  
2458  
2459  
2460  
2461  
2462  
2463  
2464

.SBTTL GLOBAL SUBROUTINES SECTION

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

\*\*\*\*\*  
: GLOBAL MACROS  
:\*\*\*\*\*

.MACRO RETURN

RTS PC

.ENDM

.MACRO CALL XXX

JSR PC,XXX

.ENDM

.MACRO PUSH ARG

.IRP X,<ARG>

MOV X,-(SP) ;STORE X ON STACK

.ENDM

.ENDM

.MACRO POP ARG

.IRP X,<ARG>

MOV (SP)+,X ;RESTORE X FROM STACK

.ENDM

.ENDM

:--

\*\*\*\*\*  
: SUBROUTINES USED IN THE INIT CODE ONLY  
:\*\*\*\*\*

:\*  
:\*MODULE 2.1 (BUILDS HARDWARE P-TABLES AND STATISTIC TABLES)  
:\*  
:\*BEGINROUTINE (MOD = 2.1 GET HARDWARE P-TABLES /BLDTBL/)  
:\* TABLE COUNTER = 0  
:\* STORE NUMBER OF UNITS INTO UNIT#  
:\* DO WHILE UNIT # IS GREATER THAN ZERO  
:\* : GET HARDWARE P-TABLE (THIS IS A SUPERVISOR CALL AND WILL SET

```

2465 : * : OR RESET BRANCH COMPLETE. THE HARDWARE
2466 : * : TABLE WAS CREATED IN THE INPUT MODULE 1.0)
2467 : * : IF BRANCH_COMPLETE IS TRUE
2468 : * : : THEN
2469 : * : : RELOCATE HARDWARE P-TABLE
2470 : * : : BUILD SPACE FOR A STATISTIC TABLE
2471 : * : : STORE UNIT # IN P-TABLE
2472 : * : : SUBTRACT TABLE LENGTH FROM MEMORY
2473 : * : : TABLE COUNTER = TABLE COUNTER +1
2474 : * : ENDF
2475 : * : DECREMENT UNIT#
2476 : * ENDDO
2477 : * CALCULATE MAXIMUM MEMORY AVAILABLE
2478 : * CALCULATE START OF MEMORY
2479 : * IF UNKNOWN TAPE OPTION
2480 : * : THEN
2481 : * : IF TABLE_COUNT GREATER THAN 1
2482 : * : : THEN
2483 : * : : SET ERROR FLAG
2484 : * : ENDF
2485 : * ENDF
2486 : * DIVIDE MEMORY IN HALF
2487 : * CALCULATE START OF SECOND HALF OF MEMORY
2488 : * ENDRoutine
2489
2490
2491
2492 015260 BLDTBL: PUSH <R1,R2,R3,R4,R5,R10>
2493 015276 005037 002424 CLR R10 ;CLEAR TEMP REGISTER
2494 015302 005005 CLR R5 ;
2495 015304 005037 002344 CLR TBLCNT ;RESET TABLE COUNTER
2496 015310 013701 002012 MOV L$UNIT,R1 ;TEMPORARY STORAGE OF # OF UNITS
2497 015314 013737 002354 002330 MOV VRMADD,PTBLAD ;PTBLAD CONTAINS STARTING ADDRESS OF P-TABLES
2498 015322 013703 002330 MOV PTBLAD,R3 ;TEMPORARY STORAGE OF FREE MEMORY
2499 015326 1$: GPHARD R5,R2 ;FIND LOCATION OF P-TABLE
2500 015334 BNCOMPLETE 3$ ;UNIT NOT FOUND
2501 015336 012704 000004 MOV #4,R4 ;COUNTER
2502 015342 012223 2$: MOV (R2)+,(R3)+ ;RELOCATE HARDWARE P-TABLE
2503 015344 005304 DEC R4 ;DECREMENT COUNTER
2504 015346 001375 BNE 2$ ;LOOP UNTIL FULL TABLE IS MOVED
2505 015350 010523 MOV R5,(R3)+ ;STORE SUPERVISOR UNIT NUMBER IN UNIT TABLE
2506 015352 063703 002342 ADD STTBLH,R3 ;GET START OF NEXT TABLE
2507 015356 006237 002346 ASR TBLNTH
2508 015362 163737 002346 002312 SUB TBLNTH,BUFSIZ ;UPDATE AVAILABLE MEMORY
2509 015370 006337 002346 ASL TBLNTH
2510 015374 063737 002346 002354 ADD TBLNTH,VRMADD ;UPDATE OF WHERE FREE MEMORY BEGINS
2511 015402 005737 002424 TST R10 ;HAVE WE BEEN HERE YET
2512 015406 001003 BNE 6$ ;YES , GET NEXT TABLE #
2513 015410 005237 002424 INC R10 ;NO , SET FOR GCR TABLE
2514 015414 000744 BR 1$ ;LOOP
2515 015416 005037 002424 6$: CLR R10 ;SET FOR PE TABLE
2516 015422 005237 002344 INC TBLCNT ;UPDATE # OF TABLES
2517 015426 005205 INC R5 ;GET NEXT LOGICAL UNIT
2518 015430 000402 BR 5$
2519 015432 005205 3$: INC R5 ;GET NEXT UNIT
2520 015434 000734 BR 1$ ;

```

```
2521 015436 005301      5$: DEC R1      ;GET NEXT UNIT
2522 015440 001332      BNE 1$      ;AM I DONE?
2523 015442 022737 000001 002206  CMP #1,CANSEQ ;CHECK FOR UNKNOWN TAPE OPTION
2524 015450 001007      BNE 4$      ;BRANCH IF NOT CHOSEN
2525 015452 022737 000001 002344  CMP #1,TBLCNT ;SEE IF ONLY ONE UNIT CHOSEN
2526 015460 001403      BEQ 4$      ;BRANCH IF YES
2527 015462 012737 000006 002320  MOV #6,ERRFLG ;SET ERROR FLAG
2528 015470 042737 000001 002312  4$: BIC #1,BUFSIZ ;MAKE EVEN BUFFER
2529 015476 013737 002354 002332  MOV VRMADD,REDBUF
2530 015504 063737 002312 002332  ADD BUFSIZ,REDBUF ;REDBUF CONTAINS STARTING ADDRESS OF READ BUFFER
2531 015512      POP <R10,R5,R4,R3,R2,R1> ;
2532 015530 000207      RTS PC      ;GO HOME
```

```
2533
2534
2535
2536
2537 ;*
2538 ;*MODULE 2.2 (CHECKS TO SEE IF DEVICE EXISTS)
2539 ;*
2540 ;*BEGINROUTINE (MOD = 2.2 /DEVCHK/)
2541 ;* SET UP UNIT TABLE
2542 ;* DO FOR 1 TO TABLE COUNT OR ERROR_FLAG SET
2543 ;* : GET CORRECT CAS REGISTER FOR UNIT
2544 ;* : SET UP TIMEOUT COUNTER
2545 ;* : SET INTERRUPT VECTOR UP
2546 ;* : GENERATE RH VECTOR TABLE
2547 ;* : STORE TM UNIT # IN MEMORY LOCATION
2548 ;* : STORE TM UNIT# IN UNIT SELECT LOCATION IN CAS
2549 ;* : FIND NON DATA TRANSFER COMMAND LOCATION FOR SPECIFIED TAPE UNIT#.
2550 ;* : CLEAR ATTENTION SUMMARY REG
2551 ;* : ENABLE INTERRUPTS
2552 ;* : LOAD TU SENSE COMMAND
2553 ;* : SEE IF NON-EXISTENT DRIVE
2554 ;* : LOWER PRIORITY
2555 ;* : DO UNTIL ERROR FLAG SET OR ATTENTION BIT SET AND CORRECT ATTENTION ADDRESS SET
2556 ;* : INCREMENT TIMEOUT COUNTER
2557 ;* : IF TIMEOUT COUNTER EQUALS MAXIMUM TIME
2558 ;* : THEN-SET ERROR_FLAG
2559 ;* : ENDDO
2560 ;* : IF ERROR_FLAG NOT SET
2561 ;* : THEN-
2562 ;* : IF DONE NOT SET
2563 ;* : THEN-SET ERROR FLAG
2564 ;* : ENDDO
2565 ;* : ENDDO
2566 ;* : IF ERROR_FLAG NOT SET
2567 ;* : THEN
2568 ;* : IF ILLEGAL DRIVE TYPE
2569 ;* : THEN-SET ERROR_FLAG
2570 ;* : ENDDO
2571 ;* : ENDDO
2572 ;* : IF ERROR_FLAG NOT SET
2573 ;* : THEN-IF START FLAG SET
2574 ;* : THEN
2575 ;* : PRINT SERIAL NUMBER
2576 ;* : ENDDO
```

```

2577          ;* : ENDF
2578          ;* ENDDO
2579          ;* CLEAR ATTENTION SUMMARY REGISTER
2580          ;*ENDROUTINE
2581
2582
2583
2584 015532      DEVCHK: PUSH    <R1,R2,R3,R4,R5>
2585 015544 012737 000001 002352      MOV    #1,TBLNUM      ;SET UP COUNTER
2586 015552 012737 036156 002376      MOV    #INSERV,INTVEC ;PUT INTERRUPT SERVICE ROUTINE INTO INTVEC
2587 015560 013703 002352      1$:  MOV    TBLNUM,R3      ;GET TABLE NUMBER
2588 015564 005303      DEC    R3              ;FIX TABLE OFFSET
2589 015566 006303      ASL    R3              ;
2590 015570 012737 000000 002316      MOV    #RHADD,DISPLC  ;GET RH ADDRESS
2591 015576 004737 036076      JSR    PC,TBLDIS      ;
2592 015602 017763 164510 002400      MOV    @DISPLC,RHADD1(R3) ;PUT ADDRESS INTO ADDRESS TABLE
2593 015610 013701 002316      MOV    DISPLC,R1      ;GET PTABLE ADDRESS
2594 015614 016103 000002      MOV    RHVEC(R1),R3   ;
2595 015620 062703 000002      ADD    #2,R3          ;GET NEXT LOCATION AFTER RHVEC
2596 015624 005013      CLR    (R3)          ;CLEAR THAT LOCATION
2597 015626 005003      CLR    R3            ;SET UP TIME OUT COUNTER
2598 015630      SETVEC RHVEC(R1),INTVEC,PRIRTY
2599 015656 062737 000010 002376      ADD    #10,INTVEC     ;GET NEXT INTERRUPT LOCATION
2600 015664 017701 164426      MOV    @DISPLC,R1     ;TEMPORARY STORAGE OF RH ADDRESS
2601 015670 017737 164422 002356      MOV    @DISPLC,RHADDR ;STORAGE FOR RH ADDRESS
2602 015676 012761 001020 000024      MOV    #1020,DIAGTS(R1)
2603 015704 012737 000004 002316      MOV    #TMNUM,DISPLC  ;FIND OFFSET FOR TM NUMBER
2604 015712 004737 036076      JSR    PC,TBLDIS      ;GET SPECIFIED LOCATION
2605 015716 017705 164374      MOV    @DISPLC,R5     ;R5 CONTAINS TMNUM
2606 015722 017737 164370 002370      MOV    @DISPLC,TM78N  ;STORAGE FOR TM78 NUMBER
2607 015730 010537 002432      MOV    R5,BITPST
2608 015734 004737 036406      JSR    PC,SHIFTL
2609 015740 017761 164352 000010 3$:  MOV    @DISPLC,UNSLCT(R1) ;STORE UNIT# IN UNIT SELECT
2610 015746 012711 040000      MOV    #40000,(R1)    ;CLEAR MASSBUS TRE BIT
2611 015752 005011      CLR    (R1)          ;CLEAR TRE SHADOW
2612 015754 000240      NOP                  ;GIVE HIM SOME TIME
2613 015756 012737 000006 002316      MOV    #TUNUM,DISPLC  ;FIND OFFSET FOR TU NUMBER
2614 015764 004737 036076      JSR    PC,TBLDIS      ;GET SPECIFIED LOCATION
2615 015770 017702 164322      MOV    @DISPLC,R2     ;R2 CONTAINS TU NUM
2616 015774 017737 164316 002372      MOV    @DISPLC,TU78N  ;STORAGE FOR TU78 NUMBER
2617 016002 006302      ASL    R2              ;INCREMENT BY WORDS
2618 016004 062702 000040      ADD    #NDTFC0,R2     ;FIND LOCATION FOR TU SENSE CMD
2619 016010 060102      ADD    R1,R2          ;FIND LOCATION FOR TUSENSE CMD
2620 016012 005037 002374      CLR    INTRPT         ;CLEAR INTERRUPT FLAG
2621 016016 012761 000377 000016      MOV    #377,ATTBIT(R1) ;CLEAR ATTENTION SUMMARY REG
2622 016024 005011      CLR    (R1)
2623 016026 005761 000052      TST    INTRDA(R1)    ;TM78 READY ??
2624 016032 100405      BMI    20$           ;YES , CONTINUE
2625 016034 012737 000005 002320      MOV    #5,ERRFLG     ;NO , FATAL ERROR
2626 016042 000137 016504      JMP    7$            ;EXIT
2627 016046 012711 000100      20$: MOV    #100,(R1)      ;ENABLE INTERRUPT
2628 016052 012712 000011      MOV    #SENSE,(R2)   ;PUT TU SENSE COMMAND INTO CAS REGISTER
2629 016056 032761 010000 000010      BIT    #10000,UNSLCT(R1) ;SEE IF NON EXISTENT DRIVE
2630 016064 001405      BEQ    4$            ;BRANCH IF DRIVE EXISTS
2631 016066 012737 000004 002320      MOV    #4,ERRFLG    ;NON EXISTENT DRIVE
2632 016074 000137 016504      JMP    7$

```



```

2633 016100      4$:  SETPRI #PRI00      ;ENABLE CPU INTERRUPTS
2634 016106      8$:  DELAY 1      ;WAIT 10 USECONDS
2635 016136
2636 016140 005737 002374      TST INTRPT      ;SEE IF RECEIVED INTERRUPT
2637 016144 001010      BNE 13$      ;BRANCH WHEN GET INTERRUPT
2638 016146 005203      INC R3      ;TIMEOUT COUNTER
2639 016150 022703 023420      CMP #10000.,R3      ;SEE IF MACHINE HUNG
2640 016154 001354      BNE 8$      ;WAIT SOME MORE
2641 016156 012737 000002 002320      MOV #2,ERRFLG      ;TIME OUT TM78 HUMG
2642 016164 000547      BR 7$      ;GET OUT
2643 016166
2644 016174 033761 002432 000016 13$:  SETPRI PRIRTY
2645 016202 001005      BIT BITPST,ATTBIT(R1) ;SEE IF CORRECT UNIT TERMINATED
2646 016204 005037 002374      BNE 5$      ;BRANCH WHEN R4=ATTENTION BIT
2647 016210 012711 000100      CLR INTRPT      ;CLEAR INTERRUPT BIT
2648 016214 000731      MOV #100,(R1)      ;ENABLE INTERRUPTS
2649 016216 005037 002374      BR 4$      ;GO BACK AND WAIT FOR CORRECT INTERRUPT
2650 016222 013761 002370 000010 5$:  CLR INTRPT      ;RESET INTERRUPT FLAG
2651 016230 016105 000036      MOV NDTICD(R1),R5 ;MOVE UNIT THAT INTERRUPTED TU REGISTER
2652 016234 000305      SWAB R5      ;GET ATTENTION ADDRESS IN LOW BYTE
2653 016236 042705 177774      BIC #177774,R5      ;MASK OFF HIGH BITS
2654 016242 023705 002372      CMP TU78N,R5      ;SEE IF CORRECT UNIT
2655 016246 001314      BNE 4$      ;BRANCH IF WRONG UNIT
2656 016250 122761 000001 000036      CMPB #1,NDTICD(R1) ;SEE IF DONE OCCURRED
2657 016256 001404      BEQ 14$      ;BRANCH WHEN DONE
2658 016260 012737 000003 002320      MOV #3,ERRFLG      ;SET UP FOR INTERRUPT ERROR
2659 016266 000506      BR 7$      ;LEAVE MODULE
2660 016270 022761 000100 000026 14$:  CMP #100,DRVTP(R1) ;SEE IF CORRECT DRIVE TYPE
2661 016276 002004      BGE 6$      ;SET ERROR FLAG IF LESS THAN
2662 016300 012737 000001 002320      MOV #1,ERRFLG      ;DRIVE TYPE NOT TM78 TYPE
2663 016306 000476      BR 7$      ;LEAVE MODULE
2664 016310 032761 040000 000020 6$:  BIT #40000,DSE0(R1) ;SEE IF TU IS THERE
2665 016316 001004      BNE 15$      ;
2666 016320 012737 000004 002320      MOV #4,ERRFLG      ;PRINT ERROR
2667 016326 000466      BR 7$      ;LEAVE MODULE
2668 016330 005002      15$:  CLR P2      ;SET UP LOOP COUNTER
2669 016332 012703 002300      MOV #SERNUM,R3      ;GET STORAGE LOCATION FOR ASCII SERIAL #
2670 016336 000361 000030      12$:  SWAB SERIAL(R1) ;GET MOST SIGNIFICANT NUMBERS
2671 016342 116105 000030      MOVB SERIAL(R1),R5 ;MOVE TWO MOST SIGNIFICANT NUMBERS TO REG
2672 016346 042705 000017      BIC #17,R5      ;MASK OFF LOWER FOUR BITS
2673 016352 006205      ASR R5      ;
2674 016354 006205      ASR R5      ;SHIFT TO LOWER FOUR BITS
2675 016356 006205      ASR R5      ;
2676 016360 006205      ASR R5      ;
2677 016362 062705 000060      ADD #60,R5      ;CONVERT TO ASCII
2678 016366 110523      MOVB R5,(R3)+      ;MOVE TO OUTPUT STORAGE
2679 016370 116105 000030      MOVB SERIAL(R1),R5 ;GET NEXT MSB
2680 016374 042705 000360      BIC #360,R5      ;MASK OFF SECOND FOUR BITS
2681 016400 062705 000060      ADD #60,R5      ;CONVERT TO ASCII
2682 016404 110523      MOVB R5,(R3)+      ;MOVE TO OUTPUT STORAGE
2683 016406 005202      INC R2      ;INCREMENT LOOP COUNTER
2684 016410 022702 000001      CMP #1,R2      ;
2685 016414 001750      BEQ 12$      ;GET NEXT TWO NUMBERS
2686 016416 005737 002276      TST START1
2687 016422 001420      BEQ 2$
2688 016424      PRINTF #SERLNB,TU78N, TM78N,RHADDR,#SERNUM

```

```

2689 016464 005237 002352 2$: INC TBLNUM ;GET NEXT UNIT
2690 016470 023737 002352 002344 CMP TBLNUM,TBLCNT ;SEE IF LAST UNIT
2691 016476 003002 BGT 7$ ;CONTINUE
2692 016500 000137 015560 JMP 1$ ;START AGAIN
2693 016504 012761 000377 000016 7$: MOV #377,ATTBIT(R1) ;CLEAR ATTENTION SUMMARY REG
2694 016512 POP <R5,R4,R3,R2,R1> ;RESTORE REGISTERS
2695 016524 000207 RTS PC ;GO HOME
2696
2697 :*MODULE 2.3 (QUESTIONS NEEDED TO BE ASKED FOR BRUTUS OPTIONS)
2698 :*
2699 :*BEGINROUTINE
2700 :* IF PROGRAM IN UNATTENDED MODE
2701 :* : THEN
2702 :* : SET ERROR FLAG
2703 :* : ELSE
2704 :* : CLEAR COMMAND LOCATIONS
2705 :* : CLEAR DATA TRANSFER FLAG
2706 :* : ASK HOW MANY COMMANDS IN STRING
2707 :* : DO FOR NUMBER OF COMMANDS IN STRING
2708 :* : : ASK FOR COMMAND
2709 :* : : IF DATA TRANSFER COMMAND
2710 :* : : : THEN
2711 :* : : : SET DATA TRANSFER FLAG
2712 :* : : : ENDF
2713 :* : : ASK FOR REPITITION OF THAT COMMAND
2714 :* : : ENDDO
2715 :* : IF DATA TRANSFER
2716 :* : : THEN
2717 :* : : IF BRUTUS WITH ERRORS
2718 :* : : : THEN
2719 :* : : : ASK ERROR AND STATISTIC INFORMATION
2720 :* : : : ELSE
2721 :* : : : INHIBIT ERROR AND STATISTICAL INFORMATION
2722 :* : : : ENDF
2723 :* : : ASK PATTERN NUMBER
2724 :* : : IF RANDOM PATTERN
2725 :* : : : THEN
2726 :* : : : ASK FOR RANDOM RECORD LENGTH
2727 :* : : : ELSE
2728 :* : : : ASK LENGTH OF RECORD
2729 :* : : : CLEAR RANDOM RECORD LENGTH
2730 :* : : : ENDF
2731 :* : : SET UP LOOP OF PATTERN ACCORDING TO HOW OFTEN IT REPEATS
2732 :* : : IF USER SELECTS TO INPUT ITS OWN PATTERN
2733 :* : : : THEN
2734 :* : : : : DO UNTIL=14
2735 :* : : : : : INPUT PATTERN
2736 :* : : : : ENDDO
2737 :* : : : ENDF
2738 :* : : ENDF
2739 :* : ENDF
2740 :*ENDROUTINE
2741
2742
2743 016526 BRUTQ: PUSH <R1,R2,R3,R4>
2744 016536 MANUAL ;SEE IF UAM FLAG IS SET

```

2745	016540					BCOMPLETE 1\$	:BRANCH IF MANUAL INTERVENTION ALLOWED
2746	016542	012737	000004	002320		MOV #4,ERRFLG	:SET ERROR FLAG UAM FLAG IS SET
2747	016550	000137	017364			JMP 6\$	:RETURN
2748	016554	005037	002206		1\$:	CLR CANSEQ	:
2749	016560	012701	002232			MOV #COMMD1,R1	:GET START OF BRUTUS SOFTWARE P-TABLE
2750	016564	005021			7\$:	CLR (R1)+	:CLEAR BRUTUS SOFTWARE P-TABLE
2751	016566	022701	002262			CMP #REP6+2,R1	:SEE IF DONE CLEARING
2752	016572	001374				BNE 7\$	:BRANCH IF NOT DONE CLEARING
2753	016574	005037	002516			CLR DATRAN	
2754	016600					GMANID MSG19,CMDSTR,D,177777,1,6,NO	:# OF CMDS IN STRING
2755	016620	012702	002232			MOV #COMMD1,R2	
2756	016624	013737	002230	002340		MOV CMDSTR,CONTER	:STORE USER ANSWER
2757	016632	005237	002340			INC CONTER	:FIX FOR BRUTUS ROUTINE
2758	016636				3\$:	GMANID MSG21,REPCMD,D,177777,3,77,NO	:GET COMMAND
2759	016656	013722	002310			MOV REPCMD,(R2)+	:GET NEXT LOCATION
2760	016662	022737	000050	002310		CMP #50,REPCMD	
2761	016670	002411				BLT 4\$	
2762	016672					GMANID MSG20,REPCMD,D,177777,1,177777,NO	:REPITION OF CMD
2763	016712	000413				BR 5\$	
2764	016714				4\$:	GMANID MSG15,REPCMD,D,177777,1,177777,NO	:RECORDS IN FILE
2765	016734	012737	000001	002516		MOV #1,DATRAN	:SET IF DATA TRANSFER
2766	016742	013722	002310		5\$:	MOV REPCMD,(R2)+	:GET NEXT LOCATION
2767	016746	005337	002230			DEC CMDSTR	:DECREMENT COUNTER
2768	016752	001331				BNE 3\$	:SEE IF THERE IS ANOTHER COMMAND
2769							
2770	016754					GMANIL MSG9,BRUERR,177777,YES	:ASK BRUTUS WITH ERRORS
2771	016770	005737	002516			TST DATRAN	:SEE IF DATA TRANSFER COMMAND
2772	016774	001434				BEQ 8\$	:BRANCH IF NON DATA TRANSFER
2773	016776	005737	002210			TST BRUERR	:SEE IF ERRORS SHOULD BE REPORTED
2774	017002	001431				BEQ 8\$	:
2775	017004					GMANIL MSG10,STAEOP,177777,YES	:STATISTICS AT END OF PASS
2776	017020					GMANIL MSG7,STAEOT,177777,YES	:STATISTICS AT EOT
2777	017034					GMANIL MSG12,RETRY,177777,YES	:RETRIES
2778	017050					GMANIL MSG17,DATCHK,177777,YES	:DATA CHECK
2779	017064	000413				BR 2\$	:
2780	017066	005037	002212		8\$:	CLR STAEOP	:
2781	017072	005037	002214			CLR STAEOT	:
2782	017076	005037	002216			CLR RETRY	:
2783	017102	005037	002226			CLR DATCHK	:
2784	017106	005737	002516			TST DATRAN	:
2785	017112	001524				BEQ 6\$	:
2786	017114				2\$:	GMANID MSG14,PATRN,D,177777,0,11,YES	:PATTERN #
2787	017134	022737	000010	002222		CMP #10,PATRN	:SEE IF RANDOM PATTERN
2788	017142	001024				BNE 13\$	:BRANCH IF NOT
2789	017144	005737	002210			TST BRUERR	:
2790	017150	001410				BEQ 14\$	:
2791	017152					PRINTF #RANDOM	:
2792	017172				14\$:	GMANIL MSG11,RALNTH,177777,YES	
2793	017206	005737	002224			TST RALNTH	
2794	017212	001012				BNE 12\$	
2795	017214				13\$:	GMANID MSG13,CHAREC,D,177777,1,177777,YES	:# OF BYTES IN RECORD
2796	017234	005037	002224			CLR RALNTH	:NO RANDOM RECORD LENGTH
2797	017240	012737	000004	002350	12\$:	MOV #4,PATLOP	:SET UP PATTERN LOOP
2798	017246	022737	000006	002222		CMP #6,PATRN	:SEE IF USER PATTERN
2799	017254	001034				BNE 9\$	:BRANCH IF NOT
2800	017256	012737	000016	002350		MOV #16,PATLOP	:SET UP PATTERN LOOP

```

2801 017264          PRINTF #INPATT          :PRINT STATEMENT
2802 017304 012704 003430  MOV #PATSIX,R4      :R4 CONTAINS OFFSET
2803 017310 012703 000016  MOV #16,R3          :SET LOOP UP
2804 017314          10$: GMANID MSG8,USEPAT,0,177777,0,177777,NO
2805 017334 013724 002306  MOV USEPAT,(R4)+    :GET AND STORE USER PATTERN
2806 017340 005303          DEC R3                :DECREMENT LOOP
2807 017342 001364          BNE 10$                :SEE IF DONE
2808 017344 000407          BR 6$                    :BRANCH IF DONE
2809 017346 022737 000007 002222 9$: CMP #7,PATRN        :SEE IF INCREMENT PATTERN
2810 017354 001003          BNE 6$                    :BRANCH IF NOT
2811 017356 012737 000200 002350 6$: MOV #200,PATLOP    :SET UP LOOP
2812 017364          POP <R4,R3,R2,R1>
2813 017374 000207          RTS PC

```

```

2814
2815
2816
2817
2818      :*
2819      :*MODULE 2.4 (CLEAR STATISTIC TABLE FOR ALL USE)
2820      :*
2821      :*BEGINROUTINE (MOD = 2.4 CLEARS STATISTICS /CLRSTA/)
2822      :* DO 1 TO TABLE_COUNT
2823      :* : MOVE UNIT# TO REG
2824      :* : MOVE BYRFHI TO DISPLC (FIRST STAT IN STATISTIC TABLE)
2825      :* : CALL TBLDIS
2826      :* : DO WHILE (RHADD + TBLNTH) IS GREATER THAN DISPLC
2827      :* : : CLEAR (DISPLC) ;CLEARS CONTENT OF DISPLAC
2828      :* : : INCREMENT DISPLC
2829      :* : ENDDO
2830      :* ENDDO
2831      :*ENDROUTINE

```

```

2832
2833
2834 017376          CLRSTA: PUSH <R2>
2835 017400 012737 000001 002352  MOV #1,TBLNUM      :SET UP UNIT NUMBER
2836 017406 012737 000012 002316  1$: MOV #SERWRT,DISPLC :GET FIRST LOCATION OF STATISTIC TABLE
2837 017414 004737 036076          JSR PC,TBLDIS     :GETS ADDRESS OF SPECIFIED LOCATION
2838 017420 013702 002316          MOV DISPLC,R2     :R2 CONTAINS ADDRESS OF FIRST STAT OF TABLE
2839 017424 063702 002342          ADD STTBLH,R2   :FIND END OF TABLE
2840 017430 005077 162662          CLR @DISPLC      :CLEAR SPECIFIED LOCATION
2841 017434 062737 000002 002316  2$: ADD #2,DISPLC    :GET NEXT LOCATION
2842 017442 020237 002316          CMP R2,DISPLC    :SEE IF REACH THE END OF STATISTIC TABLE
2843 017446 001370          BNE 2$                :IF NOT LOOP
2844 017450 005737 002444          TST DENFLG      :DID WE DO GCR TABLE YET ?
2845 017454 001003          BNE 3$                :YES GET NEXT UNIT
2846 017456 005237 002444          INC DENFLG      :NO DO NOW
2847 017462 000751          BR 1$                :LOOP
2848 017464 005037 002444          3$: CLR DENFLG      :RESET DEN FLAG
2849 017470 005237 002352          INC TBLNUM      :GET NEXT UNIT
2850 017474 023737 002352 002344  CMP TBLNUM,TBLCNT :HAVE I CLEARED ALL TABLES
2851 017502 003741          BLE 1$                :LOOP UNTIL ALL TABLE HAVE BEEN CLEARED
2852 017504          POP <R2>
2853 017506 000207          RTS PC

```

2854  
2855  
2856

2857  
2858  
2859  
2860  
2861  
2862  
2863  
2864  
2865  
2866  
2867  
2868  
2869  
2870  
2871  
2872  
2873  
2874  
2875  
2876  
2877  
2878  
2879  
2880  
2881  
2882  
2883  
2884  
2885  
2886  
2887  
2888  
2889  
2890  
2891  
2898  
2899  
2900  
2901  
2902  
2903

```

:
:*
:*MODULE 2.5 (OUTPUTS FATAL ERRORS OF USER INPUT)
:*
:*BEGINROUTINE (MOD = 2.5 OUTPUT FATAL ERRORS OF USER INPUT /ERROUT/)
:* GET UNIT TABLE ADDRESS
:* SET UP ADDRESS TO START OF ERROR_TABLE
:* ADD ERROR FLAG TO START OF ERROR_TABLE
:* MOVE ADDRESS TO PRINT ROUTINE
:* PRINT ERROR
:*ENDROUTINE

```

017510				ERROUT: PUSH	<R1,R2>	
017514	012737	000000	002316	MOV	#RHADD,DISPLC	:GET TABLE ADDRESS
017522	004737	036076		JSR	PC,TBLDIS	:
017526	017702	162564		MOV	@DISPLC,R2	:R2 CONTAINS ADDRESS OF TABLE
017532	013701	002320		MOV	ERRFLG,R1	:PUT ERROR FLAG INTO REG
017536	005301			DEC	R1	:SETTING UP CORRECT ADDRESS
017540	006301			ASL	R1	:DOUBLE BYTES TO MAKE WORD
017542	022701	000004		CMP	#4,R1	
017546	001422			BEQ	1\$	
017550	022701	000010		CMP	#10,R1	:
017554	001440			BEQ	3\$	:
017556				PRINTF	FETBL(R1),TU78N,TM78N,RHADDR	
017612	000437			BR	2\$	
017614				1\$: PRINTF	FETBL(R1),TU78N,TM78N,RHADDR,NDTICD(R2)	
017654	000416			BR	2\$	
017656				3\$: PRINTF	FETBL(R1),TM78N,RHADDR,INTRDA(R2)	
017712				2\$: POP	<R2,R1>	
017716	000207			RTS	PC	

```

:*****
:SUBROUTINES USED IN TEST MODULE ONLY
:*****

```

```

2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950

```

```

: *
: *MODULE 3.1 SEQUENCE DEFINITION
: *
: *BEGINROUTINE (MOD = 3.1 SEQUENCE DEFINITION /SEQDEF/)
: * GET RH ADDRESS
: * IF NOT IN RETRY PROCESS
: * : THEN
: * : IF ALL END OF TAPE FLAG SET
: * : : THEN
: * : : : LOAD A REWIND COMMAND
: * : : : LOAD REPITITION TO 1
: * : : : SET REWIND FLAG
: * : : : CLEAR EOT FLAG
: * : : : CLEAR ALLEOT FLAG
: * : : : CLEAR RECORD #
: * : : : CLEAR FILE #
: * : : : IF BRUTUS OPTION
: * : : : : THEN
: * : : : : : SET SER BIT IN CAS REGISTERS
: * : : : : ELSE
: * : : : : : IF RETRIES NOT ALLOWED
: * : : : : : : THEN
: * : : : : : : : SET SER BIT
: * : : : : : : ELSE
: * : : : : : : : IF NON-DATA TRANSFER COMMAND
: * : : : : : : : : THEN
: * : : : : : : : : : CLEAR SER BIT
: * : : : : : : : : ELSE
: * : : : : : : : : : IF WRITE COMMAND
: * : : : : : : : : : : THEN
: * : : : : : : : : : : : SET SER BIT
: * : : : : : : : : : : ELSE
: * : : : : : : : : : : : CLEAR SER BIT
: * : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : ELSE
: * : : : : : : : : : : : : : : CLEAR HARD ERROR COUNT
: * : : : : : : : : : : : : : : CALL SEQUENCE SELECT
: * : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : ELSE
: * : : : : : : : : : : : : : : : CALL RETRY SEQUENCE
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : : : : : : : ENDIF
: * : : : : : : : : : ~~~~~

```

2951	017720				SEQDEF: PUSH	<R1,R3>	
2952	017724	012737	000000	002316	MOV	#RHADD,DISPLC	:GET RH ADDRESS
2953	017732	004737	036076		JSR	PC,TBLDIS	:
2954	017736	017701	162354		MOV	@DISPLC,R1	:R1 CONTAINS RH ADDRESS
2955	017742	005737	002604		TST	RTYFLG	:SEE IF RETRY IN PROGRESS
2956	017746	001035			BNE	1\$	:BRANCH IF YES
2957	017750	023737	002600	002344	CMP	ALLEOT,TBLCNT	:SEE IF ALL UNITS AT EOT
2958	017756	001037			BNE	2\$	:BRANCH IF NOT ALL AT EOT
2959	017760	012737	000007	002452	MOV	#REWIND,CMMAND	:REWIND ALL UNITS
2960	017766	013737	002452	002462	MOV	CMMAND,RETRCD	

```
2961 017774 012737 000001 002602      MOV      #1,REPITN
2962 020002 005237 002510                INC      DENSWT
2963 020006 012737 000001 002466      MOV      #1,AUTORW      ;SET AUTO REWIND FLAG
2964 020014 005037 002540                CLR      EOTFLG        ;CLEAR EOT FLAG
2965 020020 005037 002600                CLR      ALLEOT        ;CLEAR ALL EOT FLAG
2966 020024 005037 002534                CLR      DRPFLG        ;CLEAR DROP UNIT FLAG
2967 020030 005037 002566                CLR      RECORD
2968 020034 005037 002570                CLR      FILENM
2969 020040 000415                BR       3$
2970 020042 013703 002604      1$:    MOV      RTYFLG,R3      ;R3 CONTAINS RETRY NUMBER
2971 020046 006303                ASL      R3              ;CONVERT TO TABLE OFFSET
2972 020050 004773 003212                JSR      PC,@RTYSEQ(R3) ;DO RETRY SEQUENCE
2973 020054 000407                BR       3$
2974 020056 005037 002476      2$:    CLR      HRDCNT
2975 020062 013703 002206                MOV      CANSEQ,R3      ;R3 CONTAINS CANSEQ NUMBER
2976 020066 006303                ASL      R3              ;CONVERT TO TABLE OFFSET
2977 020070 004773 002754                JSR      PC,@SEQSEL(R3) ;DO CORRECT SEQUENCE
2978 020074 022737 000001 002314  3$:    CMP      #1,BRTEST      ;SEE IF BRUTUS OPTION
2979 020102 001004                BNE      4$              ;BRANCH IF NOT
2980 020104 012761 100000 000014      MOV      #BIT15,CMDADR(R1) ;SET SER FLAG
2981 020112 000425                BR       7$
2982 020114 005737 002216      4$:    TST      RETRY        ;SEE IF RETRIES ALLOWED
2983 020120 001004                BNE      5$              ;BRANCH IF YES
2984 020122 012761 100000 000014      MOV      #BIT15,CMDADR(R1) ;SET SER FLAG
2985 020130 000416                BR       7$
2986 020132 022737 000050 002452  5$:    CMP      #50,CMMAND    ;SEE IF NDT COMMAND
2987 020140 101010                BHI      6$              ;BRANCH IF N D T
2988 020142 022737 000070 002452      CMP      #70,CMMAND    ;SEE IF WRITE COMMAND
2989 020150 103404                BLO      6$              ;BRANCH IF READ COMMAND
2990 020152 012761 100000 000014      MOV      #BIT15,CMDADR(R1) ;SET SER ON WRITE COMMAND
2991 020160 000402                BR       7$              ;GO HOME
2992 020162 005061 000014      6$:    CLR      CMDADR(R1)  ;CLEAR SER FLAG
2993 020166                7$:    POP      <R3,R1>      ;RESTORE REGISTERS
2994 020172 000207                RTS      PC              ;GO HOME
```

```
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
```

```
;*
;*BRUTUS SEQUENCE MODULE 3.1.1
;*
;*BEGINROUTINE (MOD - 3.1.1 BRUTUS SEQUENCE /BRTSEQ/)
;* IF INITIALIZE IS SET
;* : THEN-LOAD ADDRESS OF COMMANDS
;* : LOAD COUNTER WITH MAXIMUM COMMAND
;*
;* ENDF
;* LOAD COMMAND
;* CLEAR INITIALIZE
;* GET NEXT COMMAND ADDRESS
;* GET HOW MANY TIMES TO REPEAT COMMAND
;* LOAD RETRY
;* DECREMENT COUNT
;* IF COUNT IS ZERO
;* : THEN-CLEAR COMMAND
;* : RESET INITIALIZE
;* ENDF
;*ENDROUTINE
```

```

3017
3018
3019
3020 020174          BRTSEQ:
3021 020174 005737 002326 3$:   TST   INTLZE           ;SEE IF COMMAND POINTER SHOULD BE RESET
3022 020200 001406          BEQ   1$                ;BRANCH IF SEQUENCE IN PROGRESS
3023 020202 012737 002232 002616  MOV  #COMMD1,CMDSTG    ;TEMPORARY STORAGE FOR COMMAND
3024 020210 013737 002340 002620  MOV  CONTER,COUNT      ;COUNTER FOR MAXIMUM COMMANDS
3025 020216 017737 162374 002452 1$:   MOV  @CMDSTG,CMMAND    ;PUT COMMAND INTO EXECUTE LOCATION
3026 020224 005037 002326          CLR   INTLZE
3027 020230 062737 000002 002616  ADD  #2,CMDSTG         ;GET NEXT COMMAND
3028 020236 013737 002452 002462  MOV  CMMAND,RETRCD     ;STORE COMMAND TO BE EXECUTED IN
3029                                ;RETRY COMMAND LOCATION
3030 020244 017737 162346 002602  MOV  @CMDSTG,REPITN    ;
3031 020252 062737 000002 002616  ADD  #2,CMDSTG         ;GET NEXT COMMAND
3032 020260 005337 002620          DEC  COUNT             ;MAXIMUM LEFT TO BE ISSUED
3033 020264 001005          BNE  2$                ;BRANCH IF NOT DONE WITH SEQUENCE
3034 020266 005037 002452          CLR  CMMAND
3035 020272 012737 000001 002326  MOV  #1,INTLZE
3036 020300 000207          RTS   PC
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069 020302 005037 002224  UNTAPE: CLR  RALNTH      ;CLEAR RANDOM RECORD LENGTH
3070 020306 005737 002326          TST  INTLZE           ;SEE IF WENT THROUGH INIT CODE
3071 020312 001431          BEQ  1$                ;BRANCH IF NOT
3072 020314 005037 002326          CLR  INTLZE           ;CLEAR INIT CODE FLAG

