

MAINDEC: -08-DIRXA-C-D
TITLE: -RX8/RX01 DIAGNOSTIC PROGRAM
AUTHOR: -MIKE STURAK
-HANK POULTER
-DON RICE
DATE: -JULY, 1976
MAINTAINER: -DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1975, 1976
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORTATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

1. ABSTRACT
2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 PRELIMINARY DIAGNOSTIC PROGRAMS
3. LOADING PRODEEDURE
4. STARTING PROCEDURE
 - 4.1 STARTING ADDRESS LOCATIONS
 - 4.2 OPERATOR ACTION
 - 4.3 PROGRAM ACTION
 - 4.4 TEST PARAMATER SELECTIONS
 - 4.4.1 RESTRICTIONS
5. OPERATING PROCEDURE
6. ERROR DETECTION
 - 6.1 PROGRAM DEFINITIONS OF:
 - 6.1.1 WRITE ERROR
 - 6.1.2 READ ERROR (CRC)
 - 6.1.3 CRC AND DATA ERROR
 - 6.1.4 NO CRC BUT DATA ERROR
 - 6.1.5 CRC BUT NO DATA ERROR
 - 6.1.5.1 SUMCHECK ERROR
 - 6.1.6 SEEK ERROR
 - 6.1.7 PARITY ERROR
 - 6.2 DIFINITIVE ERROR CODES
 - 6.3 UNEXPECTED OR MISSING ERROR CONDITIONS:
 - 6.3.1 MISSING D.D. MARK
 - 6.3.2 UNEXPECTED D.D. MARK
 - 6.3.3 MISSING ERROR FLAG
 - 6.3.4 UNEXPECTED RX01 IRQ
 - 6.3.5 DEVICE TEST HUNG
 - 6.4 SYSTEM FAILURES: (UNKNOWN IRQ)

7. ERROR REPORTING

7.1 EXAMPLES

7.1.1 EXAMPLES OF INTERFACE / CONTROL RELATED ERRORS

7.1.2 EXAMPLES OF DRIVE / DATA RELATED ERRORS

7.1.2.1 SEEK ERROR

7.1.2.2 WRITE-CRC AND DATA ERROR

7.1.2.3 READ-CRC AND DATA ERROR

7.1.2.4 WRITE-CRC BUT NO DATA ERROR

7.2 ERROR RECOVERY

7.3 FATAL ERROR RECOVERY

8. MANUAL INTERVENTION

8.1 FIELD ENGINEERING TROUBLE SHOOTING AIDS

9. DIAGNOSTIC TEST STRATEGY

10. CONSOLE PACKAGE ADDENDUM

11. APT-8 INTERFACES

12. PROGRAM LISTING

1. ABSTRACT

//////////

THIS PROGRAM IS DESIGNED TO TEST THE RX01 FLEXIBLE DISKETTE SUBSYSTEM CONFIGURED AS A COMPLETE SUBSYSTEM, (RX8 INTERFACE, RX01 CONTROL, AND DISKETTE DRIVES), OR AS A PARTIAL SUBSYSTEM (INTERFACE, AND/OR CONTROL). OBVIOUSLY THE MAXIMUM RESOLUTION POSSIBLE IS ACHIEVED BY TESTING THE RX01 DISKETTE SUBSYSTEM CONFIGURED AS A COMPLETE SUBSYSTEM.

REVISION C

THE ADDITION OF THE CONSOLE PACKAGE.
THE REMOVAL OF TEST THREE (3).
SEE LISTING LOCATION FOR EXPLANATION
THE REMOVAL OF DECIMAL CONVERSION ROUTINES.
THE ADDITION OF APT-8 INTERFACES.

2. REQUIREMENTS

//////////

2.1 EQUIPMENT

//////////

A PDP-8/E TYPE GENERATION COMPUTER WITH 4K OF CORE, CONSOLE SWITCHES, TELETYPE, AND AN RX01 DISKETTE SUBSYSTEM.
THE DIAGNOSTIC CAN RUN WITHOUT HARDWARE SWITCH IF THE CONSOLE PACKAGE IS USED. IF THE CONSOLE PACKAGE IS ACTIVE 8K OF CORE IS NEEDED.

2.2 STORAGE

//////////

THIS PROGRAM IS DESIGNED TO BE RUN STAND ALONE AND MUST OCCUPY PROGRAM LOCATIONS 0 THRU 7577 OF FIELD 0.
THE CONSOLE PACKAGE OCCUPIES FIELD 1 LOCATIONS 200 TO 2000.
FIELD 1 IS NEEDED ONLY IF THE CONSOLE PACKAGE IS BEING USED.

2.3 PRELIMINARY DIAGNOSTIC PROGRAMS

////////////////////////////////////

THIS PROGRAM ASSUMES THAT THE HOST PDP IS FUNCTIONING CORRECTLY.

3. LOADING PROCEDURE
////////////////////

THIS PROGRAM IS IN BINARY FORMAT. TO LOAD THIS PROGRAM INTO CORE,
FOLLOW THE INSTRUCTIONS PUBLISHED FOR THE PARTICULAR BINARY FORMAT
LOADER BEING USED.

4. STARTING PROCEDURE
////////////////////

4.1 STARTING ADDRESS LOCATIONS
////////////////////

THIS PROGRAM HAS 2 STARTING ADDRESS LOCATIONS:

200 * DIAGNOSTIC PROGRAM *

STARTING THIS PROGRAM AT PROGRAM LOCATION 200, AND SUCCESSFULLY SUPPLYING
THE NECESSARY TEST PARAMETERS REQUESTED, DIRECTS THIS PROGRAM TO BEGIN
TESTING THE RX01 SUBSYSTEM AS A SUBSYSTEM OR PARTIAL SUBSYSTEM.

THE ADDRESS THAT THE CONSOLE PACKAGE WILL START THE DIAGNOSTIC
AT IS LOCATION 200.

201 * RESTART OF DIAGNOSTIC PROGRAM *

STARTING THIS PROGRAM AT PROGRAM LOCATION 201 DIRECTS THE PROGRAM TO
CONTINUE TESTING THE RX01 SUBSYSTEM AS A SUBSYSTEM OR PARTIAL SUBSYSTEM
USING THE PARAMETERS SUPPLIED AT STARTING ADDRESS 200.

4.2 OPERATOR ACTION
////////////////////

CONFIGURE THE ACCUMULATOR SWITCHES TO REPRESENT THE OCTAL PROGRAM STARTING
ADDRESS LOCATION DESIRED, PRESS " LOAD ADDRESS ", PRESS " CLEAR ", AND
FINALLY PRESS " CONT " INUE. IF THE CONSOLE PACKAGE IS ACTIVE
THEN ISSUEING THE R DIRXA COMMAND WILL LOAD THE PROGRAM FROM
A DISKETTE.

4.3 PROGRAM ACTION
////////////////////

THE PROGRAM HAS PRINTED ON THE TELEPRINTER THE PRESENT " MAINDEC "
REVISION, AND, IF THE PROGRAM STARTING ADDRESS PROGRAM LOCATION IS:

* 200 *

AND THE CONSOLE PACKAGE IS ACTIVE THE DIAGNOSTIC WILL PRINT
"REMOVE DIAGNOSTIC DISKETTE" AND WAIT FOR A *** CONTROL E ***
THE PROGRAM WILL CONTINUE RUNNING AFTER THE OPERATOR REMOVES THE
DIAGNOSTIC DISKETTE AND REPLACES IT WITH A SCRATCH DISKETTE.
TO CONTINUE TYPE CONTROL E.

THE PROGRAM IS REQUESTING TEST PARAMATER INFORMATION TO BE SUPPLIED BY
BY THE OPERATOR VIA THE PDP SWITCHES.

* 201 *

THE PROGRAM HAS RECONFIRMED PREVIOUS TEST PARAMATER SELECTIONS BY PRINTING
APPROPRIATE MESSAGES ON THE TELEPRINTER, AND HAS RESUMED TESTING.

4.4 TEST PARAMETER SELECTIONS
////////////////////

AFTER STARTING THIS PROGRAM AT STARTING ADDRESS 200, THE PROGRAM WILL PRINT
" SELECT PARAMATERS (INCLUDING DEVICE CODE) " AND HALT OR PRINT
THE SWITCH REGISTER QUESTION IF THE CONSOLE PACKAGE IS ACTIVE.

THE OPERATOR WILL THEN CONFIGURE CONSOLE SWITCHES 0, AND 1 TO REPRESENT
DISKETTE DRIVE SELECTION, CONSOLE SWITCH 2 TO REPRESENT RX01 MICROCONTROLLER
CABLE CONNECTION TO RX8 INTERFACE, CONSOLE SEITCHES 3, 4, AND 5 TO REPRESENT
RX8 INTERFACE IOT DEVICE CODE, AND FINALLY CONSOLE SWITHCHES 7 THRU 11 TO
REPRESENT THE [STARTING] TEST , AND THEN PRESS " CONT " INUE.(SEE RESTRICTIONS)

C O N S O L E
S W I T C H E S

0 1 2 3 4 5 6 7 8 9 10 11
U U C X X X - T T T T T
(STARTING TEST)

- 0 = 0 - DISKETTE DRIVE(S) IS READY
- 1 = 0 - (POWER APPLIED / DOOR CLOSED)
- 2 = 0 - RX01 MICROCONTROL CABLED TO RX8 INTERFACE
- 3 = - FOR STANDARD DEVICE CODE
- 4 = - POSITION CONSOLE SWITCHES
- 5 = - 3-4-5 = 0
- 7 = - TO EXECUTE ALL TESTS
- 8 = - CONFIGURE CONSOLE SWITCHES
- 9 = - 7 THRU 11
- 10 = - EQUIVALENT TO ZERO
- 11 = - (THIS IS THE STARTING TEST)

THE PROGRAM WILL PRINT A CONFIRMATION MESSAGE, THEN BEGIN TESTING THE RX01 SUBSYSTEM AS A SUBSYSTEM OR PARTIAL SUBSYSTEM WITH THE FIRST TEST CONFIGURED WITHIN CONSOLE SWITCHES 7 THRU 11.

TO TEST THE RX01 DISKETTE SUBSYSTEM AS A COMPLETE SUBSYSTEM, CONFIGURE CONSOLE SWITCHES 0, 1, AND 2 TO REPRESENT ZERO. THE PROGRAM WILL ASSUME ALL DRIVE(S) ARE READY (POWER APPLIED / DOOR CLOSED).

NOTE, HOWEVER, THAT TESTING THE RX01 DISKETTE SUBSYSTEM AS A PARTIAL SUBSYSTEM IS MORE THAN MERELY CONFIGURING CONSOLE SWITCHES 0, 1, AND 2 TO REPRESENT VALUES OTHER THAN ZERO.

IF ONLY THE CONTROL / INTERFACE PARTIAL SUBSYSTEM (NO DRIVES) IS THE THE UNIT UNDER TEST (UUT), THEN THE DISKETTE DRIVE(S) CABLE (IF ANY) MUST BE PHYSICALLY REMOVED FROM THE RX01 CONTROLLER, OR, THE DRIVES MUST BE RENDERED " NOT READY " (DOOR OPEN), THEN THE CONSOLE SWITCHES 0, AND 1 EACH MUST BE CONFIGURED TO REPRESENT A " 1 ".

