

RX211,V21,02

RX02 FCTN/LGC
CZR XFA0

AH E626A MC

COPYRIGHT 1979

FICHE 1 OF 1

NOV 1979

digital

MADE IN USA

IDENTIFICATION

B 1

SEQ 0001

PRODUCT CODE: AC-E625A-MC
PRODUCT NAME: CZRXFAO RX02 FCTN/LGC
PRODUCT DATE: 10 AUG 79
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: L. S. PRUCHA

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 RESPONSIBLIITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADE MARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.2.1	HARDWARE REQUIREMENTS
1.2.2	SOFTWARE REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
1.6	MEMORY MAP
2.0	OPERATING INSTRUCTIONS
2.1	HARDWARE QUESTIONS
2.2	SOFTWARE QUESTIONS
3.0	ERROR INFORMATION
3.1	SYSTEM FATAL ERRORS
3.2	DEVICE FATAL ERRORS
3.3	HARD ERRORS
3.4	SOFT ERRORS
3.5	ERROR PRINTOUT FORMAT
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
5.1	DEVICE REGISTERS
5.2	DEVICE PROTOCOL
5.3	DEVICE HARDWARE CONFIGURATION
6.0	TEST SUMMARIES
7.0	LISTING INDEX
7.1	LISTING

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS PROGRAM CONTAINS A FUNCTION TEST OPTION AND A LOGIC TEST OPTION. A USER MAY SELECT TO RUN THE FUNCTION TEST ONLY, LOGIC TEST ONLY OR BOTH. THE DIAGNOSTIC WILL DEFAULT TO RUN THE LOGIC TEST ONLY. THE FUNCTION TEST WILL PERFORM A FUNCTIONAL EVALUATION OF THE DEVICE. IT WILL VERIFY THAT THE DRIVES CAN SEEK, THAT DATA CAN BE WRITTEN AND READ AND THAT DRIVE STATUS IS CORRECT. THE LOGIC TEST WILL ANALYZE DEVICE FAILURES, REPORT FAILING FIELD REPLACEABLE UNITS AND PROVIDE EXTENSIVE INFORMATION ON THE NATURE OF THE ERROR.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

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

1.2.2 SOFTWARE REQUIREMENTS

THIS DIAGNOSTIC IS DESIGNED TO RUN WITH THE DIAGNOSTIC SUPERVISOR AS DESCRIBED IN PARAGRAPH 2.0.

1.3 RELATED DOCUMENTS AND STANDARDS

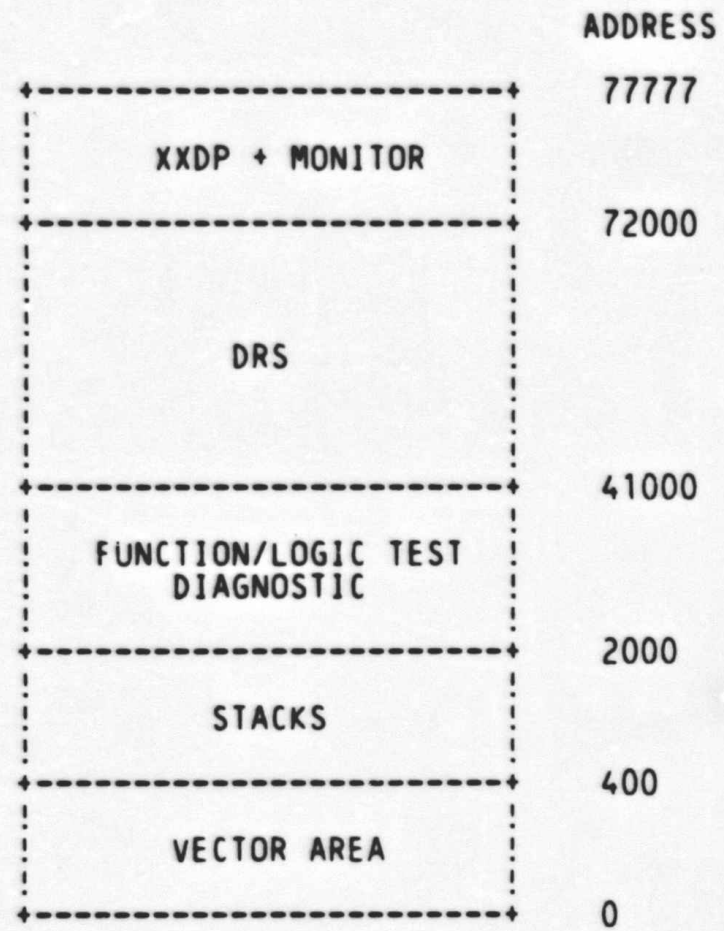
XXDP+ USERS MANUAL

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

NONE

1.5 ASSUMPTIONS

THIS DIAGNOSTIC ASSUMES THAT ALL HARDWARE OTHER THAN THE RXV21/RX211 INTERFACE OR RX02 SUBSYSTEM BEING TESTED WORKS PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, MEMORY, ETC., DOES NOT FUNCTION PROPERLY.

MEMORY LAYOUT ON 16K MACHINE - XXDP ENVIRONMENT

IN A MACHINE WITH MORE MEMORY FREE SPACE WILL OCCUR BETWEEN
THE DIAGNOSTIC AND THE DRS.

THIS IS A REV C SUPERVISOR DIAGNOSTIC: FOR OPERATING INSTRUCTIONS, PLEASE SEE CHAPTER 5 OF XXDP+ OPERATOR'S MANUAL. THEY ARE NO LONGER INCLUDED IN THE DIAGNOSTIC LISTING BECAUSE IT IS DESIRED THAT A CHANGE IN THOSE INSTRUCTIONS NOT REQUIRE A RE-ASSEMBLY OF ALL SUPERVISOR DIAGNOSTICS.

THE FOLLOWING SERIES OF QUESTIONS COMPRISE THE PARAMETERS NECESSARY TO IDENTIFY EACH FLOPPY DISK SUBSYSTEM.

RX ADDRESS -

THIS PARAMETER DEFINES THE BASE BUS ADDRESS FOR THE FLOPPY DISK SUBSYSTEM INTERFACE.

VECTOR ADDRESS -

THIS PARAMETER DEFINES THE INTERRUPT VECTOR ADDRESS FOR THE FLOPPY DISK SUBSYSTEM INTERFACE.

DRIVE -

THIS PARAMETER DEFINES THE FLOPPY DISK SUBSYSTEM DRIVE NUMBER (0 - 1).

EXPANSION-TYPE -

THIS PARAMETER IS TO BE USED FOR FUTURE EXPANSION. TYPE A CARRIAGE RETURN.

BR-LEVEL -

THIS PARAMETER DEFINES THE BR-LEVEL OF THE FLOPPY DISK SUBSYSTEM INTERFACE. A BR LEVEL OF 0 -> 7 WILL BE ACCEPTED.

SOFTWARE QUESTIONS

SEQ 0006

THE FOLLOWING SERIES OF QUESTIONS ARE INTENDED TO PROVIDE SELECTION OF VARIOUS TEST OPTIONS.

TEST HELP -

IF ANSWER IS YES "Y" THEN A SHORT HELP DESCRIPTION ON USE OF THIS DIAGNOSTIC WILL BE TYPED.

LOGIC TEST MODE -

IF ANSWER IS YES "Y" THEN THE LOGIC TESTS WILL BE DONE. THESE TESTS PROVIDE EXTENSIVE TESTING OF THE FLOPPY DISK SUBSYSTEM LOGIC. FAILING FIELD REPLACEABLE UNITS WILL BE CALLED OUT AND EXTENSIVE ERROR INFORMATION WILL BE REPORTED. THE AMOUNT OF ERROR INFORMATION MAYBE SUPPRESSED WITH THE "DRS" "IXE" FLAG.

FUNCTION TEST MODE -

IF ANSWER IS YES "Y" THEN THE FUNCTION TESTS WILL BE DONE. THESE TESTS PROVIDE A QUICK VERIFICATION THAT THE FLOPPY DISK SUBSYSTEM IS FUNCTIONAL, ONLY VERY BASIC ERROR REPORTING IS DONE, MEDIA RELATED ERRORS ARE IGNORED.

DEVICE FATAL THRESHOLD LEVEL -

THE DEVICE FATAL THRESHOLD LEVEL (DFTL) IS INITIALLY SET=1. THIS THRESHOLD LEVEL EQUALS THE NUMBER OR HARD ERRORS THAT WILL CAUSE A DEVICE FATAL ERROR WHEN THE DRS "EVL" FLAG IS SET.

NON-EXISTENT MEMORY ADDRESS -

THIS ADDRESS IS USED BY THE DIAGNOSTIC TO TEST THE RX CAPABILITY TO DETECT NON EXISTENT MEMORY (VIA BUS TIME OUT). THIS IS ONLY TESTED DURING THE NON EXISTENT MEMORY TEST. THE STANDARD 160000 DIAGNOSTIC ADDRESS IS USED BY DEFAULT.

EXTENDED ADDRESS BITS -

THESE BITS ARE USED DURING THE NPR & NON EXISTENT MEMORY TESTS TO TEST THE RX EXTENDED MEMORY CAPABILITIES. BITS 13 & 12 ARE SET IN THE RXCS REGISTER CORRESPONDING TO BITS 1 & 0 SET BY THE USER.

TEST CONTROL FLAGS -

IF ANSWER IS YES "Y", THEN THE FOLLOWING QUESTION WILL BE ASKED.

PRINT ONLY 10 DATA ERRORS & CONTINUE

IF THIS QUESTION IS ANSWERED NO "N", THEN ALL ERRORS IN THE RX DATA BUFFER WILL BE PRINTED. A YES ANSWER "Y" WILL CAUSE ONLY THE FIRST 10 BYTES IN ERROR TO BE PRINTED.

3.0 ERROR INFORMATION

THIS PROGRAM HAS THREE TYPES OF ERROR CLASSIFICATIONS; SYSTEM FATAL, DEVICE FATAL, AND HARD ERRORS.

3.1 SYSTEM FATAL ERRORS

SYSTEM FATAL ERRORS ARE USED TO INDICATE THAT AN ERROR WAS DETECTED BY THE DIAGNOSTIC SUPERVISOR IN RELATION TO LOADING/ CONTROLLING THE DIAGNOSTIC PROCESS. WHEN A SYSTEM FATAL ERROR IS DETECTED THE UNIT IS USUALLY DROPPED.

THE CONTENT OF EACH ERROR IS SUCH THAT IT SHOULD BE SELF - EXPLANATORY. HOWEVER, THE MESSAGES UTILIZE SOME TERMS THAT ARE SPECIFIC TO THE FLOPPY DISK SUBSYSTEM, AND MAY REQUIRE SOME GETTING USE TO.

3.2 DEVICE FATAL ERRORS

DEVICE FATAL ERRORS ARE A RESULT OF:

1. REACHING A DEVICE FATAL THRESHOLD LEVEL ("DVTL"). THIS LEVEL IS INITIALLY SET=1, BUT MAY BE MODIFIED BY THE OPERATOR. AN "DVTL" =1 WILL CAUSE 1 HARD ERROR TO BE CLASSIFIED A DEVICE FATAL ERROR.
2. AN ERROR THAT IS CONSIDERED FATAL TO THE DEVICE, BUT TESTING WILL CONTINUE.

3.3 HARD ERRORS

HARD ERRORS ARE A RESULT OF: A NON-RECOVERABLE ERROR

3.5 ERROR PRINTOUT FORMAT

EACH ERROR WILL BE PRINTED OUT USING THE STANDARD "DRS" HEADER.

3.5.1 FUNCTION TESTS

THE SECOND LINE PRINTED OUT WILL GIVE THE TEST TITLE
THE THIRD LINE PRINTED OUT WILL IDENTIFY THE ERROR. IF IT
IS A CSR ERROR THE ACTUAL AND EXPECTED RESULTS WILL BE DISPLAYED.

EXAMPLE ERROR PRINTOUT:

```
CZRFXAO HRD ERR 00004 ON UNIT 01 TST 010 SUB 000 PC:003476
POSITIONING - FNC TST
CSR- ERROR
    REG ACTUAL=005520
    REG EXPECT=037566
```

3.5.2 LOGIC TESTS

THE SECOND AND THIRD LINES WILL BE PRINTED AS DESCRIBED FOR
THE FUNCTION TESTS.
DEPENDING ON THE TYPE OF ERROR ADDITIONAL ACTUAL AND EXPECTED
RESULTS WILL BE DISPLAYED. THEN THE TEST WILL CALL OUT WHICH ARE
THE MOST LIKELY FIELD REPLACEABLE UNITS "FRU'S" THAT ARE
FAILING. ALL CURRENT DEVICE REGISTERS ARE THEN DISPLAYED,
INCLUDING A DATA BUFFER DUMP IF DATA WAS BAD.

EXAMPLE ERROR PRINTOUT:

```
CZRFXAO DEV FTL ERR 00019 ON UNIT 01 TST 024 SUB 000 PC:003476
WRD CNT INTEGRITY PRT:1 - LGC TST
WORD COUNT ERROR
    REG ACTUAL=000020
    REG EXPECT=000000
```

POSSIBLE FAILING "FRU'S":
CONTROLLER - M7744
INTERFACE - M8256

```
UNIT#1 RXCSR=014440 RXESR=010040 CMD=000437 ->READ ERR CODE
ERROR CODE=230 ->WORD CNT OVF.
WORD CNT=020
CUR TRK DV0=76. CUR TRK DRV1= 0.
TARGET TRK =76. TARGET SEC =10. SOFT STAT=060 BAD TRK=15.
```


4.0 PERFORMANCE AND PROGRESS REPORTS

NONE

5.0 DEVICE INFORMATION

5.1 DEVICE REGISTERS

	! <FUNCTION>!															
	!15	!14	!13	!12	!11	!10	!09	!08	!07	!06	!05	!04	!03	!02	!01	!00
RXCS:	ERR	INT	XM	XM	RX2		SID	DEN	TR	IE	DON	DRV	FUN	FUN	FUN	GO
RXWC:	X	X	X	X	X	X	X	X								WORD COUNT
RXBA:	BUS ADDRESS															
RXES:	X	X	X	X	NXM	WC	SID	DRV	DRV	DEL	DSK	DEN	AC	INT	SID	CRC
						OVF	#1	#1	RDY	DAT	DEN	ERR	LOW	DON	RDY	
RXTA:	X	X	X	X	X	X	X	X	0							TRACK ADDRESS
RXSA:	X	X	X	X	X	X	X	X	0	0	0					SECTOR ADDRESS
RXDB:	DATA BUFFER															

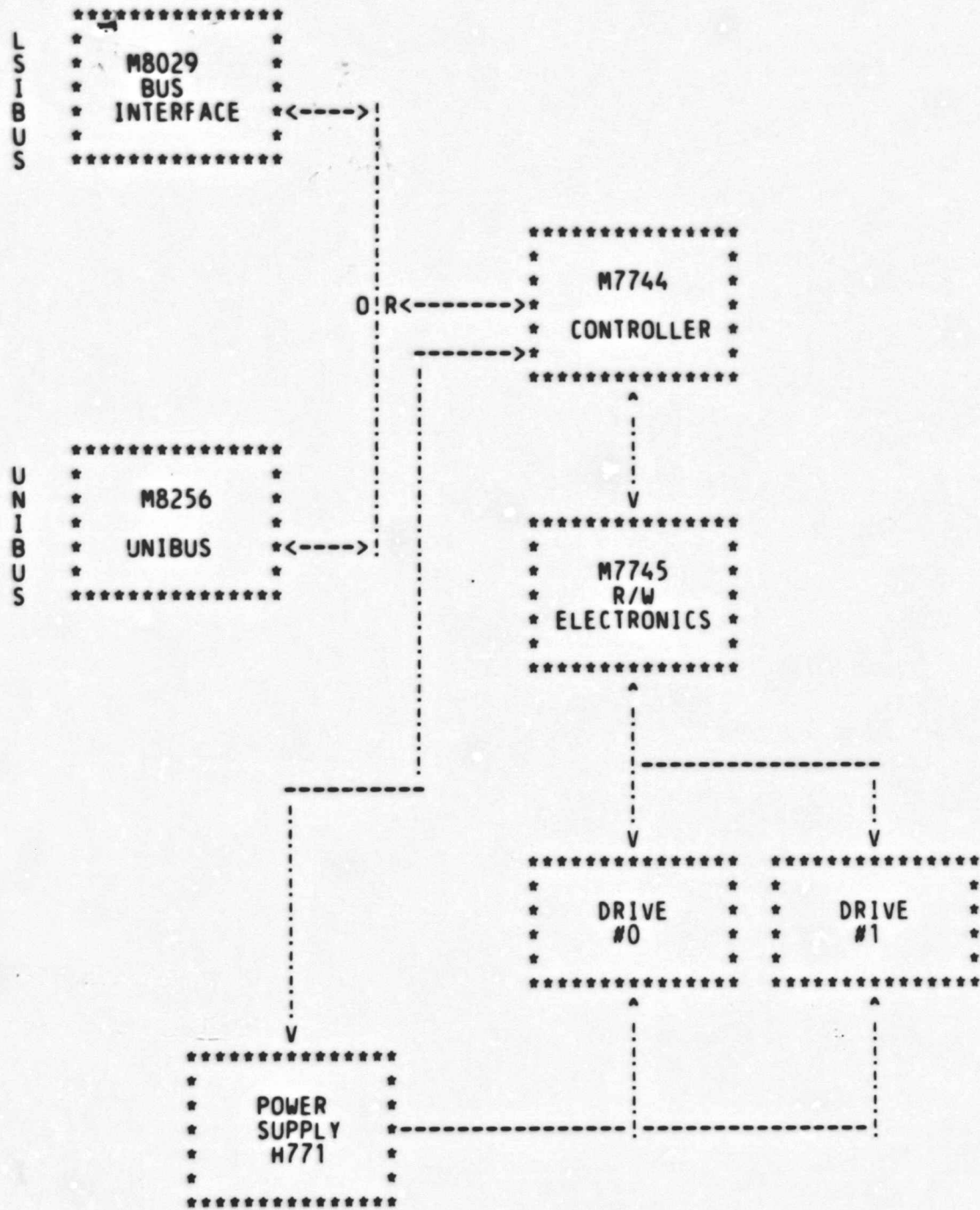
READ ERROR CODE REGISTERS - (SEE LABEL "XERUUT")

WORD	!15	!14	!13	!12	!11	!10	!09	!08	!07	!06	!05	!04	!03	!02	!01	!00
#1	WORD COUNT								ERROR CODE							
#2	CURRENT TRACK DRV #1								CURRENT TRACK DRIVE #0							
#3	TARGET SECTOR								TARGET TRACK							
#4	BAD TRACK-ONLY VALID IF ERRCODE=150								UNT	DV1	HD	DV0	X	X	X	LCD
									SEL	DEN	LD	DEN				DEN

RX02 FUNCTIONAL PROCESS

FUNCTION CODE BIT # 3 2 1	FUNCTION	PROCEDURE (PROTOCOL)
0 0 0	FILL BUFFER	Function Word --->TR--->WC--->TR--->BA--->DONE
0 0 1	EMPTY BUFFER	Function Word --->TR--->WC--->TR--->BA--->DONE
0 1 0	WRITE SECTOR	Function Word --->TR--->SA--->TR--->TA--->DONE
0 1 1	READ SECTOR	Function Word --->TR--->SA--->TR--->TA--->DONE
1 0 0	SET DENSITY	Function Word --->TR--->VW--->DONE
1 0 1	READ MAINT. STATUS	Function Word --->DONE
1 1 0	WRITE SECTOR with deleted data	Function Word --->TR--->SA--->TR--->TA--->DONE
1 1 1	READ ERROR CODE	Function Word --->TR--->BA--->DONE

TR = wait for TR BIT
 DONE = wait for DONE BIT
 BA = BUS ADDRESS (output to RX)
 VW = VERIFICATION WORD (output to RX)
 WC = WORD COUNT (output to RX)
 SA = SECTOR ADDRESS (output to RX)
 TA = TRACK ADDRESS (output to RX)



6.0 TEST SUMMARIES

SEQ 0012

TEST 1 - INITIALIZE - FNC TST

TEST TO VERIFY THAT AN RX INITIALIZE WILL RETURN THE DEVICE TO A VALID STATE.

DESCRIPTION:

1. DO BUS INITIALIZE
2. IF RX ERR BIT IS SET REPORT ERROR
3. CALL PROGRAMMED INITIALIZE
4. IF RX ERR BIT IS SET REPORT ERROR

TEST 2 - READ ERROR CODE - FNC TST

TEST TO VERIFY THAT THE DEVICE WILL COMPLETE A READ ERROR CODE COMMAND WITHOUT ENCOUNTERING AN ERROR.

DESCRIPTION:

1. CALL PROGRAMMED INITIALIZE
2. IF RX ERR BIT IS SET REPORT ERROR
3. CALL READ ERROR CODE
4. IF RX ERR BIT IS SET REPORT ERROR

TEST 3 - FILL BUFFER - FNC TST

TEST TO VERIFY THE DEVICE BUFFER WILL FILL WITH NO RESULTING ERROR.

DESCRIPTION:

1. CALL FILL BUFFER
2. IF RX ERR BIT IS SET REPORT ERROR

TEST 4 - EMPTY BUFFER - FNC TST

TEST TO VERIFY THE DEVICE BUFFER WILL EMPTY WITHOUT ERRORS.

DESCRIPTION:

1. CALL EMPTY BUFFER
2. IF RX ERR BIT IS SET REPORT ERROR

TEST 5 - READ STATUS - FNC TST

TEST TO VERIFY THAT A DEVICE MAINTENANCE READ STATUS (RXES) COMMAND WILL EXECUTE WITHOUT ERROR.

DESCRIPTION:

1. CALL READ STATUS
2. IF RX ERR BIT IS SET REPORT ERROR

TEST 6 - FILL & EMPTY BUFFER - FNC TST

TEST TO VERIFY THE DEVICE BUFFER DATA IS VALID AFTER A FILL/EMPTY BUFFER COMMAND SEQUENCE.

DESCRIPTION:

1. SETUP RANDOM DATA PATTERN
2. CALL FILL BUFFER
3. IF RX ERR BIT IS SET REPORT ERROR
4. CALL EMPTY BUFFER
5. IF RX ERR BIT IS SET REPORT ERROR
6. CALL DATA CHECK

TEST 7 - READ & WRITE SECTOR - FNC TST

TEST TO VERIFY THE DEVICE WILL READ AND WRITE IN BOTH DENSITIES WITHOUT AN ERROR.

DESCRIPTION:

1. SETUP TO DO TEST IN WRONG DENSITY
2. CALL WRITE SECTOR
3. IF RX ERR BIT IS NOT SET REPORT ERROR
4. CALL READ SECTOR
5. IF RX ERR BIT IS NOT SET REPORT ERROR
6. SETUP CORRECT DENSITY
7. CALL WRITE SECTOR
8. IF RX ERR BIT IS SET REPORT ERROR
9. CALL READ SECTOR
10. IF RX ERR BIT IS SET REPORT ERROR

TEST 8 - WRITE SECTOR DELETED DATA - FNC TST

TEST TO VERIFY THAT THE DEVICE WILL WRITE A DELETED DATA MARK ON THE DISKETTE WITHOUT ERROR.

DESCRIPTION:

1. SETUP TEST TO CORRECT DENSITY AND DELETED DATA MODE
2. CALL WRITE SECTOR DELETED DATA
3. IF RX ERR BIT IS SET REPORT ERROR
4. CALL READ SECTOR
5. IF RX ERR BIT IS SET REPORT ERROR
6. CLEAR DELETED DATA MODE
7. CALL WRITE SECTOR (CLEAR DELETED DATA MARK)
8. IF RX ERR BIT IS SET REPORT ERROR

TEST 9 - SET DENSITY - FNC TST

TEST TO VERIFY THE DEVICE WILL CHANGE DENSITIES WITHOUT INCURRING AN ERROR.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. CALL SET DENSITY
3. IF RX ERR BIT IS SET REPORT ERROR
4. CALL READ SECTOR
5. IF RX ERR BIT IS SET OR DENSITY NOT CORRECT REPORT ERROR
6. SETUP DENSITY = OPPOSITE DISK DENSITY
7. CALL SET DENSITY
8. IF RX ERR BIT IS SET REPORT ERROR
9. CALL READ SECTOR
10. IF RX ERR BIT IS SET OR DENSITY NOT CORRECT REPORT ERROR
11. SETUP DENSITY = DISK DENSITY
12. CALL SET DENSITY
13. IF RX ERR BIT IS SET REPORT ERROR

TEST 10 - POSITIONING - FNC TEST

TEST TO VERIFY THE DEVICE WILL CHANGE SECTORS AND TRACKS WITHOUT INCURRING AN ERROR.

DESCRIPTION:

1. SETUP RANDOM TRACK PATTERN AND DENSITY = DISK DENSITY
2. CALL GET A TRACK & SECTOR
3. CALL READ SECTOR
4. IF RX ERR BIT IS SET REPORT ERROR
5. DO 2. -> 4. UNTIL 76. TRACKS DONE

TEST 11 - CSR BITS - LGC TST

TEST TO VERIFY THAT THE READ/WRITE BITS OF THE CONTROL AND STATUS REGISTER CAN BE WRITTEN INTO AND READ AND OTHERWISE BEHAVE AS EXPECTED.

DESCRIPTION:

1. LOAD RX CSR WITH 1'S
2. CHECK & REPORT THAT ALL BITS THAT SHOULD SET, DO SET
3. LOAD RX CSR WITH 0'S
4. CHECK & REPORT THAT ALL BITS THAT SHOULD NOT BE SET, ARE NOT SET

TEST 12 - DBR BITS - LGC TST

TEST TO VERIFY THAT THE READ/WRITE BITS OF THE DATA BUFFER REGISTER CAN BE WRITTEN INTO AND READ AS EXPECTED.

DESCRIPTION:

1. WRITE RX DBR WITH ALL 1'S
2. CHECK & REPORT ALL BITS THAT SHOULD & SHOULD NOT BE SET
3. WRITE RX DBR WITH ALL 0'S
4. CHECK & REPORT ALL BITS THAT SHOULD & SHOULD NOT BE SET

TEST 13 - CSR-DBR COMMON BYTE - LGC TST

TEST TO VERIFY THAT THE LOWER BYTE OF THE RXCS MAPS INTO THE RXDB AND THEREFORE CHECK WRITE ONLY BITS OF THE RXCS.

DESCRIPTION:

1. LOAD RX CSR LOW BYTE WITH ALL 1'S (EXCEPT BIT#0)
2. CHECK & REPORT IF RX DBR LOW BYTE NOT EQUAL TO ALL 1'S (EXCEPT BIT#0 & BIT#3)
3. LOAD RX CSR LOW BYTE WITH ALL 0'S
4. CHECK & REPORT IF RX DBR LOW BYTE NOT EQUAL TO ALL 0'S

TEST 14 - BUS INITIALIZE - LGC TST

TEST TO VERIFY THAT THE INTERFACE BOARD WILL COMPLETE A BUS INITIALIZE.

DESCRIPTION:

1. ISSUE BUS INITIALIZE
2. CHECK & REPORT IF ERROR BIT OR AC LOW BIT ARE SET OR IF DONE BIT IS NOT SET

TEST 15 - PROGRAMMED INITIALIZE - LGC TST

TEST TO VERIFY THAT THE INTERFACE BOARD WILL COMPLETE A PROGRAMMED INITIALIZE.

DESCRIPTION:

1. CALL PROGRAMMED INITIALIZE
2. CALL DEVICE STATE CHECK
3. CHECK & REPORT ERRORS

TEST 16 - POWER FAIL - LGC TST

TEST TO VERIFY THAT THE ACLOW CIRCUITS OPERATE AS EXPECTED.

DESCRIPTION:

1. IF MANNUAL INTERVENTION ALLOWED ASK OPERATOR TO POWER DOWN RX
2. IF POWERED DOWN, THEN CHECK & REPORT IF AC LOW BIT NOT SET
3. ASK OPERATOR TO POWER UP RX
4. IF POWERED UP, THEN INITIALIZE, CHECK & REPORT IF AC LOW BIT SET

TEST 17 - CONTROLLER-INTERFACE - LGC TST

TEST TO VERIFY THAT THE INTERFACE BOARD STATE SEQUENCER IS FUNCTIONAL. ALSO TO VERIFY THE CONTROLLER-INTERFACE HANDSHAKE BY TRYING FUNCTIONS WITH MINIMUM READ/WRITE BOARD INVOLVEMENT.

DESCIRPTION:

1. CALL READ ERROR CODE
2. IF ERROR, THEN REPORT ERROR
3. CALL FILL BUFFER
4. IF ERROR, THEN REPORT ERROR
5. CALL EMPTY BUFFER
6. IF ERROR, THEN REPORT ERROR
7. CALL READ STATUS

8. IF ERROR, THEN REPORT ERROR

0 2

SEQ 0016

TEST 18 - NPR - LGC TST

TEST TO VERIFY THAT THE NPR LOGIC WILL STORE A WORD IN MEMORY.

DESCRIPTION:

1. SETUP MEMORY LOCATION
2. CALL READ ERROR CODE (TO WRITE OVER LOCATION)
3. IF ERROR, THEN REPORT NPR ERROR
4. SETUP BUFFER AREAS BEGIN, END & END+1
5. CALL FILL BUFFER
6. IF ERROR, THEN REPORT ERROR
7. CALL EMPTY BUFFER
8. IF ERROR, THEN REPORT ERROR
9. CHECK BUFFER AREAS BEGIN & END SHOULD CHANGE & END+1 SHOULD NOT, REPORT AS NPR ERROR, IF CONDITIONS NOT MET

TEST 19 - NPR NON-EXISTENT MEM - LGC TST

TEST TO VERIFY THAT THE NPR NON-EXISTEND MEMORY LOGIC WILL TIME OUT WHEN GIVEN AN ILLEGAL ADDRESS.

DESCRPTION:

1. SETUP BUS TRAPS AND NONEXSISTANT MEMORY ADDRESS
2. CALL READ ERROR CODE
3. IF RX CSR ERROR BIT OR RX ESR NXM BIT NOT SET, THEN CALL ERROR
4. CALL INITIALIZE (CLEAR RX ERROR)
5. CLEAR BUS TRAP VECTOR

TEST 20 - INTERRUPT - LGC TST

TEST TO VERIFY THAT THE INTERRUPTS CAN BE SET AND THAT THE DEVICE RESPONDS AS EXPECTED.

DESCRIPTION:

1. SET PROCESSOR PRIORITY = 7 (NO INTERRUPTS)
2. SET RX INTERRUPT BIT & SETUP LOWER PRIORITY
3. CALL WATCH TO LOWER PROCESSOR PRIORITY & WAIT FOR INTERRUPT
4. CALL ERROR IF DID NOT INTERRUPT
5. CLEAR RX INTERRUPT BIT

TEST 21 - PRIORITY LVL - LGC TST

TEST TO VERIFY THAT THE DEVICE PRIORITY IS SET TO THE CORRECT LEVEL.

DESCRIPTION:

1. SETUP PROCESSOR PRIORITY = 7 (NO INTERRUPTS)
2. DO SET PROCESSOR PRIORITY
3. SET RX INTERRUPT BIT
4. IF INTERRUPT OCCURED, THEN CHECK LEVEL & REPORT IF PROCSSOR PRIORITY NOT LOWER THAN RX
5. IF INTERRUPT DID NOT OCCUR, THEN SETUP NEXT LOWER PROCESSOR PRIORITY & START AT 2. AGAIN

TEST 22 - INITIALIZE CONTROL - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE INITIALIZE.

DESCRIPTION:

1. CALL PROGRAMMED INITIALIZE
2. IF ERROR, THEN REPORT ERROR

TEST 23 - DATA BUF INTEGRITY - LGC TST

TEST TO VERIFY ALL BITS OF DATA BUFFER, VARIOUS PATTERNS WILL BE USED.

DESCRIPTION:

1. SETUP RANDOM DATA PATTERN
2. CALL FILL BUFFER
3. IF ERROR, THEN REPORT ERROR
4. CALL EMPTY BUFFER
5. IF ERROR, THEN REPORT ERROR
6. CALL DATA CHECK
7. SETUP NEW DATA PATTERN
8. DO 2. -> 7. UNTIL ALL DATA PATTERNS DONE

TEST 24 - WRD CNT INTEGRITY - LGC TST

TEST TO VERIFY ALL BITS OF WORD COUNT REGISTER AND CHECK THAT EXCEEDING THE WORD COUNT FOR DISKETTE DENSITY WILL BE DETECTED.

DESCRIPTION:

1. SETUP BUFFER LENGTH = 128.
2. CALL FILL BUFFER
3. IF ERROR, THEN REPORT ERROR
4. CALL READ ERROR CODE
5. IF ERROR, THEN REPORT ERROR
6. IF RX WORD COUNT NOT = 0, THEN CALL ERROR
7. DECREMENT WORD COUNT TO RX, DO 2. -> 6. UNTIL WORD COUNT TO RX IS = 0

TEST 25 - CONTROLLER-READ*WRITE ELECT - LGC TST

TEST TO VERIFY MINIMAL CONTROLLER BOARD-READ/WRITE ELECTRONICS BOARD INTERFACE VIA INITIALIZE OF A SELECTED DRIVE.

DESCRIPTION:

1. CALL PROGRAMMED INITIALIZE
2. IF ERROR, THEN REPORT ERROR
3. CALL READ ERROR CODE
4. IF ERROR, THEN REPORT ERROR

TEST 26 - READ SECTOR-PRT:1 - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A READ SECTOR IN BOTH DENSITIES AND RETURN A VALID ERROR CODE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. CALL READ SECTOR
3. CALL READ ERROR CODE
4. IF ERROR, THEN REPORT ERROR
5. SETUP DENSITY = OPPOSITE DISK DENSITY
6. CALL READ SECTOR
7. CALL READ ERROR CODE
8. IF ERROR, THEN REPORT ERROR

TEST 27 - POSITIONING - LGC TST

TEST TO VERIFY THAT THE DRIVE WILL READ THE HEADERS ON ALL TRACKS OF THE DRIVE AS EXPECTED.

DESCRIPTION:

1. SETUP RANDOM TRACKS MODE
2. CALL GET A TRACK
3. CALL READ SECTOR
4. CALL READ ERROR CODE
5. IF TRACK OR OTHER ERROR, THEN REPORT ERROR
6. DO 2. -> 5. UNTIL 76. TRACKS DONE

TEST 28 - WRITE SECTOR-PRT:1 - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A WRITE SECTOR IN BOTH DENSITIES AND RETURN A VALID ERROR CODE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. CALL WRITE SECTOR
3. IF ERROR, THEN REPORT ERROR
4. SETUP DENSITY = OPPOSITE DISK DENSITY
5. CALL WRITE SECTOR
6. IF NO DENSITY ERROR, THEN REPORT ERROR

TEST TO VERIFY THAT THE DEVICE SET A DELETED DATA MARK ON THE DISKETTE HEADER AND RETRIEVE THIS DATA ON A KNOWN GOOD DISKETTE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. SETUP DELETED DATA MODE
3. CALL WRITE SECTOR
4. IF ERROR, THEN REPORT ERROR
5. CALL READ SECTOR
6. IF RX CSR DELETED DATA BIT NOT SET, THEN REPORT ERROR
7. CLEAR DELETED DATA MODE
8. CALL WRITE SECTOR (CLEAR DELETED DATA MARK)

TEST 30 - SET DENSITY - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE SET DENSITY IN BOTH DENSITIES. THE VALID WORD WILL ALSO BE CHECKED. ALSO TO VERIFY THAT THE DRIVE WILL READ IN BOTH DENSITIES, WITHOUT ERRORS.

DESCRIPTION:

1. GET & SAVE DISK DENSITY
2. SETUP DENSITY = SINGLE
3. CALL SET DENSITY
4. IF ERROR, THEN REPORT ERROR
5. SETUP INVALID KEY WORD = ASCII "K"
6. CALL SET DENSITY
7. IF NO DENSITY ERROR, THEN REPORT ERROR
8. SETUP VALID KEY WORD = ASCII "I"
9. SETUP DENSITY = DOUBLE
10. CALL SET DENSITY
11. IF ERROR, THEN REPORT ERROR
12. CHECK DISK DENSITY & REPORT IF NOT SET = DOUBLE
13. IF SAVED DISK DENSITY = DOUBLE, THEN SET DENSITY = SINGLE AND CALL SET DENSITY

TEST 31 - SECTOR ADR - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL HANDLE ALL LEGAL AND ILLEGAL SECTOR ADDRESSES PROPERLY.

DESCRIPTION:

1. GET A SECTOR
2. CALL READ SECTOR
3. CALL READ ERROR CODE
4. IF SECTOR ADDRESS NOT = RX SECTOR ADDRESS OR OTHER ERROR, THEN REPORT ERROR
5. DO 1. -> 4. UNTIL ALL SECTORS DONE OR ERROR OCCURS
6. SETUP SECTOR = 0 (ILLEGAL SECTOR)
7. CALL READ SECTOR
8. CALL READ ERROR CODE
9. IF NO SECTOR ERROR OR IF OTHER ERROR, THEN REPORT ERROR

TEST 32 - TRACK ADR - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL HANDLE ALL LEGAL AND ILLEGAL TRACK ADDRESSES PROPERLY.

DESCRIPTION:

1. GET A TRACK
2. CALL READ SECTOR
3. CALL READ ERROR CODE
4. IF TRACK ADDRESS NOT = RX TRACK ADDRESS, THEN CALL ERROR
OR IF OTHER TRACK ERROR OCCURED, THEN CALL ERROR
5. DO 1. -> 4. UNTIL ALL TRACKS DONE OR FINI FLAG SET (COMMAND ERROR)
6. SETUP ILLEGAL TRACK
7. CALL READ SECTOR
8. CALL READ ERROR CODE
9. IF TRACK ADDRESS NOT = RX TRACK ADDRESS OR
IF ERROR CODE NOT = 40 (TRACK > 76.), THEN CALL ERROR

TEST 33 - READ SECTOR-PRT:2 - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A READ SECTOR IN BOTH DENSITIES AND RETURN A VALID ERROR CODE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. CALL READ SECTOR
3. CALL READ ERROR CODE
4. IF ERROR, THEN REPORT ERROR
5. SETUP DENSITY = OPPOSITE DISK DENSITY
6. CALL READ SECTOR
7. CALL READ ERROR CODE
8. IF NOT DENSITY ERROR, THEN REPORT ERROR

TEST 34 - WRITE SECTOR-PRT:2 - LGC TST

TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A WRITE SECTOR IN BOTH DENSITIES AND RETURN A VALID ERROR CODE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. CALL WRITE SECTOR
3. IF ERROR, THEN REPORT ERROR
4. SETUP DENSITY = OPPOSITE DISK DENSITY
5. CALL WRITE SECTOR
6. IF NOT DENSITY ERROR, THEN REPORT ERROR

TEST TO VERIFY THAT THE DEVICE SET A DELETED DATA MARK ON THE DISKETTE HEADER AND RETRIEVE THIS DATA ON A KNOWN GOOD DISKETTE. THIS IS DONE IN OPPOSITE DENSITY OF PART: 1.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. SETUP DELETED DATA MODE
3. CALL WRITE SECTOR
4. CALL READ ERROR CODE
5. IF ERROR, THEN REPORT ERROR
6. CALL READ SECTOR
7. IF RX ESR DELETED DATA BIT NOT SET OR OTHER ERROR, THEN REPORT ERROR

TEST 36 - DISKETTE & DENSITY DATA CHECK - LGC TST

TEST TO VERIFY WITH A KNOWN GOOD DISKETTE THAT THE DEVICE WILL READ AND WRITE TO THE DISKETTE WITHOUT DATA ERRORS. BOTH DENSITIES WILL BE DONE.

DESCRIPTION:

1. SETUP DENSITY = DISK DENSITY
2. SETUP RANDOM DATA PATTERN
3. GET A TRACK & SECTOR
4. CALL FILL BUFFER
5. CALL WRITE SECTOR
6. SETUP TO CLEAR RX INTERNAL BUFFER
7. CALL FILL BUFFER
8. CALL READ SECTOR
9. CALL EMPTY BUFFER
10. CALL DATA CHECK
11. DO 3. -> 10. UNTIL AT LEAST ONE SECTOR OF EACH TRACK IS ACCESSED
12. SETUP DENSITY = OPPOSITE DISK DENSITY
13. CALL SET DENSITY
14. DO 3. -> 13. UNTIL BOTH DENSITIES DONE

2-	23	PROGRAM HEADER
2-	92	DISPATCH TABLE
3-	108	DEFAULT HARDWARE P-TABLE
3-	129	LOAD DEVICE PROTECTION
3-	141	SOFTWARE P-TABLE
5-	189	GLOBAL EQUATES SECTION
12-	664	GLOBAL DATA SECTION
12-	713	- READ ERROR CODE BUFFER
15-	798	GLOBAL TEXT SECTION
17-	837	GLOBAL ERROR REPORT SECTION
19-	891	- MOD U.ERR.ERR - ERROR
23-	994	- MOD U.SFT.ENV - ERROR NUMBER EVALUATION
25-	1048	- MOD U.PRT.PET - PRINT ERROR TYPE
25-	1121	- MOD U.ERR.IDT - GET & PRINT ERROR IDENTIFICATION MESSAGE
29-	1204	- ERROR MESSAGES
29-	1265	- MOD U.SFT.FRU - GET & PRINT FRU'S IDENT
33-	1369	- FRU MESSAGES
35-	1403	- FRU CALLOUT - PRESETUP FOR TESTS
37-	1451	- FRU CALLOUT - PRESETUP FOR ERROR CODE
39-	1474	- MOD U.ERR.PCE - PRINT COMMAND ERROR
41-	1521	- COMMAND ERROR MESSAGE TABLE
43-	1547	- MOD U.ERR.PRE - PRINT REGISTER ERROR
43-	1564	- MOD U.PRT.SCP - PRINT SECTORS
45-	1595	- MOD U.PRT.TKP - PRINT TRACKS
47-	1653	- MOD U.ERR.CLE - CLEAR ERRORS
49-	1676	GLOBAL SUBROUTINES SECTION
49-	1742	- MOD U.1.0 - RANDOM GENERATOR
51-	1840	- MOD U.DEV.INT - INITIALIZE DEVICE
51-	1864	- MOD U.DEV.CLD - CLEAR DEVICE
53-	1885	- MOD U.DEV.FLB - FILL BUFFER
55-	1921	- MOD U.DEV.EMB - EMPTY BUFFER
57-	1957	- MOD U.DEV.WRT - WRITE SUBROUTINE
59-	1993	- MOD U.DEV.RED - READ SUBROUTINE
61-	2028	- MOD U.DEV.SDN - SET DENSITY
63-	2060	- MOD U.DEV.RST - READ STATUS
65-	2087	- MOD U.DEV.REC - READ ERROR CODE
67-	2123	- MOD U.DEV.CMD - SETUP DEVICE COMMAND
67-	2144	- MOD U.DEV.SSC - SETUP SUBSYSTEM COMMANDS
69-	2170	- MOD U.DEV.CEC - DEVICE COMMAND ERROR CHECK
71-	2199	- MOD U.DEV.WAT - WAIT SUBROUTINE
71-	2223	- MOD U.DEV.DRC - DEVICE DONE CHECK
73-	2248	- MOD U.DEV.WCH - WATCH DOG TIMER
75-	2294	- MOD U.DEV.WDN - AWAIT DONE BIT SUBROUTINE
77-	2325	- MOD U.DEV.WTR - AWAIT TRANSFER READY SUBROUTINE
79-	2366	- MOD U.DEV.REG - GET DEVICE REGISTERS
79-	2389	- MOD U.DEV.ITR - INTERRUPT HANDLER
81-	2409	- MOD U.SFT.DPT - SET DATA PATTERN
83-	2511	- MOD U.SFT.GTK - GET TRACK
85-	2554	- MOD U.SFT.GSC - GET SECTOR
87-	2583	- MOD U.SFT.DCK - DATA CHECK
89-	2642	- MOD U.SFT.CDB - CLEAR DATA BUFFER
91-	2655	- MOD U.SFT.RCR - REGISTER CHECK & REPORT
93-	2766	- MOD U.SFT.SRC - SETUP REGISTER CHECK
95-	2808	- MOD U.SFT.BTK - BITS SET/NOT SET CHECK
99-	2909	- PRESETUP REGISTER TABLES
101-	2929	- MOD U.SET.GEN - GET ERROR CODE-ERR #
103-	2975	- MOD U.PRT.STA - PRINT UNIT STATUS

105-	3022	- MOD U.PRT.EC - PRINT UNIT ERROR CODE
107-	3065	- UNIT ERROR CODE MESSAGES
107-	3091	- MOD U.SFT.GEO - GET ERROR CODE OFFSET
107-	3101	- MOD U.SFT.CRS - CLEAR REGISTERS
109-	3113	- MOD U.SFT.DSC - DEVICE STATE CHECK
110-	3152	- MOD U.SFT.DRC - DEVICE READY CHECK
112-	3175	- MOD U.SFT.DDC - DEVICE DENSITY CK
114-	3230	- MOD U.SFT.TKE - TRACK ERROR CHECK
118-	3305	- MOD U.SFT.ECK - ERROR CHECK
122-	3389	- MOD U.SFT.ENC - ERROR NEG TEST CHECK
124-	3423	- MOD U.SFT.DBG - TEST STATUS
124-	3442	- MOD U.SFT.CDC - COMPLIMENT DENSITY CONTROL
124-	3456	- MOD U.SFT.SDC - SETUP DENSITY CONTROL
126-	3471	- MOD U.PRT.UNT - PRINT UNIT IDENT
126-	3489	- MOD U.PRT.DID - PRINT DRIVE IDENT
128-	3519	- MOD U.TST.FTS - FUNCTION TEST SETUP
128-	3537	- MOD U.TST.LTS - LOGIC TEST SETUP
128-	3558	- MOD U.TST.SFG - SETUP TEST FLAGS
130-	3569	- MOD U.SFT.SDC - SETUP DEVICE COMMANDS
130-	3588	- MOD U.TST.CCR - CLEAR TEST CTRS & ERROR REGS
130-	3610	- MOD U.TST.T76 - SET TRACK=76
131-	3633	REPORT CODING SECTION
132-	3690	INITIALIZE SECTION
133-	3779	- MOD 1.1 - UNPACK HARDWARE P-TABLES
133-	3817	- MOD 1.2 - INITIALIZE TABLES
134-	3829	CLEANUP CODING SECTION
134-	3859	DROP UNIT SECTION
134-	3896	AUTO DROP UNIT SECTION
134-	3905	ADD UNIT SECTION
135-	3985	TEST 0 - ADDRESSING TEST
136-	4028	- MOD U.SFT.TRP - BUS TRAP HANDLER
137-	4053	TEST 1 - INITIALIZE - FNC TST
139-	4087	TEST 2 - READ ERROR CODE - FNC TST
141-	4120	TEST 3 - FILL BUFFER - FNC TST
143-	4147	TEST 4 - EMPTY BUFFER - FNC TST
145-	4175	TEST 5 - READ STATUS - FNC TST
147-	4205	TEST 6 - FILL & EMPTY BUFFER - FNC TST
149-	4246	TEST 7 - READ & WRITE SECTOR - FNC TST
153-	4310	TEST 8 - WRITE SECTOR DELETED DATA - FNC TST
155-	4352	TEST 9 - SET DENSITY - FNC TST
159-	4415	TEST 10 - POSITIONING - FNC TST
160-	4456	TEST 11 - CSR BITS - LGC TST
163-	4518	TEST 12 - DBR BITS - LGC TST
166-	4574	TEST 13 - CSR-DBR COMMON BYTE - LGC TST
169-	4633	TEST 14 - BUS INITIALIZE - LGC TST
172-	4689	TEST 15 - PROGRAMMED INITIALIZE - LGC TST
173-	4723	TEST 16 - POWER FAIL - LGC TST
176-	4788	TEST 17 - CONTROLLER-INTERFACE - LGC TST
179-	4837	TEST 18 - NPR - LGC TST
183-	4926	- MOD U.SFT.NAT - ADDRESS NPR ADDRESS TEST
184-	4973	TEST 19 - NPR NON-EXISTENT MEM - LGC TST
187-	5022	TEST 20 - INTERRUPT - LGC TST
188-	5059	TEST 21 - PRIORITY LVL - LGC TST
191-	5135	TEST 22 - INITIALIZE CONTROL - LGC TST
192-	5163	TEST 23 - DATA BUF INTEGRITY - LGC TST
195-	5222	TEST 24 - WRD CNT INTEGRITY - LGC TST
198-	5288	TEST 25 - CONTROLLER-READ*WRITE ELECT - LGC TST

201- 5331	TEST 26 - READ SECTOR-PRT:1 - LGC TST
204- 5385	TEST 27 - POSITIONING - LGC TST
207- 5438	TEST 28 - WRITE SECTOR-PRT:1 - LGC TST
210- 5490	TEST 29 - DELETED DATA WRITE PRT:1 - LGC TST
213- 5545	TEST 30 - SET DENSITY - LGC TST
216- 5630	TEST 31 - SECTOR ADR - LGC TST
219- 5726	TEST 32 - TRACK ADR - LGC TST
222- 5833	TEST 33 - READ SECTOR-PRT:2 - LGC TST
225- 5889	TEST 34 - WRITE SECTOR-PRT:2 - LGC TST
228- 5945	TEST 35 - DELETED DATA WRITE PRT:2 - LGC TST
231- 6004	TEST 36 - DISKETTE & DENSITY DATA CHECK - LGC TST
236- 6116	HARDWARE PARAMETER CODING SECTION
238- 6188	SOFTWARE PARAMETER CODING SECTION
240- 6250	- RX02 FILL BUFFER AREA
240- 6258	- RX02 EMPTY BUFFER AREA
240- 6273	- PATCH AREA


```
.NLIST SEQ,LD,BIN,CND
```

```
7.1  LISTING
```

```
.REPT 0
```

```
.ENDR
```

```
9  
22  
23  
49  
51  
52  
53  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
74  
75  
76  
77  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

```
002000
```

```
002000
```

```
002000
```

```
002000
```

```
002122
```

```
002154
```

```
002162
```

```
.LIST SEQ,BIN  
.TITLE PROGRAM HEADER AND TABLES  
.SBTTL PROGRAM HEADER
```

```
.ENABL ABS,AMA  
.=2000  
.NLIST BEX,MD
```

```
BGNMOD
```

```
:++  
: THE PROGRAM HEADER IS THE INTERFACE BETWEEN  
: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.  
:--
```

```
POINTER BGNSW,BGNSFT,BGNAU,BGNDU,ERRTBL,BGNSETUP
```

```
HEADER CZRXFA0,0,0,170,0  
DESCRIPT <RX02 FUNCTION-LOGIC TEST>
```

```
DEVTYP <RX02>
```

```
.SBTTL DISPATCH TABLE
```

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

```
DISPATCH 36
```

108
109
110
111
112
113
114 002274
115 002276 177170
116 002300 000264
117 002302 000000
118 002304 000000
119 002306 000005
125 002310
126
129
130
131
132
133
134 002310
135 002310 000000
136 002312 177777
137 002314 000004
138 002316
140
141
142
143
144
145
146 002316
147 002320 000001
148 002322 000000
149 002324 000001
150 002326 000000
151 002330 000000
152 002332 000020
153 002334 000000
154 002336 000114
155 002340 000001
156 002342 000032
157 002344 160000
158 002346 000000
165 002350
166
167
168
169
170
171
172
173
174 002350

.SBTTL DEFAULT HARDWARE P-TABLE

++
: THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
: THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
: IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES.
--

BGNHW DFPTBL
.WORD 177170 ;UNIBUS ADDRESS
.WORD 264 ;VECTOR ADDRESS
.WORD 0 ;DRIVE #
.WORD 0 ;FUTURE EXPANSION
.WORD 5 ;BR LEVEL #'S
ENDHW

.SBTTL LOAD DEVICE PROTECTION

++
: LOAD DEVICE PROTECTION TABLE - USED TO CHECK HARDWARE P-TABLE
: AGAINST LOAD DEVICE.
--

BGNPROT
.WORD 0 ;P-TABLE OFFSET->CSR
.WORD -1 ;P-TABLE OFFSET->VECTOR-DON'T CARE
.WORD 4 ;P-TABLE OFFSET->DRIVE
ENDPROT

.SBTTL SOFTWARE P-TABLE

++
: THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
: PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
--

BGNSW SFPTBL
DVTL: .WORD 1 ;HARD ERROR->DEVICE FATAL THRESHOLD LEVEL
.WORD 0 ;CONTROL WORD FOR SOFTWARE P-TABLES
TSTMOD:: .WORD 1 ;TEST MODE
TSTPAT:: .WORD 0 ;TEST PATTERN #
TRKSEQ:: .WORD 0 ;TRACK SEQUENCE #
SWREG:: .WORD 20 ;SOFTWARE SWITCH REG
OD:: .WORD 0 ;OUTSIDE DIA. TRACK LIMIT
ID:: .WORD 114 ;INSIDE DIA. TRACK LIMIT.
MINSEC:: .WORD 1 ;MINIMUM SECTOR LIMIT
MAXSEC:: .WORD 32 ;MAXIMUM SECTOR LIMIT
NXMADR:: .WORD 160000 ;NON-EXISTENT MEMORY-ADR
XADBIT:: .WORD 0 ;EXTENDED ADDRESS BITS
ENDSW