```

```

;*
;*UNKNOWN TAPE MODULE 3.1.2
;*
;*BEGINROUTINE (MODULE 3.1.2 - UNKNOWN TAPE /UNTAPE/)
;* IF INITIALIZE SET
;* : THEN
;* : CLEAR INITIAILZE FLAG
;* : SELECT RETRIES
;* : SELECT RECORD LENGTH OF 2K BYTES
;* : CLEAR FILE #
;* : CLEAR RECORD #
;* : CLEAR DATA COMPARE
;* : LOAD REWIND COMMAND
;* : SET REPITITION OF COMMAND TO 1
;* : ELSE
;* : IF TAPE IS REWINDING FROM EOT
;* : : THEN
;* : : CLEAR EOT REWINDING FLAG
;* : : MOVE 0 INTO COMMAND FOR END OF PASS
;* : : ELSE
;* : : : LOAD READ FORWARD COMMAND
;* : : : INCREMENT RECORD #
;* : : : LOAD REPITITION OF COMMAND TO 1
;* : : : LOAD RETRY COMMAND
;* : : ENDIF
;* : ENDIF
;*ENDROUTINE

```



```

3073 020320 012737 000001 002216      MOV      #1,RETRY      ;SELECT RETRIES
3074 020326 012737 004000 002220      MOV      #4000,CHAREC ;SELECT 2K RECORDS
3075 020334 005037 002570          CLR      FILENM       ;CLEAR FILE COUNTER
3076 020340 005037 002566          CLR      RECORD       ;CLEAR RECORD COUNTER
3077 020344 005037 002226          CLR      DATCHK       ;NO DATA COMPARE
3078 020350 012737 000007 002452      MOV      #REWIND,CMMAND ;LOAD REWIND
3079 020356 013737 002452 002462      MOV      CMMAND,RETRCD
3080 020364 012737 000001 002602      MOV      #1,REPITN    ;LOAD REPITITION
3081 020372 000137 020450          JMP      3$           ;GET OUT
3082 020376 005737 002510          1$: TST      DENSWT     ;SEE IF REWINDING
3083 020402 001407          BEQ      2$           ;BRANCH IF NOT
3084 020404 005037 002510          CLR      DENSWT     ;RESET REWIND FLAG
3085 020410 012737 000000 002452      MOV      #0,CMMAND    ;SET UP FOR EOP
3086 020416 000137 020450          JMP      3$           ;GET OUT
3087 020422 012737 000071 002452      2$: MOV      #REDFWD,CMMAND ;SET UP READ COMMAND
3088 020430 005237 002566          INC      RECORD      ;INCREMENT RECORD COUNTER
3089 020434 012737 000001 002602      MOV      #1,REPITN    ;SET UP REPITITION
3090 020442 013737 002452 002462      MOV      CMMAND,RETRCD ;SET UP RETRY COMMAND
3091 020450 000207          3$: RTS      PC      ;GO HOME

```

3092  
3093  
3094  
3095  
3096  
3097  
3098  
3099  
3100  
3101  
3102  
3103  
3104  
3105  
3106  
3107  
3108  
3109  
3110  
3111  
3112  
3113  
3114  
3115  
3116  
3117  
3118  
3119  
3120  
3121  
3122  
3123  
3124  
3125  
3126  
3127  
3128

```

;*
;*FIELD SERVICE SEQUENCE #1
;*
;*BEGINROUTINE (MODULE 3.1.3 - FIELD SERVICE 1 /FIELD1/)
;* IF INITIALIZE SET
;* : THEN
;* : CLEAR INITIALIZE
;* : SET UP COUNT FOR DO LOOP
;* : ENABLE RETRIES
;* : SET RECORD LENGTH TO 2K BYTES
;* : SELECT RANDOM PATTERN
;* : SET RANDOM RECORD LENGTH
;* : SET UP HOW OFTEN PATTERN REPEATS ITSELF IN WORDS
;* : CLEAR FILE #
;* : ENABLE DATA COMPARE
;* : LOAD REWIND COMMAND
;* : CLEAR RECORD #
;* : LOAD REPITITION OF COMMAND
;* : GET ADDRESS OF WHERE COMMANDS START
;* : ELSE
;* : IF TAPE IS REWINDING FROM EOT
;* : THEN
;* : CLEAR RECORD #
;* : RESET COUNT FOR DO LOOP
;* : CLEAR EOT REWINDING FLAG
;* : IF DENSITY IS NOT GCR
;* : THEN
;* : : LOAD ADDRESS FOR GCR COMMANDS
;* : : ELSE
;* : : LOAD ADDRESS FOR PE
;* : : SET EOP FLAG
;* : : ENDDIF
;* : ELSE

```

```
3129 : * : : LOAD COMMAND
3130 : * : : GET ADDRESS OF NEXT COMMAND
3131 : * : : LOAD RETRY COMMAND
3132 : * : : LOAD REPITITION OF COMMAND TO 1
3133 : * : : IF WRITE COMMAND
3134 : * : : : THEN
3135 : * : : : INCREMENT RECORD #
3136 : * : : ENDFIF
3137 : * : : DECREMENT LOOP COUNTER
3138 : * : : IF LOOP COUNTER EQUAL TO ZERO
3139 : * : : : THEN
3140 : * : : : RESET LOOP COUNTER
3141 : * : : : RESET COMMAND ADDRESS
3142 : * : : ENDFIF
3143 : * : : ENDFIF
3144 : * ENDFIF
3145 : * ENDRoutine
```

```
3148 020452 005737 002326 FIELD1:
3149 020452 001452 TST INTLZE ;SEE IF WENT THRU INIT CODE
3150 020456 001452 BEQ 1$ ;BRANCH IF NOT
3151 020460 005037 002326 CLR INTLZE ;CLEAR INITIALIZE FLAG
3152 020464 012737 000003 002620 MOV #3,COUNT ;SET UP LOOP COUNTER
3153 020472 012737 000001 002216 MOV #1,RETRY ;RETRIES? YES
3154 020500 012737 004000 002220 MOV #4000,CHAREC ;2K SIZE RECORDS
3155 020506 012737 000010 002222 MOV #10,PATRN ;RANDOM PATTERN, YES
3156 020514 005237 002224 INC RALNTH ;SET RANDOM RECORD LENGTH
3157 020520 012737 000004 002350 MOV #4,PATLOP ;SET UP PATTERN LOOP COUNTER
3158 020526 005037 002570 CLR FILENM ;CLEAR FILE NUMBER
3159 020532 012737 000001 002226 MOV #1,DATCHK ;DATA CHECKED, YES
3160 020540 012737 000007 002452 MOV #REWIND,CMMAND ;LOAD REWIND COMMAND
3161 020546 013737 002452 002462 MOV CMMAND,RETRCD
3162 020554 005037 002566 CLR RECORD ;CLEAR RECORD NUMBER
3163 020560 012737 000001 002602 MOV #1,REPITN
3164 020566 012737 002766 002616 MOV #CANA1,CMDSTG ;LOAD PE SEQUENCE
3165 020574 005037 002444 CLR DENFLG ;SET PE STATISTIC FLAG
3166 020600 000137 020766 JMP 5$ ;GET OUT
3167 020604 005737 002510 1$: TST DENSWT ;TEST IF REWINDING
3168 020610 001433 BEQ 3$ ;BRANCH IF NOT
3169 020612 005037 002566 CLR RECORD ;RESET RECORD COUNT
3170 020616 012737 000003 002620 MOV #3,COUNT
3171 020624 005037 002510 CLR DENSWT ;RESET DENSITY SWITCH FLAG
3172 020630 022737 002776 002616 CMP #CANB1,CMDSTG ;SEE IF PE MODE
3173 020636 101406 BLOS 2$ ;BRANCH IF GCR
3174 020640 005237 002444 INC DENFLG ;SET GCR STAT FLAG
3175 020644 012737 002776 002616 MOV #CANB1,CMDSTG ;SET GCR MODE
3176 020652 000412 BR 3$
3177 020654 012737 002766 002616 2$: MOV #CANA1,CMDSTG ;SET PE MODE
3178 020662 005037 002444 CLR DENFLG ;SET PE STATISTIC FLAG
3179 020666 012737 000000 002452 MOV #0,CMMAND ;SET FOR END OF PASS
3180 020674 000137 020766 JMP 5$
3181 020700 017737 161712 002452 3$: MOV @CMDSTG,CMMAND ;LOAD COMMAND
3182 020706 062737 000002 002616 ADD #2,CMDSTG ;SET UP FOR NEXT COMMAND
3183 020714 013737 002452 002462 MOV CMMAND,RETRCD ;LOAD RETRY COMMAND
3184 020722 012737 000001 002602 MOV #1,REPITN ;LOAD REPITITION
```

GLOBAL AREAS  
ZTMJA3.P11

MACY11 30(1046)  
02-JUL-80 11:53

02-JUL-80 15:09 PAGE 3-35  
GLOBAL SUBROUTINES SECTION

SEQ 0058

3185	020730	022737	000067	002452		CMP	#67,CMMAND	;SEE IF WRITE
3186	020736	103402				BLO	4\$	;BRANCH IF NOT
3187	020740	005237	002566			INC	RECORD	;INCREMENT RECORD COUNT
3188	020744	005337	002620		4\$:	DEC	COUNT	;TEST LOOP
3189	020750	001006				BNE	5\$	;BRANCH IF NOT DONE
3190	020752	012737	000003	002620		MOV	#3,COUNT	;RESET TO START OF LOOP
3191	020760	162737	000006	002616		SUB	#6,CMDSTG	;RESET COMMAND ADDRESS
3192	020766	000207			5\$:	RTS	PC	;RETURN

3193  
3194  
3195  
3196  
3197  
3198  
3199  
3200  
3201  
3202  
3203  
3204  
3205  
3206  
3207  
3208  
3209  
3210  
3211  
3212  
3213  
3214  
3215  
3216  
3217  
3218  
3219  
3220  
3221  
3222  
3223  
3224  
3225  
3226  
3227  
3228  
3229  
3230  
3231  
3232  
3233  
3234  
3235  
3236  
3237  
3238  
3239  
3240

```

;*
;*FIELD SERVICE SEQUENCE #2
;*
;*BEGINROUTINE (MODULE 3.1.4 - FIELD SERVICE 2 /FIELD2/)
;* IF INITIALIZE SET
;* : THEN
;* :   CLEAR INITIALIZE
;* :   SET UP COUNT FOR DO LOOP
;* :   ENABLE RETRIES
;* :   SET RECORD LENGTH TO 2K BYTES
;* :   START PATTERN AT ZERO
;* :   CLEAR RANDOM RECORD LENGTH FLAG
;* :   SET UP HOW OFTEN PATTERN REPEATS ITSELF IN WORDS
;* :   CLEAR FILE #
;* :   ENABLE DATA COMPARE
;* :   LOAD REWIND COMMAND
;* :   CLEAR RECORD #
;* :   LOAD REPITITION OF COMMAND
;* :   GET ADDRESS OF WHERE COMMANDS START
;* : ELSE
;* :   IF TAPE IS REWINDING FROM EOT
;* :   : THEN
;* :   :   CLEAR RECORD #
;* :   :   RESET COUNT FOR DO LOOP
;* :   :   CLEAR EOT REWINDING FLAG
;* :   :   IF DENSITY IS NOT GCR
;* :   :   : THEN
;* :   :   :   LOAD ADDRESS FOR GCR COMMANDS
;* :   :   : ELSE
;* :   :   :   LOAD ADDRESS FOR PE
;* :   :   :   SET EOP FLAG
;* :   :   ENDIF
;* :   ELSE
;* :   :   LOAD COMMAND
;* :   :   GET ADDRESS OF NEXT COMMAND
;* :   :   LOAD RETRY COMMAND
;* :   :   LOAD REPITITION OF COMMAND TO 1
;* :   :   IF GCR TAPE MARK COMMAND
;* :   :   : THEN
;* :   :   :   MOVE ONE TO REPITITION
;* :   :   :   INCREMENT FILE #
;* :   :   : ELSE
;* :   :   :   IF SPACE REVERSE FILE
;* :   :   :   : THEN

```

```

3241      : * : : : : DECREMENT FILE #
3242      : * : : : : MOVE ONE TO REPITITION
3243      : * : : : : ELSE
3244      : * : : : : IF SPACE FORWARD FILE
3245      : * : : : : THEN
3246      : * : : : : INCREMENT FILE #
3247      : * : : : : MOVE ONE TO REPITITION
3248      : * : : : : ELSE
3249      : * : : : : MOVE 24 TO RECORDS IN FILE
3250      : * : : : : ENDIF
3251      : * : : : : ENDIF
3252      : * : : : : DECREMENT COMMAND COUNTER
3253      : * : : : : IF COMMAND COUNTER EQUAL ZERO
3254      : * : : : : THEN
3255      : * : : : : RESET COMMAND COUNTER
3256      : * : : : : RESET COMMAND ADDRESS
3257      : * : : : : GET NEXT PATTERN
3258      : * : : : : IF USER PATTERN
3259      : * : : : : THEN
3260      : * : : : : GET NEXT PATTERN
3261      : * : : : : ELSE
3262      : * : : : : IF RANDOM PATTERN
3263      : * : : : : THEN
3264      : * : : : : GET NEXT PATTERN
3265      : * : : : : ENDIF
3266      : * : : : : ENDIF
3267      : * : : : : IF LAST PATTERN
3268      : * : : : : THEN
3269      : * : : : : START PATTERN LOOP OVER
3270      : * : : : : ENDIF
3271      : * : : : : ENDIF
3272      : * : : : : ENDIF
3273      : * : : : : ENDIF
3274      : * : : : : ENDIF
3275      : * : : : : *ENDROUTINE
3276

```

```

3277 020770 005737 002326 FIELD2: TST INTLZE ;SEE IF WENT THRU INIT CODE
3278 020774 001452 BEQ 1$ ;BRANCH IF NOT
3279 020776 005037 002326 CLR INTLZE ;CLEAR INITIALIZE FLAG
3280 021002 012737 000006 002620 MOV #6,COUNT ;SET UP LOOP COUNTER
3281 021010 012737 000001 002216 MOV #1,RETRY ;RETRIES? YES
3282 021016 012737 004000 002220 MOV #4000,CHAREC ;2K SIZE RECORDS
3283 021024 005037 002222 CLR PATRN ;START WITH ZERO PATTERN
3284 021030 005037 002224 CLR RALNTH ;CLEAR RECORD LENGTH
3285 021034 012737 000004 002350 MOV #4,PATLOP ;SET UP PATTERN LOOP COUNTER
3286 021042 005037 002570 CLR FILENM ;CLEAR FILE NUMBER
3287 021046 012737 000001 002226 MOV #1,DATCHK ;DATA CHECKED, YES
3288 021054 012737 000007 002452 MOV #REWIND,CMMAND ;LOAD REWIND COMMAND
3289 021062 013737 002452 002462 MOV CMMAND,RETRCD
3290 021070 012737 000001 002566 MOV #1,RECORD ;SET RECORD NUMBER
3291 021076 012737 000001 002602 MOV #1,REPITN
3292 021104 005037 002444 CLR DENFLG ;SET PE STATISTIC FLAG
3293 021110 012737 003004 002616 MOV #CAN2,CMDSTG ;LOAD PE SEQUENCE
3294 021116 000137 021452 JMP 5$ ;GET OUT
3295 021122 005737 002510 1$: TST DENSWT ;TEST IF REWINDING
3296 021126 001440 BEQ 3$ ;BRANCH IF NOT

```

```

3297 021130 005037 002222          CLR      PATTRN          ;START WITH ZERO PATTERN AFTER REWIND
3298 021134 012737 000001 002566    MOV      #1,RECORD      ;RESET RECORD NUMBER
3299 021142 005037 002570          CLR      FILENM
3300 021146 012737 000006 002620    MOV      #6,COUNT
3301 021154 005037 002510          CLR      DENSWT        ;RESET DENSITY SWITCH FLAG
3302 021160 022737 003022 002616    CMP      #CANB2,CMDSTG  ;SEE IF PE MODE WAS SET
3303 021166 101406          BLOS    2$             ;BR IF GCR WAS SET
3304 021170 005237 002444          INC      DENFLG        ;SET GCR STATISTIC FLAG
3305 021174 012737 003022 002616    MOV      #CANB2,CMDSTG  ;SET GCR MODE
3306 021202 000412          BR      3$
3307 021204 012737 003004 002616 2$:  MOV      #CANA2,CMDSTG  ;SET PE MODE
3308 021212 005037 002444          CLR      DENFLG        ;SET PE STATISTIC FLAG
3309 021216 012737 000000 002452    MOV      #0,CMMAND     ;SET FOR END OF PASS
3310 021224 000137 021452          JMP      5$
3311 021230 017737 161362 002452 3$:  MOV      @CMDSTG,CMMAND ;LOAD COMMAND
3312 021236 062737 000002 002616    ADD     #2,CMDSTG      ;SET UP FOR NEXT COMMAND
3313 021244 013737 002452 002462    MOV     CMMAND,RETRCD  ;LOAD RETRY COMMAND
3314 021252 022737 000017 002452    CMP     #WTMGCR,CMMAND ;SEE IF WRITE TM COMMAND
3315 021260 103406          BLO     6$             ;BRANCH IF NOT
3316 021262 012737 000001 002602    MOV     #1,REPITN     ;SET REPITITION TO ONE
3317 021270 005237 002570          INC     FILENM        ;INCREMENT FILE NUMBER
3318 021274 000427          BR      4$
3319 021276 022737 000027 002452 6$:  CMP     #SPRVFL,CMMAND ;SEE IF SPACE REVERSE FILE
3320 021304 001006          BNE     7$             ;BRANCH IF NOT
3321 021306 005337 002570          DEC     FILENM
3322 021312 012737 000001 002602    MOV     #1,REPITN     ;SET REPITITION TO ONE
3323 021320 000415          BR      4$
3324 021322 022737 000025 002452 7$:  CMP     #SPFWFL,CMMAND ;SEE IF SPACE FORWARD COMMAND
3325 021330 001006          BNE     10$            ;BRANCH IF NOT
3326 021332 005237 002570          INC     FILENM
3327 021336 012737 000001 002602    MOV     #1,REPITN
3328 021344 000403          BR      4$
3329 021346 012737 000030 002602 10$: MOV     #30,REPITN    ;LOAD REPITITION
3330 021354 005337 002620 4$:  DEC     COUNT         ;TEST LOOP
3331 021360 001034          BNE     5$             ;BRANCH IF NOT DONE
3332 021362 012737 000006 002620    MOV     #6,COUNT      ;RESET TO START OF LOOP
3333 021370 162737 000014 002616    SUB     #14,CMDSTG    ;RESET COMMAND ADDRESS
3334 021376 005237 002222          INC     PATTRN        ;GET NEXT PATTERN
3335 021402 022737 000006 002222    CMP     #6,PATTRN     ;SEE IF USER PATTERN
3336 021410 001003          BNE     8$             ;BRANCH IF NO
3337 021412 005237 002222          INC     PATTRN        ;GET NEXT PATTERN
3338 021416 000415          BR      5$
3339 021420 022737 000010 002222 8$:  CMP     #10,PATTRN    ;SEE IF RANDOM PATTERN
3340 021426 001003          BNE     9$             ;BRANCH IF NO
3341 021430 005237 002222          INC     PATTRN        ;GET NEXT PATTERN
3342 021434 000406          BR      5$
3343 021436 022737 000011 002222 9$:  CMP     #11,PATTRN    ;SEE IF LAST PATTERN
3344 021444 101002          BHI     5$             ;BRANCH IF NO
3345 021446 005037 002222          CLR     PATTRN        ;START AT FIRST PATTERN
3346 021452 000207          RTS     PC             ;RETURN
3347
3348
3349
3350
3351
3352

```

```

3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408

```

```

: *
: * FUNCTIONAL TESTS -- CANNED SEQUENCE 4
: *
: * BEGINROUTINE (MODULE 3.1.5 - FUNCTIONAL TESTS /FUNCT/)
: *
: * IF INITIALIZE SET
: * : THEN
: * :   CLEAR INITIALIZE
: * :   CLEAR RANDOM RECORD FLAG
: * :   CLEAR FILE #
: * :   CLEAR RECORD #
: * :   DISABLE DATA CHECKING
: * :   SET UP ONE REPITION PER COMMAND
: * :   ENABLE RETRIES
: * :   LOAD FUNCTION TABLE COMMAND LIST
: * :   200 BYTES PER RECORD
: * :   GO TO CORRECT FUNCTION TEST
: * :   LOAD RETRY COMMAND
: * :   RETURN TO SEQDEF
: * : ELSE
: * :   GO TO CORRECT FUNCTION TEST
: * :   LOAD RETRY COMMAND
: * :   RETURN TO SEQDEF
: * ENDIF
: * ENDROUTINE

```

```

FUNCT:
      PUSH      <R1>
      TST       INTLZE      ;SEE IF WENT THRU INIT CODE
      BEQ       1$         ;BRANCH IF NOT
      CLR       INTLZE      ;CLEAR INITIALIZE FLAG
      MOV       #1,RETRY    ;RETRIES? YES
      MOV       #200,CHAREC ;START WITH 200 BYTE RECORDS
      CLR       RALNTH      ;CLEAR RANDOM RECORD LENGTH
      CLR       RECORD      ;CLEAR RECORD NUMBER
      CLR       FILENM      ;CLEAR FILE NUMBER
      CLR       DATCHK      ;DATA CHECKED, NO
      CLR       MESNUM      ;SET MESSAGE POINTER
      MOV       #FCNTBL,CMDSTG ;LOAD FUNCTION TABLE ADDRESS
      1$: MOV     @CMDSTG,R1  ;GET FUNCTION TABLE ADDRESS
      JSR       PC,(R1)     ;GO TO CORRECT FUNCTION TEST
      MOV       CMMAND,RETRCD
      POP       <R1>
      RTS       PC          ;RETURN

```

```

: * BEGINSUBROUTINE (NOOP FUNCTIONAL TEST -- F.NOP)
: *
: * SET REPEAT COUNT TO 10
: * LOAD NOOP COMMAND
: * INCREMENT POINTER TO NEXT TEST
: * RETURN
: * ENDSUBROUTINE

```

```

F.NOP:
      MOV       #10,REPITN  ;NOOP TEST
                        ;REPEAT 10 TIMES

```

```

021454
021454
021456 005737 002326
021462 001425
021464 005037 002326
021470 012737 000001 002216
021476 012737 000200 002220
021504 005037 002224
021510 005037 002566
021514 005037 002570
021520 005037 002226
021524 005037 002442
021530 012737 003036 002616
021536 017701 161054 1$:
021542 004711
021544 013737 002452 002462
021552
021554 000207

```

```
3409 021564 012737 000003 002452      MOV      #NOOP,CMMAND      ;LOAD COMMAND
3410 021572 005037 002442              CLR      MESNUM           ;SET MESSAGE POINTER
3411 021576 062737 000002 002616      ADD      #2,CMDSTG        ;NEXT STAGE
3412 021604              RETURN                    ;
3413
3414      ;*BEGINSUBROUTINE (REWIND FUNCTIONAL TEST -- -- F.REW)
3415      ;*
3416      ;* SET REPEAT COUNT OF 1
3417      ;* LOAD REWIND COMMAND
3418      ;* SET POINTER TO NEXT STAGE
3419      ;* RETURN
3420      ;*ENDSUBROUTINE
3421
3422      ;*BEGINSUBROUTINE (SENSE COMMAND ISSUED TO CHECK BOT STATUS AFTER REWIND)
3423      ;*
3424      ;* LOAD SENSE COMMAND
3425      ;* SET REPITION COUNT TO 1
3426      ;* SET CHECK BOT WHEN DONE FLAG
3427      ;* SET POINTER TO NEXT TEST
3428      ;* RETURN
3429      ;*ENDSUBROUTINE
3430
3431 021606      F.REW:              ;REWIND TEST
3432 021606 012737 000001 002602      MOV      #1,REPITN        ;ONE TIME
3433 021614 012737 000007 002452      MOV      #REWIND,CMMAND   ;LOAD REWIND COMMAND
3434 021622 062737 000002 002616      ADD      #2,CMDSTG        ;NEXT TEST
3435 021630      RETURN
3436
3437 021632      F.SNS:              ;ISSUE SENSE TO CHECK BOT STATUS
3438 021632 012737 000002 002442      MOV      #2,MESNUM        ;SET MESSAGE POINTER
3439 021640 012737 000011 002452      MOV      #SENSE,CMMAND   ;LOAD COMMAND
3440 021646 012737 177777 002434      MOV      #-1,CHKBOT      ;SET FLAG
3441 021654 062737 000002 002616      ADD      #2,CMDSTG        ;NEXT TEST
3442 021662 012737 000001 002602      MOV      #1,REPITN        ;ONE TIME ONLY
3443 021670      RETURN
3444
3445
3446      ;*BEGINSUBROUTINE (READ/WRITE FUNCTIONAL SUBROUTINES)
3447      ;*
3448      ;*BEGINROUTINE (WRITE)
3449      ;*
3450      ;* IF WRITE PE
3451      ;* : THEN
3452      ;* :   LOAD WRITE PE COMMAND
3453      ;* :   USE 1000 BYTES PER RECORD
3454      ;* :   SET REPITION TO 200
3455      ;* : ELSE
3456      ;* :   LOAD WRITE GCR COMMAND
3457      ;* :   USE 1000 BYTES/RECORD
3458      ;* :   USE 1000 RECORDS
3459      ;* ENDIF
3460      ;* CLEAR CHECK BOT STATUS FLAG FROM LAST TEST
3461      ;* USE DATA PATTERN OF ALL ONES
3462      ;* CLEAR RECORD #
3463      ;* CLEAR FILE #
3464      ;* SET POINTER TO NEXT STAGE
```

```

3465      ;*ENDROUTINE
3466      ;*
3467      ;*BEGINROUTINE (READ SUBROUTINES)
3468      ;*
3469      ;* IF READ REVERSE
3470      ;* : THEN
3471      ;* :   LOAD READ REVERSE COMMAND
3472      ;* : ELSE
3473      ;* :   LOAD READ FORWARD COMMAND
3474      ;* ENDIF
3475      ;* CLEAR FILE NUMBER
3476      ;* CLEAR RECORD NUMBER
3477      ;* SET POINTER TO NEXT TEST
3478      ;* RETURN
3479      ;*ENDROUTINE
3480      ;*
3481      ;*ENDSUBROUTINE
3482
3483      021672      F.WPE:      ;WRITE PE
3484      021672      005037      002444      CLR      DENFLG      ;USE PE STAT TABLE
3485      021676      012737      000004      002442      MOV      #4,MESNUM      ;SET MESSAGE POINTER
3486      021704      012737      000061      002452      MOV      #WRTPE,CMMAND ;LOAD COMMAND
3487      021712      012737      001000      002220      MOV      #1000,CHAREC ;1000 BYTES/REC
3488      021720      012737      000200      002602      MOV      #200,REPITN  ;200 RECORDS
3489      021726      000413      BR      F.WCOM      ;GO TO COMMON CODE
3490
3491      021730      F.WGCR:      ;WRITE GCR
3492      021730      005237      002444      INC      DENFLG      ;SET GCR STAT FLAG
3493      021734      012737      000063      002452      MOV      #WRTGCR,CMMAND ;LOAD COMMAND
3494      021742      012737      001000      002220      MOV      #1000,CHAREC ;1000 BYTES/REC
3495      021750      012737      001000      002602      MOV      #1000,REPITN  ;1000 RECORDS
3496      ; BR      F.WCOM      ;GO TO COMMON CODE
3497
3498      021756      012737      000001      002222      F.WCOM: MOV      #1,PATRN      ;PATTER OF ALL ONES
3499      021764      005037      002434      CLR      CHKBOT      ;CLEAR THIS FLAG
3500      021770      005037      002566      CLR      RECORD      ;CLEAR REC #
3501      021774      005037      002570      CLR      FILENM      ;CLEAR FILE #
3502      022000      062737      000002      002616      ADD      #2,CMDSTG      ;NEXT STAGE
3503      022006      RETURN
3504
3505      022010      012737      000077      002452      F.RDR1: MOV      #REDREV,CMMAND ;LOAD READ REVERSE
3506      022016      000403      BR      F.RD1      ;GO TO COMMON CODE
3507      022020      012737      000071      002452      F.RDF1: MOV      #REDFWD,CMMAND ;LOAD READ FORWARD
3508      022026      012737      001000      002220      F.RD1:  MOV      #1000,CHAREC ;1000 BYTES/REC
3509      022034      012737      000200      002602      MOV      #200,REPITN  ;200 RECORDS
3510      022042      062737      000002      002616      ADD      #2,CMDSTG      ;NEXT STAGE
3511      022050      RETURN
3512
3513      ; READ GCR MODE
3514
3515      022052      012737      000077      002452      F.RDR2: MOV      #REDREV,CMMAND ;LOAD READ REVERSE
3516      022060      000403      BR      F.RD2      ;GO TO COMMON CODE
3517      022062      012737      000071      002452      F.RDF2: MOV      #REDFWD,CMMAND ;LOAD READ FORWARD
3518      022070      012737      001000      002220      F.RD2:  MOV      #1000,CHAREC ;1000 BYTES/REC
3519      022076      012737      001000      002602      MOV      #1000,REPITN  ;1000 RECORDS
3520      022104      062737      000002      002616      ADD      #2,CMDSTG      ;NEXT COMMAND

```



3521 022112

RETURN

3522  
3523  
3524  
3525  
3526  
3527  
3528  
3529  
3530  
3531  
3532;\*BEGINSUBROUTINE (WRITE CHECK FUNCTIONAL TEST -- F.WC)  
;\* REWIND TAPE TO BOT  
;\* WRITE A RECORD 100 BYTES LONG IN PE MODE  
;\* ISSUE A WRITE CHECK REVERSE  
;\* ISSUE A WRITE CHECK FORWARD  
;\* ASSURE "WCE" DOES NOT SET  
;\* REPEAT FOR GCR MODE  
;\*ENDSUBROUTINE3533  
3534  
3535  
3536  
3537  
3538  
3539  
3540  
3541  
3542  
3543  
3544  
3545022114 012737 000006 002442 F.WC1: MOV #6,MESNUM ;SET MESSAGE POINTER  
022122 005737 002510 TST DENSWT ;GCR ?  
022126 001006 BNE 1\$ ;YES  
022130 005037 002444 CLR DENFLG ;SET PE STAT FLAG  
022134 012737 000061 002452 MOV #WRTPE,CMMAND ;LOAD WRITE COMMAND  
022142 000405 BR 2\$ ;BRANCH  
022144 012737 000063 002452 1\$: MOV #WRTGCR,CMMAND ;LOAD WRITE GCR COMMAND  
022152 005237 002444 INC DENFLG ;SET GCR STAT FLAG  
022156 012737 004000 002220 2\$: MOV #4000,CHAREC ;4000 BYTES/REC  
022164 012737 000001 002602 MOV #1,REPITN ;ONE TIME  
022172 062737 000002 002616 ADD #2,CMDSTG ;NEXT STAGE  
022200 RETURN3546  
3547  
3548  
3549  
3550022202 012737 000001 002602 F.WC2: MOV #1,REPITN ;ONE TIME  
022210 012737 000057 002452 MOV #WCKREV,CMMAND ;LOAD WRITE CHECK REVERSE  
022216 062737 000002 002616 ADD #2,CMDSTG ;NEXT STAGE  
022224 RETURN ;RETURN3551  
3552  
3553  
3554  
3555  
3556  
3557  
3558  
3559  
3560  
3561022226 005737 002510 F.WC3: TST DENSWT ;WHICH DENSITY  
022232 001406 BEQ 1\$ ;GCR NOW  
022234 062737 000002 002616 ADD #2,CMDSTG ;NEXT TEST  
022242 005037 002510 CLR DENSWT ;CLEAR FLAG  
022246 000405 BR 4\$ ;EXIT  
022250 005237 002510 1\$: INC DENSWT ;SET GCR  
022254 012737 003070 002616 MOV #F.WC,CMDSTG ;AGAIN  
022262 012737 000001 002602 4\$: MOV #1,REPITN ;ONE TIME  
022270 012737 000051 002452 MOV #WCKFWD,CMMAND ;WRITE CHECK FORWARD  
022276 RETURN3562  
3563  
3564  
3565  
3566  
3567  
3568  
3569  
3570  
3571  
3572  
3573  
3574  
3575  
3576;\*BEGINROUTINE (SPACING FUNCTIONAL TEST -- F.SP1)  
;\*  
;\*BEGINSUBROUTINE (REWIND TAPE)  
;\*  
;\* CALL F.REW TO REWIND THE TAPE  
;\* SET RECORD LENGTH COUNTER TO 100  
;\* SET POINTER TO NEXT STAGE  
;\* RETURN  
;\*ENDSUBROUTINE;\*BEGINSUBROUTINE (FORMAT TAPE FOR SPACING TEST -- F.SP2)  
;\*  
;\* SET REPEAT COUNT TO 1  
;\* LOAD RECORD LENGTH FROM COUNTER "F.TMP1"

```
3577      :* IF DENSWT IS ZERO
3578      :*   THEN
3579      :*   :   LOAD WRITE PE COMMAND
3580      :*   :   ELSE
3581      :*   :   LOAD WRITE GCR COMMAND
3582      :*   ENDIF
3583      :* INCREMENT COUNTER 'F.TMP1' FOR NEXT RECORD LENGTH
3584      :* IF 'F.TMP1' IS GREATER THAN 200
3585      :*   THEN
3586      :*   :   SET PROGRAM POINTER TO NEXT STAGE
3587      :*   :   LOAD COUNTERS FOR NEXT STAGE
3588      :*   :   F.TMP1 WITH 77
3589      :*   :   F.TMPR WITH 100
3590      :*   :   F.TMPF WITH 176
3591      :*   ELSE
3592      :*   :   RETURN
3593      :*   ENDIF
3594      :*ENDSUBROUTINE
3595
3596      :*BEGINSUBROUTINE (DO SPACING VARYING RECORD LENGTHS)
3597      :*
3598      :* (*SPACE REVERSE -- F.SP3*)
3599      :*
3600      :* LOAD SPACE REVERSE COMMAND
3601      :* SET REPITION FROM COUNTER 'F.TMP1'
3602      :* SET POINTER TO NEXT STAGE
3603      :* RETURN
3604      :*
3605      :* (*READ REVERSE -- F.SP4*)
3606      :*
3607      :* LOAD READ REVERSE COMMAND
3608      :* LOAD RECORD LENGTH FROM COUNTER F.R
3609      :* SET REPEAT COUNT TO 1
3610      :* SET NEXT STAGE POINTER
3611      :*
3612      :* (*SPACE FORWARD RECORD -- F.SP5*)
3613      :*
3614      :* DECREMENT SPACING COUNTER
3615      :* IF SPACING COUNTER (F.A) IS = 0
3616      :*   THEN
3617      :*   :   EXIT TEST
3618      :*   :   ELSE
3619      :*   :   LOAD SPACE FORWARD COMMAND
3620      :*   :   LOAD REPEAT COUNT FROM COUNTER F.A
3621      :*   :   SET POINTER TO NEXT STAGE
3622      :*   :   RETURN
3623      :*   ENDIF
3624      :*
3625      :* (*READ FORWARD 1 RECORD -- F.SP6*)
3626      :*
3627      :* LOAD READ FORWARD COMMAND
3628      :* SET REPEAT COUNT TO 1
3629      :* SET RECORD LENGTH FROM COUNTER F.F
3630      :* DECREMENT SPACE COUNTER F.A
3631      :* IF F.A = 0
3632      :*   THEN
```

```

3633      :*      : SET POINTER TO NEXT TEST
3634      :*      : ELSE
3635      :*      : SET POINTER TO SPACE REVERSE SUBTEST
3636      :*      : ENDIF
3637      :*      : *ENDSUBROUTINE
3638
3639      :      : TEMPORARY LOCATIONS FOR SPACING TEST
3640
3641      022300 000000
3642      022302 000000
3643      022304 000000
3644
3645
3646      :      : SPACING FUNCTIONAL ROUTINES
3647
3648      022306
3649      022312 012737 000010 002442
3650      022320 012737 000100 022300
3651      022326
3652
3653      022330 012737 000001 002602
3654      022336 013737 022300 002220
3655      022344 005737 002510
3656      022350 001006
3657      022352 005037 002444
3658      022356 012737 000061 002452
3659      022364 000405
3660      022366 012737 000063 002452
3661      022374 005237 002444
3662      022400 005237 022300
3663      022404 022737 000200 022300
3664      022412 003401
3665      022414
3666      022416 062737 000002 002616
3667      022424 012737 000077 022300
3668      022432 012737 000100 022302
3669      022440 012737 000176 022304
3670      022446
3671
3672      022450 012737 000023 002452
3673      022456 013737 022300 002602
3674      022464 062737 000002 002616
3675      022472
3676
3677      022474 012737 000077 002452
3678      022502 013737 022302 002220
3679      022510 012737 000001 002602
3680      022516 062737 000002 002616
3681      022524
3682
3683      022526 005337 022300
3684      022532 001441
3685      022534 012737 000021 002452
3686      022542 013737 022300 002602
3687      022550 062737 000002 002616
3688      022556

F.TMP1: .WORD 0
F.TMPR: .WORD 0
F.TMPF: .WORD 0

F.SP1: CALL F.REW ;USE REWIND CODE
      MOV #10,MESNUM ;SET MESSAGE POINTER
      MOV #100,F.TMP1 ;SET UP TEMP FOR NEXT STAGE
      RETURN

F.SP2: MOV #1,REPITN ;ONLY ONCE
      MOV F.TMP1,CHAREC ;LOAD BYTES/REC
      TST DENSWT ;WHICH DENSITY
      BNE 1$ ;BR FOR GCR
      CLR DENFLG ;SET PE STAT FLAG
      MOV #WRTPE,CMMAND ;LOAD PE WRITE
      BR 2$ ;JUMP OVER SET GCR
1$: MOV #WRTGCR,CMMAND ;LOAD GCR WRITE
      INC DENFLG ;SET GCR STAT FLAG
2$: INC F.TMP1 ;INCREASE RECORD SIZE
      CMP #200,F.TMP1 ;200 BYTES/REC YET ?
      BLE 3$ ;YES , QUIT
      RETURN ;NO , HANG IN THERE
3$: ADD #2,CMDSTG ;NEXT SECTION
      MOV #77,F.TMP1 ;SET UP COUNTERS FOR NEXT PART
      MOV #100,F.TMPR ;REVERSE READ SIZE
      MOV #176,F.TMPF ;FORWARD READ SIZE
      RETURN

F.SP3: MOV #SPRVRC,CMMAND ;LOAD SPACE REV RECORD
      MOV F.TMP1,REPITN ;REPEAT F.TMP1 TIMES
      ADD #2,CMDSTG ;NEXT PART
      RETURN

F.SP4: MOV #REDREV,CMMAND ;LOAD READ REVERSE
      MOV F.TMPR,CHAREC ;LOAD BYTES/REC
      MOV #1,REPITN ;ONE RECORD
      ADD #2,CMDSTG ;NEXT PART
      RETURN

F.SP5: DEC F.TMP1 ;DECREASE SPACE COUNT
      BEQ F.SPDN ;IF ZERO QUIT
      MOV #SPFWRC,CMMAND ;LOAD SPACE FORWARD COMMAND
      MOV F.TMP1,REPITN ;LOAD SPACE COUNT
      ADD #2,CMDSTG ;NEXT STAGE
      RETURN