IF ONLY THE INTERFACE PARTIAL SUBSYSTEM (NO DRIVES AND CONTROL) IS THE UUT THEN THE BC05-L CABLE FROM THE INTERFACE TO THE CONTROL MUST BE PHYSICALLY

MOVED, ONLY THEN MAY CONSOLE SEITCHES 0, 1, AND 2 BE CONFIGURED TO REPRESENT A " 1 " .

4.4.1 RESTRICTIONS
////////////////////

A SELECT GROUP OF TESTS ARE DESIGNED TO CONFIRM THE RESULTS OF PREVIOUS TESTS. THE FOLLOWING IS AN OVERVIEW OF THOSE TESTS WHICH REQUIRE A PREVIOUS TESTS TO HAVE EXECUTED CORRECTLY.

...FOR THIS TEST TO FUNCTION CORRECTLY...	/	...THIS TEST MUST HAVE PREVIOUSLY EXECUTED CORRECTLY...
---	---	---

T14, T15 / T13

TEST 14, AND TEST 15 EMPTY THE SECTOR BUFFER IN 12-BIT MODE COMPARING THE CONTENTS TO THE EXPECTED DATA PATTERN FILLED BY TEST 13.

T17, T20 / T16

TEST 17, AND TEST 20 EMPTY THE SECTOR BUFFER IN 8-BIT MODE COMPARING THE CONTENTS TO THE EXPECTED DATA PATTERN FILLED WITHIN TEST 16.

T24, T25, T26 / T22

THE SECTOR BUFFER SHOULD REMAIN UNCHANGED AND CONTAIN THE DATA PATTERN FILLED WITHIN TEST 22 AFTER THE RX01 CONTROL ATTEMPTS TO SEEK SECTOR 0.

T27 / T22

THE CONTENTS OF THE SECTOR BUFFER SHOULD REMAIN UNCHANGED AND CONTAIN THE DATA PATTERN FILLED WITHIN TEST 22 AFTER THE RX01 CONTROLLER ATTEMPTS TO SEEK AN ILLEGAL TRACK GREATER THAN 114.

T32 / T31

TEST 32 ISSUES AN INITIALIZE WHICH PERFORMS AN " IMPLIED READ " OF TRACK 1 / SECTOR 1. THE PROGRAM COMPARES THE DATA FROM THE SECTOR BUFFER WHICH SHOULD BE EQUIVALENT TO THE DATA WRITTEN WITHIN TEST 31 (FIRST WRITE EVER).

TEST 33 ISSUES THE FIRST READ EVER OF THE TRACK / SECTOR WRITTEN WITHIN
TEST 31 AND COMPARES THE DATA EXPECTING THE DATA PATTERN OF TEST 31.

5. OPERATING PROCEDURE
////////////////////////////////////

THE OPERATOR MAY CONTROL THE DYNAMIC ACTION OF THE PROGRAM BY APPROPRI-
ATELY CONFIGURING THE ACCUMULATOR SWITCHES AS DESCRIBED BELOW.
IF THE CONSOLE PACKAGE IS ACTIVE THEN THE SETTING OF THE
SWITCH REGISTER UNDER DYNAMIC CONDITIONS IS POSSIBLE BY
TYPING CONTROL D THIS WILL PRINT THE SWITCH REGISTER
QUESTION AND ALLOW CHANGES TO IT THEN CONTINUE RUNNING.

0 = 1 - HALT AT DETECTION OF AN ERROR
1 = 1 - HALT AT THE END OF A TEST PASS
2 = 1 - (LONG) DATA COMPARISON ERROR PRINTOUT
3 = 1 - INHIBIT ERROR PRINTOUTS
4 = 1 - LOCK SCOPE LOOP ON ERROR
5 = 1 - LOCK SCOPE LOOP ON TEST (OK OR NOT)
6 = 1 - INHIBIT THE ISSUING OF [INIT] AT ERROR
11= 1 - DISABLE THE RINGING OF THE BELL AT AN ERROR

6. ERROR DETECTION

6.1 PROGRAM DEFINITIONS

THIS PROGRAM HAS DEFINED THE FOLLOWING AS ERRORS:

6.1.1 WRITE ERROR
////////////////////////////////////

A WRITE ERROR IS A READ ERROR IF THE DATA BEING READ IS OF UNKNOWN
QUALITY (THE DATA BEING READ IS BEING READ FOR THE FIRST TIME AFTER ITS
WRITING) .

6.1.2 READ (CRC) ERROR - (TRANSFER REGISTER STATUS BIT 11 = 1)
////////////////////////////////////

A READ ERROR IS A READ ERROR WHERE THE QUALITY OF THE DATA BEING READ
IS KNOWN (THE DATA BEING READ HAD BEEN READ SUCCESSFULLY SOME TIME PRE-
VIOUSLY).

6.1.3 CRC AND DATA ERROR

////////////////////

6.1.4 NO CRC BUT DATA ERROR

////////////////////

6.1.5 CRC BUT NO DATA ERROR

////////////////////

THESE DATA ERRORS ARE DETECTED WHEN THE PROGRAM IS VERIFYING THE DATA THAT " SHOULD HAVE BEEN READ " WITH THE DATA THAT " ACTUALLY WAS READ " BY COMPARING THE " BAD " COLUMN TO THE " GOOD " COLUMN.

WORD# GOOD BAD

1 (TRACK IDENTIFICATION BITS 5-11)
2 (SECTOR IDENTIFICATION BITS 8-11)

WORDS 3 THRU 62 (IF 12-BIT MODE), OR
BYTES 3 THRU 126 (IF 8-BIT MODE) CONTAIN
THE OPERATORS PPP SELECTION.

63 (OR BYTE 127) - THE SUM OF ALL WORDS 1 THRU 62
OR BYTES 1 THRU 127.

64 OR (BYTE 128) - THE NEGATIVE OF 2 TIMES
THE VALUE OF WORD # 63 OR BYTE # 127.

6.1.5.1 SUMCHECK ERROR

////////////////////

THE PROGRAM DETECTS A " SUM-CHECK " ERROR BY SUMMING ALL THE ACTUAL (BAD) DATA COLUMN AND COMPARING THAT SUM TO 0.

THE REASON FOR THE FIRST 2 WORDS/BYTES CONTAINING TRACK / SECTOR IDENTIFICATION CODES IS TO DETECT ADDRESSING ERRORS.

THE REASON FOR THE LAST 2 WORDS/BYTES CONTAINING CHECKSUM INFORMATION IS TO DISTINGUISH BETWEEN WHAT MIGHT RESEMBLE AN ADDRESSING ERROR (IF THE PROGRAM DETECTED AN ERROR WHEN COMPARING THE FIRST 2 WORDS/BYTES) AND A CRC ERROR.

6.1.6 SEEK ERROR - (NO ALLOCATED TRANSFER REGISTER STATUS BIT)
////////////////

A SEEK ERROR HAS BEEN DEFINED AS " NOT A CRC ", AND " NOT A PARITY " ERROR.

6.1.7 PARITY ERROR - (TRANSFER REGISTER STATUS BIT 10 = 1)
////////////////

A PARITY ERROR IS AN ERROR WHICH RESULTS FROM AN INCORRECT TRANSFER OF THE COMMAND WORD FROM THE RX8 INTERFACE TO THE RX01 MICROPROCESSOR CONTROL.

6.2 DEFINITIVE ERROR CODES
////////////////

THE RX01 MICROCONTROLLER HAS DEFINED ERROR CODES AND MEANINGS WHICH ARE AVAILABLE TO THE PROGRAM BY ISSUING COMMAND #7 TO " READ THE B-CODE ".

A DEFINITIVE ERROR CODE REPRESENTS [WHERE] WITHIN A MICRO-FUNCTION THE ERROR WAS DETECTED.

THE FOLLOWING ARE THE DEFINITIVE ERROR CODES AND MEANINGS:

0	-	NO ERROR
10	-	DRIVE 0 FAILED TO SEE HOME FROM INITIALIZE
20	-	DRIVE 1 FAILED TO SEE HOME FROM INITIZLIAE
30	-	HOME FOUND WHEN STEPPING OUT 10 TRACKS FROM INIT
40	-	TRIED TO ACCESS A TRACK GREATER THAN 77(DECIMAL)
50	-	HOME WAS FOUND BEFORE DESIRED TRACK
60	-	SELF DIAGNOSTIC ERROR
70	-	DESIRED SECTOR NOT FOUND AFTER SAMPLING 52 HEADERS
100	-	WRITE PROTECT ERROR
110	-	MORE THAN 40US AND NO SEP CLOCK DETECTED
120	-	A PREAMBLE COULD NOT BE FOUND
130	-	PREAMBLE FOUND BUT NO ID MARD FOUND IN TIME
140	-	CRC ERROR ON SUPPOSIDLY GOOD HEADER
150	-	GOOD HEADER(NO CRC ERROR) BUT TRACK COMPARE ERROR
160	-	IDAM NOT FOUND IN TOME
170	-	DATA AM NOT FOUND IN TIME
200	-	DATA CRC ERROR
210	-	ALL PARITY ERRORS

6.3 UNEXPECTED OR MISSING ERROR CONDITIONS

////////////////////////////////////

6.3.1 MISSING DD MARK

////////////////////////////////////

THIS ERROR MAY OCCUR WHEN THE PROGRAM EXPECTED A DELETED DATA MARK BUT NONE OCCURED.

6.3.2 UNEXPECTED DD MARK

////////////////////////////////////

THIS ERROR MAY OCCUR WHEN THE PROGRAM HAD NOT EXPECTED A DELETED DATA MARK BUT ONE OCCURED.

6.3.3 MISSING ERROR FLAG

////////////////////////////////////

THIS ERROR MAY OCCUR WHEN THE CONTENTS OF THE TRANSFER REGISTER AT DONE TIME ARE NOT 0, AND THE ERROR FLAG IS CLEARED.

6.3.4 UNEXPECTED RX01 IRQ

////////////////////////////////////

THIS ERROR MAY OCCUR WHEN THE PROGRAM HAS NOT YET ENABLED THE RX8 INTERRUPT ENABLE FLIP-FLOP BUT AN INTERRUPT OCCURED.

6.3.5 DEVICE TEST HUNG

////////////////////////////////////

THIS ERROR MAY OCCUR WHEN THE PROGRAM EXPECTS BUT FAILED TO RECIEVE A PROGRAM INTERRUPT REQUEST FROM THE RX01 SUBSYSTEM WITHIN AN ALLOTTED PERIOD OF TIME (APPROXIMATELY 4 SECONDS).

6.4 UNKNOWN IRQ

////////////////////////////////////

THIS ERROR MAY OCCUR WHEN THE PROGRAM HAS FAILED TO IDENTIFY THE DEVICE ISSUING A PROGRAM INTERRUPT REQUEST.