```

:      | 15! 14! 13! 12! 11! 10! 09! 08! 07! 06! 05! 04! 03! 02! 01! 00!
:-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
:SWREG: |   | PRT |   | SID |   |   |   |   |   |   |   | TEN | SEK |   | FUN |
:      |   | STA |   | FLG |   |   |   |   |   |   |   | DAT | CAL |   | TST |
:-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

ENDMOD

188
189
215
225
226
227
273
291
297
326
332
344
416
422
452
458
488 002350
489
490
491
492
493
494
495 002350

.TITLE GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

-----< TEST MACROS >-----
: THIS SECTION CONTAINS MACROS USED THROUGHOUT THE TESTS

BGNMOD

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

EQUALS

: BIT DIFINITIONS

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

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

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

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

: PRIORITY LEVEL DEFINITIONS

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

: OPERATOR FLAG BITS

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


```

497          ;***** PROGRAM EQUIVALENTS *****
498
499          000010 DLDCMD = BIT3      ;DEL. DATA CMD BIT-----<CSR>
500          100000 ERRBIT = BIT15    ;ERROR BIT-----<CSR>
501          040000 RXINIT = BIT14    ;RXINIT BIT-----<CSR>
502          004000 RX2BIT = BIT11    ;RX02 BIT-----<CSR>
503          001000 SIDE1 = BIT9      ;SIDE #1 BIT-----<ESR> & <CSR>
504          000400 DENBIT = BIT8     ;DENSITY BIT-----<CSR>
505          000200 TRBIT = BIT7      ;TR BIT-----<CSR>
506          000040 DNBIT = BIT5      ;DONE BIT-----<CSR>
507          000020 DRV1 = BIT4       ;DRIVE 1-----<CSR>
508          004000 NXMBIT = BIT11    ;NON-EXISTENT MEM-----<ESR>
509          002000 WCOVRF = BIT10    ;WORD COUNT OVERFLOW----<ESR>
510          000400 DRIVE1 = BIT8     ;DRIVE #1 BIT-----<ESR>
511          000200 DRVRDY = BIT7     ;DRIVE READY BIT-----<ESR>
512          000100 DLDBIT = BIT6     ;DEL. DATA BIT-----<ESR>
513          000040 DRVDEN = BIT5     ;DRIVE DENSITY-----<ESR>
514          000020 DENERR = BIT4     ;DENSITY ERROR-----<ESR>
515          000010 ACLOW = BIT3      ;AC LOW BIT-----<ESR>
516          000004 INITDN = BIT2     ;INITIALIZE DONE BIT----<ESR>
517          000002 SIDRDY = BIT1     ;SIDE READY BIT-----<ESR>
518          000001 CRCERB = BIT0     ;CRC ERROR BIT-----<ESR>
519          000004 BTRP4 = 4         ;BUS TRAP LOC #4 - TRAP HANDLER
520          000006 BTRP6 = 6         ;BUS TRAP LOC #4 - PSW
521          000001 LOGICT = BIT0     ;LOGIC TEST BIT-----<SWREG>
522          000002 FUNCTT = BIT1     ;FUNCTION TEST BIT-----<SWREG>
523          010000 SIDFLG = BIT12    ;SIDE FLAG SOFT-P TABLE-<SWREG>
524          000400 ITK = BIT8       ;INITIALIZE TRACKS FLAG      <TKSCFG>
525          001000 ISC = BIT9       ;INITIALIZE SECTORS FLAG    <TKSCFG>
526          000001 STK = BIT0       ;SEQUENCE TRACKS FLAG      <TKSCFG>
527          000002 SSC = BIT1       ;SEQUENCE SECTORS FLAG     <TKSCFG>
528          000000 RTK = 0         ;RANDOM TRACKS FLAG        <TKSCFG>
529          000000 RSC = 0         ;RANDOM SECTORS FLAG       <TKSCFG>
530          000004 ILTK = BIT2     ;ILLEGAL TRACKS FLAG      <TKSCFG>
531
532          ;***** DEVICE COMMANDS *****
533
534          000000 FBCMD = 0         ;FILL BUFFER CMD ;
535          000002 EBCMD = 2         ;EMPTY BUFFER CMD
536          000004 WSCMD = 4         ;WRITE SECTOR
537          000006 RSCMD = 6         ;READ SECTOR
538          000010 SDCMD = 10        ;SET DENSITY
539          000012 STCMD = 12        ;STATUS
540          000014 WDDCMD = 14       ;WRITE DELETED DATA CMD
541          000016 RECCMD = 16      ;READ ERROR CODE CMD
    
```

```

544          ;***** ERROR NUMBER EQUIVALENTS *****
545
546          000002      WRTErr =      2.          :WRITE ERR          -HRD
547          000003      RDERR =      3.          :READ ERR           -HRD
548          000004      CRCERR =     4.          :CRC ERR            -HRD
549          000005      DATERR =     5.          :DATA ERR           -HRD
550          000006      SEKERR =     6.          :SEEK ERR           -HRD
551          000007      DLDERR =     7.          :DELETED DATA ERR -HRD
552          ;-----
553
554          000012      FILERR =    10.          :FILL BUFFER ERR   -HRD
555          000013      EMPERR =    11.          :EMPTY BUFFER ERR  -HRD
556          000014      INTNDN =    12.          :INTERRUPT, NO DONE ERR -HRD
557          000015      DNNINT =    13.          :DONE, NO INTERRUPT ERR -HRD
558          000016      ERRNST =    14.          :ERROR NOT SET ERR -HRD
559          000017      ILLERC =    15.          :ILLEGAL ERROR CODE -HRD
560          000020      DENDSK =    16.          :DENSITY OF DISK-NOT ERR -HRD
561          000021      RECERR =    17.          :READ ERROR CODE ERR -HRD
562          ;-----
563
564          000023      WCERR =     19.          :WORD COUNT ERROR  -DVCFTL*
565          000024      SDRDYE =    20.          :SIDE READY        -DVCFTL
566          000025      DVRDYE =    21.          :DRIVE READY       -DVCFTL
567          000026      SIDWRG =    22.          :SIDE WRONG        -DVCFTL
568          000027      DRVWRG =    23.          :DRIVE WRONG       -DVCFTL
569          000030      DENERR =    24.          :DENSITY ERR       -DVCFTL
570          000031      DENMIX =    25.          :DENSITY MIXED ON DISK ERR -DVCFTL
571          000032      DLDTER =    26.          :DELETED DATA ERR -DVCFTL
572          000033      CSRERR =    27.          :RXCSR-ERR         -DVCFTL
573          000034      DBRERR =    28.          :RXESR-ERR         -DVCFTL
574          000035      STDNER =    29.          :SET DENSITY ERR   -DVCFTL
575          000036      SDKYWD =    30.          :SET DENSITY KEYWORD (VARIFY) -DVCFTL
576          000037      ACLOWD =    31.          :AC LOW            -DVCFTL
577          000040      ALGO2E =    32.          :ALGO2 ERROR       -DVCFTL
578          000041      TRKAER =    33.          :TRACK ADDRESS     -DVCFTL
579          000042      SECAER =    34.          :SECTOR ADDRESS    -DVCFTL
580          ;-----
581
582          000050      ACLOWF =    40.          :AC LOW FATAL ERR  -SYSFTL
583          000051      WCOVFE =    41.          :WORD COUNT OVERFLOW ERR -SYSFTL
584          000052      NXMERR =    42.          :NON-EXISTENT MEMORY ERR -SYSFTL
585          000053      NPRERR =    43.          :NPR LOGIC ERR     -SYSFTL
586          000054      PRILEV =    44.          :PRIORITY LEVEL ERR -SYSFTL
587          000055      DATABF =    45.          :DATA BUFFER INTEG ERR -SYSFTL
588          000056      HDSFDG =    46.          :HARDWARE SELF DIAG ERR -SYSFTL
589          000057      NOTRBT =    47.          :"TR" BIT TIME OUT ERR -SYSFTL
590          000060      NODNBT =    48.          :"DONE" BIT TIBIT TIME OUT ERR -SYSFTL
591          000061      NOITDB =    49.          :NO "INIT DONE" BIT ERR -SYSFTL
592          000062      NOITDP =    50.          :NO PROG "INIT DONE" BIT ERR -SYSFTL
593          000063      DNNOTR =    51.          :"DONE" BIT, NO "TR" BIT -SYSFTL
594          ;-----
    
```



```

597          ;***** FRU CALLOUT MESSAGE EQUIVALENTS *****
598          000000 INTERF =          0          :INTERFACE=0
599          000002 CONTRL =          2*1. :FRUM1   :CONTROLLER
600          000004 RWELEC =          2*2. :FRUM2   :R-W ELECTRONICS
601          000006 PHYDRV =          2*3. :FRUM3   :PHYSICAL DRIVE
602          000010 CABLES =          2*4. :FRUM4   :CABLES
603          000012 POWRSP =          2*5. :FRUM5   :POWER SUPPLY
604          000014 DISKET =          2*6. :FRUM6   :DISKETTE
605          000016 INTFSW =          2*7. :FRUM7   :INTERFACE SWITCHES
606          000020 NPRJPR =          2*8. :FRUM8   :NPR JUMPER
607          000022 CONTSW =          2*9. :FRUM9   :CONTROLLER SWITCHES
608          000024 INTFCB =          2*10. :FRUM10  :INTERFACE CABLE
609          000026 DOOROP =          2*11. :FRUM11  :DOOR OPEN
610          000030 DISKSP =          2*12. :FRUM12  :DISK SPINNING-DRIVE BELT
611          000032 MOTOR  =          2*13. :FRUM13  :MOTOR AC POWER NOT ROTATING
612          000034 NOPWR  =          2*14. :FRUM14  :POWER CORD, BLOWN FUSE, DRIVE POWER
613
614
615

```

```

615          ;***** TEST FLAGS REGISTER EQUIVALENTS (FLAGST), *****
616          000001 REGCK  =          BIT0      :REGISTER CHECK
617          000002 DDCFLG =          BIT1      :DOUBLE DENSITY CONTROL FLAG (DD=1)
618          000004 DATCK  =          BIT2      :DATA CHECK
619          000010 DLPDN  =          BIT3      :DO LOOP DONE
620          000020 EMBUFF =          BIT4      :EMPTY BUFFER-<USED BY DATA CHECK>
621          000040 FUNTST =          BIT5      :FUNCTION TEST FLAG
622          000100 HDRPRT =          BIT6      :ERROR CALL HEADER PRINT
623          000200 RECFLG =          BIT7      :READ ERROR CODE FLAG
624          001000 TRKDON =          BIT9      :TRACK DONE
625          002000 SECDON =          BIT10     :SECTOR DONE
626          004000 NEGST  =          BIT11     :NEGATIVE TEST FLAG
627          010000 ILLGAL =          BIT12     :ILLEGAL FLAG
628          020000 CKERR  =          BIT13     :CHECK ERROR WORDS FLAG
629          040000 HRDERR =          BIT14     :HARD ERROR
630          100000 ERRFLG =          BIT15     :ERROR
631
632
633

```

```

632          ;***** PROGRAM/PRINT FLAGS REGISTER EQUIV (FLAGSP) *****
633
634          000001 TKPRT  =          BIT0      :TRACKS PRINT
635          000002 SCPRT  =          BIT1      :SECTORS PRINT
636          000004 RGPRT  =          BIT2      :REGISTERS PRINT
637          000010 PROPRT =          BIT3      :PROTOCOL LEVEL PRINT
638          000100 HDRPRT =          BIT6      :HEADER PRINT
639          000200 RECTST =          BIT7      :ERROR CODE TEST (INVOKE ERROR CODE)
640          000400 LSIFLG =          BIT8      :LSI FLAG
641          010000 FONZFG =          BIT12     :FONZ FLAG
642          040000 RESFLG =          BIT14     :RESTART FLAG
643          100000 STAFLG =          BIT15     :START FLAG
644
645
646

```

```

646          ;***** "SYS ERR" & "TYP ERR" REGISTER EQUIVALENTS *****
647          000020 CMDERR =          BIT4      :COMMAND ERROR
648          004000 DVFERR =          BIT11     :DEVICE FATAL ERROR
649          002000 SYFERR =          BIT10     :SYSTEM FATAL ERROR
650

```

664
 665
 666
 667
 668
 669
 670
 671
 676
 677 002350 000000
 678 002352 000000
 679 002354 000000
 680 002356 000000
 681 002360 000000
 682 002362 000000
 683 002364 000000
 684 002366 000000
 685 002370 000000
 686 002372 000000
 687 002374 000000
 688 002376 000000
 689
 690 002400 000000
 691 002402 000000
 692 002404 000000
 693 002406 000000
 694 002410 000000
 695 002412 000000
 696 002414 000000
 697 002416 000000
 698 002420 000000
 699
 700 002422 000000
 701 002424 000000
 702 002426 000000
 703 002430 000000
 704 002432 000000
 705 002434 000000
 706 002436 000000
 707 002440 000000
 708
 709
 710
 711
 712
 713
 714
 715 002442 000
 716 002443 000
 717 002444 000
 718 002445 000
 719 002446 000
 720 002447 000
 721 002450 000
 722 002451 000
 723

.SBTTL GLOBAL DATA SECTION

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

--
 : STORAGE FOR DEVICE REGISTERS

```

-----
RXCS:  .WORD  0      ;UNIT BUS ADR-CSR          <UUT *>
RXDB:  .WORD  0      ;UNIT BUS ADR-DBR        <UUT *>
VECT:  .WORD  0      ;UNIT VECTOR             <UUT *>
RXPRI: .WORD  0      ;PRIORITY FOR DEVICE INTERRUPTS <UUT *>
EMPADR: .WORD  0      ;EMPTY BUFFER ADDRESS
FILADR: .WORD  0      ;FILL BUFFER ADDRESS
RECADR: .WORD  0      ;READ ERROR CODE ADDRESS
EXTADR: .WORD  0      ;EXTENDED ADDRESS (BITS: #12 & #13)
WDCNT:  .WORD  0      ;WORD COUNT
VARIFY: .WORD  0      ;VARIFY WORD
TRACK:  .WORD  0      ;TRACK ADR
SECTOR: .WORD  0      ;SECTOR ADR
    
```

DEVICE

PARAMETERS

```

-----
CMD:    .WORD  0      ;COMMAND WORD-TO DEVICE
DELDAT: .WORD  0      ;DELETED DATA FLAG & WORD    <CMD>
INTERT: .WORD  0      ;INTERRUPT WORD              <CMD>
DRIVE:  .WORD  0      ;DRIVE WORD                  <CMD*>
SIDE:   .WORD  0      ;SIDE WORD                   <CMD*>
DENSITY: .WORD  0     ;DENSITY CONTROL WORD       <CMD>
DENSTA: .WORD  0     ;DENSITY STATUS WORD-DRIVE DENSITY
PRIORT: .WORD  0     ;PRIORITY OF INTERRUPT HANDLER-WATCH DOG
DRVOFF: .WORD  0     ;DRIVE BYTE OFFSET
    
```

COMMAND

PARAMETERS

```

-----
ERRCMD: .WORD  0     ;ERROR COMMAND
LCMD:   .WORD  0     ;LAST COMMAND
LRXCSR: .WORD  0     ;LAST RX CSR STORAGE
LRXESR: .WORD  0     ;LAST RX ESR STORAGE
RXCSR:  .WORD  0     ;RX CSR STORAGE
RXESR:  .WORD  0     ;RX ESR STORAGE
REGEXP: .WORD  0     ;REGISTER EXPECTED
REGACT: .WORD  0     ;REGISTER ACTUAL
    
```

DEVICE
 ERROR

&

STATUS
 INFO

* = INFO FROM HARDWARE P-TABLES

.SBTTL - READ ERROR CODE BUFFER

```

-----
XERUUT: .BYTE  0     ;ERROR CODE UUT
WC:     .BYTE  0     ;WORD COUNT UUT
CTK0:   .BYTE  0     ;CUR TRK DRV#0
CTK1:   .BYTE  0     ;CUR TRK DRV#1
TTRK:   .BYTE  0     ;TARGET TRACK
TSEC:   .BYTE  0     ;TARGET SECTOR
SFTSTS: .BYTE  0     ;MICRO CODE SOFT STATUS
BTRK:   .BYTE  0     ;BAD TRACK ADR (ONLY APPLIES IF ERR CODE = 150)
    
```


798
799
800
801
802
803
804
805
806
807
808
809
815
816
817
818
819
826
827

.SBTTL GLOBAL TEXT SECTION

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

:
: NAMES OF DEVICES SUPPORTED BY PROGRAM
:

:
: FORMAT STATEMENTS USED IN PRINT CALLS
:


```
891 .SBTTL - MOD U.ERR.ERR - ERROR
892 -----
893 BGNSUB ERR
894 IF ERR NBR NOT=0 [F]
895 : THEN-SET ERR SAVE = ERR NUMBER
896 : CALL ERROR NUMBER EVALUATION
897 : SETUP ERROR BLOCK CODE ADDRESS
898 : CALL PRINT ERROR TYPE
899 : IF PROGRAM FLAGS-PRT REGS ??? SET [J]
900 : THEN-IF ERRNBR=CSR ERROR [I]
901 : : THEN-CALL PRINT REGS
902 : : ENDF
903 : ENDF
904 : IF COMMAND ERROR SET IN TYPERR [B]
905 : : THEN-CALL PRINT COMMAND ERROR
906 : ENDF
907 : IF FUNCTION TEST NOT SET [A]
908 : : THEN-IF PRINT FLAGS=REGS PRINT [E]
909 : : : THEN-CALL REGISTERS PRINT
910 : : : ENDF
911 : : IF PRINT FLAG=SECTOR PRINT [G]
912 : : : THEN-CALL SECTOR PRINT
913 : : : ENDF
914 : : IF PRINT FLAG=TRACK PRINT [C]
915 : : : THEN-CALL TRACKS PRINT
916 : : : ENDF
917 : : CALL PRINT FRU
918 : : CALL PRINT UNIT STATUS
919 : : ELSE-IF SWITCH REGISTER BIT #14 SET [D]
920 : : : THEN-CALL PRINT UNIT STATUS
921 : : : ENDF
922 : : ENDF
923 : IF ERR SAVE = ERR OLD [K]
924 : : THEN - INCREMENT ERROR CTR
925 : : IF ERROR CTR = 10 ERRORS [L]
926 : : : THEN - SET ABORT = 20
927 : : : ENDF
928 : : ELSE - SET ERR OLD = ERR SAVE
929 : : : CLEAR ERR SAVE
930 : : : CLEAR ERR CTR
931 : : ENDF
932 : CALL CLEAR ERRORS
933 : ENDF
934 ENDSUB
935 -----
```



```

938 003060 000240          ERROR:  NOP
939 003062 005737 002520  IFERR:  TST      ERRNBR      ;IF ERR NBR,
940 003066 001520          BEQ      EFERR      ;NOT=0, THEN
941 003070 013737 002520 003336  MOV     ERRNBR,ERRSAV ;SAVE ERROR NUMBER
942 003076 004737 003344          CALL    ERNBEV      ;CALL ERROR NUMBER EVALUATION
943 003102 012737 003502 002524  MOV     #ERIDNT,ERRBLK ;SETUP ERROR BLOCK CODE ADDRESS
944 003110 004737 003474          CALL    PTERTY      ;CALL PRINT ERROR TYPE
945 003114 032737 000004 002500 1JERR:  BIT     #RGPRT,FLAGSP ;IF PROGRAM FLAG-PRT FLAG
946 003122 001006          BNE     IBERR      ;NOT SET, THEN
947 003124 022737 000033 002520 1IERR:  CMP     #CSRERR,ERRNBR ;IF CSR ERR
948 003132 001002          BNE     IBERR      ;THEN,
949 003134 004737 007564          CALL    PRTREG      ;CALL PRINT REGS
950 003140 032737 000020 002460 1BERR:  BIT     #CMDERR,TYPERR ;IF TYPERR-COMMAND ERROR
951 003146 001402          BEQ     IAERR      ;SET, THEN
952 003150 004737 007040          CALL    PRTCDE      ;CALL PRINT COMMAND ERRORS
953 003154 032737 000040 002476 1AERR:  BIT     #FUNTST,FLAGST ;IF FLAGS-FUNCTION TEST
954 003162 001027          BNE     IDERR      ;NOT SET, THEN
955 003164 032737 000004 002500 1EERR:  BIT     #RGPRT,FLAGSP ;IF PROGRAM FLAGS=REGS PRINT
956 003172 001402          BEQ     IGERR      ;THEN
957 003174 004737 007564          CALL    PRTREG      ;CALL PRINT REGS
958 003200 032737 000002 002500 1GERR:  BIT     #SCPRT,FLAGSP ;IF PROGRAM FLAG=SECTOR
959 003206 001402          BEQ     ICERR      ;BIT SET, THEN
960 003210 004737 007674          CALL    PRTSEC      ;
961 003214 032737 000001 002500 1CERR:  BIT     #TKPRT,FLAGSP ;IF PROGRAM FLAGS=TRK PRINT
962 003222 001402          BEQ     ECERR      ;BIT SET, THEN
963 003224 004737 010002          CALL    PRTRK      ;CALL PRINT TRACKS
964 003230 004737 005404          ECERR:  CALL    PRTFRU    ;CALL PRINT FRU
965 003234 004737 015240          CALL    PRTSTA    ;CALL PRINT UNIT STATUS
966 003240 000406          BR     EAERR      ;BR TO END 'A'
967 003242 032737 040000 002332 1DERR:  BIT     #BIT14,SWREG ;IF SWITCH REG BIT 14
968 003250 001402          BEQ     EAERR      ;SET, THEN
969 003252 004737 015240          CALL    PRTSTA    ;CALL PRINT UNIT STATUS
970 003256 000240          EAERR:  NOP
971 003260 023737 003336 003340 1KERR:  CMP     ERRSAV,ERROLD ;IF SAVED ERR & OLD ERR
972 003266 001011          BNE     LKERR      ;EQUAL, THEN
973 003270 005237 003342          INC     ERRCTR      ;INCREMENT ERR CTR
974 003274 022737 000012 003342 1LERR:  CMP     #10,ERRCTR ;IF 10 ERRS OF SAME KIND
975 003302 012737 000020 002452  MOV     #20,ABORT    ;SET ABORT FLAG
976 003310 000407          BR     EFERR      ;BR TO END 'F'
977 003312 013737 003336 003340 1KERR:  MOV     ERRSAV,ERROLD ;SETUP OLD ERR FOR NEXT CK
978 003320 005037 003336          CLR     ERRSAV      ;CLEAR OUT SAVED ERR
979 003324 005037 003342          CLR     ERRCTR      ;CLEAR ERR CTR
980 003330 004737 010300  EFERR:  CALL    CLRERR    ;CALL CLEAR ERRORS
981 003334 000207  XERROR: RETURN      ;RETURN
982 -----
983 003336 000000  ERRSAV: 0          ;SAVED ERR
984 003340 000000  ERROLD: 0          ;OLD ERR
985 003342 000000  ERRCTR: 0          ;ERR CTR
986 -----
  
```


994
 995
 996
 997
 998
 999
 1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1040
 1041
 1042
 1043
 1044
 1045

```

.SBTTL - MOD U.SFT.ENV - ERROR NUMBER EVALUATION.
-----
BGNSUB
: IF ERR NBR > 39. [A]
: THEN-SET SYSTEM FATAL ERR->ERRTYP
: ELSE
:   IF ERR NBR > 19. [B]
:   : THEN-SET DEVICE FATAL ERR->ERRTYP
:   : ELSE
:   :   IF ERR NBR > 0. [D]
:   :   : THEN-SET HARD ERR->ERRTYP
:   :   :   ENDIF
:   :   ENDIF
:   ENDIF
ENDIF
IF ERRTYP=HARD ERROR [F]
: THEN-
:   IF EVAL SET IN DRS FLAGS [G]
:   : THEN-INCREMENT HARD ERR THRESHOLD LEVEL
:   :   IF HARD ERR THRESHOLD LEVEL=SET LEVEL [H]
:   :   : THEN-SET DEV FATAL ERR->ERRTYP
:   :   ENDIF
:   ENDIF
ENDIF
ENDSUB
-----
ERNBEV: NOP
IAENV:  CMP    #39.,ERRNBR    ; IF ERR NBR > 39.
      BHI     IBENV          ; THEN
      CLR     ERRTYP         ; SET ERRTYP=SYS FTL
      BR      IFENV          ; BR TO IF 'F'
IBENV:  CMP    #19.,ERRNBR    ; IF ERR NBR > 19.
      BHI     IDENV          ; THEN
      MOV     #1,ERRTYP       ; SET ERRTYP=DVC FTL
      BR      IFENV          ; BR TO IF 'F'
IDENV:  TST    ERRNBR         ; IF ERR NBR > 0
      BEQ     IFENV          ; THEN
      MOV     #2,ERRTYP       ; SET ERRTYP=HARD ERR
      IFENV:  CMP    #2,ERRTYP ; IF ERRTYP = HARD ERR
      BNE     EFENV          ; THEN
IGENV:  BIT    #BIT2,FLGDRS   ; IF EVAL IN DRS FLAGS
      BEQ     EFENV          ; SET, THEN
      INC     HETLCT          ; INCREMENT HARD ERR THRESHOLD LEVEL CTR
IHENV:  CMP    DRTL,HETLCT    ; IF DEVICE FTL THRES LVL=SFT LEV
      BHI     EFENV          ; THEN
      MOV     #1,ERRTYP       ; SET ERRTYP=DEV FTL ERR
      CLR     HETLCT          ; CLEAR HARD ERR THRES LVL CTR
EFENV:  NOP
XERNBE: RETURN                ; RETURN
-----
HETLCT: 0                      ; HARD ERROR THRESHOLD LEVEL CTR
-----

```


1048
1051
1052
1053
1054
1055
1056
1057 003474 000240
1058 003476
1059 003500 000207
1060
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134 003502
1135 003502 013701 002520
1136 003506 006301
1137 003510 000240
1138 003512 016101 003534
1139 003516 004737 002550
1140 003522 013701 003534
1141 003526 004737 002550
1142 003532
1143

```
.SBTTL - MOD U.PRT.PET - PRINT ERROR TYPE
-----
:BGNSUB
:CALL ERROR - REVC
:ENDSUB
-----
PTERTY: NOP
        ERROR
        RETURN          ;RETURN
-----
.SBTTL - MOD U.ERR.IDT - GET & PRINT ERROR IDENTIFICATION MESSAGE
-----
:BGMSG      ERIDENT
:LET R1=ERROR #
:DOUBLE R1 FOR ADDRESSING - MESSAGE OFFSET
:LET R1=ERR MSG TABLE ADD + MESSAGE OFFSET
:PRINT BASIC R1
:PUT MESSAGE TABLE ADDRESS IN R1
:PRINT BASIC R1
:ENDMSG
-----
BGMSG      ERIDENT
MOV        ERRNBR,R1      ;GET ERR #
ASL        R1             ;DOUBLE IT FOR ADDRESSING
NOP
MOV        ERMSTB(R1),R1  ;GET ERR MSG ADR FROM TABLE
CALL      PRTBOS          ;CALL PRINT BASIC NO ARG
MOV        ERMSTB,R1     ;GET REST OF ERR MSG FROM TABLE
CALL      PRTBOS          ;CALL PRINT BASIC NO ARG
ENDMSG
-----
```


			ERROR MESSAGE TABLE	ERROR DESCRIPTION	-CLASS
1146			:		
1147			:		
1148	003534	003706	ERMSTB: .WORD ERMS0	:ERROR	
1149	003536	003706	.WORD ERMS0	:ERR#0	
1150	003540	003717	.WORD ERMS2	:ERR#2	:WRITE -HARD
1151	003542	003731	.WORD ERMS3	:ERR#3	:READ -HARD
1152	003544	003742	.WORD ERMS4	:ERR#4	:CRC -HARD
1153	003546	003752	.WORD ERMS5	:ERR#5	:DATA ERR -HARD
1154	003550	003763	.WORD ERMS6	:ERR#6	:SEEK -HARD
1155	003552	003774	.WORD ERMS7	:ERR#7	:DEL DATA -HARD
1156	003554	003706	.WORD ERMS0	:ERR#8	-HARD
1157	003556	003706	.WORD ERMS0	:ERR#9	-HARD
1158	003560	004015	.WORD ERMS10	:ERR#10	:FILL BUFFER -HARD
1159	003562	004035	.WORD ERMS11	:ERR#11	:EMPTY BUFFER -HARD
1160	003564	004056	.WORD ERMS12	:ERR#12	:INTR-NO DONE -HARD
1161	003566	004112	.WORD ERMS13	:ERR#13	:DONE-NO INTR -HARD
1162	003570	004146	.WORD ERMS14	:ERR#14	:ERR-NOT SET -HARD
1163	003572	004177	.WORD ERMS15	:ERR#15	:ILLEG ERR CODE -HARD
1164	003574	004226	.WORD ERMS16	:ERR#16	:DISK DENSITY MIXED OR WRG -HARD
1165	003576	004264	.WORD ERMS17	:ERR#17	:READ ERROR CODE-ERROR WRG -HARD
1166	003600	003706	.WORD ERMS0	:ERR#18	
1167	003602	004314	.WORD ERMS19	:ERR#19	:WORD COUNT
1168	003604	004333	.WORD ERMS20	:ERR#20	:SIDE NOT RDY -DEVFTL
1169	003606	004356	.WORD ERMS21	:ERR#21	:DRIVE NOT RDY -DEVFTL
1170	003610	004402	.WORD ERMS22	:ERR#22	:SIDE RESPONDING WRG -DEVFTL
1171	003612	004434	.WORD ERMS23	:ERR#23	:DRIVE RESPONDING WRG -DEVFTL
1172	003614	004467	.WORD ERMS24	:ERR#24	:DENSITY -DEVFTL
1173	003616	004503	.WORD ERMS25	:ERR#25	:DENSITY DISK -DEVFTL
1174	003620	004532	.WORD ERMS26	:ERR#26	:DEL DATA -DEVFTL
1175	003622	004553	.WORD ERMS27	:ERR#27	:CSR -DEVFTL
1176	003624	004564	.WORD ERMS28	:ERR#28	:DBR -DEVFTL
1177	003626	003706	.WORD ERMS0	:ERR#29	-DEVFTL
1178	003630	004625	.WORD ERMS30	:ERR#30	:SET DENSITY KEYWORD -DEVFTL
1179	003632	004655	.WORD ERMS31	:ERR#31	:AC LOW -DEVFTL
1180	003634	004670	.WORD ERMS32	:ERR#32	:ALGO2 -DEVFTL
1181	003636	004711	.WORD ERMS33	:ERR#33	:TRACK ADDRESS -DEVFTL
1182	003640	004733	.WORD ERMS34	:ERR#34	:SECTOR ADDRESS -DEVFTL
1183	003642	003706	.WORD ERMS0	:ERR#35	
1184	003644	003706	.WORD ERMS0	:ERR#36	
1185	003646	003706	.WORD ERMS0	:ERR#37	
1186	003650	003706	.WORD ERMS0	:ERR#38	
1187	003652	003706	.WORD ERMS0	:ERR#39	
1188	003654	004756	.WORD ERMS40	:ERR#40	:AC LOW FATAL -SYSFTL
1189	003656	004777	.WORD ERMS41	:ERR#41	:WORD COUNT OVERFLOW -SYSFTL
1190	003660	005027	.WORD ERMS42	:ERR#42	:NON-EXISTENT MEM -SYSFTL
1191	003662	005054	.WORD ERMS43	:ERR#43	:NON PROCESSOR REQUEST -SYSFTL
1192	003664	005102	.WORD ERMS44	:ERR#44	:PRIORITY LEVEL -SYSFTL
1193	003666	005125	.WORD ERMS45	:ERR#45	:DATA BUFFER INTEG -SYSFTL
1194	003670	005153	.WORD ERMS46	:ERR#46	:HARDWARE SELF DIAG -SYSFTL
1195	003672	005202	.WORD ERMS47	:ERR#47	: "TR" BIT TIME OUT -SYSFTL
1196	003674	005230	.WORD ERMS48	:ERR#48	: "DONE" BIT TIME OUT -SYSFTL
1197	003676	005260	.WORD ERMS49	:ERR#49	:NO BUS "INIT DONE" -SYSFTL
1198	003700	005307	.WORD ERMS50	:ERR#50	:NO PROG "INIT DONE" -SYSFTL
1199	003702	005337	.WORD ERMS51	:ERR#51	: "DONE" SET->WAITING FOR "TR" BIT -SYSFTL
1200	003704	003706	.WORD ERMS0	:ERR#52	-SYSFTL
1201			:		


```

1204      .SBTTL - ERROR MESSAGES
1205      -----
1206 003706      045      101      040 ERMS0: .ASCIZ /%A ERROR/
1207      :ERMS1: .ASCIZ /%A 1 ?/
1208 003717      045      101      040 ERMS2: .ASCIZ /%A WRITE/
1209 003731      045      101      040 ERMS3: .ASCIZ /%A READ/
1210 003742      045      101      040 ERMS4: .ASCIZ /%A CRC/
1211 003752      045      101      040 ERMS5: .ASCIZ /%A DATA/
1212 003763      045      101      040 ERMS6: .ASCIZ /%A SEEK/
1213 003774      045      101      040 ERMS7: .ASCIZ /%A DELETED DATA/
1214      :ERMS8: .ASCIZ /%A 8 ?/
1215      :ERMS9: .ASCIZ /%A 9 ?/
1216 004015      045      101      040 ERMS10: .ASCIZ /%A FILL BUFFER/
1217 004035      045      101      040 ERMS11: .ASCIZ /%A EMPTY BUEFER/
1218 004056      045      101      040 ERMS12: .ASCIZ /%A INTERRUPT-NO "DONE" BIT/
1219 004112      045      101      040 ERMS13: .ASCIZ /%A "DONE" BIT-NO INTERRUPT/
1220 004146      045      101      040 ERMS14: .ASCIZ /%A ERROR BIT NOT SET-ON/
1221 004177      045      101      040 ERMS15: .ASCIZ /%A ILLEGAL ERROR CODE/
1222 004226      045      101      040 ERMS16: .ASCIZ /%A DISK DENSITY MIXED OR WRG/
1223 004264      045      101      040 ERMS17: .ASCIZ /%A RD ERR CODE-ERR WRG/
1224      :ERMS18: .ASCIZ /%A 18 ?/
1225 004314      045      101      040 ERMS19: .ASCIZ /%A WORD COUNT/
1226 004333      045      101      040 ERMS20: .ASCIZ /%A SIDE NOT READY/
1227 004356      045      101      040 ERMS21: .ASCIZ /%A DRIVE NOT READY/
1228 004402      045      101      040 ERMS22: .ASCIZ /%A WRONG SIDE RESPONDING/
1229 004434      045      101      040 ERMS23: .ASCIZ /%A WRONG DRIVE RESPONDING/
1230 004467      045      101      040 ERMS24: .ASCIZ /%A DENSITY/
1231 004503      045      101      040 ERMS25: .ASCIZ /%A DISK-MIXED DENSITY/
1232 004532      045      101      040 ERMS26: .ASCIZ /%A DELETED DATA/
1233 004553      045      101      040 ERMS27: .ASCIZ /%A CSR-/
1234 004564      045      101      040 ERMS28: .ASCIZ /%A DBR-/
1235 004575      045      101      040 ERMS29: .ASCIZ /%A DENSITY DID NOT SET/
1236 004625      045      101      040 ERMS30: .ASCIZ /%A SET DENSITY KEYWORD/
1237 004655      045      101      040 ERMS31: .ASCIZ /%A AC LOW/
1238 004670      045      101      040 ERMS32: .ASCIZ /%A ALGO2 FAILED/
1239 004711      045      101      040 ERMS33: .ASCIZ /%A TRACK ADDRESS/
1240 004733      045      101      040 ERMS34: .ASCIZ /%A SECTOR ADDRESS/
1241      :ERMS35: .ASCIZ /%A 35 ?/
1242      :ERMS36: .ASCIZ /%A 36 ?/
1243      :ERMS37: .ASCIZ /%A 37 ?/
1244      :ERMS38: .ASCIZ /%A 38 ?/
1245      :ERMS39: .ASCIZ /%A 39 ?/
1246 004756      045      101      040 ERMS40: .ASCIZ /%A AC LOW FATAL/
1247 004777      045      101      040 ERMS41: .ASCIZ /%A WORD COUNT OVERFLOW/
1248 005027      045      101      040 ERMS42: .ASCIZ /%A NON-EXISTENT MEM/
1249 005054      045      101      040 ERMS43: .ASCIZ /%A NON-PROCESSOR REQ/
1250 005102      045      101      040 ERMS44: .ASCIZ /%A PRIORITY LEVEL/
1251 005125      045      101      040 ERMS45: .ASCIZ /%A DATA BUFFER INTEG/
1252 005153      045      101      040 ERMS46: .ASCIZ /%A HARDWARE SELF DIAG/
1253 005202      045      101      040 ERMS47: .ASCIZ /%A "TR" BIT TIME OUT/
1254 005230      045      101      040 ERMS48: .ASCIZ /%A "DONE" BIT TIME OUT/
1255 005260      045      101      040 ERMS49: .ASCIZ /%A NO BUS "INIT DONE"/
1256 005307      045      101      040 ERMS50: .ASCIZ /%A NO PROG "INIT DONE"/
1257 005337      045      101      040 ERMS51: .ASCIZ /%A "DONE" SET->WAITING FOR "TR" BIT/
1258      :ERMS52: .ASCIZ /%A 52 ?/
1259      .EVEN      :800. BYTES-->680.
1260
  
```



```

1306 005404 000240 PRTFRU: NOP ;
1307 005406 005737 002454 IAFRU: TST FIN ;IF FINI FLAG
1308 005412 001404 BEQ IBFRU ;SET, THEN
1309 005414 012737 006640 005610 MOV #TOFT0,FRUTAD ;SET FRU TBL ADR=FINI FRU TBL ADR
1310 005422 000431 BR EAFRU ;BR TO WHILE 'E'
1311 005424 105737 002442 IBFRU: TSTB XERUUT ;IF ERROR CODE
1312 005430 001412 BEQ LBFRU ;NOT=0, AND
1313 005432 032737 000200 002476 BIT #RECFLG,FLAGST ;FLAGS-READ ERROR CODE BIT
1314 005440 001406 BEQ LBFRU ;SET, THEN
1315 005442 004737 017106 CALL GTECOF ;CALL GET ERROR CODE OFFSET
1316 005446 016137 006660 005610 MOV TOFTB(R1),FRUTAD ;GET ERROR CODE FRU TABLE ADDRESS
1317 005454 000414 BR EAFRU ;BR TO WHILE 'E'
1318 005456 013701 002466 LBFRU: MOV TSTID,R1 ;GET TEST TABLE ADDRESS
1319 005462 005721 WCFRU: TST (R1)+ ;DO WHILE TABLE ENTRY NOT=-1
1320 005464 100376 BPL WCFRU ;ADVANCE TABLE ADDRESS
1321 005466 005721 WDFRU: TST (R1)+ ;DO WHILE TABLE ENTRY NOT=-1
1322 005470 100376 BPL WDFRU ;ADVANCE TABLE ADDRESS
1323 005472 013702 002470 MOV TCMDCT,R2 ;GET TEST COMMAND CTR
1324 005476 006302 ASL R2 ;DOUBLE IT
1325 005500 060201 ADD R2,R1 ;SETUP FRU TABLE ADDRESS
1326 005502 011137 005610 MOV (R1),FRUTAD ;SET FRU TABLE ADR=ABOVE ADDRESS
1327 005506 012701 005650 EAFRU: MOV #FRUM00,R1 ;SET FRU MSG HEADER
1328 005512 004737 002550 CALL PRTBOS ;CALL PRINT BASIC-NO ARG
1329 005516 105777 000066 WEFRU: TSTB @FRUTAD ;DO WHILE TABLE ENTRY
1330 005522 100430 BMI EEFRU ;NOT=-1
1331 005524 105777 000060 IFFRU: TSTB @FRUTAD ;IF TABLE ENTRY
1332 005530 001014 BNE LFFRU ;EQUALS 0, THEN
1333 005532 032737 000400 002500 IGFRU: BIT #LSIFLG,FLAGSP ;IF LSI FLAG BIT-PROGRAM FLAGS
1334 005540 001403 BEQ LGFRU ;SET, THEN
1335 005542 012701 005713 MOV #FRUM0A,R1 ;SET LSI INTERFACE MSG
1336 005546 000402 BR EGFRU ;BR TO END 'G'
1337 005550 012701 005746 LGFRU: MOV #FRUM0B,R1 ;SET UNIBUS INTERFACE MSG
1338 005554 004737 002550 EGFRU: CALL PRTBOS ;CALL PRINT BASIC-NO ARG
1339 005560 000406 BR EFRU ;BR TO END 'G'
1340 005562 117701 000022 LFFRU: MOV @FRUTAD,R1 ;SETUP PRINT FRU MSG OFFSET FROM TABLE
1341 005566 016101 005612 MOV FRUTBM(R1),R1 ;SET FRU MSG ADR FROM TABLE
1342 005572 004737 002550 CALL PRTBOS ;CALL PRINT BASIC-NO ARG
1343 005576 005237 005610 EFRU: INC FRUTAD ;INCREMENT FRU TABLE ADDRESS
1344 005602 000745 BR WEFRU ;END DO 'E'
1345 005604 000240 EEFRU: NOP ;
1346 005606 000207 XPTFRU: RETURN ;RETURN
1347 -----
1348 005610 000000 FRUTAD: 0 ;FRU TABLE ADDRESS
1349 -----

```

1352	005612	000000	FRUTBM:	.WORD	0
1353	005614	006001		.WORD	FRUM1
1354	005616	006035		.WORD	FRUM2
1355	005620	006070		.WORD	FRUM3
1356	005622	006120		.WORD	FRUM4
1357	005624	006140		.WORD	FRUM5
1358	005626	006173		.WORD	FRUM6
1359	005630	006221		.WORD	FRUM7
1360	005632	006255		.WORD	FRUM8
1361	005634	006324		.WORD	FRUM9
1362	005636	006361		.WORD	FRUM10
1363	005640	006426		.WORD	FRUM11
1364	005642	006451		.WORD	FRUM12
1365	005644	006504		.WORD	FRUM13
1366	005646	006544		.WORD	FRUM14

1367
 1368
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1400

.SBTTL - FRU MESSAGES

045	116	045	FRUM00:	.ASCIZ	/%N%A	POSSIBLE FAILING 'FRU'S': %N/
045	123	061	FRUM0A:	.ASCIZ	/%S11%A	INTERFACE - M8029%N/
045	123	061	FRUM0B:	.ASCIZ	/%S11%A	INTERFACE - M8256%N/
045	123	061	FRUM1:	.ASCIZ	/%S11%A	CONTROLLER - M7744%N/
045	123	061	FRUM2:	.ASCIZ	/%S11%A	R-W ELECT - M7745%N/
045	123	061	FRUM3:	.ASCIZ	/%S11%A	PHYSICAL DRIVE%N/
045	123	061	FRUM4:	.ASCIZ	/%S11%A	CABLES%N/
045	123	061	FRUM5:	.ASCIZ	/%S11%A	POWER SUPPLY-H771%N/
045	123	061	FRUM6:	.ASCIZ	/%S11%A	BAD DISKETTE%N/
045	123	061	FRUM7:	.ASCIZ	/%S11%A	INTERFACE SWITCHES%N/
045	123	061	FRUM8:	.ASCIZ	/%S11%A	NPR JUMPER - PDP-11 BACKPLANE%N/
045	123	061	FRUM9:	.ASCIZ	/%S11%A	CONTROLLER SWITCHES%N/
045	123	061	FRUM10:	.ASCIZ	/%S11%A	INTERFACE->CONTROLLER CABLE%N/
045	123	061	FRUM11:	.ASCIZ	/%S11%A	DOOR OPEN%N/
045	123	061	FRUM12:	.ASCIZ	/%S11%A	BROKEN DRIVE BELT%N/
045	123	061	FRUM13:	.ASCIZ	/%S11%A	DRIVE MOTOR - AC POWER%N/
045	123	061	FRUM14:	.ASCIZ	/%S11%A	POWER CORD, BLOWN FUSE, DRIVE POWER, POWER SUPPLY %N/

.EVEN . :506. BYTES


```

1403      .SBTTL - FRU CALLOUT - PRESETUP FOR TESTS
1404      ;-----
1405      TN=0
1406      INFCTL=TOFT0      ;INTERFACE & CONTROLLER
1407      FRUTB 0,INTERF,CONTRL,INTFCB
           TOFT0: .BYTE INTERF
           .BYTE CONTRL
           .BYTE INTFCB
           .BYTE -1
1408      ;-----
1409      INTONL=TOFT40     ;INTERFACE ONLY
1410      FRUTB 40,INTERF
           TOFT40: .BYTE INTERF
           .BYTE -1
1411      ;-----
1412      CTLINF=TOFT41    ;CONTROLLER & INTERFACE
1413      FRUTB 41,CONTRL,INTERF
           TOFT41: .BYTE CONTRL
           .BYTE INTERF
           .BYTE -1
1414      ;-----
1415      CTLRWE=TOFT42    ;
1416      FRUTB 42,CONTRL,RWELEC
           TOFT42: .BYTE CONTRL
           .BYTE RWELEC
           .BYTE -1
1417      ;-----
1418      CTLONL=TOFT43    ;
1419      FRUTB 43,CONTRL,INTFCB
           TOFT43: .BYTE CONTRL
           .BYTE INTFCB
           .BYTE -1
1420      ;-----
1421      .EVEN
    
```

1424
 1425
 1426 006660 000000
 1427 006662 006736
 1428 006664 006742
 1429 006666 000000
 1430 006670 006746
 1431 006672 006751
 1432 006674 000000
 1433 006676 006755
 1434 006700 000000
 1435 006702 006763
 1436 006704 006770
 1437 006706 006776
 1438 006710 000000
 1439 006712 007002
 1440 006714 007006
 1441 006716 007012
 1442 006720 007016
 1443 006722 000000
 1444 006724 007022
 1445 006726 007025
 1446 006730 007030
 1447 006732 007035
 1448 006734 177777
 1449
 1450
 1451
 1452
 1453 000000
 1454 006736
 006736 006
 006737 002
 006740 004
 006741 377
 1455 006742
 006742 006
 006743 002
 006744 004
 006745 377
 1456 006746
 006746 000
 006747 002
 006750 377
 1457 006751
 006751 004
 006752 006
 006753 002
 006754 377
 1458 006755
 006755 014
 006756 004
 006757 006
 006760 002
 006761 000
 006762 377
 1459 006763

ERROR CODE - FRU CALLOUT ADDRESS TABLE

```

:-----
:
:-----
TOFTB: .WORD 0
        .WORD TOFT1
        .WORD TOFT2
        .WORD 0
        .WORD TOFT4
        .WORD TOFT5
        .WORD 0
        .WORD TOFT7
        .WORD 0
        .WORD TOFT11
        .WORD TOFT12
        .WORD TOFT13
        .WORD 0
        .WORD TOFT15
        .WORD TOFT16
        .WORD TOFT17
        .WORD TOFT20
        .WORD 0
        .WORD TOFT22
        .WORD TOFT23
        .WORD TOFT24
        .WORD TOFT25
        .WORD -1
:-----

```

.SBTTL - FRU CALLOUT - PRESETUP FOR ERROR CODE

```

:-----
:
:-----
TN=0
FRUTB 1,PHYDRV,CONTRL,RWELEC
TOFT1: .BYTE PHYDRV
        .BYTE CONTRL
        .BYTE RWELEC
        .BYTE -1
FRUTB 2,PHYDRV,CONTRL,RWELEC
TOFT2: .BYTE PHYDRV
        .BYTE CONTRL
        .BYTE RWELEC
        .BYTE -1
FRUTB 4,INTERF,CONTRL
TOFT4: .BYTE INTERF
        .BYTE CONTRL
        .BYTE -1
FRUTB 5,RWELEC,PHYDRV,CONTRL
TOFT5: .BYTE RWELEC
        .BYTE PHYDRV
        .BYTE CONTRL
        .BYTE -1
FRUTB 7,DISKET,RWELEC,PHYDRV,CONTRL,INTERF
TOFT7: .BYTE DISKET
        .BYTE RWELEC
        .BYTE PHYDRV
        .BYTE CONTRL
        .BYTE INTERF
        .BYTE -1
FRUTB 11,DISKET,RWELEC,PHYDRV,CONTRL

```


	006763	014		TOFT11:	.BYTE	DISKET
	006764	004			.BYTE	RWELEC
	006765	006			.BYTE	PHYDRV
	006766	002			.BYTE	CONTRL
	006767	377			.BYTE	-1
1460	006770		FRUTB	12,DISKET,RWELEC,PHYDRV,CONTRL,POWRSP		
	006770	014		TOFT12:	.BYTE	DISKET
	006771	004			.BYTE	RWELEC
	006772	006			.BYTE	PHYDRV
	006773	002			.BYTE	CONTRL
	006774	012			.BYTE	POWRSP
	006775	377			.BYTE	-1
1461	006776		FRUTB	13,DISKET,RWELEC,CONTRL		
	006776	014		TOFT13:	.BYTE	DISKET
	006777	004			.BYTE	RWELEC
	007000	002			.BYTE	CONTRL
	007001	377			.BYTE	-1
1462	007002		FRUTB	15,RWELEC,PHYDRV,CONTRL		
	007002	004		TOFT15:	.BYTE	RWELEC
	007003	006			.BYTE	PHYDRV
	007004	002			.BYTE	CONTRL
	007005	377			.BYTE	-1
1463	007006		FRUTB	16,RWELEC,PHYDRV,CONTRL		
	007006	004		TOFT16:	.BYTE	RWELEC
	007007	006			.BYTE	PHYDRV
	007010	002			.BYTE	CONTRL
	007011	377			.BYTE	-1
1464	007012		FRUTB	17,DISKET,RWELEC,CONTRL		
	007012	014		TOFT17:	.BYTE	DISKET
	007013	004			.BYTE	RWELEC
	007014	002			.BYTE	CONTRL
	007015	377			.BYTE	-1
1465	007016		FRUTB	20,DISKET,RWELEC,CONTRL		
	007016	014		TOFT20:	.BYTE	DISKET
	007017	004			.BYTE	RWELEC
	007020	002			.BYTE	CONTRL
	007021	377			.BYTE	-1
1466	007022		FRUTB	22,RWELEC,CONTRL		
	007022	004		TOFT22:	.BYTE	RWELEC
	007023	002			.BYTE	CONTRL
	007024	377			.BYTE	-1
1467	007025		FRUTB	23,INTERF,CONTRL		
	007025	000		TOFT23:	.BYTE	INTERF
	007026	002			.BYTE	CONTRL
	007027	377			.BYTE	-1
1468	007030		FRUTB	24,DISKET,CONTRL,INTERF,RWELEC		
	007030	014		TOFT24:	.BYTE	DISKET
	007031	002			.BYTE	CONTRL
	007032	000			.BYTE	INTERF
	007033	004			.BYTE	RWELEC
	007034	377			.BYTE	-1
1469	007035		FRUTB	25,INTERF,CONTRL		
	007035	000		TOFT25:	.BYTE	INTERF
	007036	002			.BYTE	CONTRL
	007037	377			.BYTE	-1