```

```

3689
3690 022560 012737 000071 002452 F.SP6: MOV #REDFWD,CMMAND ;LOAD READ FORWARD COMMAND
3691 022566 012737 000001 002602 MOV #1,REPITN ;ONE READ
3692 022574 013737 022304 002220 MOV F.TMPF,CHAREC ;LOAD BYTES/REC
3693 022602 005237 022302 INC F.TMPR ;DEC REVERSE COUNTER
3694 022606 005337 022304 DEC F.TMPF ;DECREMENT FORWARD COUNTER
3695 022612 005337 022300 DEC F.TMP1 ;DECREMENT SPACE COUNTER
3696 022616 022737 000001 022300 CMP #1,F.TMP1 ;DONE ?
3697 022624 001404 BEQ F.SPDN ;YES , EXIT
3698 022626 012737 003104 002616 MOV #F.SPRV,CMDSTG ;LOOP UNTIL DONE
3699 022634 RETURN
3700
3701 ; ONCE SPACING FACTOR GOES TO ZERO
3702
3703 022636 005737 002510 F.SPDN: TST DENSWT ;PE ?
3704 022642 001006 BNE 1$ ;NO,SET GCR
3705 022644 005237 002510 INC DENSWT ;
3706 022650 012737 003100 002616 MOV #F.SP,CMDSTG ;RESET SEQUENCE ADDRESS
3707 022656 RETURN
3708 022660 005037 002510 1$: CLR DENSWT ;CLEAR FLAG
3709 022664 012737 003114 002616 MOV #F.SPND+2,CMDSTG ;NEXT TEST
3710 022672 RETURN
3711
3712 ;*BEGINSUBROUTINE (TAPE MARK WRITE/READ)
3713 ;* REWIND TO BOT
3714 ;* ISSUE A WRITE TAPE MARK COMMAND PE
3715 ;* SPACE REVERSE FILE (TO TAPE MARK)
3716 ;* SPACE FORWARD FILE (TO TAPE MARK)
3717 ;* REPEAT FOR GCR TAPE MARK
3718 ;*ENDSUBROUTINE
3719
3720 ; TAPE MARK WRITE / READ FUNCTIONAL TEST
3721
3722
3723 022674 012737 000012 002442 F.TM1: MOV #12,MESNUM ;SET MESSAGE POINTER
3724 022702 005737 002510 TST DENSWT ;WHICH DENSITY
3725 022706 001406 BEQ 1$ ;BRANCH FOR PE
3726 022710 005237 002444 INC DENFLG ;SET GCR STAT FLAG
3727 022714 012737 000017 002452 MOV #WTMGCR,CMMAND ;LOAD GCR TAPE MARK COMMAND
3728 022722 000405 BR 2$ ;
3729 022724 012737 000015 002452 1$: MOV #WTMPE,CMMAND ;LOAD WRITE PE TAPE MARK
3730 022732 005037 002444 CLR DENFLG ;SET PE STAT FLAG
3731 022736 012737 000001 002602 2$: MOV #1,REPITN ;ONE TIME
3732 022744 005037 002566 CLR RECORD ;
3733 022750 005037 002570 CLR FILENM ;FILE # 0
3734 022754 005037 002220 CLR CHAREC ;SET TO ZERO
3735 022760 062737 000002 002616 ADD #2,CMDSTG ;SET NEXT PART
3736 022766 RETURN
3737
3738 ; SPACE REVERSE TAPE MARK
3739
3740 022770 012737 000027 002452 F.TM2: MOV #SPRVFL,CMMAND ;LOAD COMMAND
3741 022776 012737 000001 002602 MOV #1,REPITN ;ONE TIME
3742 023004 062737 000002 002616 ADD #2,CMDSTG ;NEXT
3743 023012 RETURN
3744

```

```

3745 ; SPACE FORWARD TAPE MARK
3746
3747 023014 012737 000025 002452 F.TM3: MOV #SPFWFL,CMMAND ;LOAD FORWARD SPACE
3748 023022 012737 000001 002602 MOV #1,REPITN ;ONE TAPE MARK
3749 023030 005737 002510 TST DENSWT ;CHANGE DENSITIES
3750 023034 001406 BEQ 1$ ;YES , USE GCR NOW
3751 023036 062737 000002 002616 ADD #2,CMDSTG ;NO QUIT
3752 023044 005037 002510 CLR DENSWT ;CLEAR FLAG
3753 023050 000405 BR 2$ ;QUIT
3754 023052 005237 002510 1$: INC DENSWT ;SET GCR MODE
3755 023056 162737 000006 002616 SUB #6,CMDSTG ;LOOP BACK
3756 023064 2$: RETURN
3757
3758 ;*BEGINSUBROUTINE (TAPE MARK SPACE TEST)
3759 ;* REWIND TAPE
3760 ;* WRITE ON TAPE THE FOLLOWING PATTERN
3761 ;*
3762 ;* TM:20 REC:TM:40 REC:TM:60 REC:TM:80 REC:TM:
3763 ;*
3764 ;* SPACE REVERSE 200 RECORDS
3765 ;* TAPE SHOULD STOP ON EACH RECORD
3766 ;* 5 SPACE REVERSE COMMANDS WILL HAVE TO BE ISSUED
3767 ;* BOT SHOULD NEVER BE REACHED
3768 ;* REPEAT FOR GCR MODE
3769 ;*ENDSUBROUTINE
3770
3771 023066 F.TMS1: CALL F.REW ;REWIND TAPE
3772 023072 012737 000024 022300 MOV #20.,F.TMP1 ;LOAD RECORD COUNT
3773 023100 RETURN
3774
3775 ; FORMAT THE TAPE
3776
3777 023102 F.TMS2: CALL F.TM1 ;WRITE A TAPE MARK
3778 023106 RETURN
3779
3780 023110 012737 000014 002442 F.TMS3: MOV #14,MESNUM ;SET MESSAGE POINTER
3781 023116 012737 000061 002452 MOV #WRTPE,CMMAND ;LOAD WRITE COMMAND
3782 023124 013737 022300 002602 MOV F.TMP1,REPITN ;LOAD RECORD COUNT
3783 023132 012737 001000 002220 MOV #1000,CHAREC ;1000 BYTES / REC
3784 023140 062737 000024 022300 ADD #20.,F.TMP1 ;INCREMENT RECORD COUNT
3785 023146 022737 000144 022300 CMP #100.,F.TMP1 ;DONE ?
3786 023154 002007 BGE 1$ ;NO , LOOP
3787 023156 062737 000002 002616 ADD #2,CMDSTG ;YES
3788 023164 012737 000310 022300 MOV #200.,F.TMP1 ;SET UP TOTAL RECORD COUNT
3789 023172 000403 BR 2$
3790 023174 162737 000002 002616 1$: SUB #2,CMDSTG ;BACK TO WRITE A TM
3791 023202 2$: RETURN
3792
3793 ; HERE WE SPACE REVERSE
3794
3795 023204 012737 177777 002436 F.TMS4: MOV #-1,CHKUTM ;SET EXPECT UN-EXPECTED TM
3796 023212 012737 000001 002602 MOV #1,REPITN ;ONE RECORD
3797 023220 012737 000023 002452 MOV #SPRVRC,CMMAND ;LOAD SPACE REVERSE COMMAND
3798 023226 005337 022300 DEC F.TMP1 ;DECREMENT RECORD COUNT
3799 023232 001020 BNE 3$ ;GO DO IT
3800 023234 005037 002436 CLR CHKUTM ;CLEAR UN-EXPECTED TM FLAG

```

```

3801 023240 005737 002510          TST    DENSWT          ;DID WE DO GCR YET
3802 023244 001006                   BNE    2$             ;YES , BRANCH
3803 023246 005237 002510          INC    DENSWT          ;SET GCR NOW
3804 023252 012737 003124 002616  MOV    #F.TMS,CMDSTG  ;DO IT AGAIN
3805 023260 000405                   BR     3$             ;RETURN
3806 023262 005037 002510          2$:   CLR    DENSWT          ;RESET FLAG
3807 023266 062737 000002 002616  ADD    #2,CMDSTG      ;NEXT TEST
3808 023274                   3$:   RETURN
3809
3810
3811                               ;*BEGINSUBROUTINE (WORST CASE VACUUM COLUMN TEST)
3812                               ;* REWIND
3813                               ;* FORMAT TAPE WITH 25 RECORDS OF 20 BYTES EACH
3814                               ;* REWIND TAPE
3815                               ;* CLEAR SPACE COUNT
3816                               ;* SPACE FORWARD "SPACE COUNT + 1" TIMES
3817                               ;* IF SPACE COUNT > 25
3818                               ;* THEN
3819                               ;* REWIND
3820                               ;* SPACE FORWARD "SPACE COUNT - 1" TIMES
3821                               ;* IF SPACE COUNT = 0
3822                               ;* THEN
3823                               ;* REPEAT FOR GCR MODE
3824                               ;* ELSE
3825                               ;* REWIND
3826                               ;* ENDIF
3827                               ;* ELSE
3828                               ;* REWIND
3829                               ;* ENDIF
3830                               ;*ENDSUBROUTINE
3831
3832 023276 012737 000016 002442  F.VC1:  MOV    #16,MESNUM      ;SET MESSAGE POINTER
3833 023304 005737 002510          TST    DENSWT          ;WHICH DENSITY
3834 023310 001006                   BNE    1$             ;BR FOR GCR
3835 023312 005037 002444          CLR    DENFLG          ;SET PE STAT FLAG
3836 023316 012737 000061 002452  MOV    #WRTPE,CMMAND  ;PE WRITE
3837 023324 000405                   BR     2$             ;ENTER COMMON CODE
3838 023326 012737 000063 002452  1$:   MOV    #WRTGCR,CMMAND ;GCR WRITE
3839 023334 005237 002444          INC    DENFLG          ;SET GCR STAT FLAG
3840 023340 012737 000025 002602  2$:   MOV    #25,REPITN    ;25 RECORDS
3841 023346 012737 000020 002220  MOV    #20,CHAREC     ;20 BYTES/REC
3842 023354 005037 022300          CLR    F.TMP1          ;CLEAR RECORD COUNT
3843 023360 062737 000002 002616  ADD    #2,CMDSTG      ;NEXT SECTION
3844 023366                   RETURN                ;RETURN
3845
3846 023370 005237 022300          F.VC2:  INC    F.TMP1          ;INCREMENT SPACING COUNT
3847 023374 023727 022300 000025  CMP    F.TMP1,#25     ;DONE ?
3848 023402 003404                   BLE    1$             ;NOT YET
3849 023404 062737 000002 002616  ADD    #2,CMDSTG      ;YES , NEXT SECTION
3850 023412 000411                   BR     10$            ;
3851 023414 013737 022300 002602  1$:   MOV    F.TMP1,REPITN  ;LOAD SPACE COUNT
3852 023422 012737 000021 002452  MOV    #SPFWRC,CMMAND ;LOAD SPACE FORWARD
3853 023430 162737 000002 002616  SUB    #2,CMDSTG      ;ISSUE A REWIND
3854 023436                   10$:  RETURN
3855
3856 023440 162737 000001 022300  F.VC3:  SUB    #1,F.TMP1          ;DECREMENT SPACE COUNT

```

```

3857 023446 001011          BNE      1$          ;BR IF NOT DONE
3858 023450 005737 002510   TST      DENSWT     ;REALLY DONE <BOTH DENS>
3859 023454 001020          BNE      10$        ;YES
3860 023456 005237 002510   INC      DENSWT     ;NO SET GCR NOW
3861 023462 012737 003134 002616  MOV      #F.VC,CMDSTG ;AGAIN
3862 023470 000417          BR       11$
3863 023472 013737 022300 002602 1$:  MOV      F.TMP1,REPITN ;LOAD SPACE COUNT
3864 023500 012737 000021 002452  MOV      #SPFWRC,CMMAND ;LOAD COMMAND
3865 023506 162737 000002 002616  SUB      #2,CMDSTG
3866 023514 000405          BR       11$        ;RETURN
3867 023516 005037 002510 10$:  CLR      DENSWT     ;CLEAR FLAG
3868 023522 062737 000002 002616  ADD      #2,CMDSTG   ;NEXT TEST
3869 023530          11$:  RETURN
3870
3871          ;*BEGINSUBROUTINE (ERASE FUNCTIONAL TEST)
3872          ;* REWIND
3873          ;* ISSUE 200 -- 3" ERASE COMMANDS (40 FEET)
3874          ;* REWIND
3875          ;* ISSUE READ FORWARD
3876          ;* EXPECT NOT CAPABLE
3877          ;* REPEAT FOR GCR
3878          ;*ENDSUBROUTINE
3879
3880
3881 023532 012737 000020 002442  F.ER1:  MOV      #20,MESNUM ;SET MESSAGE POINTER
3882 023540 005737 002510   TST      DENSWT     ;WHICH DENSITY
3883 023544 001006          BNE      1$          ;BRANCH FOR GCR
3884 023546 005037 002444   CLR      DENFLG     ;SET PE STAT FLAG
3885 023552 012737 000035 002452  MOV      #ERGPE,CMMAND ;LOAD ERASE PE
3886 023560 000405          BR       2$
3887 023562 012737 000037 002452 1$:  MOV      #ERGGCR,CMMAND ;ERASE GCR
3888 023570 005237 002444   INC      DENFLG     ;SET GCR STAT FLAG
3889 023574 012737 000310 002602 2$:  MOV      #200.,REPITN ;200 TIMES
3890 023602 062737 000002 002616  ADD      #2,CMDSTG   ;NEXT PART
3891 023610          RETURN
3892          ; READ TAPE TO MAKE SURE IT WAS ERASED
3893
3894 023612 012737 000001 002602  F.ER2:  MOV      #1,REPITN ;ONE TIME
3895 023620 012737 000071 002452  MOV      #REDFWD,CMMAND ;LOAD READ
3896 023626 062737 000002 002616  ADD      #2,CMDSTG   ;NEXT PART
3897 023634 005037 002440   CLR      ERASED     ;CLEAR FLAG
3898 023640          RETURN
3899
3900          ; CHECK TO SEE THAT NOT CAPABLE OCCURED
3901
3902 023642 005737 002440   F.ER3:  TST      ERASED ;DID WE GET A NON CAPABLE INT CODE
3903 023646 001004          BNE      1$          ;YES , EXIT
3904 023650          ERRDF 38.,ERM037,ERRMS1 ;PRINT ERASE FAILED
3905 023660 005737 002510 1$:  TST      DENSWT     ;WHICH DENSITY
3906 023664 001006          BNE      2$          ;BRANCH IF GCR WAS DONE
3907 023666 005237 002510   INC      DENSWT     ;SET GCR
3908 023672 012737 003150 002616  MOV      #F.ER,CMDSTG ;SET START OF ERASE TEST
3909 023700 000407          BR       4$          ;EXIT
3910 023702 005037 002510 2$:  CLR      DENSWT     ;CLEAR DENSITY FLAG
3911 023706 062737 000002 002616  ADD      #2,CMDSTG   ;NEXT TEST
3912 023714 005037 002440   CLR      ERASED     ;

```

3913 023720

4\$: RETURN

3914  
3915  
3916  
3917  
3918  
3919  
3920  
3921  
3922  
3923  
3924  
3925  
3926  
3927  
3928  
3929  
3930  
3931  
3932  
3933  
3934  
3935  
3936  
3937  
3938  
3939  
3940  
3941  
3942  
3943  
3944  
3945  
3946  
3947  
3948  
3949  
3950  
3951  
3952  
3953  
3954  
3955  
3956  
3957  
3958  
3959  
3960  
3961  
3962  
3963  
3964  
3965  
3966  
3967  
3968

023722 005037 002450  
023726  
023732 012737 000001 022300  
023740 005037 002566  
023744 005037 002570  
023750  
023752 012737 000022 002442  
023760 012737 000010 002222  
023766 012737 000020 002602  
023774 013737 022300 002220  
024002 005737 002510  
024006 001006  
024010 005037 002444  
024014 012737 000061 002452  
024022 000405  
024024 005237 002444  
024030 012737 000063 002452  
024036 062737 000002 002616  
024044  
024046 012737 000020 002602  
024054 012737 000077 002452

```
;*BEGINSUBROUTINE (SHORT AND LONG RECORD TEST)
;* REWIND TAPE
;* SET BYTE COUNT TO ZERO
;* WRITE 20 RECORDS IN PE
;* READ REVERSE 20 RECORDS
;* READ FORWARD 20 RECORDS
;* INCREMENT BYTE COUNT
;* IF BYTE COUNT IS GREATER THAN 20
;* THEN
;* CONTINUE
;* ELSE
;* GO BACK AND WRITE 20 RECORDS AGAIN
;* ENDF
;* REWIND TAPE
;* SET BYTE COUNT TO MAX
;* WRITE 2 RECORDS IN PE
;* READ REVERSE 2 RECORDS
;* READ FORWARD 2 RECORDS
;* DECREMENT BYTE COUNT
;* IF BYTE COUNT IS MAX-4
;* THEN
;* CONTINUE
;* ELSE
;* WRITE 2 MORE RECORDS
;* READ REV THEN FORWARD OVER THEM
;* ENDF
;* REPEAT OPERATION FOR GCR RECORDING MODE
;*ENDSUBROUTINE
```

: SHORT RECORDS

```
F.SH1: CLR BIGREC ;CLEAR BIG RECORD FLAG
CALL F.REW ;REWIND
MOV #1,F.TMP1 ;START WITH ONE BYTE PER REC
CLR RECORD ;RECORD 0
CLR FILENM ;FILE 0
RETURN ;RETURN

F.SH2: MOV #22,MESNUM ;SET MESSAGE ADDRESS
MOV #10,PATRN ;SET DATA PATTERN
MOV #20,REPITN ;20 RECORDS
MOV F.TMP1,CHAREC ;LOAD BYTE COUNT
TST DENSWT ;PE OR GCR
BNE 1$ ;BR IF GCR
CLR DENFLG ;SET PE STATISTICS
MOV #WRTPE,CMMAND ;ISSUE A WRITE PE
BR 2$ ;
1$: INC DENFLG ;SET GCR STATISTICS
MOV #WRTGCR,CMMAND ;GCR WRITE
2$: ADD #2,CMDSTG ;NEXT PART
RETURN

F.SH3: MOV #20,REPITN ;20 RECORDS
MOV #REDREV,CMMAND ;READ REVERSE
```



GLOBAL AREAS MACY11 30(1046) 02-JUL-80 15:09 PAGE 3-49  
 ZTMJA3.P11 02-JUL-80 11:53 GLOBAL SUBROUTINES SECTION

SEQ 0072

```

3969 024062 013737 022300 002220      MOV      F.TMP1,CHAREC      ;LOAD BYTE COUNT
3970 024070 062737 000002 002616      ADD      #2,CMDSTG        ;NEXT
3971 024076                                RETURN
3972
3973 024100 012737 000020 002602  F.SH4:  MOV      #20,REPITN      ;20 RECORDS
3974 024106 012737 000071 002452      MOV      #REDFWD,CMMAND   ;READ FORWARD
3975 024114 013737 022300 002220      MOV      F.TMP1,CHAREC   ;LOAD BYTE COUNT
3976 024122 005237 022300      INC      F.TMP1          ;INCREMENT BYTE COUNT
3977 024126 022737 000021 022300      CMP      #21,F.TMP1      ;DONE ?
3978 024134 001004      BNE     1$              ;NO , BRANCH
3979 024136 062737 000002 002616      ADD      #2,CMDSTG        ;NEXT
3980 024144 000403      BR      2$              ;EXIT
3981 024146 012737 003164 002616  1$:    MOV      #F.SHRT,CMDSTG   ;LOOP
3982 024154      2$:    RETURN
3983
3984                                ;      LONG RECORD TEST
3985
3986 024156                                F.LNG1: CALL     F.REW          ;REWIND
3987 024162 005037 002566      CLR     RECORD
3988 024166 005037 002570      CLR     FILENM          ;FILE 0
3989 024172 012737 177777 022300      MOV     #-1,F.TMP1      ;SET LARGEST BYTE COUNT
3990 024200      RETURN
3991
3992 024202 012737 000002 002602  F.LNG2: MOV     #2,REPITN    ;2 RECORDS
3993 024210 013737 022300 002220      MOV     F.TMP1,CHAREC   ;LOAD BYTE COUNT
3994 024216 012737 000377 002446      MOV     #000377,LNGDAT  ;SET LONG RECORD DATA
3995 024224 012737 177777 002450      MOV     #-1,BIGREC      ;SET BIGRECORD FLAG
3996 024232 005737 002510      TST     DENSWT          ;PE OR GCR ?
3997 024236 001006      BNE     1$              ;BR IF GCR
3998 024240 005037 002444      CLR     DENFLG          ;SET PE STATISTICS
3999 024244 012737 000061 002452      MOV     #WRTPPE,CMMAND  ;WRITE PE
4000 024252 000405      BR      2$              ;
4001 024254 005237 002444      1$:    INC     DENFLG      ;SET GCR STATISTICS
4002 024260 012737 000063 002452      MOV     #WRTGCR,CMMAND  ;WRITE GCR
4003 024266 062737 000002 002616  2$:    ADD     #2,CMDSTG      ;NEXT
4004 024274      RETURN
4005
4006 024276 012737 000002 002602  F.LNG3: MOV     #2,REPITN    ;2 RECORDS
4007 024304 012737 000077 002452      MOV     #REDREV,CMMAND  ;READ REVERSE
4008 024312 013737 022300 002220      MOV     F.TMP1,CHAREC   ;LOAD BYTE COUNT
4009 024320 062737 000002 002616      ADD     #2,CMDSTG        ;NEXT
4010 024326      RETURN
4011
4012 024330 012737 000002 002602  F.LNG4: MOV     #2,REPITN    ;2 RECORDS
4013 024336 012737 000071 002452      MOV     #REDFWD,CMMAND  ;READ FORWARD
4014 024344 013737 022300 002220      MOV     F.TMP1,CHAREC   ;LOAD BYTE COUNT
4015 024352 005337 022300      DEC     F.TMP1          ;COUNT DOWN
4016 024356 022737 177773 022300      CMP     #177773,F.TMP1  ;DONE ?
4017 024364 001404      BEQ     2$              ;YES , BRANCH
4018 024366 012737 003174 002616  1$:    MOV     #F.LONG,CMDSTG  ;LOOP UNTIL DONE
4019 024374 000416      BR      3$              ;EXIT
4020 024376 005737 002510      2$:    TST     DENSWT          ;DONE
4021 024402 001006      BNE     4$              ;YES , BRANCH
4022 024404 005237 002510      INC     DENSWT          ;NO , SET GCR
4023 024410 012737 003162 002616      MOV     #F.SL,CMDSTG    ;GO BACK TO START
4024 024416 000405      BR      3$              ;

```

```

4025 024420 005037 002510      4$: CLR      DENSWT      ;RESET FLAGS
4026 024424 062737 000002 002616  ADD      #2,CMDSTG    ;NEXT
4027 024432      3$: RETURN
4028
4029
4030      ;*BEGINSUBROUTINE (CAPSTAN WORST CASE TEST)
4031      ;* REWIND TAPE
4032      ;* ERASE 400' OF TAPE SET PE (1600 ERASE 3'')
4033      ;* WAIT 10 SECONDS
4034      ;* REWIND AT HIGH SPEED
4035      ;*ENDSUBROUTINE
4036
4037 024434 005037 002450      F.CP1: CLR      BIGREC      ;CLEAR FROM LAST TEST
4038 024440 012737 000024 002442  MOV      #24,MESNUM    ;SET MESSAGE POINTER
4039 024446 012737 003100 002602  MOV      #1600.,REPITN ;400 FEET
4040 024454 012737 000037 002452  MOV      #ERGGCR,CMMAND ;ERASE COMMAND
4041 024462 062737 000002 002616  ADD      #2,CMDSTG    ;SET FOR NEXT STAGE
4042 024470      RETURN
4043
4044 024472 012701 000600      F.CP2: MOV      #600,R1    ;LOOP COUNT
4045 024476      1$: DELAY 250      ;DELAY
4046 024526      BREAK      ;KEEP ALIVE
4047 024530      DEC      R1
4048 024532 001361      BNE      1$
4049 024534 000240      PATCH: NOP
4050 024536 000240      NOP
4051 024540 000240      NOP
4052 024542      CALL     F.REW      ;TEMPORARY PATCHING LOCATIONS
4053 024546      RETURN      ;REWIND TAPE
4054      ;RETURN
4055 024550 012737 003036 002616  F.END: MOV      #FCNTBL,CMDSTG ;SET FOR NEXT PASS
4056 024556 005037 002452      CLR      CMMAND      ;LOAD END OF PASS COMMAND
4057 024562      RETURN
4058
4059
4060
4061      ;
4062      ;RETRY SUBROUTINES
4063      ;
4064
4065
4066 024564 005737 002470      RTRY01: TST     VISIT     ;SEE IF FIRST TIME THROUGH LOOP
4067 024570 001016      BNE      2$           ;BRANCH IF NOT
4068 024572 005237 002470      INC     VISIT     ;SET VISIT FLAG
4069 024576 022737 000077 002452  CMP     #REDREV,CMMAND ;SEE IF READ REVERSE COMMAND
4070 024604 001404      BEQ     1$           ;BRANCH IF READ REVERSE
4071 024606 012737 000023 002452  MOV     #SPRVRC,CMMAND ;LOAD SPACE REVERSE COMMAND,
4072 024614 000411      BR      3$           ;GET OUT
4073 024616 012737 000021 002452  1$: MOV     #SPFWRC,CMMAND ;LOAD SPACE FORWARD COMMAND
4074 024624 000405      BR      3$           ;
4075 024626 013737 002462 002452  2$: MOV     RETRCD,CMMAND ;EXECUTE RETRY COMMAND
4076 024634 005037 002470      CLR     VISIT     ;CLEAR VISIT FLAG
4077 024640 000207      3$: RTS     PC
4078
4079
4080 024642 013737 002462 002452  RTRY02: MOV     RETRCD,CMMAND ;REISSUE COMMAND

```

```

4081 024650 000207          RTS      PC
4082
4083
4084
4085 024652 022737 000077 002462 RTRY03: CMP      #REDREV,RETRCD ;SEE IF READ REVERSE COMMAND
4086 024660 001404          BEQ      1$      ;BRANCH IF READ REVERSE
4087 024662 012737 000077 002452          MOV      #REDREV,CMMAND ;LOAD READ REVERSE
4088 024670 000403          BR       2$      ;
4089 024672 012737 000071 002452 1$:      MOV      #REDFWD,CMMAND ;LOAD READ
4090 024700 000207          2$:      RTS      PC
4091
4092 024702 005737 002470          PTRY04: TST     VISIT    ;FIND WHAT LOOP
4093 024706 001006          BNE     1$      ;BRANCH IF DATA TRANS LOOP
4094 024710 005237 002470          INC     VISIT    ;SET VISIT FLAG
4095 024714 012737 000023 002452          MOV      #SPRVRC,CMMAND ;LOAD SPACE REVERSE CODE
4096 024722 000441          BR      4$      ;GET OUT
4097 024724 023737 002556 002560 1$:      CMP      WRTCNT,WRTYCT ;SEE IF ISSUE WRITE COMMAND
4098 024732 001426          BEQ     3$      ;BRANCH IF ISSUE WRITE COMMAND
4099 024734 005237 002546          INC     BADCNT   ;INCREMENT # OF BAD TRIES
4100 024740 005037 002556          CLR     WRTCNT   ;CLEAR WRITE RETRY COUNT
4101 024744 005037 002560          CLR     WRTYCT   ;CLEAR WRITE RETRY COUNT
4102 024750 022737 000063 002462          CMP      #WRTGCR,RETRCD ;SEE IF PE OR GCR
4103 024756 001006          BNE     2$      ;BRANCH IF PE
4104 024760 012737 000037 002452          MOV      #ERGGCR,CMMAND ;LOAD ERASE 3 INCHES GCR
4105 024766 005237 002514          INC     ERAFLG
4106 024772 000415          BR      4$      ;CONTINUE RETRY
4107 024774 012737 000035 002452 2$:      MOV      #ERGPE,CMMAND ;LOAD ERASE 3 INCHES PE
4108 025002 005237 002514          INC     ERAFLG
4109 025006 000407          BR      4$      ;CONTINUE RETRY
4110 025010 013737 002462 002452 3$:      MOV      RETRCD,CMMAND ;REISSUE WRITE
4111 025016 005037 002470          CLR     VISIT    ;SET UP FOR SPACE
4112 025022 005237 002556          INC     WRTCNT   ;SET UP FOR SUCCESSIVE WRITES
4113 025026 000207          4$:      RTS      PC
4114
4115
4116 025030 022737 000061 002452 RTRY05: CMP      #WRTPE,CMMAND ;SEE IF WRITE PE
4117 025036 001004          BNE     1$      ;BRANCH IF WRITE GCR
4118 025040 012737 000041 002452          MOV      #CLFLPE,CMMAND ;SET UP CLOSE PE TAPE
4119 025046 000403          BR      2$      ;
4120 025050 012737 000043 002452 1$:      MOV      #CLFGCR,CMMAND ;SET UP CLOSE GCR TAPE
4121 025056 005037 002604          2$:      CLR     RTYFLG
4122 025062 000207          RTS      PC ;RETURN
4123
4124 025064 005737 002470          RTRY06: TST     VISIT    ;SEE IF HERE BEFORE
4125 025070 001006          BNE     1$      ;BRANCH IF YES
4126 025072 005237 002470          INC     VISIT    ;SET VISIT VLAG
4127 025076 012737 000023 002452          MOV      #SPRVRC,CMMAND ;SPACE REVERSE
4128 025104 000417          BR      4$      ;BRANCH OUT
4129 025106 005037 002470          1$:      CLR     VISIT
4130 025112 022737 000063 002462          CMP      #WRTGCR,RETRCD ;SEE IF PE
4131 025120 001404          BEQ     2$      ;BRANCH IF GCR
4132 025122 012737 000041 002452          MOV      #CLFLPE,CMMAND ;CLOSE FILE PE
4133 025130 000403          BR      3$      ;
4134 025132 012737 000043 002452 2$:      MOV      #CLFGCR,CMMAND ;CLOSE FILE GCR
4135 025140 005037 002604          3$:      CLR     RTYFLG
4136 025144 000207          4$:      RTS      PC ;GO HOME

```

4137  
4138  
4139  
4140  
4141  
4142  
4143  
4144  
4145  
4146  
4147  
4148  
4149  
4150  
4151  
4152  
4153  
4154  
4155  
4156  
4157  
4158  
4159  
4160  
4161  
4162  
4163  
4164  
4165  
4166  
4167  
4168  
4169  
4170  
4171  
4172  
4173  
4174  
4175  
4176  
4177  
4178  
4179  
4180  
4181  
4182  
4183  
4184  
4185  
4186  
4187  
4188  
4189  
4190  
4191  
4192

```

: *
: * ISSUE COMMAND MODULE 3.2
: *
: * BEGINROUTINE (MOD = 3.2 ISSUE COMMAND /ISSCMD/)
: * FIND CORRECT CAS REG WITH RH ADD AND TM #
: * LOAD TM # IN CAS REGISTER
: * IF CMD IS LESS THAN 50
: * : THEN
: * :   CLR DATRAN FLAG
: * :   LOAD TM# IN CAS REG
: * :   CLEAR ATTENTION SUMMARY REGISTER
: * :   CLEAR TRE BIT IN CS1
: * :   ENABLE INTERRUPTS
: * :   LOAD CMD IN APPROPRIATE TU LOCATION IN CAS REG
: * : ELSE
: * :   SET DATRAN FLAG
: * :   SET UP RH REGISTER 14
: * :   CALL 3.2.1 (SETUP DATA BUFFER)
: * :   LOAD BYTE COUNT INTO CAS REG
: * :   LOAD TU# IN CMD ADDRESS IN CAS REG
: * :   CLEAR TRE BIT IN CS1
: * :   ENABLE INTERRUPTS
: * :   LOAD COMMAND INTO CAS REG
: * ENDIF
: * ENDROUTINE
    
```

```

ISSCMD: PUSH    <R1,R2>
4168 025146      012737 000000 002316      MOV    #RHADD,DISPLC ;MOVE INFORMATION DESIRED INTO TABLE DISPL
4169 025160      004737 036076          JSR    PC,TBLDIS    ;GET RHADD FOR SPECIFIED TABLE
4170 025164      017701 155126          MOV    @DISPLC,R1   ;TEMP STORAGE
4171 025170      012737 000004 002316      MOV    #TMNUM,DISPLC ;FIND OFFSET FOR TM NUMBER
4172 025176      004737 036076          JSR    PC,TBLDIS    ;GET ADDRESS
4173 025202      017761 155110 000010      MOV    @DISPLC,UNSLCT(R1) ;PUT UNIT NUMBER INTO UNIT SELECT
4174 025210      022737 000050 002452      CMP    #50,CMMAND   ;SEE IF MOTION OR TRANSFER COMMAND
4175 025216      100425          BMI    1$          ;BRANCH IF DATA TRANSFER COMMAND
4176 025220      005037 002516          CLR    DATRAN       ;CLEAR FLAG
4177 025224      012737 000006 002316      MOV    #TUNUM,DISPLC ;FIND OFFSET FOR TU NUMBER
4178 025232      004737 036076          JSR    PC,TBLDIS    ;GET ADDRESS
4179 025236      017702 155054          MOV    @DISPLC,R2   ;R2 CONTAINS TU NUMBER
4180 025242      006302          ASL    R2           ;SHIFT FOR WORDS
4181 025244      062702 000040          ADD    #NDTFCO,R2   ;GETTING LOCATION FOR MOTION COMMANDS
4182 025250      060102          ADD    R1,R2        ;LOCATION FOR COMMAND
4183 025252      012711 040000          MOV    #40000,(R1)  ;CLEAR TRE BIT
4184 025256      005011          CLR    (R1)         ;CLEAR TRE BIT IN TM78 MEMORY
4185 025260      012711 000100          MOV    #100,(R1)   ;SET INTERRUPT ENABLE BIT
4186 025264      013712 002452          MOV    CMMAND,(R2) ;PUT COMMAND INTO CAS
4187 025270      000444          BR     2$          ;
4188
4189 025272      012737 000001 002516 1$: MOV    #1,DATRAN    ;SET DATA TRANSFER FLAG
4190 025300      042761 077777 000014      BIC    #77777,CMDADR(R1) ;SET FORMAT , SKIP COUNT,AND RECORD CNT TO 0
4191 025306      004737 025410          JSR    PC,SETDBF    ;SETUP DATA BUFFER
4192 025312      013761 002456 000006      MOV    BUFFER,BYTCNT(R1) ;SETUP BYTE COUNT
    
```

```

4193 025320 012737 000006 002316      MOV      #TUNUM,DISPLC      ;FIND OFFSET FOR TU NUMBER
4194 025326 004737 036076              JSR      PC,TBLDIS         ;GET ADDRESS
4195 025332 057761 154760 000014      BIS      @DISPLC,CMDADR(R1) ;PUT TU INTO COMMAND ADR
4196 025340 013737 002452 002454      MOV      CMMAND,EICMMD
4197 025346 062737 000100 002454      ADD      #100,EICMMD      ;SET INTERRUPT ENABLE BIT
4198 025354 012711 040000              MOV      #40000,(R1)      ;CLEAR TRE BIT IN CS1
4199 025360 005011              CLR      (R1)             ;CLEAR TRE BIT IN TM78 MEMORY
4200 025362 005737 002450              TST      BIGREC          ;LARGE RECORD FLAG SET
4201 025366 001403              BEQ      20$             ;NO , PROCEED AS USUAL
4202 025370 052761 000010 000010      BIS      #10,UNSLCT(R1)  ;SET BUS ADDRESS INCREMENT INHIBIT
4203 025376 013711 002454      20$:    MOV      EICMMD,(R1)  ;LOAD COMMAND
4204 025402              2$:    POP      <R2,R1>     ;RESTORE REGISTERS
4205 025406 000207      RTS      PC

```

```

4206
4207
4208
4209
4210
4211
4212      ;*SET UP DATA BUFFER MOD = 3.2.1
4213      ;*
4214      ;*BEGINROUTINE (MOD = 3.2.1 SET UP DATA BUFFER /SETDBF/)
4215      ;* IF RANDOM RECORDS
4216      ;* : THEN
4217      ;* : DO UNTIL CORRECT SIZE
4218      ;* : : CALL RANDOM GEN
4219      ;* : ENDDO
4220      ;* : STORE BYTES IN RECORD
4221      ;* : IF BUFFER NOT=CHARACTERS PER RECORD
4222      ;* : : THEN
4223      ;* : : RESET BUFFER GENERATION FLAG
4224      ;* : ENDF
4225      ;* ENDF
4226      ;* IF # OF BYTES IS GREATER THAN BUFFSIZE (BYTE) (RECORDS)
4227      ;* : THEN
4228      ;* : PRINT BUFFER REQUESTED TOO LARGE, USING LARGEST AVAILABLE (X)
4229      ;* : MOVE BUFFER TO TEMPORARY LOCATION
4230      ;* : ELSE
4231      ;* : MOVE #_OF_BYTES TO TEMPORARY LOCATION
4232      ;* ENDF
4233      ;* IF ODD # OF BYTES IN BUFFER
4234      ;* : THEN
4235      ;* : INCREMENT BUFFER
4236      ;* ENDF
4237      ;* DIVIDE BYTES BY TWO
4238      ;* TWO'S COMPLEMENT BYTES
4239      ;* MOVE BYTES TO MTWC, (MASS BUS WORD COUNT)
4240      ;* TWO'S COMPLEMENT BYTES
4241      ;* CALCULATE END OF BUFFER
4242      ;* IF READ COMMAND
4243      ;* : THEN
4244      ;* : ZERO READ BUFFER
4245      ;* : IF READ REVERSE
4246      ;* : : THEN
4247      ;* : : IF ODD BUFFER
4248      ;* : : : THEN

```

```

4249      : * : : : ADJUST SKIP COUNT
4250      : * : : : PUT END OF BUFFER INTO MASS BUS ADDRESS
4251      : * : : : ENDF
4252      : * : : : ELSE
4253      : * : : : PUT END OF READ BUFFER INTO MASS BUS ADDRESS
4254      : * : : : ENDF
4255      : * : : : ELSE
4256      : * : : : LOAD BEGINNING OF READ BUFFER INTO MASSBUS ADDRESS
4257      : * : : : ENDF
4258      : * : : : IF WRITE COMMAND
4259      : * : : : THEN
4260      : * : : : IF NOT SAME PATTERN
4261      : * : : : THEN
4262      : * : : : CALL GENBUF (3.2.1.1)
4263      : * : : : SET PATTERN # GENERATION FLAG
4264      : * : : : ENDF
4265      : * : : : ENDF
4266      : * : : : ENDRoutine
4267
4268
4269