7. ERROR REPORTING

////////////////////

ALL ERRORS DETECTED WILL BE REPORTED IF AC SW 3 = 0. THE PROGRAM HAS TWO ERROR REPORTING SCHEMES.

7.1.1 EXAMPLES IF INTERFACE / CONTROL RELATED ERRORS

THE FOLLOWING INFORMATION IS PRINTED FOR ALL INTERFACE / CONTROL RELATED ERRORS.

ERR	FAT	FAST	-	EAC	GOOD	PASS
ERR			-	PROGRAM ADDRESS OF THE EPROP		
FAT			-	FIRST ADDRESS OF THE TEST IN ERROR		
FAST			-	FIRST ADDRESS OF THE SUBTEST WITHIN THE TEST		
-			-	SEE ASSEMBLY LISTING FOR MAP		
EAC			-	ERROR AC (ACTUAL) RESULT OF TEST		
GOOD			-	EXPECTED RESULT OF TEST		
PASS			-	PASS # AT ERROR		

7.1.2 EXAMPLES OF DRIVE / DATA RELATED ERRORS

THE FOLLOWING INFORMATION IS PRINTED FOR ALL DRIVE / DATA RELATED ERRORS.

CMND	XDR	CODE	RSTA	START	TARGET	PASS
CMND			-	COMMAND TO THE RX01 MICROCONTROLLER		
XDR			-	CONTENTS OF THE TRANSFER REGISTER AT ERROR/DONE		
CODE			-	DEFINITIVE ERROR CODE (VIA COMMAND #7)		
RSTA			-	STATUS (VIA COMMAND #5)		
START			-	STARTING TRACK/SECTOR ACTUATOR POSITION		
TARGET			-	TARGET TRACK/SECTOR ACTUATOR POSITION		
PASS			-	PASS # AT ERROR (TO 16777215 DECIMAL)		

7.1.2.1 SEEK ERROR

CMND	XDR	CODE	RSTA	START	TARGET	PASS
0014	0100	0120	0300	[HOME]	1 1	
INIT	0	0120	0200	[HOME]	[HOME]	

A SEEK ERROR OCCURED WHILE TRYING TO " WRITE DELETED DATA " (CMND #14)

ONTO TRACK 1 SECTOR 1.

COMD	-	WRITE DELETED DATA
XDR	-	DELETED DATA MARK
CODE	-	A PREAMBLE COULD NOT BE FOUND
RSTA	-	DRIVE READY + DELETED DATA
START	-	HOME POSITION
TARGET	-	TRACK 1, SECTOR 1

THEN THE PROGRAM ISSUED AN INITIALIZE AT AN ATTEMPT TO RECOVER FROM
THE " SEEK " ERROR.

CMND - MEANS IOT 67X7 [INIT] WAS ISSUED
XDR - MUST BE SEEK (NOT CRC OR PARITY)
CODE - A PREAMBLE COULD NOT BE FOUND
RSTA - DRIVE READY
START - HOME POSITION
TARGET - HOME POSITION

7.1.2.2 WRITE-CRC AND DATA ERROR

CMND	XDR	CODE	RSTA	START	TARGET	PASS
0026	0001	3200	0201	100,30	100,1	
WRITE-CRC AND DATA ERROR						
WORD	GOOD	BAD				
4	5435	5473				
5	6617	5437				
6	6303	4606				
SUMCHECK IS 1253						
TOTAL BAD=60						

WHILE READING SECTOR 7 OF TRACK 100 THE PROGRAM DETECTED A CRC ERROR.

THE PROGRAM EXPANDS THE STANDARD ERROR FORMAT TO INCLUDE DATA COM-
PARISON INFORMATION IF THE TEST IS A DATA COMPARISON TEST.

IF AC SWITCH 2 =1 THEN A [LONG] DATA COMPARISON PRINTOUT WOULD
HAVE OCCURRED OF [ALL] THE WORDS/BYTES IN ERROR.

THE WORD " WRITE " WITHIN THE EXPANSION MEANS THAT THE DATA OF
SECTOR 7 HAD NEVER BEEN READ BEFORE, THEREFORE THE PROGRAM ASSUMED IT
WAS WRITTEN INCORRECTLY.

7.1.2.3 READ-CRC AND DATA ERROR

CMND	XDR	CODE	RSTA	START	TARGET	PASS
0026	0001	0200	0201	100,30	100,1	
READ-CRC AND DATA ERROR						
WORD	GOOD	BAD				
4	5435	5477				
5	6617	5437				
6	6303	5406				
SUM-CHECK IN 1257						
TOTAL BAD=60						

WHILE SEEKING SECTOR 1 OF TRACK 100 THE PROGRAM DETECTED A CRC ERROR.
THE PROGRAM EXPANDS THE STANDARD ERROR FORMAT TO INCLUDE DATA COMPARISON
INFORMATION IF THE TEST IS A DATA COMPARISON TEST.

IF AC SWITCH 2 = 1 THEN A [LONG] DATA COMPARISON ERROR PRINTOUT WOULD
HAVE OCCURRED OF [ALL] THE WORDS/BYTES IN ERROR.

THE WORD " READ " WITHIN THE EXPANSION MEANS THAT THE DATA OF
SECTOR 7 HAD BEEN READ SOME TIME PREVIOUSLY, THEREFORE THE PROGRAM
ASSUMES THAT THE DATA WAS WRITTEN CORRECTLY BUT READ INCORRECTLY.

7.1.2.4 WRITE-CRC BUT NO DATA ERROR

CMND	XDR	CODE	RSTA	START	TARGET	PASS
0026	0001	0200	0201	100,30	100,1	

WRITE - CRC BUT NO DATA ERROR

A CRC ERROR WAS DETECTED AFTER READING SECTOR 1 OF TRACK 100.

THE WORD " WRITE " WITHIN THE DATA EXPANSION MEANS THAT THE DATA
OF SECTOR 1 TRACK 100 HAD NEVER BEEN READ BEFORE THEREFORE THE PROGRAM
ASSUMED IT WAS WRITTEN INCORRECTLY, AND BECAUSE THE PROGRAM DID NOT DETECT
A DATA COMPARISON ERROR, IT ASSUMED THAT THE 2 CRC CHARACTERS WERE WRITTEN
INCORRECTLY.

7.2 ERROR RECOVERY ////////////////////

THE PROGRAM WILL ATTEMPT TO RETRY ALL ERRORING FUNCTIONS 10 TIMES.
A RECOVERABLE ERROR (SOFT) IS ONE WHICH DISAPPEARS WITHIN 10 PROGRAM
RETRYS. AN UNRECOVERABLE ERROR (HARD) IS ONE WHICH REMAINS AFTER 10
PROGRAM RETRYS.

THE PROGRAM WILL ISSUE IOT 67X7 [INIT] FOR ALL ERRORS BUT
DEFINITIVE ERROR CODES 140, 200, AND 210.

140 - CRC ERROR ON SUPPOSIDELY GOOD HEADER
200 - DATA CRC ERROR
210 - PARITY ERROR

THESE ERRORS ARE [NOT] SEEK TYPE ACTUATOR RELATED ERRORS.
IF A HARD WRITE ERROR IS DETECTED THE PROGRAM ABORTS FURTHER TESTING
THIS PASS OF THAT TRACK/SECTOR BUT CONTINUES TESTING TRACKS. ON THE REMAINING
A SYMPATHETIC HARD READ ERROR MAY OCCUR (PATTERN DEPENDENT) IF THE
HARD WRITE ERROR HAD OCCURRED WITHIN A TEST WHICH WOULD EVENTUALLY READ
THAT SECTOR AND A SYMPATHETIC HARD DATA COMPARISON ERROR MAY OCCUR IF THAT
TEST WAS TO VERIFY THE DATA TO A KNOWN PATTERN.

7.3 FATAL ERROR RECOVERY ////////////////////////////////////

IF THE PROGRAM DETECTS ANY OF THE SUCCEEDING FATAL ERROR CONDITIONS,
THE PROGRAMS RECOVERY WILL BE THAT OF EXITING THE PRESENT TEST.

- (A) HARD PARITY ERROR
- (B) A SELECTED DRIVE BECOMING NOT READY
- (C) NO EXPECTED RX01 INTERRUPT REQUEST
- (D) MISSING ERROR FLAG
- (E) LOG OVERFLOW
- (F) DEVICE TEST HUNG.

8. MANUAL INTERVENTION ////////////////////////////////////

8.1 FIELD ENGINEERING TROUBLE SHOOTING AIDS ////////////////////////////////////

THE FIELD ENGINEER, BY ALTERING THE CONTENTS OF SPECIFIC PROGRAM
MAINTENANCE LOCATIONS, IS ABLE TO DIRECT THE PROGRAM TO PERFORM TESTING
UPON A PARTICULAR AREA [WINDOW] OF THE DISKETTE INSTEAD OF THE ENTIRE
SURFACE. THESE PROGRAM LOCATIONS ARE LABELED "OD" "ID", "FIRST, AND "LAST".

"OD" (OUTSIDE DIAMETER), PROGRAM LOCATION 30, IS THE INITIAL OUTER-
MOST TRACK THE PROGRAM WILL ACCESS.

"ID" (INSIDE DIAMETER), PROGRAM LOCATION 31, IS THE FINAL INNERMOST
INNERMOST TRACK THE PROGRAM WILL ACCESS.

"FIRST", PROGRAM LOCATION 32, IS THE FIRST SECTOR TO BE ACCESSED
OF A TRACK.

"LAST", PROGRAM LOCATION 33, IS THE LAST SECTOR TO BE ACCESSED OF A TRACK.

THE STANDARD ASSEMBLED CONTENTS OF THESE FIELD ENGINEERING MAINTENANCE LOCATIONS ARE:

*30

OD,	52	/INITIAL TRACK TO TEST
ID,	53	/FINAL TRACK TO TEST
FIRST,	1	/FIRST SECTOR OF A TRACK
LAST,	32	/LAST SECTOR OF A TRACK

THESE ARE THE ONLY FIELD ENGINEERING MAINTENANCE PROGRAM LOCATIONS DESIGNED TO BE EXTERNALLY ALTERED.

THE PROGRAM WILL PRINT A MAINTENANCE VERIFICATION MESSAGE IF THE CONTENTS OF THE MAINTENANCE LOCATIONS ARE NOT THE ASSEMBLED STANDARDS.

THE OCTAL CONTENTS OF THESE MAINTENANCE LOCATIONS MUST BE WITHIN THE RESTRICTED LIMITS WHICH ARE:

0 <= OD <= ID
0 <= ID <= 114
1 <= FIRST <= LAST
1 <= LAST <= 32

THE PROGRAM VERIFIES THE CONTENTS OF EACH MAINTENANCE LOCATION. THE PROGRAM WILL SET INTO THE MAINTENANCE LOCATION THE STANDARD VALUE IF THE DESIRED CONTENTS WERE NOT WITHIN THE REQUIRED SPECIFIED LIMITS.