1470
1471

: .EVEN

1474
 1475
 1476
 1477
 1478
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1490
 1491
 1492
 1493
 1494
 1495 007040 000240
 1496 007042 012701 007204
 1497 007046 004737 002550
 1498 007052 013702 002422
 1499 007056 032702 040000
 1500 007062 001405
 1501 007064 012701 007452
 1502 007070 012702 000012
 1503 007074 000404
 1504 007076 042702 177761
 1505 007102 016201 007164
 1506 007106 004737 002550
 1507 007112 012701 007216
 1508 007116 004737 002550
 1509 007122 022702 000012
 1510 007126 001415
 1511 007130 032737 000010 002500
 1512 007136 001411
 1513 007140 013702 002472
 1514 007144 012701 007502
 1515 007150 004737 002734
 1516 007154 042737 000020 002460
 1517 007162 000207
 1518

```

.SBTTL - MOD U.ERR.PCE - PRINT COMMAND ERROR
-----
: BGNSUB
:   SETUP & PRINT COMMAND FORMAT MESSAGE
:   GET COMMAND
:   IF INITIALIZE COMMAND
:   :   THEN-SETUP INIT COMMAND MSG
:   :   ELSE-CLEAR TOP BITS & GO BIT
:   :   CLEAR TOP BITS & GO BIT
:   :   GET COMMAND MSG ADDRESS FROM TABLE (INDEXED BY COMMAND)
:   ENDF
:   CALL PRINT
:   SETUP & PRINT END OF COMMAND ERROR
:   IF PROTOCOL TYPE COMMAND
:   :   THEN-IF PRINT FLAGS=PRINT PROTOCOL SET
:   :   :   THEN-SETUP & PRINT PROTOCOL ERR
:   :   ENDF
:   ENDF
: ENDSUB
-----

PRTCDE: NOP      ;
: MOV      #CMFTMS,R1      ;SETUP COMMAND FORMAT MESSAGE
: CALL    PRTBOS          ;CALL PRINT BASIC-NO ARG
: MOV      ERRCMD,R2      ;GET COMMAND
IAPCE:  BIT      #BIT14,R2  ;IF INITIALIZE BIT
: BEQ     LAPCE           ;SET, THEN
: MOV      #CMDM8,R1      ;SET PROGRAMMED INIT MSG
: MOV      #12,R2         ;SET R2 TO SHOW COMMAND WITH NO PROTOCOL
: BR      EAPCE          ;BR TO END 'A'
LAPCE:  BIC      #177761,R2 ;CLEAR TOP BITS & GO BIT
: MOV      CMDMSG(R2),R1  ;GET COMMAND MSG ADR FROM TABLE
EAPCE:  CALL    PRTBOS          ;CALL PRINT BASIC-NO ARG
: MOV      #CMERMS,R1     ;SETUP "COMMAND ERR" MSG
: CALL    PRTBOS          ;CALL PRINT BASIC-NO ARG
IBPCE:  CMP      #12,R2     ;IF R2 CONTAINS PROTOCOL TYPE COMMAND
: BEQ     XPCE           ;THEN
: BIT      #PROPRT,FLAGSP ;IF PRINT PROTOCOL FLAG=FLAGSP
: BEQ     XPCE           ;SET, THEN
: MOV      PROTCT,R2     ;SETUP PRINT PROTOCOL CNT
: MOV      #CMDPE,R1     ;SETUP PRINT PROTOCOL ERR MSG
: CALL    PRTX1S         ;PRINT MSG
: BIC     #CMDERR,TYPERR  ;CLEAR TYP ERR COMMAND ERROR
XPCE:  RETURN          ;RETURN
-----

```


				.SBTTL - COMMAND ERROR MESSAGE TABLE	
1521					
1522					
1523	007164	007237			CMDMSG: .WORD CMDM0
1524	007166	007255			.WORD CMDM1
1525	007170	007274			.WORD CMDM2
1526	007172	007313			.WORD CMDM3
1527	007174	007331			.WORD CMDM4
1528	007176	007347			.WORD CMDM5
1529	007200	007374			.WORD CMDM6
1530	007202	007430			.WORD CMDM7
1531	007204	045	116	045	CMFTMS: .ASCIZ /%N%S6%A->/
1532	007216	045	101	040	CMERMS: .ASCIZ /%A COMMAND ERROR/
1533	007237	045	101	106	CMDM0: .ASCIZ /%AFILL BUFFER/
1534	007255	045	101	105	CMDM1: .ASCIZ /%AEMPTY BUFFER/
1535	007274	045	101	127	CMDM2: .ASCIZ /%AWRITE SECTOR/
1536	007313	045	101	122	CMDM3: .ASCIZ /%AREAD SECTOR/
1537	007331	045	101	123	CMDM4: .ASCIZ /%ASET DENSITY/
1538	007347	045	101	122	CMDM5: .ASCIZ /%AREAD MAINT. STATUS/
1539	007374	045	101	127	CMDM6: .ASCIZ /%AWRITE SECTOR-DELETED DATA/
1540	007430	045	101	122	CMDM7: .ASCIZ /%AREAD ERROR CODE/
1541	007452	045	101	120	CMDM8: .ASCIZ /%APROGRAMMED INITIALIZE/
1542	007502	045	116	045	CMDPE: .ASCIZ /%N%S8%APROTOCOL FAILED-WAITING TO PASS WORD #%01/
1543					.EVEN
1544					

```

1547      .SBTTL - MOD U.ERR.PRE - PRINT REGISTER ERROR
1548      :-----:
1549
1550 007564 000240      PRTREG: NOP      ;
1551 007566 012701 007620      MOV      #PRTGMS,R1      ;SETUP REGISTER MESSAGE
1552 007572 013702 002440      MOV      REGACT,R2      ;SETUP REG ACTUAL
1553 007576 013703 002436      MOV      REGEXP,R3      ;SETUP REG EXPECTED
1554 007602 004737 002612      CALL     PRTB2S      ;CALL PRINT BASIC-2 ARG
1555 007606 005037 002440      CLR      REGACT      ;CLEAR OLD RESULTS
1556 007612 005037 002436      CLR      REGEXP      ;CLEAR OLD RESULTS
1557 007616 000207      RETURN     ;RETURN
1558      :-----:
1559 007620      045      116      045  PRTGMS: .ASCII /%N%S6%AREG ACTUAL=%0%N/
1560 007646      045      123      066  .ASCIZ  /%S6%AREG EXPECT=%0%N/
1561      .EVEN
1562      :-----:
1563
1564      .SBTTL - MOD U.PRT.SCP - PRINT SECTORS
1565      :-----:
1566      : BGNSUB
1567      : IF READ ERROR CODE FLAG SET
1568      :   THEN-SETUP PRINT EXPECTED SECTOR
1569      :   SETUP PRINT DEVICE SECTOR
1570      :   CALL PRINT
1571      : ENDIF
1572      : ENDSUB
1573      :-----:
1574
1575 007674 000240      PRTSEC: NOP      ;
1576 007676 032737 000200 002476  IASCP: BIT      #RECFLG,FLAGST ;IF READ ERROR CODE FLAG
1577 007704 001424      BEQ      XSCP      ;SET, THEN
1578 007706 013702 002376      MOV      SECTOR,R2      ;SETUP EXPECTED SECTOR
1579 007712 012701 010156      MOV      #EXMS,R1      ;SETUP EXPECTED MSG
1580 007716 004737 002550      CALL     PRTB0S      ;CALL PRINT BASIC-0 ARG
1581 007722 012701 007760      MOV      #ADSCMS,R1     ;SETUP SECTOR MSG
1582 007726 004737 002570      CALL     PRTB1S      ;CALL PRINT BASIC-1 ARG
1583 007732 113702 002447      MOVB     TSEC,R2      ;SETUP DEVICE SECTOR
1584 007736 012701 010220      MOV      #TGMS,R1      ;SETUP TARGET MSG
1585 007742 004737 002550      CALL     PRTB0S      ;CALL PRINT BASIC-0 ARG
1586 007746 012701 007760      MOV      #ADSCMS,R1     ;SETUP SECTOR MSG
1587 007752 004737 002570      CALL     PRTB1S      ;CALL PRINT BASIC-1 ARG
1588 007756 000207      XSCP:  RETURN     ;RETURN
1589      :-----:
1590 007760      045      101      040  ADSCMS: .ASCIZ  /%A SECTOR=%D2%A./
1591      .EVEN
1592      :-----:

```


1595
 1596
 1597
 1598
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1610
 1611
 1612
 1613
 1614 010002 004737 020600
 1615 010006 032737 000200 002476
 1616 010014 001445
 1617 010016 013702 002374
 1618 010022 012701 010156
 1619 010026 004737 002550
 1620 010032 012701 010257
 1621 010036 004737 002570
 1622 010042 005737 002406
 1623 010046 001403
 1624 010050 113702 002445
 1625 010054 000402
 1626 010056 113702 002444
 1627 010062 012701 010177
 1628 010066 004737 002550
 1629 010072 012701 010257
 1630 010076 004737 002570
 1631 010102 113702 002446
 1632 010106 012701 010220
 1633 010112 004737 002550
 1634 010116 012701 010257
 1635 010122 004737 002570
 1636 010126 000412
 1637 010130 013702 002374
 1638 010134 012701 010241
 1639 010140 004737 002550
 1640 010144 012701 010257
 1641 010150 004737 002570
 1642 010154 000207
 1643
 1644 010156 045 116 045
 1645 010177 045 116 045
 1646 010220 045 116 045
 1647 010241 045 116 045
 1648 010257 045 101 040
 1649
 1650

```

.SBTTL - MOD U.PRT.TKP - PRINT TRACKS
-----
BGNSUB
CALL PRINT UNIT IDENT
IF READ ERROR CODE FLAG SET
: THEN-SETUP PRINT EXPECTED TRACK
: CALL PRINT 1 PARAMETER
: IF DRIVE #1 SELECTED
: THEN-SETUP CURRENT TRACK DRV1-PRINT
: ELSE-SETUP CURRENT TRACK DRVO-PRINT
: ENDF
: CALL PRINT 1 PARAMETER
: SETUP PRINT DRIVE TARGET TRACK
: CALL PRINT 1 PARAMETER
: ELSE-SETUP PRINT ERROR ON TRACK
: CALL PRINT 1 PARAMETER
: ENDF
ENDSUB
-----
PRTRK: CALL PRTDID ;CALL PRINT DRIVE IDENT
IATKP: BIT #RECFLG,FLAGST ;IF READ ERROR CODE FLAG
BEQ LATKP ;FLAG SET, THEN
MOV TRACK,R2 ;SETUP EXPECTED TRACK
MOV #EXMS,R1 ;SETUP EXPECTED MSG
CALL PRTBOS ;CALL PRINT BASIC-0 ARG
MOV #ADTKMS,R1 ;SETUP PRINT TRACK
CALL PRTB1S ;CALL PRINT BASIC-1 PAR.
IBTKP: TST DRIVE ;IF DRIVE #1
BEQ LBTKP ;SELECTED, THEN
MOVB CTK1,R2 ;SETUP CUR TRK D1-PRT
BR EBTKP ;BR TO END 'B'
LBTKP: MOVB CTK0,R2 ;SETUP CUR TRK D0-PRT
EBTKP: MOV #CDMS,R1 ;SETUP DRIVE CURRENT MSG
CALL PRTBOS ;CALL PRINT BASIC-0 ARG
MOV #ADTKMS,R1 ;SETUP PRINT TRACK
CALL PRTB1S ;CALL PRINT BASIC-1 PAR.
MOV TTRK,R2 ;SETUP TARGET TRACK
MOV #TGMS,R1 ;SETUP TARGET MSG
CALL PRTBOS ;CALL PRINT BASIC-0 ARG
MOV #ADTKMS,R1 ;SETUP PRINT TRACK
CALL PRTB1S ;CALL PRINT BASIC-1 PAR.
BR XTKPRT ;BR TO EXIT
LATKP: MOV TRACK,R2 ;SETUP ERROR TRACK
MOV #ERTKMS,R1 ;SETUP ERROR TRACK MSG
CALL PRTBOS ;CALL PRINT BASIC-0 ARG
MOV #ADTKMS,R1 ;SETUP PRINT TRACK
CALL PRTB1S ;CALL PRINT BASIC-1 PAR.
XTKPRT: RETURN
:
-----
EXMS: .ASCIZ /%N%S6% EXPECTED/
CDMS: .ASCIZ /%N%S6% CUR DRV/
TGMS: .ASCIZ /%N%S6% TARGET/
ERTKMS: .ASCIZ /%N% ERROR ->/
ADTKMS: .ASCIZ /%A TRACK=%D2%. /
.EVEN
-----

```

1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664 010300 000240
1665 010302 005037 002442
1666 010306 005037 002520
1667 010312 005037 002516
1668 010316 012737 003502 002524
1669 010324 005037 002454
1670 010330 005037 002452
1671 010334 042737 100000 002476
1672 010342 000207
1673

.SBTTL - MOD U.ERR.CLE - CLEAR ERRORS

: BGNSUB
: CLEAR ERROR NUMBER
: CLEAR ERROR TYPE
: CLEAR ERROR BLOCK
: CLEAR FIN
: CLEAR ABORT
: ENDSUB

CLRERR: NOP
CLR XERUUT : CLEAR READ ERR CODE WORD
CLR ERNBR : CLEAR ERROR NUMBER
CLR ERRTP : CLEAR ERROR TYPE
MOV #ERIDNT,ERRBLK : CLEAR ERROR BLOCK
CLR FIN : CLEAR FINI
CLR ABORT : CLEAR ABORT
BIC #ERRFLG,FLAGST : CLEAR FLAGST ERR FLAG
RETURN : RETURN

1676
 1677
 1678
 1679
 1680
 1681
 1682
 1742
 1743
 1744
 1745
 1752
 1758
 1765
 1771
 1778
 1787
 1795
 1801
 1802
 1809
 1815
 1816 010344 012700 000001
 1817 010350 063700 010432
 1818 010354 063700 010434
 1819 010360 042700 170000
 1820 010364 000241
 1821 010366 006100
 1822 010370 006100
 1823 010372 010037 010432
 1824 010376 005000
 1825 010400 013700 010434
 1826 010404 006000
 1827 010406 006000
 1828 010410 063700 010432
 1829 010414 042700 170000
 1830 010420 010037 010434
 1831 010424 010037 010436
 1832 010430 000207
 1833
 1834 010432 000000
 1835 010434 000000
 1836 010436 000000
 1837

.SBTTL GLOBAL SUBROUTINES SECTION

```

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

.SBTTL - MOD U.1.0 - RANDOM GENERATOR

```

:++
: FUNCTIONAL DESCRIPTION:
: SUBROUTINE TO GENERATE A RANDOM NUMBER
: INPUTS: NONE
: IMPLICIT INPUTS: NONE
: OUTPUTS: RANUM
: IMPLICIT OUTPUTS: NONE
: SUBORDINATE ROUTINES USED: NONE
: FUNCTIONAL SIDE EFFECTS: NONE
: CALLING SEQUENCE: SUB
:--
    
```

```

:----- RANDOM GENERATOR -----
RANGEN: MOV    #1,R0
        ADD    RAN1,R0
        ADD    RAN2,R0
        BIC    #170000,R0
        CLC
        ROL    R0
        ROL    R0
        MOV    R0,RAN1
        CLR    R0
        MOV    RAN2,R0
        ROR    R0
        ROR    R0
        ADD    RAN1,R0
        BIC    #170000,R0
        MOV    R0,RAN2
        MOV    R0,RANUM
        RTS    PC
:-----
RAN1:   0
RAN2:   0
RANUM:  0
:-----
    
```

```
1840 .SBTTL - MOD U.DEV.INT - INITIALIZE DEVICE
1841 :++
1842 : FUNCTIONAL DESCRIPTION: SUBR TO SEND INITIALIZE TO DEVICE.-ERROR CK
1843 : INPUTS: NONE
1844 : IMPLICIT INPUTS: ERROR BIT
1845 : OUTPUTS: DEVICE INITIALIZE
1846 : IMPLICIT OUTPUTS:
1847 : SUBORDINATE ROUTINES USED: COMMAND ERR CK, GET DEV. REGS, WAIT DONE
1848 : FUNCTIONAL SIDE EFFECTS:
1849 : CALLING SEQUENCE: SUBR
1850 :--
```

```
1851 -----
1852 :
1853 :
1854 010440 012737 040000 002400 INTIAL: MOV #40000,CMD ;SET INT COMMAND
1855 010446 013777 002400 171674 MOV CMD,@RXCS ;INIT UNIT 0
1856 010454 004737 012032 CALL AWDN ;GO AWAIT DONE
1857 010460 004737 011544 XINT: CALL CDERCK ;CALL COMMAND ERROR CK
1858 010464 004737 012244 CALL GETREG ;CALL GET DEV REGS
1859 010470 000207 RTS PC ;RETURN
1860 -----
```

```
1861
1862
1863 .SBTTL - MOD U.DEV.CLD - CLEAR DEVICE
1864 :++
1865 : FUNCTIONAL DESCRIPTION: SUBR TO SEND INIT TO DEVICE - NO ERROR CK
1866 : INPUTS: NONE
1867 : IMPLICIT INPUTS: NONE
1868 : OUTPUTS: DEVICE INITIALIZE
1869 : IMPLICIT OUTPUTS:
1870 : SUBORDINATE ROUTINES USED: A WAIT "DONE"
1871 : FUNCTIONAL SIDE EFFECTS:
1872 : CALLING SEQUENCE: SUBR
1873 :--
```

```
1874 -----
1875 :
1876 :
1877 :
1878 010472 012701 040000 CLRDEV: MOV #40000,R1 ;SET INITIALIZE COMMAND
1879 010476 010177 171646 MOV R1,@RXCS ;CLEAR DEVICE
1880 010502 004737 012032 CALL AWDN ;AWAIT DONE
1881 010506 000207 RETURN ;RETURN
1882 -----
```



```

1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899 010510 004737 011634
1900 010514 005737 002454
1901 010520 001035
1902 010522 012737 000001 011542
1903 010530 004737 011502
1904 010534 053737 002366 002400
1905 010542 013777 002400 171600
1906 010550 004737 012110
1907 010554 005737 002454
1908 010560 001015
1909 010562 013777 002370 171562
1910 010570 004737 012110
1911 010574 005737 002454
1912 010600 001005
1913 010602 013777 002362 171542
1914 010610 004737 011610
1915 010614 004737 011544
1916 010620 004737 012244
1917 010624 000207
1918
  
```

```

.SBTTL - MOD U.DEV.FLB - FILL BUFFER
:++
: FUNCTIONAL DESCRIPTION: SUBR TO SEND FILL BUFFER COMMAND TO DEVICE.
: INPUTS: NONE
: IMPLICIT INPUTS: NONE
: OUTPUTS: FILL BUFFER TO RX
: IMPLICIT OUTPUTS:
: SUBORDINATE ROUTINES USED: SET COMMANDS, WAIT "DONE", WAIT "TR"
: FUNCTIONAL SIDE EFFECTS:
: CALLING SEQUENCE:
:--
:-----
FILBUF: CALL DVDNCK ;CALL DEVICE READY CHECK
        TST FIN ;IF FINI FLAG SET
        BNE XFILBF ;NOT SET, THEN
        MOV #1,NCMD ;SET FILL BUFFER COMMAND
        CALL SETSCD ;CALL SET SUBSYS COMMAND - MOD U.DEV.SSC
        BIS EXTADR,CMD ;SET EXT. ADR. BITS
        MOV CMD,@RXCS ;LOAD COMMAND
        CALL AWTR ;WAIT FOR "TR"
IBFLB: TST FIN ;IF FINI FLAG
        BNE XFILBF ;EQUALS ZERO THEN
        MOV WDCNT,@RXDB ;LOAD WORD COUNT FOR OUTPUT BUFFER
        CALL AWTR ;WAIT FOR "TR"
ICFLB: TST FIN ;IF FINI FLAG
        BNE XFILBF ;EQUALS ZERO THEN
        MOV FILADR,@RXDB ;LOAD BASE ADDRESS FOR OUTPUT BUFFER
        CALL WAIT ;WAIT FOR "DONE" OR INTERRUPT
XFILBF: CALL CDERCK ;CALL COMMAND ERROR CHECK
        CALL GETREG ;CALL GET DEV REGS
        RTS PC ;RETURN
:-----
  
```


1957
 1958
 1959
 1960
 1961
 1962
 1963
 1964
 1965
 1966
 1967
 1968
 1969
 1970
 1971
 1972
 1973
 1974
 1975
 1976
 1977
 1978
 1979
 1980
 1981
 1982
 1983
 1984
 1985
 1986
 1987
 1988
 1989
 1990

```

.SBTTL - MOD U.DEV.WRT - WRITE SUBROUTINE
:++
: FUNCTIONAL DESCRIPTION: SUBR TO SEND WRITE SECTOR TO DEVICE.
: INPUTS: NONE
: IMPLICIT INPUTS: DELETED DATA MODE
: OUTPUTS: WRITE SECTOR TO RX
: IMPLICIT OUTPUTS:
: SUBORDINATE ROUTINES USED: SETUP COMMANDS, WAIT "DONE", WAIT "TR"
: FUNCTIONAL SIDE EFFECTS:
: CALLING SEQUENCE:
:--
:-----
WRITE: CALL DVDNCK ;CALL DEVICE READY CHECK
      TST FIN ;IF FINI FLAG
      BNE XWRITE ;EQUALS ZERO THEN
      MOV #5,NCMD ;SET TO WRITE SECTOR
      BIS DELDAT,NCMD ;SETUP WRITE DELETED DATA, IF SET
      CALL SETSCD ;CALL SET SUBSYS COMMAND
      CALL SETDCD ;CALL SET DEVICE COMMAND - MOD U.DEV.CMD
WRITE1: MOV CMD,@RXCS ;LOAD COMMAND
      CALL AWTR ;GO AWAIT TRANSFER READY "TR"
IBWRT: TST FIN ;IF FINI FLAG
      BNE XWRITE ;EQUALS ZERO THEN
      MOV SECTOR,@RXDB ;LOAD SECTOR ADDRESS
      CALL AWTR ;GO AWAIT TRANSFER READY "TR"
ICWRT: TST FIN ;IF FINI FLAG
      BNE XWRITE ;EQUALS ZERO THEN
      MOV TRACK,@RXDB ;LOAD TRACK ADDRESS
      CALL WAIT ;WAIT FOR INTERRUPT OR "DONE"
XWRITE: CALL GETREG ;CALL GET DEV REGS
      RTS PC ;RETURN
:-----

```

```

1993 .SBTTL - MOD U.DEV.RED - READ SUBROUTINE
1994 :++
1995 : FUNCTIONAL DESCRIPTION: SUBR TO SEND READ SECTOR TO DEVICE.
1996 : INPUTS: NONE
1997 : IMPLICIT INPUTS: NONE
1998 : OUTPUTS: READ SECTOR TO RX
1999 : IMPLICIT OUTPUTS:
2000 : SUBORDINATE ROUTINES USED: SETUP COMMANDS, WAIT "DONE", WAIT "TR"
2001 : FUNCTIONAL SIDE EFFECTS:
2002 : CALLING SEQUENCE:
2003 :--
2004 :-----
2005 :
2006 :
2007 011062 004737 011634 READ: CALL DVDNCK ;CALL DEVICE READY CHECK
2008 011066 005737 002454 TST FIN ;IF FINI FLAG
2009 011072 001034 BNE XREAD ;EQUALS ZERO, THEN
2010 011074 012737 000007 011542 MOV #7,NCMD ;SET READ COMMAND
2011 011102 004737 011502 CALL SETSCD ;CALL SET SUBSYS COMMAND
2012 011106 004737 011462 CALL SETDCD ;CALL SET DEVICE COMMAND - MOD U.DEV.SDC
2013 011112 013777 002400 171230 READ1: MOV CMD,@RXCS ;LOAD COMMAND
2014 011120 004737 012110 CALL AWTR ;GO AWAIT TRANSFER READY
2015 011124 005737 002454 IBRED: TST FIN ;IF FINI FLAG
2016 011130 001015 BNE XREAD ;EQUALS ZERO, THEN
2017 011132 013777 002376 171212 MOV SECTOR,@RXDB ;LOAD SECTOR ADDRESS
2018 011140 004737 012110 CALL AWTR ;GO AWAIT TRANSFER READY
2019 011144 005737 002454 ICRED: TST FIN ;IF FINI FLAG
2020 011150 001005 BNE XREAD ;EQUALS ZERO, THEN
2021 011152 013777 002374 171172 MOV TRACK,@RXDB ;LOAD TRACK ADDRESS
2022 011160 004737 011610 CALL WAIT ;WAIT FOR INTERRUPT OR "DONE"
2023 011164 004737 012244 XREAD: CALL GETREG ;CALL GET DEV REGS
2024 011170 000207 RETURN ;RETURN
2025 :-----

```


2028
 2029
 2030
 2031
 2032
 2033
 2034
 2035
 2036
 2037
 2038
 2039
 2040
 2041
 2042 011172 004737 011634
 2043 011176 005737 002454
 2044 011202 001024
 2045 011204 012737 000011 011542
 2046 011212 004737 011502
 2047 011216 004737 011462
 2048 011222 013777 002400 171120
 2049 011230 004737 012110
 2050 011234 005737 002454
 2051 011240 001005
 2052 011242 013777 002372 171102
 2053 011250 004737 011610
 2054 011254 004737 011544
 2055 011260 004737 012244
 2056 011264 000207
 2057

.SBTTL - MOD U.DEV.SDN - SET DENSITY
 ;++
 ; FUNCTIONAL DESCRIPTION: SUBR TO SEND SET DENSITY COMMAND TO DEVICE.
 ; INPUTS: NONE
 ; IMPLICIT INPUTS: DENSITY
 ; OUTPUTS: SET DENSITY TO RX
 ; IMPLICIT OUTPUTS:
 ; SUBORDINATE ROUTINES USED: SETUP COMMANDS, WAIT "DONE", WAIT "TR"
 ; FUNCTIONAL SIDE EFFECTS:
 ; CALLING SEQUENCE:
 ;--

```

-----
SETDN:  CALL  DVDNCK      ;CALL DEVICE READY CHECK
        TST   FIN        ;IF FINI FLAG
        BNE  XSETDN     ;NOT SET, THEN
        MOV  #11,NCMD   ;SETUP DENSITY COMMAND
        CALL SETSCD     ;CALL SET SUBSYS COMMAND
        CALL SETDCD     ;CALL SET DEVICE COMMAND - MOD U.DEV.SDC
        MOV  CMD,@RXCS  ;SEND COMMAND
        CALL AWTR       ;GO AWAIT "TR"
IBSDN:  TST   FIN        ;IF FINI FLAG IS
        BNE  XSETDN     ;ZERO
        MOV  VARIFY,@RXDB ;SEND VARIFY WORD
        CALL WAIT       ;WAIT FOR "DONE" OR INTERRUPT
XSETDN: CALL  CDERCK     ;CALL COMMAND ERROR CHECK
        CALL  GETREG     ;CALL GET DEV REGS
        RTS   PC        ;RETURN
-----
  
```

```

2060 .SBTTL - MOD U.DEV.RST - READ STATUS
2061 :++
2062 : FUNCTIONAL DESCRIPTION: SUBR TO SEND READ STATUS COMMAND TO DEVICE.
2063 : INPUTS: NONE
2064 : IMPLICIT INPUTS: NONE
2065 : OUTPUTS: READ STATUS TO RX
2066 : IMPLICIT OUTPUTS: NONE
2067 : SUBORDINATE ROUTINES USED: SETUP COMMANDS, WAIT "DONE", WAIT "TR"
2068 : FUNCTIONAL SIDE EFFECTS: NONE
2069 : CALLING SEQUENCE:
2070 :--
2071 :
2072 :-----
2073 :
2074 011266 004737 011634 RDSTAT: CALL DVDNCK ;CALL DEVICE READY CHECK
2075 011272 022737 000060 002520 CMP #NODNBT,ERRNBR ;IF ERRNBR NOT SET=
2076 011300 001416 BEQ XRDSTA ;"NO DONE BIT", THEN
2077 011302 012737 000013 011542 MOV #13,NCMD ;SET READ STATUS
2078 011310 004737 011502 CALL SETSCD ;CALL SET SUBSYS COMMAND
2079 011314 004737 011462 CALL SETDCD ;CALL SET DEVICE COMMAND - MOD U.DEV.SDC
2080 011320 013777 002400 171022 MOV CMD,@RXCS ;SEND COMMAND
2081 011326 004737 011610 CALL WAIT ;GO AWAIT "DONE" OR INTERRUPT
2082 011332 004737 012244 CALL GETREG ;CALL GET DEV REGS
2083 011336 000207 XRDSTA: RETURN ;RETURN
2084 :-----

```



```

2087 .SBTTL - MOD U.DEV.REC - READ ERROR CODE
2088 :++
2089 : FUNCTIONAL DESCRIPTION: SUBR TO SEND READ ERROR CODE TO DEVICE.
2090 : INPUTS: EXTENDED ADDRESS BITS, FINI FLAG, SETUP COMMAND WORD
2091 : IMPLICIT INPUTS: NONE
2092 : OUTPUTS: READ ERROR CODE FLAG, READ ERROR CODE TO DRIVE, READ ERROR CODE NEW CMD
2093 : IMPLICIT OUTPUTS: NONE
2094 : SUBORDINATE ROUTINES USED: SET DEVICE CMD, WAIT, WAIT FOR "TR", GET
2095 :                               REG, CMD ERR CK
2096 : FUNCTIONAL SIDE EFFECTS: NONE
2097 : CALLING SEQUENCE: -
2098 :--
2099 :-----
2100
2101 011340 004737 011634 RDERCD: CALL DVDNCK ;CALL DEVICE READY CHECK
2102 011344 022737 000060 002520 CMP #NODNBT,ERRNBR ;IF ERRNBR NOT SET=
2103 011352 001440 BEQ XRDERC ;"NO DONE BIT", THEN
2104 011354 012737 000017 011542 MOV #17,NCMD ;SET ERROR CODE COMMAND
2105 011362 004737 011502 CALL SETSCD ;CALL SET SUBSYS COMMAND
2106 011366 042737 000400 002400 BIC #DENBIT,CMD ;CLEAR DENSITY BIT FROM CMD
2107 011374 053737 002366 002400 BIS EXTADR,CMD ;SET EXTENDED ADDRESS BITS
2108 011402 013777 002400 170740 MOV CMD,@RXCS ;SEND COMMAND
2109 011410 004737 012110 CALL AWTR ;THEN GO AWAIT "TR"
2110 011414 005737 002454 IBREC: TST FIN ;IF FINI FLAG
2111 011420 001015 BNE XRDERC ;EQUALS ZERO THEN
2112 011422 013777 002364 170722 MOV RECADR,@RXDB ;SET BASE ADDR FOR READ ERR CODE
2113 011430 004737 011610 CALL WAIT ;GO AWAIT "DONE" OR INTERRUPT
2114 011434 005737 002454 ICREC: TST FIN ;IF FINI FLAG
2115 011440 001005 BNE XRDERC ;NOT SET, THEN
2116 011442 052737 000200 002476 BIS #RECFLG,FLAGST ;SET PRINT ERROR CODE OR FLAG
2117 011450 004737 012244 CALL GETREG ;CALL GET DEV REGS
2118 011454 004737 011544 XRDERC: CALL CDERCK ;CALL COMMAND ERROR CHECK
2119 011460 000207 RTS PC ;RETURN
2120 :-----

```

2123 .SBTTL - MOD U.DEV.CMD - SETUP DEVICE COMMAND
2124 :++
2125 : FUNCTIONAL DESCRIPTION: SUBR TO SETUP DEVICE COMMAND WORD - I.E.,
2126 : SET DRIVE & SIDE BITS
2127 : INPUTS: NONE
2128 : IMPLICIT INPUTS: SIDE & DRIVE BITS, COMMAND
2129 : OUTPUTS: COMMAND WORD FOR DEVICE
2130 : IMPLICIT OUTPUTS: NONE
2131 : SUBORDINATE ROUTINES USED: NONE
2132 : FUNCTIONAL SIDE EFFECTS: NONE
2133 : CALLING SEQUENCE: SUBR
2134 :--
2135 :
2136 :-----

2137									
2138	011462	000240				SETDCD: NOP			
2139	011464	053737	002406	002400		BIS DRIVE,CMD			:SETUP DRIVE BIT
2140	011472	053737	002410	002400		BIS SIDE,CMD			:SETUP SIDE BIT
2141	011500	000207				RETURN			:RETURN

2142 :-----
2143 :
2144 .SBTTL - MOD U.DEV.SSC - SETUP SUBSYSTEM COMMANDS
2145 :++
2146 : FUNCTIONAL DESCRIPTION: SUBR TO SETUP SUBSYSTEM COMMANDS - I.E.
2147 : SET BITS THAT ARE NOT DRIVE RELATED
2148 : INPUTS: NEW COMMAND
2149 : IMPLICIT INPUTS: COMMAND, DENSITY, INTERRUPT BIT
2150 : OUTPUTS: COMMAND
2151 : IMPLICIT OUTPUTS: LAST COMMAND, PROTOCOL CTR
2152 : SUBORDINATE ROUTINES USED: NONE
2153 : FUNCTIONAL SIDE EFFECTS: NONE
2154 : CALLING SEQUENCE: SUBR
2155 :--
2156 :-----

2157	011502	000240				SETSCD: NOP			
2158	011504	013737	002400	002424		MOV CMD,LCMD			:SAVE LAST COMMAND
2159	011512	013737	011542	002400		MOV NCMD,CMD			:SETUP NEW COMMAND
2160	011520	0C5037	002472			CLR PROTCT			:CLEAR TEST COMMAND PROTOCOL COUNTER
2161	011524	053737	002412	002400		BIS DENSTY,CMD			:SETUP DENSITY BIT
2162	011532	053737	002404	002400		BIS INTERT,CMD			:SETUP INTERRUPT BIT
2163	011540	000207				RETURN			:

2164 :-----
2165 NCMD: 0 ;NEW COMMAND
2166 :-----
2167 :


```
2170 .SBTTL - MOD U.DEV.CEC - DEVICE COMMAND ERROR CHECK
2171 -----
2172 :++
2173 : FUNCTIONAL DESCRIPTION: SUBR TO CHECK FOR DEVICE COMMAND FATAL ERRORS.
2174 : INPUTS: NONE
2175 : IMPLICIT INPUTS: FIN FLAG, FLAGS(NEG TEST), ERR NBR
2176 : OUTPUTS: NONE
2177 : IMPLICIT OUTPUTS: ERROR CONDITION
2178 : SUBORDINATE ROUTINES USED: ERROR
2179 : FUNCTIONAL SIDE EFFECTS: DROP UNIT & CLEAN UP
2180 : CALLING SEQUENCE: SUBR
2181 :--
2182 -----
2183 :
2184 :
2185 011544 000240 CDERCK: NOP ;
2186 011546 005737 002454 TST FIN ;IF FINI FLAG
2187 011552 001415 BEQ XCEC ;SET, THEN
2188 011554 005737 002520 TST ERRNBR ;IF ERROR NUMBER
2189 011560 001412 BEQ XCEC ;NOT=0, THEN
2190 011562 032737 004000 002476 BIT #NEGTST,FLAGST ;IF NEG TEST FLAG
2191 011570 001006 BNE XCEC ;NOT SET, THEN
2192 011572 004737 003060 CALL ERROR ;CALL ERROR-MOD
2193 011576 DODU UNIT ;DROP UNIT
2194 011604 DOCLN ;DO CLEAN-UP
2195 011606 000207 XCEC: RETURN ;RETURN
2196 -----
```

```

2199 .SBTTL - MOD U.DEV.WAT - WAIT SUBROUTINE
2200 :++
2201 : FUNCTIONAL DESCRIPTION: SUBR TO DETERMINE TO WAIT FOR "DONE" OR INTERRUPTS
2202 : INPUTS: DEVICE COMMAND
2203 : IMPLICIT INPUTS: NONE
2204 : OUTPUTS: NONE
2205 : IMPLICIT OUTPUTS: NONE
2206 : SUBORDINATE ROUTINES USED: WATCH & A WAIT DONE
2207 : FUNCTIONAL SIDE EFFECTS: NONE
2208 : CALLING SEQUENCE: SUBR
2209 :--
2210
2211
2212
  
```

```

-----
2213 011610 032737 000100 002400 WAIT: BIT #100,CMD ;IF COMMAND-INTERRUPT BIT
2214 011616 001403 BEQ 1$ ;SET, THEN
2215 011620 004737 011662 CALL WATCH ;CALL WATCH DOG WAIT FOR INTERRUPT
2216 011624 000402 BR XWAIT ;BR TO END
2217 011626 004737 012032 1$: CALL AWDN ;ELSE, CALL WAIT FOR DONE
2218 011632 000207 XWAIT: RETURN ;RETURN
-----
  
```

```

2220
2221
2222
2223 .SBTTL - MOD U.DEV.DRC - DEVICE DONE CHECK
2224 :++
2225 : FUNCTIONAL DESCRIPTION: SUBR TO CK IF DEVICE IS READY TO ACCEPT A CMD
2226 : INPUTS: NONE
2227 : IMPLICIT INPUTS: DONE BIT
2228 : OUTPUTS: NONE
2229 : IMPLICIT OUTPUTS: NONE
2230 : SUBORDINATE ROUTINES USED: WATCH & A WAIT DONE
2231 : FUNCTIONAL SIDE EFFECTS: NONE
2232 : CALLING SEQUENCE: SUBR
2233 :--
2234
2235
2236
  
```

```

-----
2237 011634 005003 DVDNCK: CLR R3 ;CLEAR REC
2238 011636 032777 000040 170504 1$: BIT #DNBIT,@RXCS ;IF DEVICE DONE
2239 011644 001005 BNE XDVRCK ;NOT SET
2240 011646 005203 INC R3 ;BUMP TIMEOUT COUNTER
2241 011650 001372 BNE 1$ ;IF TIME OUT, THEN
2242 011652 BRESET ;EXECUTE BUS RESET
2243 011654 004737 012032 CALL AWDN ;CALL A WAIT "DONE"
2244 011660 000207 XDVRCK: RETURN ;RETURN
-----
2245
  
```



```

2248 .SBTTL - MOD U.DEV.WCH - WATCH DOG TIMER
2249 :++
2250 : FUNCTIONAL DESCRIPTION: SUBR TO WATCH DOG DEVICE 'DONE' & INTERRUPTS
2251 : INPUTS: PROCESSOR LOW PRIORITY
2252 : IMPLICIT INPUTS: DEVICE 'DONE' & INTERRUPTS
2253 : OUTPUTS: DONE TIMEOUT ERROR, NO INTERRUPT ERROR
2254 : IMPLICIT OUTPUTS: NONE
2255 : SUBORDINATE ROUTINES USED: NONE
2256 : FUNCTIONAL SIDE EFFECTS: NONE
2257 : CALLING SEQUENCE: SUBR
2258 :--
2259 :
2260 :-----
2261 :
2262 011662 005037 012030 WATCH: CLR DNFLAG ;CLEAR DONE FLAG
2263 011666 SETPRI PRIORT ;SET PROCESSOR PRI - ALLOW INTERRUPTS
2264 011674 013704 012024 MOV DX,R4 ;SET DELAY MULT
2265 011700 013703 012026 BAUWCH: MOV DLY,R3 ;SET DELAY
2266 011704 005737 012030 IBUWCH: TST DNFLAG ;IF INTERRUPTS DONE FLAG
2267 011710 001413 BEQ LBUWCH ;IS SET, THEN
2268 011712 032777 000040 170430 ICUWCH: BIT #DNBIT,@RXCS ;IF DONT BIT
2269 011720 001035 BNE XUWCH ;IS NOT SET, THEN
2270 011722 012737 000014 002520 MOV #INTNDN,ERRNBR ;SET ERROR #-NO DONE BIT
2271 011730 012737 010000 002460 MOV #BIT12,TYPERR ;SET INTERR, BUT NO DONE ERROR
2272 011736 000426 BR XUWCH ;BR TO MOD 'EXIT'
2273 011740 LBUWCH: BREAK
2274 011742 005303 DEC R3 ;DECREMENT DELAY COUNT
2275 011744 001357 UDUWCH: BNE IBUWCH ;DO UNIT DELAY COUNT=0
2276 011746 005304 DEC R4 ;DECREMENT DELAY MULT
2277 011750 001353 UAUWCH: BNE BAUWCH ;DO UNTIL DELAY MULT=0
2278 011752 032777 000040 170370 IEUWCH: BIT #DNBIT,@RXCS ;IF DONE BIT IS
2279 011760 001407 BEQ LEUWCH ;SET, THEN
2280 011762 012737 000015 002520 MOV #DNNINT,ERRNBR ;SET ERR #-DONE, NO INTR
2281 011770 052737 020000 002460 BIS #BIT13,TYPERR ;SET DONE, BUT NO INTERRUPT ERROR
2282 011776 000406 BR XUWCH ;BR TO MOD 'EXIT'
2283 012000 052737 000020 002456 LEUWCH: BIS #BIT4,SYSERR ;SET NO DONE T.O. ERROR
2284 012006 012737 000060 002520 MOV #NODNBT,ERRNBR ;SET ERR #-NO DONE BIT
2285 012014 XUWCH: SETPRI #PRI07 ;SET PROCESSOR PRI=7 - NO INTERRUPTS
2286 012022 000207 RTS PC ;RETURN
2287 :-----
2288 012024 000010 DX: 10 ;DELAY MULT
2289 012026 100000 DLY: 100000 ;DELAY
2290 012030 000000 DNFLAG: 0 ;DONE FLAG
2291 ;MOD U.2.3.4 ---- END MODULE -----

```

```

2294 .SBTTL - MOD U.DEV.WDN - AWAIT DONE BIT SUBROUTINE
2295 :++
2296 : FUNCTIONAL DESCRIPTION: SUBR TO WAIT FOR DEVICE "DONE" BIT
2297 : INPUTS: TIMEOUT PASS COUNTER
2298 : IMPLICIT INPUTS: DEVICE "DONE" BIT, (RXCSR), DONE WAIT MULTIPLIER
2299 : OUTPUTS: "DONE" BIT TIMEOUT ERROR
2300 : IMPLICIT OUTPUTS: NONE
2301 : SUBORDINATE ROUTINES USED: GET DEVICE REGISTERS
2302 : FUNCTIONAL SIDE EFFECTS: NONE
2303 : CALLING SEQUENCE: SUBR
2304 :--
  
```

```

2307 -----
2308 012032 005004 AWDN: CLR R4 ;RESET TIME OUT MULTIPLIER
2309 012034 005003 1$: CLR R3 ;PRESET TIME OUT COUNTER
2310 012036 032777 000040 170304 2$: BIT #DNBIT,@RXCS ;SEE IF DONE SET
2311 012044 001020 BNE 3$ ;IF SO: BR
2312 012046 BREAK ;TEMPORARY RETURN TO MONITOR
2313 012050 005203 INC R3 ;BUMP TIME OUT COUNTER
2314 012052 001371 BNE 2$ ;IF NOT TIMED OUT: BR
2315 012054 005204 INC R4 ;INCREMENT TIMEOUT MULTIPLIER
2316 012056 023704 002474 CMP DNWTMT,R4 ;IF ON 2ND
2317 012062 101364 BHI 1$ ;TIMEOUT PASS, THEN
2318 012064 012737 000060 002520 MOV #NODNBT,ERRNBR ;SET ERR #-NO DONE BIT
2319 012072 052737 000020 002456 BIS #BIT4,SYSERR ;SET NO DONE BIT ON SYSTEM ERROR
2320 012100 012737 000001 002454 MOV #1,FIN ;EXIT THIS COMMAND
2321 012106 000207 3$: RTS PC ;EXIT
2322 -----
  
```



```

2325 .SBTTL - MOD U.DEV.WTR - AWAIT TRANSFER READY SUBROUTINE
2326 :++
2327 : FUNCTIONAL DESCRIPTION: SUBR TO WAIT FOR DEVICE "TR" BIT
2328 : INPUTS: NONE
2329 : IMPLICIT INPUTS: DEVICE "TR", "DONE" & CSR, ESR
2330 : OUTPUTS: "TR" TIMEOUT ERROR, NO DONE BIT, PROTOCOL COUNTER
2331 : IMPLICIT OUTPUTS: NONE
2332 : SUBORDINATE ROUTINES USED: GET DEVICE REGISTERS
2333 : FUNCTIONAL SIDE EFFECTS: NONE
2334 : CALLING SEQUENCE: SUBR
2335 :--
  
```

```

-----
2339 012110 005237 002472 AWTR: INC PROTCT ;INCREMENT TEST PROTOCOL COUNTER
2340 012114 005004 CLR R4 ;PRESET TIMEOUT MULTIPLIER
2341 012116 005003 CLR R3 ;PRESET TIME OUT COUNTER
2342 012120 032777 000040 170222 2$: BIT #DNBIT,@RXCS ;IF DONE BIT
2343 012126 001013 BNE 3$ ;NOT SET, THEN
2344 012130 032777 000200 170212 BIT #TRBIT,@RXCS ;SEE IF TRANSFER READY SET
2345 012136 001041 BNE 5$ ;IF SO: BR
2346 012140 BREAK ;TEMPORARY RETURN TO MONITOR
2347 012142 005203 INC R3 ;BUMP TIME OUT COUNTER
2348 012144 001365 BNE 2$ ;IF NOT TIMED OUT: BR
2349 012146 005204 INC R4 ;INCREMENT TIMEOUT MULTIPLIER
2350 012150 022704 000004 CMP #4,R4 ;IF ON 2ND
2351 012154 101360 BHI 1$ ;TIMEOUT PASS, THEN
2352 012156 012737 000001 002454 3$: MOV #1,FIN ;EXIT THIS COMMAND
2353 012164 052737 000020 002460 BIS #CMDERR,TYPERR ;**** ERROR ON COMMAND ****
2354 012172 013737 002400 002422 MOV CMD,ERRCMD ;SETUP ERROR COMMAND
2355 012200 012737 000057 002520 MOV #NOTRBT,ERRNBR ;SET ERR #=NO "TR" BIT
2356 012206 052737 000200 002456 BIS #TRBIT,SYSERR ;SET SYS ERR=NO "TR" BIT
2357 012214 032777 000040 170126 BIT #DNBIT,@RXCS ;IF DONE BIT
2358 012222 001004 BNE 4$ ;NOT SET, THEN
2359 012224 052737 000020 002456 BIS #BIT4,SYSERR ;SET NO DONE BIT EITHER
2360 012232 000403 RR 5$ ;BR TO EXIT
2361 012234 012737 000063 002520 4$: MOV #DNNOTR,ERRNBR ;SET ERR #="DONE" NO "TR"
2362 012242 000207 5$: RTS PC ;RETURN
-----
  
```

```
2366 .SBTTL - MOD U.DEV.REG - GET DEVICE REGISTERS
2367 :++
2368 : FUNCTIONAL DESCRIPTION: SUBROUTINE TO GET RX02 CSR & ESR
2369 : INPUTS: NONE
2370 : IMPLICIT INPUTS: DEVICE CSR & ESR
2371 : OUTPUTS: DEVICE CSR & ESR
2372 : IMPLICIT OUTPUTS: OLD CSR & ESR
2373 : SUBORDINATE ROUTINES USED: NONE
2374 : FUNCTIONAL SIDE EFFECTS: NONE
2375 : CALLING SEQUENCE: SUBR
2376 :--
```

```
-----
2379 GETREG: MOV     RXCSR,LRXCSR    ;SAVE LAST CSR
2380 012244 013737 002432 002426      MOV     RXESR,LRXESR    ;SAVE LAST ESR
2381 012252 013737 002434 002430      MOV     @RXCS,RXCSR    ;GET RXCSR FOR PRINT
2382 012260 017737 170064 002432      MOV     @RXDB,RXESR    ;GET RXESR FOR PRINT
2383 012266 017737 170060 002434      RETURN                    ;RETURN
2384 012274 000207
2385 :-----
```

```
2386
2387
2388 .SBTTL - MOD U.DEV.ITR - INTERRUPT HANDLER
2389 :++
2390 : FUNCTIONAL DESCRIPTION: ;DEVICE INTERRUPT HANDLER
2391 : INPUTS: NONE
2392 : IMPLICIT INPUTS: DEVICE "DONE" BIT & INTERRUPT BIT
2393 : OUTPUTS: DONE FLAG
2394 : IMPLICIT OUTPUTS: NONE
2395 : SUBORDINATE ROUTINES USED: NONE
2396 : FUNCTIONAL SIDE EFFECTS: NONE
2397 : CALLING SEQUENCE: DEVICE INTERRUPT
2398 :--
```

```
-----
2402 INTRHD: MOV     #1,DNFLAG    ;SET DONE FLAG
2403 012276 012737 000001 012030      RTI     ;RETURN FROM INTERRUPT
2404 012304 000002
2405 :-----
2406
```



```

2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422 012306 042737 000377 012372
2423 012314 005037 012654
2424 012320 005737 012660
2425 012324 001003
2426 012326 012737 000007 012660
2427 012334 013704 012660
2428 012340 005304
2429 012342 006304
2430 012344 150437 012372
2431 012350 012704 036224
2432 012354 013705 002370
2433 012360 006305
2434 012362 062705 036222
2435 012366 162705 000004
2436 012372 000777
2437 012374 000137 012430
2438 012400 000137 012446
2439 012404 000137 012456
2440 012410 000137 012524
2441 012414 000137 012532
2442 012420 000137 012556
2443 012424 000137 012566
2444
2445 012430 005037 012656
2446 012434 004737 012614
2447 012440 005705
2448 012442 001463
2449 012444 000773
2450
2451 012446 112737 000377 012656
2452 012454 000767
2453
2454 012456 112737 000376 012656
2455 012464 000261
2456 012466 012702 000000
2457 012472 103001
2458 012474 005202
2459 012476 004737 012614
2460 012502 005705
2461 012504 001442
2462 012506 000241
2463 012510 005702
2464 012512 001401
2465 012514 000261

.SBTTL - MOD U.SFT.DPT - SET DATA PATTERN
-----
PAT #  SUBROUTINE  DATA PATTERN
-----
0      RANDAT      NO PATTERN SPECIFIED (FORCE RANDOM DATA)
1      DATA0      ALL ZEROS
2      DATA1      ALL ONES
3      FLOAT0      FLOATING ZERO
4      FLOAT1      FLOATING ONE
5      PAT125      ALTERNATING BITS
6      PAT333      ALTERNATING PAIRS OF BITS
7      RANDAT      RANDOM
-----
STDATP: BIC      #377,@#BRONPT  ;CLEAR BRANCH OFFSET
        CLR      SUM          ;SET UP FOR ACCUMULATION OF CHECK SUM
        TST      PAT          ;IF NO PATTERN SPECIFIED FORCE PATTERN 7
        BNE     1$
        MOV     #7,PAT
        MOV     PAT,R4        ;GET PATTERN BITS
        DEC     R4           ;ADJUST FOR CORRECT OFFSET
        ASL     R4
        BISB    R4,@#BRONPT  ;INSERT OFFSET
        MOV     #DATPAT+2,R4  ;SET UP ADDRESS OF FIRST BYTE
        MOV     WDCNT,R5     ;SETUP WORD COUNT
        ASL     R5           ;DOUBLE WORD COUNT FOR ADR
        ADD     #DATPAT,R5   ;ADD DATA PATTERN ADR
        SUB     #4,R5        ;ADJ. FOR CHECKSUM
BRONPT: BR      .           ;BRANCH BY OFFSET SELECTED
        JMP     DATA0       ;000 DATA BYTE
        JMP     DATA1       ;377 DATA BYTE
        JMP     FLOAT0      ;FLOAT A 0 THROUGH ALL 1'S
        JMP     FLOAT1      ;FLOAT A 1 THROUGH ALL 0'S
        JMP     PAT125      ;125/052 DATA WORD
        JMP     PAT333      ;314/063 DATA WORD
        JMP     RANDAT      ;RANDOM DATA BYTE
-----
DATA0:  CLR      DATBYT
PG:     JSR      PC,LOAD     ;GO LOAD THE DATA BUFFER
        TST      R5         ;IF R5
        BEQ     END131      ;NOT =0 ,THEN
        BR      PG
-----
DATA1:  MOVB    #377,DATBYT
        BR      PG
-----
FLOAT0: MOVB    #376,DATBYT  ;SET UP A ONES FIELD
XPG:   SEC
        SEC      ;SET THE C BIT TO ROTATE THROUGH THE DATA
1$:    MOV     #0,R2        ;CLR R2 (CAN'T USE "CLR" AS IT CLEARS "C" BIT)
        BCC     2$         ;BR IF THE "C" BIT IS CLEARED
        INC     R2         ;SET R2 IF NOT
2$:    JSR      PC,LOAD     ;GO LOAD THE DATA BUFFER
        TST      R5         ;IF R5
        BEQ     END131      ;NOT ZERO THEN
        CLC
        TST     R2         ;IS R2 NONZERO
        BEQ     3$
        SEC
        SEC      ;YES, SET THE "C" BIT
  
```