```

```

4270      025410      SETDBF: PUSH      <R1,R2,R3,R4,R5>
4271      025422      023727 002452 000067      CMP      CMMAND,#67      :SEE IF READ COMMAND
4272      025430      101025      BHI      10$      :BRANCH IF YES
4273      025432      005737 002224      TST      RALNTH      :SEE IF RANDOM RECORD LENGTH
4274      025436      001422      BEQ      10$      :BRANCH IF NOT
4275      025440      023737 002564 002352      CMP      RECTBL,TBLNUM :SEE IF SAME COMMAND DIFFERENT TU
4276      025446      001416      BEQ      10$      :BRANCH IF YES
4277      025450      005737 002604      TST      RTYFLG      :RETRY IN PROGRESS?
4278      025454      001022      BNE      9$      :BRANCH AROUND BUF REGEN IF SO
4279      025456      004737 026120      JSR      PC,RANGEN    :GET RANDOM RECORD LENGTH
4280      025462      042737 170000 002640      BIC      #170000,RANS2 :KEEP RECOR LENGTH LOWER THAN 4K
4281      025470      005737 002640      TST      RANS2      :SEE IF ZERO
4282      025474      001770      BEQ      11$      :BRANCH IF YES
4283      025476      013737 002640 002220      MOV      RANS2,CHAREC :STORE NEW BYTES PER RECORD
4284      025504      023737 002456 002220      10$: CMP      BUFFER,CHAREC :SEE IF SAME BUFFER SIZE
4285      025512      001403      BEQ      9$      :BRANCH IF NOT
4286      025514      012737 177777 002426      MOV      #-1,GENFLG   :NEW BUFFER GENERATION FLAG
4287      025522      013705 002220      9$: MOV      CHAREC,R5     :R5 CONTAINS BYTE COUNT
4288      025526      005737 002450      TST      BIGREC      :BIG RECORD FLAG SET
4289      025532      001020      BNE      1$      :YES SKIP THIS SECTION
4290      025534      023737 002220 002312      CMP      CHAREC,BUFSIZ :SEE IF BYTE COUNT TOO LARGE
4291      025542      101414      BLOS     1$      :BRANCH IF ACCEPTABLE BYTE COUNT
4292      025544      013705 002312      MOV      BUFSIZ,R5    :R5 HAS LARGEST BYTE COUNT ALLOWED
4293      025550      PRINTF #ERRBUF,BUFSIZ :PRINT ERROR
4294      025574      012737 000000 002316      1$: MOV      #RHADD,DISPLC :GET RH ADDRESS
4295      025602      004737 036076      JSR      PC,TBLDIS    :R1 CONTAINS RH ADDRESS
4296      025606      017701 154504      MOV      @DISPLC,R1   :BUFFER CONTAINS # OF BYTES TO TRANSFER
4297      025612      010537 002456      MOV      R5,BUFFER    :RESTORE BYTE COUNT
4298      025616      010537 002220      MOV      R5,CHAREC    :R4 CONTAINS #OF BYTES
4299      025622      010504      MOV      R5,R4        :SEE IF ODD BYTE
4300      025624      032705 000001      BIT      #1,R5        :BRANCH IF BYTE COUNT EVEN
4301      025630      001401      BEQ      2$      :MAKE EVEN
4302      025632      005205      INC      R5
4303      025634      000241      2$: CLC
4304      025636      006005      ROR      R5          :MAKE WORD

```

```

4305 025640 005405          NEG      R5          ;COMPLIMENT WORD COUNT
4306 025642 010561 000002  MOV      R5,WRDCNT(R1) ;LOAD WORD COUNT
4307 025646 005737 002450  TST      BIGREC      ;BIG RECORD FLAG SET
4308 025652 001404          BEQ      20$         ;NO , CARRY ON
4309 025654 012761 002446 000004  MOV      #LNGDAT,BSADDR(R1) ;SET ADDRESS TO BIG RECORD DATA
4310 025662 000510          BR       7$         ;EXIT THIS SUBROUTINE
4311 025664 063704 002354      20$:  ADD      VRMADD,R4     ;R4 CONTAINS LAST MEMORY ADDRESS
4312 025670 010437 002460      MOV      R4,BUFLST   ;STORE LAST ADDRESS OF BUFFER
4313 025674 022737 000051 002452  CMP      #WCKFWD,CMMAND ;WRITE CHECK FORWARD
4314 025702 001464          BEQ      6$         ;USE THE LAST WRITE BUFFER FOR COMPARE
4315 025704 022737 000061 002452  CMP      #61,CMMAND   ;WRITE COMMAND PE ?
4316 025712 001460          BEQ      6$         ;BR IF YES
4317 025714 022737 000063 002452  CMP      #63,CMMAND   ;WRITE GCR ?
4318 025722 001454          BEQ      6$         ;BR IF WRITE COMMAND
4319 025724 022737 000057 002452  CMP      #WCKREV,CMMAND ;SEE IF WRITE CHECK REVERSE
4320 025732 001003          BNE     14$         ;NO
4321 025734 013704 002354      MOV      VRMADD,R4     ;LOAD LAST WRITE BUFFER ADDRESS
4322 025740 000412          BR       15$         ;USE READ REVERSE COMMON CODE
4323 025742 022737 000077 002452  14$:  CMP      #REDREV,CMMAND ;SEE IF READ REVERSE COMMAND
4324 025750 001404          BEQ      3$         ;BRANCH IF YES
4325 025752 013761 002332 000004  MOV      REDBUF,BSADDR(R1) ;SET UP STARTING READ BUFFER ADDRESS
4326 025760 000423          BR       4$         ;
4327 025762 013704 002332      3$:  MOV      REDBUF,R4     ;R4 CONTAINS START OF READ BUFFER
4328 025766 063704 002456      15$:  ADD      BUFFER,R4     ;R4 CONTAINS LAST ADDRESS OF BUFFER
4329 025772 032737 000001 002456  BIT      #1,BUFFER    ;SEE IF ODD BYTE
4330 026000 001407          BEQ      8$         ;BRANCH IF NOT
4331 026002 052761 000400 000014  BIS      #400,CMDADR(R1) ;ADJUST SKIP COUNT FOR ODD BYTE
4332 026010 005304          DEC      R4          ;ADJUST ADDRESS FOR ODD BYTE
4333 026012 010461 000004      MOV      R4,BSADDR(R1) ;STORE READ REVERSE ADDRESS
4334 026016 000404          BR       4$         ;
4335 026020 005304      8$:  DEC      R4          ;MAKE CORRECT LAST ADDRESS
4336 026022 005304          DEC      R4          ;
4337 026024 010461 000004      MOV      R4,BSADDR(R1) ;LOAD LAST ADDRESS OF BUFFER
4338 026030 013703 002332      4$:  MOV      REDBUF,R3     ;R3 CONTAINS START OF READ BUS ADDRESS
4339 026034 063703 002456      ADD      BUFFER,R3     ;R3 CONTAINS LAST ADDRESS
4340 026040 013702 002332      MOV      REDBUF,R2     ;R2 CONTAINS READ BUFFER ADDRESS
4341 026044 005022      5$:  CLR      (R2)+         ;CLEAR READ BUFFER
4342 026046 020302          CMP      R3,R2        ;SEE IF DONE
4343 026050 101375          BHI     5$          ;BRANCH IF NOT
4344 026052 000403          BR       12$         ;BRANCH IF YES
4345 026054 013761 002354 000004  6$:  MOV      VRMADD,BSADDR(R1) ;LOAD START OF WRITE ADDRESS
4346 026062 023737 002222 002426  12$:  CMP      PATRN,GENFLG  ;SEE IF SAME PATTERN
4347 026070 001405          BEQ      7$         ;BRANCH TO END
4348 026072 004737 026160          JSR      PC,GENBUF    ;GENERATE BUFFER
4349 026076 013737 002222 002426      MOV      PATRN,GENFLG  ;SET PATTERN GENERATION FLAG
4350 026104          7$:  POP      <R5,R4,R3,R2,R1>
4351 026116 000207          RTS      PC

```

;\*BEGINROUTINE (MODULE 3.2.1.1 = RANDOM DATA)

```

4352
4353
4354
4355
4356
4357 026120          RANGEN: PUSH      <R5>
4358 026122 012705 003466      MOV      #PATEGT,R5   ;GET PATTERN LOCATION
4359 026126 063737 002640 002636  1$:  ADD      RANS2,RANB2  ;GENERATE RANDOM NUMBER
4360 026134 063737 002636 002640      ADD      RANB2,RANS2  ;

```

```

4361 026142 013725 002640      MOV      RANS2,(R5)+      ;STORE #
4362 026146 022705 003476      CMP      #PATEGT+10,R5   ;SEE IF DONE
4363 026152 001365                BNE      1$              ;BRANCH IF NOT
4364 026154                POP      <R5>
4365 026156 000207      RTS      PC

```

```

4366
4367
4368      ;*BEGINROUTINE (MODULE 3.2.1.2 = GCR PATTERN GENERATION)
4369      ;* STORE START OF WRITE BUFFER
4370      ;* STORE PATTERN #
4371      ;* MAKE PATTERN # WORD OFFSET
4372      ;* DO UNTIL BUFFER FULL
4373      ;* : IF RANDOM PATTERN
4374      ;* : : THEN
4375      ;* : : CALL RANDOM GENERATOR
4376      ;* : : ENDF
4377      ;* : IF GCR PATTERN
4378      ;* : : THEN
4379      ;* : : CALL GCR PATTERN GENERATOR
4380      ;* : : SET GCR FLAG
4381      ;* : : ENDF
4382      ;* : CLEAR PATTERN COUNTER
4383      ;* : GET PATTERN ADDRESS
4384      ;* : DO UNTIL PATTERN BUFFER FULL
4385      ;* : : INCREMENT PATTERN COUNT
4386      ;* : : IF INCREMENT PATTERN
4387      ;* : : : THEN
4388      ;* : : : GENERATE NEXT WORD
4389      ;* : : : ENDF
4390      ;* : ENDDO
4391      ;* ENDDO
4392      ;* CLEAR GCR PATTERN
4393      ;* INITIALIZE INCREMENT PATTERN
4394      ;*ENDROUTINE
4395

```

```

4396 026160      GENBUF: PUSH      <R3,R4,R5>
4397 026166 013704 002354      MOV      VRMADD,R4      ;R4 CONTAINS START OF BUFFER
4398 026172 013703 002222      MOV      PATRN,R3      ;GET PATTERN #
4399 026176 006303                ASL      R3              ;MAKE INTO WORD OFFSET
4400 026200 022737 000010 002222 1$:  CMP      #10,PATRN     ;SEE IF RANDOM PATTERN
4401 026206 001002                BNE      6$              ;BRANCH IF NOT
4402 026210 004737 026120      JSR      PC,RANGEN     ;CALL RANDOM GENERATOR
4403 026214 022737 000011 002222 6$:  CMP      #11,PATRN     ;SEE IF GCR PATTERN
4404 026222 101005                BHI      2$              ;BRANCH IF NOT GCR
4405 026224 004737 026342      JSR      PC,GCRGEN     ;CALL GCR GENERATOR
4406 026230 012737 000001 002550      MOV      #1,GCRFLG     ;
4407 026236 005037 002536 2$:    CLR      PATCNT        ;PATTERN OFFSET
4408 026242 016305 003324      MOV      PATGEN(R3),R5 ;R5 CONTAINS PATTERN ADDRESS
4409 026246 012524 3$:    MOV      (R5)+,(R4)+   ;GENERATE BYTE OF BUFFER
4410 026250 023704 002460      CMP      BUFLST,R4     ;SEE IF LAST BYTE
4411 026254 101421                BLOS     5$              ;BRANCH IF DONE
4412 026256 005237 002536      INC      PATCNT        ;GET NEXT BYTE
4413 026262 022737 000007 002222      CMP      #7,PATRN     ;SEE IF 256 PATTERN
4414 026270 001006                BNE      4$              ;BRANCH IF NOT
4415 026272 005305                DEC      R5              ;STAY IN SAME BYTE
4416 026274 105215                INCB     (R5)

```



```

4417 026276 105215          INCB   (R5)
4418 026300 005305          DEC    R5
4419 026302 105215          INCB   (R5)
4420 026304 105215          INCB   (R5)
4421 026306 023737 002350 002536 4$:  CMP    PATLOP,PATCNT ;SEE IF DONE
4422 026314 001354          BNE    3$ ;BRANCH IF NOT DONE
4423 026316 000730          BR     1$ ;BRANCH TO GET MORE PATTERN
4424 026320 005037 002550          CLR    GCRFLG
4425 026324 012737 000400 003464 5$:  MOV    #400,PATSVN
4426 026332          POP    <R5,R4,R3>
4427 026340 000207          RTS    PC
4428
4429          ;*BEGINROUTINE (MODULE 3.2.1.2.1 = GCR BUFFER)
4430          ;* GET PATTERN STORAGE ADDRESS
4431          ;* IF BEGINNING OF PATTERN
4432          ;* : THEN
4433          ;* : CLEAR BINARY COUNTERS
4434          ;* ENDIF
4435          ;* DO UNTIL GCR BUFFER FILLED
4436          ;* : DO UNTIL MAXIMUM GCR LOOP REACHED
4437          ;* : : CHANGE BIT PATTERN
4438          ;* : : STORE BINARY COUNTER
4439          ;* : ENDDO
4440          ;* : IF LSB NOT SET
4441          ;* : : THEN
4442          ;* : : STORE BYTE A
4443          ;* : : ELSE
4444          ;* : : STORE BYTE B
4445          ;* : ENDIF
4446          ;* : INCREMENT LOOP COUNTER
4447          ;* : SHIFT RIGHT FOR NEXT BYTE
4448          ;* : IF BUFFER FILLED
4449          ;* : : THEN
4450          ;* : : INCREMENT GCR BUFFER
4451          ;* : ENDIF
4452          ;* ENDDO
4453          ;*ENDROUTINE
4454
4455          GCRGEN: PUSH    <R5>
4456 026342          MOV    #PATNIN,R5 ;R5 CONTAINS ADDRESS OF PATTERN STORAGE
4457 026344 012705 003476          TST    GCRFLG ;SEE IF BEGINNING OF PATTERN
4458 026350 005737 002550          BNE    1$ ;BRANCH IF NOT
4459 026354 001006          CLR    GCRBIT
4460 026356 005037 002552          CLR    GCRCNT ;SET UP BINARY COUNTER
4461 026362 005037 002576          CLR    GCRRPT
4462 026366 005037 002574          CLR    GCRRPT,GCRLOP ;SEE IF MAXIMUM LOOP COUNTER
4463 026372 023737 002574 002630 1$:  BNE    2$ ;BRANCH IF NOT REACHED
4464 026400 001010          CLR    GCRRPT ;START PATTERN LOOP OVER
4465 026402 005037 002574          INC    GCRCNT ;CHANGE BIT PATTERN
4466 026406 005237 002576          MOV    GCRCNT,GCRBIT ;SET GCRBIT WITH BINARY COUNTER
4467 026412 013737 002576 002552          BR     1$
4468 026420 000764          BR
4469 026422 032737 000001 002552 2$:  BIT    #1,GCRBIT ;SEE IF LSB SET
4470 026430 001003          BNE    3$ ;BRANCH IF SET
4471 026432 113725 002624          MOVB  BYTEA,(R5)+ ;MOVE BYTE A INTO PATTERN LOCATION
4472 026436 000402          BR     4$
4473 026440 113725 002626          MOVB  BYTEB,(R5)+ ;MOVE BYTE B INTO PATTERN LOCATION

```

```

4473 026444 005237 002574      4$:  INC      GRRPT      ;INCREMENT LOOP COUNTER
4474 026450 006037 002552      ROR      GCRBIT      ;ROTATE TO SEE IF LSB SET
4475 026454 022705 003506      CMP      #PATNIN+10,R5 ;SEE IF BUFFER FILLED
4476 026460 001344      BNE      1$          ;BRANCH IF NOT
4477 026462      POP      <R5>      ;
4478 026464 000207      RTS      PC          ;GO HOME
4479

```

```

4480      ;*
4481      ;*MODULE 3.3 COMMAND TERMINATION
4482      ;*
4483
4484
4485

```

```

4486      ;*BEGINROUTINE (MOD = 3.3 COMMAND TERMINATION /CMDTRM/)
4487      ;* CLR REGISTERS
4488      ;* IF COMMAND WAS A REWIND
4489      ;* : THEN
4490      ;* : : DO UNTIL ALL UNITS INTERRUPTED OR TIMED_OUT
4491      ;* : : : CLEAR INTERRUPT FLAG
4492      ;* : : : ENABLE INTERRUPTS
4493      ;* : : : DO UNTIL INTERRUPT FLAG SET OR TIME_OUT REACHED
4494      ;* : : : INCREMENT TIME_OUT COUNTER
4495      ;* : : : ENDDO
4496      ;* : : : IF TIME_OUT HAS OCCURRED
4497      ;* : : : : THEN
4498      ;* : : : : : IF BRUTUS W/ERRORS OR CANNED SEQUENCE
4499      ;* : : : : : : THEN
4500      ;* : : : : : : : PRINT UNIT(S) THAT TIMED OUT
4501      ;* : : : : : : : IF CANNED SEQUENCE OPTION
4502      ;* : : : : : : : : THEN
4503      ;* : : : : : : : : : SET EOT FLAG FOR UNIT(S) TIMED OUT
4504      ;* : : : : : : : : : ELSE
4505      ;* : : : : : : : : : INCREMENT ALL EOT FLAG
4506      ;* : : : : : : : : : ENDDIF
4507      ;* : : : : : : : ELSE
4508      ;* : : : : : : : : INCREMENT ALL EOT FLAG
4509      ;* : : : : : : : ENDDIF
4510      ;* : : : : ELSE
4511      ;* : : : : IF ILLEGAL INTERRUPT
4512      ;* : : : : : THEN
4513      ;* : : : : : : IF BRUTUS W/ERRORS OR CANNED SEQUENCE
4514      ;* : : : : : : : THEN
4515      ;* : : : : : : : : PRINT INTERRUPT THAT WAS UNEXPECTED
4516      ;* : : : : : : : ENDDIF
4517      ;* : : : : : : : : CLEAR INTERRUPT FLAG
4518      ;* : : : : : : : ELSE
4519      ;* : : : : : : : : IF BRUTUS W/ERRORS OR CANNED SEQUENCE
4520      ;* : : : : : : : : : THEN
4521      ;* : : : : : : : : : : CALL RESPECTIVE INTERRUPT SUBROUTINE
4522      ;* : : : : : : : : : : CLEAR INTERRUPT IN CAS SUMMARY REG
4523      ;* : : : : : : : : : ENDDIF
4524      ;* : : : : : : : : IF INPUT TU ON LINE INTERRUPT
4525      ;* : : : : : : : : : THEN
4526      ;* : : : : : : : : : : INCREMENT ALL UNITS INTERRUPTED
4527      ;* : : : : : : : : : ENDDIF
4528      ;* : : : : : : : ENDDIF

```



```

4585      : * : : : : : : : : : THEN
4586      : * : : : : : : : : : CALL MODULE 3.3.2
4587      : * : : : : : : : : : ENDIF
4588      : * : : : : : : : : : ELSE
4589      : * : : : : : : : : : PRINT ERROR
4590      : * : : : : : : : : : ENDIF
4591      : * : : : : : : : : : ELSE
4592      : * : : : : : : : : : IF DATA COMPARE SET
4593      : * : : : : : : : : : THEN
4594      : * : : : : : : : : : CALL MODULE 3.3.2
4595      : * : : : : : : : : : ENDIF
4596      : * : : : : : : : : : ENDIF
4597      : * : : : : : : : : : ENDIF
4598      : * : : : : : : : : : ENDIF
4599      : * : : : : : : : : : ENDIF
4600      : * : : : : : : : : : ENDROUTINE
4601
4602      CMDTRM: PUSH      <R1,R2,R3,R4,R5>
4603      026466          CLR      R10
4604      026500 005037 002424      CLR      R1
4605      026504 005001          CLR      R2
4606      026506 005002          CLR      R3
4607      026510 005003          CLR      R4
4608      026512 005004          CLR      R5
4609      026514 005005          CLR      R5
4610      026516 022737 000007 002452      CMP      #REWIND,CMMAND ;SEE IF REWIND WAS EXECUTED
4611      026524 001402          BEQ      1$ ;BRANCH IF A REWIND COMMAND
4612      026526 000137 027530          JMP      15$
4613      026532 006303          1$: ASL      R3 ;SHIFT FOR THE NUMBER OF UNITS
4614      026534 052703 000001          BIS      #BIT00,R3 ;SET A BIT FOR EACH UNIT
4615      026540 005337 002612          DEC      RWDFLG ;GET NEXT UNIT
4616      026544 001372          BNE      1$ ;BRANCH IF AL BITS FOR ALL UNITS NOT SET
4617      026546 005037 002374          2$: CLR      INTRPT ;CLEAR INTERRUPT FLAG SET IN SERVICE ROUTINE
4618      026552          SETPRI #PRI00 ;ENABLE INTERRUPTS
4619      026560 005737 002374          3$: TST      INTRPT ;SEE IF INTERRUPT RECEIVED
4620      026564 001121          BNE      7$ ;BRANCH IF INTERRUPT
4621      026566 005237 002424          INC      R10 ;INCREMENT TIME OUT COUNTER
4622      026572          BREAK
4623      026574          DELAY 22
4624      026624 022737 177777 002424      CMP      #-1,R10 ;SEE IF TIME OUT REACHED
4625      026632 001352          BNE      3$ ;IF NO TIME OUT LIMIT REACHED THEN BRANCH
4626      026634 005005          CLR      R5
4627      026636 022737 000001 002314      CMP      #1,BRTEST ;CHECKING FOR BRUTUS OPTION
4628      026644 001465          BEQ      6$ ;BRANCH IF BRUTUS CHOSEN
4629      026646 010304          MOV      R3,R4 ;R4 CONTAINS A BIT SET FOR EACH UNIT REWINDING
4630      026650 043704 002562          BIC      ALLINT,R4 ;R4 HAS UNITS THAT TIMED OUT
4631      026654 005737 002210          TST      BRUERR ;SEE IF BRUTUS W/ERRORS
4632      026660 001006          BNE      4$ ;BRANCH IF NOT CANNED SEQUENCE
4633      026662 050437 002540          BIS      R4,EOTFLG ;SET EOT FLAG FOR EACH UNIT THAT TIMED OUT
4634      026666 005237 002600          INC      ALLEOT
4635      026672 050437 002534          BIS      R4,DRPFLG ;SET DROP FLAG FOR SCHEDULER
4636      026676 050437 002562          4$: BIS      R4,ALLINT ;SET INTERRUPT FLAG FOR EACH UNIT THAT TIMED OUT
4637      026702 005205          5$: INC      R5 ;SET UP UNIT COUNTER
4638      026704 005704          TST      R4 ;SEE IF YOU GOT ALL UNIT ERRORS
4639      026706 001002          BNE      50$
4640      026710 000137 027470          JMP      14$ ;BRANCH IF DONE PRINTING ERRORS
4641      026714 006204          50$: ASR      R4 ;SEE IF ERROR SHOULD BE PRINTED FOR NEXT UNIT

```

```

4641 026716 103371          BCC      5$          ;BRANCH IF NO ERROR
4642 026720 010537 002352  MOV      R5,TBLNUM   ;TBLNUM CONTAINS UNIT FOR DISPLACEMENT SUB
4643 026724 012737 000000 002316  MOV      #RHADD,DISPLC ;LOOKING FOR RH ADDRESS
4644 026732 004737 036076          JSR      PC,TBLDIS
4645 026736 017701 153354          MOV      @DISPLC,R1   ;R1CONTAINS RH ADDRESS
4646 026742 013702 002316          MOV      DISPLC,R2   ;R2 CONTAINS P-TABLE ADDRESS
4647 026746 017737 153344 002356  MOV      @DISPLC,RHADDR ;STORAGE FOR RH ADDRESS
4648 026754 016237 000004 002370  MOV      TMNUM(R2),TM78N ;STORAGE FOR TM78 NUMBER
4649 026762 016237 000006 002372  MOV      TUNUM(R2),TU78N ;STORAGE FOR TU78 NUMBER
4650 026770 005237 002542          INC      TIMEOUT
4651 026774 016237 000010 002074  MOV      SPUNIT(R2),L$LUN
4652 027002          ERRDF  1.,ERM001,ERRMS1
4653 027012 005037 002542          CLR      TIMEOUT
4654 027016 000731          BR      5$          ;GO LOOK FOR ANOTHER UNIT TIMED OUT
4655 027020 005237 002562 6$:      INC      ALLINT  ;FAKE AN INTERRUPT FOR BRUTUS OPTION
4656 027024 000137 027470          JMP      14$
4657
4658          ;CHECKING TO SEE IF ILLEGAL INTERRUPT
4659
4660 027030 005037 002374 7$:      CLR      INTRPT
4661 027034          SETPRI  PRIORITY
4662 027042 013701 002532          MOV      MASBUS,R1   ;R1 CONTAINS RH ADDRESS OF INTERRUPT GENERATED
4663 027046 013737 002532 002356  MOV      MASBUS,RHADDR
4664 027054 012711 000100          MOV      #100,(R1)   ;RE ENABLE INTERRUPT BIT
4665 027060 005761 000016          TST      ATTBIT(R1) ;IS THERE AN ATTENTION SET
4666 027064 001011          BNE      8$          ;YES, GO HANDLE INTERUPT
4667 027066 016137 000010 002430  MOV      UNSLCT(R1),BNRYNB ;SET UNIT THAT CAUSED MASS BUS ERROR
4668 027074 042737 177770 002430  BIC      #177770,BNRYNB ;MASK DOWN TO UNIT NO. ONLY
4669 027102 005037 002506          CLR      TAPMON     ;NO ATTENTION BIT
4670 027106 000410          BR      13$         ;PROCEED AS USUAL
4671 027110 016137 000016 002506 8$:      MOV      ATTBIT(R1),TAPMON ;TAPMON CONTAINS TM ATTENTION BIT
4672 027116 013737 002506 002430  MOV      TAPMON,BNRYNB ;SET UP FOR SHIFT SUBROUTINE
4673 027124 004737 036360          JSR      PC,SHIFTR   ;CONVERT BIT POSITION TO BINARY NUMBER
4674 027130 013737 002430 002370 13$:     MOV      BNRYNB,TM78N ;TM78N CONTAINS BINARY EQUIVALENT TO BIT POSITION
4675 027136 013761 002370 000010  MOV      TM78N,UNSLCT(R1) ;SET UNIT SELECT WITH TM# THAT INTERRUPTED
4676 027144 016137 000036 002372  MOV      NDTICD(R1),TU78N ;TU78N CONTAINS ATTENTION ADDRESS
4677 027152 000337 002372          SWAB     TU78N
4678 027156 116137 000036 002526  MOVB     NDTICD(R1),ICCODE ;ICCODE CONTAINS INTERRUPT
4679 027164 113737 002372 002522  MOVB     TU78N,EXTERR
4680 027172 042737 177403 002522  BIC      #177403,EXTERR ;EXTERR CONTAINS ERROR
4681 027200 006237 002522          ASR      EXTERR     ; INFORMATION!
4682 027204 006237 002522          ASR      EXTERR
4683 027210 042737 177774 002372  BIC      #177774,TU78N ;TU78N NOW CONTAINS TU78 NUMBER
4684 027216 013737 002344 002420  MOV      TBLCNT,R8   ;R8 CONTAINS NUMBER OF UNITS
4685 027224 013702 002330          MOV      PTBLAD,R2  ;R2 CONTAINS ADDRESS OF TABLE
4686 027230 012737 000001 002352  MOV      #1,TBLNUM
4687 027236 021237 002532 9$:      CMP      (R2),MASBUS ;LOOKING FOR MASSBUS THAT CAUSE INTERRUPT
4688 027242 001435          BEQ      12$         ;BRANCH IF FOUND UNIT WITH CORRECT RH
4689 027244 063702 002346 10$:     ADD      TBLNTH,R2   ;GET NEXT TABLE, R5 CONTAINS ADDRESS OF NEXT TABLE
4690 027250 063702 002346          ADD      TBLNTH,R2  ;NEXT TABLE IS AFTER GCR UNIT TABLE
4691 027254 005237 002352          INC      TBLNUM
4692 027260 005337 002420          DEC      R8         ;LOOP UNIT COUNTER
4693 027264 001364          BNE      9$         ;TRY COMPARE AGAIN
4694
4695          ;MUST BE AN ILLEGAL INTERRUPT
4696

```

```

4697 027266 022737 000001 002314      CMP      #1,BRTEST      ;SEE IF BRUTUS OPTION
4698 027274 001413                    BEQ      11$           ;BRANCH IF YES
4699 027276 005237 002542                    INC      TIMEOUT
4700 027302 012737 000143 002074      MOV      #143,L$LUN
4701 027310                    ERRDF   2.,ERM002,ERRMS2
4702 027320 005037 002542                    CLR      TIMEOUT
4703 027324 013761 002506 000016 11$:   MOV      TAPMON,ATBIT(R1) ;CLEARS ATTENTION SUMMARY INTERRUPT BIT
4704 027332 000137 026546                    JMP      2$           ;GO WAIT FOR ANOTHER INTERRUPT
4705 027336 026237 000004 002370 12$:   CMP      TMNUM(R2),TM78N ;SEE IF THIS TM GENERATED INTERRUPT
4706 027344 001337                    BNE     10$           ;BRANCH IF WRONG TM NUMBER
4707 027346 026237 000006 002372      CMP      TUNUM(R2),TU78N ;SEE IF THIS TU GENERATED INTERRUPT
4708 027354 001333                    BNE     10$           ;BRANCH IF WRONG TU NUMBER
4709
4710      ;MUST BE LEGAL INTERRUPT
4711
4712 027356 016105 000036      MOV      NDTICD(R1),R5 ;R5 CONTAINS INTERRUPT RECEIVED
4713 027362 042705 177400      BIC      #177400,R5
4714 027366 022705 000035      CMP      #35,R5      ;SEE IF UNDEFINED INTERRUPT
4715 027372 101002                    BHI     47$           ;BRANCH IF NO
4716 027374 012705 000016      MOV      #16,R5      ;SET UP FOR UNDEFINED INTERRUPT
4717 027400 006305                    ASL     R5            ;MAKE WORD ADDRESS OFFSET
4718 027402 016237 000010 002074 47$:   MOV      SPUNIT(R2),L$LUN
4719 027410 004775 002662      JSR      PC,@INTTBL(R5) ;CALL RESPECTIVE INTERRUPT SUBROUTINE
4720 027414 013761 002506 000016      MOV      TAPMON,ATBIT(R1) ;CLEAR ATTENTION SUMMARY INTERRUPT BIT
4721 027422 022705 000016      CMP      #16,R5      ;SEE IF REWINDING INTERRUPT
4722 027426 001420                    BEQ     14$           ;BRANCH IF REWINDING
4723 027430 022705 000036      CMP      #36,R5      ;SEE IF ONLINE INTERUPT
4724 027434 001415                    BEQ     14$           ;YES , DON'T COUNT THIS AS DONE
4725 027436 163737 002344 002420      SUB     TBLCNT,R8    ;GET CORRECT UNIT THAT INTERRUPTED
4726 027444 005437 002420      NEG     R8            ;GET CORRECT NUMBER
4727 027450 013737 002420 002432      MOV     R8,BITPST    ;SET UP FOR BINARY TO BIT POSITION
4728 027456 004737 036406      JSR     PC,SHIFTL    ;SHIFT
4729 027462 053737 002432 002562      BIS     BITPST,ALLINT ;SET UNITS THAT HAVE INTERRUPTED
4730 027470 005037 002424 14$:   CLR     R10          ;RESET COUNTER
4731 027474 020337 002562      CMP     R3,ALLINT    ;SEE IF ALL UNITS INTERRUPTED
4732 027500 001404                    BEQ     30$           ;RETURN TO SCHEDULER
4733 027502 012711 000100      MOV     #100,(R1)    ;ENABLE INTERUPTS IN CASE WE DID EXT SENSE.
4734 027506 000137 026546      JMP     2$           ;BRANCH IF ALL UNITS NOT INTERRUPTED
4735 027512 005037 002562 30$:   CLR     ALLINT       ;RESET ALL UNIT INTERRUPT FLAG
4736 027516 013737 002344 002352      MOV     TBLCNT,TBLNUM
4737 027524 000137 030552      JMP     25$          ;GO HOME
4738
4739
4740      ;IF NO REWIND COMMAND
4741
4742
4743 027530 005237 002464 15$:   INC     STAFLG       ;SET ONE UNIT STAT FLAG
4744 027534 012737 000000 002316      MOV     #RHADD,DISPLC ;GETTING CORRECT RH ADDRESS
4745 027542 004737 036076      JSR     PC,TBLDIS
4746 027546 017701 152544      MOV     @DISPLC,R1   ;R1 CONTAINS RH ADDRESS
4747 027552 005004                    CLR     R4            ;SET UP TIME OUT COUNTER
4748 027554 013702 002316      MOV     DISPLC,R2    ;GET TABLE ADDRESS
4749 027560 010137 002356      MOV     R1,RHADDR    ;STORE RH ADDRESS
4750 027564 016237 000004 002370      MOV     TMNUM(R2),TM78N ;STORE TM #
4751 027572 016237 000006 002372      MOV     TUNUM(R2),TU78N ;STORE TU #
4752 027600 005037 002374      CLR     INTRPT       ;CLEAR INTERRUPT FLAG

```

```

4753 027604          SETPRI #PRI00          ;ENABLE INTERRUPTS
4754 027612 005737 002374 16$: TST INTRPT          ;SEE IF INTERRUPT RECEIVED
4755 027616 001055          BNE 19$          ;BRANCH IF SO
4756 027620 005204          INC R4          ;INCREMENT TIME OUT COUNTER
4757 027622          BREAK
4758 027624 022704 177777    CMP #-1,R4          ;SEE IF TIMEOUT REACHED
4759 027630 001370          BNE 16$          ;BRANCH IF UNIT HAS NOT TIMED OUT
4760 027632 005237 002544    INC TIMER
4761 027636 022737 003000 002544  CMP #3000,TIMER
4762 027644 001362          BNE 16$
4763 027646 005037 002544    CLR TIMER
4764 027652 022737 000001 002314  CMP #1,BRTEST          ;SEE IF BRUTUS SEQUENCE
4765 027660 001002          BNE 40$          ;RETURN TO SCHEDULER
4766 027662 000137 030552    JMP 25$
4767 027666 005237 002542 40$: INC TIMEOUT
4768 027672 016237 000010 002074  MOV SPUNIT(R2),L$LUN ;FILL IN UNIT # FOR APT
4769 027700          ERRDF 3.,ERM001,ERRMS1
4770 027710 005037 002542    CLR TIMEOUT
4771 027714 005737 002210    TST BRUERR          ;SEE IF BRUTUS W/ERRORS
4772 027720 001402          BEQ 41$          ;BRANCH IF NOT
4773 027722 000137 030552    JMP 25$          ;BRANCH IF BRUTUS W/ERRORS TO RETURN
4774 027726 053737 002530 002540 41$: BIS UNITFL,EOTFLG ;SET EOT FLAG FOR UNIT THAT TIMED OUT
4775 027734 005237 002600    INC ALLEOT
4776 027740 053737 002530 002534  BIS UNITFL,DRPFLG ;SET FLAG THAT TELLS SCHEDULER A UNIT WAS DROPPED
4777 027746 000137 030552    JMP 25$          ;RETURN TO SCHEDULER
4778
4779          ;: INTERRUPT RECEIVED
4780
4781
4782 027752 005037 002374 19$: CLR INTRPT          ;
4783 027756          SETPRI PRITY          ;RAISE PRIORITY
4784 027764 013737 002532 002356  MOV MASBUS,RHADDR ;RHADDR CONTAINS RH THAT INTERRUPTED
4785 027772 005737 002516    TST DATRAN          ;SEE IF DATA TRANSFER
4786 027776 001161          BNE 22$          ;GO SERVICE INTERRUPT
4787 030000 013701 002356    MOV RHADDR,R1
4788 030004 005761 000016    TST ATTBIT(R1)      ;TEST FOR MASSBUS TERMINATION OF NDT INT
4789 030010 001007          BNE 26$          ;BRANCH TO HANDLE ATTENTION INTERRUPT
4790 030012 016137 000010 002430  MOV UNSLCT(R1),BNRYNB ;GET UNIT NUMBER OF FAULT
4791 030020 042737 177770 002430  BIC #177770,BNRYNB ;MASK DOWN TO UNIT NUMBER ONLY
4792 030026 000405          BR 27$          ;GO HANDLE INTERRUPT CODE
4793 030030 016137 000016 002430 26$: MOV ATTBIT(R1),BNRYNB ;GET TM
4794 030036 004737 036360    JSR PC,SHIFTR      ;CONVERT TO NUMBER
4795 030042 013737 002430 002432 27$: MOV BNRYNB,BITPST
4796 030050 004737 036406    JSR PC,SHIFTL      ;CONVERT TM NUMBER TO BIT POSITION
4797 030054 013705 002432    MOV BITPST,R5      ;R5 CONTAINS TM CLEAR BIT
4798 030060 013761 002430 000010  MOV BNRYNB,UNSLCT(R1);SELECT UNIT
4799 030066 013737 002430 002370  MOV BNRYNB,TM78N    ;STORE TM #
4800 030074 026237 000004 002430  CMP TMNUM(R2),BNRYNB;SEE IF TM THAT IS EXPECTED
4801 030102 001402          BEQ 28$          ;BRANCH TO SEE CORRECT TU
4802 030104 005237 002502    INC WTMFLG          ;SET WRONG TM FLAG
4803 030110 016103 000036 28$: MOV NDTICD(R1),R3 ;R3 CONTAINS NON DATA TRANSFER INTERRUPT
4804 030114 122703 000032    CMPB #32,R3        ;SEE IF TM FAULT B
4805 030120 001420          BEQ 20$          ;PROCESS INTERRUPT
4806 030122 122703 000034    CMPB #34,R3        ;SEE IF MB FAULT B
4807 030126 001415          BEQ 20$          ;PROCESS INTERRUPT
4808 030130 000303          SWAB R3          ;GET TU UNIT