NOTE, THAT TRACK 0 IS NOT INCLUDED WITHIN THE STANDARD [WINDOW] OF TESTABLE TRACKS. THE REASON FOR THIS IS NOT TO INADVERTENTLY DESTROY THE FORMATTED CONTENTS OF TRACK 0.

TO EXPAND THE WINDOW OF TESTABLE TRACKS INCLUDING TRACK 0, THE CONTENTS OF PROGRAM LOCATIONS " OD " MUST BE ZERO.

IN SUMMARY, IF THE CONTENTS OF PROGRAM LOCATIONS OD, ID, FIRST, AND LAST WERE 30, 30, 1, 1, RESPECTIVELY, THE PROGRAM WOULD PERFORM SELECTED TESTING ONLY UPON TRACK 30, SECTOR 1.

9. DIAGNOSTIC TEST STRATEGY
////////////////////////////////////

THE PDP-8 DIAGNOSTIC PACKAGE, WHICH CONTAINS THE RX8/RX01 DIAGNOSTIC PROGRAM, MAINDEC-08-DIRXA-, AND THE RX8/RX01 DATA RELIABILITY/EXERCISER PROGRAM, MAINDEC-08-DIRXB-, IS DESIGNED TO COMPLETELY TEST THE RX8/RX01 SUBSYSTEM WHEN EACH PROGRAM IS RUN TO ITS NATURAL COMPLETION COMMENCING WITH THE RX8/RX01 DIAGNOSTIC PROGRAM.

THE RX8 DIAGNOSTIC PROGRAM CONTAINS EFFICIENT MAINTENANCE SCOPING LOOPS, WHILE THE RX8/RX01 DATA RELIABILITY/EXERCISER PROGRAM CONTAINS DISKETTE COMPATABILITY VERIFICATION.

10. CONSOLE PACKAGE
//////////

TABLE OF CONTENTS

- 10.1. ABSTRACT
- 10.2. REQUIREMENTS
- 10.3. RESTRICTIONS
- 10.4. STANDARD OPERATING PROCEDURE
 - 10.4.1 INITIALIZATION
 - 10.4.2 CONTROL CHARACTERS
 - 10.4.3 WAITING MESSAGE
 - 10.4.4 SWITCH REGISTER MESSAGE
 - 10.4.5 END OF PASS MESSAGE
- 10.5. ERRORS
 - 10.5.1 ERROR HALTS
- 10.6. SWITCH REGISTER SETTINGS
 - 10.6.1 OPERATING SWITCHES
 - 10.6.2 ERROR SWITCHES
- 10.7. LOCATION CHANGES
- 10.8. PROGRAM DESCRIPTION
- 10.9. DIALOGUE FOR CONSOLE PACKAGE
- 10.10. LISTING

10.1. ABSTRACT

THE CONSOLE PACKAGE HAS BEEN ADDED TO THIS DIAGNOSTIC TO ALLOW THE PROGRAM TO RUN WITH NO HARDWARE SWITCH REGISTER AND TO HAVE COMMUNICATIONS WITH THE DIAGNOSTIC VIA A TERMINAL.

THE DIAGNOSTIC CAN BE RUN IN TWO MODES WITH THE CONSOLE PACKAGE . 1) RUNNING WITH THE CONSOLE PACKAGE ACTIVE - THIS ALLOWS THE OPERATOR CONTROL OF THE DIAGNOSTIC THROUGH THE TERMINAL, THE DIAGNOSTIC WILL ASK FOR THE VALUE OF THE PSEUDO SWITCH REGISTER, BEFORE CONTINUING WITH EXECUTION OF THE DIAGNOSTIC ALL ERROR WILL BE PRINTED ON THE TERMINAL AND THE NUMBER OF PASSES WILL BE PRINTED. THERE WILL BE NO HALTS EXECUTED .

2) CONSOLE PACKAGE NOT ACTIVE-THIS WILL RESULT IN THE USE OF HALTS FOR ERROR, HALTS AT END OF PASS IF SELECTED, USE OF THE HARDWARE SWITCH REGISTER , NOT ASKING THE SWITCH QUESTION.

10.2. REQUIREMENTS

10.3. RESTRICTIONS

- 1) RUNNING THE CONSOLE PACKAGE REQUIRES THAT THE PSEUDO SWITCH REGISTER BE USED.
- 2) ONCE RUNNING THE CONSOLE PACKAGE NONACTIVE AND NOW DESIRE TO RUN IT ACTIVE. ONE MUST RELOAD THE DIAGNOSTIC AND INITILIZE FOR A ACTIVE CONSOLE PACKAGE.
- 3.) THE CONSOLE PACKAGE IS LOCATED IN FIELD 1 THERE MUST BE 8K OF CORE TO RUN THE CONSOLE PACKAGE.

10.4. STANDARD OPERATION PROCEDURE

10.4.1 INITIALIALIZATION

FOR A ACTIVE CONSOLE PACKAGE

- 1.) LOAD ADDRESS 0021
- 2.) SET TO 0000 INDICATOR FOR USING THE PSEUDO SWITCH REGISTER
- 3.) LOAD ADDRESS 0022

- 4.) SET SR3=1 (400) INDICATOR FOR USING A ACTIVE CONSOLE PACKAGE
- 5.) LOAD STARTING ADDRESS OF PROGRAM AND BEGIN

FOR A NON ACTIVE CONSOLE PACKAGE

- 1.) LOAD ADDRESS 0021
- 2.) SET SR0=1 (4000) TO INDICATE A HARDWARE SWITCH REGISTER
- 3.) LOAD ADDRESS 0022
- 4.) SET TO 0000 TO INDICATE A DEACTIVE CONSOLE PACKAGE
- 5.) LOAD STARTING ADDRESS OF PROGRAM AND BEGIN

10.4.2 CONTROL CHARACTERS -----

CONTROL CHARACTERS ARE USED TO GIVE THE OPERATOR THE ABILITY TO PERFORM THE FOLLOWING FUNCTIONS.
NOTE: THE PROGRAM WILL RESPOND TO THE CONTROL CHARACTER IN FIVE (5) SECONDS OR LESS.

CONTROL C -----

THIS WILL START THE LOADER THAT IS IN LOCATION 7600.

CONTROL R -----

THIS WILL RESTART THE PROGRAM AND REASK THE SWITCH REGISTER QUESTION.

CONTROL E -----

THIS WILL CONTINUE THE PROGRAM FROM A ERROR IF ALLOWED BY THE DIAGNOSTIC OR FROM A WAITING STATEMENT.

CONTROL S -----

THIS WILL STOP PROGRAM EXECTUION AND WAIT FOR A CONTROL Q BEFORE PRINTING . THIS IS A NONPRINTING CHARACTER.

CONTROL Q -----

THIS IS TO CONTINUE A PROGRAM AFTER A CONTROL Q IS TYPED. THIS IS A NONPRINTING CHARACTER.

10.4.3 WAITING MESSAGE

THE WAITING MESSAGE IS USED TO ALLOW THE OPERATOR TIME TO MAKE A DECISION AS TO WHAT CONTROL CHARACTER TO TYPE. THIS MESSAGE MAY APPEAR AT THE END OF PASS MESSAGE IF THE HALT ON ERROR BIT IS SET. THE CONTROL CHARACTERS MAY NOW BE USED TO PERFORM THE NEEDED FUNCTION. THE WAITING MESSAGE MAY BE PRINTED AFTER A ERROR MESSAGE IF THE HALT ON ERROR BIT IS SET. HERE AGAIN THE CONTROL CHARACTERS MAY BE USED. THE WAITING MESSAGE MAY BE PRINTED IF OPERATOR INTERVENTION IS REQUIRED..

10.4.4

SWITCH REGISTER MESSAGE

THIS MESSAGE IS USED TO SETUP THE PSEUDO SWITCH REGISTER BEFORE PROGRAM EXECUTION TAKES PLACE. THE SWITCH REGISTER IS SETUP WHEN THE FOURTH CHARACTER IS ENTERED OF A CARRIAGE RETURN IS TYPED

SR=0000 4000

UNDER SCORING INDICATES OPERATOR RESPONSE

10.4.5

END OF PASS

A INDICATION WILL BE GIVEN WHEN THE DIAGNOSTIC HAS MADE A SUCESSFULL PASS. THE PRINT OUT WILL INDICATE THE DIAGNOSTIC MAINDEC NUMBER THE WORD PASS AND A FOUR DIGIT PASS NUMBER. A PASS WILL BE A TIME PERIOD RATHER THAN A PROGRAM PASS OF THE DIAGNOSTIC. THE TIME PERIOD WILL BE IN THE RANGE OF ONE (1) TO FIVE (5) MINUTES. IF THE DIAGNOSTIC MAKES A PROGRAM PASS IN THE 1 TO 5 MINUTE RANGE THEN THE PASS COUNT WILL BE THE SAME AS THE NUMBER OF PROGRAM PASSES. IF THE PROGRAM MAKES A PROGRAM PASS IN LESS THEN ONE MINUTE THEN THE PASS COUNT WILL NOT BE THE SAME AS THE PASS COUNTER THE PASS COUNTER WILL REFLECT MORE THEN ON PROGRAM PASS. THE NUMBER OF PROGRAM PASSES REQUIRED FOR "A PASS MESSAGE CAN BE FOUND IN LOCATION CALLED "CNTVAL". IF HALT AT END OF PASS IS SET THEN THE PASS MESSAGE WILL BE PRINTED AND A WAITING STATEMENT WILL ALSO BE PRINTED. A CONTROL CHARACTER IS NEEDED TO CONTINUE FROM THIS MESSAGE. THE FORMAT OF THE END OF PASS MESSAGE IS

NAME PASS 0001 (- OR I OR C OR D)
- AN ERROR OCCUPED DURING I,C,D
I INTERFACE TEST OK RX8
C RX8 AND RX01 TESTED OK
D RX8 AND RX01 AND DRIVE TESTED OK

10.5. ERRORS

UPON DETECTION OF A ERROR THE DIAGNOSTIC WILL DO ONE OF THE FOLLOWING OPERATIONS:

1.) PRINT THE ERROR MESSAGE FOR THE ERROR CONDITION
CHECK THE SWITCH REGISTER TO SEE IF THE PROGRAM SHOULD HALT
IF HALT ON ERROR IS SELECTED THEN WAITING WILL BE PRINTED
TO GO ON FROM THE ERROR TYPE CONTROL E
REFERE TO THE LISTING AT THE LOCATION PRINTED IN THE ERROR
PC FOR THE CAUSE OF THE ERROR.

10.5.1 ERROR HALTS

CONSOLE PACKAGE DEACTIVE WILL CAUSE NO ERROR MESSAGE TO BE PRINTED. A HALT WILL REPLACE THE ERROR CALL IN THE CODE AND THE DIAGNOSTIC WILL THEN GO TO THAT HALT. REFER TO THE LISTING FOR THE CAUSE OF THE ERKOR, THE ERROR LOCATION WILL BE THE SAME IF THE CONSOLE PACKAGE WAS ACTIVE.