```

2466 012516 106137 012656      3$:   ROLB   DATBYT
2467 012522 000761             BR     1$
2468                               -----
2469 012524 005037 012656      FLOAT1: CLR   DATBYT
2470 012530 000755             BR     XPG
2471                               -----
2472 012532 112737 000125 012656  PAT125: MOVB  #125,DATBYT
2473 012540 004737 012614      XXPG:  JSR   PC,LOAD
2474 012544 005705             TST   R5           ;IF R5
2475 012546 001421             BEQ   END131       ;NOT ZERO THEN
2476 012550 105137 012656             COMB  DATBYT
2477 012554 000771             BR     XXPG
2478                               -----
2479 012556 112737 000333 012656  PAT333: MOVB  #333,DATBYT
2480 012564 000765             BR     XXPG
2481                               -----
2482 012566 004737 010344      RANDAT: JSR   PC,RANGEN ;GET RANDOM NUMBER
2483 012572 113737 010436 012656      MOVB  RANUM,DATBYT
2484 012600 004737 012614             JSR   PC,LOAD
2485 012604 005705             TST   R5           ;IF R5
2486 012606 001401             BEQ   END131       ;NOT ZERO THEN
2487 012610 000766             BR     RANDAT
2488                               -----
2489 012612 000207      END131: RTS   PC           ;RETURN.
2490                               -----
2491
2492
2493                               -----
2494 012614 063737 012656 012654  LOAD:  ADD   DATBYT,SUM ;ACCUMULATE THE PATTERN CHECK SUM
2495 012622 113724 012656             MOVB  DATBYT,(R4)+ ;LOAD THE DATA BUFFER
2496 012626 020504             CMP   R5,R4       ;HAVE 124 BYTES BEEN GENERATED
2497 012630 001401             BEQ   1$         ;IF YES, RETURN
2498 012632 000407             BR    ENLDL      ;IF NO, RETURN TO PATTERN GENERATOR
2499 012634 113724 012654      1$:   MOVB  SUM,(R4)+ ;PUT CHECKSUM INTO TABLE
2500 012640 005137 012654             COM   SUM        ;COMPLIMENT CHECKSUM
2501 012644 113714 012654             MOVB  SUM,(R4)   ;PUT COMP CHECK SUM INTO TABLE
2502 012650 005005             CLR   R5        ;CLEAR TEMP #5 - FLAG DONE MODULE
2503 012652 000207      ENLDL: RTS   PC           ;RETURN
2504                               -----
2505 012654 000000      SUM:   0
2506 012656 000000      DATBYT: 0
2507 012660 000000      PAT:   0
2508                               -----
;MOD 1.3.1 ----- END MODULE -----

```



```

2511          .SBTTL - MOD U.SFT.GTK - GET TRACK
2512          :-----
2513
2514 012662 000240          GETTRK: NOP
2515 012664 032737 000400 002510 IAGTK: BIT #ITK,TKSCFG ;IF INITIALIZE TRK IS
2516 012672 001423          BEQ ICGTK ;SET, THEN
2517 012674 042737 000400 002510 BIC #ITK,TKSCFG ;RESET INITIALIZE TRK FLG
2518 012702 013737 002336 013100 MOV ID,TRKCNT ;GET INSIDE TRACK
2519 012710 163737 002334 013100\ SUB OD,TRKCNT ;GET OUTSIDE TRACK
2520 012716 005237 013100 INC TRKCNT ;INCREMENT # OF TRACKS
2521 012722 013737 002334 002374 MOV OD,TRACK ;SET TRACK=O.D.
2522 012730 005337 002374 DEC TRACK ;DECREMENT TRACK
2523 012734 042737 001000 002476 BIC #TRKDON,FLAGST ;CLEAR TRACK DONE FLAG
2524 012742 032737 000004 002510 ICGTK: BIT #ILTK,TKSCFG ;IF TK/SC FLAGS=ILLEGAL TRACK
2525 012750 001416          BEQ LBGTK ;BIT SET, THEN
2526 012752 012737 000115 002374 MOV #77.,TRACK ;SET TRACK=77=ILLEGAL TRACK
2527 012760 052737 001000 002476 BIS #TRKDON,FLAGST ;SET TRACK DONE FLAG
2528 012766 000443          BR XGTK ;BR TO EXIT
2529 012770 032737 000001 002510 IBGTK: BIT #STK,TKSCFG ;IF TK & SE FLAG=SEQ TRK FLAG
2530 012776 001403          BEQ LBGTK ;SET, THEN
2531 013000 005237 002374 INC TRACK ;INCREMENT TRACK
2532 013004 000426          BR EBGTK ;BR TO END 'B'
2533 013006 004737 010344          LBGTK: CALL RANGEN ;GET A RANDOM NUMBER
2534 013012 042737 177600 010436 BIC #177600,RANUM ;CLEAR ALL BUT LOW 7 BITS
2535 013020 123737 010436 002336 IDCOMP: CMPB RANUM,ID ;IF RANUM LARGER THAN ID ADDRESS
2536 013026 003401          BLE ODCOMP ;THEN
2537 013030 000766          BR LBGTK ;BR TO GET ANOTHER RANDOM NUMBER
2538 013032 123737 010436 002334 ODCOMP: CMPB RANUM,OD ;IF RANUM SMALLER THAN OD ADDRESS
2539 013040 002001          BGE PRESCK ;THEN
2540 013042 000761          BR LBGTK ;BR TO GET ANOTHER RANDOM NUMBER
2541 013044 123737 010436 002374 PRESCK: CMPB RANUM,TRACK ;IF RANUM EQUALS PRESENT TRACK
2542 013052 001755          BEQ LBGTK ;GET ANOTHER RANDOM NUMBER
2543 013054 013737 010436 002374 MOV RANUM,TRACK ;RANUM OK PUT IT IN TARGET TRACK
2544 013062 005337 013100          EBGTK: DEC TRKCNT ;IF TOTAL # OF TRACKS
2545 013066 001003          BNE XGTK ;DONE, THEN
2546 013070 052737 001000 002476 BIS #TRKDON,FLAGST ;THEN SET TRACK DONE FLAG
2547 013076 000207          XGTK: RTS PF
2548          :-----
2549 013100 000000          TRKCNT: .WORD 0 ;DRV TRK TABLE LOCATOR
2550 013102 000000          INITTK: .WORD 0 ;INITIALIZE TRK FLAG
2551          :-----
  
```

```

2554          .SBTTL - MOD U.SFT.GSC - GET SECTOR
2555          ;-----
2556
2557 013104 000240          GETSEC: NOP          ;
2558 013106 032737 001000 002510 IAGSC: BIT      #ISC,TKSCFG      ;IF TK/SC FLAGS=INIT SECTORS BIT
2559 013114 001411          BEQ      IBGSC          ;SET, THEN
2560 013116 042737 001000 002510 BIC      #ISC,TKSCFG      ;CLEAR THE FLAG
2561 013124 012737 000001 002376 MOV      #1,SECTOR      ;SET SECTOR=1
2562 013132 042737 002000 002476 BIC      #SECDON,FLAGST ;CLEAR FLAGST-SECTOR DONE FLAG
2563 013140 105737 002510          IBGSC: TSTB     TKSCFG          ;IF SEQUENCE SECTOR
2564 013144 001416          BEQ      BCGSC          ;SET, THEN
2565 013146 062737 000001 002376 ADD      #1,SECTOR      ;BUMP SECTOR ADDRESS
2566 013154 022737 000033 002376 CMP      #33,SECTOR     ;IF SECTORS
2567 013162 101030          BHI      XGSC          ;DONE, THEN
2568 013164 012737 000001 002376 MOV      #1,SECTOR      ;SET SECTOR=1
2569 013172 052737 002000 002476 BIS      #SECDON,FLAGST ;SET FLAGST-SECTOR DONE FLAG
2570 013200 000421          BR      XGSC          ;BR EXIT
2571 013202 004737 010344          BCGSC: CALL    RANGEN      ;BGN DO 'C'-CALL RANDOM NO. GENERATOR
2572 013206 042737 177740 010436 BIC      #177740,RANUM  ;CLEAR TOP BITS RANDOM NUM.
2573 013214 123727 010436 000033 UCGSC: CMPB     RANUM,#27. ;DUNTIL RANUM < 27.
2574 013222 103367          BHIS    BCGSC          ;
2575 013224 105737 010436          IDGSC: TSTB     RANUM      ;IF RANDOM NO.
2576 013230 001002          BNE     EDGSC          ;EQUALS ZERO, THEN
2577 013232 105237 010436          INCB    RANUM          ;SET RANUM = 1
2578 013236 113737 010436 002376 EDGSC: MOVB    RANUM,SECTOR ;SET SECTOR ADR = RANDOM NO.
2579 013244 000207          XGSC:  RTS      PC
2580          ;-----

```



```

2583          .SBTTL - MOD U.SFT.DCK - DATA CHECK
2584          ;-----
2585
2586 013246 005037 013520 DATAACK: CLR DAERCT          :CLEAR DATA ERR COUNT
2587 013252 052737 000100 002500      BIS #HDRPRT,FLAGSP :SET PRINT HEADER FLAG
2588 013260 013737 002370 013514      MOV WDCNT,BYTCNT   :SAVE WORD COUNT
2589 013266 006337 013514      ASL BYTCNT        :DOUBLE IT SO BYTE COUNT
2590 013272 005037 013516      CLR BYTNUM        :CLEAR BYTE NUMBER
2591 013276 013705 013516      BADCK: MOV BYTNUM,R5 :SETUP BYTE NUMBER FOR AUTO INDEX
2592 013302 116501 036222      MOV#B DATPAT(R5),R1 :SET TEMP#1=DATA SOURCE BYTE
2593 013306 116502 036622      MOV#B DATBUF(R5),R2 :SET TEMP#2=DATA BUFFER BYTE
2594 013312 120102      IBDCCK: CMPB R1,R2 :IF SOURCE BYTE & BUFFER BYTE
2595 013314 001465      BEQ EBDCK         :NOT EQUAL
2596 013316 005237 013520      INC DAERCT        :INCREMENT DATA ERR COUNT
2597 013322 023727 013520 000012 IEDCK: CMP DAERCT,#10. :IF OVER 10 DATA ERRORS
2598 013330 103404      BLO TFDCK        :THEN
2599 013332 032737 000020 002332 IFDCK: BIT #20,SWREG :IF PRINT ONLY 10 DATA ERROR FLAG
2600 013340 001053      BNE EBDCK        :IS NOT SET, THEN
2601 013342 110137 013522      TFDCK: MOV#B R1,DATASB :GET DATA SHOULD BE->PRINT
2602 013346 110237 013524      MOV#B R2,DATAWS   :GET DATA WAS->PRINT
2603 013352 032737 000100 002500 IMDCK: BIT #HDRPRT,FLAGSP :IF PRINT HEADER
2604 013360 001431      BEQ EMDCK        :OK, THEN
2605 013362 042737 000100 002500      .IC #HDRPRT,FLAGSP :CLEAR PRINT HEADER
2606 013370 012737 000005 002520      MOV #DATER,ERRNBR :SETUP ERR NBR= DATA ERR
2607 013376 004737 003060      CALL ERROR       :CALL ERROR
2608 013402 032737 000020 002476 INDCK: BIT #EMBUFF,FLAGST :IF EMPTY BUFFER BIT
2609 013410 001011      BNE ENDCK        :NOT SET, THEN
2610 013412 012701 013526      MOV #DMSG1B,R1   :SETUP MSG FORMAT
2611 013416 013702 002374      MOV TRACK,R2    :SETUP TRACK # PRT
2612 013422 013703 002376      MOV SECTOR,R3   :SETUP SECTOR # PRT
2613 013426 004737 002612      CALL PRTB2S     :CALL PRINT BASIC-2 ARG
2614 013432 000400      BR ENDCK        :BR TO END 'N'
2615 013434 012701 013561      ENDCK: MOV #DMSG1,R1 :SETUP MSG FORMAT
2616 013440 004737 002550      CALL PRTB0S     :CALL PRINT BASIC-0 ARG
2617 013444 012701 013615      EMDCK: MOV #DMSG2,R1 :SETUP MSG FORMAT
2618 013450 013702 013516      MOV BYTNUM,R2   :SETUP BYTE #
2619 013454 013703 013522      MOV DATASB,R3  :SETUP DATA SHOULD BE
2620 013460 013704 013524      MOV DATAWS,R4 :SETUP DATA WAS
2621 013464 004737 002636      CALL PRTB3S     :CALL PRINT BASIC-3 ARG
2622 013470 005237 013516      EBDCK: INC BYTNUM :INCREMENT BYTE #
2623 013474 005337 013514      DEC BYTCNT      :DECREMENT BYTE COUNT
2624 013500 005737 013514      UADCK: TST BYTCNT :DOUNTIL BYTE COUNT
2625 013504 003274      BGT BADCK       :EQUALS 0
2626 013506 004737 013642      ENDDCK: CALL CLRDAT :CALL CLEAR DATA BUFFER
2627 013512 000207      RTS PC         :RETURN
2628
2629 013514 000000      BYTCNT: 0       :BYTE COUNT
2630 013516 000000      BYTNUM: 0       :BYTE NUMBER
2631 013520 000000      DAERCT: 0       :DATA ERR COUNT
2632 013522 000000      DATASB: 0       :DATA SHOULD BE
2633 013524 000000      DATAWS: 0      :DATA WAS
2634
2635 013526 045 116 045 DMSG1B: .ASCIZ /%N% TRK#%D3%A. SEC#%D2%A./
2636 013561 045 116 045 DMSG1: .ASCIZ /%N% BYTE#%S2%AGOOD%S6%ABAD/
2637 013615 045 116 045 DMSG2: .ASCIZ /%N%S3%D3%S2%B8%S2%B8/
2638          .EVEN
2639          ;-----

```

2642
2643
2644
2645 013642 012705 036622
2646 013646 012704 000200
2647 013652 005025
2648 013654 005304
2649 013656 005704
2650 013660 001374
2651 013662 000207
2652

```
.SBTTL - MOD U.SFT.CDB - CLEAR DATA BUFFER  
;-----  
CLRDAT: MOV #DATBUF,R5 ;GET BEGIN OF DATA BUFFER  
          MOV #128.,R4 ;SET WORD LENGTH OF TABLE  
BACDB: CLR (R5)+ ;CLEAR WORD IN DATA BUFFER TABLE  
        DEC R4 ;DECREMENT WORD COUNT  
        TST R4 ;DO UNTIL  
UACDB: BNE BACDB ;ALL TABLE WORDS ZEROED  
        RETURN ;RETURN  
;-----
```


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

.SBTTL - MOD U.SFT.RCR - REGISTER CHECK & REPORT

```

BGNSUB
  IF FINI FLAG NOT SET
  : THEN-
  :   GET TEST TABLE ADDRESS
  :   DOUNTIL TEST TABLE ENTRY=-1
  :   : ADVANCE TEST TABLE ADDRESS
  :   ENDDO
  :   ADVANCE TEST TABLE ADDRESS
  :   GET COMMAND COUNTER
  :   DOUBLE COMMAND COUNTER
  :   ADDRESS OF REG TABLE THIS CMD=CMD COUNTER + TEST TABLE ADR
  :   GET ADDRESS OF REG TABLE THIS COMMAND
  :   SET RXCSR COMPARE WORD=COMPARE WORD FROM TABLE
  :   SET RXCSR MASK WORD=DON'T CARE BITS FROM REG TABLE
  :   SET RXESR COMPARE WORD=COMPARE WORD FROM TABLE
  :   SET RXESR MASK WORD=DON'T CARE BITS FROM REG TABLE
  :   SETUP CSR REG CK
  :   IF RXCSR NOT=CSRCMP
  :   : THEN-CALL CK BITS
  :   ENDF
  :   SETUP ESR REG CK
  :   IF ESR NOT=ESRCMP
  :   : THEN-CALL CK BITS
  :   ENDF
  ENDF
  GET REGISTER ERR TABLE PTR
  TERMINATE ERROR TABLE
  ENDF
  NOP
ENDSUB
  
```

2689 013664 000240
 2690 013666 005037 014670
 2691 013672 005737 002454
 2692 013676 001160
 2693 013700 004737 014276
 2694 013704 013701 002466
 2695 013710 005721
 2696 013712 100401
 2697 013714 000775
 2698 013716 013702 002470
 2699 013722 006302
 2700 013724 060201
 2701 013726 011103
 2702 013730 012337 014260
 2703 013734 012337 014262
 2704 013740 012337 014264
 2705 013744 011337 014266
 2706 013750 013701 002432
 2707 013754 042701 172027
 2708 013760 043701 014262
 2709 013764 043737 014262 014270
 2710 013772 053737 014270 014260
 2711 014000 023701 014260

```

REGSCK: NOP ;
          CLR RGETPT ;CLEAR REG ERROR TABLE PTR
IARCR: TST FIN ;IF FINI FLAG
        BNE EARCR ;NOT SET, THEN
        CALL SURGCK ;CALL SETUP REGS CHECK
        MOV TSTID,R1 ;GET TEST TABLE ADDRESS
UBRCR: TST (R1)+ ;DO UNTIL TEST TABLE ENTRY
        BMI EBRCR ;EQUALS -1, ADVANCE TEST TABLE ADRS
        BR UBRCR ;END DO 'B'
EBRCR: MOV TCMDCT,R2 ;GET TEST COMMAND CTR
        ASL R2 ;DOUBLE COMMAND CTR
        ADD R2,R1 ;CAL ADRS OF REG TABLE FOR THIS CMD
        MOV (R1),R3 ;GET ADRS FROM TEST TABLE
        MOV (R3)+,CSRCMP ;SET RXCSR COMPARE WORD=TABLE CSR CMP
        MOV (R3)+,CSRMSK ;SET RXCSR MASK WORD=TABLE CSR MSK
        MOV (R3)+,ESRCMP ;SET RXESR COMPARE WORD=TABLE ESR CMP
        MOV (R3),ESRMSK ;SET RXESR MASK WORD=TABLE ESR MSK
CSRCHK: MOV RXCSR,R1 ;GET RXCS
        BIC #172027,R1 ;CLEAR OFF WRITE ONLY BIT-CK DRV SELECT BIT ****
        BIC CSRMSK,R1 ;MASK OFF BITS DON'T CARE ABOUT
        BIC CSRMSK,CSRSET ;MASK OFF CSRSET BITS DON'T CARE
        BIS CSRSET,CSRCMP ;SET CSR COMMAND VARIABLE BITS
        CMP CSRCMP,R1 ;IF RXCS CONTAINS
  
```

```

2712 014004 001437          BEQ      4$          :ERRORS, THEN
2713 014006 013737 014260 002436  MOV     CSRCMP,REGEXP :SAVE EXPECTED
2714 014014 010137 002440          MOV     R1,REGACT    :SAVE ACTUAL
2715 014020 032737 000002 002332  BIT     #FUNCTT,SWREG :IF FUNCTION TEST
2716 014026 001403          BEQ     1$          :MODE, THEN
2717 014030 005237 014274          INC     FTERCT       :INCREMENT ERROR COUNT
2718 014034 000420          BR      3$          :BR TO REST OF SETUP
2719 014036 010137 014702          1$:  MOV     R1,BADWRD   :SET BAD WORD
2720 014042 013737 014260 014700  MOV     CSRCMP,CMPWRD :SET COMPARE WORD
2721 014050 012737 000004 014676  MOV     #4,BITOFF    :SET # BITS TO OFFSET WORD
2722 014056 012737 000014 014674  MOV     #12.,BITLIM  :SET # BITS TO CHECK
2723 014064 012737 014776 014704  2$:  MOV     #CSERTB,RTBADR :SET REG TAB ADR=CSR
2724 014072 004737 014522          CALL   CKBITS       :FIND BAD BITS & RELATED ERR #
2725 014076 012701 100000          3$:  MOV     #ERRFLG,R1  :SET ERR
2726 014102 000401          BR      XCSRCK      :BR TO END
2727 014104 005001          4$:  CLR     R1          :CLEAR ERRORS
2728 014106 050137 002476  XCSRCK: BIS     R1,FLAGST :SET FLAGST ERR BIT-IF ERRORS
2729 014112 013701 002434  ESRCHK: MOV     RXESR,R1  :GET RXES
2730 014116 042701 176000          BIC     #176000,R1   :MASK OFF BITS NOT USED IN RXES
2731 014122 043701 014266          BIC     ESRMSK,R1    :MASK OFF BITS DON'T CARE ABOUT
2732 014126 043737 014266 014272  BIC     ESRMSK,ESRSET :MASK OFF ESRSET BITS DON'T CARE
2733 014134 053737 014272 014264  BIS     ESRSET,ESRCMP :SET ESR COMMAND VARIABLE BITS
2734 014142 023701 014264          CMP     ESRCMP,R1   :IF RXES CONTAINS
2735 014146 001431          BEQ     4$          :ERRORS, THEN
2736 014150 032737 000002 002332  BIT     #FUNCTT,SWREG :IF FUNCTION TEST
2737 014156 001403          BEQ     1$          :MODE, THEN
2738 014160 005237 014274          INC     FTERCT       :INCREMENT ERROR COUNT
2739 014164 000417          BR      3$          :BR TO REST OF SETUP
2740 014166 010137 014702          1$:  MOV     R1,BADWRD   :SET BAD WORD
2741 014172 013737 014264 014700  MOV     ESRCMP,CMPWRD :SET COMPARE WORD
2742 014200 005037 014676          CLR    BITOFF       :SET BIT OFFSET
2743 014204 012737 000014 014674  MOV     #12.,BITLIM  :SET # BITS TO CHECK
2744 014212 012737 014746 014704  2$:  MOV     #ESERTB,RTBADR :SET REG ERR TAB ADR=ESR
2745 014220 004737 014522          CALL   CKBITS       :FIND BAD BITS & RELATED ERR #
2746 014224 012701 100000          3$:  MOV     #ERRFLG,R1  :SET ERR
2747 014230 000401          BR      XESRCK      :BR TO END
2748 014232 005001          4$:  CLR     R1          :CLEAR ERRORS
2749 014234 050137 002476  XESRCK: BIS     R1,FLAGST :SET TEST ERROR FLAG, IF ERRORS
2750 014240 013705 014670  EARCR: MOV     RGETPT,R5 :GET REG ERR TBL PTR
2751 014244 006305          ASL     R5          :DOUBLE REG ERROR TAB PTR FOR ADDRESSING
2752 014246 012765 177777 014706  MOV     #-1,RGERTB(R5) :TERMINATE TBL
2753 014254 000240          NOP                    :
2754 014256 000207          XREGCK: RTS     PC    :RETURN
2755
-----
2756 014260 000000          CSRCMP: 0          :CSR COMPARE WORD
2757 014262 000000          CSRMSK: 0          :CSR MASK WORD
2758 014264 000000          ESRCMP: 0          :ESR COMPARE WORD
2759 014266 000000          ESRMSK: 0          :ESR MASK WORD
2760 014270 000000          CSRSET: 0          :CSR SET - SETUP REGS CK
2761 014272 000000          ESRSET: 0          :ESR SET - SETUP REGS CK
2762 014274 000000          FTERCT: 0          :FUNCTION TEST ERROR COUNTER
2763
-----

```



```

2766 .SBTTL - MOD U.SFT.SRC - SETUP REGISTER CHECK
2767 -----
2768 014276 000240 SURGCK: NOP ;
2769 014300 005037 014272 CLR ESRSET ;CLEAR ESR SET
2770 014304 032737 040000 002400 IGSRC: BIT #RXINIT,CMD ;IF CMD WAS RX INITIALIZE
2771 014312 001406 BEQ EGSRC ;THEN
2772 014314 042737 001000 002400 BIC #SIDE1,CMD ;CLEAR SIDE #1 SELECT BIT
2773 014322 042737 000400 002400 BIC #DRIVE1,CMD ;CLEAR DRIVE #1 SELECT BIT
2774 014330 013705 002400 EGSRC: MOV CMD,R5 ;GET COMMAND
2775 014334 042705 177761 BIC #177761,R5 ;CLEAR ALL BUT COMMAND
2776 014340 022705 000016 IASRC: CMP #16,R5 ;IF COMMAND = READ ERROR CODE
2777 014344 001015 BNE EASRC ;THEN
2778 014346 032737 000200 002500 IFSRC: BIT #RECTST,FLAGSP ;IF FLAGSP NOT=REC TEST
2779 014354 001011 BNE EASRC ;THEN
2780 014356 013737 002424 002400 MOV LCMD,CMD ;SET COMMAND=LAST COMMAND
2781 014364 013737 002426 002432 MOV LRXCSR,RXCSR ;GET LAST RXCSR
2782 014372 013737 002430 002434 MOV LRXESR,RXESR ;GET LAST RXESR
2783 014400 013705 002400 EASRC: MOV CMD,R5 ;GET COMMAND
2784 014404 010537 014270 MOV R5,CSRSET ;SETUP CRS SET
2785 014410 042737 176277 014270 BIC #176277,CSRSET ;SAVE ONLY: SIDE,DENS,INTR ENA,(DRV SEL CK) BITS
2786 014416 032705 001000 IBSRC: BIT #SIDE1,R5 ;IF SIDE #1 SELECTED
2787 014422 001403 BEQ ICSRC ;THEN
2788 014424 052737 001000 014272 BIS #SIDE1,ESRSET ;SETUP ESR SET -> SIDE1
2789 014432 032705 000020 ICSRC: BIT #DRV1,R5 ;IF DRIVE #1 SELECTED
2790 014436 001403 BEQ IDSRC ;THEN
2791 014440 052737 000400 014272 BIS #DRIVE1,ESRSET ;SETUP ESRSET -> DRIVE1
2792 014446 032705 000400 IDSRC: BIT #DENBIT,R5 ;IF DOUBLE DENSITY SELECTED
2793 014452 001403 BEQ EDSRC ;THEN
2794 014454 052737 000040 014272 BIS #DRV DEN,ESRSET ;SETUP ESR SET = DOUBLE DENSITY
2795 014462 042705 177761 EDSRC: BIC #177761,R5 ;CLEAR ALL BUT COMMAND
2796 014466 005737 002402 IESRC: TST DELDAT ;IF DELETED DATA MODE
2797 014472 001411 BEQ EESRC ;AND
2798 014474 022705 000006 CMP #RSCMD,R5 ;COMMAND=READ SECTOR
2799 014500 001403 BEQ 1$ ;OR
2800 014502 022705 000014 CMP #WDDCMD,R5 ;COMMAND-WRITE DELETED DATA SECTOR
2801 014506 001003 BNE EESRC ;THEN
2802 014510 052737 000100 014272 1$: BIS #DLDBIT,ESRSET ;SETUP ESR SET ->DELETED DATA BIT
2803 014516 000240 EESRC: NOP ;
2804 014520 000207 XSRC: RETURN ;RETURN
2805 -----

```



```

2808          .SBTTL - MOD U.SFT.BTK - BITS SET/NOT SET CHECK
2809          ;-----
2810
2811 014522 013702 014700      CKBITS: MOV      CMPWRD,R2      :GET COMPARE WORD
2812 014526 013701 014702      MOV      BADWRD,R1     :GET BAD WORD
2813 014532 040201              BIC      R2,R1         :SET R1=BITS THAT SHOULDN'T BE SET
2814 014534 005102              COM      R2             :COMPLIMENT COMPARE WORD
2815 014536 053702 014702      BIS      BADWRD,R2     :SET BAD BITS
2816 014542 005102              COM      R2             :SET R2=BITS THAT SHOULD BE SET
2817 014544 050201              BIS      R2,R1         :SET R1=ALL BITS THAT SHOULD OR SHOULDN'T BE SET
2818 014546 005737 014676      TST      BITOFF        :IF BIT OFFSET
2819 014552 001407              BEQ      2$            :NOT=0, THEN
2820 014554 005337 014676      1$: DEC      BITOFF        :
2821 014560 000241              CLC                      :CLEAR CARRY
2822 014562 006001              ROR      R1             :
2823 014564 005737 014676      TST      BITOFF        :IF BIT OFFSET
2824 014570 001371              BNE      1$            :EQUALS 0, THEN
2825 014572 005037 014672      2$: CLR      BITCNT        :CLEAR BIT COUNTER
2826 014576 032701 000001      3$: BIT      #1,R1       :IF LSB
2827 014602 001417              BEQ      4$            :NOT=0, THEN
2828 014604 013702 014672      MOV      BITCNT,R2     :GET BIT COUNTER
2829 014610 006302              ASL      R2             :DOUBLE IT FOR ADDRESSING
2830 014612 063702 014704      ADD      RTBADR,R2     :ADD REG TABLE ADR
2831 014616 011203              MOV      (R2),R3       :GET ERR# THIS BIT ERROR FROM TABLE
2832 014620 005703              TST      R3            :IF ERR #
2833 014622 001407              BEQ      4$            :NOT=0, THEN
2834 014624 013704 014670      MOV      RGETPT,R4     :SET UP REG ERR TABLE POINTER
2835 014630 006304              ASL      R4             :DOUBLE IT FOR ADDRESSING
2836 014632 010364 014706      MOV      R3,RGERTB(R4) :SET THIS ERR# IN TABLE OF REG ERRORS
2837 014636 005237 014670      INC      RGETPT        :ADVANCE TABLE POINTER TO NEXT LOCATION
2838 014642 005237 014672      4$: INC      BITCNT        :INCREMENT BIT COUNTER
2839 014646 000241              CLC                      :CLEAR CARRY
2840 014650 006001              ROR      R1             :SHIFT NEXT BIT FOR TEST
2841 014652 023737 014674 014672  CMP      BITLIM,BITCNT  :IF ALL BITS SPECIFIED
2842 014660 101346              BHI      3$            :DONE, THEN
2843 014662 005037 014672      CLR      BITCNT        :RESET BIT COUNT
2844 014666 000207      XCRBIT: RETURN        :RETURN
2845          ;-----
2846 014670 000000      RGETPT: 0              :REG ERROR TABLE POINTER
2847 014672 000000      BITCNT: 0              :BIT COUNTER
2848 014674 000000      BITLIM: 0              :BIT REGISTER LIMIT
2849 014676 000000      BITOFF: 0              :BIT REGISTER OFFSET
2850 014700 000000      CMPWRD: 0              :COMPARE WORD
2851 014702 000000      BADWRD: 0              :BAD WORD
2852 014704 000000      RTBADR: 0              :REGISTER ERROR TABLE ADDRESS
2853          ;-----
  
```


2856
 2857
 2858 014706 000000
 2859 014710 177777
 2860 014712 177777
 2861 014714 177777
 2862 014716 177777
 2863 014720 177777
 2864 014722 177777
 2865 014724 177777
 2866 014726 177777
 2867 014730 177777
 2868 014732 177777
 2869 014734 177777
 2870 014736 177777
 2871 014740 177777
 2872 014742 177777
 2873 014744 177777

REGISTER ERROR #'S - TABLE

```

RGERTB: .WORD 0
         .WORD -1
         .WORD -1
         .WORD -1
         .WORD -1
         .WORD -1
         .WORD -1
         .WORD -1
         .WORD -1
         .WORD -1
         .WORD -1
         .WORD -1
         .WORD -1
         .WORD -1
         .WORD -1
  
```

2874
 2875
 2876
 2877

TABLE - ESR ERROR #'S

```

ESERTB: .WORD CRCERR      ;BIT #00 - CRC ERR
         .WORD SDRDYE      ;BIT #01 - SIDE 1 RDY
         .WORD NOITDP      ;BIT #02 - INIT DONE
         .WORD ACLOWF!NEGTST ;BIT #03 - AC LOW
         .WORD DENERR!NEGTST ;BIT #04 - DEN ERR
         .WORD DENDSK!NEGTST ;BIT #05 - DRV DEN-->NOT ERROR
         .WORD DLDTER      ;BIT #06 - DEL DATA
         .WORD DVRDYE      ;BIT #07 - DRV RDY
         .WORD DRVWRG      ;BIT #08 - UNIT SEL
         .WORD SIDWRG      ;BIT #09 - HEAD SEL
         .WORD WCOVFE!NEGTST ;BIT #10 - WC OVFL
         .WORD NXMERR!NEGTST ;BIT #11 - NXM
  
```

2890
 2891
 2892
 2893

TABLE - CSR ERROR #'S

```

CSERTB: .WORD CSRERR      ;BIT #04 - UNIT SEL      - R/W
         .WORD CSRERR      ;BIT #05 - "DONE"        - R
         .WORD CSRERR      ;BIT #06 - INTER ENB    - R/W
         .WORD CSRERR      ;BIT #07 - "TR"         - R
         .WORD CSRERR      ;BIT #08 - DENSITY      - R/W
         .WORD CSRERR      ;BIT #09 - HEAD SEL    - R/W
         .WORD CSRERR      ;BIT #10 -              -
         .WORD CSRERR      ;BIT #11 - RX02        - R
         .WORD CSRERR      ;BIT #12 -              - W
         .WORD CSRERR      ;BIT #13 -              - W
         .WORD CSRERR      ;BIT #14 -              - W
         .WORD CSRERR      ;BIT #15 - ERR BIT     - R
  
```

2894 014776 000033
 2895 015000 000033
 2896 015002 000033
 2897 015004 000033
 2898 015006 000033
 2899 015010 000033
 2900 015012 000033
 2901 015014 000033
 2902 015016 000033
 2903 015020 000033
 2904 015022 000033
 2905 015024 000033
 2906

```

2909      .SBTTL - PRESETUP REGISTER TABLES
2910      ;-----
2911
2912      TN=0
2913      015026 000000
                REGTB 1,04040,0,0,-1      :RXCS ONLY
                TORT1: .WORD 04040      :RXCSR SHOULD BE
                .WORD 0      :RXCSR DONT CARE
                .WORD 0      :RXESR SHOULD BE
                .WORD -1      :RXESR DONT CARE
2914      015036 004040
                REGTB 2,04040,0,0,0      :RXCS & RXES/ALL
                TORT2: .WORD 04040      :RXCSR SHOULD BE
                .WORD 0      :RXCSR DONT CARE
                .WORD 0      :RXESR SHOULD BE
                .WORD 0      :RXESR DONT CARE
2915      015046 004040
                REGTB 3,04040,0,4,177773 :RXCS & RXES INITIALIZE CK
                TORT3: .WORD 04040      :RXCSR SHOULD BE
                .WORD 0      :RXCSR DONT CARE
                .WORD 4      :RXESR SHOULD BE
                .WORD 177773      :RXESR DONT CARE
2916      015056 004040
                REGTB 4,04040,0,204,1440 :RXCS & RXES INITIALIZE ALL CK
                TORT4: .WORD 04040      :RXCSR SHOULD BE
                .WORD 0      :RXCSR DONT CARE
                .WORD 204      :RXESR SHOULD BE
                .WORD 1440      :RXESR DONT CARE
2917      015066 004040
                REGTB 5,04040,0,200,60   :RXCS & RXES READ STATUS CK
                TORT5: .WORD 04040      :RXCSR SHOULD BE
                .WORD 0      :RXCSR DONT CARE
                .WORD 200      :RXESR SHOULD BE
                .WORD 60      :RXESR DONT CARE
2918      015076 004040
                REGTB 6,04040,0,0,1440   :RXCS & RXES NO DISK OPERATION
                TORT6: .WORD 04040      :RXCSR SHOULD BE
                .WORD 0      :RXCSR DONT CARE
                .WORD 0      :RXESR SHOULD BE
                .WORD 1440      :RXESR DONT CARE
2919      ;-----
2920      015026 CSONLY = TORT1      :RXCS ONLY
2921      015036 CSESAL = TORT2      :RXCS & RXES ALL
2922      015046 CEINIT = TORT3      :RXCS & RXES INITIALIZE CK
2923      015056 CSESIT = TORT4      :RXCS & RXES INITIALIZE ALL
2924      015066 CSESRS = TORT5      :RXCS & RXES READ STATUS CK
2925      015076 CSESND = TORT6      :RXCS & RXES NO DISK OPERATION
2926      ;-----
    
```



```

2929      .SBTTL - MOD U.SET.GEN - GET ERROR CODE-ERR #
2930      ;-----
2931
2932 015106 005002      GTECEN: CLR      R2      ;CLEAR TEMP REG #2
2933 015110 105737 002442  IAGEN:  TSTB     XERUUT   ;IF X ERR CODE UUT
2934 015114 001422      BEQ      XGTECN   ;NOT=0, THEN
2935 015116 122737 000260 002442  IBGEN:  CMPB     #260,XERUUT ;IF ERR CODE UUT
2936 015124 101003      BHI      LBGEN    ;EXCEEDS 260, THEN
2937 015126 012702 000017      MOV     #ILLERC,R2 ;SET ERR CODE #
2938 015132 000407      BR      EBGEN    ;BR TO END 'B'
2939 015134 052737 100000 002476  LBGEN:  BIS     #ERRFLG,FLAGST ;SET FLAGS-ERR FLAG
2940 015142 004737 017106      CALL   GTECOF   ;CALL GET ERROR CODE OFFSET
2941 015146 016102 015164      MOV     ECERNTR(R1),R2 ;GET ERROR CODE ERR # FROM TABLE
2942 015152 010237 002462  EBGEN:  MOV     R2,RECERN ;READ ERR CODE ERR #
2943 015156 010237 020164      MOV     R2,ECERNB ;PASS ERR CODE ERR # TO "ERRCHK" MOD
2944 015162 000207      XGTECN: RETURN   ;RETURN
2945      ;-----
2946
2947      ;      ERROR CODE ERROR # TABLE
2948      ;-----
2949
2950 015164 000000      ECERNTR: .WORD   ; 00->NO ERROR      -
2951 015166 000006      .WORD   SEKERR   ; 10->NO HOME DRVO   -SEEK
2952 015170 000006      .WORD   SEKERR   ; 20->NO HOME DRV1   -SEEK
2953 015172 000017      .WORD   ILLERC   ; 30-> --            -
2954 015174 004041      .WORD   TRKAER!NEGTST ; 40->ACC TK > 76   -TRACK ERR
2955 015176 000006      .WORD   SEKERR   ; 50->HOME BEFORE TRK -SEEK
2956 015200 000017      .WORD   ILLERC   ; 60-> --            -
2957 015202 004003      .WORD   RDERR!NEGTST ; 70->NO SEC-52 TRIES -READ
2958 015204 000017      .WORD   ILLERC   ; 100-> --           -
2959 015206 000003      .WORD   RDERR    ; 110->NO STEP CLOCK -READ
2960 015210 000003      .WORD   RDERR    ; 120->NO PREAMBLE   -READ
2961 015212 000003      .WORD   RDERR    ; 130->PREAMBLE-NO I.D. -READ
2962 015214 000017      .WORD   ILLERC   ; 140-> --            -
2963 015216 000006      .WORD   SEKERR   ; 150->GD TRK NOT=TRK -SEEK
2964 015220 000003      .WORD   RDERR    ; 160->TOO MY TRIES IDAM -READ
2965 015222 000003      .WORD   RDERR    ; 170->DATA AM NOT FND  -READ
2966 015224 000004      .WORD   CRCERR   ; 200->CRC           -CRC
2967 015226 000017      .WORD   ILLERC   ; 210-> --            -
2968 015230 000056      .WORD   HDSFDG   ; 220->SELF DIAG     -SELF DIAG
2969 015232 004051      .WORD   WCOVFE!NEGTST ; 230->WRD COUNT OVF  -WRD CTOV
2970 015234 004030      .WORD   DENERR!NEGTST ; 240->DENSITY ERR   -DEN ERR
2971 015236 004036      .WORD   SDKYWD!NEGTST ; 250->WRG KEYWD-S.D. -WRG KEY
2972      ;-----
  
```

```

2975      .SBTTL - MOD U.PRT.STA - PRINT UNIT STATUS
2976      ;-----
2977
2978 015240 012701 015446      PRTSTA: MOV      #IDENT1,R1      ;SETUP FORMAT MSG
2979 015244 013702 002512      MOV      UNTPRT,R2      ;SETUP UNIT PRT
2980 015250 013703 002432      MOV      RXCSR,R3      ;SETUP RXCSR
2981 015254 013704 002434      MOV      RXESR,R4      ;SETUP RXESR
2982 015260 013705 002400      MOV      CMD,R5      ;SETUP COMMAND
2983 015264 004737 002664      CALL     PRTB4S      ;CALL PRINT BASIC 4-PARM.
2984 015270 005737 002400      IBSTA:  TST      CMD      ;IF CMD
2985 015274 001417      BEQ      IASTA      ;NOT = 0, THEN
2986 015276 032737 040000 002400  ICSTA:  BIT      #BIT14,CMD      ;IF PROG INIT
2987 015304 001403      BEQ      LCSTA      ;THEN
2988 015306 012701 007452      MOV      #CMDM8,R1      ;SETUP PROG INIT MSG
2989 015312 000406      BR      ECSTA      ;BR TO END 'C'
2990 015314 013705 002400      LCSTA:  MOV      CMD,R5      ;GET COMMAND
2991 015320 042705 177761      BIC      #177761,R5      ;CLR ALL BUT CMD
2992 015324 016501 007164      MOV      CMDMSG(R5),R1      ;GET CMD MSG
2993 015330 004737 002550      ECSTA:  CALL     PRTB0S      ;CALL PRINT BASIC 0 - PAR
2994 015334 032737 000200 002476  IASTA:  BIT      #RECFLG,FLAGST      ;IF ERR CODE FLAG
2995 015342 001435      BEQ      XPTSTA      ;SET, THEN
2996 015344 004737 015744      CALL     PRTECD      ;CALL PRINT ERROR CODE
2997 015350 004737 017124      CALL     CLRREGS      ;CALL CLEAR REGISTER
2998 015354 012701 015531      MOV      #XER2,R1      ;SETUP FORMAT MSG
2999 015360 113702 002443      MOV      WC,R2      ;SETUP WORD COUNT
3000 015364 113703 002444      MOV      CTK0,R3      ;SETUP CTK0
3001 015370 113704 002445      MOV      CTK1,R4      ;SETUP CTK1
3002 015374 004737 003002      CALL     PRTX3S      ;CALL PRINT-EXT 3 PARAMETERS
3003 015400 012701 015627      MOV      #XER3,R1      ;SETUP FORMAT MSG
3004 015404 113702 002446      MOV      TTRK,R2      ;SETUP TTRK
3005 015410 113703 002447      MOV      TSEC,R3      ;SETUP TSEC
3006 015414 113704 002450      MOV      SFTSTS,R4      ;SETUP SFTSTS
3007 015420 113705 002451      MOV      BTRK,R5      ;SETUP BTRK
3008 015424 004737 003030      CALL     PRTX4S      ;CALL PRINT-EXT 4 PAR
3009 015430 042737 000200 002476  XPTSTA: BIC      #RECFLG,FLAGST      ;CLEAR ERROR CODE FLAG
3010 015436 005037 015444      CLR      ERRREG      ;CLEAR ERROR REGISTER
3011 015442 000207      RTS      PC      ;RETURN
3012      ;-----
3013 015444 000000      ERRREG: 0      ;
3014      ;-----
3015 015446      045      116      045  IDENT1: .ASCIZ  /%N% UNIT#%01% RXCSR=%0% RXESR=%0% CMD=%0% ->/
3016 015531      045      116      045  XER2:  .ASCIZ  /%N% WORD CNT=%03%N% CUR TRK DV0=%D2%. CUR TRK DV1=%D2%. /
3017 015627      045      116      045  XER3:  .ASCIZ  /%N% TARGET TRK =%D2%. TARGET SEC =%D2%. SOFT STAT=%03% BAD TRK=%D2%.%
3018      .EVEN
3019      ;-----
    
```



```

3022      .SBTTL - MOD U.PRT.EC - PRINT UNIT ERROR CODE
3023      ;-----
3024
3025 015744 012701 016014      PRTECD: MOV      #XER1,R1      ;SETUP FORMAT MSG
3026 015750 113702 002442      MOV      XERUUT,R2      ;GET ERROR CODE
3027 015754 042702 177400      BIC      #177400,R2     ;CLEAR TOP R2
3028 015760 004737 002734      CALL     PRTX1S         ;CALL PRINT EXTENDED-1 ARG
3029 015764 105737 002442      TSTB     XERUUT         ;IF ERROR
3030 015770 001410                BEQ      ENDXER         ;NOT=0, THEN
3031 015772 004737 017106      CALL     GTECOF         ;CALL GET ERROR CODE OFFSET
3032 015776 016101 016042      MOV      ECTAB-2(R1),R1 ;SET ADR OF ERROR MSG FOR PRINT
3033 016002 004737 002714      CALL     PRTXOS         ;CALL PRINT EXTENDED-NO ARG
3034 016006 105037 002442      CLRB     XERUUT         ;CLEAR ERROR CODE
3035 016012 000207                ENDXER: RTS      PC      ;RETURN
3036      ;-----
3037 016014      045      116      045  XER1:  .ASCIZ  /%N%  ERR CODE=%03%  ->/
3038      .EVEN
3039      ;-----
3040
3041 016044 016136      ECTAB:  .WORD  EC1
3042 016046 016175      .WORD  EC2
3043 016050 016116      .WORD  EC0
3044 016052 016234      .WORD  EC4
3045 016054 016274      .WORD  EC5
3046 016056 016116      .WORD  EC0
3047 016060 016337      .WORD  EC7
3048 016062 016116      .WORD  EC0
3049 016064 016407      .WORD  EC11
3050 016066 016451      .WORD  EC12
3051 016070 016477      .WORD  EC13
3052 016072 016116      .WORD  EC0
3053 016074 016567      .WORD  EC15
3054 016076 016640      .WORD  EC16
3055 016100 016670      .WORD  EC17
3056 016102 016716      .WORD  EC20
3057 016104 016116      .WORD  EC0
3058 016106 016753      .WORD  EC22
3059 016110 017013      .WORD  EC23
3060 016112 017033      .WORD  EC24
3061 016114 017052      .WORD  EC25
3062      ;-----

```

3065					.SBTTL - UNIT ERROR CODE MESSAGES
3066					-----
3067	016116	045	101	040	EC0: .ASCIZ /%A ILL ERR CODE/
3068	016136	045	101	116	EC1: .ASCIZ /%ANO HOME ON INITIALIZE DRV 0./
3069	016175	045	101	116	EC2: .ASCIZ /%ANO HOME ON INITIALIZE DRV 1./
3070					;EC3: .ASCIZ /%A ILL ERR CDE./
3071	016234	045	101	124	EC4: .ASCIZ /%A TRIED TO ACCESS A TRACK > 76./
3072	016274	045	101	110	EC5: .ASCIZ /%A HOME FOUND BEFORE DESIRED TRACK./
3073					;EC6: .ASCIZ /%A ILL ERR CDE./
3074	016337	045	101	065	EC7: .ASCIZ /%A 52 HEADERS PASSED & SECTOR NOT FOUND./
3075					;EC10: .ASCIZ /%A ILL ERR CDE./
3076	016407	045	101	116	EC11: .ASCIZ /%ANO STEPCLK SEEN IN 40 MICROSEC./
3077	016451	045	101	120	EC12: .ASCIZ /%A PREAMBLE NOT FOUND./
3078	016477	045	101	120	EC13: .ASCIZ /%A PREAMBLE FOUND BUT NO ID MARK IN TIME./
3079	016550	045	101	111	EC14: .ASCIZ /%A ILL ERR CDE./
3080	016567	045	101	107	EC15: .ASCIZ /%A GOOD HEADER TRACK ADR NOT=SELECTED TRK/
3081	016640	045	101	111	EC16: .ASCIZ /%A IDAM->TOO MANY TRIES./
3082	016670	045	101	116	EC17: .ASCIZ /%ANO DATA AM IN TIME./
3083	016716	045	101	103	EC20: .ASCIZ /%A CRC ERR ON READING SECTOR./
3084					;EC21: .ASCIZ /%A ILL ERR CDE./
3085	016753	045	101	122	EC22: .ASCIZ /%A R-W ELECT. FAILED MAINT. TST./
3086	017013	045	101	127	EC23: .ASCIZ /%A WORD CNT OVFL./
3087	017033	045	101	104	EC24: .ASCIZ /%A DENSITY ERR./
3088	017052	045	101	123	EC25: .ASCIZ /%A SET DENSITY WRG KEY WORD./

3089					;
3090					.EVEN
3091					.SBTTL - MOD U.SFT.GEO - GET ERROR CODE OFFSET
3092					-----
3093					
3094	017106	013701	002442		GTECOF: MOV XERUUT,R1 ;SAVE EXTENDED ERROR CODE IN TEMP #1
3095	017112	006201			ASR R1 ;FORMAT E.C.
3096	017114	006201			ASR R1 ;FORMAT E.C. FOR ADR
3097	017116	042701	177700		BIC #177700,R1 ;CLR TOP BYTE
3098	017122	000207			RETURN ;RETURN
3099					-----

3100					
3101					.SBTTL - MOD U.SFT.CRS - CLEAR REGISTERS
3102					-----
3103					
3104	017124	005001			CLRRGS: CLR R1
3105	017126	005002			CLR R2
3106	017130	005003			CLR R3
3107	017132	005004			CLR R4
3108	017134	005005			CLR R5
3109	017136	000207			RETURN ;RETURN
3110					-----

3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172

.SBTTL - MOD U.SFT.DRC - DEVICE READY CHECK

: BGNSUB
: CALL CLEAR DEVICE
: IF RXES DRV RDY NOT SET [A]
: THEN-SET SYS ERR-DRV RDY ERR
: SETUP ERR # DRV RDY ERR
: CALL ERR
: ENDIF
: ENDSUB

3164 017310 000240
3165 017312 004737 010472
3166 017316 032777 000200 163026
3167 017324 001010
3168 017326 052737 000040 002456
3169 017334 012737 000025 002520
3170 017342 004737 003060
3171 017346 000207

DVRYCK: NOP ;
CALL CLRDEV ;CALL CLEAR DEVICE
IADRC: BIT #DRV RDY, @RXDB ;IF RXDB-DRIVE RDY
BNE EADRC ;NOT SET, THEN
BIS #BIT5, SYSERR ;SET SYS ERR=DRV RDY ERR
MOV #DVRDYE, ERRNBR ;SET ERR NBR=DRV RDY ERROR
CALL ERROR ;CALL ERROR
EADRC: RETURN ;BR TO EXIT

```

3175 .SBTTL - MOD U.SFT.DDC - DEVICE DENSITY CK
3176 -----
3177 : BGNSUB
3178 : CALL DEVICE READY CK
3179 : IF SYS ERR=DEVICE READY ERR NOT SET
3180 : : THEN-SET TRACK=0, SECTOR=10
3181 : : CALL READ SECTOR
3182 : : IF FINI NOT SET [A]
3183 : : : THEN
3184 : : : IF RXES DRIVE DENSITY=DOUBLE DEN [B]
3185 : : : : THEN-SET DENSITY STATUS=DOUBLE DENSITY
3186 : : : : ELSE-SET DENSITY STATUS=SINGLE DENSITY
3187 : : : ENDF
3188 : : SET TRACK=76, SECTOR=10
3189 : : CALL READ SECTOR
3190 : : IF RXES DRIVE DENSITY NOT=DENSITY STATUS [C]
3191 : : : THEN-
3192 : : : : SETUP ERROR # & ERROR MSG=>'DISKETTE-MIXED DENSITY'
3193 : : : : CALL ERROR
3194 : : : : DO DROP UNIT
3195 : : : ENDF
3196 : : ENDF
3197 : ENDSUB
3198 -----
3199
3200 017350 004737 017310 DENCHK: CALL DVRYCK ;CALL DEVICE READY CK
3201 017354 032737 000040 002456 IDDDC: BIT #BITS,SYSERR ;IF SYS ERR=DEVICE RDY ERR
3202 017362 001054 BNE EADDC ;NOT SET, THEN
3203 017364 005037 002374 CLR TRACK ;SET TRACK=0
3204 017370 012737 000012 002376 MOV #10.,SECTOR ;SET SECTOR=10
3205 017376 004737 011062 CALL READ ;CALL READ SECTOR
3206 017402 005737 002454 IADDC: TST FIN ;IF FINI
3207 017406 001042 BNE EADDC ;NOT SET, THEN
3208 017410 032777 000040 162734 IBDDC: BIT #DRV DEN,@RXDB ;IF DRIVE DEN-DOUBLE DEN BIT
3209 017416 001404 BEQ LBDDC ;SET, THEN
3210 017420 012737 000400 002414 MOV #DENBIT,DENSTA ;SET DENSITY STATUS=DOUBLE DEN
3211 017426 000402 BR EBDDC ;BR TO END 'B'
3212 017430 005037 002414 LBDDC: CLR DENSTA ;SET DENSITY STATUS=SINGLE DEN
3213 017434 012737 000114 002374 EBDDC: MOV #76.,TRACK ;SET TRACK=76.
3214 017442 004737 011062 CALL READ ;CALL READ SECTOR
3215 017446 017701 162700 MOV @RXDB,R1 ;GET RXES
3216 017452 042701 177737 BIC #^CDRV DEN,R1 ;CLEAR ALL BUT DRIVE DENSITY
3217 017456 006301 ASL R1 ; ADV DRIVE DENSITY
3218 017460 006301 ASL R1 ; SO EQUAL TO
3219 017462 006301 ASL R1 ; DENSITY STATUS
3220 017464 020137 002414 ICDDC: CMP R1,DENSTA ;IF RXES DRIVE DENSITY & DENSITY STATUS
3221 017470 001411 BEQ EADDC ;NOT=, THEN
3222 017472 012737 000020 002520 MOV #DENDSK,ERRNBR ;SET ERR NBR=DISK DENSITY ERROR
3223 017500 004737 003060 CALL ERROR ;CALL ERROR-MOD
3224 017504 DODU UNIT ;DROP UNIT
3225 017512 DOCLN ;DO CLEAN
3226 017514 000207 EADDC: RETURN ;RETURN
3227 -----
    