```

```

4809 030132 042703 177774      BIC      #177774,R3      ;MASK OFF GARBAGE
4810 030136 010337 002372      MOV      R3,TU78N      ;STORE TU#
4811 030142 005737 002502      TST     WTMFLG        ;SEE IF WRONG TM
4812 030146 001005                BNE     20$           ;PROCESS INTERRUPT
4813 030150 026203 000006      CMP     TUNUM(R2),R3  ;SEE IF CORRECT TU
4814 030154 001402                BEQ     20$           ;PROCESS INTERRUPT
4815 030156 005237 002500      INC     WTUFLG        ;SET WRONG TU FLAG
4816 030162 116103 000036      20$:    MOVB   NDTICD(R1),R3  ;GET INTERRUPT CODE
4817 030166 042703 177700      BIC     #177700,R3    ;MASK OFF EXTRA BITS
4818 030172 010337 002526      MOV     R3,ICCODE     ;STORE INTERRUPT CODE
4819 030176 016137 000036 002522  MOV     NDTICD(R1),EXTERR ;
4820 030204 000337 002522      SWAB   EXTERR        ;
4821 030210 042737 177403 002522  BIC     #177403,EXTERR ;GET EXTENDED INFORMATION
4822 030216 006237 002522      ASR    EXTERR        ;
4823 030222 006237 002522      ASR    EXTERR        ;
4824 030226 006303                ASL    R3            ;MAKE WORD DISPLACEMENT
4825 030230 005737 002502      TST     WTMFLG        ;SEE IF WRONG TM INTERRUPT
4826 030234 001003                BNE     29$           ;BRANCH IF YES
4827 030236 005737 002500      TST     WTUFLG        ;SEE IF WRONG TU INTERRUPT
4828 030242 001404                BEQ     33$           ;BRANCH IF NO
4829 030244 012737 000143 002074 29$:    MOV     #143,L$LUN    ;STORE ASYNCHRONOUS INTERRUPT IN UNIT #
4830 030252 000403                BR     34$           ;PROCESS INTERRUPT
4831 030254 016237 000010 002074 33$:    MOV     SPUNIT(R2),L$LUN ;STORE P TABLE UNIT # FOR APT
4832 030262 022703 000071 34$:    CMP     #71,R3       ;SEE IF NON-EXISTENT INTERRUPT
4833 030266 101002                BHI     45$           ;BRANCH IF NOT
4834 030270 012703 000034      MOV     #34,R3       ;SET FOR NON-EXISTENT INTERRUPT
4835 030274 004773 002662      45$:    JSR    PC,@INTTBL(R3) ;CALL RESPECTIVE INTERRUPT SUBROUTINE
4836 030300 010561 000016      MOV     R5,ATTBIT(R1) ;CLEAR INTERRUPT BIT
4837 030304 022737 000143 002074  CMP     #143,L$LUN    ;SEE IF CORRECT UNIT
4838 030312 001412                BEQ     31$           ;BRANCH IF NOT
4839 030314 022737 000017 002526  CMP     #17,ICCODE    ;SEE IF TU ON LINE
4840 030322 001406                BEQ     31$           ;BRANCH IF YES
4841 030324 022737 000033 002526  CMP     #33,ICCODE    ;SEE IF TU FAULT B
4842 030332 001402                BEQ     31$           ;BRANCH IF YES
4843 030334 005237 002504      INC     INTROK        ;SET CORRECT INTERRUPT RECEIVED
4844 030340 000463      31$:    BR     17$           ;
4845 030342 116103 000012 22$:    MOVB   INTCDE(R1),R3  ;GET DATA TRANSFER INTERRUPT CODE
4846 030346 042703 177700      BIC     #177700,R3    ;MASK OFF EXTRA BITS
4847 030352 010337 002526      MOV     R3,ICCODE     ;
4848 030356 016137 000012 002522  MOV     INTCDE(R1),EXTERR ;
4849 030364 000337 002522      SWAB   EXTERR        ;
4850 030370 042737 177403 002522  BIC     #177403,EXTERR ;
4851 030376 006237 002522      ASR    EXTERR        ;
4852 030402 006237 002522      ASR    EXTERR        ;
4853 030406 006303                ASL    R3            ;MAKE WORD DISPLACEMENT
4854 030410 016237 000010 002074  MOV     SPUNIT(R2),L$LUN ;
4855 030416 022703 000071      CMP     #71,R3       ;SEE IF NON-EXISTENT INTERRUPT
4856 030422 101002                BHI     44$           ;BRANCH IF NOT
4857 030424 012703 000034      MOV     #34,R3       ;SET FOR NON-EXISTENT INTERRUPT
4858 030430 004773 002662      44$:    JSR    PC,@INTTBL(R3) ;CALL RESPECTIVE INTERRUPT SUBROUTINE
4859 030434 005061 000012      CLR    INTCDE(R1)    ;CLEAR DATA TRANSFER INTERRUPT CODE
4860 030440 005237 002504      INC     INTROK        ;SET CORRECT INTERRUPT RECEIVED
4861 030444 022737 000001 002314  CMP     #1,BRTEST     ;SEE IF BRUTUS OPTION
4862 030452 001416                BEQ     17$           ;END IF BRUTUS OPTION
4863 030454 022737 000070 002452  CMP     #70,CMMAND    ;SEE IF READ COMMAND
4864 030462 101012                BHI     17$           ;END OF ROUTINE ON WRITE

```



```

4865 030464 005737 002512          TST      DTCOMP          ;SEE IF SHOULD DATA COMPARE
4866 030470 001407                   BEQ      17$              ;BRANCH IF NOT
4867 030472 005037 002512          CLR      DTCOMP          ;RESET FLAG
4868 030476 005737 002226          23$:    TST      DATCHK          ;SEE IF DATA COMPARE OPTION CHOSEN
4869 030502 001407                   BEQ      17$              ;BRANCH IF NOT
4870 030504 004737 035122          JSR      PC,COMPARE      ;DO THE DATA CHECK
4871 030510 122761 000000 000016 17$:  CMPB    #0,ATTBIT(R1)    ;SEE IF SECOND INTERRUPT
4872 030516 001003                   BNE     21$              ;BRANCH IF NOT
4873 030520 005737 002504          TST      INTROK          ;SEE IF CORRECT INTERRUPT
4874 030524 001012                   BNE     25$              ;BRANCH IF SO
4875 030526 005037 002516          21$:    CLR      DATRAN          ;CLEAR DATA TRANSFER
4876 030532 005037 002502          CLR      WTMFLG          ;
4877 030536 005037 002500          CLR      WTUFLG          ;
4878 030542 012711 000100          MOV     #100,(R1)       ;REENABLE INTERRUPTS
4879 030546 000137 027530          JMP     15$              ;PROCESS SECOND INTERRUPT
4880 030552 005037 002504          25$:    CLR      INTROK          ;
4881 030556 005037 002502          CLR      WTMFLG          ;CLEAR WRONG TM FLAG
4882 030562 005037 002500          CLR      WTUFLG          ;CLEAR WRONG TU FLAG
4883 030566                   POP     <R5,R4,R3,R2,R1>
4884 030600 000207          RTS      PC

```

4885  
4886  
4887  
4888  
4889  
4890  
4891  
4892  
4893  
4894  
4895  
4896  
4897  
4898  
4899  
4900  
4901  
4902  
4903  
4904  
4905  
4906  
4907  
4908  
4909  
4910  
4911  
4912  
4913  
4914  
4915  
4916  
4917  
4918  
4919  
4920

```

;
; INTERRUPT SERVICE ROUTINES MODULE 3.3.1
;

```

```

;*BEGINROUTINE (MOD = 3.3.1 INTERRUPT RESPONSES)
;* BEGINSUBROUTINE (OO MASS BUS TERMINATION)
;* : CLEAR TIMER
;* : IF NOT BRUTUS OPTION
;* : THEN
;* : PRINT ERROR MESSAGE
;* : IF NOT BRUTUS WITH ERROR
;* : THEN
;* : GET UNIT #
;* : SET EOT FLAG
;* : SET DROP UNIT FLAG
;* : PRINT UNIT DROPPED
;* : ENDF
;* : ENDF
;* : SET TM CLEAR
;* : DO UNTIL TM READY SET OR TIME OUT REACHED
;* : IF TM READY NOT SET
;* : THEN
;* : INCREMENT TIMER
;* : IF TIMER REACHES MAX
;* : THEN
;* : IF NOT BRUTUS OPTION
;* : THEN
;* : PRINT MESSAGE
;* : ENDF
;* : ENDF
;* : ENDF
;* : ENDDO

```

```
4921 : * ENDSUBROUTINE
4922 : *
4923 : *
4924 : * BEGINSUBROUTINE (01 DONE)
4925 : * : IF NOT BRUTUS OPTION
4926 : * : : THEN
4927 : * : : GET RH ADDRESS
4928 : * : : IF TRANSFER ERROR
4929 : * : : : THEN
4930 : * : : : CALL RH TRANSFER SUBROUTINE
4931 : * : : : ENDF
4932 : * : : IF NOT DATA TRANSFER COMMAND
4933 : * : : : THEN
4934 : * : : : IF EOT BIT SET
4935 : * : : : : THEN
4936 : * : : : : IF CLOSE FILE PE
4937 : * : : : : : THEN
4938 : * : : : : : INCREMENT ALL EOT FLAG
4939 : * : : : : ENDF
4940 : * : : : : IF CLOSE FILE GCR
4941 : * : : : : : THEN
4942 : * : : : : : INCREMENT ALL EOT FLAG
4943 : * : : : : ENDF
4944 : * : : : ENDF
4945 : * : : ELSE
4946 : * : : STORE TABLE ADDRESS
4947 : * : : IF READ COMMAND
4948 : * : : : THEN
4949 : * : : : SET DATA COMPARE EXTENDED SENSE FLAG
4950 : * : : : IF RETRY FLAG SET
4951 : * : : : : THEN
4952 : * : : : : CLEAR RETRY FLAG
4953 : * : : : : INCREMENT READ RECOVERY
4954 : * : : : : PRINT RECOVERED
4955 : * : : : : INCREMENT APPROPRIATE READ FOR A STATUS ERROR
4956 : * : : : ENDF
4957 : * : : : IF ERROR CORRECT
4958 : * : : : : THEN
4959 : * : : : : PRINT ERROR
4960 : * : : : : IF DOUBLE TRACK ERROR
4961 : * : : : : : THEN
4962 : * : : : : : INCREMENT APPROPRIATE STAT
4963 : * : : : : : ELSE
4964 : * : : : : : INCREMENT SINGLE TRACK STAT
4965 : * : : : : ENDF
4966 : * : : : ENDF
4967 : * : : ELSE
4968 : * : : IF RETRY NOT SET
4969 : * : : : THEN
4970 : * : : : IF DONE+1
4971 : * : : : : THEN
4972 : * : : : : SETUP FOR WRITE RETRY
4973 : * : : : : ELSE
4974 : * : : : : UPDATE WRITE STATUS ERROR
4975 : * : : : ENDF
4976 : * : : ELSE
```

4977  
4978  
4979  
4980  
4981  
4982  
4983  
4984  
4985  
4986  
4987  
4988  
4989  
4990  
4991  
4992  
4993  
4994  
4995  
4996  
4997  
4998  
4999  
5000  
5001  
5002  
5003  
5004  
5005  
5006  
5007  
5008  
5009  
5010  
5011  
5012  
5013  
5014  
5015  
5016  
5017  
5018  
5019  
5020  
5021  
5022  
5023  
5024  
5025  
5026  
5027  
5028  
5029  
5030  
5031  
5032

```

: * : : : : : IF DONE+0
: * : : : : : : THEN
: * : : : : : : IF WRITE RETRY 4TH TIME
: * : : : : : : : THEN
: * : : : : : : : UPDATE STATUS OR MEDIA ERROR
: * : : : : : : : PRINT WRITE RECOVERY
: * : : : : : : : CLEAR RETRY FLAGS
: * : : : : : : ELSE
: * : : : : : : INCREMENT RETRY COUNT
: * : : : : : : ENDIF
: * : : : : : ELSE
: * : : : : : IF WRITE BAD 4TH TIME
: * : : : : : : THEN
: * : : : : : : PRINT UNRECOVERABLE
: * : : : : : : INCREMENT STAT
: * : : : : : : CLEAR RETRY FLAG
: * : : : : : : IF NOT BRUTUS WITH ERROR
: * : : : : : : : THEN
: * : : : : : : : GET TABLE #
: * : : : : : : : DROP UNIT
: * : : : : : : : SET EOT
: * : : : : : : : SET DROP FLAG
: * : : : : : : : INCREMENT ALL EOT
: * : : : : : : : PRINT DROP UNIT
: * : : : : : : : INCREMENT DROP STAT
: * : : : : : : : ENDIF
: * : : : : : : ELSE
: * : : : : : : PRINT ERROR
: * : : : : : : ENDIF
: * : : : : : : ENDIF
: * : : : : : : ENDIF
: * : : : : : : ENDIF
: * : : : : : ELSE
: * : : : : : CLEAR RETRY FLAG
: * : : : : : ENDIF
: * : : : : : ENDSUBROUTINE
: *
: * BEGINSUBROUTINE (MOD = 3.3.2.1 TRANSFER OR PARITY MASSBUS ERRORS)
: * : STORE UNIT NUMBER
: * : PRINT ERROR
: * : CREATE TABLE OFFSET
: * : INCREMENT MASSBUS ERROR COUNT
: * : IF MAXIMUM PARITY ERROR REACHED
: * : : THEN
: * : : PRINT UNIT DROPPED
: * : : SET EOT FLAG
: * : : SET DROP FLAG
: * : : CLEAR MAXIMUM ERROR COUNT
: * : : INCREMENT DROP UNIT
: * : : ELSE
: * : : IF DATA TRANSFER COMMAND
: * : : : THEN
: * : : : IF RETRIES ALLOWED
```

```
5033 : * : : : : THEN
5034 : * : : : : IF RETRY NOT ALREADY IN PROGRESS
5035 : * : : : : THEN
5036 : * : : : : CLEAR VISIT FLAG
5037 : * : : : : SET RETRY=1
5038 : * : : : : ENDIF
5039 : * : : : : ENDIF
5040 : * : : : : ENDIF
5041 : * : : : : ENDIF
5042 : * : : : : ENDSUBROUTINE
5043 : *
5044 : * BEGINSUBROUTINE (02 TM INTERRUPT)
5045 : * : IF NOT BRUTUS OPTION
5046 : * : THEN
5047 : * : IF UNKNOWN TAPE OPTION
5048 : * : THEN
5049 : * : INCREMENT FILE #
5050 : * : IF ONE RECORD
5051 : * : THEN
5052 : * : IF MORE THAN ONE FILE
5053 : * : THEN
5054 : * : CLEAR RECORD
5055 : * : JMP LEOT
5056 : * : ELSE
5057 : * : CLEAR RECORD
5058 : * : ENDIF
5059 : * : ENDIF
5060 : * : ELSE
5061 : * : PRINT ERROR
5062 : * : IF DATA TRANSFER COMMAND
5063 : * : THEN
5064 : * : UPDATE STATUS ERROR READ FWD OR REVERSE
5065 : * : ENDIF
5066 : * : ENDIF
5067 : * : ENDIF
5068 : * : ENDSUBROUTINE
5069 : *
5070 : *
5071 : *
5072 : * BEGINSUBROUTINE (03 BOT)
5073 : * : IF NO BRUTUS OPTION
5074 : * : THEN
5075 : * : PRINT ERROR (UNEXPECTED BEGINNING OF TAPE)
5076 : * : IF CANNED SEQUENCE
5077 : * : THEN
5078 : * : SET EOT FLAG (DROP UNIT)
5079 : * : PRINT UNIT DROPPED
5080 : * : UPDATE DROPPED UNIT STAT
5081 : * : ENDIF
5082 : * : STATUS ERRORS READ REV
5083 : * : ENDIF
5084 : * : ENDSUBROUTINE
5085 : *
5086 : *
5087 : *
5088 : * BEGINSUBROUTINE (04 EOT INTERRUPT RECEIVED)
```

```
5089 : * : GET TABLE #
5090 : * : IF NOT BRUTUS
5091 : * : : THEN
5092 : * : : IF EOT FLAG SET
5093 : * : : : THEN
5094 : * : : : INCREMENT ALL EOT FLAG
5095 : * : : : ELSE
5096 : * : : : IF BRUTUS W/ERRORS
5097 : * : : : : THEN
5098 : * : : : : PRINT ERROR
5099 : * : : : : ENDIF
5100 : * : : : IF DATA TRANSFER
5101 : * : : : : THEN
5102 : * : : : : IF EXTENDED ERROR 1
5103 : * : : : : : THEN
5104 : * : : : : : IF NOT UNKNOWN TAPE
5105 : * : : : : : : THEN
5106 : * : : : : : : CLEAR RETRY FLAGS
5107 : * : : : : : : SET UP TO SPACE REVERSE AND CLOSE FILE
5108 : * : : : : : : ENDIF
5109 : * : : : : : ELSE
5110 : * : : : : : IF NOT UNKNOWN TAPE
5111 : * : : : : : : THEN
5112 : * : : : : : : SET UP TO CLOSE FILE
5113 : * : : : : : : ENDIF
5114 : * : : : : : : UPDATE BYTES READ
5115 : * : : : : : : ENDIF
5116 : * : : : : : ENDIF
5117 : * : : : : SET EOT FLAG
5118 : * : : : : ENDIF
5119 : * : : : : ENDIF
5120 : * : : : : ENDSUBROUTINE
5121 : * : : : :
5122 : * : : : :
5123 : * : : : :
5124 : * : : : : BEGINROUTINE (05 LEOT)
5125 : * : : : : IF BRUTUS WITH ERRORS
5126 : * : : : : : THEN
5127 : * : : : : : : PRINT MESSAGE
5128 : * : : : : : : ENDIF
5129 : * : : : : IF NOT BRUTUS OPTION
5130 : * : : : : : THEN
5131 : * : : : : : : GET TABLE #
5132 : * : : : : : : INCREMENT ALL EOT
5133 : * : : : : : : SET EOT FLAG
5134 : * : : : : : : ENDIF
5135 : * : : : : : : ENDRROUTINE
5136 : * : : : :
5137 : * : : : :
5138 : * : : : :
5139 : * : : : : BEGINSUBROUTINE (10 FPT, 11 NOT READY, 12 NOT AVL, 13 OFFLINE, 14 NON EX)
5140 : * : : : : IF NOT BRUTUS OPTION
5141 : * : : : : : THEN
5142 : * : : : : : : IF INTERRUPT=10
5143 : * : : : : : : : THEN
5144 : * : : : : : : : PRINT ERROR (A WRITE WAS ATTEMPTED ON A FILE PROTECTED TAPE)
```

```
5145 : * : : ENDF
5146 : * : : IF INTERRUPT=11
5147 : * : : : THEN
5148 : * : : : PRINT ERROR (TU IS ON LINE BUT TAPE IS REWINDING OR LOADING)
5149 : * : : ENDF
5150 : * : : IF INTERRUPT=12
5151 : * : : : THEN
5152 : * : : : PRINT ERROR (TU IS NOT SWITCHED TO THIS PORT, BUT IS ON LINE.)
5153 : * : : ENDF
5154 : * : : IF INTERRUPT=13
5155 : * : : : THEN
5156 : * : : : PRINT ERROR (TU IS NOT SWITCHED ON LINE BUT TAPE IS REWINDING OR LOADI
5157 : * : : ENDF
5158 : * : : IF INTERRUPT=14
5159 : * : : : THEN
5160 : * : : : PRINT ERROR (TU DOES NOT EXIST OR POWER IS OFF)
5161 : * : : ENDF
5162 : * : : IF INTERRUPT=15
5163 : * : : : THEN
5164 : * : : : PRINT ERROR (BLANK TAPE)
5165 : * : : ENDF
5166 : * : : IF INTERRUPT=16
5167 : * : : : THEN
5168 : * : : : PRINT ERROR (ILLEGAL INTERRUPT)
5169 : * : : ENDF
5170 : * : : IF CANNED SEQUENCE
5171 : * : : : THEN
5172 : * : : : SET EOT FLAG (DROP UNIT)
5173 : * : : ENDF
5174 : * : : IF NOT BRUTUS W/ERRORS
5175 : * : : : THEN
5176 : * : : : GET TABLE #
5177 : * : : : SET EOT FLAG
5178 : * : : : SET DROP UNIT FLAG
5179 : * : : : INCREEMNT ALL EOT FLAG
5180 : * : : : PRINT UNIT DROPPED
5181 : * : : : INCREMENT DROP UNIT STATISTIC
5182 : * : : ENDF
5183 : * : ENDF
5184 : * ENDSUBROUTINE
5185 : *
5186 : *
5187 : *
5188 : * BEGINSUBROUTINE (17 ON LINE)
5189 : * : IF NOT BRUTUS OPTION
5190 : * : : THEN
5191 : * : : PRINT MESSAGE
5192 : * : ENDF
5193 : * ENDSUBROUTINE
5194 : *
5195 : *
5196 : *
5197 : * BEGINSUBROUTINE (20 LONG RECORD, 21 SHORT RECORD)
5198 : * : IF NOT BRUTUS OPTION
5199 : * : : THEN
5200 : * : : IF NOT UNKNOWN TAPE
```

```

5201      : * : : : THEN
5202      : * : : : IF INTERRUPT=20
5203      : * : : : : THEN
5204      : * : : : : PRINT ERROR LONG RECORD
5205      : * : : : : ELSE
5206      : * : : : : PRINT ERROR SHORT RECORD
5207      : * : : : : ENDIF
5208      : * : : : IF READ FORWARD COMMAND
5209      : * : : : THEN
5210      : * : : : IF ABLE TO ISSUE READ SENSE
5211      : * : : : THEN
5212      : * : : : : IF DOUBLE TRACK ERROR
5213      : * : : : : : THEN
5214      : * : : : : : UPDATE DOUBLE TRACK READ FORWARD STAT
5215      : * : : : : : ENDIF
5216      : * : : : : IF SINGLE TRACK ERROR AND NO DOUBLE TRACK ERROR
5217      : * : : : : : THEN
5218      : * : : : : : UPDATE SINGLE TRACK READ FORWARD STAT
5219      : * : : : : : ENDIF
5220      : * : : : : ENDIF
5221      : * : : : INCREMENT STATUS ERROR READ FORWARD
5222      : * : : : ELSE
5223      : * : : : IF ABLE TO ISSUE READ SENSE COMMAND
5224      : * : : : THEN
5225      : * : : : : IF DOUBLE TRACK ERROR
5226      : * : : : : : THEN
5227      : * : : : : : UPDATE DOUBLE TRACK RR STAT
5228      : * : : : : : ENDIF
5229      : * : : : : IF SINGLE TRACK AND NOT DOUBLE TRACK ERROR
5230      : * : : : : : THEN
5231      : * : : : : : UPDATE SINGLE TRACK RR STAT
5232      : * : : : : : ENDIF
5233      : * : : : : INCREMENT STATUS ERROR RR
5234      : * : : : : ENDIF
5235      : * : : : ENDIF
5236      : * : : : IF READ COMMAND
5237      : * : : : THEN
5238      : * : : : : UPDATE BYTES READ FORWARD
5239      : * : : : : ELSE
5240      : * : : : : UPDATE BYTES READ REVERSE
5241      : * : : : : ENDIF
5242      : * : : : ENDIF
5243      : * : : : ENDSUBROUTINE
5244      : * : : :
5245      : * : : :
5246      : * : : :
5247      : * : : : BEGINSUBROUTINE (22 RETRY)
5248      : * : : : IF NOT BRUTUS OPTION
5249      : * : : : : THEN
5250      : * : : : : : PRINT ERROR (THE INITIAL OPERATION SHOULD BE REPEATED)
5251      : * : : : : : IF RETRIES SELECTED
5252      : * : : : : : : THEN
5253      : * : : : : : : SET RETRY FLAG=2
5254      : * : : : : : : ENDIF
5255      : * : : : : : ENDIF
5256      : * : : : : ENDSUBROUTINE

```

```
5257 : *
5258 : *
5259 : *
5260 : * BEGINSUBROUTINE (24 READ OPP)
5261 : : IF NOT BRUTUS OPTION
5262 : : : THEN
5263 : : : PRINT ERROR (THE INITIAL READ SHOULD BE PERFORMED IN THE OPPOSITE DIRECTION)
5264 : : : IF RETRIES SELECTED
5265 : : : : THEN
5266 : : : : SET RETRY FLAG=3
5267 : : : : ENDF
5268 : : : ENDF
5269 : * ENDSUBROUTINE
5270 : *
5271 : *
5272 : *
5273 : * BEGINSUBROUTINE (24 UNREAD)
5274 : : IF NOT BRUTUS OPTION
5275 : : : THEN
5276 : : : PRINT ERROR (READ RETRIES HAVE FAILED TO READ THE RECORD)
5277 : : : UPDATE NON RECOVERABLE READ REVERSE OR
5278 : : : UPDATE NON RECOVERABLE READ FORWARD
5279 : : : ENDF
5280 : : CLEAR RETRY FLAG
5281 : * ENDSUBROUTINE
5282 : *
5283 : *
5284 : *
5285 : * BEGINSUBROUTINE (25 ERROR)
5286 : : IF NOT BRUTUS OPTION
5287 : : : THEN
5288 : : : PRINT ERROR WITH EXTENDED SENSE
5289 : : : IF NO RETRY ALLOWED
5290 : : : : THEN
5291 : : : : : IF READ COMMAND
5292 : : : : : : THEN
5293 : : : : : : INCREMENT STATUS ERROR READ F OR R.
5294 : : : : : : ELSE
5295 : : : : : : INCREMENT WRITE STATS
5296 : : : : : ENDF
5297 : : : : ELSE
5298 : : : : : IF WRITE
5299 : : : : : : THEN
5300 : : : : : : CLEAR VISIT FLAG
5301 : : : : : : IF RETRIED MAXIMUM
5302 : : : : : : : THEN
5303 : : : : : : : PRINT ERROR
5304 : : : : : : : UPDATE STATS
5305 : : : : : : : CLEAR RETRY COUNT
5306 : : : : : : : CLEAR RETRY FLAG
5307 : : : : : : : IF NOT BRUTUS WITH ERRORS
5308 : : : : : : : : THEN
5309 : : : : : : : : SET EOT FLAG
5310 : : : : : : : : INCREMENT ALL EOT FLAG
5311 : : : : : : : : SET DROP FLAG
5312 : : : : : : : : PRINT UNIT DROPPED
```



```

5313      : * : : : : : : INCREMENT UNIT DROPPED FLAG
5314      : * : : : : : : ELSE
5315      : * : : : : : : MOV 4 TO RETRY FLAG
5316      : * : : : : : : ENDIF
5317      : * : : : : : : ENDIF
5318      : * : : : : : : ENDIF
5319      : * : : : : : : ENDIF
5320      : * : : : : : : ENDIF
5321      : * : : : : : : ENDSUBROUTINE
5322      : * : : : : : :
5323      : * : : : : : :
5324      : * : : : : : :
5325      : * : : : : : : BEGINSUBROUTINE (26 EOT ERROR)
5326      : * : : : : : : IF BRUTUS W/ERRORS OR CANNED SEQUENCE NOT EQUAL TO 1
5327      : * : : : : : : THEN
5328      : * : : : : : : PRINT ERROR (A WRITE ERROR HAS OCCURRED BEYOND THE EOT MARKER)
5329      : * : : : : : : SET RETRY FLAG=6
5330      : * : : : : : : INCREMENT ALL EOT
5331      : * : : : : : : INCREMENT EOT FLAG
5332      : * : : : : : : ENDIF
5333      : * : : : : : : ENDSUBROUTINE
5334      : * : : : : : :
5335      : * : : : : : :
5336      : * : : : : : :
5337      : * : : : : : : BEGINSUBROUTINE (27 BAD TAPE)
5338      : * : : : : : : IF NOT BRUTUS
5339      : * : : : : : : THEN
5340      : * : : : : : : PRINT ERROR (TAPE POSITION HAS BEEN LOST)
5341      : * : : : : : : IF CANNED SEQUENCE OR UNKNOWN TAPE
5342      : * : : : : : : THEN
5343      : * : : : : : : SET EOT FLAG (DROP UNIT)
5344      : * : : : : : : INCREMENT ALL EOT FLAG
5345      : * : : : : : : INCREMENT DROP FLAG
5346      : * : : : : : : UPDATE DROP UNIT STAT
5347      : * : : : : : : ENDIF
5348      : * : : : : : : ENDIF
5349      : * : : : : : : ENDSUBROUTINE
5350      : * : : : : : :
5351      : * : : : : : :
5352      : * : : : : : :
5353      : * : : : : : :
5354      : * : : : : : :
5355      : * : : : : : :
5356      : * : : : : : : BEGINSUBROUTINE (30 TM FAULT A, 31 TU FAULT A)
5357      : * : : : : : : IF NOT BRUTUS
5358      : * : : : : : : THEN
5359      : * : : : : : : PRINT ERROR (TAPE UNIT HAS FAILED)
5360      : * : : : : : : IF CANNED SEQUENCE OR UNKNOWN TAPE
5361      : * : : : : : : THEN
5362      : * : : : : : : SET EOT FLAG (DROP UNIT)
5363      : * : : : : : : SET DROP FLAG
5364      : * : : : : : : INCREMENT ALL EOT FLAG
5365      : * : : : : : : PRINT UNIT DROPPED
5366      : * : : : : : : INCREMENT DROP UNIT STAT
5367      : * : : : : : : ENDIF
5368      : * : : : : : : ENDIF

```

```

5369      : * ENDSUBROUTINE
5370      : *
5371      : *
5372      : *
5373      : *
5374      : * BEGINSUBROUTINE (33 TU FAULT B)
5375      : * : IF NOT BRUTUS
5376      : * : : THEN
5377      : * : : PRINT ERROR
5378      : * : : ENDF
5379      : * ENDSUBROUTINE
5380      : *
5381      : *
5382      : *
5383      : * BEGINSUBROUTINE (32 TM FAULT, 34 MB FAULT)
5384      : * : CLEAR TIME OUT COUNTER
5385      : * : FIND RH ADDRESS THAT INTERRUPTED
5386      : * : FIND TM THAT INTERRUPTED
5387      : * : IF NOT BRUTUS
5388      : * : : THEN
5389      : * : : PRINT CORRECT ERROR MESSAGE
5390      : * : : IF CANNED SEQUENCE
5391      : * : : : THEN
5392      : * : : : DROP ALL UNITS ON TM
5393      : * : : : SET EOT FLAG
5394      : * : : : SET DROP FLAG
5395      : * : : : INCREMENT APPROPRIATE ALL EOT FLAG
5396      : * : : : INCREMENT APPROPRIATE DROP UNIT STAT
5397      : * : : : ENDF
5398      : * : : ENDF
5399      : * : : INITIATE A TM CLEAR
5400      : * : : DO UNTIL TM READY OR TIME OUT
5401      : * : : : INCREMENT TIMEOUT COUNTER
5402      : * : : : TEST TM READY
5403      : * : : ENDDO
5404      : * : : IF NOT BRUTUS
5405      : * : : : THEN
5406      : * : : : : IF TIME OUT
5407      : * : : : : : THEN
5408      : * : : : : : PRINT UNIT THAT TIMED OUT
5409      : * : : : : ENDF
5410      : * : : ENDF
5411      : * ENDSUBROUTINE
5412      : * ENDRoutine
5413      : *
5414      : * INTERRUPT SERVICE ROUTINES MODULE 3.3.1
5415      : *
5416      : *
5417      : *

```

```

5418 030602 MBTERM: PUSH <R4>
5419 030604 005004 CLR R4 ;SET UP TIME OUT COUNTER
5420 030606 022737 000001 002314 CMP #1, BRTEST ;SEE IF BRUTUS OPTION
5421 030614 001424 BEQ 1$ ;BRANCH IF BRUTUS
5422 030616 005737 002516 TST DATRAN ;DATA TRANSFER TERMINATION ?
5423 030622 001005 BNE 4$ ;YES , PRINT CORRECT MESSAGE
5424 030624 ERRDF 33.,ERM035,ERRMS1 ;NO , PRINT NON DATA XFER MESSAGE

```

```

5425 030634 000404          BR      5$          :CONTINUE
5426 030636          4$:  ERRDF  33.,ERM032,ERRMS1
5427 030646 005737 002210  5$:  TST   BRUERR          :SEE IF BRUTUS W/ERRORS
5428 030652 001005          BNE   1$          :BRANCH IF BRUTUS W/ERRORS
5429 030654 013737 002352 002432  MOV   TBLNUM,BITPST
5430 030662 004737 035024  JSR   PC,DROPIT          :GO DROP THE UNIT
5431 030666 052761 040000 000000 1$:  BIS   #BIT14,BASE(R1)  :SET RH11/70 ERROR CLEAR
5432 030674 042761 040000 000000  BIC   #BIT14,BASE(R1)  :CLEAR 'TRE' BIT IN TM78
5433 030702 052761 040000 000052  BIS   #BIT14,INTRDA(R1) :SET TM CLEAR
5434 030710 032761 100000 000052 3$:  BIT   #BIT15,INTRDA(R1) :SEE IF TM READY
5435 030716 001014          BNE   2$          :BRANCH IF SET
5436 030720 005204          INC   R4
5437 030722 022704 177777  CMP   #-1,R4          :SEE IF TIMED OUT
5438 030726 001370          BNE   3$
5439 030730 022737 000001 002314  CMP   #1,BRTEST
5440 030736 001404          BEQ   2$          :IF BRUTUS
5441 030740          ERRDF  29.,ERM028,ERRMS1 :RETURN
5442 030750          2$:  POP   <R4>
5443 030752 000207          RTS   PC
5444
5445
5446 030754          DONE:  PUSH  <R1,R2,R3,R4,R5>
5447 030766 022737 000001 002314  CMP   #1,BRTEST
5448 030774 001004          BNE   8$          :IF NOT BRUTUS THEN CONTINUE
5449 030776 005037 002604  CLR   RTYFLG
5450 031002 000137 032100  JMP
5451 031006 012737 000000 002316 8$:  MOV   #RHADD,DISPLC    :SET UP TO FIND UNIT
5452 031014 004737 036076  JSR   PC,TBLDIS        :GET ADDRESS
5453 031020 017701 151272  MOV   @DISPLC,R1       :STORE RH ADDRESS IN R1
5454 031024 011103          MOV   (R1),R3          :R3 CONTAINS CNTNTS OF RH ADDRESS
5455 031026 032703 060000  BIT   #60000,R3        :SEE IF TRE OR MCPE ERROR
5456 031032 001404          BEQ   9$          :BRANCH IF NO ERRORS
5457 031034 004737 032114  JSR   PC,RHTREB        :CALL TRE OR MCPE ERROR ROUTINE
5458 031040 000137 032100  JMP
5459 031044 005737 002516  9$:  TST   DATRAN          :SEE IF DATA TRANSFER
5460 031050 001062          BNE   10$         :BRANCH IF YES
5461 031052 013737 002352 002432  MOV   TBLNUM,BITPST
5462 031060 004737 036406  JSR   PC,SHIFTL        :GET UNIT BIT MAP
5463 031064 006237 002432  ASR   BITPST           :CORRECT MAP
5464 031070 033737 002432 002540  BIT   BITPST,EOTFLG    :SEE IF EOT FLAG SET
5465 031076 001023          BNE   25$         :BRANCH IF YES
5466 031100 005737 002434  TST   CHKBOT          :CHECK BOT STATUS FLAG SET
5467 031104 001416          BEQ   26$         :NO EXIT
5468 031106 032761 002000 000020  BIT   #BIT10,DSEO(R1)  :IS BOT SET
5469 031114 001012          BNE   26$         :YES , EXIT
5470 031116          ERRDF  40.,ERM036,ERRMS1 :NO , ERROR SHOULD BE !
5471 031126 053737 002432 002534  BIS   BITPST,DRPFLG    :DROP UNIT
5472 031134 053737 002432 002540  BIS   BITPST,EOTFLG
5473 031142 000137 032100  26$:  JMP   7$          :GET OUT
5474 031146 022737 000041 002452 25$:  CMP   #CLFLPE,CMMAND   :SEE IF CLOSE TAPE PE
5475 031154 001004          BNE   11$         :BRANCH IF NOT
5476 031156 005237 002600  INC   ALLEOT
5477 031162 000137 032100  JMP   7$          :GET OUT
5478 031166 022737 000043 002452 11$:  CMP   #CLFGCR,CMMAND   :SEE IF CLOSE TAPE GCR
5479 031174 001402          BEQ   14$         :BRANCH IF YES
5480 031176 000137 032100  JMP   7$

```

```

5481 031202 005037 002604      14$: CLR      RTYFLG      ;RESET RETRY FLAG
5482 031206 005237 002600      INC      ALLEOT
5483 031212 000137 032100      JMP      7$          ;GET OUT
5484 031216 013702 002316      MOV      DISPLC,R2   ;STORE UNIT TABLE ADDRESS IN R2
5485 031222 022737 000070 002452 10$: CMP      #70,CMMAND ;SEE IF READ OR WRITE COMMAND
5486 031230 101146      BHI      3$          ;BRANCH IF WRITE COMMAND
5487 031232 005237 002512      INC      DTCOMP      ;SET FOR DATA COMPARE
5488 031236 005737 002604      TST      RTYFLG      ;SEE IF RETRY SET
5489 031242 001423      BEQ      2$          ;BRANCH IF NOT
5490 031244 005037 002604      CLR      RTYFLG      ;RESET RETRY FLAG
5491 031250 022737 000077 002462  CMP      #REDREV,RETRCD
5492 031256 001003      BNE      15$
5493 031260 005262 000016      INC      SERRRV(R2) ;UPDATE STATUS ERROR READ REV
5494 031264 000402      BR      16$
5495 031266 005262 000014      15$: INC      SERRFO(R2) ;UPDATE STATUS ERROR READ FWD
5496 031272      16$: PRINTB #FORM22
5497 031312 032761 002000 000012 2$: BIT      #2000,INTCDE(R1) ;SEE IF COMMAND EXECUTED PERFECTLY
5498 031320 001456      BEQ      4$          ;BRANCH IF YES
5499 031322 005237 002524      INC      EXTDSN      ;UPDATE EXTENDED SENSE FLAG
5500 031326      ERRSOFT 6.,ERM005,ERRMS1
5501 031336 005737 002606      TST      SENFLG      ;SEE IF READ SENSE FAILED
5502 031342 001045      BNE      4$          ;BRANCH IF YES
5503 031344 022737 000077 002452  CMP      #REDREV,CMMAND ;SEE IF READ REVERSE
5504 031352 001421      BEQ      13$        ;BRANCH IF YES
5505 031354 132737 000002 003251  BITB     #2,ECCSTA    ;SEE IF SINGLE TRACK ERROR
5506 031362 001403      BEQ      12$        ;BRANCH IF NOT
5507 031364 005262 000052      INC      DBLTRF(R2) ;INCREMENT DOUBLE TRACK ERROR FORWARD
5508 031370 000432      BR      4$
5509 031372 132737 000001 003251 12$: BITB     #1,ECCSTA    ;SEE IF DOUBLE TRACK ERROR
5510 031400 001003      BNE      23$        ;BRANCH IF NOT
5511 031402 005262 000132      INC      OTHRFO(R2)
5512 031406 000423      BR      4$
5513 031410 005262 000056      23$: INC      SNGTRF(R2) ;INCREMENT SINGLE TRACK ERROR FORWARD
5514 031414 000420      BR      4$
5515 031416 132737 000002 003251 13$: BITB     #2,ECCSTA    ;SEE IF DOUBLE TRACK ERROR READ REV
5516 031424 001403      BEQ      17$        ;BRANCH IF NOT
5517 031426 005262 000054      INC      DBLTRR(R2) ;UPDATE STATE
5518 031432 000411      BR      4$
5519 031434 132737 000001 003251 17$: BITB     #1,ECCSTA    ;SEE IF SINGLE TRACK ERROR READ REV
5520 031442 001003      BNE      24$        ;BRANCH IF NOT
5521 031444 005262 000134      INC      OTHRRV(R2)
5522 031450 000402      BR      4$
5523 031452 005262 000060      24$: INC      SNGTRR(R2) ;UPDATE STATS
5524 031456 022737 000070 002452 4$: CMP      #70,CMMAND ;SEE IF READ COMMAND
5525 031464 101030      BHI      3$          ;BRANCH IF NOT
5526 031466 022737 000077 002452  CMP      #REDREV,CMMAND ;SEE IF READ REVERSE COMMAND
5527 031474 001412      BEQ      1$          ;BRANCH IF READ REVERSE
5528 031476 012703 000040      MOV      #BYRFLO,R3 ;STORE OFFSETS
5529 031502 012704 000036      MOV      #BYRFMD,R4 ;IN REGISTERS
5530 031506 012705 000034      MOV      #BYRFHI,R5 ;FOR DECIMAL CONVERSION SUBROUTINE
5531 031512 004737 035640      JSR      PC,DECMAL  ;STORE BYTES READ
5532 031516 000137 032100      JMP      7$          ;GO TO RETURN
5533 031522 012703 000046      1$: MOV      #BYRRLO,R3 ;STORE OFFSETS
5534 031526 012704 000044      MOV      #BYRRMD,R4 ;IN REGISTERS
5535 031532 012705 000042      MOV      #BYRRHI,R5 ;FOR CONVERSION AND UPDATE PURPOSES
5536 031536 004737 035640      JSR      PC,DECMAL  ;STORE BYTES READ REVERSE