10.6. SWITCH REGISTER SETTINGS

THE FOLLOWING SWITCH REGISTER SETTINGS ARE USED BY THE CONSOLE PACKAGE, THESE SWITCH REGISTER SETTINGS ARE VALID WHEN USING THE HARDWARE SWITCH REGISTER AND THE PSEUDO SWITCH REGISTER.

10.6.1 OPERATING SWITCHES

THE CONSOLE PACKAGE USES THE LOCATIONS 20 21 22 FOR THE FOLLOWING PURPOSES.

LOCATION 20
PSEUDO SWITCH REGISTER

LOCATION 21
HARDWARE IDENTIFIER 1

LOCATION 22
HARDWARE IDENTIFIER 2

LOCATION 0022

BIT	OCTAL VALUE	FUNCTION WHEN 0	FUNCTION WHEN 1
---	-----	-----	-----
0	4000	NOT ON ACT8A LINE	ON ACT 8A LINE
1	2000	NOT ON ACT 8E LINE	ON ACT 8E LINE
2	1000	NOT YET DEFINED	
3	400	DEACTIVE CONSOLE PACKAGE	ACTIVE CONSOLE PACKAGE

6.2 ERROR RELATED SWITCHES

THESE ARE THE SWITCH REGISTER SETING THAT THE CONSOLE PACKAGE
WILL RECOGNISE.

BIT	OCTAL VALUE	FUNCTION WHEN 0	FUNCTION WHEN 1
---	-----	-----	-----
0	4000	CONTINUE AFTER ERROR	STOP AFTER ERROR
1	2000	CONTINUE AFTER END OF PASS	STOP AFTER END OF PASS
2	1000	SHORT ERROR PRINTOUT	LONG DATA COMPARISON ERROR PRINTOUT
3	400	PRINT ERROR MESSAGES	DO NOT PRINT ERRORS
4	200	NO SCOPE LOOP ON ERROR	LOCK SCOPE LOOP ON ERROR
5	100	NO SCOPE LOOP ON TEST	LOCK SCOPE LOOP ON TEST(OK OR NOT)
6	40	ISSUE[INIT]AT ERROR	DO NOT ISSUE [INIT] AT ERROR
11	1	RING BELL ON ERROR	NO BELL RINGING

10.7. LOCATION CHANGES

THE FOLLOWING LOCATIONS CAN BE CHANGED TO MEET THE SPECIFIC
NEED FOR MODIFICATION OF THE DIAGNOSTIC.

CNTVAL IS THE LOCATION FOR THE VALUE OF THE
 NUMBER OF PROGRAM PASSES NEED TO
 PRINT THE END OF PASS MESSAGE.

FILLER IS THE LOCATION SET FOR THE NUMBER OF
 FILLER CHARACTERS AFTER A CRLF SET TO FOUR (4)

10.8. PROGRAM DESCRIPTION

PARAMETER SELECTIONS:

BITS 0-1

SELECT DRIVES: 0000 =SELECT DRIVES 0 AND 1

4000 =SELECT DRIVE 1

2000 =SELECT DRIVE 0

BIT2

DRIVE/CONTROLLER TESTS

0000 =DO DRIVE AND CONTROLLER

1000 =DO CONTROLLER ONLY

BITS 3-5

DEVICE CODE SELECTION

PUT ONE DIGITE DEVICE CODE IN HERE

THE PROGRAM ASSUMES THE FIRST DIGIT TO BE 7

EXAMPLE :IF DEVICE CODE IS 75 ENTER A 500

BITS 7-11

TEST SELECTION

PUT IN TEST YOU WISH TO RUN

A 00 TEST SELECTION IS ALL TESTS

10.9. DIALOGUE FOR CONSOLE PACKAGE

THE DIALOGUE FOR THIS DIAGNOSTIC IS:

R DIRXA (CR)

MAINDEC-08-DIRXA-B

/THE -B IS THE REVISION LEVEL

REMOVE DIAGNOSTIC DISKETTE

/THE PROGRAM WILL WAIT FOR
/A CONTROL E BEFORE CONTINUING

SELECT PARAMATERS(INCLUDING DEVICE CODES)

SR=0500

/DEVICE CODE SELECTED IS 75
/0500=DRIVES 0 AND 1 DEVICE CODE 75
/4500=DRIVE 1 DEVICE CODE 75
/2500=DRIVE 0 DEVICE CODE 75

TEST PARAMATERS: 0500

/VERIFICATION FOR PARAMETERS

OD=0001 ID=0114 FIRST= 0001 LAST= 0032 /DISKETTE RANGE SELECTED

SR=0500 6000

/6000= SWITCH SETTING STOP ON ERROR AND
/AT END OF PASS

11. APT-8 INTERFACES

11.1 DESCRIPTION

TWO INTERFACES HAVE BEEN PROVIDED WHICH WILL ALLOW THIS DIAGNOSTIC TO RUN UNDER THE STANDARD APT-8 SYSTEM. THESE INTERFACES ARE:

1. TIMING INTERFACE
2. ERROR INTERFACE

EACH WILL BE EXPLAINED IN MORE DETAIL.

11.2 SETUP

IN ORDER TO RUN UNDER APT-8, ADDRESSES 20 AND 22 MUST BE ESTABLISHED PRIOR TO RUNNING THE PROGRAM UNDER APT-8 CONTROL. THE FOLLOWING INFORMATION MUST BE INDICATED:

1. DEVICE CODE OF RX01 CONTROLLER UNDER TEST.
2. RX01 CONTROLLER CABLED TO INTERFACE.
3. DRIVE OR DRIVES TO BE TESTED.
4. DIAGNOSTIC RUNNING UNDER THE APT-8 SYSTEM.
5. STARTING TEST NUMBER IF OTHER THAN THAT FOR ACCEPTANCE.

ADDRESS 20

ADDRESS 20 IS USED TO ESTABLISH ALL BUT ITEM 4. THE SET UP IS THE SAME FOR THAT OF THE STANDARD SWITCH REGISTER FUNCTION.

0	1	2	3	4	5	6	7	8	9	10	11
U	U	C	X	X	X	-	T	T	T	T	T
(STARTING TEST)											

- 0 = 0 - DISKETTE DRIVE(S) IS READY
- 1 = 0 - (POWER APPLIED / DOOR CLOSED)
- 2 = 0 - RX01 MICROCONTROL CABLED TO RX8 INTERFACE
- 3 = - FOR STANDARD DEVICE CODE
- 4 = - POSITION CONSOLE SWITCHES
- 5 = - 3-4-5 = 0
- 7 = - TO EXECUTE ALL TESTS
- 8 = - CONFIGURE CONSOLE SWITCHES
- 9 = - 7 THRU 11
- 10 = - EQUIVALENT TO ZERO
- 11 = - (THIS IS THE STARTING TEST)

NOTE: IF MORE THAN ONE DEVICE CODE IS AVAILABLE ON THE SYSTEM THE DIAGNOSTIC WILL HAVE TO RELOADED AND THE PROPER DEVICE CODE SELECTED.

ADDRESS 22

THIS ADDRESS IS USED TO INDICATE THAT THE PROGRAM IS RUNNING ON APT-8, THE NUMBER OF DRIVES TO BE DONE, AND IF SINGLE DRIVE TESTING.

BIT ZERO MUST BE A ONE (1) TO INDICATE THAT THE DIAGNOSTIC IS RUNNING ON APT-8.

CAUTION:

WHILE UNDER APT-8 CONTROL THE HARDWARE SWITCH REGISTER IS FUNCTIONAL. IT IS RECOMMENDED THAT THE SWITCH REGISTER BE SET TO ZERO PRIOR TO START-UP.

11.3 APT-8 INTERFACES.

11.3.1 TIMING

APT-8 IS NOTIFIED OF PROGRAM RUN BETWEEN .2 SECONDS AND 2.0 SECONDS. THIS WILL ALLOW THE DIAGNOSTIC TO RUN UNDER THE MUCH SLOWER MOS MEMORY WITHOUT CAUSING APT-8 TO GIVE A TIMEOUT ERROR.

11.3.2 ERRORS

ONLY THE ERROR PC IS REPORTED TO APT-8. THE TYPE OF ERROR CAN BE DETERMINED FROM THE CORRESPONDING ADDRESS IN THE PROGRAM LISTING. THERE IS A POSSIBILITY THAT A TIMEOUT ERROR MAY OCCUR. THIS IS CAUSED BY THE ERROR "HUNG DEVICE". THE PROGRAM WILL HAVE TO BE RERUN IN DUMP MODE IF THIS SHOULD HAPPEN.

11.4 LOADING PRECAUTIONS

THIS PROGRAM SHOULD BE LOADED IN SRCIPT MODE INDICATING TO APT-8 THAT CORE SUMCHECKS ARE TO BE IGNORED.

12. PROGRAM LISTING

```

1      /RX8 RX01 DIAGNOSTIC DIRXA-C
2      0001      FIELD 1
3      /CONSOL SRC=V1R4- CONSOL PACKAGE
4
5
6
7      /SET UP A LAS TO BE EQUALL TO THE CALL C0CKSW
8
9      /PROGRAM SHOULD CHECK FOR A CONTROL CHARACTER FROM THE CONSOL
10     /EVERY FIVE SECONDS OR LESS
11
12
13
14     /SETUP CNTVAL FOR A RANGE OF 1 TO 4 MINUTES FOR C8PASS TO PRINT PASS
15     /SETUP OF CNTVAL WILL BE FOUND IN C8PASS
16     /THIS VALUE SHOULD BE A POSITIVE NUMBER.
17
18
19     /SET UP XDOSW AS THE VALUE NEEDED FOR A RETURN FOR CONTROL R
20     /RETURN TO ASK THE SWITCH REGISTER QUESTION.
21
22     /CHANGE 1 AND 2 APRIL 16 1975
23
24     /CHANGE 3 APRIL 18,1975
25
26     /CHANGE 4 APRIL 22 1975
27
28     /CHANGE 5 APRIL 23 1975
29
30
31     /CHANGE 6 APRIL 24,1975
32
33
34     /CHANGE 7 APRIL 25,1975
35
36
37
38     /THE CALL TABLE IS A CONDITIONAL ASSEMBLY.
39     / TO ASSEMBLE THE CALL REMOVE THE / BEFORE CONSOL =0.
40     /IN COMBINING THE CONSOL PACKAGE TO A DIAGNOSTIC.
41     / THE CALL TABLE IS TO BE AT THE BEGINNING OF A PROGRAM.
42     /CONSOL=0
43     6661      PSKF= 6661
44     6662      PCLF= 6662
45     6663      PSKE= 6663
46     6664      PSTB= 6664
47     6665      FSIE= 6665
48     6004      GTF= 6004
49     7701      ACL= 7701
50     6007      CAF= 6007
51     7421      MQL= 7421
52     4461      HLT= HALT
53
54     7501      /#6      MQA= 7501
55     /#6