```

3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265

```
.SBTTL - MOD U.SFT.TKE - TRACK ERROR CHECK
-----
BGNSUB
: IF LAST COMMAND=READ OR WRITE SECTOR [A]
: THEN-IF FLAG=READ ERROR CODE BIT SET [B]
: : THEN-IF DRIVE #0 SELECTED [C]
: : : THEN-IF CURRENT TRK DRV #0 NOT=TRACK [D]
: : : : THEN-
: : : : IF FLAGS=NEG TST NOT SET [E]
: : : : : THEN-SETUP ERROR #
: : : : : SET PRINT TRACKS-PRINT FLAGS
: : : : : CALL ERROR REPORT
: : : : : ENDIF
: : : : ENDIF
: : : ELSE-IF CURRENT TRK DRV #1 NOT=TRACK [F]
: : : : THEN-
: : : : IF FLAGS=NEG TST NOT SET [G]
: : : : : THEN-SETUP ERROR
: : : : : SET PRINT TRACKS-PRINT FLAGS
: : : : : CALL ERROR REPORT
: : : : : ENDIF
: : : : ENDIF
: : : ELSE-IF ERROR ON COMMAND [H]
: : : : THEN-
: : : : IF FLAGS=NEG TEST NOT SET [I]
: : : : : THEN-SETUP ERR #
: : : : : SET PRINT TRACKS-PRINT FLAGS
: : : : : CALL ERR REPORT
: : : : : ENDIF
: : : : ENDIF
: : : ENDIF
: : ENDIF
: ENDIF
: NOP
: ENDSUB
-----
```



```

3268 017516 000240 TKERCK: NOP
3269 017520 022737 000017 002424 IATKE: CMP #17,LCMD ;IF LAST COMMAND
3270 017526 001471 BEQ EATKE ; WAS
3271 017530 032737 000004 002400 BIT #4,CMD ; READ OR WRITE
3272 017536 001465 BEQ EATKE ; THEN
3273 017540 032737 000200 002476 IBTKE: BIT #RECFLG,FLAGST ;IF FLAGS=READ ERROR CODE BIT
3274 017546 001442 BEQ IHTKE ;SET, THEN
3275 017550 005737 002406 ICTKE: TST DRIVE ;IF DRIVE# 0
3276 017554 001016 BNE IFTKE ;SELECTED, THEN
3277 017556 123737 002444 002374 IDTKE: CMPB CTK0,TRACK ;IF CURRENT TRACK DRIVE 0 & TRACK
3278 017564 001452 BEQ EATKE ;NOT=, THEN
3279 017566 032737 004000 002476 IETKE: BIT #NEGST,FLAGST ;IF FLAGS=NEG TEST BIT
3280 017574 001046 BNE EATKE ;NOT SET, THEN
3281 017576 012737 000041 002520 MOV #TRKAER,ERRNBR ;SET ERR NBR=TRACK ADDRESS ERROR
3282 017604 004737 003060 CALL ERROR ;CALL ERROR
3283 017610 000440 BR EATKE ;BR TO END 'A'
3284 017612 123737 002445 002374 IFTKE: CMPB CTK1,TRACK ;IF CURRENT TRACK DRIVE 1 & TRACK
3285 017620 001434 BEQ EATKE ;NOT=, THEN
3286 017622 032737 004000 002476 IGTKE: BIT #NEGST,FLAGST ;IF FLAGS=NE TEST BIT
3287 017630 001030 BNE EATKE ;NOT SET, THEN
3288 017632 012737 000041 002520 MOV #TRKAER,ERRNBR ;SET ERR NBR=TRACK ADDRESS ERROR
3289 017640 052737 000001 002500 BIS #TKPRT,FLAGSP ;SET PRINT TRACKS FLAG-PROGRAM FLAGS
3290 017646 004737 003060 CALL ERROR ;CALL ERROR
3291 017652 000417 BR EATKE ;BR TO END 'A'
3292 017654 005737 002432 IHTKE: TST RXCSR ;IF ERROR ON COMMAND (READ OR WRITE)
3293 017660 100014 BPL EATKE ;SET, THEN
3294 017662 032737 004000 002476 IITKE: BIT #NEGST,FLAGST ;IF FLAGS=NEG TEST BIT
3295 017670 001010 BNE EATKE ;SET, THEN
3296 017672 012737 000041 002520 MOV #TRKAER,ERRNBR ;SET ERR NBR=TRACK ADDRESS ERROR
3297 017700 052737 000001 002500 BIS #TKPRT,FLAGSP ;SET PRINT TRACKS FLAG
3298 017706 004737 003060 CALL ERROR ;CALL ERROR
3299 017712 000240 EATKE: NOP
3300 017714 042737 000001 002500 BIC #TKPRT,FLAGSP ;CFAR PRINT TRACKS FLAG
3301 017722 000207 XTKECK: RETURN ;RETURN
3302 -----

```

3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337

```
.SBTTL - MOD U.SFT.ECK - ERROR CHECK
-----
BGNSUB
: IF REG CHECK SET [A]
: THEN-CALL REGISTER CHECK
ENDIF
: IF READ ERROR CODE SET [B]
: THEN-IF FLAGSP=READ ERROR CODE TEST NOT SET [N]
: : THEN-CALL READ ERROR CODE CHECK
: : CALL ERROR NEG TEST CK
: ENDIF
ENDIF
: IF ERROR FLAG SET [C]
: THEN
: IF ERR NUMBER NOT SET=SYSFTL ERROR [D]
: : THEN-CLEAR REG ERR #
: : DOWHILE REG ERR # TABLE ENTRY NOT=-1 [E]
: : : SET TEMP R2=REG ERR # TABLE ENTRY
: : : IF TEMP REG #2 > REG ERR # [I]
: : : THEN-SET REG ERR #=TEMP REG
: : : ENDIF
: : ENDDO
: : IF REG ERR # > ERR CODE ERR # [M]
: : : THEN-SET ERR NUMBER=REG ERR #
: : : ELSE-SET ERR NUMBER=ERR CODE ERR #
: : ENDIF
: ENDIF
: CLEAR REG ERR #
: CLEAR ERR CODE ERR #
: CALL ERROR
ENDIF
ENDSUB
-----
```



```

3340 017724 000240 ERRCHK: NOP
3341 017726 032737 000001 002476 IAECK: BIT #REGCK,FLAGST ;IF FLAGS=REG CK BIT
3342 017734 001402 BEQ IBECK ;SET, THEN
3343 017736 004737 013664 CALL REGSCK ;CALL REGISTER CHECK
3344 017742 032737 000200 002476 IBECK: BIT #RECFLG,FLAGST ;IF FLAGS=READ ERROR CODE BIT
3345 017750 001420 BEQ ICECK ;SET, THEN
3346 017752 032737 000200 002500 INECK: BIT #RECTST,FLAGSP ;IF FLAGSP=READ ERROR CODE TEST
3347 017760 001014 BNE ICECK ;NOT SET, THEN
3348 017762 032737 100000 002432 IOECK: BIT #ERRBIT,RXCSR ;IF RXCSR ERR BIT
3349 017770 001410 BEQ ICECK ;SET, THEN
3350 017772 004737 015106 CALL GTECEN ;CALL GET READ ERROR CODE ERR #
3351 017776 013702 020164 MOV ECERNB,R2 ;PASS ERROR CODE ERR # TO "NEG TEST CK" MOD
3352 020002 004737 020170 CALL ERNTCK ;CALL ERROR NEG TEST CHECK
3353 020006 010237 020164 MOV R2,ECERNB ;SAVE REC ERR
3354 020012 032737 100000 002476 ICECK: BIT #ERRFLG,FLAGST ;IF FLAGS=ERROR FLAG
3355 020020 001460 BEQ XERRCK ;SET, THEN
3356 020022 022737 000047 002520 IDECK: CMP #39,ERRNBR ;IF ERR NUMBER NOT=SYS FTL ERR
3357 020030 103434 BLO EDECK ;THEN
3358 020032 005037 020166 CLR RGERNB ;CLEAR REGISTER ERROR #
3359 020036 005001 CLR R1 ;CLEAR REGISTER ERROR TABLE PTR
3360 020040 005761 014706 WEECK: TST RGERTB(R1) ;DOWHILE REG ERR TABLE ENTRY
3361 020044 100413 BMI IMECK ;NOT=-1, THEN
3362 020046 016102 014706 MOV RGERTB(R1),R2 ;PASS REG ERR # TABLE ENTRY TO "NEG TEST CK" MOD VIA "R2"
3363 020052 004737 020170 CALL ERNTCK ;CALL ERROR NEG TEST CHECK
3364 020056 020237 020166 IIECK: CMP R2,RGERNB ;IF TEMP R2 > REG ERR NBR
3365 020062 103402 BLO EIECK ;THEN
3366 020064 010237 020166 MOV R2,RGERNB ;SET REG ERR NUMBER=R2
3367 020070 005721 EIECK: TST (R1)+ ;INCREMENT INDEX
3368 020072 000762 EEECK: BR WEECK ;BR TO DOWHILE 'E'
3369 020074 023737 020166 020164 IMECK: CMP RGERNB,ECERNB ;IF REG ERR# > ERR CODE ERR#
3370 020102 103404 BLO LMECK ;THEN
3371 020104 013737 020166 002520 MOV RGERNB,ERRNBR ;SET ERR NUMBER=REG ERR #
3372 020112 000403 BR EDECK ;BR TO END 'D'
3373 020114 013737 020164 002520 LMECK: MOV ECERNB,ERRNBR ;SET ERR NUMBER=ERR CODE ERR#
3374 020122 000240 EDECK: NOP
3375 020124 032737 020000 002332 IPECK: BIT #BIT13,SWREG ;IF SW REG BIT #13
3376 020132 001402 BEQ EPECK ;SET, THEN
3377 020134 004737 020240 CALL TSTDBG ;**
3378 020140 005037 020166 EPECK: CLR RGERNB ;CLEAR REG ERR #
3379 020144 005037 020164 CLR ECERNB ;CLEAR ERR CODE ERR #
3380 020150 004737 003060 CALL ERROR ;CALL ERROR
3381 020154 042737 000200 002500 BIC #RECFLG,FLAGSP ;CLEAR RD ERR CODE FLG
3382 020162 000207 XERRCK: RETURN ;RETURN
-----
3383
3384 020164 000000 ECERNB: 0 ;ERR CODE ERR #
3385 020166 000000 RGERNB: 0 ;REG ERR #
-----
3386
  
```

```

3389 .SBTTL - MOD U.SFT.ENC - ERROR NEG TEST CHECK
3390 -----
3391 : BGNSUB
3392 : IF TEMP REG #2=NEG TEST FLAG SET [A]
3393 : THEN-CLEAR NEG TEST FLAG FROM ERR #
3394 : IF FLAGS=NEG TEST FLAG SET [B]
3395 : THEN-IF NEG TEST ERR #=SET NEG TEST ERR [C]
3396 : THEN-CLEAR THE ERROR
3397 : ELSE-IF REG #2=DISK ERROR [D]
3398 : THEN-CLEAR-NOT ERROR
3399 : ENDIF
3400 : ENDIF
3401 : ENDIF
3402 : ENDSUB
3403 -----
3404
3405 ERNTCK: NOP ;
3406 020170 000240 IAENC: BIT #NEGTST,R2 ;IF TEMP REG=NEG TEST FLAG
3407 020172 032702 004000 BEQ XENTCK ;SET, THEN
3408 020176 001417 BIC #NEGTST,R2 ;CLEAR NEG TEST FLAG
3409 020200 042702 004000 IBENC: BIT #NEGTST,FLAGST ;IF FLAGS=NEG TEST BIT
3410 020204 032737 004000 002476 BEQ XENTCK ;SET, THEN
3411 020212 001411 ICENC: CMP NGTSER,R2 ;IF NEG TEST ERR # & SET NEG TEST ERR
3412 020214 023702 002464 BNE IDENC ;ARE EQUAL, THEN
3413 020220 001002 CLR R2 ;OK, CLEAR THE ERROR !!
3414 020222 005002 BR XENTCK ;BR TO IF 'I'
3415 020224 000404 IDENC: CMP #DENDSK,R2 ;IF DISK DEN
3416 020226 022702 000020 BNE XENTCK ;ERROR, THEN
3417 020232 001001 CLR R2 ;CLEAR-NOT ERROR<-----
3418 020234 005002 XENTCK: RETURN ;RETURN
3419 020236 000207
3420 -----
  
```



```

3423          .SBTTL - MOD U.SFT.DBG - TEST STATUS
3424          ;-----
3425
3426 020240 013702 002476  TSTDBG: MOV     FLAGST,R2
3427 020244 013703 002500      MOV     FLAGSP,R3
3428 020250 013704 002522      MOV     ERRMSG,R4
3429 020254 012701 020306      MOV     #TSDGMS,R1
3430 020260 004737 002636      CALL    PRTB3S
3431 020264 012701 020364      MOV     #TSDGM1,R1
3432 020270 013702 020166      MOV     RGERNB,R2
3433 020274 013703 020164      MOV     ECERNB,R3
3434 020300 004737 002612      CALL    PRTB2S
3435 020304 000207      RETURN          ;RETURN
3436          ;-----
3437 020306      045      116      045  TSDGMS: .ASCIZ  /%N%A->FLAGST=%OXA FLAGSP=%OXA ERRMSG ADR=%O%N/
3438 020364      045      101      040  TSDGM1: .ASCIZ  /%A REG ERR #%OXA ERR CODE ERR #%O%N/
3439          .EVEN
3440          ;-----
3441          .SBTTL - MOD U.SFT.CDC - COMPLIMENT DENSITY CONTROL
3442          ;-----
3443
3444
3445 020430 000240  CDENC: NOP
3446 020432 005737 002412  IACDC: TST     DENSTY          ;IF CONTROL DENSITY
3447 020436 001406      BEQ     LACDC          ;EQUALS DOUBLE, THEN
3448 020440 042737 000002 002476  BIC     #DDCFLG,FLAGST ;CLEAR DOUBLE DENSITY CONTROL FLAG
3449 020446 005037 002412      CLR     DENSTY          ;SET CONTROL DENSITY=SINGLE
3450 020452 000406      BR     XCDENC          ;BR TO 'X'
3451 020454 012737 000400 002412  LACDC: MOV     #DENBIT,DENSTY ;SET CONTROL DENSITY=DOUBLE
3452 020462 052737 000002 002476  BIS     #DDCFLG,FLAGST ;SET DOUBLE DENSITY CONTROL FLAG
3453 020470 000207  XCDENC: RETURN          ;RETURN
3454          ;-----
3455          .SBTTL - MOD U.SFT.SDC - SETUP DENSITY CONTROL
3456          ;-----
3457
3458
3459 020472 013737 002414 002412  SDENC: MOV     DENSTA,DENSTY ;SET DENSTY CONTROL=DENSITY STATUS
3460 020500 005737 002414  IASDC: TST     DENSTA          ;IF DENSITY STATUS SET TO
3461 020504 001407      BEQ     LASDC          ;DOUBLE DENSITY, THEN
3462 020506 052737 000002 002476  BIS     #DDCFLG,FLAGST ;SET DOUBLE DENSITY CONTROL FLAG
3463 020514 012737 000200 002370  MOV     #128.,WDCNT      ;SET WORD COUNT=128
3464 020522 000406      BR     XSDC           ;BR TO EXIT
3465 020524 042737 000002 002476  LASDC: BIC     #DDCFLG,FLAGST ;CLEAR DOUBLE DENSITY CONTROL FLAG
3466 020532 012737 000100 002370  MOV     #64.,WDCNT      ;SET WORD COUNT=64
3467 020540 000207  XSDC:  RETURN          ;RETURN
3468          ;-----

```

```

3471 .SBTTL - MOD U.PRT.UNT - PRINT UNIT IDENT
3472 -----
3473 :BGNSUB
3474 :   GET UNIT #
3475 :   GET UNIT MSG
3476 :   CALL PRINT-1 ARG
3477 :ENDSUB
3478 -----
3479
3480 020542 013702 002512      MOV      UNTPRT,R2      ;GET UNIT #
3481 020546 012701 020560      MOV      #PTUTMS,R1    ;GET UNIT MSG
3482 020552 004737 002570      CALL     PRTB1S        ;CALL PRINT BASIC-1 ARG
3483 020556 000207              RETURN                 ;RETURN
3484 -----
3485 020560      045      116      045  PTUTMS: .ASCIZ  /%N% UNIT #%D2/
3486 :   .EVEN
3487 :-----
3488
3489 .SBTTL - MOD U.PRT.DID - PRINT DRIVE IDENT
3490 -----
3491 :BGNSUB
3492 :   GET DRIVE #
3493 :   GET SIDE #
3494 :   IF DOUBLE SIDED DEVICE
3495 :   : THEN-SETUP PRINT IDENT DOUBLE SIDED DEVICE
3496 :   :   CALL PRINT BASIC-2 PAR.
3497 :   : ELSE-SETUP PRINT IDENT SINGLE SIDED DEVICE
3498 :   :   CALL PRINT BASIC-1 PAR.
3499 :   ENDIF
3500 :ENDSUB
3501 -----
3502
3503 020600 013702 002514      PRTDID: MOV      DRVPR2,R2      ;SETUP R2=DRV #
3504 020604 012701 020657      MOV      #IDSSMS,R1    ;SETUP PRINT IDENT SINGLE SIDED DEVICE
3505 020610 004737 002570      CALL     PRTB1S        ;CALL PRINT BASIC-1 PAR.
3506 020614 032737 010000 002332  IADID:  BIT      #SIDFLG,SWREG ;IF DOUBLE SIDED DEVICE
3507 020622 001406              BEQ      XPTDID        ;FLAG SET, THEN
3508 020624 013702 002515      MOV      SIDPRT,R2    ;SETUP R3=SID #
3509 020630 012701 020642      MOV      #IDSSMS,R1    ;SETUP PRINT IDENT DOUBLE SIDED DEVICE
3510 020634 004737 002570      CALL     PRTB1S        ;CALL PRINT BASIC-2 PAR.
3511 020640 000207              XPTDID: RETURN         ;RETURN
3512 -----
3513 020642      045      101      040  IDSSMS: .ASCIZ  /%A SIDE #%01/
3514 020657      045      116      045  IDSSMS: .ASCIZ  /%N% DRIVE #%01/
3515 :   .EVEN
3516 :-----
  
```



```

3519
3520
3521
3522
3523
3524
3525
3526
3527
3528 020700 000240
3529 020702 004737 021122
3530 020706 012737 000040 002476
3531 020714 017737 161546 002522
3532 020722 052737 000001 002476
3533 020730 004737 021014
3534 020734 000207
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549 020736 000240
3550 020740 004737 021122
3551 020744 042737 000040 002476
3552 020752 017737 161510 002522
3553 020760 004737 020772
3554 020764 004737 021014
3555 020770 000207
3556
3557
3558
3559
3560 020772 000240
3561 020774 013701 002466
3562 021000 005721
3563 021002 012137 002476
3564 021006 111137 002500
3565 021012 000207
3566
  
```

```

.SBTTL - MOD U.TST.FTS - FUNCTION TEST SETUP
-----
: BGNSUB
:   SET FUNCTION TEST BIT-FLAGS
:   SETUP TEST IDENT MSG IN 'ERRMSG'
:   SET FLAGS REGISTER CHECK
:   NOP
: ENDSUB
-----
FTSTUP: NOP
: CALL CLRCR           ; CALL CLEAR CTRS & REGS
: MOV #FUNTST,FLAGST  ; SET FUNCTION TEST BIT-FLAGS
: MOV @TSTID,ERRMSG   ; SETUP TEST IDENT MSG
: BIS #REGCK,FLAGST   ; SET FLAGS-REGISTER CHECK
: CALL SUDVCD         ; CALL SETUP DEVICE COMMANDS
: RETURN              ; RETURN
-----
.SBTTL - MOD U.TST.LTS - LOGIC TEST SETUP
-----
: BGNSUB
:   CLEAR FUNCTION TEST BIT-FLAGS
:   SETUP TEST IDENT MSG IN 'ERRMSG'
:   GET TEST TABLE ADDRESS
:   INCREMENT TO NEXT ADDRESS
:   SET ANY FLAGS FROM THAT ADDRESS
:   SET FLAGS REGISTER CHECK
:   NOP
: ENDSUB
-----
LTSTUP: NOP
: CALL CLRCR           ; CALL CLEAR CTRS & REGS
: BIC #FUNTST,FLAGST  ; CLEAR FUNCTION TEST BIT-FLAGS
: MOV @TSTID,ERRMSG   ; SETUP TEST IDENT MSG
: CALL SUTSFG         ; CALL SETUP TEST FLAGS
: CALL SUDVCD         ; CALL SETUP DEVICE COMMANDS
: RETURN              ; RETURN
-----
.SBTTL - MOD U.TST.SFG - SETUP TEST FLAGS
-----
SUTSFG: NOP
: MOV TSTID,R1        ; GET TEST TABLE ADDRESS
: TST (R1)+           ; INC TEST TABLE ADDRESS
: MOV (R1)+,FLAGST   ; SET TEST FLAGS FROM TABLE
: MOVB (R1),FLAGSP   ; SET PRINT FLAGS FROM TABLE
: RETURN              ; RETURN
-----
  
```

```

3569      .SBTTL - MOD U.SFT.SDC - SETUP DEVICE COMMANDS
3570      ;-----
3571
3572 021014 012737 036622 002360 SUDVCD: MOV    #DATBUF,EMPADR ;SETUP EMPTY BUFFER ADDRESS
3573 021022 012737 000111 002372      MOV    #'1,VARIFY ;SETUP SET DENSITY KEYWORD='I'
3574 021030 012737 036222 002362      MOV    #DATPAT,FILADR ;SETUP FILL BUFFER ADDRESS
3575 021036 032737 000002 002476 1$: BIT    #DDCFLG,FLAGST ;IF DOUBLE DENSITY FLAGS
3576 021044 001407          BEQ    2$ ;SET, THEN
3577 021046 012737 000400 002412      MOV    #DENBIT,DENSTY ;SET DEVICE DENSITY=DOUBLE
3578 021054 012737 000200 002370      MOV    #128.,WDCNT ;SET WORD COUNT=DOUBLE DEN SIZE
3579 021062 000405          BR/   3$ ;BR
3580 021064 005037 002412 2$: CLR    DENSTY ;SET DEVICE DENSITY=SINGLE
3581 021070 012737 000100 002370      MOV    #64.,WDCNT ;SET WORD COUNT=SINGLE DEN SIZE
3582 021076 012737 002442 002364 3$: MOV    #XERUUT,RECADR ;SET READ ERROR CODE ADR=NORMAL ADR
3583 021104 012737 000001 002374      MOV    #1,TRACK ;SETUP TRACK=1
3584 021112 012737 000001 002376      MOV    #1,SECTOR ;SETUP SECTOR=1
3585 021120 000207          RETURN ;RETURN
  
```

```

3586      ;-----
3587
3588      .SBTTL - MOD U.TST.CCR - CLEAR TEST CTRS & ERROR REGS
3589      ;-----
3590      ; BGNSUB
3591      ; CLEAR ANY ERRORS FROM PREVIOUS TESTS
3592      ; ENDSUB
3593      ;-----
3594
  
```

```

3595 021122 005037 002400 CLRCR: CLR    CMD ;CLEAR COMMAND WORD
3596 021126 005037 002454      CLR    FIN ;CLEAR COMMAND FINI FLAG
3597 021132 005037 002460      CLR    TYPERR ;CLEAR TYPE ERROR
3598 021136 005037 002470      CLR    TCMDCCT ;CLEAR TEST COMMAND CTR
3599 021142 005037 002442      CLR    XERUUT ;CLEAR READ ERR CODE WORD
3600 021146 005037 002510      CLR    TKSCFG ;CLEAR TRK & SEC FLAGS
3601 021152 005037 002402      CLR    DELDAT ;CLEAR DELETED DATA MODE
3602 021156 005037 002504      CLR    TTEMP1 ;CLEAR TEST TEMP #1
3603 021162 000240          NOP ;
3604 021164 000240          NOP ;
3605 021166 000240          NOP ;
3606 021170 000240          NOP ;
3607 021172 000207          RETURN ;RETURN
  
```

```

3608      ;-----
3609
3610      .SBTTL - MOD U.TST.T76 - SET TRACK=76
3611      ;-----
3612
3613 021174 012737 000114 002374 SITK76: MOV    #76.,TRACK ;SET TRACK=76.
3614 021202 012737 000012 002376      MOV    #10.,SECTOR ;SET SECTOR=10.
3615 021210 000207          RETURN ;RETURN
  
```

```

3616      ;-----
3617
3618 021212          ENDMOD
3619
  
```



```
3632 .TITLE MISCELLANEOUS SECTIONS
3633 .SBTTL REPORT CODING SECTION
3661
3662 021212 BGNMOD
3663
3664 :++
3665 : THE REPORT CODING SECTION CONTAINS THE
3666 : "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
3667 :--
3668
3669 021212 BGNRPT
3675
3676 021212 ENDRPT
3677
3688 .EVEN
```

3690
 3691
 3692
 3693
 3694
 3695
 3696
 3697 021214
 3698
 3699 021214 000240
 3704 021216
 3706 021224
 3707 021232
 3708 021234
 3709 021242
 3710 021244 042737 140000 002500
 3711 021252
 3712 021260
 3713 021262
 3714 021264
 3715 021266 052737 000400 002500
 3716 021274 022737 004177 002120
 3717 021302 101007
 3718 021304 052737 010000 002500
 3719 021312 000403
 3720 021314 042737 000400 002500
 3721 021322 052737 100000 002500
 3722 021330 000414
 3723 021332
 3724 021340
 3725 021342 052737 040000 002500
 3726 021350 000404
 3727 021352
 3728 021360
 3729 021362 012737 177777 021526
 3730 021370 062737 000001 021526
 3731 021376 023737 002012 021526
 3732 021404 001426
 3733 021406
 3734 021420
 3735 021422 004737 021742
 3736 021426 004737 021572
 3737 021432
 3738 021460 000414
 3739 021462
 3740 021502 012737 000001 002452
 3741 021510
 3742 021512 000240
 3749 021514 013737 021526 002512
 3760 021522
 3761
 3762 021524 000000
 3763 021526 177777
 3764
 3765 021530 045 116 045
 3766
 3767

```

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

      BGNINIT

INIT:  NOP
      RFLAGS  FLGDRS
      REDEF   #EF.CONTINUE      ;IF CONTINUE
      BCOMPLETE XINIT          ;NOT SET, THEN
      REDEF   #EF.PWR           ;IF POWER FAIL
      BCOMPLETE XINIT          ;NOT SET, THEN
      BIC     #RESFLG!STAFLG,FLAGSP ;CLEAR RESTART & START FLAGS
START: REDEF   #EF.START        ;IF START FLAG
      BNCOMPLETE RESTAR        ;SET, THEN
START0: READBUS                ;IF BUS IS "LSI-BUS"
      BNCOMPLETE UN1           ;THEN
      BIS     #LSIFLG,FLAGSP    ;SET LSI FLAG-PROGRAM FLAGS
      CMP     #4177,L$HIMEM     ;IF HI MEMORY (41776=HI LIMIT 124K)
      BHI     START1            ;IS 124K OR HIGHER, THEN
      BIS     #FONZFG,FLAGSP    ;SET LSI 11/23 FLAG
      BR      START1            ;BR TO "START1"
UN1:   BIC     #LSIFLG,FLAGSP   ;CLEAR LSI FLAG-PROGRAM FLAGS
START1: BIS     #STAFLG,FLAGSP  ;SET START FLAG
      BR      SETUP            ;BR TO "SET UP"
RESTAR: REDEF   #EF.RESTART    ;IF RESTART FLAG
      BNCOMPLETE NEW           ;SET, THEN
      BIS     #RESFLG,FLAGSP    ;SET RESTART FLAG
      BR      SETUP            ;BR TO "SETUP"
NEW:   REDEF   #EF.NEW         ;IF NEW PASS FLAG
      BNCOMPLETE NEXT         ;THEN
SETUP: MOV     #-1,UNIT        ;SETUP TO START GETING UNITS OVER
NEXT:  ADD     #1,UNIT         ;BUMP UNIT TO NEXT UNIT
      CMP     L$UNIT,UNIT      ;IF "DRS" UNIT CNT & DIAG UNIT
      BEQ     INITER           ;NOT EXCEEDED, THEN
      GPHARD  UNIT,PLOC        ;GET NEXT UNIT
      BNCOMPLETE NEXT         ;IF FOUND A UNIT,THEN
      CALL    INTTBL           ;CALL INITIALIZE TABLES
      CALL    UNPKHP           ;UNPACK HARDWARE P-TABLES
      SETVEC  VECT,#INTRHD,#PRIO7
      BR      XINIT
INITER: PRINTF #INTER1        ;PRINT "TOO MANY UNITS"
      MOV     #1,ABORT        ;SET ABORT FLAG
      DOCLN
XINIT: NOP
      MOV     UNIT,UNTPRT      ;SET USER # = LOGICAL UNIT #
      ENDINIT

-----
PLOC:  .WORD  0                ;P-TABLE LOCATION
UNIT:  .WORD -1                ;LOGICAL UNIT# UNDER TEST
-----
INTER1: .ASCIZ /%N%ASTART OVER -> TOO MANY UNITS/
      .EVEN
-----

```



```

3779      .SBTTL - MOD 1.1 - UNPACK HARDWARE P-TABLES
3780      ;-----
3781
3782 021572 013701 021524      UNPKHP: MOV      PLOC,R1          ;SAVE P-TABLE LOCATION
3783 021576 012137 002350      MOV      (R1)+,RXCS        ;LOAD UNIT BUS ADR-CSR
3784 021602 013737 002350 002352  MOV      RXCS,RXDB         ;LOAD UNIT BUS ADR-DBR
3785 021610 062737 000002 002352  ADD      #2,RXDB           ;SET UNIT BUS ADR-DBR
3786 021616 012137 002354      MOV      (R1)+,VECT        ;LOAD UNIT VECTOR
3787 021622 005721      IA11:  TST      (R1)+        ;IF DRIVE #0
3788 021624 001007      BNE     LA11              ;THEN
3789 021626 005037 002406      CLR     DRIVE             ;SETUP TO SELECT DRIVE #0
3790 021632 005037 002420      CLR     DRVOFF            ;SETUP DRIVE BYTE OFFSET DRVO
3791 021636 105037 002514      CLRB   DRVPRT            ;SET PRINT DRV #=0
3792 021642 000411      BR     IB11              ;BR TO IF 'B'
3793 021644 012737 000020 002406  LA11:  MOV      #DRV1,DRIVE  ;SETUP TO SELECT DRIVE #1
3794 021652 012737 000001 002420  MOV      #1,DRVOFF         ;SETUP DRIVE BYTE OFFSET DRV1
3795 021660 112737 000001 002514  MOVB    #1,DRVPRT         ;SET PRINT DRV #=1
3796 021666 005721      IB11:  TST      (R1)+        ;IF SIDE #0 SELECTED
3797 021670 001005      BNE     LB11              ;THEN
3798 021672 005037 002410      CLR     SIDE             ;SETUP TO SELECT SIDE #0
3799 021676 105037 002515      CLRB   SIDPRT            ;SET PRINT SID #=0
3800 021702 000406      BR     EB11              ;BR TO END 'B'
3801 021704 012737 001000 002410  LB11:  MOV      #SIDE1,SIDE  ;SETUP TO SELECT SIDE #1
3802 021712 112737 000001 002515  MOVB    #1,SIDPRT         ;SET PRINT SID #=1
3803 021720 011102      EB11:  MOV      (R1),R2       ;GET DEVICE PRIORITY
3804 021722 116237 021732 002356  MOVB    PRITAB(R2),RXPRI  ;SETUP PROPER DEVICE PRIORITY
3805 021730 000207      RETURN                    ;RETURN
    
```

```

3806      ;-----
3807 021732      000      PRITAB: .BYTE  PRI00          ;PRIORITY 0
3808 021733      040      .BYTE  PRI01          ;PRIORITY 1
3809 021734      100      .BYTE  PRI02          ;PRIORITY 2
3810 021735      140      .BYTE  PRI03          ;PRIORITY 3
3811 021736      200      .BYTE  PRI04          ;PRIORITY 4
3812 021737      240      .BYTE  PRI05          ;PRIORITY 5
3813 021740      300      .BYTE  PRI06          ;PRIORITY 6
3814 021741      340      .BYTE  PRI07          ;PRIORITY 7
    
```

```

3815      ;-----
3816
3817      .SBTTL - MOD 1.2 - INITIALIZE TABLES
3818      ;-----
3819
    
```

```

3820 021742 000240      INTTBL: NOP                    ;
3821 021744 012701 002452  MOV      #ABORT,R1          ;GET ADDRES SOF TABLE TO CLEAR
3822 021750 012702 000010  MOV      #10,R2           ;SET TABLE LENGTH
3823 021754 005021      1$:  CLR     (R1)+          ;CLEAR LOCATOIN
3824 021756 005302      DEC     R2               ;DECREMENT TABLE COUNT
3825 021760 001375      BNE     1$              ;IF DONE, THEN
3826 021762 000207      RETURN                    ;RETURN
3827
    
```

3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3845
3856
3857
3858
3859
3860
3861
3862
3863
3864
3870
3875
3876
3877
3878
3879
3880
3891
3892
3893
3896
3897
3898
3899
3900
3901
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3918
3919
3920
3931
3932
3933

021764
021764
021772
021774

021776
021776 010002
022000 012701 022012
022004 004737 002570
022010

022012 045 116 045

022054
022054 004737 022066
022060

022062
022062 000240
022064

022066

```
.SBTTL CLEANUP CODING SECTION
;
; ++
; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
; --
      BGNCLN
      CLRVEC VECT      ;CLEAR VECTOR
      BRESET          ;BUS RESET
      ENDCLN
      .EVEN
;-----
.SBTTL DROP UNIT SECTION
;
; ++
; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE TO NO
; LONGER BE TESTED.
;-----
      BGNDU
      MOV      R0,R2      ;GET LOGICAL UNIT #
      MOV      #DUMSG1,R1 ;SET DROP MSG
      CALL     PRTB1S     ;CALL PRINTB 1 ARG
      ENDDU
;-----
DUMSG1: .ASCIZ  /%N%A DROP UNIT#%D1%A FROM TEST%N/
;-----
      .EVEN
;-----
.SBTTL AUTO DROP UNIT SECTION
;-----
      BGNAUTO
      CALL     ADRTST     ;CALL ADDRESSING TST
      ENDAUTO
;-----
;-----
.SBTTL ADD UNIT SECTION
;-----
;
; ++
; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
; TO THE TEST CYCLE.
;-----
      BGNAU
      NOP
      ENDAU
;-----
      .EVEN
      ENDMOD
```


3936
 3947
 3983 022066
 3984
 3985
 3986
 3987
 3988
 3989
 3990
 3991
 3992
 3993
 3994
 3995
 3996
 3997
 3998
 3999
 4000
 4001
 4002
 4003
 4004
 4005
 4006
 4007
 4008 022066 000240
 4009 022070 005037 002452
 4010 022074
 4011 022122 017701 160222
 4012 022126
 4013 022134 005737 002452
 4014 022140 001413
 4015 022142 012701 022212
 4016 022146 013702 002350
 4017 022152
 4018 022162
 4019 022170 000207
 4020
 4021 022172 101 104 104
 4022 022212 045 101 040
 4023 022246 045 101 040
 4024
 4025
 4026

```

.TITLE HARDWARE TESTS
      BGNMOD
.SBTTL TEST 0 - ADDRESSING TEST
**
: TEST TO ASSURE THAT THE DEVICE WILL RESPOND WITHOUT A BUS TRAP.
-----
      BGNSUB
        SETUP TEST
        SETUP BUS TRAPS
        READ RXCSR
        RESET BUS TRAPS
        IF TRAP
        :   THEN-SET SYSTEM FATAL FLAG
        :   :   CALL FUNCTION TEST ERROR
        :   :   REPORT BUS TRAP ON RXCSR
        ENDIF
        READ RXDBR
        IF TRAP
        :   THEN-SET SYSTEM FATAL FLAG
        :   :   CALL FUNCTION TEST ERROR
        :   :   REPORT BUS TRAP ON RXDBR
        ENDIF
        RESET BUS TRAPS
      ENDSUB
-----
ADRTST: NOP          :
        CLR          ABORT          :CLEAR ABORT FLAG
        SETVEC      #BTRP4,#TRAP,#PRI07
        MOV         @RXCS,R1        :READ RXCSR
        CLRVEC      #BTRP4
        TST         ABORT          :IF ABORT FLAG
        BEQ         1$              :SET, THEN
        MOV         #TRPMS1,R1      :SET TRAP MESSAGE
        MOV         RXCS,R2        :SET TRAP ADDRESS
        ERRSF      60,TOMSG,PRTB1
        DODU        UNIT
        1$:         RETURN          :RETURN
-----
TOMSG:  .ASCIZ    /ADDRESSING TEST/
TRPMS1: .ASCII    /%A BUS TRAP AT ADDRESS:%06%N/
        .ASCIZ    /%A INTERFACE BAD OR NOT SET TO ABOVE ADDRESS/
        .EVEN
-----

```

- MOD U.SFT.TRP - BUS TRAP HANDLER

```

4028 .SBTTL - MOD U.SFT.TRP - BUS TRAP HANDLER
4029 :++
4030 : FUNCTIONAL DESCRIPTION: SUBR TO HANDLE DEVICE BUS TRAP
4031 : INPUTS: NONE
4032 : IMPLICIT INPUTS: BUS TRAP
4033 : OUTPUTS: ABORT FLAG
4034 : IMPLICIT OUTPUTS: NONE
4035 : SUBORDINATE ROUTINES USED: NONE
4036 : FUNCTIONAL SIDE EFFECTS: NONE
4037 : CALLING SEQUENCE: INTERRUPT
4038 :--

```

```

4040 -----
4041
4042 022324 005237 002452 TRAP: INC ABORT ;SET ABORT FLAG
4043 022330 000002 RTI ;RETURN FROM TRAP INTERRUPT
4044 -----

```

```

4045 :
4046 : TEST SETUP DEFINITIONS
4047 :
4048 :
4049 : FRUS1=0
4050 : TN=0
4051 : FUNCT=1

```



```

4053          .SBTTL TEST 1 - INITIALIZE - FNC TST
          022332 000414          BR          BGNT1          ;BR TO BGN TST
          022334 040          040          111 T1MSG: .ASCIZ / INITIALIZE - FNC TST/
          .EVEN
    
```

```

4054
4055          :++
4056          : TEST TO VERIFY THAT AN RX INITIALIZE WILL RETURN THE DEVICE TO A VALID
4057          : STATE.
4058          :-----
    
```

```

4059          : BGNTST
4060          : IF FUNCTION TEST
4061          : THEN-SETUP TEST I.D.
4062          : CALL FUNCTION TEST SETUP
4063          : BUS INITIALIZE
4064          : CALL ERROR CHECK
4065          : CALL DEVICE STATE CHECK
4066          : INCREMENT COMMAND PTR
4067          : PROGRAM INITIALIZE RX
4068          : CALL ERROR CHECK
4069          : ENDIF
4070          : ENDTST
4071          :-----
    
```

```

4073 022364          TSETUP
          022364 012737 022444 002466 BGNT1: MOV          #T1TBL,TSTID          ;SETUP TEST ID TBL-TEST# 1
          022372 032737 000002 002324 IAT1:  BIT          #FUNCTI,TSTMOD          ;IF TEST MODE=FUNCTION TEST
          022400 001417          BEQ          XT1          ;BIT SET, THEN
          022402 004737 020700          CALL          FTSTUP          ;CALL FUNCTION TEST SETUP
4074 022406          BRESET
4075 022410 004737 011610          CALL          WAIT
4076 022414 004737 012244          CALL          GETREG          ;CALL GET REGS
4077 022420 004737 017724          CALL          ERRCHK          ;CALL ERROR CHECK
4078 022424 004737 017140          CALL          DVSTCK          ;CALL DEVICE CK
4079 022430 004737 010440          CALL          INTIAL          ;CALL PROG INITIALIZE
4080 022434 004737 017724          CALL          ERRCHK          ;CALL ERROR CHECK
4081 022440          XT1:  EXIT          TST
4082 022444          REGTBL CSONLY          REGS1=CSONLY
          015026
4083 022444          TTBL
          022444 022334          T1TBL: .WORD          T1MSG
          022446 177777          .WORD          -1
          022450          T1RTB:
          022450 015026          .WORD          REGS1
          022452 177777          .WORD          -1
4084 022454          ENDTST
    
```

4087 .SBTTL TEST 2 - READ ERROR CODE - FNC TST
 022456 000416 BR BGNT2 ;BR TO BGN TST
 022460 040 122 T2MSG: .ASCIZ / READ ERROR CODE - FNC TST/
 .EVEN

4088
 4089 : **
 4090 : TEST TO VERIFY THAT THE DEVICE WILL COMPLETE A READ ERROR CODE COMMAND
 4091 : WITHOUT ENCOUNTERING AN ERROR.
 4092 :-----

4093 : BGNTST
 4094 : IF FUNCTION TEST
 4095 : THEN-SETUP TEST IDENT
 4096 : CALL FUNCTION TEST SETUP
 4097 : PROGRAM INITIALIZE RX
 4098 : CALL ERROR CHECK
 4099 : SETUP ERROR CODE ADDRESS
 4100 : CALL READ ERROR CODE
 4101 : CALL ERROR CHECK
 4102 : ENDIF
 4103 : ENDTST
 4104 :-----

4106 022514 TSETUP
 022514 012737 022604 002466 BGNT2: MOV #T2TBL,TSTID ;SETUP TEST ID TBL-TEST# 2
 022522 032737 000002 002324 IAT2: BIT #FUNCTI,TSTMOD ;IF TEST MODE=FUNCTION TEST
 022530 001423 BEQ XT2 ;BIT SET, THEN
 022532 004737 020700 CALL FTSTUP ;CALL FUNCTION TEST SETUP
 4107 022536 004737 010440 CALL INTIAL ;CALL PROGRAM INITIALIZE
 4108 022542 004737 017724 CALL ERRCHK ;CALL ERROR CHECK
 4109 022546 012737 002442 002364 MOV #XERUUT,RECADR ;SETUP READ ERROR CODE ADDRESS
 4110 022554 052737 000200 002500 BIS #RECTST,FLAGSP ;SET READ ERROR CODE TEST=FLAGSP
 4111 022562 004737 011340 CALL RDERCD ;CALL READ ERROR CODE
 4112 022566 004737 017724 CALL ERRCHK ;CALL ERROR CHECK
 4113 022572 042737 000200 002500 BIC #RECTST,FLAGSP ;CLEAR READ ERROR CODE TEST=FLAGSP
 4114 022600 XT2: EXIT TST
 4115 022604 REGTBL CSONLY REGS1=CSONLY
 4116 022604 015026 TTBL
 022604 022460 T2TBL: .WORD T2MSG
 022606 177777 .WORD -1
 022610 T2RTB: .WORD REGS1
 022610 015026 .WORD -1
 022612 177777
 4117 022614 ENDTST

4120 .SBITL TEST 3 - FILL BUFFER - FNC TST
 022616 000414 BR BGNT3 ;BR TO BGN TST
 022620 040 106 T3MSG: .ASCIZ / FILL BUFFER - FNC TST/
 .EVEN

4121
 4122 :
 4123 :
 4124 :
 4125 :
 4126 :
 4127 :
 4128 :
 4129 :
 4130 :
 4131 :
 4132 :
 4133 :
 4134 :
 4135 :
 4136 :

```

:++
: TEST TO VERIFY THE DEVICE BUFFER WILL FILL WITH NO RESULTING ERROR.
-----
: BGNTST
: IF FUNCTION TEST
: THEN-SETUP TEST IDENT
:     SETUP DENSITY CONTROL
:     CALL SETUP DEVICE COMMANDS
:     CALL FILL BUFFER
:     NOP
: ENDIF
: ENDTST
-----
    
```

4136 022650 TSETUP
 022650 012737 022720 002466 BGNT3: MOV #T3TBL,TSTID ;SETUP TEST ID TBL-TEST# 3
 022656 032737 000002 002324 IAT3: BIT #FUNCTI,TSTMOD ;IF TEST MODE=FUNCTION TEST
 022664 001413 BEQ XT3 ;BIT SET, THEN
 022666 004737 020700 CALL FTSTUP ;CALL FUNCTION TEST SETUP
 4137 022672 052737 000002 002476 BIS #DDCFLG,FLAGST ;SET DOUBLE DENSITY CONTROL FLAG COMMANDS
 4138 022700 004737 021014 CALL SUDVCD ;CALL SETUP DEVICE COMMANDS
 4139 022704 004737 010510 CALL FILBUF
 4140 022710 004737 017724 CALL ERRCHK ;CALL ERROR CHECK
 4141 022714 XT3: EXIT TST
 4142 022720 REGTBL CSONLY
 015026 REGS1=CSONLY
 4143 022720 TTBL
 022720 022620 T3TBL: .WORD T3MSG
 022722 177777 .WORD -1
 022724 T3RTB:
 022724 015026 .WORD REGS1
 022726 177777 .WORD -1
 4144 022730 ENDTST

4147 .SBTTL TEST 4 - EMPTY BUFFER - FNC TST
 022732 000415 BR BGNT4 ;BR TO BGN TST
 022734 040 040 105 T4MSG: .ASCIZ / EMPTY BUFFER - FNC TST/
 .EVEN

4148
 4149 :++
 4150 : TEST TO VERIFY THE DEVICE BUFFER WILL EMPTY WITHOUT ERRORS.
 4151 -----

4152 BGNTST
 4153 : IF FUNCTION TEST
 4154 : THEN-SETUP TEST IDENT
 4155 : SETUP DENSITY CONTROL
 4156 : CALL SETUP DEVICE COMMANDS
 4157 : CALL EMPTY BUFFER
 4158 : CALL ERROR CHECK
 4159 : NOP
 4160 : ENDIF
 4161 : ENDTST
 4162 -----

4164 022766 TSETUP
 022766 012737 023036 002466 BGNT4: MOV #T4TBL,TSTID ;SETUP TEST ID TBL-TEST# 4
 022774 032737 000002 002324 IAT4: BIT #FUNCTT,TSTMOD ;IF TEST MODE=FUNCTION TEST
 023002 001413 BEQ XT4 ;BIT SET, THEN
 023004 004737 020700 CALL FTSTUP ;CALL FUNCTION TEST SETUP
 4165 023010 052737 000002 002476 BIS #DDCFLG,FLAGST ;SET DOUBLE DENSITY CONTROL FLAG
 4166 023016 004737 021014 CALL SUDVCD ;CALL SETUP DEVICE COMMANDS
 4167 023022 004737 010626 CALL EMPBUF ;CALL EMPTY BUFFER
 4168 023026 004737 017724 CALL ERRCHK ;CALL ERROR CHECK
 4169 023032 XT4: EXIT TST
 4170 023036 REGTBL CSONLY
 015026 REGS1=CSONLY
 4171 023036 TIBL
 023036 022734 T4TBL: .WORD T4MSG
 023040 177777 .WORD -1
 023042 T4RTB:
 023042 015026 .WORD REGS1
 023044 177777 .WORD -1
 4172 023046 ENDTST

4175 .SBTTL TEST 5 - READ STATUS - FNC TST
 023050 000414 BR BGNT5 ;BR TO BGN TST
 023052 040 122 T5MSG: .ASCIZ / READ STATUS - FNC TST/
 .EVEN

4176
 4177 :++
 4178 : TEST TO VERIFY THAT A DEVICE MAINTENANCE READ STATUS (RXES) COMMAND
 4179 : WILL EXECUTE WITHOUT ERROR.

4180 -----
 4181 : BGNTST
 4182 : IF FUNCTION TEST
 4183 : THEN-SETUP TEST IDENT
 4184 : SETUP DENSITY CONTROL=SINGLE
 4185 : CALL SETUP DEVICE COMMANDS
 4186 : CALL READ MAINT STATUS
 4187 : CALL ERROR CHECK
 4188 : NOP
 4189 : ENDIF
 4190 : ENDTST
 4191 -----
 4192

4193 023102 TSETUP
 023102 012737 023154 002466 BGNT5: MOV #T5TBL,TSTID ;SETUP TEST ID TBL-TEST# 5
 023110 032737 000002 002324 IAT5: BIT #FUNCTI,T5TMOD ;IF TEST MODE=FUNCTION TEST
 023116 001414 BEQ XT5 ;BIT SET, THEN
 023120 004737 020700 CALL FTSTUP ;CALL FUNCTION TEST SETUP
 4194 023124 042737 000002 002476 BIC #DDCFLG,FLAGST ;CLEAR DOUBLE DENSITY CONTROL FLAG
 4195 023132 000240 NOP ;
 4196 023134 004737 021014 CALL SUDVCD ;CALL SETUP DEVICE COMMANDS
 4197 023140 004737 011266 CALL RDSTAT ;CALL READ MAINT STATUS
 4198 023144 004737 017724 CALL ERRCHK ;CALL ERROR CHECK
 4199 023150 XT5: EXIT TST
 4200 023154 REGTBL CSONLY
 015026 REGS1=CSONLY
 4201 023154 TTBL
 023154 023052 T5TBL: .WORD T5MSG
 023156 177777 .WORD -1
 023160 T5RTB:
 023160 015026 .WORD REGS1
 023162 177777 .WORD -1
 4202 023164 ENDTST

4205 .SBTTL TEST 6 - FILL & EMPTY BUFFER - FNC TST
 023166 000420 BR BGNT6 ;BR TO BGN TST
 023170 040 106 T6MSG: .ASCIZ / FILL & EMPTY BUFFER - FNC TST/
 .EVEN

4206
 4207 :
 4208 : TEST TO VERIFY THE DEVICE BUFFER DATA IS VALID AFTER A FILL/EMPTY
 4209 : BUFFER COMMAND SEQUENCE.
 4210 -----

4211 : BGNTST
 4212 : IF FUNCTION TEST
 4213 : THEN-SETUP TEST IDENT
 4214 : SETUP DENSITY CONTROL=DOUBLE
 4215 : CALL SETUP DEVICE COMMANDS
 4216 : SET DATA PATTERN=RANDOM
 4217 : CALL DATA PATTERN SETUP
 4218 : CALL FILL BUFFER
 4219 : CALL ERROR CHECK
 4220 : CALL EMPTY BUFFER
 4221 : CALL ERROR CHECK
 4222 : SET EMPTY BUFFER FLAG
 4223 : CALL DATA CHECK
 4224 : ENDIF
 4225 : ENDTST
 4226 -----

4227
 4228 023230 TSETUP
 023230 012737 023340 002466 BGNT6: MOV #T6TBL,TSTID ;SETUP TEST ID TBL-TEST# 6
 023236 032737 000002 002324 IAT6: BIT #FUNCTI,TSTMOD ;IF TEST MODE=FUNCTION TEST
 023244 001433 BEQ XT6 ;BIT SET, THEN
 023246 004737 020700 CALL FTSTUP ;CALL FUNCTION TEST SETUP
 4229 023252 052737 000002 002476 BIS #DDCFLG,FLAGST ;SET DOUBLE DENSITY CONTROL FLAG
 4230 023260 004737 021014 CALL SUDVCD ;CALL SETUP DEVICE COMMANDS
 4231 023264 005037 012660 CLR PAT ;SET DATA PATTERN=RANDOM
 4232 023270 004737 012306 CALL STDATP ;CALL SET DATA PATTERN
 4233 023274 004737 010510 CALL FILBUF ;CALL FILL BUFFER
 4234 023300 004737 017724 CALL ERRCHK ;CALL ERROR CHECK
 4235 023304 004737 010626 CALL EMPBUF ;CALL EMPTY BUFFER
 4236 023310 004737 017724 CALL ERRCHK ;CALL ERROR CHECK
 4237 023314 052737 000020 002476 BIS #EMBUFF,FLAGST ;SET EMPTY BUFFER FLAG
 4238 023322 004737 013246 CALL DATAK ;CALL DATA CHECK
 4239 023326 042737 000020 002476 BIC #EMBUFF,FLAGST ;CLEAR EMPTY BUFFER FLAG
 4240 023334 XT6: EXIT TST
 4241 023340 REGTBL CSONLY REGS1=CSONLY
 015026
 4242 023340 TTBL
 023340 023170 T6TBL: .WORD T6MSG
 023342 177777 .WORD -1
 023344 T6RTB:
 023344 015026 .WORD REGS1
 023346 177777 .WORD -1
 4243 023350 ENDTST

4246 .SBTTL TEST 7 - READ & WRITE SECTOR - FNC TST
023352 000420 BR BGNT7 ;BR TO BGN TST
023354 040 122 T7MSG: .ASCIZ / READ & WRITE SECTOR - FNC TST/
.EVEN

4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277

```

:++
: TEST TO VERIFY THE DEVICE WILL READ AND WRITE IN BOTH DENSITIES WITHOUT
: AN ERROR.
-----
: BGNTST
: IF FUNCTION TEST
: THEN-SETUP TEST IDENT
: CALL DENSITY CHECK
: SETUP TRACK=0, SECTOR=10
: CLEAR ENDDO FLAG
: BGND
: SET DENSITY CONTROL WORD=OPPOSITE DENSITY STATUS
: SET NEGATIVE TEST FLAG
: SETUP EXPECTED DEN ERR
: CALL WRITE SECTOR
: CALL ERROR CK
: CALL READ SECTOR
: CALL ERROR CK
: SET DENSITY CONTROL WORD=DOUBLE DEN
: CALL WRITE SECTOR
: CALL ERROR CK
: CALL READ SECTOR
: CALL ERROR CK
: CALL SET TRACK=76, SECTOR=10
: COMP ENDDO FLAG
: DUNTIL ENDDO FLAG=0
: ENDIF
: ENDTST
-----
```

```

4280 023414          TSETUP
      023414 012737 023602 002466 BGNT7: MOV #T7TBL,TSTID ;SETUP TEST ID TBL-TEST# 7
      023422 032737 000002 002324 IAT7: BIT #FUNCTT,TSTMOD ;IF TEST MODE=FUNCTION TEST
      023430 001462          BEQ XT7 ;BIT SET, THEN
      023432 004737 020700          CALL FTSTUP ;CALL FUNCTION TEST SETUP
4281 023436 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
4282 023442 005037 002374          CLR TRACK ;SET TRACK=0
4283 023446 012737 000012 002376          MOV #10.,SECTOR ;SET SECTOR=10.
4284 023454 005037 002504          CLR TTEMP1 ;CLEAR ENDDO FLAG
4285 023460 000240          BBT7: NOP ;
4286 023462 004737 020472          CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
4287 023466 042737 004000 002476          BIC #NEGTST,FLAGST ;CLEAR NEG TEST FLAG
4288 023474 004737 010744          CALL WRITE ;CALL WRITE SECTOR
4289 023500 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
4290 023504 004737 011062          CALL READ ;CALL READ SECTOR
4291 023510 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
4292 023514 004737 020430          CALL CDENC ;CALL COMPLIMENT DENSITY CONTROL
4293 023520 052737 004000 002476          BIS #NEGTST,FLAGST ;SET NEG TEST FLAG
4294 023526 012737 000030 002464          MOV #DENERR,NGTSER ;SETUP EXPECTED NEG TEST ERR=DEN ERR
4295 023534 004737 010744          CALL WRITE ;CALL WRITE SECTOR
4296 023540 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
4297 023544 004737 011062          CALL READ ;CALL READ SECTOR
4298 023550 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
4299 023554 005137 002504          COM TTEMP1 ;COMPLIMENT ENDDO FLAG
4300 023560 004737 020430          CALL CDENC ;CALL COMPLIMENT DENSITY CONTROL
4301 023564 004737 021174          CALL STTK76 ;CALL SET TRACK=76.
4302 023570 005737 002504          UBT7: TST TTEMP1 ;DOUNTIL ENDDO FLAG
4303 023574 001331          BNE BBT7 ;EQUALS 0
4304 023576          XT7: EXIT TST
4305 023602          REGTBL CSESAL
                                     REGS1=CSESAL
4306 023602          TTBL
      023602 023354          T7TBL: .WORD T7MSG
      023604 177777          .WORD -1
      023606          T7RTB:
      023606 015036          .WORD REGS1
      023610 177777          .WORD -1
4307 023612          ENDTST
    
```


4310 .SBTTL TEST 8 - WRITE SECTOR DELETED DATA - FNC TST
 023614 000423 BR BGNT8 ;BR TO BGN TST
 023616 040 127 T8MSG: .ASCIZ / WRITE SECTOR DELETED DATA - FNC TST/
 .EVEN

4311
 4312
 4313
 4314
 4315
 4316
 4317
 4318
 4319
 4320
 4321
 4322
 4323
 4324
 4325
 4326
 4327
 4328
 4329
 4330
 4331
 4332
 4333

++
 TEST TO VERIFY THAT THE DEVICE WILL WRITE A DELETED DATA MARK ON THE
 DISKETTE WITHOUT ERROR.