```

```

5537 031542 000137 032100      JMP      7$          ;GO TO RETURN
5538 031546 005737 002604      3$:  TST      RTYFLG      ;TEST RETRY
5539 031552 001027          BNE      5$          ;BRANCH IF RETRY IN PROGRESS
5540 031554 032761 002000 000012  BIT      #2000,INTCDE(R1) ;SEE IF DONE+1
5541 031562 001412          BEQ      18$         ;BRANCH IF NO
5542 031564 012737 000004 002604  MOV      #4,RTYFLG     ;SETUP WRITE RETRY
5543 031572 005237 002524          INC      EXTDSN
5544 031576          ERRSOFT 36.,ERM003,ERRMS1
5545 031606 000534          BR      7$
5546 031610 012703 000032      18$:  MOV      #BYWRLO,R3     ;STORE OFFSETS
5547 031614 012704 000030      MOV      #BYWRMD,R4     ;IN REGISTERS
5548 031620 012705 000026      MOV      #BYWRHI,R5
5549 031624 004737 035640      JSR      PC,DECIMAL     ;UPDATE STATS
5550 031630 000523          BR      7$
5551 031632 032761 002000 000012  5$:  BIT      #2000,INTCDE(R1) ;SEE IF DONE+1
5552 031640 001047          BNE      21$         ;BRANCH IF DONE+1
5553 031642 022737 000003 002560  CMP      #3,WRTYCT      ;SEE IF WRITE RETRY COUNT=4TH TIME
5554 031650 001111          BNE      6$          ;BRANCH IF LESS THAN FOURTH TIME
5555 031652 012703 000032      MOV      #BYWRLO,R3     ;STORE
5556 031656 012704 000030      MOV      #BYWRMD,R4     ;OFFSETS INTO
5557 031662 012705 000026      MOV      #BYWRHI,R5     ;REGISTERS
5558 031666 004737 035640      JSR      PC,DECIMAL     ;UPDATE BYTES WRITTEN
5559 031672 005737 002514      TST      ERAFLG        ;SEE IF MEDIA ERROR OR STATUS ERROR
5560 031676 001003          BNE      19$         ;BRANCH IF MEDIA ERROR
5561 031700 005262 000130      INC      OTHWRT(R2)     ;UPDATE STATUS ERROR
5562 031704 000402          BR      20$
5563 031706 005262 000050      19$:  INC      MEDIAE(R2)     ;UPDATE MEDIA ERROR
5564 031712          20$:  PRINTB  #FORM22
5565 031732 005037 002560      CLR      WRTYCT         ;RESET COUNTER
5566 031736 005037 002556      CLR      WRTCNT        ;RESET WRITE RETRY COUNTER
5567 031742 005037 002546      CLR      BADCNT        ;RESET
5568 031746 005037 002514      CLR      ERAFLG        ;FLAGS
5569 031752 005037 002604      CLR      RTYFLG
5570 031756 000450          BR      7$          ;BRANCH OUT
5571 031760 022737 000004 002546  21$:  CMP      #4,BADCNT     ;SEE IF FAILED 4 TIMES
5572 031766 001407          BEQ      22$         ;BRANCH IF YES
5573 031770 005237 002524      INC      EXTDSN        ;UPDATE EXTENDED SENSE
5574 031774          ERRSOFT 4.,ERM003,ERRMS1
5575 032004 000435          BR      7$
5576 032006 005037 002546      22$:  CLR      BADCNT        ;RESET ALL RETRY COUNTERS
5577 032012 005037 002514      CLR      ERAFLG
5578 032016 005037 002560      CLR      WRTYCT
5579 032022 005037 002556      CLR      WRTCNT
5580 032026 005262 000020      INC      NRECWR(R2)     ;UPDATE STATS
5581 032032 005237 002524      INC      EXTDSN        ;EXTENDED SENSE
5582 032036          ERRHRD 35.,ERM034,ERRMS1
5583 032046 005037 002604      CLR      RTYFLG        ;RESET RETRY FLAG
5584 032052 005737 002210      TST      BRUERR        ;SEE IF BRUTUS W/ERRORS
5585 032056 001010          BNE      7$          ;BRANCH IF YES
5586 032060 013737 002352 002432  MOV      TBLNUM,BITPST  ;GET TABLE #
5587 032066 004737 035024      JSR      PC,DROPIT     ;GO DROP THE UNIT
5588 032072 000402          BR      7$
5589 032074 005237 002560      6$:  INC      WRTYCT        ;ANOTHER SUCCESSFUL PASS OF RETRY
5590 032100          7$:  POP      <R5,R4,R3,R2,R1>
5591 032112 000207          RTS      PC
5592

```

```

5593
5594
5595 032114          RHTREB: PUSH    <R3,R4>
5596 032120 016237 000010 002074  MOV    SPUNIT(R2),L$LUN
5597 032126          ERRDF  32.,ERM031,ERRMS1
5598 032136 013704 002352          MOV    TBLNUM,R4          ;GET TABLE NUMBER
5599 032142 005304          DEC    R4                ;CREATE CORRECT OFFSET
5600 032144 006304          ASL    R4
5601 032146 005264 002642          INC    TRECNT(R4)        ;INCREMENT ERROR COUNT
5602 032152 023764 002614 002642 4$:  CMP    PARMAX,TRECNT(R4) ;SEE IF MAXIMUM PARITY ERRORS HAVE
5603                                     ;BEEN REACHED
5604 032160 001036          BNE    1$                ;BRANCH IF MAXIMUM NOT REACHED
5605 032162 005737 002210          TST    BRUERR           ;BRUTUS W/ERRORS
5606 032166 001030          BNE    2$                ;BRANCH IF YES
5607 032170          PRINTB #FORM08
5608 032210 010437 002432          MOV    R4,BITPST
5609 032214 004737 036406          JSR    PC,SHIFTL        ;CALCULATE CORRECT BIT POSITION
5610 032220 006237 002432          ASR    BITPST           ;CONVERT FOR EOT FLAG
5611 032224 053737 002432 002540  BIS    BITPST,EOTFLG    ;SET EOT FLAG
5612 032232 005237 002600          INC    ALLEOT
5613 032236 053737 002432 002534  BIS    BITPST,DRPFLG    ;SET DROP UNIT FLAG
5614 032244 005262 000126          INC    DRPUNT(R2)
5615 032250 005064 002642          CLR    TRECNT(R4)      ;CLEAR TRE COUNTER
5616 032254 000416          BR     3$                ;GO HOME
5617 032256 005737 002516          1$:  TST    DATRAN           ;SEE IF DATA TRANSFER COMMAND
5618 032262 001413          BEQ    3$                ;GO HOME IF NON-DATA TRANSFER
5619 032264 005737 002216          TST    RETRY            ;SEE IF RETRY ALLOWED
5620 032270 001410          BEQ    3$                ;GO HOME IF RETRY NOT ALLOWED
5621 032272 005737 002604          TST    RTYFLG          ;SEE IF RETRY IN PROGRESS
5622 032276 001005          BNE    3$                ;GO HOME IF RETRY IN PROGRESS
5623 032300 005037 002470          CLR    VISIT
5624 032304 012737 000001 002604  MOV    #1,RTYFLG        ;SET UP FOR RETRY
5625 032312          POP    <R4,R3>         ;RESTORE REGISTERS
5626 032316 000207          RTS    PC                ;GO HOME
5627
5628 032320 005737 002436          TM:   TST    CHKUTM      ;SEE IF FUNCTIONAL SEQUENCE CAUSED THIS
5629 032324 001054          BNE    5$                ;YES HE SURE DID , EXIT
5630 032326 022737 000001 002314  CMP    #1,BRTEST        ;SEE IF BRUTUS OPTION CHOSEN
5631 032334 001450          BEQ    5$                ;BRANCH IF BRUTUS TO RETURN
5632 032336 022737 000001 002206  CMP    #1,CANSEQ        ;SEE IF UNKNOWN TAPE OPTION
5633 032344 001022          BNE    1$                ;BRANCH IF NOT UNKNOWN TAPE
5634 032346 005237 002570          INC    FILENM           ;INCREMENT FILE NUMBER
5635 032352 022737 000001 002566  CMP    #1,RECORD        ;SEE IF LEOT
5636 032360 001010          BNE    4$                ;BRANCH IF NOT
5637 032362 022737 000001 002570  CMP    #1,FILENM        ;SEE IF FILE #=1
5638 032370 001404          BEQ    4$                ;BRANCH IF YES
5639 032372 005037 002566          CLR    RECORD           ;CLEAR RECORD #
5640 032376 000137 033000          JMP    LEOT              ;JUMP TO LEOT
5641 032402 005037 002566          4$:  CLR    RECORD
5642 032406 000137 032456          JMP    5$                ;BRANCH FOR UNKNOWN TAPE TO RETURN
5643 032412 005237 002524          1$:  INC    EXTDSN
5644 032416          ERRDF  7.,ERM006,ERRMS1
5645 032426 005737 002516          TST    DATRAN           ;SEE IF DATA TRANSFER COMMAND
5646 032432 001411          BEQ    5$                ;RETURN IF NOT DATA TRANSFER COMMAND
5647 032434 022737 000077 002452  CMP    #REDREV,CMMAND    ;SEE IF READ REVERSE COMMAND
5648 032442 001403          BEQ    2$                ;BRANCH IF READ REVERSE COMMAND

```

```

5649 032444 005262 000014          INC      SERRFO(R2)      ;INCREMENT STATUS ERRORS READ
5650 032450 000402                   BR        5$
5651 032452 005262 000016          2$: INC      SERRRV(R2)      ;INCREMENT STATUS ERRORS READ REVERSE
5652 032456 000207          5$: RTS      PC          ;GO HOME
5653
5654 032460 022737 000001 002314 BOT:  CMP      #1,BRTEST      ;SEE IF BRUTUS OPTION
5655 032466 001423                   BEQ      2$              ;BRANCH IF BRUTUS OPTION
5656 032470 005237 002524          INC      EXTDSN
5657 032474                   ERRDF  8.,ERM007,ERRMS1
5658 032504 005737 002210          TST      BRUERR          ;SEE IF BRUTUS W/ERRORS
5659 032510 001005                   BNE      1$              ;BRANCH IF BRUTUS W/ERRORS
5660 032512 013737 002352 002432  MOV      TBLNUM,BITPST  ;STORE TABLE NUMBER IN BIT POSITION
5661 032520 004737 035024          JSR      PC,DROPIT      ;GO DROP THE UNIT
5662 032524 005737 002516          1$: TST      DATRAN      ;SEE IF DATA TRANSFER COMMAND
5663 032530 001402                   BEQ      2$              ;BRANCH IF NOT
5664 032532 005262 000016          INC      SERRRV(R2)      ;INCREMENT STATUS ERROR READ REVERSE
5665 032536 000207          2$: RTS      PC          ;GO HOME
5666
5667 032540                   EOT:  PUSH     <R3,R4,R5>
5668 032546 013737 002352 002432  MOV      TBLNUM,BITPST  ;FIND CORRECT UNIT
5669 032554 004737 036406          JSR      PC,SHIFTL      ;TO SET UP DROP FLAG
5670 032560 006237 002432          ASR      BITPST         ;CONVERT FOR EOT FLAG
5671 032564 022737 000001 002314  CMP      #1,BRTEST      ;SEE IF BRUTUS OPTION
5672 032572 001476                   BEQ      5$
5673 032574 033737 002432 002540  BIT      BITPST,EOTFLG  ;SEE IF AT EOT
5674 032602 001403                   BEQ      6$              ;BRANCH IF NOT
5675 032604 005237 002600          INC      ALLEOT         ;SET ALL UNITS EOT FLAG
5676 032610 000467                   BR        5$              ;GET OUT
5677 032612 005737 002210          6$: TST      BRUERR      ;SEE IF BRUTUS W/ERRORS
5678 032616 001426                   BEQ      1$              ;BRANCH IF NOT
5679 032620                   PRINTB #FORM28
5680 032640                   PRINTB #FORM01,RHADDR, TM78N,TU78N
5681 032674 005737 002516          1$: TST      DATRAN      ;SEE IF DATA TRANSFER COMMAND
5682 032700 001425                   BEQ      4$              ;BRANCH IF NOT
5683 032702 005737 002522          TST      EXTERR        ;SEE IF EXTENDED ERROR 1 OR 0
5684 032706 001412                   BEQ      2$              ;BRANCH IF ZERO
5685 032710 005037 002560          CLR      WRTYCT         ;CLEAR WRITE
5686 032714 005037 002556          CLR      WRTCNT        ;RETRY
5687 032720 005037 002546          CLR      BADCNT        ;FLAGS
5688 032724 012737 000006 002604  MOV      #6,RTYFLG      ;SPACE REVERSE AND CLOSE FILE
5689 032732 000413                   BR        3$
5690 032734 012703 000032          2$: MOV      #BYWRLO,R3  ;STORE OFFSETS
5691 032740 012704 000030          MOV      #BYWRMD,R4    ;INTO
5692 032744 012705 000026          MOV      #BYWRHI,R5    ;REGISTERS
5693 032750 004737 035640          JSR      PC,DECMAL      ;UPDATE # OF BYTES WRITTEN
5694 032754 012737 000005 002604  4$: MOV      #5,RTYFLG  ;SET UP TO CLOSE FILE
5695 032762 053737 002432 002540  3$: BIS      BITPST,EOTFLG ;SET EOT BIT
5696 032770          5$: POP      <R5,R4,R3>
5697 032776 000207          RTS      PC
5698
5699 033000 022737 000001 002314 LEOT:  CMP      #1,BRTEST      ;SEE IF BRUTUS OPTION
5700 033006 001420                   BEQ      1$              ;BRANCH IF NOT
5701 033010                   ERRDF  10.,ERM009,ERRMS1
5702 033020 013737 002352 002432  MOV      TBLNUM,BITPST  ;SET UP
5703 033026 004737 036406          JSR      PC,SHIFTL      ;TO DROP UNIT
5704 033032 006237 002432          ASR      BITPST         ;CONVERT FOR EOT FLAG

```

```

5705 033036 005237 002600          INC      ALLEOT      ;
5706 033042 053737 002432 002540  BIS      BITPST,EOTFLG ;BY SETTING EOT FLAG
5707 033050 000207          1$:      RTS      PC
5708
5709
5710
5711 033052 000207          IREWND: RTS      PC      ;GO HOME
5712
5713
5714 033054 022737 000001 002314  FPT:     CMP      #1,BRTEST  ;SEE IF BRUTUS OPTION
5715 033062 001416          BEQ      FPTOUT    ;BRANCH IF BRUTUS OPTION CHOSEN
5716 033064 005237 002524          INC      EXTDSN
5717 033070          ERRDF  11.,ERM010,ERRMS1
5718 033100 005737 002210          CHEAT:  TST      BRUERR    ;SEE IF BRUTUS W/ERRORS OPTION CHOSEN
5719 033104 001005          BNE      FPTOUT    ;BRANCH TO RETURN IF BRUTUS W/ERRORS
5720 033106 013737 002352 002432  MOV      TBLNUM,BITPST ;SET UP
5721 033114 004737 035024          JSR      PC,DROPIT  ;GO DROP THE UNIT
5722 033120 000207          FPTOUT: RTS      PC
5723
5724 033122 000207          INOOP:  RTS      PC      ;RETURN
5725
5726 033124 022737 000001 002314  NOTRDY: CMP      #1,BRTEST  ;SEE IF BRUTUS OPTION
5727 033132 001406          BEQ      1$        ;BRANCH IF BRUTUS OPTION
5728 033134          ERRDF  12.,ERM011,ERRMS1
5729 033144 000137 033100          JMP      CHEAT     ;LOOP TO ABOVE CODE IN FPT
5730 033150 000207          1$:      RTS      PC      ;GO HOME
5731
5732 033152 022737 000001 002314  NOTAVL: CMP      #1,BRTEST  ;SEE IF BRUTUS OPTION
5733 033160 001410          BEQ      1$        ;BRANCH IF BRUTUS OPTION
5734 033162 005237 002524          INC      EXTDSN
5735 033166          ERRDF  13.,ERM012,ERRMS1
5736 033176 000137 033100          JMP      CHEAT     ;LOOP TO ABOVE CODE IN FPT
5737 033202 000207          1$:      RTS      PC      ;GO HOME
5738
5739 033204 022737 000001 002314  OFFLNE: CMP      #1,BRTEST  ;SEE IF BRUTUS OPTION
5740 033212 001406          BEQ      1$        ;BRANCH IF BRUTUS OPTION
5741 033214          ERRDF  14.,ERM013,ERRMS1
5742 033224 000137 033100          JMP      CHEAT     ;LOOP TO ABOVE CODE IN FPT
5743 033230 000207          1$:      RTS      PC      ;GO HOME
5744
5745 033232 022737 000001 002314  NONEX:  CMP      #1,BRTEST  ;SEE IF BRUTUS OPTION
5746 033240 001406          BEQ      1$        ;BRANCH IF BRUTUS W/ERRORS
5747 033242          ERRDF  15.,ERM014,ERRMS1
5748 033252 000137 033100          JMP      CHEAT     ;LOOP TO ABOVE CODE IN FPT
5749 033256 000207          1$:      RTS      PC      ;GO HOME
5750
5751 033260 022737 000004 002206  BLKTAP: CMP      #4,CANSEQ  ;FUNCTIONAL SEQUENCE ?
5752 033266 001003          BNE      2$        ;NO
5753 033270 005237 002440          INC      ERASED    ;YES , SET ERASED FLAG
5754 033274 000412          BR      1$        ;EXIT
5755 033276 022737 000001 002314  2$:     CMP      #1,BRTEST  ;SEE IF BRUTUS OPTION
5756 033304 001406          BEQ      1$        ;GO HOME
5757 033306          ERRDF  16.,ERM015,ERRMS1
5758 033316 000137 033100          JMP      CHEAT     ;LOOP TO ABOVE CODE IN FPT
5759 033322 000207          1$:      RTS      PC
5760

```



```

5761 033324 022737 000001 002314 ILLINT: CMP #1,BRTEST ;SEE IF BRUTUS
5762 033332 001406 BEQ 1$ ;BRANCH IF BRUTUS OPTION
5763 033334 ERRDF 5.,ERM004,ERRMS1
5764 033344 000137 033100 JMP CHEAT ;LOOP TO ABOVE CODE IN FPT
5765 033350 000207 1$: RTS PC ;GO HOME
5766
5767 033352 022737 000001 002314 ONLINE: CMP #1,BRTEST ;SEE IF BRUTUS OPTION
5768 033360 001404 BEQ 1$ ;BRANCH IF BRUTUS
5769 033362 ERRDF 17.,ERM016,ERRMS1
5770 033372 000207 1$: RTS PC ;RETURN
5771
5772 033374 S4TREC:
5773 033374 LONREC: PUSH <R3,R4,R5>
5774 033402 022737 000001 002314 CMP #1,BRTEST ;SEE IF BRUTUS
5775 033410 001523 BEQ 10$ ;YES, GO HOME
5776 033412 005037 002604 CLR RTYFLG ;CLEAR RETRY FLAG
5777 033416 022737 000001 002206 CMP #1,CANSEQ ;SEE IF UNKNOWN TAPE
5778 033424 001470 BEQ 8$ ;YES, BRANCH
5779 033426 005237 002524 INC EXTDSN ;DO EXTENDED SENSE
5780 033432 022737 000020 002526 CMP #20,ICCODE ;SEE IF LONG RECORD
5781 033440 001005 BNE 1$ ;BRANCH IF NOT
5782 033442 ERRDF 18.,ERM017,ERRMS1
5783 033452 000404 BR 2$
5784 033454 1$: ERRDF 19.,ERM018,ERRMS1
5785 033464 022737 000077 002452 2$: CMP #REDREV,CMMAND ;SEE IF READ REVERSE
5786 033472 001423 BEQ 5$ ;BRANCH IF RR
5787 033474 005737 002606 TST SENFLG ;SEE IF READ SENSE FAILED
5788 033500 001015 BNE 4$ ;BRANCH IF FAILED
5789 033502 132737 000002 003251 BITB #2,ECCSTA ;SEE IF DOUBLE TRACK READ FORWARD
5790 033510 001403 BEQ 3$ ;BRANCH IF NOT
5791 033512 005262 000052 INC DBLTRF(R2) ;INCREMENT DOUBLE TRACK ERROR FORWARD
5792 033516 000406 BR 4$ ;GO UPDATE BYTES
5793 033520 132737 000001 003251 3$: BITB #1,ECCSTA ;SEE IF SINGLE TRACK READ FORWARD
5794 033526 001402 BEQ 4$ ;
5795 033530 005262 000056 INC SNGTRF(R2) ;INCREMENT SINGLE TRACK ERROR FORWARD
5796 033534 005262 000014 4$: INC SERRFO(R2) ;INCREMENT STATUS ERROR READ FORWARD
5797 033540 000422 BR 8$ ;
5798 033542 005737 002606 5$: TST SENFLG ;SEE IF READ SENSE FAILED
5799 033546 001015 BNE 7$ ;BRANCH IF FAILED
5800 033550 132737 000002 003251 BITB #2,ECCSTA ;SEE IF DOUBLE TRACK READ REVERSE
5801 033556 001403 BEQ 6$ ;BRANCH IF NOT
5802 033560 005262 000054 INC DBLTRR(R2) ;INCREMENT DOUBLE TRACK READ REVERSE
5803 033564 000406 BR 7$ ;
5804 033566 132737 000001 003251 6$: BITB #1,ECCSTA ;SEE IF SINGLE TRACK READ REVERSE
5805 033574 001402 BEQ 7$ ;BRANCH IF NOT
5806 033576 005262 000060 INC SNGTRR(R2) ;INCREMENT SINGLE TRACK READ REVERSE
5807 033602 005262 000016 7$: INC SERRRV(R2) ;INCREMENT STATUS ERROR READ REVERSE
5808 033606 022737 000077 002452 8$: CMP #REDREV,CMMAND ;SEE IF READ REVERSE COMMAND
5809 033614 001411 BEQ 9$ ;BRANCH IF RR
5810 033616 012703 000040 MOV #BYRFLO,R3 ;STORE STATISTICS
5811 033622 012704 000036 MOV #BYRFMD,R4 ;OFFSETS IN
5812 033626 012705 000034 MOV #BYRFHI,R5 ;REGISTERS
5813 033632 004737 035640 JSR PC,DECIMAL ;STORE DECIMAL BYTES READ
5814 033636 000410 BR 10$ ;GO HOME
5815 033640 012703 000046 9$: MOV #BYRRLO,R3 ;STORE STATISTICS
5816 033644 012704 000044 MOV #BYRRMD,R4 ;OFFSETS IN

```

```

5817 033650 012705 000042      MOV      #BYRRHI,R5      ;REGISTERS
5818 033654 004737 035640      JSR      PC,DECMAL      ;STORE DECIMAL BYTES RR
5819 033660      10$:  POP      <R5,R4,R3>
5820 033666 000207      RTS      PC              ;RETURN
5821
5822
5823 033670 022737 000001 002314  IRETRY:  CMP      #1,BRTEST      ;SEE IF BRUTUS OPTION
5824 033676 001414      BEQ      5$              ;BRANCH IF YES
5825 033700 005237 002524      INC      EXTDSN
5826 033704      ERRSOFT 20.,ERM019,ERRMS1
5827 033714 005737 002216      TST      RETRY          ;SEE IF RETRIES SELECTED
5828 033720 001403      BEQ      5$              ;BRANCH IF NOT
5829 033722 012737 000002 002604      MOV      #2,RTYFLG      ;SET RETRY FLAG
5830 033730 000207      5$:  RTS      PC              ;GO HOME
5831
5832 033732 022737 000001 002314  READOP:  CMP      #1,BRTEST      ;SEE IF BRUTUS OPTION
5833 033740 001414      BEQ      5$              ;
5834 033742 005237 002524      INC      EXTDSN
5835 033746      ERRSOFT 21.,ERM020,ERRMS1
5836 033756 005737 002216      TST      RETRY          ;SEE IF RETRIES SELECTED
5837 033762 001403      BEQ      5$              ;BRANCH IF NO RETRIES SELECTED
5838 033764 012737 000003 002604      MOV      #3,RTYFLG      ;SET RETRY FLAG
5839 033772 000207      5$:  RTS      PC              ;GO HOME
5840
5841 033774 022737 000001 002314  UNRDAB:  CMP      #1,BRTEST      ;SEE IF BRUTUS OPTION
5842 034002 001417      BEQ      2$              ;BRANCH IF BRUTUS OPTION
5843 034004 005237 002524      INC      EXTDSN
5844 034010      ERRHRD  22.,ERM021,ERRMS1
5845 034020 022737 000077 002462      CMP      #REDREV,RETRCD ;SEE IF READ REVERSE COMMAND
5846 034026 001403      BEQ      1$              ;
5847 034030 005262 000022      INC      NRECRE(R2)     ;INCREMENT NON-RECOVERABLE READS
5848 034034 000402      BR      2$              ;
5849 034036 005262 000024      1$:  INC      NRECRR(R2)   ;INCREMENT NON-RECOVERABLE READ REVERSE
5850 034042 005037 002604      2$:  CLR      RTYFLG
5851 034046 000207      RTS      PC
5852
5853 034050 022737 000001 002314  ERROR:  CMP      #1,BRTEST      ;SEE IF BRUTUS OPTION
5854 034056 001500      BEQ      5$              ;
5855 034060 022737 000004 002546      CMP      #4,BADCNT      ;SEE IF HARD ERROR
5856 034066 001436      BEQ      6$              ;BRANCH IF SO
5857 034070 005237 002524      INC      EXTDSN
5858 034074      ERRSOFT 23.,ERM022,ERRMS1
5859 034104 005737 002216      TST      RETRY          ;ARE RETRIES ALLOWED
5860 034110 001021      BNE      3$              ;
5861 034112 022737 000070 002452      CMP      #70,CMMAND     ;SEE IF READ OR WRITE OPERATION
5862 034120 101012      BHI      2$              ;BRANCH IF WRITE COMMAND
5863 034122 022737 000077 002452      CMP      #REDREV,CMMAND ;SEE IF READ REVERSE COMMAND
5864 034130 001403      BEQ      1$              ;BRANCH IF READ REVERSE
5865 034132 005262 000014      INC      SERRFO(R2)     ;INCREMENT READ STATUS ERROR
5866 034136 000450      BR      5$              ;
5867 034140 005262 000016      1$:  INC      SERRRV(R2)   ;INCREMENT READ REVERSE STATUS ERROR
5868 034144 000445      BR      5$              ;
5869 034146 005262 000012      2$:  INC      SERWRT(R2)   ;INCREMENT WRITE STATUS ERROR
5870 034152 000442      BR      5$              ;
5871 034154 023727 002452 000070  3$:  CMP      CMMAND,#70     ;SEE IF READ OR WRITE
5872 034162 101036      BHI      5$              ;BRANCH IF READ

```

```

5873 034164 005037 002470      6$: CLR VISIT ;CLEAR WRITE VISIT FLAG
5874 034170 022737 000004 002546 CMP #4,BADCNT ;SEE IF RETRIED ENOUGH
5875 034176 001025 BNE 4$ ;BRANCH IF NOT DONE
5876 034200 005037 002546 CLR BADCNT ;RESET BAD COUNT
5877 034204 005262 000020 INC NRECWR(R2) ;UPDATE NON-RECOVERABLE WRITE
5878 034210 005237 002524 INC EXTDSN ;
5879 034214 ERRHRD 35.,ERM034,ERRMS1 ;
5880 034224 005037 002604 CLR RTYFLG ;RESET RETRIES
5881 034230 005737 002210 TST BRUERR ;SEE IF BRUTUS W/ERRORS
5882 034234 001011 BNE 5$ ;BRANCH IF B W/ERRORS
5883 034236 013737 002352 002432 MOV TBLNUM,BITPST ;GET UNIT #
5884 034244 004737 035024 JSR PC,DROPIT ;GO DROP THE UNIT
5885 034250 000403 BR 5$ ;
5886 034252 012737 000004 002604 4$: MOV #4,RTYFLG ;SET RETRY FLAG
5887 034260 000207 5$: RTS PC ;GO HOME
5888
5889 034262 022737 000001 002314 EOTERR: CMP #1,BRTEST ;SEE IF BRUTUS OPTION
5890 034270 001426 BEQ 3$ ;IF BRUTUS GO HOME
5891 034272 005737 002210 TST BRUERR ;SEE IF BRUTUS W/ERRORS
5892 034276 001404 BEQ 1$ ;BRANCH IF NOT
5893 034300 ERRSOFT 24.,ERM023,ERRMS1 ;
5894 034310 013737 002352 002432 1$: MOV TBLNUM,BITPST ;
5895 034316 004737 036406 JSR PC,SHIFTL ;MAKE TABLE NUMBER CORRESPOND TO BIT POSITION
5896 034322 006237 002432 ASR BITPST ;CONVERT FOR EOT FLAG
5897 034326 053737 002432 002540 BIS BITPST,EOTFLG ;DROP UNIT
5898 034334 005037 002470 CLR VISIT ;
5899 034340 012737 000006 002604 MOV #6,RTYFLG ;SET UP FOR RETRY SIX
5900 034346 000207 3$: RTS PC ;GO HOME
5901
5902 034350 022737 000001 002314 BADTAP: CMP #1,BRTEST ;SEE IF BRUTUS OPTION
5903 034356 001416 BEQ 4$ ;GO HOME
5904 034360 005237 002524 INC EXTDSN ;
5905 034364 ERRDF 25.,ERM024,ERRMS1 ;
5906 034374 005737 002210 TST BRUERR ;SEE IF BRUTUS W/ERRORS
5907 034400 001005 BNE 4$ ;BRANCH IF BRUTUS W/ERRORS
5908 034402 013737 002352 002432 MOV TBLNUM,BITPST ;
5909 034410 004737 035024 JSR PC,DROPIT ;GO DROP THE UNIT
5910 034414 000207 4$: RTS PC ;
5911
5912 034416 022737 000001 002314 TMFALT: CMP #1,BRTEST ;SEE IF BRUTUS OPTION
5913 034424 001416 BEQ TMOUT ;BRANCH IF BRUTUS OPTION
5914 034426 005237 002524 INC EXTDSN ;
5915 034432 ERRDF 26.,ERM025,ERRMS1 ;
5916 034442 005737 002210 CHEATA: TST BRUERR ;SEE IF BRUTUS W/ERRORS
5917 034446 001005 BNE TMOUT ;BRANCH IF BRUTUS W/ERRORS
5918 034450 013737 002352 002432 MOV TBLNUM,BITPST ;
5919 034456 004737 035024 JSR PC,DROPIT ;GO DROP THE UNIT
5920 034462 000207 TMOUT: RTS PC ;GO HOME
5921
5922 034464 022737 000001 002314 TUFALT: CMP #1,BRTEST ;SEE IF BRUTUS OPTION
5923 034472 001410 BEQ 1$ ;BRANCH IF BRUTUS OPTION
5924 034474 005237 002524 INC EXTDSN ;
5925 034500 ERRDF 27.,ERM026,ERRMS1 ;
5926 034510 000137 034442 JMP CHEATA ;LOOP TO ABOVE CODE IN TMFALT
5927 034514 000207 1$: RTS PC ;GO HOME
5928

```

```

5929 034516 022737 000001 002314 TUFALB: CMP #1,BRTEST ;SEE IF BRUTUS OPTION
5930 034524 001404 BEQ 1$ ;BRANCH IF BRUTUS
5931 034526 ERRDF 30.,ERM029,ERRMS2
5932 034536 000207 1$: RTS PC ;RETURN
5933
5934 034540 MBFALT:
5935 034540 TMFALB: PUSH <R1,R2,R3,R4>
5936 034550 005004 CLR R4 ;SET UP TIME OUT COUNTER
5937 034552 013701 002532 MOV MASBUS,R1 ;R1 CONTAINS RH ADDRESS
5938 034556 016137 000016 002430 MOV ATTBIT(R1),BNRYNB ;STORE ATTENTION BIT IN BINARY NUMBER
5939 034564 004737 036360 JSR PC,SHIFTR ;CONVERT BIT POSITION TO BINARY NUMBER
5940 034570 053761 002430 000010 BIS BNRYNB,UNSLCT(R1) ;SET UNIT SELECT WITH A TM NUMBER
5941 034576 013737 002430 002370 MOV BNRYNB,TM78N
5942 034604 022737 000001 002314 CMP #1,BRTEST ;SEE IF BRUTUS OPTION
5943 034612 001450 BEQ 4$ ;BRANCH TO TM READY
5944 034614 022737 000034 002526 CMP #34,ICCODE ;SEE IF MB FAULT
5945 034622 001005 BNE 8$ ;BRANCH IF NOT
5946 034624 ERRDF 31.,ERM030,ERRMS2
5947 034634 000404 BR 9$
5948 034636 8$: ERRDF 28.,ERM027,ERRMS2
5949 034646 005737 002210 9$: TST BRUERR ;SEE IF BRUTUS W/ERRORS
5950 034652 001030 BNE 4$ ;BRANCH IF BRUTUS W/ERRORS TO TM READY
5951 034654 012703 000001 MOV #1,R3 ;R3 CONTAINS TABLE NUMBER
5952 034660 013702 002330 MOV PTBLAD,R2 ;R2 CONTAINS TABLE ADDRESS
5953 034664 021201 1$: CMP (R2),R1 ;SEE IF SAME RH
5954 034666 001411 BEQ 2$ ;BRANCH IF SAME RH
5955 034670 005203 3$: INC R3 ;GET NEXT TABLE
5956 034672 020337 002344 CMP R3,TBLCNT ;SEE IF LOOK AT ALL TABLES
5957 034676 101016 BHI 4$ ;BRANCH IF LOOK AT ALL TABLES
5958 034700 063702 002346 6$: ADD TBLNTH,R2 ;GET ADDRESS OF NEXT TABLE
5959 034704 063702 002346 ADD TBLNTH,R2
5960 034710 000765 BR 1$ ;BRANCH TO COMPARE
5961 034712 023762 002430 000004 2$: CMP BNRYNB,TMNUM(R2) ;SEE IF SAME TM NUMBER
5962 034720 001367 BNE 6$ ;BRANCH IF TM'S DON'T MATCH
5963 034722 010337 002432 MOV R3,BITPST ;BIT POSITION CONTAINS TABLE NUMBER
5964 034726 004737 035024 JSR PC,DROPIT ;GO DROP THE UNIT
5965 034732 000756 BR 3$ ;GET NEXT UNIT
5966 034734 052761 040000 000052 4$: BIS #BIT14,INTRDA(R1) ;SET TM CLEAR
5967 034742 032761 100000 000052 10$: BIT #BIT15,INTRDA(R1) ;SEE IF TM READY
5968 034750 001020 BNE 5$ ;BRANCH IF TM READY IS SET
5969 034752 005204 INC R4 ;INCREMENT TIME OUT COUNTER
5970 034754 022704 177777 CMP #-1,R4 ;SEE IF REACHED TIME OUT
5971 034760 001370 BNE 10$ ;NO TIME OUT REACHED
5972 034762 022737 000001 002314 CMP #1,BRTEST ;SEE IF BRUTUS W/ERRORS
5973 034770 001410 BEQ 5$
5974 034772 005237 002542 INC TIMEOUT
5975 034776 ERRDF 29.,ERM028,ERRMS2
5976 035006 005037 002542 CLR TIMEOUT
5977 035012 5$: POP <R4,R3,R2,R1>
5978 035022 000207 RTS PC
5979
5980
5981 035024 004737 036406 DROPIT: JSR PC,SHIFTL ;CONVERT UNIT TO BIT POSITION
5982 035030 006237 002432 ASR BITPST ;CONVERT FOR EOT FLAG
5983 035034 033737 002432 002534 BIT BITPST,DRPFLG ;ALREADY DROPPED ?
5984 035042 001014 BNE 1$ ;BRANCH IF DROPPED

```

```

5985 035044 053737 002432 002534      BIS      BITPST,DRPFLG      ;ELSE - SET DROP FLAG
5986 035052 033737 002432 002540      BIT      BITPST,EOTFLG   ;WAS UNIT AT EOT ?
5987 035060 001005                BNE      1$              ;IF SO DONT INCREMENT ALLEOT
5988 035062 053737 002432 002540      BIS      BITPST,EOTFLG   ;DROP UNIT
5989 035070 005237 002600                INC      ALLEOT          ;
5990 035074                1$:    PRINTB #FORM08      ;
5991 035114 005262 000126                INC      DRPUNT(R2)     ;UPDATE STATS
5992 035120 000207                RTS      PC              ;GO KEEP UP THE GOOD WORK
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040

```

```

;*BEGINROUTINE (MOD = 3.3.5 DATA COMPARE)
;* CLEAR COUNTERS
;* STORE WRITE BUFFER ADDRESS
;* STORE READ BUFFER ADDRESS
;* STORE BYTES PER RECORD
;* DO UNTIL LOOP FLAG
;* : IF EVEN BYTE RECORD
;* : : THEN
;* : : COMPARE BUFFER
;* : : ELSE
;* : : COMPARE BYTE
;* : : ENDF
;* : IF NOT EQUAL
;* : : THEN
;* : : COMPARE LOWEST BYTE
;* : : IF NOT EQUAL
;* : : : THEN
;* : : : STORE EXPECTED
;* : : : STORE ACTUAL
;* : : : STORE BYTE # IN RECORD
;* : : : CALL DROP DR PICK ROUTINE
;* : : : INCREMENT BAD BYTE COUNT
;* : : : IF DID NOT PRINT ERROR
;* : : : : THEN
;* : : : : INCREMENT PRINT ERROR FLAG
;* : : : : IF READ FWD COMMAND
;* : : : : : THEN
;* : : : : : INCREMENT FWD DATA ERROR
;* : : : : : ELSE
;* : : : : : INCREMENT REVERSE DATA ERROR
;* : : : : ENDF
;* : : : : STORE UNIT NUMBER
;* : : : : PRINT ERROR
;* : : : : ELSE
;* : : : : IF MAXIMUM BYTES NOT PRINTED
;* : : : : : THEN
;* : : : : : PRINT BYTE BAD
;* : : : : : UPDATE BYTE BAD
;* : : : : ENDF
;* : : : ENDF
;* : : ELSE
;* : : GET NEXT BYTE IN BOTH BUFFERS
;* : : DECREMENT BYTE COUNT
;* : : IF CHECKED HIGH BYTE
;* : : : THEN
;* : : : GET NEXT BYTE IN BOTH BUFFERS