```

```

56     0200      *200
57
58     /*****
59     /C8PASS
60     /THIS IS CALLED AT THE END OF EACH PROGRAM COMPLETION
61     /THE VALUE OF** CNTVAL** WILL BE DETERMINED BY THE TIME IT TAKES
62     /THE PROGRAM TO COMPLETE THIS MANY C8PASS TO BE IN THE 1 TO 4 MINUTE
63     /RANGE
64     /      C8PASS=JMS      XC8PAS
65     /EX.      C8PASS
66     /      HLT              /HALT IF NON CONSOL PACKAGE
67     /      JMP      START1  /CONTINUE RUNNING THIS PROGRAM
68     /RETURN TO LOCATION CALL PLUS ONE WITH THE AC=0 IF NON CONSOL PACKAGE AND HLT
69     /IF CONTINUE TO RUN THEN RETURN TO CALL PLUS2 AC=0
70
71     0200 0000  XC8PAS, 0
72     0201 4777* JMS  C0CK22      /CHECK IF CONSOL IS ACTIVE
73     0202 5210 JMP  DOPACK      /IS CLASSIC
74     0203 4267 JMS  XC8SW       /CHECK SR SETTING
75     0204 0376 AND  (400       /FOR HALT ON END OF C8PASS
76     0205 7640 SZA CLA      /I= HALT 0 CONTINUE
77     0206 5234 JMP  C8EXT1     /DO ONE EXIT
78     0207 5233 JMP  C8BY1     /CONTINUE ON RUNNING PROGRAM
79     0210 4236 JMS  CKCOUT     /CLASS CHECK C8PASS COUNT
80     0211 5233 JMP  C8BY1     /C8PASS COUNT NOT DONE REDO PROGRAM
81     0212 2255 ISZ  PASCNT     /C8PASS COUNT DONE SET C8PASS COUNT
82
83     0213 4775* /#2      JMS  XC8CRL
84
85     0214 4317 /#2      JMS  XC8PNT      /C8PRNT BUFFER
86     0215 0260 MESPAS
87     0216 1255 TAD  PASCNT     /GET NUMBER
88     0217 4774* JMS  XC8OCT     /CONVERT IT TO ASCII
89     0220 6201 CDF 0
90     0221 1653 TAD I      XMX      /GET THE CHAR TO PRINT
91     0222 6211 CDF 10
92     0223 3236 DCA  CKCOUT     /STORE FOR PRINTING
93     0224 4317 JMS  XC8PNT     /PRINT IT
94     0225 0236 CKCOUT
95     0226 4775* JMS  XC8CRL     /DO A CARRIAGE RETURN
96     0227 4267 JMS  XC8SW       /CHECK A HALT AT END OF C8PASS
97     0230 0376 AND  (400       /MASK BIT
98     0231 7640 SZA CLA      /HALT =1 NO SKIP CONTINUE =0
99     0232 4773* JMS  XC8INGU     /STOP PROGRAM EXECUTION-LOOK FOR INPUT
100    0233 2260 C8BY1, ISZ  XC8PAS     /BUMP RETURN
101    0234 6203 C8EXT1, CDF CIF 0
102    0235 5600 JMP  I      XC8PAS
103    0236 0000 CKCOUT, 0
104    0237 1256 TAD  DOSET      /CHECK IF SET UP NEEDED
105    0240 7640 SZA CLA      /0=SET UP C8PASS COUNT VALUE
106
107    0241 5246 JMP  NOSET     /I=C8PASS COUNT VALUE OK
108    0242 1257 TAD  CNTVAL     /C8PASS COUNT VALUE ON
109    0243 7040 CMA
110    0244 3254 DCA  DOCNT     /GET COUNT VALUE FOR THIS PROG
                    /SET TO NEGATIVE
                    /STORE IN HERE

```

```

111 0245 2256          ISZ  DOSET          /INDICATE VALUE SET UP
112
113 0246 2254          NOSET, ISZ  DOCNT          /COUNT THE NUMBER OF PASSES
114          /#2
115 0247 5233          JMP  C8BY1          /EXIT FOR ANOTHER PASS
116 0250 3256          DCA  DOSET          /SET TO C8PNT C8PASS
117 0251 2236          ISZ  CKCOUT          /BUMP RETURN FOR
118 0252 5636          JMP I CKCOUT          /C8PASS C8TYPE OUT
119 0253 0556          XMX,  MX          /LOCATION OF CHAR TO PRINT AT PASS FROM FLD 0
120 0254 0000          DOCNT, 0
121 0255 0000          PASCNT, 0
122 0256 0000          DOSET, 0
123 0257 0000          CNTVAL, 0
124 0260 0411          MESPAS, TEXT      "DIRXA-C PASS "
      0261 2230
      0262 0155
      0263 0340
      0264 2001
      0265 2323
      0266 4000

125
126
127
128          /*****
129          /C8CKSW
130          /ROUTINE THAT WILL CHECK WHEPE TO READ THE
131          /C8 SWITCHES FROM IE, FROM PANEL OR PSEUDO C8SWIT REGISTER
132          / C8CKSW= JMS XC8SW
133          /EX  C8CKSW          /READ THE C8SWIT REGISTER
134          /RETURN WITH THE CONTENTS OF SWITCH REGISTER
135
136          /RETUPN TO NEXT LOCATION FOLLOWING CALL WITH THE AC= TO VALUE OF C8SWIT SETTING
137
138
139 0267 0000          XC8SW, 0
140          /#1
141 0270 7200          CLA
142 0271 4772          JMS  XC8CKP          /CLEAR AC
143          /#2
144 0272 7000          NOP          /GO CHECK THE IF ANY CONTRL
145          /#2
146          /#1
147 0273 6201          CDF 00
148 0274 1771          TAD I (21          /GET WD FOR INDICATOR
149 0275 6211          CDF I0
150 0276 7710          SPA CLA          /CHECK IF FROM PANEL 4000
151 0277 7614          7614          /DO LAS AND SKIPGET FROM PANEL WITH LAS
152 0300 5302          JMP  CRGET1          /GET CONTENTS OF LOC 20 FLD 00
153 0301 5305          JMP  C8EXT2          /EXIT COMMON
154 0302 6201          CRGET1, CDF 00
155 0303 1770          TAD I (20          /PSEUDO SW
156 0304 6211          CDF I0
157 0305 5667          CREXT2, JMP I XC8SW          /EXIT WITH STATUS BIT IN AC.
158
159

```

```

160          /*****
161
162          /C8TTYI
163          /THIS ROUTINE WILL LOOK FOR A INPUT FROM THE CONSOL
164          / C8TTYI= JMS XC8TTY
165          /EX. C8TTYI          /READ CHAR FROM THE CONSOL DEVICE
166          /RETURN TO CALL PLUS ONE AC CONTAINS THE CHAR
167
168          /
169          /
170 0306 0000          XC8TTY, 0
171 0307 6031          KSF
172 0310 5307          JMP  .-1          /LOOK FOR KEYBOARD FLAG
173 0311 6036          KRB
174 0312 0367          AND  (177          /GET CHAR
175 0313 1366          TAD  (200          /MASK FOR 7 BITS
176 0314 3765          DCA  CHAR          /ADD THE EIGHTH BIT
177 0315 1765          TAD  CHAR          /STORE IT
178 0316 5706          JMP I XC8TTY          /EXIT
179
180
181          /*****
182
183          /C8PNT
184
185          /THIS ROUTINE WILL TYPE THE CONTENTS OF THE C8 PRINT BUFFER. THE LOCATION
186          /OF THE BUFFER WILL BE IN THE ADDR FOLLOWING THE CALL. C8 PRINTING OF THE BUFFER
187          /WILL STOP WHEN A 00 CHAR IS DETECTED. CHARACTERS ARE PACKED 2 PER WORD.
188
189          / C8PNT= JMS XC8PNT
190
191
192          /EX. C8PNT          /C8PNT THE CONTENTS OF THE FOLLOWING BUFFER
193          / MESS77          /LOCATION OF C8PNT BUFFER
194          /C8PNT WILL USE THE LOCATION FOLLOWING THE CALL AS THE POINTER FOR THE
195          /C8PNT ROUTINE. RETURN TO CALL PLUS TWO WITH AC= 0
196
197
198
199
200 0317 0000          XC8PNT, 0
201 0320 7300          CIA CLL
202 0321 1717          TAD I XC8PNT          /GET C8PNT BUFFERS STARTING LOCATION
203 0322 3352          DCA  PTSTOR          /STORE IN PTSTOR
204 0323 2317          ISZ  XC8PNT          /BUMP RETURN
205 0324 1752          CBD01, TAD I PTSTOR          /GET DATA WORD
206 0325 0364          AND  (7700          /MASK FOR LEFT BYTE
207 0326 7450          SNA
208 0327 5717          JMP I XC8PNT          /CHECK IF 00 TERMINATE
209 0330 7500          SNA          /EXIT
210 0331 7020          CML          /IS AC MINUS
211 0332 7001          IAC          /MAKE CHAR A 300 AFTER ROTATE
212 0333 7012          RTR          /MAKE CHAR A 200 AFTER ROTATE
213 0334 7012          RTR
214 0335 7012          RTP          /PUT CHAR IN BITS 4-11 MAKE IT 8 BIT ASCII

```

```

215 0336 4763* JMS XC8TYP /C8PRNT IT ON CONSOLE
216 0337 1752 TAD I PTSTOR /GET DATA WORD
217 0340 0362 AND (0077) /MASK FOR RIGHT BYTE
218 0341 7450 SNA /CHECK IF 00 TERMINATOR
219 0342 5717 JMP I XC8PNT //EXIT
220 0343 1361 TAD (3740) /ADD FUDGE FACTOR TO DETERMINE IF 200
221 0344 7500 SMA /OR 300 IS TO BE ADD TO CHAR
222 0345 1360 TAD (100) /ADD 100
223 0346 1357 TAD (240) /ADD 200
224 0347 4763* JMS XC8TYPE /C8TYPE ONLY BITS 4-11
225 0350 2352 ISZ PTSTOR /BUMP POINTER FOR NEXT WORD
226 0351 5324 JMP C8D01 /DO AGAIN
227 0352 0000 PTSTOR, 0 /STOR FOR C8PRNT BUFFER
228 0353 0000 STOPNT, 0 /0000 C8PRNT 7777=DO NOT C8PRNT
229
230
231 /*****
232
233
234 0357 0240
235 0360 0100
236 0361 3740
237 0362 0077
238 0363 1106
239 0364 7700
240 0365 1104
241 0366 0200
242 0367 0177
243 0370 0020
244 0371 0021
245 0372 1050
246 0373 0635
247 0374 1007
248 0375 1032
249 0376 0400
250 0377 1330
251 0400
PAGE
252 /*****
253
254 /C8CNTR
255 /THIS ROUTINE WILL CHECK FOR THE PRESENCE OF CONTROL CHARACTERS
256 /IT WILL CHECK FOR THE FOLLOWING CHAR C-K-Q-O-L-S
257 / C8CNTR= JMS XC8CNT
258
259 /EX. C8CNTR /CHECK FOR CONTROL CHARACTER
260 / JMP ANYTHING /LOC FOLLOWING CALL IS FOR CONTINUING THE PROGRAM
261 / JMP ANYTHING /LOC. IS FOR RETURN IF INMODE SET AND NOT CNTRL CHAR
262 /
263 /RETURN IS TO CALL PLUS ONE IFCONTINUE
264 /RETURN IS TO CALL PLUS TWO IF INMOD SET AND NOT CONTROL CHAR
265 /
266 /RETURN IS TO CALL PLUS TWO IF INMODE IS NOT SET AND NO
267 /CONTROL CHAR ..THIS WILL PRINT THE CHARACTER AND A ?
268 /CLEAP THE AC AND RETURN CALL+2.