```

-----
BGNTST
  IF FUNCTION TEST
  : THEN-SETUP TEST IDENT
  :   SET TRACK=76, SECTOR=10
  :   CALL DENSITY CHECK
  :   SET DELETED DATA FLAG
  :   SET DENSITY CONTROL WORD=DISK DENSITY
  :   CALL WRITE SECTOR
  :   CALL ERROR CK
  :   CALL READ SECTOR SECTOR
  :   CALL ERROR CK
  :   CLEAR DELETED DATA FLAG
  :   CALL WRITE SECTOR
  :   CALL ERROR CK
  ENDIF
ENDTST
    
```

```

-----
TSETUP
BGNT8: MOV #T8TBL,TSTID ;SETUP TEST ID TBL-TEST# 8
IAT8:  BIT #FUNCTI,TSTMOD ;IF TEST MODE=FUNCTION TEST
      BEQ XT8 ;BIT SET, THEN
      CALL FTSTUP ;CALL FUNCTION TEST SETUP
      CALL STTK76 ;CALL SET TRACK=76.
      CALL DENCHK ;CALL DENSITY CHECK
      CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
      MOV #DLDCMD,DELDT ;SETUP DELETED DATA COMMAND MODE
      CALL WRITE ;CALL WRITE SECTOR
      CALL ERRCHK ;CALL ERROR CHECK
      CALL READ ;CALL READ SECTOR
      CALL ERRCHK ;CALL ERROR CHECK
      CLR DELDAT ;CLEAR DELETED DATA COMMAND MODE
      CALL WRITE ;CALL WRITE SECTOR
      CALL ERRCHK ;CALL ERROR CHECK
      NOP ;
XT8:  EXIT TST
      REGTBL CSESAL

REGS1=CSESAL

TTBL
T8TBL: .WORD T8MSG
      .WORD -1
T8RTB: .WORD REGS1
      .WORD -1

ENDTST
    
```

4349 024002

4352 .SBTTL TEST 9 - SET DENSITY - FNC TST
024004 000414 BR BGNT9 ;BR TO BGN TST
024006 040 040 123 T9MSG: .ASCIZ / SET DENSITY - FNC TST/
.EVEN

4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384

```

:++
: TEST TO VERIFY THE DEVICE WILL CHANGE DENSITIES WITHOUT INCURRING AN
: ERROR
-----
: BGNTST
: IF FUNCTION TEST
: THEN-SETUP TEST IDENT
: CALL DENSITY CHECK
: SET DENSITY CONTROL WORD=DISK DENSITY
: CALL SET DENSITY
: CALL ERROR CK
: CALL SET TRACK=76, SECTOR=10
: CALL READ SECTOR
: CALL ERROR CK
: SET TRACK=0
: CALL READ SECTOR
: CALL ERROR CK
: CALL COMPLIMENT DENSITY CONTROL
: CALL SET DENSITY
: CALL ERROR CK
: CALL READ SECTOR
: CALL ERROR CK
: CALL SET TRACK=76., SECTOR=10.
: CALL READ SECTOR
: CALL ERROR CK
: SET DENSITY CONTROL WORD=DISK DENSITY
: CALL SET DENSITY
: CALL ERROR CK
: ENDIF
: ENDTST
-----
```



```

4387 024036          TSETUP
      024036 012737 024210 002466 BGNT9: MOV      #T9TBL,TSTID ;SETUP TEST ID TBL-TEST# 9
      024044 032737 000002 002324 IAT9: BIT      #FUNCTT,TSTMOD ;IF TEST MODE=FUNCTION TEST
      024052 001454          BEQ      XT9          ;BIT SET, THEN
      024054 004737 020700          CALL    FTSTUP      ;CALL FUNCTION TEST SETUP
4388 024060 004737 017350          CALL    DENCHK      ;CALL DENSITY CHECK
4389 024064 004737 020472          CALL    SDENC        ;CALL SET DENSITY CONTROL=DENSITY STATUS
4390 024070 004737 011172          CALL    SETDN        ;CALL SET DENSITY
4391 024074 004737 017724          CALL    ERRCHK      ;CALL ERROR CHECK
4392 024100 004737 021174          CALL    STTK76      ;CALL SET TRACK=76.
4393 024104 004737 011062          CALL    READ         ;CALL READ SECTOR
4394 024110 004737 017724          CALL    ERRCHK      ;CALL ERROR CHECK
4395 024114 005037 002374          CLR     TRACK       ;SET TRACK=0
4396 024120 004737 011062          CALL    READ         ;CALL READ SECTOR
4397 024124 004737 017724          CALL    ERRCHK      ;CALL ERROR CHECK
4398 024130 004737 020430          CALL    CDENC        ;CALL COMPLIMENT DENSITY CONTROL
4399 024134 004737 011172          CALL    SETDN        ;CALL SET DENSITY
4400 024140 004737 017724          CALL    ERRCHK      ;CALL ERROR CHECK
4401 024144 004737 011062          CALL    READ         ;CALL READ SECTOR
4402 024150 004737 017724          CALL    ERRCHK      ;CALL ERROR CHECK
4403 024154 004737 021174          CALL    STTK76      ;CALL SET TRACK=76.
4404 024160 004737 011062          CALL    READ         ;CALL READ SECTOR
4405 024164 004737 017724          CALL    ERRCHK      ;CALL ERROR CHECK
4406 024170 004737 020472          CALL    SDENC        ;CALL SET DENSITY CONTROL=DENSITY STATUS
4407 024174 004737 011172          CALL    SETDN        ;CALL SET DENSITY
4408 024200 004737 017724          CALL    ERRCHK      ;CALL ERROR CHECK
4409 024204          XT9:    EXIT    TST
4410 024210          REGTBL CSESAL          REGS1=CSESAL

      015036          TTBL
4411 024210          024006          T9TBL: .WORD    T9MSG
      024212          177777          .WORD    -1
      024214          015036          T9RTB: .WORD    REGS1
      024216          177777          .WORD    -1
4412 024220          ENDTST
    
```

4415 .SBTTL TEST 10 - POSITIONING - FNC TST
 024222 000414 BR BGNT10 ;BR TO BGN TST
 024224 040 120 117 T10MSG: .ASCIZ / POSITIONING - FNC TST/
 .EVEN

4416
 4417 :++
 4418 : TEST TO VERIFY THE DEVICE WILL CHANGE SECTORS AND TRACKS WITHOUT
 4419 : INCURRING AN ERROR.

```

4420 -----
4421 : BGNTST
4422 : IF FUNCTION TEST
4423 : THEN-SETUP TEST IDENT
4424 : SET TRACK PATTERN=RANDOM
4425 : CALL DENSITY CHECK
4426 : SET DENSITY CONTROL WORD=DRV DENSITY
4427 : BGND0
4428 : CALL GET A TRACK
4429 : CALL GET A SECTOR
4430 : CALL READ SECTOR
4431 : CALL ERROR CK
4432 : DOUNTIL TRACKS DONE FLAG SET
4433 : NOP
4434 : ENDIF
4435 : ENDTST
4436 -----
4437 :--
4438
    
```

```

4438 024254 TSETUP
024254 012737 024352 002466 BGNT10: MOV #T10TBL,TSTID ;SETUP TEST ID TBL-TEST# 10
024262 032737 000002 002324 IAT10: BIT #FUNCTT,TSTMOD ;IF TEST MODE=FUNCTION TEST
024270 001426 BEQ XT10 ;BIT SET, THEN
024272 004737 020700 CALL FTSTUP ;CALL FUNCTION TEST SETUP
4439 024276 012737 000400 002510 MOV #ITK!RTK,TKSCFG ;SET TRK/SEC FLAGS-->TRACK=INIT & RANDOM
4440 024304 004737 017350 CALL DENCHK ;CALL DENSITY CHECK
4441 024310 004737 020472 CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
4442 024314 004737 012662 BBT10: CALL GETTRK ;CALL GET TRACK
4443 024320 004737 013104 CALL GETSEC ;CALL GET SECTOR
4444 024324 004737 011062 CALL READ ;CALL READ SECTOR
4445 024330 004737 017724 CALL ERRCHK ;CALL ERROR CHECK
4446 024334 032737 001000 002476 UBT10: BIT #TRKDON,FLAGST ;DOUNTIL FLAGS->TRACK DONE FLAG
4447 024342 001764 BEQ BBT10 ;SET
4448 024344 000240 NOP ;
4449 024346 XT10: EXIT TST
4450 024352 REGTBL CSESAL REGS1=CSESAL
4451 024352 015036 TTBL
024352 024224 T10TBL: .WORD T10MSG
024354 177777 .WORD -1
024356 T10RTB: .WORD REGS1
024356 015036 .WORD -1
024360 177777 .WORD -1
4452 024362 ENDTST
    
```


4456 .SBTTL TEST 11 - CSR BITS - LGC TST
024364 000412 BR BGNT11 ;BR TO BGN TST
024366 040 103 123 T11MSG: .ASCIZ / CSR BITS - LGC TST/
.EVEN

4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486

```

: **
: TEST TO VERIFY THAT THE READ/WRITE BITS OF THE CONTROL AND STATUS REG-
: ISTER CAN BE WRITTEN INTO AND READ AND OTHERWISE BEHAVE AS EXPECTED.
:-----
: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST TEST IDENT
: CALL RX LEGAL STATE CHECK
: WRITE RXCSR-ALL 1'S (EXCEPT BITS #14 & #1)
: SETUP EXPECTED REGISTER RESULTS
: IF RXCSR DOES NOT=037566
: THEN-SETUP ACTUAL REGISTER RESULTS
: SETUP ERRNBR=CSR ERROR
: CALL ERROR
: ENDIF
: WRITE RXCSR-ALL 0'S
: IF RXCSR DOES NOT=004040
: THEN-SETUP ACTUAL REGISTER RESULTS
: SETUP ERRNBR=CSR ERROR
: CALL ERROR
: ENDIF
: NOP
: ENDIF
: ENDTST
:-----
: BOARD CALLOUT:
: 1. INTERFACE
:-----
:--
```

```

4489 024412          TSETUP
      024412 012737 024602 002466 BGNT11: MOV #T11TBL,TSTID ;SETUP TEST ID TBL-TEST# 11
      024420 032737 000001 002324 IAT11: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      024426 001463          BEQ XT11 ;BIT SET, THEN
      024430 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
4490 024434          BRESET ;BUS RESET
4491 024436 004737 011610          CALL WAIT ;WAIT FOR DONE
4492 024442 004737 017140          CALL DVSTCK ;CALL DEVICE STATE CHECK
4493 024446 012777 137776 155674          MOV #137776,@RXCS ;WRITE RXCSR=ALL 1'S (EXCEPT BIT#14 & #1)
4494 024454 032737 000400 002500 IDT11: BIT #LSIFLG,FLAGSP ;IF LSI FLG - FLAGSP
4495 024462 001404          BEQ LDT11 ;SET, THEN
4496 024464 012737 005560 002436          MOV #5560,REGEXP ;SETUP EXPECTED REG RESULTS = 5560
4497 024472 000403          BR IBT11 ;BR TO IF 'B'
4498 024474 012737 037566 002436 LDT11: MOV #037566,REGEXP ;SETUP EXPECTED REGISTER RESULTS=037566
4499 024502 023777 002436 155640 IBT11: CMP REGEXP,@RXCS ;IF RXCSR NOT=EXPECTED REGISTER
4500 024510 001410          BEQ EBT11 ;THEN
4501 024512 017737 155632 002440          MOV @RXCS,REGACT ;SETUP ACTUAL REGISTER
4502 024520 012737 000033 002520          MOV #CSRERR,ERRNBR ;SET ERRNBR=CSRERR
4503 024526 004737 003060          CALL ERROR ;CALL ERROR
4504 024532 012737 004040 002436 EBT11: MOV #4040,REGEXP ;SETUP EXPECTED REGISTER RESULTS=4040
4505 024540 012777 000000 155602          MOV #0,@RXCS ;WRITE RXCSR=ALL 0'S
4506 024546 023777 002436 155574 ICT11: CMP REGEXP,@RXCS ;IF RXCSR NOT=EXPECTED REGISTER
4507 024554 001410          BEQ XT11 ;THEN
4508 024556 017737 155566 002440          MOV @RXCS,REGACT ;SETUP ACTUAL REGISTER
4509 024564 012737 000033 002520          MOV #CSRERR,ERRNBR ;SET ERRNBR=CSR ERR
4510 024572 004737 003060          CALL ERROR ;CALL ERROR
4511 024576          XT11: EXIT TST
4512 024602          REGTBL
4513 024602          TTP 0, RGPRT
      024602 024366          T11TBL: .WORD T11MSG
      024604 000000          .WORD 0
      024606 000004          .WORD RGPRT
      024610 177777          .WORD -1
      024612          T11RTB:
      024612 177777          .WORD -1
4514 024614          FRUTBL INTONL          T11FTB:
      024614          .WORD INTONL
      024614 006644          .WORD -1
      024616 177777
4515 024620          ENDTST
    
```


4518 .SBTTL TEST 12 - DBR BITS - LGC TST
024622 000412 BR BGNT12 ;BR TO BGN TST
024624 040 104 102 T12MSG: .ASCIZ / DBR BITS - LGC TST/
.EVEN

4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548

++
: TEST TO VERIFY THAT THE READ/WRITE BITS OF THE DATA BUFFER REGISTER
: CAN BE WRITTEN INTO AND READ AS EXPECTED.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: WRITE RXDBR-ALL 1'S
: SETUP EXPECTED REGISTER RESULTS
: IF RXDBR NOT=173767
: THEN-SETUP ACTUAL REGISTER RESULTS
: SETUP ERR NBR=DBR ERR
: CALL ERROR
: ENDIF
: WRITE RXDBR-ALL 0'S
: SETUP EXPECTED REGISTER RESULTS
: IF RXDBR NOT=000000
: THEN-SETUP ACTUAL REGISTER RESULTS
: SET ERRNBR=DBR ERR
: CALL ERROR
: ENDIF
: NOP
: ENDIF
: ENDTST

: BOARD CALLOUT:
: 1. INTERFACE
:-----
:--

```

4551 024650          TSETUP
      024650 012737 025012 002466 BGNT12: MOV #T12TBL,TSTID ;SETUP TEST ID TBL-TEST# 12
      024656 032737 000001 002324 IAT12: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      024664 001450          BEQ XT12 ;BIT SET, THEN
      024666 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
4552 024672          BRESET ;BUS RESET
4553 024674 004737 011610          CALL WAIT ;WAIT FOR DONE
4554 024700 012777 177777 155444          MOV #-1,@RXDB ;WRITE RXDBR-ALL 1'S
4555 024706 012737 173767 002436          MOV #173767,REGEXP ;SETUP EXPECTED REGISTER RESULTS=173767
4556 024714 023777 002436 155430 IBT12: CMP REGEXP,@RXDB ;IF RXDBR NOT=EXPECTED REGISTER
4557 024722 001410          BEQ EBT12 ;THEN
4558 024724 017737 155422 002440          MOV @RXDB,REGACT ;SETUP ACTUAL REGISTER RESULTS
4559 024732 012737 000034 002520          MOV #DBRERR,ERRNBR ;SET ERRNBR=DBR ERR
4560 024740 004737 003060          CALL ERROR ;CALL ERROR
4561 024744 005037 002436          EBT12: CLR REGEXP ;SETUP EXPECTED REGISTER RESULTS=0'S
4562 024750 012777 000000 155374          MOV #0,@RXDB ;WRITE RXDBR=ALL 0'S
4563 024756 023777 002436 155366 ICT12: CMP REGEXP,@RXDB ;IF RXDBR NOT=EXPECTED REGISTER
4564 024764 001410          BEQ XT12 ;THEN
4565 024766 017737 155360 002440          MOV @RXDB,REGACT ;SETUP ACTUAL REGISTER RESULTS
4566 024774 012737 000034 002520          MOV #DBRERR,ERRNBR ;SET ERRNBR=DBR ERR
4567 025002 004737 003060          CALL ERROR ;CALL ERROR
4568 025006          XT12: EXIT TST
4569 025012          TTBL 0,RGPRT
      025012 024624          T12TBL: .WORD T12MSG
      025014 000000          .WORD 0
      025016 000004          .WORD RGPRT
      025020 177777          .WORD -1
      025022          T12RTB:
      025022 177777          .WORD -1
4570 025024          FRUTBL INTONL T12FTB:
      025024          .WORD INTONL
      025024 006644          .WORD -1
      025026 177777
4571 025030          ENDTST
    
```



```

4574      025032 000420      .SBTTL TEST 13 - CSR-DBR COMMON BYTE - LGC TST
          025034 040      103 123 T13MSG: BR BGNT13 ;BR TO BGN TST
          .ASCIZ / CSR-DBR COMMON BYTE - LGC TST/
          .EVEN

```

```

4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604

```

```

:++
: TEST TO VERIFY THAT THE LOWER BYTE OF THE RXCS MAPS INTO THE RXDB AND
: THEREFORE CHECK WRITE ONLY BITS OF THE RXCS.
-----

```

```

: BGNTST
:   IF LOGIC TEST
:   : THEN-SETUP TEST IDENT
:   :   WRITE RXCSR LOW BYTE-ALL 1'S (EXCEPT BIT #1)
:   :   SETUP EXPECTED REGISTER RESULTS
:   :   IF RXDBR LOW BYTE NOT=376
:   :   : THEN-SETUP ACTUAL REGISTER RESULTS
:   :   :   SETUP ERR NBR=CSR ERR
:   :   :   CALL ERROR
:   :   ENDIF
:   :   WRITE RXCSR LOW BYTE-ALL 0'S
:   :   SETUP EXPECTED REGISTER RESULTS
:   :   IF RXDBR LOW BYTE NOT=000
:   :   : THEN-SETUP ACTUAL REGISTER RESULTS
:   :   :   SETUP ERR NBR=CSR ERR
:   :   :   CALL ERROR
:   :   ENDIF
:   ENDIF
: ENDTST
-----

```

```

: BOARD CALLOUT:
: 1. INTERFACE
-----
:--

```

```

4607 025074          TSETUP
      025074 012737 025242 002466 BGNT13: MOV    #T13TBL,TSTID ;SETUP TEST ID TBL-TEST# 13
      025102 032737 000001 002324 IAT13:  BIT    #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      025110 001452          BEQ    XT13      ;BIT SET, THEN
      025112 004737 020736          CALL   LTSTUP   ;CALL LOGIC TEST SETUP
4608 025116          BRESET      ;BUS RESET
4609 025120 004737 011610          CALL   WAIT     ;WAIT FOR DONE
4610 025124 012777 000376 155216          MOV    #376,@RXCS ;WRITE RXCSR LOW BYTE-ALL IF (EXCEPT BIT #1)
4611 025132 012737 000366 002436          MOV    #366,REGEXP ;SETUP EXPECTED REGISTER RESULTS=366
4612 025140 123777 002436 155204 IBT13:  CMPB   REGEXP,@RXDB ;IF RXDBR LOW BYTE NOT=376
4613 025146 001413          BEQ    EBT13   ;THEN
4614 025150 117737 155176 002440          MOVB  @RXDB,REGACT ;SETUP ACTUAL REGISTER RESULTS
4615 025156 042737 177400 002440          BIC   #177400,REGACT ;CLEAR TOP BYTE
4616 025164 012737 000033 002520          MOV    #CSRERR,ERRNBR ;SET ERRNBR=CSR ERR
4617 025172 004737 003060          CALL   ERROR   ;CALL ERROR
4618 025176 005037 002436          EBT13:  CLR    REGEXP   ;SETUP EXPECTED REGISTER RESULTS=0'S
4619 025202 112777 000000 155140          MOVB  #0,@RXCS   ;WRITE RXDBR=ALL 0'S
4620 025210 123777 002436 155134 ICT13:  CMPB   REGEXP,@RXDB ;IF RXDBR NOT=EXPECTED RESULTS
4621 025216 001407          BEQ    XT13
4622 025220 005037 002440          CLR    REGACT   ;SETUP ACTUAL REGISTER RESULTS
4623 025224 012737 000033 002520          MOV    #CSRERR,ERRNBR ;SETUP ERRNBR=CSR ERR
4624 025232 004737 003060          CALL   ERROR   ;CALL ERROR
4625
4626 025236          XT13:  EXIT    TST
4627
4628 025242          TTBL    0, RGPRT
      025242 025034          T13TBL: .WORD  T13MSG
      025244 000000          .WORD  0
      025246 000004          .WORD  RGPRT
      025250 177777          .WORD  -1
      025252          T13RTB: .WORD  -1
      025252 177777          .WORD  -1
4629 025254          FRUTBL  INTONL T13FTB: .WORD  INTONL
      025254          .WORD  -1
      025254 006644
      025256 177777
4630 025260          ENDTST
  
```


4633 .SBTTL TEST 14 - BUS INITIALIZE - LGC TST
025262 000415 BR BGNT14 ;BR TO BGN TST
025264 040 102 125 T14MSG: .ASCIZ / BUS INITIALIZE - LGC TST/
.EVEN

4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657

```

:++
: TEST TO VERIFY THAT THE INTERFACE BOARD WILL COMPLETE A BUS INITIALIZE.
:-----
: BGNTST
:   IF LOGIC TEST
:   : THEN-SETUP TEST IDENT
:   :   ISSUE BUS INITIALIZE
:   :   IF RXCSR ERROR BIT SET
:   :   : THEN-IF RXESR AC LOW BIT SET
:   :   :   THEN-SETUP ERROR
:   :   :   CALL ERROR
:   :   ENDIF
:   : ENDIF
:   : NOP
:   ENDIF
: ENDTST
:-----
: BOARD CALLOUT:
:   1. INTERFACE
:   2. CONTROLLER
:-----
:--
```

```

4660
4661 025316          TSETUP
      025316 012737 025462 002466 BGNT14: MOV    #T14TBL,TSTID ;SETUP TEST ID TBL-TEST# 14
      025324 032737 000001 002324 IAT14:  BIT    #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      025332 001451          BEQ    XT14      ;BIT SET, THEN
      025334 004737 020736          CALL   LTSTUP   ;CALL LOGIC TEST SETUP
4662 025340          BRESET
4663 025342 032777 100000 155000 IBT14:  BIT    #ERRBIT,@RXCS ;IF RXCSR ERROR BIT
4664 025350 001442          BEQ    XT14      ;SET, THEN
4665 025352 032777 000004 154772 ICT14:  BIT    #INITDN,@RXDB ;IF RXESR=INIT DONE
4666 025360 001406          BEQ    IDT14     ;SET, THEN
4667 025362 012737 000061 002520          MOV    #NOITDB,ERRNBR ;SET ERR NBR=NO INIT DONE-BUS
4668 025370 004737 003060          CALL   ERROR    ;CALL ERROR
4669 025374 000430          BR     XT14      ;BR TO EXIT
4670 025376 032777 000030 154746 IDT14:  BIT    #DENERR,@RXDB ;IF RXESR=DENSITY ERR
4671 025404 001406          BEQ    IET14     ;SET, THEN
4672 025406 012737 000020 002520          MOV    #DENDSK,ERRNBR ;SET ERR NBR=DISK DEN ERR
4673 025414 004737 003060          CALL   ERROR    ;CALL ERROR
4674 025420 000416          BR     XT14      ;BR TO EXIT
4675 025422 032777 000010 154722 IET14:  BIT    #ACLOW,@RXDB ;IF RXESR NOT=INITIALIZE DONE BIT
4676 025430 001006          BNE    LET14     ;SET, THEN
4677 025432 012737 000050 002520          MOV    #ACLOWF,ERRNBR ;SET ERR NBR=NO INIT DONE-BIT
4678 025440 004737 003060          CALL   ERROR    ;CALL ERROR
4679 025444 000404          BR     XT14      ;BR TO EXIT
4680 025446 004737 011340          LET14: CALL   RDERCD   ;CALL READ ERROR CODE
4681 025452 004737 017724          XT14: CALL   ERRCHK   ;CALL ERROR CHECK
4682 025456          XT14:  EXIT    TST
4683 025462          REGTBL
4684 025462          TTBL   REGCK,0
      025462 025264          T14TBL: .WORD  T14MSG
      025464 000001          .WORD  REGCK
      025466 000000          .WORD  0
      025470 177777          .WORD  -1
      025472          T14RTB: .WORD  -1
      025472 177777          .WORD  -1
4685 025474          FRUTBL INFCTL T14FTB: .WORD  INFCTL
      025474          .WORD  -1
      025474 006640
      025476 177777
4686 025500          ENDTST
    
```



```

4689      025502 000421          .SBTTL TEST 15 - PROGRAMMED INITIALIZE - LGC TST
          025504 040          120 122 T15MSG: BR BGNT15 ;BR TO BGN TST
          .ASCIZ / PROGRAMMED INITIALIZE - LGC TST/
          .EVEN

4690
4691      :++
4692      : TEST TO VERIFY THAT THE INTERFACE BOARD WILL COMPLETE A PROGRAMMED
4693      : INITIALIZE.
4694      :-----
4695      : BGNTST
4696      : IF LOGIC TEST
4697      : THEN-SETUP TEST IDENT
4698      : CALL PROGRAMMED INITIALIZE
4699      : CALL DEVICE STATE CK
4700      : CALL ERROR CHECK
4701      : NOP
4702      : ENDIF
4703      : ENDTST
4704      :-----
4705      : BOARD CALLOUT:
4706      : 1. INTERFACE
4707      : 2. CONTROLLER
4708      :-----
4709      :--
4710
4711      025546          TSETUP
          025546 012737 025610 002466 BGNT15: MOV #T15TBL,TSTID ;SETUP TEST ID TBL-TEST# 15
          025554 032737 000001 002324 IAT15: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
          025562 001410          BEQ XT15 ;BIT SET, THEN
          025564 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
4712 025570 004737 010440          CALL INTIAL ;CALL PROG INITIALIZE
4713 025574 004737 017140          CALL DVSTCK ;CALL DEVICE STATE CK
4714 025600 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
4715 025604          XT15: EXIT TST
4716
4717 025610          REGTBL CSESIT
          015056          REGS1=CSESIT
4718 025610          TTBL REGCK,0
          025610 025504          T15TBL: .WORD T15MSG
          025612 000001          " .WORD REGCK
          025614 000000          .WORD 0
          025616 177777          .WORD -1
          025620          T15RTB:
          025620 015056          .WORD REGS1
          025622 177777          .WORD -1
4719 025624          FRUTBL INFCTL
          025624          T15FTB:
          025624 006640          .WORD INFCTL
          025626 177777          .WORD -1
4720 025630          ENDTST
    
```

4723 .SBTTL TEST 16 - POWER FAIL - LGC TST
025632 000413 BR BGNT16 ;BR TO BGN TST
025634 040 120 117 T16MSG: .ASCIZ / POWER FAIL - LGC TST/
.EVEN

4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753

```

:++
: TEST TO VERIFY THAT THE ACLOW CIRCUITS OPERATE AS EXPECTED.
-----
: BGNTST
:   IF LOGIC TEST [A]
:   : THEN-SETUP TEST IDENT
:   :   IF MANUAL INTERVENTION ALLOWED [B]
:   :   : THEN-ASK OPERATOR TO POWER DOWN RX02 ONLY
:   :   :   IF OPERATION COMPLETE [C]
:   :   :   : THEN-CALL PROGRAMMED INITIALIZE
:   :   :   :   SETUP EXPECTED ERROR=AC LOW
:   :   :   :   SET NEG TEST FLAG=TEST FLAGS
:   :   :   :   CALL ERROR CHECK
:   :   :   :   ASK OPERATOR TO POWER UP RX02
:   :   :   :   IF OPERATION COMPLETE [D]
:   :   :   :   : THEN-CLEAR OUT EXPECTED ERROR
:   :   :   :   :   CLEAR NEG TEST FLAG=TEST FLAGS
:   :   :   :   :   CALL INITIAL
:   :   :   :   :   CALL ERROR CHECK
:   :   :   :   ENDIF
:   :   :   ENDIF
:   :   ENDIF
: ENDTST
-----
: BOARD CALLOUT:
:   1. INTERFACE
-----
:--
```



```

4756 025662          TSETUP
      025662 012737 026102 002466 BGNT16: MOV      #T16TBL,TSTID ;SETUP TEST ID TBL-TEST# 16
      025670 032737 000001 002324 IAT16:  BIT      #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      025676 001477          BEQ      XT16      ;BIT SET, THEN
      025700 004737 020736          CALL     LTSTUP    ;CALL LOGIC TEST SETUP
4757 025704 005037 002504          CLR      TTEMP1   ;SET TEMP1=0
4758 025710          IBT16:  MANUAL          ;IF MANUAL INTERVENTION
4759 025712          BNCOMPLETE XT16    ;ALLOWED, THEN
4760 025714          PRINTF  #PWRMS,UNIT ;PRINT MSG
4761 025740          GMANIL  PWDNRY,TTEMP1,1,YES
4762 025754 032737 000001 002504 ICT16:  BIT      #1,TTEMP1 ;IF RX02 IS
4763 025762 001445          BEQ      XT16      ;POWERED DOWN, THEN
4764 025764 004737 010440          CALL     INTIAL   ;CALL INITIALIZE
4765 025770 012737 000050 002464          MOV      #ACLOWF,NGTSER ;SETUP EXPECTED ERROR=AC LOW
4766 025776 052737 004000 002476          BIS      #NEGTST,FLAGST ;SET NEG TEST FLAG=TEST FLAGS
4767 026004 004737 017724          CALL     ERRCHK   ;CALL ERROR CHECK
4768 026010          PRINTF  #PWRMS,UNIT ;PRINT MSG
4769 026034          GMANIL  PWUPRY,TTEMP1,1,YES
4770 026050 032737 000002 002504 IDT16:  BIT      #2,TTEMP1 ;IF RX02 IS
4771 026056 001407          BEQ      XT16      ;POWERED UP, THEN
4772 026060 042737 004000 002476          BIC      #NEGTST,FLAGST ;CLEAR NEG TEST FLAG=TEST FLAGS
4773 026066 004737 010440          CALL     INTIAL   ;CALL INITIALIZE
4774 026072 004737 017724          CALL     ERRCHK   ;
4775 026076          XT16:  EXIT      TST
4776 026102          REGTBL  CSESIT
      015056          REGS1=CSESIT
4777 026102          TTBL    REGCK,0
      026102 025634          T16TBL: .WORD   T16MSG
      026104 000001          .WORD   REGCK
      026106 000000          .WORD   0
      026110 177777          .WORD   -1
      026112          T16RTB:
      026112 015056          .WORD   REGS1
      026114 177777          .WORD   -1
4778 026116          FRUTBL  INTONL
      026116          T16FTB:
      026116 006644          .WORD   INTONL
      026120 177777          .WORD   -1
4779
4780 026122          045      116      045  PWRMS:  .ASCIZ  /%NZA IS FLOPPY SYSTEM CONTAINING UNIT #%02/
4781 026175          040      040      120  PWDNRY: .ASCIZ  / POWERED DOWN/
4782 026214          040      040      120  PWUPRY: .ASCIZ  / POWERED UP/
4783
4784
4785 026232          ENDTST

```

```

4788      026234 000420      .SBTTL TEST 17 - CONTROLLER-INTERFACE - LGC TST
          026236 040      103 117 T17MSG: BR      BGN17      ;BR TO BGN TST
          .ASCIZ / CONTROLLER-INTERFACE - LGC TST/
          .EVEN

```

```

4789
4790      :++
4791      : TEST TO VERIFY THAT THE INTERFACE BOARD STATE SEQUENCER IS FUNCTIONAL.
4792      : ALSO TO VERIFY THE CONTROLLER-INTERFACE HANDSHAKE BY TRYING FUNCTIONS
4793      : WITH MINIMUM READ/WRITE BOARD INVOLVEMENT.

```

```

4794      :-----
4795      : BGN17
4796      : IF LOGIC TEST
4797      : THEN-SETUP TEST IDENT
4798      : SET PROTOCOL CHECK (TEST SETUP SETS VIS TEST TABLE)
4799      : CALL READ ERROR CODE
4800      : CALL ERROR CHECK
4801      : CALL FILL BUFFER
4802      : CALL ERROR CHECK
4803      : CALL EMPTY BUFFER
4804      : CALL ERROR CHECK
4805      : CALL READ MAINT STATUS
4806      : CALL ERROR CHECK
4807      : ENDIF
4808      : ENDTST

```

```

4809      :-----
4810      : BOARD CALLOUT:
4811      : 1. CONTROLLER
4812      : 2. INTERFACE
4813      :-----
4814      :--

```



```

4817 026276          TSETUP          ;
      026276 012737 026404 002466 BGNT17: MOV #T17TBL,TSTID ;SETUP TEST ID TBL-TEST# 17
      026304 032737 000001 002324 IAT17: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      026312 001432          BEQ XT17 ;BIT SET, THEN
      026314 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
4818 026320 052737 000200 002500          BIS #RECTST,FLAGSP ;SET READ ERROR CODE TEST=FLAGSP
4819 026326 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
4820 026332 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
4821 026336 042737 000200 002500          BIC #RECTST,FLAGSP ;CLEAR READ ERROR CODE TEST=FLAGSP
4822 026344 004737 010510          CALL FILBUF ;CALL FILL BUFFER
4823 026350 004737 017724          CALL ERRCHK ;CALL ERROR HECK
4824 026354 004737 010626          CALL EMPBUF ;CALL EMPTY BUFFER
4825 026360 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
4826 026364 005237 002470          INC TCMDCT ;INCREMENT TST COMMAND CTR *****
4827 026370 004737 011266          CALL RDSTAT ;CALL READ MAINTENANCE STATUS
4828 026374 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
4829 026400          XT17: EXIT TST
4830
4831 026404          REGTBL CSESND,CSESRS          REGS1=CSESND
      015076          REGS2=CSESRS
      015066
4832 026404          TTBL REGCK,PROPRT
      026404 026236          T17TBL: .WORD T17MSG
      026406 000001          .WORD REGCK
      026410 000010          .WORD PROPRT
      026412 177777          .WORD -1
      026414          T17RTB:
      026414 015076          .WORD REGS1
      026416 015066          .WORD REGS2
      026420 177777          .WORD -1
4833 026422          FRUTBL CTLINF          T17FTB:
      026422          .WORD CTLINF
      026422 006646          .WORD -1
      026424 177777
4834 026426          ENDTST
    
```

4837 026430 000410 .SBTTL TEST 18 - NPR - LGC TST
026432 040 116 120 T18MSG: BR BGNT18 ;BR TO BGN TST
.ASCIZ / NPR - LGC TST/
.EVEN

4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875

;++
: TEST TO VERIFY THAT THE NPR LOGIC WILL STORE A WORD IN MEMORY.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: SET ERROR CODE STORAGE=1'S
: CALL READ ERROR CODE
: IF ERROR CODE STORAGE=1'S
: THEN-CALL ERROR
: ENDIF
: SET WORD COUNT=128.
: SET DATA PATTERN=0'S
: CALL SET DATA PATTERN
: DATA BUFFER AREA=1'S (BEGIN, END & END+1)
: SET DENSITY CONTROL=DOUBLE DENSITY
: CALL FILL BUFFER
: CALL LOGIC TEST ERROR CK
: CALL EMPTY BUFFER
: CALL ERROR CK
: IF BEGIN DATA BUFFER AREA NOT=0'S
: THEN-SETUP NPR ERROR
: CALL ERROR
: ENDIF
: IF END DATA BUFFER AREA NOT=0'S
: THEN-CALL NPR ERROR
: CALL ERROR
: ENDIF
: IF END+1 DATA BUFFER NOT=1'S
: THEN-CALL NPR ERROR
: CALL ERROR
: ENDIF
: ENDIF
: ENDTST

--


```

4878 026452          TSETUP
      026452 012737 026752 002466 BGNT18: MOV #T18TBL,TSTID ;SETUP TEST ID TBL-TEST# 18
      026460 032737 000001 002324 IAT18: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      026466 001527          BEQ XT18 ;BIT SET, THEN
      026470 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
4879 026474 012737 177777 002442          MOV #-1,XERUUT ;SET READ ERROR CODE STORAGE=1'S
4880 026502 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
4881 026506 022737 177777 002442 IBT18: CMP #-1,XERUUT ;IF READ ERROR CODE STORAGE=1'S
4882 026514 001005          BNE EBT18 ;THEN
4883 026516 012737 000053 002520          MOV #NPRERR,ERRNBR ;SET ERR NUMBER=NPR ERROR
4884 026524 004737 003060          CALL ERROR ;CALL ERROR
4885 026530 042737 000200 002476 EBT18: BIC #RECFLG,FLAGST ;CLEAR RED ERR COD FLG = FLAGS TST
4886 026536 012737 000200 002370          MOV #128.,WDCNT ;SET DEVICE WORD COUNT=128
4887 026544 012737 000001 012660          MOV #1,PAT ;SET DATA PAT=ALL ZEROS
4888 026552 004737 012306          CALL STDATP ;CALL SET DATA PATTERN
4889 026556 012702 177777          MOV #-1,R2 ;SET R2=ALL 1'S
4890 026562 013737 002370 002504          MOV WDCNT,TTEMP1 ;SET TTEMP1=WORD COUNT
4891 026570 006337 002504          ASL TTEMP1 ;DOUBLE IT FOR ADDRESSING WORDS IN MEM
4892 026574 162737 000004 002504          SUB #4,TTEMP1 ;ADJUST TO END OF BUFFER
4893 026602 013701 002504          MOV TTEMP1,R1 ;SET R1=TEMP1
4894 026606 010237 036622          MOV R2,DATBUF ;SET DATA BUFFER BEGIN=1'S
4895 026612 110261 036622          MOV R2,DATBUF(R1) ;SET DATA BUFFER END=1'S
4896 026616 005201          INC R1 ;BUMP INDEX
4897 026620 110261 036622          MOV R2,DATBUF(R1) ;SET DATA BUFFER END +1=1'S
4898 026624 012737 000400 002412          MOV #DENBIT,DENSTY ;SET DENSITY CONTROL=DOUBLE DENSITY
4899 026632 004737 010510          CALL FILBUF ;CALL FILL BUFFER
4900 026636 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
4901 026642 004737 010626          CALL EMPBUF ;CALL EMPTY BUFFER
4902 026646 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
4903 026652 005737 036622          ICT18: TST DATBUF ;IF DATA BUFFER BEGIN
4904 026656 001406          BEQ ECT18 ;NOT=0, THEN
4905 026660 012737 000053 002520          MOV #NPRERR,ERRNBR ;SETUP ERRNBR=NPR ERR
4906 026666 004737 003060          CALL ERROR ;CALL ERROR
4907 026672 000425          BR XT18 ;BR TO EXIT
4908 026674 013701 002504          ECT18: MOV TTEMP1,R1 ;SET R1=TEMP1
4909 026700 105761 036622          IDT18: TSTB DATBUF(R1) ;IF DATA BUFFER END
4910 026704 001406          BEQ EDT18 ;NOT=0, THEN
4911 026706 012737 000053 002520          MOV #NPRERR,ERRNBR ;SETUP ERRNBR=NPR ERR
4912 026714 004737 003060          CALL ERROR ;CALL ERROR
4913 026720 000412          BR XT18 ;BR TO EXIT
4914 026722 005201          EDT18: INC R1 ;BUMP INDEX
4915 026724 126127 036622 177777          CMPB DATBUF(R1),#-1 ;IF DATA BUFFER END +1
4916 026732 001405          BEQ XT18 ;NOT=1'S, THEN
4917 026734 012737 000053 002520          MOV #NPRERR,ERRNBR ;SETUP ERRNBR=NPR ERR
4918 026742 004737 003060          CALL ERROR ;CALL ERROR
4919 026746          XT18: EXIT TST
4920 026752          REGTBL CSESND
      015076          REGS1=CSESND
4921 026752          TTBL REGCK,0
      026752 026432          T18TBL: .WORD T18MSG
      026754 000001          .WORD REGCK
      026756 000000          .WORD 0
      026760 177777          .WORD -1
      026762          T18RTB:
      026762 015076          .WORD REGS1
      026764 177777          .WORD -1
4922 026766          FRUTBL INFCTL
    
```

026766
026766 006640
026770 177777
4923 026772

ENDTST

T18FTB:
.WORD INFCTL
.WORD -1

4973 .SBTTL TEST 19 - NPR NON-EXISTENT MEM - LGC TST
027104 000420 BR BGNT19 ;BR TO BGN TST
027106 040 116 120 T19MSG: .ASCIZ / NPR NON-EXISTENT MEM - LGC TST/
.EVEN

4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998

```

: **
: TEST TO VERIFY THAT THE NPR NON-EXISTEND MEMORY LOGIC WILL TIME OUT
: WHEN GIVEN AN ILLEGAL ADDRESS.
-----
: BGNTST
:   IF LOGIC TEST
:   : THEN-SETUP TEST IDENT
:   :   IF NOT FONZ WITH 124K
:   :   : THEN-SETUP BUS TRAPS
:   :   :   SETUP NON EXISTENT ADDRESS
:   :   :   CALL READ ERROR CODE
:   :   :   IF RXCSR ERROR BIT OR RXESR NON-EXISTENT MEMORY BIT NOT SET
:   :   :   : THEN-CALL LOGIC TEST ERROR
:   :   :   ENDIF
:   :   CLEAR ERROR SET BY TRAP
:   :   CLEAR BUS TRAP VECTOR
:   :   ENDIF
:   ENDIF
: ENDTST
-----
: BOARD CALLOUT:
: 1. INTERFACE
-----

```



```

5001 027146          TSETUP
      027146 012737 027316 002466 BGNT19: MOV #T19TBL,TSTID ;SETUP TEST ID TBL-TEST# 19
      027154 032737 000001 002324 IAT19: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      027162 001447          BEQ XT19 ;BIT SET, THEN
      027164 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5002 027170 032737 010000 002500 IBT19: BIT #FONZFG,FLAGSP ;IF FONZ 124K FLAG
5003 027176 001041          BNE XT19 ;NOT SET, THEN
5004 027200 005037 002452          CLR ABORT ;CLEAR ABORT FLAG
5005 027204          SETVEC #BTRP4,#TRAP,#PRI07
5006 027232 013737 002344 002364          MOV NXMADR,RECADR ;SETUP NON EXISTENT MEMORY ADR
5007 027240 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5008 027244 012737 000052 002464          MOV #NXMERR,NGTSER ;SETUP EXPECTED NEG TEST ERR=NXM ERR
5009 027252 042737 000200 002476          BIC #RECFLG,FLAGST ;CLEAR READ ERR CODE FLAG (SU ERR CODE NOT EVALUATED)
5010 027260 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5011 027264 005037 002452          CLR ABORT ;FLAG
5012 027270          CLRVEC #BTRP4
5013 027276 004737 010440          CALL INTIAL ;CALL PROG INITIALIZE
5014 027302          XT19: EXIT TST
5015 027306          REGTBL
5016 027306          REGTB 1,4040,400,NXMBIT,173777 ;CHECK ERR BIT & NXM ERR SET
      027306 004040          T19RT1: .WORD 4040 ;RXCSR SHOULD BE
      027310 000400          .WORD 400 ;RXCSR DONT CARE
      027312 004000          .WORD NXMBIT ;RXESR SHOULD BE
      027314 173777          .WORD 173777 ;RXESR DONT CARE
5017 027316          TTBL REGCK!NEGTST,0
      027316 027106          T19TBL: .WORD T19MSG
      027320 004001          .WORD REGCK!NEGTST
      027322 000000          .WORD 0
      027324 177777          .WORD -1
      027326          T19RTB:
      027326 027306          .WORD T19RT1
      027330 177777          .WORD -1
5018 027332          FRUTBL INTONL
      027332          T19FTB:
      027332 006644          .WORD INTONL
      027334 177777          .WORD -1
5019 027336          ENDTST
    
```

5022 .SBTTL TEST 20 - INTERRUPT - LGC TST
 027340 000413 BR BGNT20 ;BR TO BGN TST
 027342 040 111 116 T20MSG: .ASCIZ / INTERRUPT - LGC TST/
 .EVEN

5023
 5024 :++
 5025 : TEST TO VERIFY THAT THE INTERRUPTS CAN BE SET AND THAT THE DEVICE
 5026 : RESPONDS AS EXPECTED.

5027 -----
 5028 : BGNTST
 5029 : IF LOGIC TEST
 5030 : THEN-SETUP TEST IDENT
 5031 : SET PROCESSOR PRIORITY-> -> NO INTERRUPTS
 5032 : CAUSE RX INTERRUPT (WHEN PRIORITY LOWERED)
 5033 : CALL WATCH DOG TO LOWER PRIORITY & WAIT FOR INTERRUPT
 5034 : CALL ERROR
 5035 : CLEAR RX INTERRUPT BIT
 5036 : NOP
 5037 : ENDIF
 5038 : ENDTST

5039 -----
 5040 : BOARD CALLOUT:
 5041 : 1. INTERFACE
 5042 -----
 5043 :--

5044 027370 TSETUP ;
 027370 012737 027462 002466 BGNT20: MOV #T20TBL,TSTID ;SETUP TEST ID TBL-TEST# 20
 027376 032737 000001 002324 IAT20: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
 027404 001424 BEQ XT20 ;BIT SET, THEN
 027406 004737 020736 CALL LTSTUP ;CALL LOGIC TEST SETUP
 5045 027412 005037 012024 CLR DX ;SET WATCH DOG MULTIPLIER=0
 5046 027416 SETPRI #PRI07 ;SET PROCESSOR PRI=NO INTERRUPTS
 5047 027424 052777 000100 152716 BIS #100,@RXCS ;CAUSE RX TO INTERRUPT-WHEN PRI LOWERED
 5048 027432 004737 011662 CALL WATCH ;CALL WATCH DOG-LOWER PRI & WAIT FOR INTERRUPT
 5049 027436 004737 003060 CALL ERROR ;CALL ERROR
 5050 027442 042777 000100 152700 BIC #100,@RXCS ;CLEAR RX INTERRUPT BIT
 5051 027450 012737 000010 012024 MOV #10,DX ;RESET WATCH DOG MULTIPLIER
 5052 027456 XT20: EXIT TST
 5053 027462 REGTBL
 5054 027462 TTBL 0,0
 027462 027342 T20TBL: .WORD T20MSG
 027464 000000 .WORD 0
 027466 000000 .WORD 0
 027470 177777 .WORD -1
 027472 T20RTB: .WORD -1
 027472 177777
 5055 027474 FRUTBL INTONL T20FTB: .WORD INTONL
 027474 .WORD -1
 027474 006644
 027476 177777
 5056 027500 ENDTST

5059 .SBTTL TEST 21 - PRIORITY LVL - LGC TST
027502 000414 BR BGNT21 ;BR TO BGN TST
027504 040 120 122 T21MSG: .ASCIZ / PRIORITY LVL - LGC TST/
.EVEN

5060
5061 :++
5062 : TEST TO VERIFY THAT THE DEVICE PRIORITY IS SET TO THE CORRECT LEVEL.
5063 :-----
5064 : BGNTST
5065 : IF LOGIC TEST
5066 : : THEN-SETUP TEST IDENT
5067 : : LOWER WATCH DOG TIMEOUT MULTIPLIER
5068 : : SETUP PROCESSOR PRIORITY=7 (PRESET VALUE)
5069 : : BGND0
5070 : : : SET PROCESSOR PRIORITY (PRESET VALUE)
5071 : : : SET DEVICE INTERRUPT BIT TO ENABLE INTERRUPT
5072 : : : IF INTERRUPT OR ERROR OCCURRED
5073 : : : : THEN-SET DO LOOP DONE BIT -> FLAGS
5074 : : : : ELSE-LOWER SETUP PROCESSOR PRIORITY
5075 : : : : CLEAR DEVICE INTERRUPT BIT
5076 : : : ENDIF
5077 : : DOUNTIL DO LOOP DONE BIT SET, PROCESSOR PRI=0 OR NO DONE BIT ESR
5078 : : IF SETUP PROCESSOR PRI NOT=DEVICE PRIORITY
5079 : : : THEN-CALL LOGIC TEST ERROR
5080 : : : ENDIF
5081 : : : ENDIF
5082 : : : ENDTST
5083 :-----
5084 : BOARD CALLOUT:
5085 : 1. INTERFACE
5086 :-----
5087 :--