```

```

6041      : * : : : : : DECREMENT BYTE COUNT
6042      : * : : : : : ELSE
6043      : * : : : : : IF NOT LAST BYTE
6044      : * : : : : : THEN
6045      : * : : : : : CLEAR LOOP FLAG
6046      : * : : : : : ELSE
6047      : * : : : : : PRINT # OF BYTES BAD
6048      : * : : : : : SET LOOP FLAG
6049      : * : : : : : ENDIF
6050      : * : : : : : ENDIF
6051      : * : : : : : ENDIF
6052      : * : : : : : ENDIF
6053      : * : : : : : ENDDO
6054      : * : : : : : *ENDROUTINE
6055
6056      035122      COMPAR: PUSH      <R3,R4,R5>
6057      035130      005037      002554      CLR          OKAY
6058      035134      005037      002424      CLR          R10          ;CLEAR TEMPORARY REGISTERS
6059      035140      005037      002422      CLR          R9
6060      035144      013705      002354      MOV          VRMADD,R5    ;R5 CONTAINS START OF WRITE BUFFER
6061      035150      013704      002332      MOV          REDBUF,R4   ;R4 CONTAINS START OF READ BUFFER
6062      035154      013703      002220      MOV          CHAREC,R3   ;R3 CONTAINS BYTES IN RECORD
6063      035160      022703      000001      2$: CMP       #1,R3      ;SEE IF ODD BYTE BUFFER
6064      035164      001402      BEQ          11$        ;BRANCH IF YES
6065      035166      021415      CMP          (R4),(R5)   ;SEE IF BYTES COMPARE
6066      035170      001473      BEQ          4$         ;BRANCH IF YES
6067      035172      005237      002554      11$: INC       OKAY      ;SET BYTE OKAY FLAG
6068      035176      121415      CMPB        (R4),(R5)   ;SEE IF LSB OKAY
6069      035200      001467      BEQ          4$         ;BRANCH IF LSB OKAY
6070      035202      111537      002472      MOVB        (R5),EXPECT  ;STORE EXPECTED DATA
6071      035206      111437      002474      MOVB        (R4),ACTUAL  ;STORE ACTUAL DATA
6072      035212      010537      002572      MOV          R5,FIRBYT   ;FIRBYT CONTAINS ADDRESS OF FAIL BYTE
6073      035216      163737      002354      002572      SUB          VRMADD,FIRBYT ;FIRBYT NOW CONTAINS FAIL BYTE NUMBER
6074      035224      004737      035472      JSR          PC,DRPPCK   ;CALL DROP PICK ROUTINE
6075      035230      005237      002424      INC          R10        ;R10 IS NUMBER OF BYTES BAD IN RECORD
6076      035234      005737      002422      TST         R9          ;SEE IF ALREADY PRINTED ERROR
6077      035240      001023      BNE         7$         ;BRANCH IF ERROR PRINTED
6078      035242      005237      002422      INC          R9          ;SET PRINT FLAG
6079      035246      022737      000077      002452      CMP          #REDREV,CMMAND ;SEE IF READ REVERSE
6080      035254      001403      BEQ         3$         ;BRANCH IF RR
6081      035256      005262      000062      INC          DERREF(R2)  ;UPDATE READ DATA ERROR
6082      035262      000402      BR          6$         ;
6083      035264      005262      000064      3$: INC       DERRRV(R2) ;UPDATE RR DATA ERROR
6084      035270      016237      000010      002074      6$: MOV       SPUNIT(R2),L$LUN
6085      035276      ERRDF      34.,ERM033,ERRMS3
6086      035306      000424      BR          4$         ;
6087      035310      022737      000010      002422      7$: CMP       #10,R9      ;SEE IF 8 BAD BYTES FOUND
6088      035316      001420      BEQ         4$         ;BRANCH IF YES
6089      035320      PRINTB    #FORM06,FIRBYT,EXPECT,ACTUAL
6090      035354      005237      002422      INC          R9          ;UPDATES BYTE PRINTED
6091      035360      005204      4$: INC       R4          ;GET NEXT BYTE
6092      035362      005205      INC          R5          ;GET NEXT BUFFER BYTE
6093      035364      005303      DEC          R3          ;DECREMENT BYTES
6094      035366      005737      002554      TST         OKAY        ;SEE IF WORD OKAY
6095      035372      001004      BNE         12$        ;BRANCH IF NOT
6096      035374      005204      INC          R4          ;GET NEXT WORD

```

```

6097 035376 005205      INC      R5      ;GET NEXT WORD
6098 035400 005303      DEC      R3      ;DECREMENT BYTES
6099 035402 000410      BR       13$     ;
6100 035404 005703      12$: TST      R3      ;SEE IF LAST BYTE
6101 035406 001406      BEQ     13$     ;BRANCH IF YES
6102 035410 022737 000001 002554  CMP     #1,OKAY ;SEE IF CHECKED BOTH BYTES
6103 035416 001665      BEQ     11$     ;BRANCH IF NO
6104 035420 005037 002554  CLR     OKAY    ;CONTINUE AND RESET FLAG
6105 035424 005703      13$: TST      R3      ;SEE IF LAST BYTES
6106 035426 001254      BNE     2$     ;BRANCH IF NOT LAST BYTE
6107 035430 005737 002424  5$:  TST      R10     ;SEE IF ERROR OCCURRED
6108 035434 001412      BEQ     8$     ;BRANCH IF NO ERROR
6109 035436
6110 035462      PRINTB #FORM10,R10
6111 035470 000207      8$:  POP     <R5,R4,R3>
        RTS      PC      ;RETURN

```

;\*BEGINROUTINE (MODULE 3.3.5.2 = PICKS)

```

6112
6113
6114
6115
6116
6117 035472      DRPPCK: PUSH   <R4,R5>
6118 035476 013705 002472      MOV     EXPECT,R5      ;
6119 035502 042705 177400      BIC     #177400,R5     ;TAKE LOWER BYTE OF R5
6120 035506 143705 002474      BICB   ACTUAL,R5      ;R5 CONTAINS BITS DROPPED
6121 035512 001404      BEQ     1$           ;TO PICK ROUTINE
6122 035514 012704 000066      MOV     #DROPT,R4     ;R4 CONTAINS DROP OFFSETS
6123 035520 004737 035560      JSR    PC,PCKDRP     ;CALL DROP UPDATE
6124 035524 012704 000106      1$:  MOV     #PICK7,R4  ;SET UP TO FIND PICKS
6125 035530 013705 002474      MOV     ACTUAL,R5
6126 035534 042705 177400      BIC     #177400,R5     ;BIC UPPER BYTE OF R5
6127 035540 043705 002472      BIC     EXPECT,R5     ;R5 CONTAINS BITS PICKED
6128 035544 001402      BEQ     2$           ;LEAVE
6129 035546 004737 035560      JSR    PC,PCKDRP     ;CALL PICK UPDATE
6130 035552
6131 035556 000207      2$:  POP     <R5,R4>
        RTS      PC      ;

```

;\*BEGINROUTINE (MODULE 3.3.5.1 = DROPS)

```

6132
6133
6134
6135
6136
6137 035560 010437 002520      PCKDRP: MOV     R4,STORE ;STORE OFFSET
6138 035564 010537 002430      MOV     R5,BNRYNB ;
6139 035570 004737 036360      JSR    PC,SHIFTR ;FIND BIT DROPPED OR PICKED
6140 035574 013737 002430 002432  MOV     BNRYNB,BITPST ;SET UP TO CLEAR BITS DROPPED OR PICKED
6141 035602 006337 002430      ASL    BNRYNB ;MAKE OFFSET INTO WORD
6142 035606 063737 002430 002520  ADD     BNRYNB,STORE ;
6143 035614 060237 002520      ADD     R2,STORE ;ADD TABLE ADDRESS TO OFFSET
6144 035620 005277 144674      INC     @STORE ;UPDATE STATS
6145 035624 004737 036406      JSR    PC,SHIFTL ;FIND DROPPED BIT LOCATION
6146 035630 043705 002432      BIC     BITPST,R5 ;CLEAR BIT DROPPED OR PICKED
6147 035634 001351      BNE    PCKDRP ;BRANCH IF NOT DONE
6148 035636 000207      RTS    PC ;GO HOME
6149
6150
6151
6152

```

6153  
6154  
6155  
6156  
6157  
6158  
6159  
6160  
6161  
6162  
6163  
6164  
6165  
6166  
6167  
6168  
6169  
6170  
6171  
6172  
6173  
6174  
6175  
6176  
6177  
6178  
6179  
6180  
6181  
6182  
6183  
6184  
6185  
6186  
6187  
6188  
6189  
6190  
6191  
6192  
6193  
6194  
6195  
6196  
6197  
6198  
6199  
6200  
6201  
6202  
6203  
6204  
6205  
6206  
6207  
6208

035640 060205  
035642 060204  
035644 060203  
035646 066113 000006  
035652 022713 023420  
035656 101011  
035660 005214  
035662 162713 023420  
035666 022714 023420  
035672 101003  
035674 005215  
035676 162714 023420  
035702 000207  
  
035704  
035706 005003  
035710 016261 000004 000010  
035716 012761 000036 000002  
035724 005461 000002  
035730 012761 003230 000004  
035736 056261 000006 000014  
035744 012711 040000  
035750 005011  
035752 012711 000173  
035756 005037 002374  
035762  
035770 005737 002374  
035774 001023  
035776 005203  
036000  
036030 022703 023420  
036034 001355  
036036 005237 002606  
036042 000413  
036044 005037 002374  
036050 122761 000001 000012  
036056 001003  
036060 005037 002606  
036064 000402  
036066 005237 002606  
036072  
036074 000207

;; STORE BYTES TRANSFERRED IN MAXIMUM WORDS OF 9999 DECIMAL  
;

DECIMAL: ADD R2,R5 ;R5 CONTAINS HIGH BYTE ADDRESS  
ADD R2,R4 ;R4 CONTAINS MEDIUM BYTE ADDRESS  
ADD R2,R3 ;R3 CONTAINS CON BYTE ADDRESS  
ADD BYTCNT(R1),(R3) ;ADD BYTES TRANSFERRED TO LOW ORDER LOCATIONS  
CMP #23420,(R3) ;SEE IF LSB IS GREATER THAN 10000  
BHI 1\$ ;RETURN IF NOT  
INC (R4) ;PSEUDO CARRY  
SUB #23420,(R3) ;RESET LEAST SIGNIFICANT BIT  
CMP #23420,(R4) ;SEE IF MIDDLE BITS OVERFLOW  
BHI 1\$ ;RETURN  
INC (R5) ;PSEUDO CARRY  
SUB #23420,(R4) ;RESET MIDDLE BITS  
1\$: RTS PC ;

EXREAD: PUSH <R3>  
CLR R3  
MOV TMNUM(R2),UNSLCT(R1) ;LOAD TM NUMBER  
MOV #36,WRDCNT(R1) ;LOAD # OF WORDS TO TRANSFER  
NEG WRDCNT(R1) ;COMPLIMENT WORD COUNT  
MOV #COMCOD,BSADDR(R1) ;LOAD BUS ADDRESS  
BIS TUNUM(R2),CMDADR(R1) ;LOAD TU NUMBER  
MOV #40000,(R1) ;CLEAR TRE BIT IN CS1  
CLR (R1) ;CLEAR TRE BIT IN TM78 MEMORY  
MOV #173,(R1) ;ISSUE COMMAND  
CLR INTRPT ;CLEAR INTERRUPT  
SETPRI #PRI00 ;  
1\$: TST INTRPT ;SEE IF INTERRUPT RECEIVED  
BNE 2\$ ;BRANCH IF INTERRUPT RECEIVED  
INC R3 ;  
DELAY 1 ;  
CMP #10000.,R3 ;SEE IF TIME OUT REACHED  
BNE 1\$ ;GO WAIT FOR INTERRUPT  
INC SENFLG ;EXTENDED SENSE UNAVAILABLE  
BR 3\$ ;  
2\$: CLR INTRPT ;  
CMPB #1,INTCDE(R1) ;SEE IF DONE INTERRUPT  
BNE 4\$ ;BRANCH IF NOT DONE  
CLR SENFLG ;EXTENDED SENSE AVAILABLE  
BR 3\$ ;  
4\$: INC SENFLG ;EXTENDED SENSE UNAVAILABLE  
3\$: POP <R3> ;  
RTS PC ;GO HOME

\*\*\*\*\*



6209  
6210  
6211  
6212  
6213  
6214  
6215  
6216  
6217  
6218  
6219  
6220  
6221  
6222  
6223  
6224  
6225  
6226  
6227  
6228  
6229  
6230  
6231  
6232  
6233  
6234  
6235  
6236  
6237  
6238  
6239  
6240  
6241  
6242  
6243  
6244  
6245  
6246  
6247  
6248  
6249  
6250  
6251  
6252  
6253  
6254  
6255  
6256  
6257  
6258  
6259  
6260  
6261  
6262  
6263  
6264

:  
: SUBROUTINES USED THROUGHOUT THE PROGRAM  
:  
:\*\*\*\*\*

:  
: FINDS SPECIFIED LOCATION IN SPECIFIED UNIT TABLE  
:

TBLDIS: PUSH <R1,R2> ;STORE R1  
CLR R1 ;SET UP DUMMY  
MOV TBLNUM,R2 ;TEMPORARY STORAGE  
1\$: DEC R2 ;FINDS START OF TABLE  
BEQ 2\$ ;BRANCH IF ONLY ONE TABLE  
ADD TBLNTH,R1 ;  
ADD TBLNTH,R1 ;BEGINNING OF NEXT TABLE  
BR 1\$ ;ADD ANOTHER TABLE  
  
2\$: TST DENFLG ;WHICH DENSITY TABLE  
BEQ 3\$ ;BR , FOR PE TABLE  
ADD TBLNTH,R1 ;SET FOR GCR TABLE  
3\$: ADD STRTBL,R1 ;START OF TABLES IN MEMORY  
ADD R1,DISPLC ;FIND SPECIFIED DATA  
POP <R2,R1> ;RESTORE REGISTER  
RTS PC ;GO HOME

:  
: INTERRUPT SUBROUTINES  
:

INSERV: MOV RHADD1,MASBUS ;PUTS CORRECT VECTOR ADDRESS FOR  
BR 1\$ ;INTERRUPT GENERATED INTO COMMON  
MOV RHADD2,MASBUS ;LOCATION CALL MASBUS.  
BR 1\$  
MOV RHADD3,MASBUS  
BR 1\$  
MOV RHADD4,MASBUS  
BR 1\$  
MOV RHADD5,MASBUS  
BR 1\$  
MOV RHADD6,MASBUS  
BR 1\$  
MOV RHADD7,MASBUS  
BR 1\$  
MOV RHADD8,MASBUS  
1\$: INC INTRPT ;SET INTERRUPT FLAG  
RTI  
  
INTCLK: BIC #200,CSRCLK ;CLEAR MONITOR BIT IN L CLOCK CSR  
INC LCLOCK ;INCREMENT TICK COUNTER  
CMP #74,HERTZ ;SEE WHAT HERTZ CLOCK RUNS AT

```

6265 036302 001405          BEQ      1$          ;BRANCH IF 60 HERTZ
6266 036304 022737 005670 002362  CMP      #5670,LCLOCK ;SEE IF 50 HERTZ MINUTE UP
6267 036312 001405          BEQ      2$          ;BRANCH IF YES
6268 036314 000420          BR       3$          ;BRANCH IF NO
6269 036316 022737 007020 002362 1$:  CMP      #7020,LCLOCK ;SEE IF MINUTE UP
6270 036324 001014          BNE     3$          ;BRANCH IF NOT
6271 036326 005237 002324          INC     MINUTE       ;INCREMENT MINUTE
6272 036332 005037 002362          CLR     LCLOCK       ;CLEAR TICK COUNTER
6273 036336 022737 000070 002324  CMP      #70,MINUTE  ;SEE IF HOUR UP
6274 036344 001004          BNE     3$          ;BRANCH IF NO
6275 036346 005237 002322          INC     HOURS        ;INCREMENT HOURS
6276 036352 005037 002324          CLR     MINUTE       ;CLEAR MINUTE COUNTER
6277 036356 000002          3$:  RTI
6278
6279
6280
6281
6282
6283 036360          SHIFTR: PUSH    <R1>          ;STORE R1
6284 036362 012701 177777          MOV     #-1,R1
6285 036366 005201          1$:  INC     R1          ;SET UP COUNTER FOR BINARY NUMBER
6286 036370 006237 002430          ASR    BNRYNB        ;SHIFT BIT POSITION
6287 036374 103374          BCC    1$          ;BRANCH UNTIL SET BIT POSITION IS IN CARRY
6288 036376 010137 002430          MOV     R1,BNRYNB   ;STORE BINARY NUMBER INTO BNRYNB
6289 036402          POP     <R1>        ;RESTORE R1
6290 036404 000207          RTS     PC          ;GO HOME
6291
6292
6293
6294
6295
6296
6297
6298 036406          SHIFTL: PUSH    <R1>          ;STORE R1
6299 036410 012701 000001          MOV     #1,R1       ;SET UP BIT TO SHIFT
6300 036414 005337 002432          1$:  DEC     BITPST      ;SEE IF BIT SHIFTED ENOUGH
6301 036420 002402          BLT    2$          ;BRANCH WHEN DONE
6302 036422 006301          ASL    R1          ;SHIFT
6303 036424 000773          BR     1$          ;GO SEE IF SHIFT SOME MORE
6304 036426 010137 002432          2$:  MOV     R1,BITPST   ;STORE BIT POSITION INTO BITPST
6305 036432          POP     <R1>        ;RESTORE R1
6306 036434 000207          RTS     PC          ;GO HOME
6307
6308
6309
6310 036436 013737 002632 002636  RANSET: MOV     RANB,RANB2 ;RESTORE RANDOM NUMBER
6311 036444 013737 002634 002640  MOV     RANS,RANS2
6312 036452 000207          RTS     PC
6313
6314
6320 036454          ENDMOD
6321

```

6334  
 6335  
 6372  
 6373 036454  
 6374  
 6375  
 6376  
 6377  
 6378  
 6379  
 6380 036454  
 6381  
 6387  
 6388  
 6389  
 6390  
 6391  
 6392  
 6393  
 6394  
 6395  
 6396  
 6397  
 6398  
 6399  
 6400  
 6401  
 6402  
 6403  
 6404  
 6405  
 6406  
 6407  
 6408  
 6409  
 6410  
 6411 036454  
 6412 036466 013737 002330 002336  
 6413 036474 013703 002352  
 6414 036500 012701 000001  
 6415 036504 005737 002464  
 6416 036510 001014  
 6417 036512 013701 002344  
 6418 036516 012737 000001 002352  
 6419 036524 005037 002444  
 6420 036530 006037 002622 4\$:  
 6421 036534 103402  
 6422 036536 000137 037560  
 6423 036542 012737 000000 002316 1\$:  
 6424 036550 004737 036076  
 6425 036554 013702 002316  
 6426 036560  
 6427 036604  
 6428 036640 022737 000001 002204  
 6429 036646 001430  
 6430 036650 022737 000001 002206

```

.TITLE MISCELLANEOUS SECTIONS
.SBTTL REPORT CODING SECTION

      BGNMOD

:++
: THE REPORT CODING SECTION CONTAINS THE
: 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
:--

      BGNRPT

:*
:*MODULE 3.5 (OUTPUT STATISTICS)
:*
:*BEGINROUTINE (MOD = 3.5 OUTPUT STATISTICS)
:* IF CALLED BY USER COMMAND
:* : THEN-MOVE MAXIMUM # OF TABLES INTO COUNT
:* : ELSE-MOVE ONE INTO COUNT
:* ENDIF
:* DO FOR 1 TO COUNT
:* : IF UNIT DROPPED
:* : : THEN
:* : : GET STATISTIC TABLE FOR UNIT GIVEN
:* : : OUTPUT STATISTICS
:* : ENDIF
:* ENDDO
:*ENDROUTINE

      PUSH <R1,R2,R3,DENFLG>
      MOV PTBLAD,STRIBL ;SET UP DISPLACEMENT SUBROUTINE
      MOV TBLNUM,R3
      MOV #1,R1 ;COUNTER FOR # OF UNITS TO BE OUTPUT
      TST STAFLG ;SEE IF STATS CALLED FROM PROGRAM
      BNE 1$ ;BRANCH IF YES
      MOV TBLCNT,R1 ;RESET COUNTER IF END OF PASS
      MOV #1,TBLNUM
      CLR DENFLG ;START WITH PE STATISTICS
      ROR RPTFLG ;SEE IF UNIT DROPPED
      BCS 1$ ;BRANCH IF UNIT STAT HAVE BEEN PRINTED
      JMP 3$
      MOV #0,DISPLC ;MOVE OFFSET OF TABLE TO TEMPORARY STORAGE
      JSR PC,TBLDIS ;GET BEGINNING TABLE ADDRESS
      MOV DISPLC,R2 ;STORE ADDRESS OF TABLE IN REG
      PRINTS #FMT1,SPUNIT(R2) ;PRINT OUT HEADER
      PRINTS #FMT2,RHADD(R2),TMNUM(R2),TUNUM(R2)
      CMP #1,BRUTUS ;BRUTUS OPTION
      BEQ 6$ ;YES DO NOT PRINT BOTH DENSITIES
      CMP #1,CANSEQ ;UNKNOWN TAPE
    
```

6431	036656	001424				BEQ	6\$		:YES WE DON'T KNOW WHAT DENSITY
6432	036660	005737	002444			TST	DENFLG		:PE OR GCR STATISTICS ?
6433	036664	001011				BNE	5\$		:BR FOR GCR
6434	036666					PRINTS	#FMTPE		:PRINT PE HEADER
6435	036706	000410				BR	6\$		:PRINT GCR HEADER MESSAGE
6436	036710				5\$:	PRINTS	#FMTGCR		
6437	036730				6\$:	PRINTS	#FMT3		
6438	036750					PRINTS	#FMT4,SERWRT(R2),SERRFO(R2),SERRRV(R2)		
6439	037004					PRINTS	#FMT5,NRECWR(R2),NRECRE(R2),NRECRV(R2)		
6440	037040					PRINTS	#FMT6,BYWRHI(R2),BYWRMD(R2),BYWRLO(R2),BYRFHI(R2),BYRFMD(R2),BYRFLO(R2)		
6441	037110					PRINTS	#FMT7,BYRRHI(R2),BYRRMD(R2),BYRRLO(R2)		
6442	037144					PRINTS	#FMT8,MEDIAE(R2)		
6443	037170					PRINTS	#FMT9,DBLTRF(R2),DBLTRR(R2)		
6444	037220					PRINTS	#FMT10,SNGTRF(R2),SNGTRR(R2)		
6445	037250					PRINTS	#FMT11,DERREF(R2),DERRRV(R2)		
6446	037300					PRINTS	#FMT17,OTHWRT(R2),OTHRFO(R2),OTHRRV(R2)		
6447	037334					PRINTS	#FMT12		
6448	037354					PRINTS	#FMT13		
6449	037374					PRINTS	#FMT14,DROP7(R2),DROP6(R2),DROP5(R2),DROP4(R2),DROP3(R2),DROP2(R2),DROP1		
6450	037454					PRINTS	#FMT15,PICK7(R2),PICK6(R2),PICK5(R2),PICK4(R2),PICK3(R2),PICK2(R2),PICK1		
6451	037534					PRINTS	#FMT16,DRPUNT(R2)		
6452	037560	005737	002464		3\$:	TST	STAFLG		:SEE IF ONLY PRINT ONE DENSITY TABLE
6453	037564	001017				BNE	10\$		:BR OUT ONLY ONE TABLE
6454	037566	022737	000001	002204		CMP	#1,BRUTUS		:BRUTUS OPTION ?
6455	037574	001413				BEQ	10\$		:YES , ONLY ONE DENSITY
6456	037576	022737	000001	002206		CMP	#1,CANSEQ		:UNKNOWN TAPE ?
6457	037604	001407				BEQ	10\$		:YES , ONLY ONE DENSITY
6458	037606	005737	002444			TST	DENFLG		:WHICH DENSITY TABLE WAS JUST PRINTED
6459	037612	001004				BNE	10\$		:ALL DONE
6460	037614	005237	002444			INC	DENFLG		:SET FOR GCR NOW
6461	037620	000137	036542			JMP	1\$		:BACK FOR MORE
6462	037624	005237	002352		10\$:	INC	TBLNUM		:NEXT UNIT
6463	037630	005037	002444			CLR	DENFLG		:SET FOR PE
6464	037634	005301				DEC	R1		:CORRECT NUMBER OF UNITS PRINTED
6465	037636	001402				BEQ	2\$		:BRANCH IF NO
6466	037640	000137	036530			JMP	4\$		:RESTORE TBLNUM
6467	037644	010337	002352		2\$:	MOV	R3,TBLNUM		:RESET GROUP REPORT FLAG
6468	037650	012737	000377	002622		MOV	#377,RPTFLG		
6469	037656					POP	<DENFLG,R3,R2,R1>		
6470									
6471	037670					EXIT	RPT		
6472									
6473						.NLIST	BEX		
6474	037674	047045	040445	051440	FMT1:	.ASCIZ	/#N#A STATISTICAL REPORT FOR UNIT #D2#N/		
6475	037744	047045	040445	044122	FMT2:	.ASCIZ	/#N#ARH ADDRESS %06#A TM78# %D1#A TU78# %D1#N/		
6476	040026	047045	040445	040524	FMTPE:	.ASCIZ	/#N#ATAPE DENSITY -- 1600 BPI (PE)#N/		
6477	040072	047045	040445	040524	FMTGCR:	.ASCIZ	/#N#ATAPE DENSITY -- 6250 BPI (GCR)#N/		
6478	040137	045	022516	020101	FMT3:	.ASCIZ	/#N#A WRITE	READ	READ REVERS
6479	040250	047045	047045	040445	FMT4:	.ASCIZ	/#N#N#ASTATUS ERRORS	%D5#A	%D5#A
6480	040361	045	022516	047101	FMT5:	.ASCIZ	/#N#ANON-RECOVERABLE	%D5#A	%D5#A
6481	040470	047045	040445	054502	FMT6:	.ASCIZ	/#N#ABYTES TRANSFERRED %D4#Z4#Z4#A	%D4#Z4#Z4#A	%D4#Z4#Z4#A
6482	040553	045	020101	020040	FMT7:	.ASCIZ	/#N#A %D4#Z4#Z4#A		
6483	040575	045	022516	046501	FMT8:	.ASCIZ	/#N#AMEDIA ERRORS	%D5#A	--
6484	040701	045	022516	042101	FMT9:	.ASCIZ	/#N#ADouble TRACK CORRECTIONS --	%D5#A	%D5#A
6485	041010	047045	040445	044523	FMT10:	.ASCIZ	/#N#ASINGLE TRACK CORRECTIONS --	%D5#A	%D5#A
6486	041117	045	022516	042101	FMT11:	.ASCIZ	/#N#ADATA COMPARE ERRORS --	%D5#A	%D5#A

					FMT12:	.ASCIZ	/XN%XA	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	B
6487	041226	047045	047045	040445	FMT12:	.ASCIZ	/XN%XA								
6488	041341	045	022516	020101	FMT13:	.ASCIZ	/XN%XA	(TRACK7)	(TRACK6)	(TRACK5)	(TRACK3)	(TRACK9)	(TRACK1)	(TRACK8)	(TRAC
6489	041454	047045	047045	040445	FMT14:	.ASCIZ	/XN%ZADROPS	%D5%	%D5%	%D5%	%D5%	%D5%	%D5%	%D5%	%D
6490	041565	045	022516	022516	FMT15:	.ASCIZ	/XN%ZAPICKS	%D5%	%D5%	%D5%	%D5%	%D5%	%D5%	%D5%	%D
6491	041676	047045	047045	040445	FMT16:	.ASCIZ	/XN%ZATIMES DROPPED	%D5%	%D5%						
6492	041732	047045	040445	052117	FMT17:	.ASCIZ	/XN%AOTHER			%D5%		%D5%		%D5%	

```

6493
6494
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6528
6529
6530
6531
6532
6533
6534
6535
6536
6537
6538
6539
6540
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554
6555
6556
6557
6558
6559
6560
6561
6562
    
```

042042

042042

042044

```

        .LIST BEX
        .EVEN
        ENDRPT
        .SBTTL INITIALIZE SECTION
        ;++
        ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
        ; AT THE BEGINNING OF EACH PASS.
        ;--
    
```

BGNINIT

```

        ;*
        ;*BEGINROUTINE (MOD 2.0 INIT)
        ;* CLEAR START FLAG
        ;* CLEAR START RESTART FLAG
        ;* CLEAR ERROR FLAG
        ;* IF NO RESTART COMMAND
        ;* : THEN
        ;* : IF START COMMAND
        ;* : : THEN
        ;* : : INCREMENT START FLAG
        ;* : : ELSE
        ;* : : GET FAKE P-TABLES FOR END OF PASS PURPOSES
        ;* : : IF NOT CONTINUE
        ;* : : : THEN
        ;* : : : IF POWER FAIL
        ;* : : : : THEN
        ;* : : : : PRINT ERROR
        ;* : : : : HALT
        ;* : : : : ENDF
        ;* : : : ELSE
        ;* : : : GO TO END INITIALIZE
        ;* : : ENDF
        ;* : ENDF
        ;* ENDF
        ;* INCREMENT START/RESTART FLAG
        ;* CLEAR HOURS COUNTER
        ;* CLEAR MINUTE COUNTER
        ;* SET INITIALIZE FLAG
        ;* RESET PATTERN GENERATION FLAG
        ;* IF CLOCK
        ;* : THEN
        ;* : SET UP INFORMATION
    
```

```

6563 : * : ELSE
6564 : * : PRINT NO CLOCK AVAILABLE
6565 : * : ENDF
6566 : * : SIZE MEMORY
6567 : * : IF START/RESTART SET
6568 : * : THEN
6569 : * : INITIALIZE RANDOM PATTERN
6570 : * : BUILD DEVICE TABLES
6571 : * : CALL DEVICE CHECK
6572 : * : CALL BRUTUS QUESTIONS
6573 : * : CALL CLEAR STATISTICS
6574 : * : ENDF
6575 : * : SET UP BRUTUS/BRUTUS WITH ERRORS FLAG
6576 : * : ENDRoutine
6577
6578
6579 042044 005037 002276 CLR START1
6580 042050 005037 002334 CLR STRST ;INITIALIZE START OR RESTART FLAG
6581 042054 005037 002320 CLR ERRFLG ;INITIALIZE ERROR FLAG
6582 042060 READEF #EF.RESTART ;SUPERVISOR CMD TO SEE IF A RESTART CMD WAS USED
6583 042066 BNCOMPLETE 10$ ;BRANCH IF NOT FOUND
6584 042070 000442 BR 6$
6585 042072 10$: READEF #EF.START ;SUPERVISOR CMD TO SEE IF A START CMD WAS USED
6586 042100 BNCOMPLETE 14$ ;IF NO START THEN SKIP A FEW LINES
6587 042102 005237 002276 INC START1 ;INCREMENT START FLAG
6588 042106 000433 BR 6$
6589 042110 005005 14$: CLR R5
6590 042112 12$: GPHARD R5,R4 ;
6591 042120 005205 INC R5 ;DUMMY GPHARD'S SO PASS COUNT
6592 042122 023705 002012 CMP L$UNIT,R5 ;IS CORRECT
6593 042126 001371 BNE 12$
6594 042130 READEF #EF.CONTINUE ;SEE IF CONTINUE
6595 042136 BCOMPLETE 11$ ;BRANCH IF YES
6596 042140 READEF #EF.PWR ;SEE IF POWER FAILURE
6597 042146 BNCOMPLETE 9$ ;BRANCH IF NOT
6598 042150 PRINTF #FORM26 ;
6599 042170 000000 HALT ;STOP
6600 042172 000137 042712 9$: JMP 5$
6601 042176 6$: BRESET
6602 042200 DELAY 250 ;LET THINGS SETTLE (SPECIALLY TM78!!)
6603 042230 005237 002334 INC STRST ;SET START RESTART FLAG
6604 042234 005037 002322 CLR HOURS ;RESET TIME
6605 042240 005037 002444 CLR DENFLG ;USE PE FIRST
6606 042244 005037 002324 CLR MINUTE ;RESET MINUTES
6607 042250 012737 000001 002326 MOV #1,INTLZE ;SET THE INITIALIZE FLAG
6608 042256 012737 177777 002426 MOV #-1,GENFLG ;RESET GENERATION FLAG
6609 042264 CLOCK L,R1 ;SEE IF L CLOCK
6610 042274 BNCOMPLETE 1$ ;BRANCH IF NOT
6611 042276 011137 002360 MOV (R1),CSRCLK ;STORE CLOCK ADDRESS
6612 042302 016137 000002 002366 MOV 2(R1),PRIRTY ;CLOCK PRIORITY
6613 042310 016137 000006 002364 MOV 6(R1),HERTZ ;POWER IN HERTZ
6614 042316 006337 002366 ASL PRIRTY ;
6615 042322 006337 002366 ASL PRIRTY ;
6616 042326 006337 002366 ASL PRIRTY ;SHIFT FOR CORRECT PRIORITY
6617 042332 006337 002366 ASL PRIRTY ;
6618 042336 006337 002366 ASL PRIRTY ;
  
```

```

6619 042342          SETVEC 4(R1),#INTCLK,PRIRTY ;
6620 042370 162737 000040 002366  SUB   #40,PRIRTY ;GET ONE PRIORITY LESS THAN CLOCK
6621 042376          SETPRI PRIRTY ;SET PRIORITY
6622 042404 000415          BR    8$ ;
6623 042406          PRINTF #FORM25 ;PRINT NO CLOCK
6624 042426 005037 002360 1$: CLR   CSRCLK ;CLEAR CLOCK ADDRESS
6625 042432 012737 000340 002366  MOV   #340,PRIRTY ;SET PRIORITY TO SEVEN
6626 042440          MEMORY VRMADD ;GET MEMORY
6627 042446 017737 137702 002312 8$: MOV   @VRMADD,BUFSIZ ;STORE SIZE OF MEMORY IN BUFFER SIZE
6628 042454 023737 002312 002262  CMP   BUFSIZ,MINSIZ ;SEE IF 2K AVAILABLE
6629 042462 101011          BHI   11$ ;BRANCH IF YES
6630 042464          PRINTF #EMSG1 ;
6631 042504          DOCLN ;SUPERVISOR ABORT CMD
6632 042506 005737 002334 11$: TST   STRST ;
6633 042512 001477          BEQ   5$ ;
6634 042514 004737 036436  JSR   PC,RANSET ;CLEAR RANDOM PATTERN
6635 042520 004737 015260  JSR   PC,BLDTBL ;BUILD UNIT TABLES (STATS)
6636 042524 005737 002320  TST   ERRFLG ;
6637 042530 001411          BEQ   7$ ;
6638 042532          PRINTF #EMSG6 ;
6639 042552          DOCLN ;
6640 042554 013737 002330 002336 7$: MOV   PTBLAD,STRTBL ;SET UP DISPLACEMENT SUBROUTINE FOR PTABLES
6641 042562 004737 015532  JSR   PC,DEVCHK ;SEE IF TEST DEVICES REALLY EXIST
6642 042566 005737 002320  TST   ERRFLG ;CHECK IF DEVICES WERE NOT THERE
6643 042572 001403          BEQ   2$ ;SKIP IF NO ERROR
6644 042574 004737 017510  JSR   PC,ERROUT ;PRINT ERROR DEVICE NON EXISTENT
6645 042600          DOCLN ;SUPERVISOR ABORT CMD
6646
6647 042602 005737 002204 2$: TST   BRUTUS ;SEE IF BRUTUS OPTION CHOSEN
6648 042606 001004          BNE   13$ ;IF NOT BRANCH
6649 042610 005037 002210  CLR   BRUERR ;
6650 042614 000137 042706  JMP   4$ ;
6651 042620 022737 000001 002344 13$: CMP   #1,TBLCNT ;SEE IF ONLY ONE TABLE
6652 042626 001411          BEQ   3$ ;BRANCH IF ONE TABLE
6653 042630          PRINTF #EMSG3 ;
6654 042650          DOCLN ;ABORT TEST
6655
6656 042652 004737 016526 3$: JSR   PC,BRUTQ ;ASK BRUTUS QUESTIONS
6657 042656 005737 002320  TST   ERRFLG ;SEE IF MANUAL INHIBIT FLAG WAS SET
6658 042662 001411          BEQ   4$ ;BRANCH IF NOT
6659 042664          PRINTF #EMSG4 ;
6660 042704          DOCLN ;ABORT TEST
6661
6662 042706 004737 017376 4$: JSR   PC,CLRSTA ;CLEAR STATISTICS
6663
6664 042712 013737 002204 002314 5$: MOV   BRUTUS,BRTEST ;COMBINED
6665 042720 063737 002210 002314  ADD   BRUERR,BRTEST ;THE TWO PAGES
6666 042726 005737 002360  TST   CSRCLK ;DO WE HAVE A CLOCK ?
6667 042732 001403          BEQ   20$ ;NO , EXIT INIT
6668 042734 012777 000100 137416  MOV   #100,@CSRCLK ;ENABLE INTERRUPTS
6669
6670 042742          20$: EXIT  INIT
6671
6672
6673          .EVEN
6674
    
```

```

6675 042746          ENDINIT
6676
6677 042750          BGNPROT
6678 042750 000000    .WORD 0          ;CSR OFFSET
6679 042752 000004    .WORD 4          ;TM #
6680 042754 000006    .WORD 6          ;TU #
6681 042756          ENDPROT
6682
6683 042756          BGNAUTO
6684
6685 042756          ENDAUTO
6686
6698
6699
6700          .SBTTL  CLEANUP CODING SECTION
6701
6702          :++
6703          : THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
6704          : AT THE END OF EACH PASS.
6705          :--
6706
6707 042760          BGNCLN
6708
6715 042760 005737 002360    TST  CSRCLK          ;DO WE HAVE A CLOCK ?
6716 042764 001403          BEQ  1$              ;NO , DO NOT ACCESS CSRCLK
6717 042766 012777 000000 137364    MOV  #0,@CSRCLK     ;SHUT OFF CLOCK
6718 042774 005037 002464    CLR  STAFLG         ;RESET STATISTIC FLAG
6719 043000 005037 002444    CLR  DENFLG        ;SET PE STAT MODE
6720 043004 005037 002510    CLR  DENSWT        ;SET PE MODE
6721 043010 012737 000001 002326    MOV  #1,INTLZE     ;SET INITIALIZE FLAG
6722 043016 012737 000377 002622    MOV  #377,RPTFLG  ;SET STATISTIC REPORT FLAG
6723 043024          EXIT  CLN
6724
6736
6737          .EVEN
6738
6739 043030          ENDCLN
6740          .SBTTL  DROP UNIT SECTION
6741
6742          :++
6743          : THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
6744          : TO NO LONGER BE TESTED.
6745          :--
6746
6747 043032          BGNDU
6748
6754
6755 043032          EXIT  DU
6756
6768
6769          .EVEN
6770
6771 043036          ENDDU
6772          .SBTTL  ADD UNIT SECTION
6773
6774          :++
  
```



6775  
6776  
6777  
6778  
6779  
6780 043040  
6781  
6787  
6788 043040  
6789  
6801  
6802  
6803  
6804 043044  
6805  
6806 043046  
6807

: THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE  
: TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF  
: 'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.  
:--

BGNAU

EXIT AU

.EVEN

ENDAU

ENDMOD

```

6810 .TITLE HARDWARE TESTS
6821
6822 .SBTTL TEST 1:
6867
6868 043046 BGNMOD
6869
6876
6882
6883 ; SUBROUTINE ONLY USED IN THE SCHEDULER
6884
6885 ; ASYERR -- CHECK FOR ASYNCHRONOUS ERRORS
6886 ; 1.TM FAULT
6887 ; 2.TU FAULT
6888 ; 3.TU ONLINE
6889
6890 043046 ASYERR: PUSH <R4> ;SAVE R4
6891 043050 012737 000000 002316 MOV #RHADD,DISPLC ;GET TABLE ADDRESS OF RH11/70 CS1 ADDRESS
6892 043056 004737 036076 JSR PC,TBLDIS ;FOR UNIT UNDER TEST
6893 043062 013704 002316 MOV DISPLC,R4 ;SAVE TABLE ADDRESS
6894 043066 016437 000004 002432 MOV TMNUM(R4),BITPST ;GET TM #
6895 043074 004737 036406 JSR PC,SHIFTL ;MAKE BIT POSITION
6896 043100 033764 002432 000016 BIT BITPST,ATTBIT(R4) ;ANY ASYNC ERRORS ?
6897 043106 001445 BEQ 20$ ;NO , CONTINUE NORMALLY
6898 043110 016437 000010 002074 MOV SPUNIT(R4),L$LUN ;STORE SUPERVISOR UNIT NUMBER
6899 043116 016437 000004 002370 MOV TMNUM(R4),TM78N ;STORE TM78 NUMBER
6900 043124 016437 000006 002372 MOV TUNUM(R4),TU78N ;STORE TU78 NUMBER
6901 043132 011404 MOV (R4),R4 ;GET RH11/70 ADDRESS
6902 043134 010437 002356 MOV R4,RHADDR ;STORE RH ADDRESS
6903 043140 012764 040000 000000 MOV #40000,RHADD(R4) ;CLEAR 'TRE'
6904 043146 005064 000000 CLR RHADD(R4) ;CLEAR TM78'S 'TRE'
6905 043152 016404 000036 MOV NDTICD(R4),R4 ;GET INTERRUPT CODE
6906 043156 010437 002522 MOV R4,EXTERR ;STORE INT. CODE AND FAIL CODE
6907 043162 000337 002522 SWAB EXTERR ;
6908 043166 042737 177403 002522 BIC #177403,EXTERR ;MAKE ONLY FAILURE CODE
6909 043174 006237 002522 ASR EXTERR ;
6910 043200 006237 002522 ASR EXTERR ;
6911 043204 042704 177700 BIC #177700,R4 ;MASK DOWN TO INTERRUPT CODE ONLY
6912 043210 010437 002526 MOV R4,ICCODE ;STORE INTERRUPT CODE
6913 043214 006304 ASL R4 ;MAKE WORD OFFSET
6914 043216 004774 002662 JSR PC,@INTTBL(R4) ;GO TO INTERRUPT HANDLER
6915 043222 20$: POP <R4>
6916 043224 RETURN
6917
6918 043226 BGNST
6919
6925
6926
6927
6928
6929
6930
6931 ;*
6932 ;*MODULE 3.0 SCHEDULER
6933 ;*
6934 ;*BEGINROUTINE (MOD=3.0 SCHEDULER /SCHDLR/)
6935 ;* IF START OR RESTART
  