```

```

269
270 0400 0000 XC8CNT, 0
271 0401 3777* DCA ACSAVE /SAVE THE AC
272 0402 4776* JMS CRCK22 /CHECK IF CONSOL ACTIVE
273 0403 5206 JMP ,+3 /ON ACTIVE CONSOLE
274 0404 1777* TAD ACSAVE /GET AC FOR RETURN
275 0405 5600 JMP I XC8CNT /EXIT NOT ON ACTIVE CONSOLE
276 0406 6004 GTF
277 0407 3775* DCA FLSAVE
278
279 0410 7501 /#6
280 /#6 MOA
281 0411 3774* DCA MQSAVE /SAVE THE MQ
282 0412 3255 DCA INDEXA /SET DISPLACEMENT INTO TABLE B
283 0413 1257 TAD XTABLA /GET ADDR OF TABLE A
284 0414 3256 DCA GETDAT /CONTAINS POINTER TO CONTROL CHAR
285 0415 1656 REDOA, TAD I GETDAT /GET CONTROL CHAR FROM TABLE
286 0416 7450 SNA /CHECK FOR A 0 END OF TABLE
287 0417 5226 JMP DONEA /END OF TABLE NO CONTROL CHAR
288 0420 1773* TAD CHAR /COMPARE CHAR TO CONTROL CHAR
289 0421 7653 SNA CLA /0 IF MATCH
290 0422 5243 JMP GOITA /MATCH
291 0423 2255 ISZ INDEXA /NO MATCH NOT END OF TABLE REDO
292 0424 2256 ISZ GETDAT /BUMP INDEX FOR EXIT WHEN CONTROL FOUND
293 0425 5215 JMP REDOA /BUMP GETDAT FOR COMPARE OF NEXT CNTRL CHAR.
294 0426 1772* DONEA, TAD INMODE /CHECK IF PROGRAM EXPECTS CHAR
295 0427 7640 SZA CLA /1=CHAR EXPECTED 0= NO CHAR EXPECTED
296 0430 5240 JMP EXITA /CHAR EXPECTED
297 0431 1773* TAD CHAR /GET CHAR - NOT CONTROL + NOT EXPECTED
298 0432 4771* JMS XC8TYP /C8PRNT CHAR
299 0433 1370 TAD (277) /GET CODE FOR "*"
300 0434 4771* JMS XC8TYP
301 0435 4767* JMS XC8CRL
302
303 0436 2200 /#1 ISZ XC8CNT /BUMP RETURN
304 /#2
305 0437 5600 JMP I XC8CNT /EXIT CALL+2
306 /#2
307 /#1
308 0440 2200 EXITA, ISZ XC8CNT /BUMP RETURN FOR MAIN PROGRAM CHECK OF CHAR
309 0441 1773* TAD CHAR /PUT CHAR IN AC.
310 0442 5600 JMP I XC8CNT /EXIT
311 0443 1773* GOITA, TAD CHAR /GET THE CONTENTS OF CHAR
312 0444 1366 TAD (100) /ADD 100 TO FORM A GOOD ASCII CHARACTER
313 0445 3773* DCA CHAR /RESTORE COEFFCT CHAR
314 0446 1260 TAD XTABLA /GET START OF TABLE B
315 0447 1255 TAD INDEXA /GET NOW FAR INTO TABLE
316 0450 3254 DCA GOTOA /STORE IT
317 0451 1654 TAD I GOTOA /GET THE ROUTINE STARTTING ADDRESS
318 0452 3254 DCA GOTOA /STORE IT IN HERE
319 0453 5654 JMP I GOTOA /GOTO CONTROL CHAR ROUTINE
320 0454 0000 GOTOA, 0000 /ADD OF CNTRL ROUTINE TO EXECUTE
321 0455 0000 INDEXA, 0000 /DISPLACEMENT INTO CNTRL TABLE
322 0456 0000 GETDAT, 0000 /LOCATION OF ADDR OF CONTROL CHAR.
323 0457 0461 XTABLA, TABLA /ADDRS OF TABLE

```

```

324 0460 0470 XTABL, TABLB /ADDS OF TABLEB
325 0461 7575 TABLA, 7575 /CNTRL C BACK TO MONITOR 203
326 0462 7557 7557 /CNTRL Q START DISPLAYING CHAR, AGAIN 221
327 0463 7556 7556 /CNTRL R BACK TO BEGINNING OF PROGRAM 222
328 0464 7555 7555 /CNTRL S STOP SENDING CHAR TO DISPLAY WAIT FOR CNTRL Q 223
329 0465 7573 7573 /CNTRL E CONTINUE WITH PROGRAM 205
330 0466 7574 7574 /CONTROL D CHANGE SWITCH REGISTER ON FLY
331 0467 0000 0000
332 0470 0476 TABLB, CNTRLQ
333 0471 0503 CNTRLQ
334 0472 0514 CNTRLR
335 0473 0524 CNTRLS
336 0474 0542 CNTRLE
337 0475 0600 CNTRLD
338
339 /RETURN TO MONITOR
340 /#7
341 0476 3765* CNTRLQ, DCA TTYLPT /CLEAR THE LPT FLAG TO PRINT ON DISPLAY
342 /#7
343 0477 4764* JMS UPAROW /CPRINT A AND LETTER IN CHAR
344 0500 6203 CDF CIF /GO TO 0 FLD
345 0501 6007 CAF /CLEAR THE WORLD
346 0502 5763 JMP I (7600 /GO TO DIAGNOSTIC MONITOR
347
348 /
349 /START SENDING CHAR, TO THE DISPLAY
350 /THIS WILL RETURN CONTROL TO CALL THAT WAS SET BY
351 /THE CALL FOR CONTROL S.
352 /
353 /#7
354 0503 3772* CNTRLQ, DCA INMODE /CLEAR THE SOFT IN FLAG
355 0504 1340 TAD C0SETS /GET THE SOFT FLAG
356 0505 7640 SZA CLA
357 0506 4311 JMS BYRETR /FIRST ENTRY TO CNTRL Q WITH NO CNTRL S
358 0507 4762* JMS CBGET
359 0510 5600 JMP I XCBCNTR /RETURN VIA CONTROL
360 0511 3340 BYRETR, DCA C0SETS /CLEAR THE FLAG FOR CONTROL S
361 0512 4762* JMS CBGET /RESTOR
362 0513 5741 JMP I C0RETR /GO TO CALL SAVED BY CNTRL S
363
364 /
365 /GO TO THE QUESTION C0SWIT
366 /
367 /
368 0514 3765* CNTRFLR, DCA TTYLPT /SET FLAG TO TTY
369 0515 3340 DCA C0SETS /CLEAR SOFT FLAG FOR CNTRL S
370 0516 3772* DCA INMODE
371 0517 4764* JMS UPAROW /PRINT CHAR
372 0520 3761* CBBY4, DCA C0SWST /CLEAR FLAG FOR CNTRL D OR P
373 /#3
374 0521 6203 CDF CIF 00
375 0522 5723 JMP I XD0SW /GO TO ADDRS OF C0SWIT
376
377 0523 0200 /#1 XD0SW, CRSTRT /D0SW IS LABEL FOR C0SWIT QUESTION
378 /#1

```

```

379 /
380 /
381 /STOP SENDING CHAR, TO DISPLAY UNTIL A "Q" IS RECEIVED
382 /
383 /
384 /#7
385 0524 1340 CNTRLS, TAD C0SETS /IF1 DO NOT STORE IN C0RETR
386 /#7
387 0525 7640 SZA CLA
388 0526 5332 JMP C007 /DONT SET UP C0RETR
389
390 0527 7001 /#5 IAC /MAKE RETURN CALL PLUS 2
391 /#5
392 0530 1200 TAD XCBCNT /GET RETURN FOR THIS CALL
393 0531 3341 DCA C0RETR /STORE IT HERE FOR USE BE CNTRL Q
394
395 0532 2340 C0D07, ISZ C0SETS /SET FLAG TO SAVE CALL
396 /#4
397 0533 4760* JMS XC0TTY /LOOK FOR THE INPUT
398
399 0534 4762* /#5 JMS CBGET /GET REGISTERS
400
401 0535 4200 /#5 JMS XCBCNTR /CHECK FOR THE CONTROL CHAR
402 /#4
403 0536 7200 CLA
404
405 0537 5324 /#7 JMP CNTRLS /IF NOT A CNTRL Q R C REASK
406 /#7
407 /#2
408 0540 0000 C0SETS, 0
409 0541 0000 C0RETR, 0
410 /
411 /SWITCH OUTPUT FROM ONE OUTPUT DEVICE TO ANOTHER - THE TWO OUTPUTS ARE THE
412 /CONSOLE AND THE PRINTER WITH DEVICE CDOE 66.
413 /
414 /
415 /
416 /
417 /
418 /
419 /
420 /CONTROL E
421 /CONTINUE RUNNING FROM A INQUIRE OR ERROR
422 /
423 /
424 0542 4764* CNTRLE, JMS UPAROW /PRINT THE CONTROL CHAR
425 0543 4762* JMS CBGET /GET THE REGISTERS
426 0544 5600 JMP I XCBCNT /RETURN TO CALL PLUS ONE
427 /
428 /
429 /
430 0560 0306
431 0561 0745
432 0562 0624
433 0563 7600

```



```

434 0564 0615
435 0565 1132
436 0566 0100
437 0567 1032
438 0570 0277
439 0571 1106
440 0572 1105
441 0573 1104
442 0574 1315
443 0575 1316
444 0576 1330
445 0577 1314
446 0600
PAGE
/
/CONTROL D
/CHANGE THE SWITCH REGISTER ANYTIME CNTRL D AND RETURN TO
/THE PROGRAM RUNNING.
451
452 0600 4215 CNTRLD, JMS UPAROW
453 0601 1213 TAD C9SETD /CHECK IF THE RETURN ADDR IS SAFE
454 0602 7640 SZA CLA
455 0603 5207 JMP C8D011 /DO NOT CHANGE THE RETURN ADDR
456 0604 1777* TAD XC8CNT /GET THE RETURN ADDR AND SAVE IT
457 0605 3214 DCA C8RETD /SAVE THE RETURN HERE
458 0606 2213 ISZ C8SETD /INDICATE RETURN SAVED DONT DESTROY
459 0607 4251 C8D011, JMS XC8PSW /GO CHANGE THE SWITCH REGISTER
460 0610 3213 DCA C8SETD /CLEAR THE FLAG
461
462 0611 4224 /*3 JMS C8GET /RESTORE THE AC MQ LINK ETC
463
464 0612 5614 /*3 JMP I C8RETD /RETURN TO THE PROGRAM
465
466 0613 000J C8SETD, 0
467 0614 0000 C8RETD, 0
468
469
470
471
472 0615 0000 UPAROW, 0 /PRINT THE "" AND THE CHAR TYPED IN
473 0616 1376 TAD (336 /CODE FOR "
474 0617 4775* JMS XC8TYP
475 0620 1774* TAD CHAR /C8TYPE THE CHAR
476 0621 4775* JMS XC8TYP
477 0622 4773* JMS XC8CRLF
478 0623 5615 JMP I UPAROW /EXIT
479
480
481
482
483 0624 0000 /******
484 0625 7200 C8GET, 0
485
486 /*6 CLA
487 0626 1772* TAD MQSAVE