```

5090 027534          TSETUP
      027534 012737 030106 002466 BGNT21: MOV #T21TBL,TSTID ;SETUP TEST ID TBL-TEST# 21
      027542 032737 000001 002324 IAT21: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      027550 001511          BEQ XT21 ;BIT SET, THEN
      027552 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5091 027556 004737 010440          CALL INTIAL ;CALL PROG INITIALIZE
5092 027562 012737 000001 012024          MOV #1,DX ;SET WATCH DOG MULTIPLIER=1
5093 027570          SETPRI #PRI07 ;SET PROCESSOR PRI=07 (NO INTERRUPTS)
5094 027576 005005          CLR R5 ;SET PRIORITY TABLE INDEX
5095 027600 005037 002520          BBT21: CLR ERRNBR ;CLEAR ERROR NUMBER INDICATOR
5096 027604 116537 021741 002416          MOVB PRITAB+7(R5),PRIORT ;SET PROCESSOR PRIORITY TO NEW LEVEL
5097 027612 013737 002416 002504          MOV PRIORT,TTEMP1 ;SETUP TEMP1 = PROCESSOR PRI
5098 027620 062737 000040 002504          ADD #40,TTEMP1 ;NOW SETUP FOR COMPARE, I.E. ONE PRI LVL HIGHER
5099 027626 052777 000100 152514          BIS #100,@RXCS ;SET RX INTERRUPT BIT,AS PROCESSOR PRI LOWERED, INTERRUPTS
5100 027634 004737 011662          CALL WATCH ;CALL WATCH DOG TO WAIT FOR INTERRUPT
5101 027640 022737 000015 002520 ICT21: CMP #DNNINT,ERRNBR ;IF INTERRUPT OR ERROR
5102 027646 001404          BEQ ECT21 ;OCCURRED, THEN
5103 027650 052737 000010 002476          BIS #DLPDN,FLAGST ;SET DO LOOP DONE FLAG
5104 027656 000404          BR UBT21 ;BR TO DOUNTIL 'B'
5105 027660 005305          ECT21: DEC R5 ;SET INDEX TO NEXT LOWER PROCESSOR PRI
5106 027662 042777 000100 152460 EDT21: BIC #100,@RXCS ;CLEAR DEVICE INTERRUPT BIT
5107 027670 020527 177770          UBT21: CMP R5,#-8. ;DO UNTIL PROCESSOR PRI TABLE ALL DONE
5108 027674 001404          BEQ IET21 ;OR
5109 027676 032737 000010 002476          BIT #DLPDN,FLAGST ;DOUNTIL FLAGST DO LOOP DONE FLAG
5110 027704 001735          BEQ BBT21 ;SET
5111 027706 005737 002520          IET21: TST ERRNBR ;IF INTERRUPT OCCURRED
5112 027712 001026          BNE LET21 ;THEN
5113 027714 032737 000400 002500 IFT21: BIT #LSIFLG,FLAGSP ;IF FLAGSP=LSI FLAG
5114 027722 001024          BNE XT21 ;NOT SET, THEN
5115 027724 023737 002504 002356 IGT21: CMP TTEMP1,RXPRI ;IF SETUP PROCESSOR PRIORITY & RX PRIORITY
5116 027732 001420          BEQ XT21 ;DONT MATCH
5117 027734 012737 000054 002520          MOV #PRILEV,ERRNBR ;SETUP ERR NBR=PRI LEV ERR
5118 027742 004737 003060          CALL ERROR ;CALL ERROR
5119 027746 013703 002416          MOV PRIORT,R3 ;SETUP INTERRUPT PRI LEV FOR PRT
5120 027752 013702 002356          MOV RXPRI,R2 ;SETUP RX PRI LEV FOR PRINT
5121 027756 012701 030006          MOV #PRIMSG,R1 ;SETUP PRI LEV MSG
5122 027762 004737 002756          CALL PRTX2S ;PRINT MSG
5123 027766 000402          BR XT21 ;BR TO TEXT EXIT
5124 027770 004737 003060          LET21: CALL ERROR ;CALL ERROR
5125 027774 012737 000010 012024 XT21: MOV #10,DX ;RESET WATCHDOG MULTIPLIER
5126 030002          EXIT TST
5127 030006 045 116 045 PRIMSG: .ASCIZ /%N%$6%A RX SET AT PRI LEV=%03%N%$6%A INTERRUPTED AT PRI LEV=%03/
5128
5129 030106
5130 030106          T21TBL: .WORD T21MSG
      030106 027504          .WORD 0
      030110 000000          .WORD 0
      030112 000000          .WORD 0
      030114 177777          .WORD -1
      030116          T21RTB: .WORD -1
      030116 177777          FRUTBL INTONL T21FTB: .WORD INTONL
5131 030120          .WORD -1
      030120          .WORD -1
      030120 006644
      030122 177777
5132 030124          ENDTST
    
```



```

5135          .SBTTL TEST 22 - INITIALIZE CONTROL - LGC TST
          030126 000417          BR          BGNT22          ;BR TO BGN TST
          030130 040          111          116 T22MSG: .ASCIZ / INITIALIZE CONTROL - LGC TST/
                                     .EVEN

5136
5137
5138          :++
5139          : TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE INITIALIZE.
5140          :-----
5141          : BGNTST
5142          : IF LOGIC TEST
5143          : THEN-SETUP TEST IDENT
5144          : ISSUE DEVICE PROGRAMMED INITIALIZE
5145          : CALL ERRCHK
5146          : ENDIF
5147          : ENDTST
5148          :-----
5149          : BOARD CALLOUT:
5150          : 1. CONTROLLER
5151          : 2. INTERFACE
5152          :-----
5153          :--
          030166          TSETUP          ;
          030166 012737 030224 002466 BGNT22: MOV          #T22TBL,TSTID          ;SETUP TEST ID TBL-TEST# 22
          030174 032737 000001 002324 IAT22: BIT          #LOGICT,TSTMOD          ;IF TEST MODE=LOGIC TEST
          030202 001406          BEQ          XT22          ;BIT SET, THEN
          030204 004737 020736          CALL          LTSTUP          ;CALL LOGIC TEST SETUP
5154 030210 004737 010440          CALL          INTIAL          ;CALL INITIALIZE
5155 030214 004737 017724          CALL          ERRCHK          ;CALL ERROR CHECK
5156 030220          XT22: EXIT          TST
5157 030224          REGTB! CEINIT

          015046          REGS1=CEINIT
5158 030224          TTBL          REGCK, RGPRT
          030224 030130          T22TBL: .WORD          T22MSG
          030226 000001          .WORD          REGCK
          030230 000004          .WORD          RGPRT
          030232 177777          .WORD          -1
          030234          T22RTB:
          030234 015046          .WORD          REGS1
          030236 177777          .WORD          -1
5159 030240          FRUTBL          CTLINF          T22FTB:
          030240          .WORD          CTLINF
          030240 006646          .WORD          -1
          030242 177777
5160 030244          ENDTST
  
```

5163 .SBTTL TEST 23 - DATA BUF INTEGRITY - LGC TST
030246 000417 BR BGNT23 ;BR TO BGN TST
030250 040 104 101 T23MSG: .ASCIZ / DATA BUF INTEGRITY - LGC TST/
.EVEN

5164
5165
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197

```
++  
: TEST TO VERIFY ALL BITS OF DATA BUFFER, VARIOUS PATTERNS WILL BE USED.  
-----  
: BGNTST  
: IF LOGIC TEST  
: THEN-SETUP TEST IDENT  
: SETUP RANDOM DATA PATTERN  
: NOP  
: BGNDO  
: : CALL DATA PATTERN SETUP  
: : CALL FILL BUFFER  
: : IF NO ERROR (ESCAPE TEST)  
: : THEN-CALL EMPTY BUFFER  
: : : IF NO ERROR (ESCAPE TEST)  
: : : THEN-SET EMPTY BUFFER FLAG  
: : : CALL DATA CHECK  
: : : ADVANCE PATTERN COUNT  
: : : GET NEW PATTERN #  
: : : IF FOUR PATTERNS DONE  
: : : THEN-SET DO LOOP DONE  
: : : ENDF  
: : : ENDF  
: : ENDF  
: DOUNTIL DO LOOP DONE FLAG SET  
: NOP  
: ENDF  
: ENDTST  
-----  
: BOARD CALLOUT:  
: 1. CONTROLLER  
: 2. INTERFACE  
-----  
:--
```



```

5200 030306          TSETUP
      030306 012737 030430 002466 BGNT23: MOV #T23TBL,TSTID ;SETUP TEST ID TBL-TEST# 23
      030314 032737 000001 002324 IAT23: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      030322 001440          BEQ XT23 ;BIT SET, THEN
      030324 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5201 030330 012737 000001 012660          MOV #1,PAT ;SET DATA PATTERN = 0'S
5202 030336 004737 012306          BBT23: CALL STDATP ;CALL SET DATA PATTERN
5203 030342 004737 010510          CALL FILBUF ;CALL FILL BUFFER
5204 030346          ESCAPE TST ;IF NO ERROR, THEN
5205 030352 004737 010626          CALL EMPBUF ;CALL EMPTY BUFFER
5206 030356          ESCAPE TST ;IF NO ERROR, THEN
5207 030362 004737 013246          CALL DATAK ;CALL DATA CHECK
5208 030366 005237 012660          INC PAT ;ADVANCE TO NEXT DATA PATTERN
5209 030372 022737 000010 012660 ICT23: CMP #8.,PAT ;IF ALL DATA PATTERNS
5210 030400 001003          BNE UBT23 ;DONE, THEN
5211 030402 052737 000010 002476          BIS #DLPDN,FLAGST ;SET FLAGST=DO_LOOP_DONE_FLAG
5212 030410 032737 000010 002476 UBT23: BIT #DLPDN,FLAGST ;DUNTIL FLAGST-DO_LOOP_DONE_FLAG
5213 030416 001747          BEQ BBT23 ;IS SET
5214 030420 005037 012660          CLR PAT ;RESET DATA PATTERN
5215 030424          XT23: EXIT TST
5216 030430          REGTBL
5217 030430          TTBL EMBUFF,0
      030430 030250          T23TBL: .WORD T23MSG
      030432 000020          .WORD EMBUFF
      030434 000000          .WORD 0
      030436 177777          .WORD -1
      030440          T23RTB:
      030440 177777          .WORD -1
5218 030442          FRUTBL CTLINF
      030442          T23FTB:
      030442 006646          .WORD CTLINF
      030444 177777          .WORD -1
5219 030446          ENDTST
    
```

5222 .SBTTL TEST 24 - WRD CNT INTEGRITY - LGC TST
030450 000417 BR BGNT24 ;BR TO BGN TST
030452 040 127 122 T24MSG: .ASCIZ / WRD CNT INTEGRITY - LGC TST/
.EVEN

5223
5224
5225
5226
5227
5228
5229
5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5240
5241
5242
5243
5244
5245
5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256

```
++  
: TEST TO VERIFY ALL BITS OF WORD COUNT REGISTER AND CHECK THAT EXCEEDING  
: THE WORD COUNT FOR DISKETTE DENSITY WILL BE DETECTED.  
-----  
: BGMTST  
: IF LOGIC TEST  
: THEN-SETUP TEST IDENT  
: SET DENSITY CONTROL=DOUBLE  
: SET BUFFER LENGTH=128.  
: BGND0  
: CALL FILL BUFFER  
: IF NO ERROR (ESCAPE TEST)  
: THEN-CALL READ ERROR CODE  
: IF NO ERROR (ESCAPE TEST)  
: THEN-IF WORD COUNTS NOT EQUAL  
: THEN-SETUP WORD COUNT ERROR  
: CALL ERROR  
: ELSE-UPDATE WORD COUNT  
: IF WORD COUNT=0  
: THEN-SET DO LOOP DONE FLAG  
: ENDIF  
: ENDIF  
: ENDIF  
: DOUNTIL DO LOOP DONE FLAG SET  
: NOP  
: ENDIF  
: ENDTST  
-----  
: BOARD CALLOUT:  
: 1. CONTROLLER  
-----  
:--
```



```

5259 030510          TSETUP
      030510 012737 030670 002466 BGNT24: MOV #T24TBL,TSTID ;SETUP TEST ID TBL-TEST# 24
      030516 032737 000001 002324 IAT24: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      030524 001457          BEQ XT24 ;BIT SET, THEN
      030526 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5260 030532 012737 000400 002412          MOV #DENBIT,DENSTY ;SET DENSITY CONTROL=DOUBLE
5261 030540 012737 000200 002370          MOV #128.,WDCNT ;SET WORD COUNT=128.
5262 030546 004737 010510          BBT24: CALL FILBUF ;CALL FILL BUFFER
5263 030552          ESCAPE TST ;IF NO ERROR THEN
5264 030556 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5265 030562          ESCAPE TST ;IF NO ERROR THEN
5266 030566 105737 002443          ICT24: TSTB WC ;IF WORD COUNT
5267 030572 001420          BEQ LCT24 ;NOT EQUAL 0, THEN
5268 030574 012737 000023 002520          MOV #WCERR,ERRNBR ;SETUP ERR NBR=WORD COUNT ERROR
5269 030602 005037 002440          CLR REGACT ;CLEAR REG ACTUAL
5270 030606 113737 002443 002440          MOVB WC,REGACT ;SETUP WORD COUNT ACTUAL
5271 030614 005037 002436          CLR REGEXP ;SETUP WORD COUNT EXPECTED
5272 030620 004737 003060          CALL ERROR ;CALL ERROR
5273 030624 052737 000010 002476          BIS #DLPDN,FLAGST ;SET FLAGST=DO LOOP DONE FLAG
5274 030632 000410          BR UBT24 ;BR TO DOUNTIL 'B'
5275 030634 005337 002370          LCT24: DEC WDCNT ;DECREMENT WORD COUNT
5276 030640 005737 002370          IDT24: TST WDCNT ;IF WORD COUNT
5277 030644 001003          BNE UBT24 ;EQUALS ZERO, THEN
5278 030646 052737 000010 002476          BIS #DLPDN,FLAGST ;SET FLAGST=DO LOOP_DONE_FLAG
5279 030654 032737 000010 002476          UBT24: BIT #DLPDN,FLAGST ;DOUNTIL FLAGST=DO_LOOP_DONE_FLAG
5280 030662 001731          BEQ BBT24 ;SET
5281 030664          XT24: EXIT TST
5282 030670          REGTBL
5283 030670          TTBL 0, RGPRT
      030670 030452          T24TBL: .WORD T24MSG
      030672 000000          .WORD 0
      030674 000004          .WORD RGPRT
      030676 177777          .WORD -1
      030700          T24RTB: .WORD -1
5284 030702          FRUTBL CTLINF          T24FTB: .WORD CTLINF
      030702          .WORD -1
      030702 006646
      030704 177777
5285 030706          ENDTST
    
```

5288 .SBTTL TEST 25 - CONTROLLER-READ*WRITE ELECT - LGC TST
030710 000424 BR BGNT25 ;BR TO BGN TST
030712 040 103 117 T25MSG: .ASCIZ / CONTROLLER-READ*WRITE ELECT - LGC TST/
.EVEN

5289
5290
5291
5292
5293
5294
5295
5296
5297
5298
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312

;++
: TEST TO VERIFY MINIMAL CONTROLLER BOARD-READ/WRITE ELECTRONICS BOARD
: INTERFACE VIA INITIALIZE OF A SELECTED DRIVE.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: NOP
: ISSUE PROGRAMMED INITIALIZE
: CALL ERROR CK
: CALL READ ERROR CODE
: IF NO ERROR (ESCAPE TEST)
: THEN-CALL ERROR CK
: ENDIF
: NOP
: ENDIF

: ENDTST

: BOARD CALLOUT:
: 1. CONTROLLER
: 2. R/W ELECTRONICS

:--


```

5315 030762          TSETUP
      030762 012737 031054 002466 BGNT25: MOV #T25TBL,TSTID ;SETUP TEST ID TBL-TEST# 25
      030770 032737 000001 002324 IAT25: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      030776 001424          BEQ XT25 ;BIT SET, THEN
      031000 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5316 031004 004737 010440          CALL INTIAL ;CALL INITIALIZE
5317 031010 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5318 031014 005237 002470          INC TCMDC ;INCREMENT TST CMD CTR *****
5319 031020 052737 000200 002500 BIS #RECTST,FLAGSP ;SET READ ERROR CODE TEST=FLAGSP
5320 031026 004737 011340          CALL RDERCD ;CALL READ ERROR CODE-
5321 031032          ESCAPE TST ;IF NO ERROR
5322 031036 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5323 031042 042737 000200 002500 BIC #RECTST,FLAGSP ;CLEAR READ ERROR CODE TEST=FLAGSP
5324 031050          XT25: EXIT TST
5325 031054          REGTBL CEINIT,CSESND
      015046          REGS1=CEINIT
      015076          REGS2=CSESND

5326 031054          TTBL REGCK,0
      031054 030712          T25TBL: .WORD T25MSG
      031056 000001          .WORD REGCK
      031060 000000          .WORD 0
      031062 177777          .WORD -1
      031064          T25RTB:
      031064 015046          .WORD REGS1
      031066 015076          .WORD REGS2
      031070 177777          .WORD -1
5327 031072          FRUTBL CTLRWE
      031072          T25FTB:
      031072 006651          .WORD CTLRWE
      031074 177777          .WORD -1
5328 031076          ENDTST
    
```

5331 .SBTTL TEST 26 - READ SECTOR-PRT:1 - LGC TST
031100 000417 BR BGNT26 ;BR TO BGN TST
031102 040 122 105 T26MSG: .ASCIZ / READ SECTOR-PRT:1 - LGC TST/
.EVEN

5332
5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361

:+
: TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A READ SECTOR IN
: BOTH DENSITIES AND RETURN A VALID ERROR CODE.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: CALL DEVICE DENSITY CK
: SET DENSITY CONTROL=DISK DEN
: CALL READ SECTOR
: CALL READ ERROR CODE
: IF NO COMMAND ERRORS
: THEN-CALL ERROR CK
: CALL COMPLIMENT DENSITY
: CALL READ SECTOR
: CALL READ ERROR CODE
: IF NO COMMAND ERRORS
: THEN-CALL ERROR CK
: ENDIF
: NOP
: ENDIF

ENDTST

: BOARD CALLOUT:
: 1. CONTROLLER
: 2. R/W ELECTRONICS

--


```

5364 031140          TSETUP
      031140 012737 031256 002466 BGNT26: MOV #T26TBL,TSTID ;SETUP TEST ID TBL-TEST# 26
      031146 032737 000001 002324 IAT26: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      031154 001436          BEQ XT26 ;BIT SET, THEN
      031156 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5365 031162 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
5366 031166 004737 020472          CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
5367 031172 004737 011062          CALL READ ;CALL READ SECTOR
5368 031176 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5369 031202          ESCAPE TST ;IF NO COMMAND ERRORS, THEN
5370 031206 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5371 031212 004737 020430          CALL CDENC ;CALL COMPLIMENT DENSITY CONTROL
5372 031216 052737 004000 002476 BIS #NEGTEST,FLAGST ;SET FLAGST=NEG TEST FLAG
5373 031224 012737 000030 002464 MOV #DENERR,NGTSER ;SETUP NEGTEST SET ERROR=DEN ERROR
5374 031232 004737 011062          CALL READ ;CALL READ SECTOR
5375 031236 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5376 031242          ESCAPE TST ;IF NO COMMAND ERRORS, THEN
5377 031246 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5378 031252          XT26: EXIT TST
5379 031256          REGTBL CSESAL
                                REGS1=CSESAL
5380 031256          TTBL REGCK,0
      031256 031102          T26TBL: .WORD T26MSG
      031260 000001          .WORD REGCK
      031262 000000          .WORD 0
      031264 177777          .WORD -1
      031266          T26RTB:
      031266 015036          .WORD REGS1
      031270 177777          .WORD -1
5381 031272          FRUTBL CTLRWE
      031272          T26FTB:
      031272 006651          .WORD CTLRWE
      031274 177777          .WORD -1
5382 031276          ENDTST
    
```

```

5385      031300 000414      .SBTTL TEST 27 - POSITIONING - LGC TST
          031302 040      120 117 T27MSG: BR BGNT27 ;BR TO BGN TST
          .ASCIZ / POSITIONING - LGC TST/
          .EVEN

```

5386
5387
5388
5389
5390
5391
5392
5393
5394
5395
5396
5397
5398
5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415

```

:++
: TEST TO VERIFY THAT THE DRIVE WILL READ THE HEADERS ON ALL TRACKS OF
: THE DEIVE AS EXPECTED.
-----

```

```

BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
:   SET TRACK INIT FLAG
:   SET SECTOR=10
:   BGNDO
:     CALL GET TRACK
:     CALL READ ERROR CODE
:     CALL READ SECTOR
:     IF NO COMMAND ERRORS (ESCAPE TST)
:       THEN-CALL ERROR CHECK
:         CALL TRACKS ERROR CK
:         CLEAR TRACK INIT FLAG
:         NOP
:     ENDIF
:   DOUNTIL TRACKS DONE, ABORT FLAG SET, OR TRACK ERRORS=10
:   NOP
: ENDIF
ENDTST

```

```

-----
: BOARD CALLOUT:
: 1. CONTROLLER
: 2. R/W ELECTRONICS
-----
:--

```



```

5418 031332          TSETUP
      031332 012737 031446 002466 BGNT27: MOV #T27TBL,TSTID ;SETUP TEST ID TBL-TEST# 27
      031340 032737 000001 002324 IAT27: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      031346 001435          BEQ XT27 ;BIT SET, THEN
      031350 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5419 031354 012737 000400 002510          MOV #ITK!RTK,TKSCFG ;SET TRK/SEC FLAG-->TRACK=INIT & RANDOM
5420 031362 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
5421 031366 004737 020472          CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
5422 031372 012737 000010 002376          MOV #10,SECTOR ;SET SECTOR=10
5423 031400 004737 012662          BBT27: CALL GETTRK ;CALL GET TRACK
5424 031404 004737 011062          CALL READ ;CALL READ SECTOR
5425 031410 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5426 031414          ESCAPE TST ;IF NO COMMAND ERRORS, THEN
5427 031420 004737 017516          CALL TKERCK ;CALL TRACK ERROR CHECK
5428 031424 042737 000400 002510          BIC #ITK,TKSCFG ;CLEAR INT TRK FLAG
5429 031432 032737 001000 002476 UBT27: BIT #TRKDON,FLAGST ;DO UNTIL FLAGST-TRACK DONE FLAG
5430 031440 001757          BEQ BBT27 ;SET,
5431 031442          XT27: EXIT TST
5432 031446          REGTBL CSESAL
                                     REGS1=CSESAL
5433 031446          TTBL REGCK,0
      031446 031302          T27TBL: .WORD T27MSG
      031450 000001          .WORD REGCK
      031452 000000          .WORD 0
      031454 177777          .WORD -1
      031456          T27RTB:
      031456 015036          .WORD REGS1
      031460 177777          .WORD -1
5434 031462          FRUTBL CTLRWE
      031462          T27FTB:
      031462 006651          .WORD CTLRWE
      031464 177777          .WORD -1
5435 031466          ENDTST
    
```

5438 .SBTTL TEST 28 - WRITE SECTOR-PRT:1 - LGC TST
031470 000417 BR BGNT28 ;BR TO BGN TST
031472 040 127 122 T28MSG: .ASCIZ / WRITE SECTOR-PRT:1 - LGC TST/
.EVEN

5439
5440 :++
5441 : TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A WRITE SECTOR IN
5442 : BOTH DENSITIES AND RETURN A VALID ERROR CODE.
5443 -----

5444 : BGNTST
5445 : IF LOGIC TEST
5446 : THEN-SETUP TEST IDENT
5447 : CALL DEVICE DENSITY CHECK
5448 : SET DENSITY CONTROL=DISK DEN
5449 : CALL WRITE SECTOR
5450 : IF NO COMMAND ERROR (ESCAPE TEST)
5451 : THEN-CALL ERROR CHECK
5452 : CALL COMPLIMENT DENSITY CONTROL
5453 : CALL WRITE SECTOR
5454 : IF NO COMMAND ERROR (ESCAPE TEST)
5455 : THEN-CALL ERROR CHECK
5456 : ENDIF
5457 : NOP
5458 : ENDIF
5459 : NOP

5460 : ENDIF
5461 : ENDTST
5462 -----

5463 : BOARD CALLOUT:
5464 : 1. CONTROLLER
5465 : 2. R/W ELECTRONICS
5466 -----

5467 :--


```

5470 031530          TSETUP
      031530 012737 031642 002466 BGNT28: MOV #T28TBL,TSTID ;SETUP TEST ID TBL-TEST# 28
      031536 032737 000001 002324 IAT28: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      031544 001434          BEQ XT28 ;BIT SET, THEN
      031546 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5471 031552 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
5472 031556 004737 020472          CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
5473 031562 004737 010744          CALL WRITE ;CALL WRITE SECTOR
5474 031566          ESCAPE TST ;IF NO COMMAND ERROR, THEN
5475 031572 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5476 031576 004737 020430          CALL CDENC ;CALL COMPLIMENT DENSITY CONTROL
5477 031602 052737 004000 002476 BIS #NEGTST,FLAGST ;SET FLAGST-NEG TEST FLAG
5478 031610 012737 000030 002464 MOV #DENERR,NGT SER ;SETUP NEG TEST ERR ERR=DENSITY ERR
5479 031616 004737 010744          CALL WRITE ;CALL WRITE
5480 031622 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5481 031626          ESCAPE TST ;IF NO COMMAND ERROR
5482 031632 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5483 031636          XT28: EXIT TST
5484 031642          REGTBL CSESAL
      015036          REGS1=CSESAL
5485 031642          TTBL REGCK,0
      031642 031472          T28TBL: .WORD T28MSG
      031644 000001          .WORD REGCK
      031646 000000          .WORD 0
      031650 177777          .WORD -1
      031652          T28RTB:
      031652 015036          .WORD REGS1
      031654 177777          .WORD -1
5486 031656          FRUTBL CTLRWE
      031656          T28FTB:
      031656 006651          .WORD CTLRWE
      031660 177777          .WORD -1
5487 031662          ENDTST
    
```

5490 .SBTTL TEST 29 - DELETED DATA WRITE PRT:1 - LGC TST
031664 000422 BR BGNT29 ;BR TO BGN TST
031666 040 104 105 T29MSG: .ASCIZ / DELETED DATA WRITE PRT:1 - LGC TST/
.EVEN

5491
5492 :++
5493 : TEST TO VERIFY THAT THE DEVICE SET A DELETED DATA MARK ON THE DISKETTE
5494 : HEADER AND RETRIEVE THIS DATA ON A KNOWN GOOD DISKETTE.
5495 -----

5496 : BGNTST
5497 : IF LOGIC TEST
5498 : THEN-SETUP TEST IDENT
5499 : CALL DEVICE DENSITY CK
5500 : SET DENSITY CONTROL=DENSITY STATUS
5501 : SET DELETED DATA FLAG (BIT#3-CMD)
5502 : CALL WRITE SECTOR
5503 : IF NO COMMAND ERROR (ESCAPE TEST)
5504 : THEN-CALL READ SECTOR
5505 : : IF NO COMMAND ERROR (ESCAPE TEST)
5506 : : THEN-IF RXESR-DELETED DATA BIT NOT SET
5507 : : : THEN-SET ERROR NUMBER=DELETED DATA ERR
5508 : : : CALL ERROR
5509 : : : ENDF
5510 : : ENDF
5511 : : ENDF

5512 : ENDF
5513 : ENDTST
5514 :-----
5515 : BOARD CALLOUT:
5516 : 1. CONTROLLER
5517 : 2. R/W ELECTRONICS
5518 :-----
5519 :--


```

5522 031732          TSETUP
      031732 012737 032056 002466 BGNT29: MOV #T29TBL,TSTID ;SETUP TEST ID TBL-TEST# 29
      031740 032737 000001 002324 IAT29: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      031746 001441          BEQ XT29 ;BIT SET, THEN
      031750 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5523 031754 004737 017350          CALL DENCHK ;CALL DEVICE DENSITY CHECK
5524 031760 004737 020472          CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
5525 031764 012737 000010 002402          MOV #DLDCMD,DELDT ;SET DELETED DATA FLAG
5526 031772 004737 010744          CALL WRITE ;CALL WRITE SECTOR
5527 031776          IBT29: ESCAPE TST ;IF NO COMMAND ERROR, THEN
5528 032002 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5529 032006 004737 011062          CALL READ ;CALL READ SECTOR
5530 032012          ICT29: ESCAPE TST ;IF NO COMMAND ERROR, THEN
5531 032016 032777 000100 150326 IDT29: BIT #DLDBIT,@RXDB ;IF RXESR-DELETED DATA BIT
5532 032024 001006          BNE LCT29 ;NOT SET, THEN
5533 032026 012737 000032 002520          MOV #DLDTERR,ERRNBR ;SETUP ERROR NUMBER=DELETED DATA ERROR
5534 032034 004737 003060          CALL ERROR ;CALL ERROR
5535 032040 000404          BR XT29 ;EXIT TST
5536 032042 005037 002402          LCT29: CLR DELDAT ;CLEAR DEL DATA MODE
5537 032046 004737 010744          CALL WRITE ;CALL WRITE SECTOR - CLR DATA FIELD
5538 032052          XT29: EXIT TST
5539 032056          REGTBL CSESAL
                                REGS1=CSESAL
5540 032056          TTBL REGCK,0
      032056 031666          T29TBL: .WORD T29MSG
      032060 000001          .WORD REGCK
      032062 000000          .WORD 0
      032064 177777          .WORD -1
      032066          T29RTB:
      032066 015036          .WORD REGS1
      032070 177777          .WORD -1
5541 032072          FRUTBL CTLRWE
      032072          T29FTB:
      032072 006651          .WORD CTLRWE
      032074 177777          .WORD -1
5542 032076          ENDTST

```

5545 .SBTTL TEST 30 - SET DENSITY - LGC TST
032100 000414 BR BGNT30 ;BR TO BGN TST
032102 040 123 105 T30MSG: .ASCIZ / SET DENSITY - LGC TST/
.EVEN

5546
5547
5548
5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591

```

: **
: TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE SET DENSITY IN BOTH
: DENSITIES. THE VALID WORD WILL ALSO BE CHECKED. ALSO TO VERIFY THAT
: THE DRIVE WILL READ IN THE NEW DENSITY WITHOUT ERROR.
-----
: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: CALL DEVICE DENSITY CHECK
: SAVE DEVICE DENSITY
: SET DENSITY CONTROL=SINGLE DEN
: CALL SET DENSITY
: IF NO COMMAND ERROR (ESCAPE TST)
: THEN-CALL ERROR CHECK
: SET VALIDITY WORD=ASCII "K"
: CALL SET DENSITY
: CALL READ ERROR CODE
: IF NO COMMAND ERROR (ESCAPE TST)
: THEN-SET FLAGST NEG TEST FLAG
: SETUP EXPECTED ERR=S.D. KEY WD ERR
: CALL ERROR CHECK
: SET DENSITY CONTROL=DOUBLE DENSITY
: SET VALIDITY WORD=ASCII 'I'
: CALL SET DENSITY
: IF NO COMMAND ERROR (ESCAPE TST)
: THEN-CALL ERROR CHECK
: CALL DEVICE DENSITY CHECK
: IF DEVICE DENSITY NOT=SET DENSITY
: THEN-SET ERR MSG=DENSITY NOT SET
: CALL ERROR CHECK
: ENDIF
: IF SAVED DEVICE DENSITY=DOUBLE DENSITY
: THEN-SET DENSITY CONTROL=SINGLE DEN
: ENDIF
: ENDIF
: ENDIF
: ENDIF
: ENDTST
-----
: BOARD CALLOUT:
: 1. CONTROLLER
: 2. R/W ELECTRONICS
-----
: --
```



```

5594 032132          TSETUP
      032132 012737 032356 002466 BGNT30: MOV #T30TBL,TSTID ;SETUP TEST ID TBL-TEST# 30
      032140 032737 000001 002324 IAT30: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      032146 001501          BEQ XT30 ;BIT SET, THEN
      032150 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5595 032154 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
5596 032160 013737 002414 002506          MOV DENSTA,TSAVE1 ;SAVE DEVICE DENSITY
5597 032166 005037 002412          CLR DENSTY ;SET DENSITY CONTROL=SINGLE DENSITY
5598 032172 004737 011172          CALL SETDN ;CALL SET DENSITY
5599 032176          ESCAPE TST ;IF NO COMMAND ERROR, THEN
5600 032202 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5601 032206 012737 000113 002372          MOV #'K,VARIFY ;SET VALIDITY WORD=ASCII 'K'
5602 032214 004737 011172          CALL SETDN ;CALL SET DENSITY
5603 032220 004737 011340          CALL RDRCD ;CALL READ ERROR CODE
5604 032224          ESCAPE TST ;IF NO COMMAND ERROR
5605 032230 052737 004000 002476          BIS #NEGTST,FLAGST ;SET FLAGST-NEG TEST FLAG
5606 032236 012737 000036 002464          MOV #SDKYWD,NGTSER ;SETUP EXPECTED ERROR=SET DEN KEYWORD ERROR
5607 032244 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5608 032250 012737 000400 002412          MOV #DENBIT,DENSTY ;SET DENSITY CONTROL=DOUBLE DENSITY
5609 032256 012737 000111 002372          MOV #'I,VARIFY ;SET VALIDITY WORD=ASCII 'I'
5610 032264 042737 004000 002476          BIC #NEGTST,FLAGST ;CLEAR FLAGST-NEG TEST FLAG
5611 032272 004737 011172          CALL SETDN ;CALL SET DENSITY
5612 032276          ESCAPE TST ;IF NO COMMAND ERROR, THEN
5613 032302 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5614 032306 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
5615 032312 023737 002414 002412          CMP DENSTA,DENSTY ;IF DENSITY DID
5616 032320 001405          BEQ IBT30 ;NOT SET, THEN
5617 032322 012737 000035 002520          MOV #STDNER,ERRNBR ;SET ERROR NBR=DENSITY DIDN'T SET ERROR
5618 032330 004737 003060          CALL ERROR ;CALL ERROR
5619 032334 005737 002506          IBT30: TST TSAVE1 ;IF SAVED DENSITY
5620 032340 001404          BEQ XT30 ;EQUALS DOUBLE DEN, THEN
5621 032342 005037 002412          CLR DENSTY ;SET DENSITY CONTROL=SINGLE DEN
5622 032346 004737 011172          CALL SETDN ;CALL SET DENSITY
5623 032352          XT30: EXIT TST
5624 032356          REGTBL CSESAL
                                     REGS1=CSESAL
5625 032356          TTBL REGCK,0
      032356 032102          T30TBL: .WORD T30MSG
      032360 000001          .WORD REGCK
      032362 000000          .WORD 0
      032364 177777          .WORD -1
      032366          T30RTB:
      032366 015036          .WORD REGS1
      032370 177777          .WORD -1
5626 032372          FRUTBL CTLRWE
      032372          T30FTB:
      032372 006651          .WORD CTLRWE
      032374 177777          .WORD -1
5627 032376          ENDTST
    
```

5630 .SBTTL TEST 31 - SECTOR ADR - LGC TST
032400 000413 BR BGNT31 ;BR TO BGN TST
032402 040 123 105 T31MSG: .ASCIZ / SECTOR ADR - LGC TST/
.EVEN

5631
5632
5633
5634
5635
5636
5637
5638
5639
5640
5641
5642
5643
5644
5645
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676

++
: TEST TO VERIFY THAT THE CONTROLLER WILL HANDLE ALL LEGAL AND ILLEGAL
: SECTOR ADDRESSES PROPERLY.

: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: SET TRACK ADR=0
: SET SECTOR LEGAL FLAG
: SET SECTOR INIT
: BGND0
: : CALL GET SECTOR ADR
: : CALL READ SECTOR
: : CALL READ ERROR CODE
: : IF FINI FLAG NOT SET
: : THEN-
: : : IF SECTOR ADR NOT=TARGET SECTOR ADR
: : : THEN-SETUP TO PRINT ERROR
: : : : CALL ERROR
: : : ELSE-CALL ERROR CK
: : : ENDF
: : ENDF
: DUNTIL SECTORS DONE FLAG SET OR ABORT FLAG SET
: CLEAR SECTORS DONE FLAG
: SET DONE TIME OUT MULTIPLIER=100
: SET NEG TEST FLAG
: BGND0
: : CALL READ SECTOR
: : CALL READ ERROR CODE
: : IF FINI FLAG NOT SET
: : THEN-IF SECTOR ADR NOT=TARGET SECTOR ADR
: : : THEN-SET ERR=SECTOR ADR ERROR
: : : : CALL ERROR
: : : ELSE-CALL ERROR CHECK
: : : ENDF
: : ENDF
: DUNTIL SECTORS DONE FLAG SET OR FINI FLAG SET
: NOP
: ENDF
: ENDTST

: BOARD CALLOUT:
: 1. CONTROLLER
:-----
:--


```

5679 032430          TSETUP
      032430 012737 032730 002466 BGNT31: MOV #T31TBL,TSTID ;SETUP TEST ID TBL-TEST# 31
      032436 032737 000001 002324 IAT31: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      032444 001527          BEQ XT31 ;BIT SET, THEN
      032446 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5680 032452 004737 010440          CALL INTIAL ;CALL INITAILIZE
5681 032456 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
5682 032462 004737 020472          CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
5683 032466 012737 001002 002510          MOV #ISC!SSC,TKSCFG ;SETUP SECTOR FLAGS=INITIALIZE & SEQUENCE
5684 032474 004737 013104          BBT31: CALL GETSEC ;CALL GET SECTOR
5685 032500 004737 011062          CALL READ ;CALL READ SECTOR
5686 032504 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5687 032510 005737 002454          ICT31: TST FIN ;IF FINI FLAG
5688 032514 001024          BNE UBT31 ;NOT SET, THEN
5689 032516 123737 002376 002447 IDT31: CMPB SECTOR,TSEC ;IF SECTOR ADR & DEVICE TARGET SECTOR
5690 032524 001416          BEQ LDT31 ;NOT =, THEN
5691 032526 012737 000042 002520          MOV #SECAER,ERRNBR ;SETUP ERR NBR=SECTOR ADDRESS ERROR
5692 032534 052737 000002 002500          BIS #SCPRT,FLAGSP ;SET FLAGSP=-PRINT SECTOR ADDRESS FLAG
5693 032542 004737 003060          CALL ERROR ;CALL ERROR
5694 032546 042737 000002 002500          BIC #SCPRT,FLAGSP ;CLEAR FLAGSP-PRINT SECTOR ADDRESS FLAG
5695 032554 004737 010440          CALL INTIAL ;CALL INITAILIZE
5696 032560 000402          BR UBT31 ;BR TO DOUNTIL 'B'
5697 032562 004737 017724          LDT31: CALL ERRCHK ;CALL ERROR CHECK
5698 032566 005737 002452          UBT31: TST ABORT ;DOUNTIL ABORT FLAG
5699 032572 001004          BNE EBT31 ;SET OR
5700 032574 032737 002000 002476          BIT #SECDON,FLAGST ;FLAGST-SECTOR DONE FLAG
5701 032602 001734          BEQ BBT31 ;SET
5702 032604 042737 002000 002476          EBT31: BIC #SECDON,FLAGST ;CLEAR FLAGST-SECTOR DONE FLAG
5703 032612 052737 004000 002476          BIS #NEGST,FLAGST ;SET FLAGST-NEG TEST FLAG
5704 032620 012737 000003 002464          MOV #RDERR,NGTSE ;SETUP EXPECTED ERROR=READ ERROR (SECTOR NOT FOUND EC=70)
5705 032626 012737 000100 002474          MOV #100,DNWTMT ;SET DONE WAIT MULTIPLIER SO NO TIME OUT
5706 032634 012737 000000 002376          MOV #0,SECTOR ;SET SECTOR ADR=0
5707 032642 004737 011062          BET31: CALL READ ;CALL READ SECTOR
5708 032646 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5709 032652 005737 002454          IFT31: TST FIN ;IF FINI FLAG
5710 032656 001017          BNE EFT31 ;NOT SET, THEN
5711 032660 123737 002376 002447          IGT31: CMPB SECTOR,TSEC ;IF SECTOR ADR AND TARGET SECTOR
5712 032666 001411          BEQ LGT31 ;NOT EQUAL, THEN
5713 032670 052737 000002 002500          BIS #SCPRT,FLAGSP ;SET FLAGSP=-PRINT SECTOR ADDRESS FLAG
5714 032676 012737 000042 002520          MOV #SECAER,ERRNBR ;SETUP ERR NBR=SECTOR ADDRESS ERROR
5715 032704 004737 003060          CALL ERROR ;CALL ERROR
5716 032710 000402          BR EFT31 ;CALL TO END 'G'
5717 032712 004737 017724          LGT31: CALL ERRCHK ;CALL ERROR CHECK
5718 032716 012737 000004 002474          EFT31: MOV #4,DNWTMT ;RESET DONE WAIT MUTIPLIER TO NORMAL
5719 032724          XT31: EXIT TST
5720 032730          REGTBL CSESAL
      015036          REGS1=CSESAL
5721 032730          TTBL REGCK,0
      032730 032402          T31TBL: .WORD T31MSG
      032732 000001          .WORD REGCK
      032734 000000          .WORD 0
      032736 177777          .WORD -1
      032740          T31RTB:
      032740 015036          .WORD REGS1
      032742 177777          .WORD -1
5722 032744          FRUTBL CTLRWE
      032744          T31FTB:
    
```

032744 006651
032746 177777
5723 032750

ENDTST

.WORD CTLRWE
.WORD -1

5726 .SBTTL TEST 32 - TRACK ADR - LGC TST
032752 000413 BR BGNT32 ;BR TO BGN TST
032754 040 124 122 T32MSG: .ASCIZ / TRACK ADR - LGC TST/
.EVEN

5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775

```

:++
: TEST TO VERIFY THAT THE CONTROLLER WILL HANDLE ALL LEGAL AND ILLEGAL
: TRACK ADDRESSES PROPERLY.
-----
: BGNTST
: IF LOGIC TEST
: THEN-SETUP TEST IDENT
: SET TRACK INIT & SEQUENCE FLAGS
: BGND0
: : BGND0
: : : CALL GET TRACK ADR
: : : CLEAR TRACK INIT FLAG
: : : CALL READ SECTOR
: : : CALL READ ERROR CODE
: : : IF FINI FLAG NOT SET
: : : THEN-
: : : : IF TRACK ADR NOT=TARGET TRACK ADR
: : : : THEN-CALL LOGIC TEST ERROR
: : : : ENDIF
: : : : IF ILLEGAL FLAG NOT SET
: : : : THEN-IF TRACK ADR NOT=UNIT TRACK ADR
: : : : : THEN-SETUP TRACK ADR ERR & CALL ERROR
: : : : : ELSE-IF ERROR CODE=40
: : : : : THEN-CALL LOGIC TEST ERROR
: : : : : ENDIF
: : : : ELSE-IF TRACK ADR=UNIT TRACK ADR
: : : : : THEN-SETUP TRACK ADR ERR & CALL ERROR
: : : : : ELSE-IF ERROR CODE NOT=40
: : : : : THEN-CALL LOGIC TEST ERROR
: : : : : ENDIF
: : : : ENDIF
: : : : ENDIF
: : : : ENDIF
: : : : DOUNTIL TRACKS DONE FLAG SET OR ABORT FLAG SET
: : : : SET TRACK INIT FLAG
: : : : IF TRACKS LEGAL FLAG SET
: : : : : THEN-SET TRACKS ILLEGAL FLAG
: : : : : ELSE-SET TRACKS LEGAL FLAG
: : : : : ENDIF
: : : : DOUNTIL TRACKS LEGAL FLAG SET
: : : : ENDIF
: ENDTST
-----
: BOARD CALLOUT:
: 1. CONTROLLER
-----
:--
```

```

5778 033002          TSETUP
      033002 012737 033336 002466 BGNT32: MOV #T32TBL,TSTID ;SETUP TEST ID TBL-TEST# 32
      033010 032737 000001 002324 IAT32: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      033016 001545          BEQ XT32 ;BIT SET, THEN
      033020 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5779 033024 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
5780 033030 004737 020472          CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
5781 033034 012737 000401 002510          MOV #ITK!STK,TKSCFG ;SET INITIALIZE & SEQUENCE TRACKS FLAG (TRACK/SECTOR FLAGS)
5782 033042 000240          BBT32: NOP ;
5783 033044 004737 012662          BCT32: CALL GETTRK ;CALL GET TRACK ADR
5784 033050 042737 000401 002510          BIC #ITK!STK,TKSCFG ;CLEAR INITIALIZE TRACKS FLAG
5785 033056 004737 011062          CALL READ ;CALL READ SECTOR
5786 033062 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5787 033066 005737 002454          IDT32: TST FIN ;IF FINI FLAG
5788 033072 001062          BNE UCT32 ;NOT SET, THEN
5789 033074 123737 002374 002446 IET32: CMPB TRACK,TTRK ;IF TRACK ADR & TARGET TRACK
5790 033102 001405          BEQ EET32 ;NOT EQUAL, THEN
5791 033104 012737 000041 002520          MOV #TRKAER,ERRNBR ;SETUP ERR NBR=TRACK ADDRESS ERROR
5792 033112 004737 003060          CALL ERROR ;CALL ERROR
5793 033116 013705 002420          EET32: MOV DRVOFF,R5 ;SET R5=DRIVE BYTE OFFSET
5794 033122 032737 010000 002476 IFT32: BIT #ILLGAL,FLAGST ;IF ILLEGAL FLAG
5795 033130 001024          BNE IIT32 ;NOT SET, THEN
5796 033132 123765 002374 002444 IGT32: CMPB TRACK,CTKO(R5) ;IF TRACK ADR & CURRENT TRACK OF SELECTED DRV (R5)
5797 033140 001406          BEQ IHT32 ;NOT EQUAL, THEN
5798 033142 012737 000041 002520          MOV #TRKAER,ERRNBR ;SETUP ERR NBR=TRACK ADDRESS ERROR
5799 033150 004737 003060          CALL ERROR ;CALL ERROR
5800 033154 000431          BR UCT32 ;BR TO DOUNTIL 'C'
5801 033156 122737 000040 002442 IHT32: CMPB #40,XERUUT ;IF ERR CODE
5802 033164 001025          BNE UCT32 ;SET=40, THEN
5803 033166 012737 000021 002520          MOV #RECERR,ERRNBR ;SETUP ERRNBR=READ ERR CODE-ERR WRG
5804 033174 004737 003060          CALL ERROR ;CALL ERROR
5805 033200 000417          BR UCT32 ;BR TO DOUNTIL 'C'
5806 033202 123765 002374 002444 IIT32: CMPB TRACK,CTKO(R5) ;IF TRACK ADR & CURRENT TRACK OF SELECTED DRV (R5)
5807 033210 001004          BNE IJT32 ;EQUAL, THEN
5808 033212 012737 000041 002520          MOV #TRKAER,ERRNBR ;SETUP ERR NBR=TRACK ADR ERROR
5809 033220 000407          BR UCT32 ;BR TO DOUNTIL 'C'
5810 033222 122737 000040 002442 IJT32: CMPB #40,XERUUT ;IF ERR CODE NOT
5811 033230 001403          BEQ UCT32 ;SET=40
5812 033232 012737 000021 002520          MOV #RECERR,ERRNBR ;SETUP ERR NBR=READ ERR CODE-ERR WRG
5813 033240 005737 002454          UCT32: TST FIN ;DOUNTIL FINI FLAG
5814 033244 001004          BNE ECT32 ;SET OR
5815 033246 032737 001000 002476          BIT #TRKDON,FLAGST ;TRACKS DONE FLAG
5816 033254 001673          BEQ BCT32 ;SET
5817 033256 042737 001000 002476 ECT32: BIC #TRKDON,FLAGST ;CLEAR TRACKS DONE FLAG
5818 033264 005037 002510          CLR TKSCFG ;CLR TRACK FLAGS
5819 033270 052737 000004 002510          BIS #ILTK,TKSCFG ;SETUP ILLEGAL TRACKS FLAG
5820 033276 032737 010000 002476 IKT32: BIT #ILLGAL,FLAGST ;IF ILLEGAL FLAG
5821 033304 001004          BNE LKT32 ;NOT SET, THEN
5822 033306 052737 010000 002476          BIS #ILLGAL,FLAGST ;SET ILLEGAL FLAG
5823 033314 000403          BR UBT32 ;BR TO DOUNTIL 'C'
5824 033316 042737 010000 002476 LKT32: BIC #ILLGAL,FLAGST ;CLEAR ILLEGAL FLAG
5825 033324 032737 010000 002476 UBT32: BIT #ILLGAL,FLAGST ;DOUNTIL ILLEGAL FLAG CLEAR
5826 033332          XT32: EXIT TST
5827 033336          REGTBL
5828 033336          TTBL 0,0
      033336 032754          T32TBL: .WORD T32MSG
      033340 000000          .WORD 0
    
```


033342 000000
 033344 177777
 033346
 5829 033346 177777
 033350
 033350 006654
 033352 177777
 5830 033354

FRUTBL CTLONL

ENDTST

T32RTB:

T32FTB:

.WORD 0
 .WORD -1
 .WORD -1
 .WORD CTLONL
 .WORD -1

5833 033356 000417
033360 040 122 105

.SBTTL TEST 33 - READ SECTOR-PRT:2 - LGC TST
BR BGNT33 ;BR TO BGN TST
T33MSG: .ASCIZ / READ SECTOR-PRT:2 - LGC TST/
.EVEN

5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864

++
: TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A READ SECTOR IN
: BOTH DENSITIES & RETURN A VALID ERROR CODE. SIMILAR TO
: READ SECTOR PRT:1, BUT WITH DISKETTE IN OPPOSITE DENSITY.