```

```
6936 : * : THEN
6937 : * : CLEAR SCHEDULER FLAGS
6938 : * : ENDF
6939 : * : DO WHILE ALL END OF TAPE FALSE
6940 : * : CALL 3.1 (SEQUENCE DEFINITION)
6941 : * : MOVE TABLE ADDRESS TO STRTBL
6942 : * : DO TABLENUM=1, TABLE COUNTER
6943 : * : : CALL 3.2 (ISSUE COMMAND)
6944 : * : : IF COMMAND NOT = TO REWIND
6945 : * : : : THEN
6946 : * : : : CALL 3.3 (COMMAND TERMINATION)
6947 : * : : : ELSE
6948 : * : : : SET REWIND FLAG
6949 : * : : : INCREMENT REWIND COUNTER
6950 : * : : : IF REWIND COUNTER = TABLE_COUNT
6951 : * : : : : THEN
6952 : * : : : : CALL 3.3 (COMMAND TERMINATION)
6953 : * : : : : RESET REWIND COUNTER
6954 : * : : : ENDF
6955 : * : : ENDF
6956 : * : : IF DROP_UNIT SET
6957 : * : : : THEN
6958 : * : : : : CALL 3.4 (DROP UNIT)
6959 : * : : : : CALL 3.5 (OUTPUT STATISTICS)
6960 : * : : : ENDF
6961 : * : : : IF REWIND FLAG IS SET
6962 : * : : : : THEN
6963 : * : : : : : CALL 3.1 (SEQUENCE DEFINITION)
6964 : * : : : : : SUBTRACT 1 FROM TABLENUM
6965 : * : : : : ELSE-
6966 : * : : : : : IF REPETITION NOT EQUAL TO ZERO
6967 : * : : : : : : THEN-DEC REPETITION
6968 : * : : : : : ENDF
6969 : * : : : ENDF
6970 : * : : : IF END OF TAPE SET
6971 : * : : : : THEN
6972 : * : : : : : INCREMENT COUNTER
6973 : * : : : : : IF COUNTER=TABLE COUNTER
6974 : * : : : : : : THEN
6975 : * : : : : : : SET ALL_END_OF_TAPE FLAG
6976 : * : : : : : ENDF
6977 : * : : : ENDF
6978 : * : : ENDDO
6979 : * : ENDDO
6980 : * : IF AUTO REWIND
6981 : * : : THEN
6982 : * : : : IF NO STATS AT EOT
6983 : * : : : : THEN
6984 : * : : : : : IF STATS AT EOP
6985 : * : : : : : : THEN
6986 : * : : : : : : : CALL 3.5 (OUTPUT STATISTICS)
6987 : * : : : : : : ENDF
6988 : * : : : ENDF
6989 : * : : ELSE
6990 : * : : : IF STATS EOP
6991 : * : : : : THEN
```

```
6992          : * : : CALL MOD 3.5 (OUTPUT STATISTICS)
6993          : * : : ENDIF
6994          : * : : ENDIF
6995          : * : : ENDIF
6996          : * : : ENDROUTINE
6997
6998
6999 043226      PUSH      <R1,R2,R3,R4>
7000
7001 043236 005737 002334      TST      STRST      ;SEE IF START OR RESTART COMMAND
7002 043242 001406          BEQ      16$      ;BRANCH IF NOT
7003 043244 012701 002452      MOV      #CMMAND,R1 ;GET FIRST DATA WORD FROM TEST MODULE
7004 043250 005021          CLR      (R1)+    ;CLEAR FIRST DATA WORD GET NEXT ONE
7005 043252 022701 002612      CMP      #RWDFLG,R1 ;SEE IF LAST WORD
7006 043256 100374          BPL      1$      ;BRANCH IF NOT LAST LOCATION
7007 043260 013737 002330 002336 16$: MOV      PTBLAD,STRTBL ;SET UP INDEX TABLE POINTER
7008 043266 012737 000001 002352 2$: MOV      #1,TBLNUM  ;SET UP LOOP ON UNIT TABLES
7009 043274 005037 002564          CLR      RECTBL   ;RESET MULTIPLE UNIT FLAG
7010 043300 004737 017720          JSR      PC,SEQDEF ;CALL MODULE 3.1 SEQUENCE DEFINITION
7011 043304 013703 002602          MOV      REPITN,R3
7012 043310 005737 002452          TST      CMMAND   ;SEE IF END OF PASS
7013 043314 001002          BNE      3$      ;IF END OF PASS LEAVE SCHEDULER
7014 043316 000137 044000          JMP      13$
7015 043322 013737 002352 002432 3$: MOV      TBLNUM,BITPST ;SET UP UNIT COUNTER
7016 043330 004737 036406          JSR      PC,SHIFTL
7017 043334 006237 002432          ASR      BITPST  ;MUST CORRECT SHIFT FOR UNIT #
7018 043340 013701 002432          MOV      BITPST,R1
7019 043344 013703 002602          MOV      REPITN,R3 ;RESTORE REPITITION #
7020 043350 022737 000003 002206      CMP      #3,CANSEQ ;SEE IF FIELD SERVICE #2 SEQUENCE
7021 043356 003013          BGT      5$      ;BRANCH IF NO
7022 043360 022737 000077 002452      CMP      #REDREV,CMMAND ;SEE IF READ REVERSE
7023 043366 001004          BNE      18$    ;BRANCH IF NO
7024 043370 013737 002602 002566      MOV      REPITN,RECORD ;RESTORE RECORD
7025 043376 000403          BR      5$      ;CONTINUE
7026 043400 012737 000001 002566 18$: MOV      #1,RECORD  ;RESET RECORD
7027 043406 010137 002530 5$: MOV      R1,UNITFL
7028 043412 030137 002540          BIT      R1,EOTFLG ;SEE IF UNIT SHOULD BE SKIPPED
7029 043416 001131          BNE      11$    ;BRANCH IF UNIT AT EOT
7030 043420          CALL     ASYERR  ;CHECK FOR ASYNC ERRORS ?
7031 043424 030137 002540          BIT      R1,EOTFLG ;DID WE DROP UNIT OR EOT
7032 043430 001124          BNE      11$    ;YES , REPORT AND PROCEED
7033          BR      20$ ;NO , TRY TO ISSUE COMMAND
7034 043432 004737 025146          JSR      PC,ISSCMD ;CALL MODULE 3.2 (ISSUE COMMAND)
7035 043436 022737 000007 002452      CMP      #REWIND,CMMAND ;SEE IF COMMAND ISSUED WAS A REWIND
7036 043444 001017          BNE      6$      ;BRANCH ON NO REWIND
7037 043446 004737 036436          JSR      PC,RANSET ;RESET RANDOM PATTERN
7038 043452 012737 177777 002426      MOV      #-1,GENFLG ;RESET RANDOM PATTERN ON REWIND
7039 043460 005237 002612          INC      RWDFLG  ;SET REWIND FLAG
7040 043464 005237 002610          INC      RWDCNT  ;SET REWIND COUNTER
7041 043470 023737 002610 002344      CMP      RWDCNT,TBLCNT ;SEE IF ALL UNITS REWINDING
7042 043476 001121          BNE      4$      ;BRANCH AROUND COMMAND TERMINATION
7043 043500 005037 002610          CLR      RWDCNT  ;CLEAR REWIND COUNTER ALL UNITS REWINDING
7044 043504 004737 026466          JSR      PC,CMDTRM ;CALL MODULE 3.3 (COMMAND TERMINATION)
7045 043510 005137 002534          COM      DRPFLG  ;COMPLEMENT DROP FLAG
7046 043514 043737 002534 002622      BIC      DRPFLG,RPTFLG ;CLEAR UNIT BIT NOT DROPPED
7047 043522 013737 002534 002622      MOV      DRPFLG,RPTFLG
```

7048	043530	005137	002534		COM	DRPFLG	:RESET DROP FLAG
7049	043534	022737	000377	002622	CMP	#377,RPTFLG	:SEE IF ANY UNITS DROPPED FROM REWIND
7050	043542	001004			BNE	19\$	:BRANCH IF NOT
7051	043544				DORPT		
7052	043546	012703	000001		MOV	#1,R3	:SET REPITITION TO ONE
7053	043552	000415			BR	10\$	
7054	043554	012737	000377	002622	19\$:	MOV #377,RPTFLG	:RESET DROP UNIT REPORT FLAG
7055	043562	030137	002534		BIT	R1,DRPFLG	:SEE IF UNIT SHOULD BE DROPPED
7056	043566	001407			BEQ	10\$	:BRANCH IF UNIT OKAY
7057	043570				DORPT		:CALL DO REPORT ROUTINE
7058	043572	005037	002464		CLR	STAFLG	:CLEAR ONE UNIT FLAG
7059	043576	005037	002604		CLR	RTYFLG	
7060	043602	012703	000001		MOV	#1,R3	
7061	043606	005737	002604	10\$:	TST	RTYFLG	:SEE IF COMMAND SHOULD RETRY
7062	043612	001403			BEQ	15\$	:BRANCH IF NO RETRY
7063	043614	004737	017720		JSR	PC,SEQDEF	:GET RETRY SEQUENCE
7064	043620	000704			BR	20\$	:BRANCH AROUND REPITITION
7065	043622	005037	002476	15\$:	CLR	HRDCNT	:CLEAR MAXIMUM ERROR COUNT
7066	043626	013737	002462	002452	MOV	RETRCD,CMMAND	
7067	043634				BREAK		
7068	043636	022703	000001		CMP	#1,R3	:SEE IF ISSUED LAST REPITITION
7069	043642	001417			BEQ	11\$	:IF YES, THEN CONTINUE
7070	043644	005303			DEC	R3	:ELSE DECREMENT REPITITION
7071	043646	022737	000003	002206	CMP	#3,CANSEQ	:SEE IF FIELD SERVICE CANNED SEQUENCE 2
7072	043654	003254			BGT	5\$	:BRANCH IF NOT
7073	043656	022737	000077	002452	CMP	#REDREV,CMMAND	:SEE IF READ REVERSE COMMAND
7074	043664	001003			BNE	17\$	:BRANCH IF NOT
7075	043666	005337	002566		DEC	RECORD	:DECREMENT RECORD #
7076	043672	000645			BR	5\$	:CONTINUE LOOP
7077	043674	005237	002566	17\$:	INC	RECORD	:INCREMENT RECORD #
7078	043700	000642			BR	5\$	:ISSUE SAME COMMAND
7079	043702	030137	002534	11\$:	BIT	R1,DRPFLG	:SEE IF STATISTICS HAVE BEEN OUTPUTED
7080	043706	001015			BNE	4\$	:BRANCH IF DROP UNIT SET
7081	043710	030137	002540		BIT	R1,EOTFLG	:SEE IF UNIT AT EOT
7082	043714	001412			BEQ	4\$	:BRANCH IF UNIT NOT AT EOT
7083	043716	050137	002534		BIS	R1,DRPFLG	
7084	043722	005737	002214		TST	STAEOT	:SEE IF USER WANTS STATS AT EOT
7085	043726	001405			BEQ	4\$	:BRANCH IF NO
7086	043730	005237	002464		INC	STAFLG	:SET ONE UNIT FLAG
7087	043734				DORPT		:CALL DO REPORT
7088	043736	005037	002464		CLR	STAFLG	:CLEAR ONE UNIT FLAG
7089	043742	005237	002352	4\$:	INC	TBLNUM	:GET NEXT UNIT
7090	043746	005037	002514		CLR	ERAFLG	:CLEAR MEDIA ERROR FLAG
7091	043752	013737	002352	002564	MOV	TBLNUM,RECTBL	:SET TABLE NUMBERS EQUAL
7092	043760	023737	002352	002344	CMP	TBLNUM,TBLCNT	:SEE IF LAST UNIT
7093	043766	101002			BHI	7\$	:LOOP IF NOT DONE
7094	043770	000137	043322		JMP	3\$	:NOT END OF PASS
7095	043774	000137	043266	7\$:	JMP	2\$	:
7096	044000	005737	002466	13\$:	TST	AUTORW	:SEE IF REWIND FROM EOT
7097	044004	001403			BEQ	9\$	:BRANCH IF NOT
7098	044006	005737	002214		TST	STAEOT	:SEE IF STATS AT EOT
7099	044012	001006			BNE	14\$	:BRANCH IF YES
7100	044014	005737	002212	9\$:	TST	STAEOP	:SEE IF STATS AT EOP
7101	044020	001403			BEQ	14\$	:BRANCH IF NOT
7102	044022	005037	002464		CLR	STAFLG	:CLEAR SINGLE UNIT FLAG
7103	044026				DORPT		:CALL DO REPORT STATISTICS

7104	044030	005037	002466	14\$:	CLR	AUTORW
7105	044034	005037	002622		CLR	RPTFLG
7106	044040				POP	<R4,R3,R2,R1>
7107						
7108						
7109						
7110	044050				EXIT	TST
7111						
7123						
7124					.EVEN	
7125						
7126	044054				ENDTST	
7127						
7133	044056				ENDMOD	
7134						

7137  
7148  
7149  
7186  
7187  
7188  
7189  
7190  
7191  
7192  
7193  
7194  
7195  
7196  
7197  
7198  
7199  
7205  
7206  
7207  
7208  
7209  
7210  
7211  
7212  
7219  
7220  
7221  
7222  
7223  
7224  
7225  
7226  
7227  
7228  
7229  
7230  
7231  
7232  
7239  
7240  
7241  
7242  
7243  
7244  
7245  
7246  
7247  
7248  
7249  
7250  
7251  
7258  
7259  
7260  
(3)

044056  
  
  
  
  
  
  
044056  
  
044060  
044070  
044100  
044112  
  
044124  
  
  
  
  
  
  
  
  
  
044124  
  
  
  
  
  
  
  
  
  
044126  
044134  
044136  
044150  
044156  
044164  
  
  
  
  
044164  
  
  
  
044164  
044170

```
.TITLE PARAMETER CODING
.SBTTL  HARDWARE PARAMETER CODING SECTION
      BGNMOD

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

      BGNHRD

      GPRMA  MSG1,0,0,0,177777,YES
      GPRMA  MSG2,2,0,0,177777,YES
      GPRMD  MSG3,4,D,177777,0,7,YES
      GPRMD  MSG4,6,D,177777,0,3,YES

      ENDHRD

.SBTTL  SOFTWARE PARAMETER CODING SECTION

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

      BGNSFT

      GPRML  MSG5,0,177777,YES
      XFERT  1$
      GPRMD  MSG6,2,D,177777,1,4,YES
      GPRML  MSG10,6,177777,YES
      GPRML  MSG7,10,177777,YES

1$:

      .EVEN

      ENDSFT

      LASTAD
L$LAST::
```

PARAMETER CODING MACY11 30(1046) 02-JUL-80 15:09 PAGE 6-1  
ZTMJA6.P11 23-JUN-80 10:32 SOFTWARE PARAMETER CODING SECTION

SEQ 0127

7261 044170  
7262 000001 .END ENDMOD















FMT11	041117	6445	6486#		
FMT12	041226	6447	6487#		
FMT13	041341	6448	6488#		
FMT14	041454	6449	6489#		
FMT15	041565	6450	6490#		
FMT16	041676	6451	6491#		
FMT17	041732	6446	6492#		
FMT2	037744	6427	6475#		
FMT3	040137	6437	6478#		
FMT4	040250	6438	6479#		
FMT5	040361	6439	6480#		
FMT6	040470	6440	6481#		
FMT7	040553	6441	6482#		
FMT8	040575	6442	6483#		
FMT9	040701	6443	6484#		
FM.E	015145	2342	2354#		
FM.NOP	014716	2334	2346#		
FM.REW	014731	2335	2347#		
FM.SL	015172	2343	2355#		
FM.SP	015011	2338	2350#		
FM.TM	015025	2339	2351#		
FM.TS	015060	2340	2352#		
FM.WC	015226	2344	2356#		
FM.WCK	014767	2337	2349#		
FM.WR	014746	2336	2348#		
FM.WV	015106	2341	2353#		
FORM01	006516	2124	2205#	2262	5680
FORM02	006576	2126	2206#	2264	
FORM03	006653	2127	2207#	2266	
FORM04	006733	2142	2208#		
FORM05	007013	2209#	2245		
FORM06	007066	2210#	2273	6089	
FORM07	007202	2212#	2244	2272	
FORM08	007157	2211#	5607	5990	
FORM09	007243	2213#	2250	2269	
FORM10	007314	2214#	6109		
FORM11	007365	2215#	2276		
FORM12	007463	2216#	2277		
FORM13	007603	2217#	2278		
FORM14	007723	2218#	2279		
FORM15	010043	2219#	2280		
FORM16	010163	2220#	2281		
FORM17	010303	2221#	2282		
FORM18	010423	2222#	2283		
FORM19	010543	2223#	2284		
FORM20	010663	2224#	2285		
FORM21	010734	2225#	2257		
FORM22	010773	2226#	5496	5564	
FORM23	011022	2125	2143	2227#	2263
FORM24	011055	2228#	2265		
FORM25	011101	2229#	6623		
FORM26	011132	2230#	6598		
FORM27	011154	2140	2231#		
FORM28	011213	2232#	5679		
FORM29	011266	2233#	2286		
FPT	033054	1665	5714#		















PARAMETER CODING  
ZTMJA6.P11

23-JUN-80 10:32

MACY11 30(1046)

02-JUL-80 15:09 PAGE 7-13  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0141

PICK5 = 000112 G	1375#	6450												
PICK6 = 000110 G	1374#	6450												
PICK7 = 000106 G	1373#	6124	6450											
PNT = 001000 G	1330#													
PRDD 003266	1877#	2284												
PRI = 002000 G	1330#													
PRIRTY 002366 G	1534#	2598	2643	4661	4783	6612*	6614*	6615*	6616*	6617*	6618*	6619	6620*	
	6621	6625*												
PRI00 = 000000 G	1330#	2633	4617	4753	6187									
PRI01 = 000040 G	1330#													
PRI02 = 000100 G	1330#													
PRI03 = 000140 G	1330#													
PRI04 = 000200 G	1330#													
PRI05 = 000240 G	1330#													
PRI06 = 000300 G	1330#													
PRI07 = 000340 G	1330#													
PSTAT 003265	1876#	2283												
PTBLAD 002330 G	1519#	2497*	2498	4685	5952	6412	6640	7007						
RALNTH 002224	1234#	2792	2793	2796*	3069*	3156*	3284*	3386*	4273					
RAMT 003237	1854#	2277												
RANB 002632	1629#	6310												
RANB2 002636	1631#	4359*	4360	6310*										
RANDOM 005010	2180#	2791												
RANGEN 026120	4279	4357#	4402											
RANS 002634	1630#	6311												
RANSET 036436	6310#	6634	7037											
RANS2 002640	1632#	4280*	4281	4283	4359	4360*	4361	6311*						
RCMLP 003236	1853#	2280												
RDATA 003246	1861#	2284												
RDON 003240	1855#	2278												
READOP 033732	1676	5832#												
RECORD 002566	1611#	2244	2272	2967*	3076*	3088*	3162*	3169*	3187*	3290*	3298*	3387*	3500*	
	3732*	3949*	3987*	5635	5639*	5641*	7024*	7026*	7075*	7077*				
RECTBL 002564 G	1610#	4275	7009*	7091*										
REDBUF 002332 G	1520#	2529*	2530*	4325	4327	4338	4340	6061						
REDEXT= 000073 G	1453#													
REDFWD= 000071 G	1452#	1708	1713	1725	1733	3087	3507	3517	3690	3895	3974	4013	4089	
REDREV= 000077 G	1454#	1707	1712	1724	1732	3505	3515	3677	3968	4007	4069	4085	4087	
	4323	5491	5503	5526	5647	5785	5808	5845	5863	6079	7022	7073		
REND 003243	1858#	2281												
REPCMD 002310 G	1500#	2758	2759	2760	2762	2764	2766							
REPITN 002602 G	1617#	2961*	3030*	3080*	3089*	3163*	3184*	3291*	3316*	3322*	3327*	3329*	3408*	
	3432*	3442*	3488*	3495*	3509*	3519*	3542*	3546*	3558*	3653*	3673*	3679*	3686*	
	3691*	3731*	3741*	3748*	3782*	3796*	3840*	3851*	3863*	3889*	3894*	3955*	3967*	
	3973*	3992*	4006*	4012*	4039*	7011	7019	7024						
REP1 002234	1238#													
REP2 002240	1240#													
REP3 002244	1242#													
REP4 002250	1244#													
REP5 002254	1246#													
REP6 002260	1248#	2751												
RETCT 003316	1901#	2283												
RETCT1 003317	1902#	2283												
RETRCD 002462 G	1577#	2960*	3028*	3079*	3090*	3161*	3183*	3289*	3313*	3394*	4075	4080	4085	
	4102	4110	4130	5491	5845	7066								
RETRY 002216	1231#	2777	2782*	2982	3073*	3153*	3281*	3384*	5619	5827	5836	5859		







PARAMETER CODING  
ZTMJA6.P11

23-JUN-80

MACY11 30(1046)  
10:32

02-JUL-80 15:09 PAGE 7-16  
CROSS REFERENCE TABLE -- USER SYMBOLS

B 12

SEQ 0144

TIMOUT	002542 G	1601#	2238	4650*	4653*	4699*	4702*	4767*	4770*	5974*	5976*			
TM	032320	1659	5628#											
TMFALB	034540	1683	5935#											
TMFALT	034416	1681	5912#											
TMNUM =	000004 G	1340#	2603	4171	4648	4705	4750	4800	5961	6178	6427	6894	6899	
TMOUT	034462	5913	5917	5920#										
TM78N	002370 G	1535#	2124	2140	2142	2262	2606*	2650	2688	2882	2884	2886	4648*	4674*
		4675	4705	4750*	4799*	5680	5941*	6899*						
TRECNT	002642 G	1640#	5601*	5602	5615*									
TUFALB	034516	1684	5929#											
TUFALT	034464	1682	5922#											
TUNUM =	000006 G	1341#	2613	4177	4193	4649	4707	4751	4813	6182	6427	6900		
TUSELO	003273	1882#	2280											
TUSEL1	003274	1883#	2281											
TUX	003320	1903#	2284											
TU78N	002372 G	1536#	2124	2142	2262	2616*	2654	2688	2882	2884	4649*	4676*	4677*	4679
		4683*	4707	4751*	4810*	5680	6900*							
TSARGC=	000001	1168#	2122#	2124#	2125#	2126#	2127#	2136#	2140#	2142#	2143#	2152#	2244#	2245#
		2250#	2257#	2262#	2263#	2264#	2265#	2266#	2269#	2272#	2273#	2276#	2277#	2278#
		2279#	2280#	2281#	2282#	2283#	2284#	2285#	2286#	2688#	2791#	2801#	2882#	2884#
		2886#	4293#	5496#	5564#	5607#	5679#	5680#	5990#	6089#	6109#	6426#	6427#	6434#
		6436#	6437#	6438#	6439#	6440#	6441#	6442#	6443#	6444#	6445#	6446#	6447#	6448#
		6449#	6450#	6451#	6598#	6623#	6630#	6638#	6653#	6659#				
TS CODE=	004130	2754#	2758#	2762#	2764#	2770#	2775#	2776#	2777#	2778#	2786#	2792#	2795#	2804#
		7206#	7207#	7208#	7209#	7240#	7241#	7242#	7243#	7244#				
TSERRN=	000042	1125#	3904#	4652#	4701#	4769#	5424#	5426#	5441#	5470#	5500#	5544#	5574#	5582#
		5597#	5644#	5657#	5701#	5717#	5728#	5735#	5741#	5747#	5757#	5763#	5769#	5782#
		5784#	5826#	5835#	5844#	5858#	5879#	5893#	5905#	5915#	5925#	5931#	5946#	5948#
		5975#	6085#											
TSEXCP=	000000	2754#	2758#	2762#	2764#	2786#	2795#	2804#	7206#	7207#	7208#	7209#	7242#	
TSFLAG=	000040	6471#	6670#	6723#	6755#	6788#	7110#							
TSGMAN=	000000	1125#	2754#	2758#	2762#	2764#	2786#	2795#	2804#					
TSHILI=	000004	2754#	2758#	2762#	2764#	2786#	2795#	2804#	7206#	7207#	7208#	7209#	7242#	
TSLAST=	000001	1125#	7260#											
TSLOLI=	000001	2754#	2758#	2762#	2764#	2786#	2795#	2804#	7206#	7207#	7208#	7209#	7242#	
TSLSYM=	010000	1125#	1216	1257	2129	2145	2155	6509	6675	6685	6739	6771	6804	7126
		7211	7249											
TSLTNO=	000001	7260#												
TSNEST=	177777	1125#	1151#	1202#	1216#	1224#	1257#	1259#	1321#	2117#	2129#	2131#	2145#	2147#
		2155#	6320#	6373#	6380#	6509#	6517#	6675#	6677#	6681#	6683#	6685#	6707#	6739#
		6747#	6771#	6780#	6804#	6806#	6868#	6918#	7126#	7133#	7187#	7198#	7211#	7231#
		7241	7249#	7261#										
TSNSO =	000000	1151#	1259	1321#	6320	6373#	6806	6868#	7133	7187#	7261			
TSNS1 =	000005	1202#	1216	1224#	1257	2117#	2129	2131#	2145	2147#	2155	6380#	6509	6517#
		6675	6677#	6681	6683#	6685	6707#	6739	6747#	6771	6780#	6804	6918#	7126
		7198#	7211	7231#	7241	7249								
TSPTNU=	000000	1125#												
TS SAVL=	177777	1125#												
TSSEGL=	177777	1125#												
TS SUBN=	000000	1125#	6918#											
TS TAGL=	177777	1125#												
TS TAGN=	010017	1125#	1202#	1224#	2117#	2131#	2147#	6380#	6517#	6677#	6683#	6707#	6747#	6780#
		6918#	7198#	7231#										
TS TEMP=	000000	1186#	1216#	1257#	1259#	2129#	2145#	2155#	2754#	2758#	2762#	2764#	2770#	2775#
		2776#	2777#	2778#	2786#	2792#	2795#	2804#	6320#	6471#	6509#	6670#	6675#	6681#
		6685#	6723#	6739#	6755#	6771#	6788#	6804#	6806#	7110#	7126#	7133#	7206#	7207#



XRETRY	003322	1905#	2285																
X\$ALWA=	000000	1125#																	
X\$FALS=	000040	1125#																	
X\$OFFS=	000400	1125#	7241																
X\$TRUE=	000020	1125#	7241																
.	= 044170	1148#	2075#	2235#	2332#	2357#	2634	4045	4622	6191	6471	6507#	6602	6670					
		6723	6755	6788	7110	7241													



MSBYTE	1168#														
MSCHEC	6471#	6670#	6723#	6755#	6788#	7110#									
MSCNTO	2754#	2758#	2762#	2764#	2770#	2775#	2776#	2777#	2778#	2786#	2792#	2795#	2804#	7206#	7207#
	7208#	7209#	7240#	7242#	7243#	7244#									
MSCOUN	2122#	2124#	2125#	2126#	2127#	2136#	2140#	2142#	2143#	2152#	2244#	2245#	2250#	2257#	2262#
	2263#	2264#	2265#	2266#	2269#	2272#	2273#	2276#	2277#	2278#	2279#	2280#	2281#	2282#	2283#
	2284#	2285#	2286#	2688#	2791#	2801#	2882#	2884#	2886#	4293#	5496#	5564#	5607#	5679#	5680#
	5990#	6089#	6109#	6426#	6427#	6434#	6436#	6437#	6438#	6439#	6440#	6441#	6442#	6443#	6444#
	6445#	6446#	6447#	6448#	6449#	6450#	6451#	6598#	6623#	6630#	6638#	6653#	6659#		
MSDATA	1168#	1170#	1172#												
MSDECR	1216#	1257#	1259#	2129#	2145#	2155#	6320#	6509#	6675#	6681#	6685#	6739#	6771#	6804#	6806#
	7126#	7133#	7211#	7249#	7261#										
MSDEFA	2754#	2758#	2762#	2764#	2770#	2775#	2776#	2777#	2778#	2786#	2792#	2795#	2804#	7206#	7207#
	7208#	7209#	7240#	7242#	7243#	7244#									
MSENDE	1216#	1257#	1259#	2129#	2145#	2155#	6320#	6509#	6675#	6685#	6739#	6771#	6804#	6806#	7126#
	7133#	7211#	7249#	7261#											
MSERRI	3904#	4652#	4701#	4769#	5424#	5426#	5441#	5470#	5500#	5544#	5574#	5582#	5597#	5644#	5657#
	5701#	5717#	5728#	5735#	5741#	5747#	5757#	5763#	5769#	5782#	5784#	5826#	5835#	5844#	5858#
	5879#	5893#	5905#	5915#	5925#	5931#	5946#	5948#	5975#	6085#					
MSEXCP	2754#	2758#	2762#	2764#	2786#	2795#	2804#	7206#	7207#	7208#	7209#	7242#			
MSEXIT	6471#	6670#	6723#	6755#	6788#	7110#									
MSEXSE	6471#	6670#	6723#	6755#	6788#	7110#									
MSEXTJ	6471#	6670#	6723#	6755#	6788#	7110#									
MSGEN	1168#	1170#	1172#	1186#	1202#	1216#	1224#	1257#	2117#	2129#	2131#	2145#	2147#	2155#	2754#
	2758#	2762#	2764#	2770#	2775#	2776#	2777#	2778#	2786#	2792#	2795#	2804#	6380#	6509#	6517#
	6675#	6677#	6683#	6685#	6707#	6739#	6747#	6771#	6780#	6804#	6918#	7126#	7198#	7211#	7231#
	7249#	7260#													
MSGENB	2754#	2758#	2762#	2764#	2770#	2775#	2776#	2777#	2778#	2786#	2792#	2795#	2804#		
MSGETS	1216#	1257#	1259#	2129#	2145#	2155#	6320#	6509#	6675#	6681#	6685#	6739#	6771#	6804#	6806#
	7126#	7133#	7211#	7241#	7249#	7261#									
MSGETT	6471#	6670#	6723#	6755#	6788#	7110#	7241#								
MSGNGB	1151#	1168#	1170#	1172#	1186#	1202#	1224#	1321#	2117#	2131#	2147#	6373#	6380#	6517#	6677#
	6683#	6707#	6747#	6780#	6868#	7187#	7198#	7231#	7260#						
MSGNIN	1168#	1170#	1172#	1186#	1202#	1224#	2122#	2124#	2125#	2126#	2127#	2129#	2136#	2140#	2142#
	2143#	2145#	2152#	2155#	2244#	2245#	2250#	2257#	2262#	2263#	2264#	2265#	2266#	2269#	2272#
	2273#	2276#	2277#	2278#	2279#	2280#	2281#	2282#	2283#	2284#	2285#	2286#	2499#	2500#	2598#
	2633#	2634#	2635#	2643#	2688#	2744#	2745#	2754#	2758#	2762#	2764#	2770#	2775#	2776#	2777#
	2778#	2786#	2791#	2792#	2795#	2801#	2804#	2882#	2884#	2886#	3904#	4045#	4046#	4293#	4617#
	4621#	4622#	4652#	4661#	4701#	4753#	4757#	4769#	4783#	5424#	5426#	5441#	5470#	5496#	5500#
	5544#	5564#	5574#	5582#	5597#	5607#	5644#	5657#	5679#	5680#	5701#	5717#	5728#	5735#	5741#
	5747#	5757#	5763#	5769#	5782#	5784#	5826#	5835#	5844#	5858#	5879#	5893#	5905#	5915#	5925#
	5931#	5946#	5948#	5975#	5990#	6085#	6089#	6109#	6187#	6191#	6426#	6427#	6434#	6436#	6437#
	6438#	6439#	6440#	6441#	6442#	6443#	6444#	6445#	6446#	6447#	6448#	6449#	6450#	6451#	6471#
	6509#	6582#	6583#	6585#	6586#	6590#	6594#	6595#	6596#	6597#	6598#	6601#	6602#	6609#	6610#
	6619#	6621#	6623#	6626#	6630#	6631#	6638#	6639#	6645#	6653#	6654#	6659#	6660#	6670#	6675#
	6685#	6723#	6739#	6755#	6771#	6788#	6804#	7051#	7057#	7067#	7087#	7103#	7110#	7126#	7198#
	7206#	7207#	7208#	7209#	7211#	7231#	7240#	7241#	7242#	7243#	7244#	7249#	7260#		
MSGNLS	2754#	2758#	2762#	2764#	2770#	2775#	2776#	2777#	2778#	2786#	2792#	2795#	2804#		
MSGNTA	1216#	1257#	2129#	2145#	2155#	6509#	6675#	6685#	6739#	6771#	6804#	7126#	7211#	7249#	
MSGNTE	6918#														
MSHAPT	1168#														
MSHNAP	1168#														
MSINCR	1151#	1202#	1224#	1321#	2117#	2122#	2124#	2125#	2126#	2127#	2129#	2131#	2136#	2140#	2142#
	2143#	2145#	2147#	2152#	2155#	2244#	2245#	2250#	2257#	2262#	2263#	2264#	2265#	2266#	2269#
	2272#	2273#	2276#	2277#	2278#	2279#	2280#	2281#	2282#	2283#	2284#	2285#	2286#	2499#	2598#
	2633#	2635#	2643#	2688#	2744#	2754#	2758#	2762#	2764#	2770#	2775#	2776#	2777#	2778#	2786#



PARAMETER CODING  
ZTMJA6.P11

23-JUN-80

MACY11 30(1046)  
10:32

02-JUL-80 15:09 PAGE 8-3  
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0150

	6434#	6436#	6437#	6438#	6439#	6440#	6441#	6442#	6443#	6444#	6445#	6446#	6447#	6448#	6449#
	6450#	6451#	6509#	6582#	6585#	6590#	6594#	6596#	6598#	6601#	6609#	6619#	6621#	6623#	6626#
	6630#	6631#	6638#	6639#	6645#	6653#	6654#	6659#	6660#	6670#	6675#	6685#	6723#	6739#	6771#
MSBSTL	6804#	7051#	7057#	7067#	7087#	7103#	7110#	7126#							
	2122#	2124#	2125#	2126#	2127#	2129#	2136#	2140#	2142#	2143#	2145#	2152#	2155#	2244#	2245#
	2250#	2257#	2262#	2263#	2264#	2265#	2266#	2269#	2272#	2273#	2276#	2277#	2278#	2279#	2280#
	2281#	2282#	2283#	2284#	2285#	2286#	2499#	2598#	2633#	2635#	2643#	2688#	2744#	2754#	2758#
	2762#	2764#	2770#	2775#	2776#	2777#	2778#	2786#	2791#	2792#	2795#	2801#	2804#	2882#	2884#
	2886#	3904#	4046#	4293#	4617#	4621#	4652#	4661#	4701#	4753#	4757#	4769#	4783#	5424#	5426#
	5441#	5470#	5496#	5500#	5544#	5564#	5574#	5582#	5597#	5607#	5644#	5657#	5679#	5680#	5701#
	5717#	5728#	5735#	5741#	5747#	5757#	5763#	5769#	5782#	5784#	5826#	5835#	5844#	5858#	5879#
	5893#	5905#	5915#	5925#	5931#	5946#	5948#	5975#	5990#	6085#	6089#	6109#	6187#	6426#	6427#
	6434#	6436#	6437#	6438#	6439#	6440#	6441#	6442#	6443#	6444#	6445#	6446#	6447#	6448#	6449#
	6450#	6451#	6509#	6582#	6585#	6590#	6594#	6596#	6598#	6601#	6609#	6619#	6621#	6623#	6626#
	6630#	6631#	6638#	6639#	6645#	6653#	6654#	6659#	6660#	6670#	6675#	6685#	6723#	6739#	6771#
MSWORD	6804#	7051#	7057#	7067#	7087#	7103#	7110#	7126#							
	1168#	1186#	2754#	2758#	2762#	2764#	2770#	2775#	2776#	2777#	2778#	2786#	2792#	2795#	2804#
	3904#	4652#	4701#	4769#	5424#	5426#	5441#	5470#	5500#	5544#	5574#	5582#	5597#	5644#	5657#
	5701#	5717#	5728#	5735#	5741#	5747#	5757#	5763#	5769#	5782#	5784#	5826#	5835#	5844#	5858#
	5879#	5893#	5905#	5915#	5925#	5931#	5946#	5948#	5975#	6085#	6471#	6670#	6723#	6755#	6788#
	7110#	7206#	7207#	7208#	7209#	7240#	7241#	7242#	7243#	7244#	7260				
MSXFER	7241#														
POINTE	1158														
POP	2123	2137	2153	2394#	2531	2694	2812	2852	2887	2993	3395	4204	4350	4364	4426
	4477	4883	5442	5590	5625	5696	5819	5977	6110	6130	6202	6236	6289	6305	6469
	6915	7106													
PRINTB	2122	2124	2125	2126	2127	2136	2140	2142	2143	2152	2244	2245	2250	2257	2262
	2263	2264	2265	2266	2269	2272	2273	5496	5564	5607	5679	5680	5990	6089	6109
PRINTF	2688	2791	2801	2882	2884	2886	4293	6598	6623	6630	6638	6653	6659		
PRINTS	6426	6427	6434	6436	6437	6438	6439	6440	6441	6442	6443	6444	6445	6446	6447
	6448	6449	6450	6451											
PRINTX	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286				
PUSH	2120	2134	2150	2388#	2492	2584	2743	2834	2871	2951	3380	4167	4270	4357	4396
	4455	4602	5418	5446	5595	5667	5773	5935	6056	6117	6176	6222	6283	6298	6411
	6890	6999													
REDEF	6582	6585	6594	6596											
RETURN	2376#	3412	3435	3443	3503	3511	3521	3544	3549	3560	3651	3665	3670	3675	3681
	3688	3699	3707	3710	3736	3743	3756	3773	3778	3791	3808	3844	3854	3869	3891
	3898	3913	3951	3965	3971	3982	3990	4004	4010	4027	4042	4053	4057	6916	
SETPRI	2633	2643	4617	4661	4753	4783	6187	6621							
SETVEC	2598	6619													
SVC	1124#	1125													
XFER	6471#	6670#	6723#	6755#	6788#	7110#									
XFERT	7241														

. ABS. 044170 000

ERRORS DETECTED: 0

CZTMJA,CZTMJA/CRF/EQ:ONEFILE=SVC33/ML,ZTMJA1.P11,ZTMJA2.P11,ZTMJA3.P11,ZTMJA4.P11,ZTMJA5.P11,ZTMJA6.P11  
RUN-TIME: 47 49 4 SECONDS  
RUN-TIME RATIO: 498/100=4.9  
CORE USED: 19K (38 PAGES)

PARAMETER CODING  
ZTMJA6.P11

MACY11 30(1046)  
23-JUN-80 10:32

02-JUL-80 15:09 PAGE 8-4  
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0151