```
-----  
: BGNTST  
: IF LOGIC TEST  
: THEN-SETUP TEST IDENT  
: CALL DEVICE DENSITY CK  
: SET DENSITY CONTROL=DISK DEN  
: CALL READ SECTOR  
: CALL READ ERROR CODE  
: IF NO COMMAND ERRORS  
: THEN-CALL ERROR CK  
: CALL COMPLIMENT DENSITY  
: CALL READ SECTOR  
: CALL READ ERROR CODE  
: IF NO COMMAND ERRORS  
: THEN-CALL ERROR CK  
: ENDIF  
: NOP  
: ENDIF  
: ENDTST  
-----
```

BOARD CALLOUT:
1. CONTROLLER
2. R/W ELECTRONICS

--


```

5867 033416          TSETUP
      033416 012737 033534 002466 BGNT33: MOV #T33TBL,TSTID ;SETUP TEST ID TBL-TEST# 33
      033424 032737 000001 002324 IAT33: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      033432 001436          BEQ XT33 ;BIT SET, THEN
      033434 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
5868 033440 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
5869 033444 004737 020472          CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
5870 033450 004737 011062          CALL READ ;CALL READ SECTOR
5871 033454 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5872 033460          ESCAPE TST ;IF NO COMMAND ERRORS, THEN
5873 033464 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5874 033470 004737 020430          CALL CDENC ;CALL COMPLIMENT DENSITY CONTROL
5875 033474 052737 004000 002476 BIS #NEGTST,FLAGST ;SET FLAGST=NEG TEST FLAG
5876 033502 012737 000030 002464 MOV #DENERR,NGTSE ;SETUP NEGTEST SET ERROR=DEN ERROR
5877 033510 004737 011062          CALL READ ;CALL READ SECTOR
5878 033514 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5879 033520          ESCAPE TST ;IF NO COMMAND ERRORS, THEN
5880 033524 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5881 033530          XT33: EXIT TST
5882 033534          REGTBL CSESAL
      015036          REGS1=CSESAL
5883 033534          TTBL REGCK,0
      033534 033360          T33TBL: .WORD T33MSG
      033536 000001          .WORD REGCK
      033540 000000          .WORD 0
      033542 177777          .WORD -1
      033544          T33RTB:
      033544 015036          .WORD REGS1
      033546 177777          .WORD -1
5884 033550          FRUTBL CTLRWE
      033550          T33FTB:
      033550 006651          .WORD CTLRWE
      033552 177777          .WORD -1
5885
5886 033554          ENDTST

```

5889 .SBTTL TEST 34 - WRITE SECTOR-PRT:2 - LGC TST
033556 000417 BR BGNT34 ;BR TO BGN TST
033560 040 127 122 T34MSG: .ASCIZ / WRITE SECTOR-PRT:2 - LGC TST/
.EVEN

5890
5891 :++
5892 : TEST TO VERIFY THAT THE CONTROLLER WILL COMPLETE A WRITE SECTOR IN
5893 : BOTH DENSITIES & RETURN A VALID ERROR CODE, SIMILAR TO WRITE
5894 : SECTOR PRT:1, BUT WITH DISKETTE IN OPPOSITE DENSITY.

5895 -----
5896 : BGNTST
5897 : IF LOGIC TEST
5898 : THEN-SETUP TEST IDENT
5899 : CALL DEVICE DENSITY CHECK
5900 : SET DENSITY CONTROL=DISK DEN
5901 : CALL WRITE SECTOR
5902 : IF NO COMMAND ERROR (ESCAPE TEST)
5903 : THEN-CALL ERROR CHECK
5904 : CALL COMPLIMENT DENSITY CONTROL
5905 : CALL WRITE SECTOR
5906 : IF NO COMMAND ERROR (ESCAPE TEST)
5907 : THEN-CALL ERROR CHECK
5908 : ENDF
5909 : NOP
5910 : ENDF
5911 : NOP

5912 : ENDF
5913 : ENDTST
5914 -----
5915 : BOARD CALLOUT:
5916 : 1. CONTROLLER
5917 : 2. R/W ELECTRONICS
5918 -----
5919 :--
5920


```

5923 033616          TSETUP
      033616 012737 033730 002466 BGNT34: MOV #T34TBL,TSTID ;SETUP TEST ID TBL-TEST# 34
      033624 032737 000001 002324 IAT34: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      033632 001434          BEQ XT34 ;BIT SET, THEN
      033634 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP

5924
5925 033640 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
5926 033644 004737 020472          CALL SDENC ;CALL SET DENSITY CONTROL=DENSITY STATUS
5927 033650 004737 010744          CALL WRITE ;CALL WRITE SECTOR
5928 033654          ESCAPE TST ;IF NO COMMAND ERROR, THEN
5929 033660 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5930 033664 004737 020430          CALL CDENC ;CALL COMPLIMENT DENSITY CONTROL
5931 033670 052737 004000 002476 BIS #NEGTST,FLAGST ;SET FLAGST-NEG TEST FLAG
5932 033676 012737 000030 002464 MOV #DENERR,NGTSER ;SETUP NEG TEST ERR ERR=DENSITY ERR
5933 033704 004737 010744          CALL WRITE ;CALL WRITE
5934 033710 004737 011340          CALL RDERCD ;CALL READ ERROR CODE
5935 033714          ESCAPE TST ;IF NO COMMAND ERROR
5936 033720 004737 017724          CALL ERRCHK ;CALL ERROR CHECK
5937 033724          XT34: EXIT TST
5938 033730          REGTBL CSESAL

                                REGS1=CSESAL
5939 033730 015036          TTBL REGCK,0
      033730 033560          T34TBL: .WORD T34MSG
      033732 000001          .WORD REGCK
      033734 000000          .WORD 0
      033736 177777          .WORD -1
      033740          T34RTB:
      033740 015036          .WORD REGS1
      033742 177777          .WORD -1
5940 033744          FRUTBL CTLRWE
      033744          T34FTB:
      033744 006651          .WORD CTLRWE
      033746 177777          .WORD -1

5941
5942 033750          ENDTST
    
```

```

5945 .SBTTL TEST 35 - DELETED DATA WRITE PRT:2 - LGC TST
      033752 000422
      033754 040 104 105 T35MSG: .ASCIZ / DELETED DATA WRITE PRT:2 - LGC TST/
      .EVEN

```

5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976

```

:++
: TEST TO VERIFY THAT THE DEVICE SET A DELETED DATA MARK ON THE DISKETTE
: HEADER AND RETRIEVE THIS DATA ON A KNOWN GOOD DISKETTE. THIS IS DONE
: IN OPPOSITE DENSITY OF TEST 1.
-----

```

```

: BGNTST
:   IF LOGIC TEST
:   : THEN-SETUP TEST IDENT
:   :   CALL DEVICE DENSITY CK
:   :   SET DENSITY CONTROL=DENSITY STATUS
:   :   SET DELETED DATA FLAG (BIT#3-CMD)
:   :   CALL WRITE SECTOR
:   :   IF NO COMMAND ERROR (ESCAPE TEST)
:   :   : THEN-CALL READ SECTOR
:   :   :   IF NO COMMAND ERROR (ESCAPE TEST)
:   :   :   : THEN-IF RXESR-DELETED DATA BIT NOT SET
:   :   :   :   : THEN-SET ERROR NUMBER=DELETED DATA ERR
:   :   :   :   :   CALL ERROR
:   :   :   :   :   ENDIF
:   :   :   :   ENDIF
:   :   ENDIF
:   ENDIF
: ENDTST
-----

```

```

: BOARD CALLOUT:
:   1. CONTROLLER
:   2. R/W ELECTRONICS
-----
:--

```



```

5979 034020          TSETUP
      034020 012737 034150 002466 BGNT35: MOV      #T35TBL,TSTID ;SETUP TEST ID TBL-TEST# 35
      034026 032737 000001 002324 IAT35:  BIT      #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      034034 001443          BEQ      XT35      ;BIT SET, THEN
      034036 004737 020736          CALL    LTSTUP   ;CALL LOGIC TEST SETUP
5980 034042 004737 017350          CALL    DENCHK  ;CALL DEVICE DENSITY CHECK
5981 034046 004737 020472          CALL    SDENC   ;CALL SET DENSITY CONTROL=DENSITY STATUS
5982 034052 012737 000010 002402          MOV      #DLDCMD,DELDTAT ;SET DELETED DATA FLAG
5983 034060 004737 010744          CALL    WRITE  ;CALL WRITE SECTOR
5984 034064 004737 011340          CALL    RDERCD ;CALL READ ERROR CODE
5985 034070          IBT35: ESCAPE  TST      ;IF NO COMMAND ERROR, THEN
5986 034074 004737 017724          CALL    ERRCHK ;CALL ERROR CHECK
5987 034100 004737 011062          CALL    READ   ;CALL READ SECTOR
5988 034104          ICT35: ESCAPE  TST      ;IF NO COMMAND ERROR, THEN
5989 034110 032777 000100 146234 IDT35:  BIT      #DLDBIT,ARXDB ;IF RXESR-DELETED DATA BIT
5990 034116 001006          BNE      LDT35   ;NOT SET, THEN
5991 034120 012737 000032 002520          MOV      #DLDTERR,ERRNBR ;SETUP ERROR NUMBER=DELETED DATA ERROR
5992 034126 004737 003060          CALL    ERROR  ;CALL ERROR
5993 034132 000404          BR      XT35   ;BR TO EXIT TST
5994 034134 005037 002402          LDT35: CLR     DELDAT ;CLEAR DELETED DATA MODE
5995 034140 004737 010744          CALL    WRITE  ;CALL WRITE SECTOR - CLR DEL DAT FIELD
5996 034144          XT35:  EXIT   TST
5997 034150          REGTBL CSAL
      015036          REGS1=CSAL
5998 034150          TTBL   REGCK,0
      034150 033754          T35TBL: .WORD  T35MSG
      034152 000001          .WORD  REGCK
      034154 000000          .WORD  0
      034156 177777          .WORD  -1
      034160          T35RTB: .WORD  REGS1
      034160 015036          .WORD  -1
      034162 177777          FRUTBL  CTLRWE
5999 034164          T35FTB: .WORD  CTLRWE
      034164          .WORD  -1
      034164 006651
      034166 177777
6000
6001 034170          ENDTST

```

6004 .SBTTL TEST 36 - DISKETTE & DENSITY DATA CHECK - LGC TST
034172 000425 BR BGNT36 ;BR TO BGN TST
034174 040 104 111 T36MSG: .ASCIZ / DISKETTE & DENSITY DATA CHECK - LGC TST/
.EVEN

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
6041
6042
6043
6044
6045
6046
6047
6048

++
: TEST TO VERIFY WITH A KNOWN GOOD DISKETTE THAT THE DEVICE WILL READ
: AND WRITE TO THE DISKETTE WITHOUT DATA ERRORS. BOTH DENSITIES WILL
: BE DONE.

```
-----  
BGNTST  
: IF LOGIC TEST  
: THEN-SETUP TEST IDENT  
: CALL DENSITY CHECK  
: CALL SETUP DENSITY CONTROL=DENSITY STATUS  
: CLEAR DO FLAG  
: BGND0  
: : SET DATA PATTERN=RANDOM  
: : CALL DATA PATTERN GENERATOR  
: : SET TRACK & SECTOR INITIALIZE FLAG  
: : SET TRACK & SECTOR=SEQUENCE MODE  
: : BGND0  
: : CALL GET TRACK & GET SECTOR  
: : CALL FILL BUFFER  
: : CALL WRITE SECTOR  
: : SETUP TO CLEAR RX INTERNAL BUFFER  
: : CALL FILL BUFFER-CLEAR INTERNAL BUFFER  
: : SETUP DATA BUFFER  
: : CALL READ SECTOR  
: : CALL EMPTY BUFFER  
: : CALL DATA CHECK  
: : IF ERROR  
: : THEN-CALL DATA ANYLSIS ERROR  
: : ENDF  
: : DOUNTIL TRACK & SECTOR DONE OR DATA ERRORS=10  
: : CALL CHANGE DENSITY  
: : SET DENSITY CONTROL=DENSITY STATUS  
: : INCREMENT DO FLAG  
: : DOUNTIL DO FLAG=2 OR ABORT FLAG SET  
: : NOP
```

```
ENDIF  
ENDTST  
-----  
BOARD CALLOUT:  
1. CONTROLLER  
2. R/W ELECTRONICS  
-----  
--
```



```

6051
6052 034246          TSETUP
      034246 012737 034504 002466 BGNT36: MOV #T36TBL,TSTID ;SETUP TEST ID TBL-TEST# 36
      034254 032737 000001 002324 IAT36: BIT #LOGICT,TSTMOD ;IF TEST MODE=LOGIC TEST
      034262 001506          BEQ XT36 ;BIT SET, THEN
      034264 004737 020736          CALL LTSTUP ;CALL LOGIC TEST SETUP
6053 034270 005037 002504          CLR TTEMP1 ;CLEAR COUNTER (TEST TEMP #1)
6054 034274 004737 017350          CALL DENCHK ;CALL DENSITY CHECK
6055 034300 004737 020472          CALL SDENC ;CALL SETUP DENSITY CONTROL=DENSITY STATUS
6056 034304 005037 012660          BBT36: CLR PAT ;SETUP DATA PATTERN=RANDOM
6057 034310 004737 012306          CALL STDATP ;CALL SET DATA PATTERN
6058 034314 052737 001400 002510          BIS #ITK!ISC,TKSCFG ;SET TRACK & SECTOR INITIALIZE FLAGS
6059 034322 052737 000003 002510          BIS #STK!SSC,TKSCFG ;SET TRACK & SECTOR SEQUENCE MODE FLAGS
6060 034330 004737 012662          BCT36: CALL GETTRK ;CALL GET TRACK
6061 034334 004737 013104          CALL GETSEC ;CALL GET SECTOR
6062 034340 004737 010510          CALL FILBUF ;CALL FULL BUFFER
6063 034344 004737 010744          CALL WRITE ;CALL WRITE SECTOR
6064 034350 004737 013642          CALL CLRDAT ;CALL CLEAR DATA BUFFER
6065 034354 012737 036622 002362          MOV #DATBUF,FILADR ;SETUP TO CLEAR RX INTERNAL BUFFER
6066 034362 004737 010510          CALL FILBUF ;CLEAR THE BUFFER
6067 034366 012737 036222 002362          MOV #DATPAT,FILADR ;SETUP DATA BUFFER ADDRESS
6068 034374 004737 011062          CALL READ ;CALL READ SECTOR
6069 034400 004737 010626          CALL EMPBUF ;CALL EMPTY BUFFER
6070 034404 004737 013246          CALL DATAK ;CALL DATA CHECK
6071 034410 022737 000012 013520          UCT36: CMP #10,DAERCT ;DOUNTIL DATA ERROR COUNT
6072 034416 001410          BEQ ECT36 ;EQUALS 10, OR
6073 034420 032737 001000 002476          BIT #TRKDON,FLAGST ;TRACKS DONE FLAG
6074 034426 001740          BEQ BCT36 ;SET, AND
6075 034430 032737 002000 002476          BIT #SECDON,FLAGST ;SECTORS DONE FLAG
6076 034436 001734          BEQ BCT36 ;SET
6077 034440 004737 020430          ECT36: CALL CDENC ;CALL COMPLIMENT DENSITY CONTROL
6078 034444 004737 011172          CALL SETDN ;CALL SET DENSITY
6079 034450 005237 002504          INC TTEMP1 ;INCREMENT COUNTER
6080 034454 012737 000100 002370          MOV #64.,WDCNT ;SET WORD COUNT
6081 034462 005737 002454          UBT36: TST FIN ;DOUNTIL FIN FLAG
6082 034466 001004          BNE XT36 ;SET OR
6083 034470 022737 000002 002504          CMP #2,TTEMP1 ;COUNT
6084 034476 001302          BNE BBT36 ;EAUALS 2
6085 034500          XT36: EXIT TST
6086
6087 034504          REGTBL CSESAL
      015036          REGS1=CSESAL
6088 034504          TTBL REGCK,0
      034504 034174          T36TBL: .WORD T36MSG
      034506 000001          .WORD REGCK
      034510 000000          .WORD 0
      034512 177777          .WORD -1
      034514          T36RTB:
      034514 015036          .WORD REGS1
      034516 177777          .WORD -1
6089 034520          FRUTBL CTLRWE
      034520          T36FTB:
      034520 006651          .WORD CTLRWE
      034522 177777          .WORD -1
6090
6091 034524          ENDTST
    
```

HARDWARE TESTS MACRO M1110 13-AUG-79 15:33 PAGE 235
TEST 36 - DISKETTE & DENSITY DATA CHECK - LGC TST

N 13

SEQ 0169

6099 034526
6100

ENDMOD

6103
6104
6115
6116
6144
6145 034526
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156 034526
6157
6158 034530
6159 034540
6160 034550
6161 034562
6162 034574
6163
6169 034606
6170
6171 034606
6172 034621
6173 034634
6174 034647
6175 034662
6176
6177
6178
6185

.NLIST BEX,ME
.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 MSG3,4,0,177777,0,,1,,YES
GPRMD MSG4,6,0,177777,0,,1,,YES
GPRMD MSG4A,10,0,177777,0,7,YES

ENDHRD

122 130 040
126 105 103
104 122 111
105 130 120
102 122 055

MSG1: .ASCIZ /RX BUS ADR/
MSG2: .ASCIZ /VECTOR ADR/
MSG3: .ASCIZ /DRIVE # /
MSG4: .ASCIZ /EXP WRD-CR/
MSG4A: .ASCIZ /BR-LEVEL /
.EVEN

6188
6189
6190
6191
6192
6193
6194
6195
6196
6197
6198
6199 034676
6200
6201 034700
6202 034706
6203 034710
6204 034716
6205 034724
6206 034732
6207 034744
6208 034754
6209 034766
6210 034774
6211 034776
6212 035004
6219 035012
6220
6221 000015
6222 000012
6223 035012 105 130 120
6224 035044 124 105 123
6225 035057 104 111 101
6226 035106 040 040 040
6227 035160 040 040 040
6228 035213 040 040 040
6229 035301 040 040 040
6230 035332 040 040 040
6231 035403 040 040 040
6232 035461 040 040 040
6233 035552 040 040 040
6237 035651 124 131 120
6238 035677 114 117 107
6239 035722 106 125 116
6240 035745 110 101 122
6241 036014 116 117 116
6242 036053 105 130 124
6243 036124 124 105 123
6244 036150 040 040 040
6245
6246

.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 MSG6,2,1,YES
XFERF 1\$
GPRML MSG7,2,2,YES
1\$: GPRML MSG8,4,LOGICT,YES
GPRML MSG9,4,FUNCTT,YES
GPRMD MSG10,0,0,177777,1,177777,YES
GPRMA MSG14,24,0,0,177777,YES
GPRMD MSG15,26,0,030000,0,3,YES
GPRML MSG17,2,100,YES
XFERF 6\$
GPRML MSG20,12,20,YES
6\$: GPRML MSG5,12,SIDFLG,YES
ENDSFT

CR=15 ;CARRIAGE RETURN
LF=12 ;LINE FEED
MSG5: .ASCIZ /EXPANSION WORD TYPE <CR> /
MSG6: .ASCIZ /TEST HELP /
MSG7: .ASCII /DIAGNOSTIC MODES ARE: /<CR><LF>
.ASCII / LOGIC TEST, FUNCTION TEST, OR BOTH/<CR><LF>
.ASCII / -FUNCTION TESTS (1-10)/<CR><LF>
.ASCII / ACT AS QUICK VERIFY & REPORT FAILING FUNCTIONS/<CR><LF>
.ASCII / -LOGIC TESTS (11-36)/<CR><LF>
.ASCII / ANALYZE FAILURE & GIVE ERROR INFO/<CR><LF>
.ASCII / REPORT FIELD REPLACEABLE UNITS "FRU'S"/<CR><LF>
.ASCII / ->DEVICE FATAL THRESHOLD LEVEL (DVTL) IS SET = 1/<CR><LF>
.ASCII / "DVTL" = NO. OF HARD ERRS THAT CAUSE DEVICE FATAL ERR/<CR><LF>
MSG8: .ASCIZ /TYPE "CR" TO CONTINUE/
MSG9: .ASCIZ /LOGIC TEST MODE /
MSG10: .ASCIZ /FUNCTION TEST MODE/
MSG14: .ASCIZ /HARD ERR -> DEVICE FATAL THRESHOLD LVL/
MSG15: .ASCIZ /NON-EXISTANT MEM ADR (NXM TST)/
MSG17: .ASCIZ /EXTENDED ADR BITS: 13 & 12 (NPR-NXM TST)/
MSG20: .ASCIZ /TEST CONTROL FLAGS /

.EVEN

6249
6250
6251
6252
6253 036222 000400
6256
6257
6258
6259
6260
6261 036622 000400
6264 037222 000000
6265 037224 000000
6266
6273
6274
6275 037226 000000
6276 037630
6277
6278 037630
037634
6279 037634
6280
6281 037634
6282 037634
6283 037640 177170
6284 037642 000264
6285 037644 000000
6286 037646 000000
6287 037650 000005
6288 037652
6289 037652
6290 037656 177170
6291 037660 000264
6292 037662 000001
6293 037664 000000
6294 037666 000005
6295 037670
6296 037670
6297 000001

```
.SBTTL - RX02 FILL BUFFER AREA
-----
DATPAT: .REPT 256. ;DATA PATTERN - RX02 FILL BUFFER
-----

.SBTTL - RX02 EMPTY BUFFER AREA
-----
DATBUF: .REPT 256. ;DATA BUFFER - RX02 EMPTY BUFFER
        .WORD 0
        .WORD 0
-----

.SBTTL - PATCH AREA
-----
PATCH: 0 ;PATCH AREA
        =.+400
-----

LASTAD
L$LAST::
ENDMOD

BGNSETUP 2
BGNPTAB
177170
264
0
0
5
ENDPTAB
BGNPTAB
177170
264
1
0
5
ENDPTAB
ENDSETUP

.END
```

PARAMETER CODING
SYMBOL TABLE

MACRO M1110 13-AUG-79 15:33 PAGE 240-1

E 14

SEQ 0173

ABORT	002452	BGNT35	034020	CMDERR=	000020	C\$ESCA=	000010	DISKSP=	000030
ACLOW =	000010	BGNT36	034246	CMDMSG	007164	C\$ESEG=	000005	DLDBIT=	000100
ACLOWD=	000037	BGNT4	022766	CMDM0	007237	C\$ESUB=	000003	DLDCMD=	000010
ACLOWF=	000050	BGNT5	023102	CMDM1	007255	C\$ETST=	000001	DLDERR=	000007
ADR =	000020 G	BGNT6	023230	CMDM2	007274	C\$EXIT=	000032	DLDTER=	000032
ADRTST	022066	BGNT7	023414	CMDM3	007313	C\$GETB=	000026	DLPDN =	000010
ADSCMS	007760	BGNT8	023664	CMDM4	007331	C\$GETW=	000027	DLY	012026
ADTKMS	010257	BGNT9	024036	CMDM5	007347	C\$GMAN=	000043	DMSG1	013561
ALGO2E=	000040	BITCNT	014672	CMDM6	007374	C\$GPHR=	000042	DMSG1B	013526
ASSEMB=	000010	BITLIM	014674	CMDM7	007430	C\$GPLO=	000030	DMSG2	013615
AWDN	012032	BITOFF	014676	CMDM8	007452	C\$GPRI=	000040	DNBIT =	000040
AWTR	012110	BIT0 =	000001 G	CMDPE	007502	C\$INIT=	000011	DNFLAG	012030
BACDB	013652	BIT00 =	000001 G	CMERMS	007216	C\$INLP=	000020	DNNINT=	000015
BADCK	013276	BIT01 =	000002 G	CMFTMS	007204	C\$MANI=	000050	DNNOTR=	000063
BADWRD	014702	BIT02 =	000004 G	CMPPWD	014700	C\$MEM =	000031	DNWTMT	002474
BAUWCH	011700	BIT03 =	000010 G	CONTRL=	000002	C\$MSG =	000023	DOOROP=	000026
BBT10	024314	BIT04 =	000020 G	CONTSW=	000022	C\$OPEN=	000034	DRIVE	002406
BBT21	027600	BIT05 =	000040 G	CR =	000015 G	C\$PNTB=	000014	DRIVE1=	000400
BBT23	030336	BIT06 =	000100 G	CRCERB=	000001	C\$PNTF=	000017	DRV DEN=	000040
BBT24	030546	BIT07 =	000200 G	CRCERR=	000004	C\$PNTS=	000016	DRV OFF	002420
BBT27	031400	BIT08 =	000400 G	CSERTB	014776	C\$PNTX=	000015	DRV PRT	002514
BBT31	032474	BIT09 =	001000 G	CSESAL=	015036	C\$QIO =	000377	DRV RDY=	000200
BBT32	033042	BIT1 =	000002 G	CSESIT=	015056	C\$RDBU=	000007	DRVWRG=	000027
BBT36	034304	BIT10 =	002000 G	CSESND=	015076	C\$REFG=	000047	DRV1 =	000020
BBT7	023460	BIT11 =	004000 G	CSESRS=	015066	C\$RESE=	000033	DUMSG1	022012
BCGSC	013202	BIT12 =	010000 G	CSONLY=	015026	C\$REVI=	000003	DVDNCK	011634
BCT32	033044	BIT13 =	020000 G	CSRCHK	013750	C\$RFLA=	000021	DVFERR=	004000
BCT36	034330	BIT14 =	040000 G	CSR CMP	014260	C\$RPT =	000025	DVRDYE=	000025
BET31	032642	BIT15 =	100000 G	CSRERR=	000033	C\$SEFG=	000046	DVRYCK	017310
BGNT1	022364	BIT2 =	000004 G	CSRMSK	014262	C\$SPRI=	000041	DVSTCK	017140
BGNT10	024254	BIT3 =	000010 G	CSRSET	014270	C\$SVEC=	000037	DVTL	002320
BGNT11	024412	BIT4 =	000020 G	CTKO	002444	C\$TPRI=	000013	DX	012024
BGNT12	024650	BIT5 =	000040 G	CTK1	002445	DAERCT	013520	EADDC	017514
BGNT13	025074	BIT6 =	000100 G	CTLINF=	006646	DATABF=	000055	EADRC	017346
BGNT14	025316	BIT7 =	000200 G	CTLONL=	006654	DATACK	013246	EADSC	017220
BGNT15	025546	BIT8 =	000400 G	CTLRWE=	006651	DATASB	013522	EAERR	003256
BGNT16	025662	BIT9 =	001000 G	C\$AU =	000052	DATAWS	013524	EAFRU	005506
BGNT17	026276	BOE =	000400 G	C\$AUTO=	000061	DATAO	012430	EANAT	027076
BGNT18	026452	BRONPT	012372	C\$BRK =	000022	DATA1	012446	EAPCE	007106
BGNT19	027146	BTRK	002451	C\$BSEG=	000004	DATBUF	036622	EARCR	014240
BGNT2	022514	BTRP4 =	000004	C\$BSUB=	000002	DATBYT	012656	EASRC	014400
BGNT20	027370	BTRP6 =	000006	C\$CEFG=	000045	DATCK =	000004	EATKE	017712
BGNT21	027534	BYTCNT	013514	C\$CLCK=	000062	DATER =	000005	EBCMD =	000002
BGNT22	030166	BYTNUM	013516	C\$CLEA=	000012	DATPAT	036222	EBDCK	013470
BGNT23	030306	CABLES=	000010	C\$CLOS=	000035	DBRERR=	000034	EBDDC	017434
BGNT24	030510	CDENC	020430	C\$CLP1=	000006	DDCFLG=	000002	EBGEN	015152
BGNT25	030762	CDERCK	011544	C\$CVEC=	000036	DEL DAT	002402	EBGTK	013062
BGNT26	031140	CDMS	010177	C\$DCLN=	000044	DENBIT=	000400	EBI1	021720
BGNT27	031332	CEINIT=	015046	C\$DODU=	000051	DENCHK	017350	EBNAT	027030
BGNT28	031530	CKBITS	014522	C\$DRPT=	000024	DENDSK=	000020	EBCRCR	013716
BGNT29	031732	CKERR =	020000	C\$DU =	000053	DENERR=	000030	EBTKP	010062
BGNT3	022650	CLRCR	021122	C\$EDIT=	000003	DENMIX=	000031	EBT11	024532
BGNT30	032132	CLR DAT	013642	C\$ERDF=	000055	DENSTA	002414	EBT12	024744
BGNT31	032430	CLRDEV	010472	C\$ERHR=	000056	DENSTY	002412	EBT13	025176
BGNT32	033002	CLRERR	010300	C\$ERRO=	000060	DFPTBL	002276 G	EBT18	026530
BGNT33	033416	CLRRGS	017124	C\$ERSF=	000054	DIAGMC=	000000	EBT31	032604
BGNT34	033616	CMD	002400	C\$ERSO=	000057	DISKET=	000014	ECERNB	020164

PARAMETER CODING
SYMBOL TABLE

ECERNT 015164	END131 012612	ERRNST= 000016	F\$DU = 000016	IAGEN 015110
ECERR 003230	EPECK 020140	EROLD 003340	F\$END = 000041	IAGSC 013106
ECNAT 027074	ERIDNT 003502 G	ERROR 003060	F\$HARD= 000004	IAGTK 012664
ECSTA 015330	ERMSTB 003534	ERRREG 015444	F\$HW = 000013	IAI1 021622
ECTAB 016044	ERMS0 003706	ERRSAV 003336	F\$INIT= 000006	IANAT 026776
ECT18 026674	ERMS10 004015	ERRTYP 002516 G	F\$JMP = 000050	IAPCE 007056
ECT21 027660	ERMS11 004035	ERTKMS 010241	F\$MOD = 000000	IARCR 013672
ECT32 033256	ERMS12 004056	ESERTB 014746	F\$MSG = 000011	IASCP 007676
ECT36 034440	ERMS13 004112	ESRCHK 014112	F\$PROT= 000021	IASDC 020500
ECO 016116	ERMS14 004146	ESRCMP 014264	F\$PWR = 000017	IASRC 014340
EC1 016136	ERMS15 004177	ESRMSK 014266	F\$RPT = 000012	IATA 015334
EC11 016407	ERMS16 004226	ESRSET 014272	F\$SEG = 000003	IATKE 017520
EC12 016451	ERMS17 004264	EVL = 000004 G	F\$SOFT= 000005	IATKP 010006
EC13 016477	ERMS19 004314	EXMS 010156	F\$SRV = 000010	IAT1 022372
EC14 016550	ERMS2 003717	EXTADR 002366	F\$SUB = 000002	IAT10 024262
EC15 016567	ERMS20 004333	E\$END = 002100	F\$SW = 000014	IAT11 024420
EC16 016640	ERMS21 004356	E\$LOAD= 000035	F\$TEST= 000001	IAT12 024656
EC17 016670	ERMS22 004402	FBCMD = 000000	GETREG 012244	IAT13 025102
EC2 016175	ERMS23 004434	FILADR 002362	GETSEC 013104	IAT14 025324
EC20 016716	ERMS24 004467	FILBUF 010510	GETTRK 012662	IAT15 025554
EC22 016753	ERMS25 004503	FILERR= 000012	GTECEN 015106	IAT16 025670
EC23 017013	ERMS26 004532	FIN 002454	GTECOF 017106	IAT17 026304
EC24 017033	ERMS27 004553	FLAGSP 002500	G\$CNT0= 000200	IAT18 026460
EC25 017052	ERMS28 004564	FLAGST 002476	G\$DELM= 000372	IAT19 027154
EC4 016234	ERMS29 004575	FLGDRS 002502	G\$DISP= 000003	IAT2 022522
EC5 016274	ERMS3 003731	FLOAT0 012456	G\$EXCP= 000400	IAT20 027376
EC7 016337	ERMS30 004625	FLOAT1 012524	G\$HILI= 000002	IAT21 027542
EDECK 020122	ERMS31 004655	FONZFG= 010000	G\$LOLI= 000001	IAT22 030174
EDGSC 013236	ERMS32 004670	FRUMOA 005713	G\$NO = 000000	IAT23 030314
EDSRC 014462	ERMS33 004711	FRUMOB 005746	G\$OFFS= 000400	IAT24 030516
EDT18 026722	ERMS34 004733	FRUM00 005650	G\$OFFSI= 000376	IAT25 030770
EDT21 027662	ERMS4 003742	FRUM1 006001	G\$PRMA= 000001	IAT26 031146
EEECK 020072	ERMS40 004756	FRUM10 006361	G\$PRMD= 000002	IAT27 031340
EEFRU 005604	ERMS41 004777	FRUM11 006426	G\$PRML= 000000	IAT28 031536
EESRC 014516	ERMS42 005027	FRUM12 006451	G\$RADA= 000140	IAT29 031740
EET32 033116	ERMS43 005054	FRUM13 006504	G\$RADB= 000000	IAT3 022656
EFENV 003466	ERMS44 005102	FRUM14 006544	G\$RADD= 000040	IAT30 032140
EFERR 003330	ERMS45 005125	FRUM2 006035	G\$RADL= 000120	IAT31 032436
EFFRU 005576	ERMS46 005153	FRUM3 006070	G\$RADO= 000020	IAT32 033010
EFT31 032716	ERMS47 005202	FRUM4 006120	G\$XFER= 000004	IAT33 033424
EF.CON= 000036 G	ERMS48 005230	FRUM5 006140	G\$YES = 000010	IAT34 033624
EF.NEW= 000035 G	ERMS49 005260	FRUM6 006173	HDRPRT= 000100	IAT35 034026
EF.PWR= 000034 G	ERMS5 003752	FRUM7 006221	HDSFDG= 000056	IAT36 034254
EF.RES= 000037 G	ERMS50 005307	FRUM8 006255	HELP = 000000	IAT4 022774
EF.STA= 000040 G	ERMS51 005337	FRUM9 006324	HETLCT 003472	IAT5 023110
EGFRU 005554	ERMS6 003763	FRUS1 = 000000	HOE = 100000 G	IAT6 023236
EGSRC 014330	ERMS7 003774	FRUTAD 005610	HRDERR= 040000	IAT7 023422
EIECK 020070	ERNBEV 003344	FRUTBM 005612	IACDC 020432	IAT8 023672
EMBUFF= 000020	ERNTCK 020170	FTERCT 014274	IADDC 017402	IAT9 024044
EMDCK 013444	ERRBIT= 100000	FTSTUP 020700	IADID 020614	IBDCK 013312
EMPADR 002360	ERRBLK 002524 G	FUNCT = 000000	IADRC 017316	IBDDC 017410
EMPBUF 010626	ERRCHK 017724	FUNCTT= 000002	IADSC 017144	IBDSC 017152
EMPERR= 000013	ERRCMD 002422	FUNTST= 000040	IAECK 017726	IBE = 010000 G
ENDCK 013434	ERRCTR 003342	F\$AU = 000015	IAENC 020172	IBECK 017742
ENDDCK 013506	ERRFLG= 100000	F\$AUTO= 000020	IAENV 003346	IBEMB 010672
ENLD 012652	ERRMSG 002522 G	F\$BGN = 000040	IAERR 003154	IBENC 020204
ENDXER 016012	ERRNBR 002520 G	F\$CLEA= 000007	IAFRU 005406	IBENV 003364

PARAMETER CODING
SYMBOL TABLE

MACRO M1110 13-AUG-79 15:33 PAGE 240-3

G 14

SEQ 0175

IBERR	003140	ID	002336	G	IJT32	033222	LBDDC	017430	L\$EXP4	002064	G	
IBFLB	010554	IDCOMP	013020		IKERR	003260	LBFRU	005456	L\$EXP5	002066	G	
IBFRU	005424	IDDDC	017354		IKT32	033276	LBGEN	015134	LSHARD	034530	G	
IBGEN	015116	IDDSMS	020642		ILERR	003274	LBGTK	013006	LSHIME	002120	G	
IBGSC	013140	IDECK	020022		ILLERC=	000017	LB11	021704	LSHPCP	002016	G	
IBGTK	012770	IDENC	020226		ILLGAL=	010000	LBNAT	027022	LSHPTP	002022	G	
IB11	021666	IDENT1	015446		ILTK =	000004	LBTKP	010056	LSHW	002276	G	
IBNAT	027006	IDENV	003404		IMDCK	013352	LBUWCH	011740	LSICP	002104	G	
IBPCE	007122	IDERR	003242		IMECK	020074	LCMD	002424	LSINIT	021214	G	
IBREC	011414	IDGSC	013224		INDCK	013402	LCSTA	015314	LSLADP	002026	G	
IBRED	011124	IDSRC	014446		INECK	017752	LCT24	030634	LSLAST	037634	G	
IBSDN	011234	IDSSMS	020657		INFCTL=	006640	LCT29	032042	LSLOAD	002100	G	
IBSRC	014416	IDTKE	017556		INIT	021214	LDT11	024474	LSLUN	002074	G	
IBSTA	015270	IDT11	024454		INITDN=	000004	LDT31	032562	LSMREV	002050	G	
IBTKE	017540	IDT14	025376		INITER	021462	LDT35	034134	LSNAME	002000	G	
IBTKP	010042	IDT16	026050		INITTK	013102	LET14	025446	LSPRIO	002042	G	
IBT11	024502	IDT18	026700		INTERF=	000000	LET21	027770	LSPROT	002310	G	
IBT12	024714	IDT24	030640		INTERT	002404	LEUWCH	012000	LSPRT	002112	G	
IBT13	025140	IDT29	032016		INTER1	021530	LF =	000012	LSREPP	002062	G	
IBT14	025342	IDT31	032516		INTFCB=	000024	LFFRU	005562	LSREV	002010	G	
IBT16	025710	IDT32	033066		INTFSW=	000016	LGFRU	005550	LSRPT	021212	G	
IBT18	026506	IDT35	034110		INTIAL	010440	LGT31	032712	LS\$SOFT	034700	G	
IBT19	027170	IDU =	000040	G	INTNDN=	000014	LKERR	003312	LS\$SPC	002056	G	
IBT29	031776	IEDCK	013322		INTONL=	006644	LKT32	033316	LS\$PCP	002020	G	
IBT30	032334	IEERR	003164		INTRHD	012276	LMECK	020114	LS\$PTP	002024	G	
IBT35	034070	IER =	020000	G	INTTBL	021742	LOAD	012614	LS\$TA	002030	G	
IBUWCH	011704	IESRC	014466		IOECK	017762	LOE =	040000	LS\$W	002320	G	
IBWRT	011014	IETKE	017566		IPECK	020124	LOGICT=	000001	LS\$TEST	002114	G	
ICDDC	017464	IET14	025422		ISC =	001000	LOT =	000010	LS\$TIML	002014	G	
ICDSC	017160	IET21	027706		ISR =	000100	G	LRXCSR	002426	LS\$UNIT	002012	G
ICECK	020012	IET32	033074		ITK =	000400		LRXESR	002430	L10000	002310	
ICEMB	010712	IEUWCH	011752		IXE =	004000	G	LSIFLG=	000400	L10002	002350	
ICENC	020214	IFDCK	013332		ISAU =	000041		LTSTUP	020736	L10003	002532	
ICERR	003214	IFENV	003420		ISAUTO=	000041		LSACP	002110	G	L10004	002540
ICFLB	010574	IFERR	003062		ISCLN =	000041		LSAPT	002036	G	L10005	002546
ICGTK	012742	IFFRU	005524		ISDU =	000041		LSAU	022062	G	L10006	003532
ICNAT	027046	IFSRC	014346		ISHRD =	000041		LSAUT	002070	G	L10007	021212
ICPCE	007130	IFTKE	017612		ISINIT=	000041		LSAUTO	022054	G	L10010	021522
ICREC	011434	IFT21	027714		ISMOD =	000041		LS\$CCP	002106	G	L10011	021774
ICRED	011144	IFT31	032652		ISMSG =	000041		LS\$CLEA	021764	G	L10012	022010
ICSRC	014432	IFT32	033122		IS\$PROT=	000040		LS\$CO	002032	G	L10013	022060
ICSTA	015276	IGENV	003430		IS\$PTAB=	000041		LS\$DEPO	002011	G	L10014	022064
ICTKE	017550	IGERR	003200		IS\$PWR =	000041		LS\$DESC	002122	G	L10015	022454
ICT11	024546	IGFRU	005532		IS\$RPT =	000041		LS\$DESP	002076	G	L10016	022614
ICT12	024756	IGSRC	014304		IS\$SEG =	000041		LS\$DEVP	002060	G	L10017	022730
ICT13	025210	IGTKE	017622		IS\$SETU=	000041		LS\$DISP	002164	G	L10020	023046
ICT14	025352	IGT21	027724		IS\$SFT =	000041		LS\$DLY	002116	G	L10021	023164
ICT16	025754	IGT31	032660		IS\$SRV =	000041		LS\$DTP	002040	G	L10022	023350
ICT18	026652	IGT32	033132		IS\$SUB =	000041		LS\$DTYP	002034	G	L10023	023612
ICT21	027640	IHENV	003444		ISTST =	000041		LS\$DU	021776	G	L10024	024002
ICT23	030372	IHTKE	017654		JSJMP =	000167		LS\$DUT	002072	G	L10025	024220
ICT24	030566	IHT32	033156		LACDC	020454		LS\$DVTY	002154	G	L10026	024362
ICT29	032012	IEECK	020056		LA11	021644		LS\$EF	002052	G	L10027	024620
ICT31	032510	IEERR	003124		LANAT	027036		LS\$ENVI	002044	G	L10030	025030
ICT35	034104	IITKE	017662		LAPCE	007076		LS\$ERRT	002516	G	L10031	025260
ICUWCH	011712	IIT32	033202		LASDC	020524		LS\$ETP	002102	G	L10032	025500
ICWRT	011034	IJERR	003114		LATKP	010130		LS\$EXP1	002046	G	L10033	025630

PARAMETER CODING
SYMBOL TABLE

MACRO M1110 13-AUG-79 15:33 PAGE 240-4

H 14

SEQ 0176

L10034	026232	NOTRBT=	000057	PRTREG	007564	RXPRI	002356	TRKAER=	000041
L10035	026426	NPRERR=	000053	PRTSEC	007674	RX2BIT=	004000	TRKCN	013100
L10036	026772	NPRJPR=	C00020	PRTSTA	015240	SCPRT =	000002	TRKDON=	001000
L10037	027336	NXMADR	002344 G	PRTTRK	010002	SDCMD =	000010	TRKSEQ	002330 G
L10040	027500	NXMBIT=	004000	PRTXOS	002714	SDENC	020472	TRPMS1	022212
L10041	030124	NXMERR=	000052	PRTX1S	002734	SDKYWD=	000036	TSAVE1	002506
L10042	030244	OD	002334 G	PRTX2S	002756	SDRDYE=	000024	TSDGMS	020306
L10043	030446	ODCOMP	013032	PRTX3S	003002	SECAER=	000042	TSDGM1	020364
L10044	030706	ODTNOT=	000001	PRTX4S	003030	SEC DON=	002000	TSEC	002447
L10045	031076	ONEFIL=	000000	PTERTY	003474	SECTOR	002376	TSTDBG	020240
L10046	031276	OSAPTS=	000000	PTUTMS	020560	SEKERR=	000006	TSTID	002466
L10047	031466	OSAU =	000001	PWDNRY	026175	SETDCD	011462	TSTMOD	002324 G
L10050	031662	OSBGNR=	000000	PWRMS	026122	SETDN	011172	TSTPAT	002326 G
L10051	032076	OSBGNS=	000001	PWUPRY	026214	SETSCD	011502	TTEMP1	002504
L10052	032376	OSDU =	000001	RANDAT	012566	SETUP	021362	TTRK	002446
L10053	032750	OSERRT=	000001	RANGEN	010344	SFPTBL	002320 G	TYPERR	002460
L10054	033354	OSGNSW=	000001	RANUM	010436	SFTSTS	002450	TSARGC=	000002
L10055	033554	OSPOIN=	000001	RAN1	010432	SIDE	002410	TSCODE=	005130
L10056	033750	OSSETU=	000001	RAN2	010434	SIDE1 =	001000	TSERNR=	000074
L10057	034170	PAT	012660	RDERCD	011340	SIDFLG=	010000	TSEXCP=	000000
L10060	034524	PATCH	037226	RDERR =	000003	SIDPRT	002515	TSFLAG=	000040
L10061	034606	PAT125	012532	RDSTAT	011266	SIDRDY=	000002	TSFREE=	037670
L10062	035012	PAT333	012556	READ	011062	SIDWRG=	000026	TSGMAN=	000000
L10063	037640	PG	012434	READ1	011112	SSC =	000002	TSHILI=	000003
L10064	037656	PHYDRV=	000006	RECADR	002364	STAFLG=	100000	TSLAST=	000001
L10065	037652	PLOC	021524	RECCMD=	000016	START	021252	TSLOLI=	000000
L10067	037670	PNT =	001000 G	RECERN	002462	STARTO	021262	TLSYM=	010000
MAXSEC	002342 G	POWRSP=	000012	RECERR=	000021	START1	021322	TSLTNO=	000044
MINSEC	002340 G	PRESCK	013044	RECFLG=	000200	STATER	017222	TSNEST=	177777
MOTOR =	000032	PRI =	002000 G	RECTST=	000200	STCMD =	000012	TSNSO =	000000
MSG1	034606	PRILEV=	000054	REGACT	002440	STDATP	012306	TSNS1 =	000005
MSG10	035745	PRIMSG	030006	REGCK =	000001	STDNER=	000035	TSPCNT=	000000
MSG14	036014	PRIORT	002416	REGEXP	002436	STK =	000001	TSPTAB=	010066
MSG15	036053	PRITAB	021732	REGSCK	013664	STTK76	021174	TSPTHV=	000002
MSG17	036124	PRI00 =	000000 G	REGS1 =	015036	SUDVCD	021014	TSPTHV=	000002
MSG2	034621	PRI01 =	000040 G	REGS2 =	000000	SUM	012654	TSSAVL=	177777
MSG20	036150	PRI02 =	000100 G	REGS3 =	000000	SURGCK	014276	TSSGL=	177777
MSG3	034634	PRI03 =	000140 G	REGS4 =	000000	SUTSFG	020772	TSSIZE=	000016
MSG4	034647	PRI04 =	000200 G	REGS5 =	000000	SVCGBL=	000000	TSSUBN=	000000
MSG4A	034662	PRI05 =	000240 G	REGS6 =	000000	SVCINS=	177777	TSTAGL=	177777
MSG5	035012	PRI06 =	000300 G	RESFLG=	040000	SVC SUB=	177777	TSTAGN=	010070
MSG6	035044	PRI07 =	000340 G	RESTAR	021332	SVCTAG=	177777	TSTEMP=	000000
MSG7	035057	PROPRT=	000010	REVC =	000000	SVCTST=	177777.	TSTEST=	000044
MSG8	035677	PROTCT	002472	RGERNB	020166	SWREG	002332 G	TSTSTM=	177777
MSG9	035722	PRTB0	002526 G	RGERTB	014706	SYFERR=	002000	TSTSTS=	000001
NAT	026774	PRTB0S	002550	RGETPT	014670	SYSERR	002456	TSSAU =	010014
NATADR	027102	PRTB1	002534 G	RGPRT =	000004	S\$LSYM=	010000	TSSAUT=	010013
NATCTR	027100	PRTB1S	002570	RSC =	000000	TCMDCT	002470	TSSCLE=	010011
NCMD	011542	PRTB2	002542 G	RSCMD =	000006	TFDCK	013342	TSSDAT=	010067
NEGTST=	004000	PRTB2S	002612	RTBADR	014704	TGMS	010220	TSSDU =	010012
NEW	021352	PRTB3S	002636	RTK =	000000	TKERCK	017516	TSSHAR=	010061
NEXT	021370	PRTB4S	002664	RWELEC=	000004	TKPRT =	000001	TSSHW =	010000
NGTSER	002464	PRTCDE	007040	RXCS	002350	TKSCFG	002510	TSSINI=	010010
NODNBT=	000060	PRTDID	020600	RXCSR	002432	TN =	000044	TSSMSG=	010006
NOITDB=	000061	PRTECD	015744	RXDB	002352	TRACK	002374	TSSPC =	000002
NOITDP=	000062	PRTFRU	005404	RXESR	002434	TRAP	022324	TSSPRO=	010001
NOPWR =	000034	PRTGMS	007620	RXINIT=	040000	TRBIT =	000200	TSSPTA=	010066

PARAMETER CODING
SYMBOL TABLE

MACRO M1110 13-AUG-79 15:33 PAGE 240-5

I 14

SEQ 0177

TSSRPT=	010007	T14FTB	025474	T24RTB	030700	T35FTB	034164	WAIT	011610
TSSSOFF=	010062	T14MSG	025264	T24TBL	030670	T35MSG	033754	WATCH	011662
TSSSW =	010002	T14RTB	025472	T25	030710 G	T35RTB	034160	WC	002443
TSSTES=	010060	T14TBL	025462	T25FTB	031072	T35TBL	034150	WCERR =	000023
TOFTB	006660	T15	025502 G	T25MSG	030712	T36	034172 G	WCFRU	005462
TOFTO	006640	T15FTB	025624	T25RTB	031064	T36FTB	034520	WCOVFE=	000051
TOFT1	006736	T15MSG	025504	T25TBL	031054	T36MSG	034174	WCOVRF=	002000
TOFT11	006763	T15RTB	025620	T26	031100 G	T36RTB	034514	WDCNT	002370
TOFT12	006770	T15TBL	025610	T26FTB	031272	T36TBL	034504	WDDCMD=	000014
TOFT13	006776	T16	025632 G	T26MSG	031102	T4	022732 G	WDFRU	005466
TOFT15	007002	T16FTB	026116	T26RTB	031266	T4MSG	022734	WEECK	020040
TOFT16	007006	T16MSG	025634	T26TBL	031256	T4RTB	023042	WEFRU	005516
TOFT17	007012	T16RTB	026112	T27	031300 G	T4TBL	023036	WRITE	010744
TOFT2	006742	T16TBL	026102	T27FTB	031462	T5	023050 G	WRITE1	011002
TOFT20	007016	T17	026234 G	T27MSG	031302	T5MSG	023052	WRTERR=	000002
TOFT22	007022	T17FTB	026422	T27RTB	031456	T5RTB	023160	WSCMD =	000004
TOFT23	007025	T17MSG	026236	T27TBL	031446	T5TBL	023154	XADBIT	002346 G
TOFT24	007030	T17RTB	026414	T28	031470 G	T6	023166 G	XCDENC	020470
TOFT25	007035	T17TBL	026404	T28FTB	031656	T6MSG	023170	XCEC	011606
TOFT4	006746	T18	026430 G	T28MSG	031472	T6RTB	023344	XCRBIT	014666
TOFT40	006644	T18FTB	026766	T28RTB	031652	T6TBL	023340	XCSRCK	014106
TOFT41	006646	T18MSG	026432	T28TBL	031642	T7	023352 G	XDVRCK	011660
TOFT42	006651	T18RTB	026762	T29	031664 G	T7MSG	023354	XEMPBF	010732
TOFT43	006654	T18TBL	026752	T29FTB	032072	T7RTB	023606	XENTCK	020236
TOFT5	006751	T19	027104 G	T29MSG	031666	T7TBL	023602	XERNBE	003470
TOFT7	006755	T19FTB	027332	T29RTB	032066	T8	023614 G	XERRCK	020162
TOMSG	022172	T19MSG	027106	T29TBL	032056	T8MSG	023616	XERROR	003334
TORT1	015026	T19RTB	027326	T3	022616 G	T8RTB	023776	XERUUT	002442
TORT2	015036	T19RT1	027306	T3MSG	022620	T8TBL	023772	XER1	016014
TORT3	015046	T19TBL	027316	T3RTB	022724	T9	024004 G	XER2	015531
TORT4	015056	T2	022456 G	T3TBL	022720	T9MSG	024006	XER3	015627
TORT5	015066	T2MSG	022460	T30	032100 G	T9RTB	024214	XESRCK	014234
TORT6	015076	T2RTB	022610	T30FTB	032372	T9TBL	024210	XFILBF	010614
T1	022332 G	T2TBL	022604	T30MSG	032102	UACDB	013660	XGSC	013244
T1MSG	022334	T20	027340 G	T30RTB	032366	UADCK	013500	XGTECN	015162
T1RTB	022450	T20FTB	027474	T30TBL	032356	UAM =	000200 G	XGTK	013076
T1TBL	022444	T20MSG	027342	T31	032400 G	UAUWCH	011750	XINIT	021512
T10	024222 G	T20RTB	027472	T31FTB	032744	UBRCR	013710	XINT	010460
T10MSG	024224	T20TBL	027462	T31MSG	032402	UBT10	024334	XPCE	007162
T10RTB	024356	T21	027502 G	T31RTB	032740	UBT21	027670	XPG	012464
T10TBL	024352	T21FTB	030120	T31TBL	032730	UBT23	030410	XPTDID	020640
T11	024364 G	T21MSG	027504	T32	032752 G	UBT24	030654	XPTFRU	005606
T11FTB	024614	T21RTB	030116	T32FTB	033350	UBT27	031432	XPTSTA	015436
T11MSG	024366	T21TBL	030106	T32MSG	032754	UBT31	032566	XRDERC	011454
T11RTB	024612	T22	030126 G	T32RTB	033346	UBT32	033324	XRDSTA	011336
T11TBL	024602	T22FTB	030240	T32TBL	033336	UBT36	034462	XREAD	011164
T12	024622 G	T22MSG	030130	T33	033356 G	UBT7	023570	XREGCK	014256
T12FTB	025024	T22RTB	030234	T33FTB	033550	UCGSC	013214	XSCP	007756
T12MSG	024624	T22TBL	030224	T33MSG	033360	UCT32	033240	XSDC	020540
T12RTB	025022	T23	030246 G	T33RTB	033544	UCT36	034410	XSETDN	011254
T12TBL	025012	T23FTB	030442	T33TBL	033534	UDUWCH	011744	XSRC	014520
T13	025032 G	T23MSG	030250	T34	033556 G	UNIT	021526	XTKCK	017722
T13FTB	025254	T23RTB	030440	T34FTB	033744	UNPKHP	021572	XTKPR1	010154
T13MSG	025034	T23TBL	030430	T34MSG	033560	UNTPRT	002512	XT1	022440
T13RTB	025252	T24	030450 G	T34RTB	033740	UN1	021314	XT10	024346
T13TBL	025242	T24FTB	030702	T34TBL	033730	VARIFY	002372	XT11	024576
T14	025262 G	T24MSG	030452	T35	033752 G	VECT	002354	XT12	025006

PARAMETER CODING
SYMBOL TABLE

MACRO M1110 13-AUG-79 15:33 PAGE 240-6

SEQ 0178

XT13	025236	XT20	027456	XT28	031636	XT35	034144	XUWCH	012014
XT14	025456	XT21	027774	XT29	032052	XT36	034500	XWAIT	011632
XT15	025604	XT22	030220	XT3	022714	XT4	023032	XWRITE	011054
XT16	026076	XT23	030424	XT30	032352	XT5	023150	XXPG	012540
XT17	026400	XT24	030664	XT31	032724	XT6	023334	X\$ALWA=	000000
XT18	026746	XT25	031050	XT32	033332	XT7	023576	X\$FALS=	000040
XT19	027302	XT26	031252	XT33	033530	XT8	023766	X\$OFFS=	000400
XT2	022600	XT27	031442	XT34	033724	XT9	024204	X\$TRUE=	000020

. ABS. 037670 000
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 34352 WORDS (135 PAGES)

DYNAMIC MEMORY: 20620 WORDS (79 PAGES)

ELAPSED TIME: 00:15:19

CZRIFA.BIN/DS:GBL/EN:ABS:AMA,CZRIFA/NL:BEX/-SP=LB1:[1,1]SVC/MLB,SY:[203,71]CZRIFA.MAC