```

```

488 0627 7421 MGL /RESTORE MQ
489 0630 1771* TAD FLSAVE
490 0631 7004 RAL /RESTORE THE LINK
491
492 0632 7200 /*1 CLA
493
494 0633 1770* TAD ACSAVE /RESTORE THE AC
495 0634 5624 JMP I C8GET /GET THE REGISTERS
496
497
498 /******
499
500 /C8INQU
501 /C8INQU ROUTINE WILL PRINT A WAITING
502 /AND THE PROGRAM IS EXPECTING A CONTROL CHAR INPUT
503 /IF CONTINUE FROM CONTROL CHAR RETURN IS CALL PLUS ONE
504 /IF NO CONTROL CHAR ENTERED THEN WAITING IS REPRINTED
505 /AND PROGRAM WAITS FOR A CONTROL CHAR AGAIN
506 //
507
508 / C8INQU= JMS XC8INQ
509
510 /EX C8INQU /C8 WILL PRINT A WAITINGAND WAIT FOR INPUT
511 DO ANYTHING /RETURN IS CALL PLUS ONE AC =0 CONTINUE
512
513 0635 0000 XC8INQ, 0
514 0636 4767* JMS C8CK22 /CHECK IF CONSOLE
515 0637 7410 SKP /SKIP FOR ACTIVE
516 0640 5635 JMP I XC8INQ /NOT CONSOLE LEAVE
517
518 0641 4773* /*1 JMS XC8CRLF
519 0642 4766* JMS XC8PNT
520 0643 1307 WAITMES /INQUIR WAITING
521 0644 4765* JMS XC8TTY /GET CHARACTER
522
523 /*5 JMS C8GET
524
525 0646 4777* /*5 JMS XC8CNTR /CHECK IF CONTROL CHARACTER
526
527 0647 5635 /*1 JMP I XC8INQ /EXIT AND CONTINUE
528 0650 5236 JMP XC8INQ+1 /REASK
529
530
531 /******
532
533
534 /C8SWIT
535 /ROUTINE WILL CHECK IF CONSOL IS ACTIVE IF IT IS ACTIVE DISPLAY
536 /SW QUESTION, IN NOT ACTIVE IT WILL NOT PRINT THE SW QUESTIONBUT
537 /RETURN TO CALL PLUS ONE AC=0.
538 /C8SWIT WILL SET UP THE PSEUDO C8SWIT
539 /REGISTER WITH THE NEW DATA ENTERED
540 /THE TAG C8DOR AT THE START OF THE CALL IS FOR THE RETURN OF CONTROL R
541 /CHAR, THIS MAY BE CHANGED IF THIS IS NOT WHERE A GOOD RESTART
542 /OF PROGRAM IS.

```

```

543 /
544 / C8SWIT= JMS XC8PSW
545
546 /EX. C8DOK, C8SWIT /SFT UP PSEUDO C8SWIT REGISTER IF
547 /ON THE CONSOL PACKAGE, RETURN IS CALL PLUS ONE AC = 0
548
549
550
551
552 0651 0000 XC8PSW, 0
553 0652 4767* JMS C8CK22 /CHECK IF ON ACTIVE CONSOLE
554 0653 7410 SKP
555 0654 5651 JMP I XC8PSW /RETURN WITHOUT ASKING PSEUDO SWITCH
556 0655 1345 TAD C8SWST /IS THE SOFT FLAG SET FOR SWITCH?
557 0656 7640 SZA CLA /SKIP IF ONE ENTRY AT ATIME OK
558 0657 5764* JMP C8BY4 /SECOND ENTRY WITH OUT A EXIT GO TO SW QUESTION
559 0660 2345 ISZ C8SWST /FIRST ENTRY SET FLAG
560
561 /#3
562 RED01, JMS XC8CRLF
563 JMS XC8PNT /C8PNT SR=XXX
564 MESA
565 CDF 00
566 TAD I (20 /GET CONTENTS OF SW
567 CDF 10
568 JMS XC8OCTA /CONVERT IT TO ASCII
569 TAD (10 /GET SPACE
570 JMS XC8TYPE
571 ISZ INNODE /SET FLAG FOR CHAR EXECTED
572 JMS XC8ECHO /LOOK FOR INPUT
573 TAD TSTCHA /NOT CONTROL TEST IT IS LEGAL
574 CHAR /STORE NEW CHAR IN SW REG
575 CDF 00
576 DCA I (20
577 CDF 10
578 TAD (-3 /GET A MINUS 3
579 DCA TMPCNT /STORE IN TEMP COUNT
580 JMS XC8ECHO /GET NEXT CHAR
581 JMS TSTCHA /CHECK IF CR + GOOD CHAR
582 CDF 00
583 TAD I (20 /GET C8SWIT REGISTER
584 CDF 10
585 RTL CLL /ROTATE IT LEFT 3 PLACES
586 HAL
587 TAD CHAR /GET CHAR + ADD IT TO PREVIOUS CONTENTS
588 CDF 00
589 DCA I (20 /SAVE NEW CONTENTS
590 CDF 10
591 ISZ TMPCNT /BUMP COUNT
592 JMP GETCHI /JMP BACK + GET NEXT CHAR
593 JMP ENDIT /END 4 CHAR C8TYPED IN
594 TSTCHA, 0
595 CIA /CHPL CHAR IN AC
596 TAD (215 /TEST IF IT IS A CARRIAGE RETURN
597 SNA CLA /SKIP IN NOT CR
598 JMP ENDIT /WAS CARRIAGE RETURN

```

```

598 0726 1774* TAD CHAR /NOT CR, GET CHAR
599 0727 1351 TAD (-260 /CHECK IF IT IS IN RANGE
600 0730 7710 SPA CLA /IF NOT POSITIVE C8ERR CHAR SMALLER THEN 260
601 0731 5753* JMP ERR1 /C8ERR - CHAR TOO SMALL
602 0732 1774* TAD CHAR /GET CHAR
603 0733 1352 TAD (-270 /GET A -270 + CHECK IF IT IS LARGER THEN 7
604 0734 7700 SMA CLA /SKIP IF LESS THEN 7
605 0735 5753* JMP ERR1 /C8ERR ON CHAR NOT IN RANGE
606 0736 1774* TAD CHAR /GET CHAR
607 0737 0351 AND (7 /MASK FOR RIGHT BYIE
608 0740 3774* DCA CHAR /STORE IN CHAR
609
610 0741 5721 JMP I TSTCHA /GET CHAR IN AC
611 0742 4773* JMS XC8CRLF /EXIT
612 0743 3345 DCA C8SWST /CLEAR THE PSW ENTRY FLAG
613 0744 5651 JMP I XC8PSW /EXIT PSW
614 0745 0000 C8SWST, 0
615 0746 0000 TMPCNT, 0
616 0751 0007
617 0752 7510
618 0753 1000
619 0754 7520
620 0755 0215
621 0756 7775
622 0757 1072
623 0760 1105
624 0761 0040
625 0762 1007
626 0763 0020
627 0764 0520
628 0765 0306
629 0766 0317
630 0767 1330
631 0770 1314
632 0771 1316
633 0772 1315
634 0773 1032
635 0774 1104
636 0775 1106
637 0776 0336
638 0777 0400
639 1000 1377 ERR1, PAGE
640 1001 4306 TAD (277 /C8PNT
641 1002 4232 JMS XC8TYPE /?
642 1003 5776* JMS XC8CRLF /
643 1004 2322 JMP RED01 /EXIT + ASK AGAIN
644 1005 7540 MESA, TEXT "SR= "
645 1006 0000
646
647 /*****
648 /C8OCTA
649 /OCTAL TO ASCII CONVERSION
650 /THIS ROUTINE WILL TAKE THE OCTAL NUMBER IN THE AC AND CONVERT IT TO ASCII
651 /THE RESULT WILL BE PRINTED ON THE CONSOL DISPLAY

```

```

650 / C8OCTA= JMS XC8OCT
651 /
652 /EX. C8OCTA /AC CONTAINS NUMBER TO BE CHANGE
653 /
654
655 1007 0000 XC8OCT, 0
656 1010 7106 CLL RTL
657 1011 7006 RTL /POSITION THE FIRST CHAR FOR PRINTING
658 1012 3230 DCA C8TMP1 /SAVE CORRECT POSITIONED WORD HERE
659 1013 1375 TAD (-4
660 1014 3231 DCA C8CKP /STORE COUNTER IN HERE
661 1015 1230 C8D04, TAD C8TMP1 /GET FIRST NUMBER
662 1016 0374 AND (0007 /MASK
663 1017 1373 TAD (260 /ADD THE PRINT CONSTANT
664 1020 4306 JMS XC8TYPE /TYPE THE NUMBER
665 1021 1230 TAD C8TMP1 /
666 1022 7006 RTL
667 1023 7004 RAL /PUT NEXT NUMBER IN POSITION
668 1024 3230 DCA C8TMP1 /STORE IT
669 1025 2231 ISZ C8CKP /DONE YET WITH FOUR NUMBERS
670 1026 5215 JMP C8D04 /NOT YET DO MORE
671 1027 5607 JMP I XC8OCT /DONE WITH FOUR
672 1030 0000 C8TMP1, 0
673 1031 0000 C8CKP, 0
674
675 //*****
676
677 /C8CRLF
678 /TYPE CR AND LF WITH FILLERS FOLLOWING EACH LF AND CR
679 /
680 / C8CRLF= JMS XC8CRL
681 /
682 /EX. C8CRLF /C8PKMT A CR AND LF WITH FILL
683 / /RETURN TO CALL PLUS ONE AC =0
684
685 1032 0000 XC8CRL, 0
686 1033 7300 CLA CLL
687 1034 1372 TAD (215 /GET CODE FOR CR
688 1035 4306 JMS XC8TYPE
689 1036 1246 TAD C8FILLER
690 1037 7040 CMA
691 1040 3247 DCA FILCNT /STORE FILLER IN HERE
692 1041 1371 TAD (212 /GET CODE FOR LF
693 1042 4306 C8D02, JMS XC8TYPE
694 1043 2247 ISZ FILCNT /CHECK ON FILLER CHAR
695 1044 5242 JMP C8D02 /TYPE A NON PRINTING CHAR
696 1045 5632 JMP I XC8CRL /EXIT
697
698 1046 0004 /#1/#2 CRFILLER, 0004 /FILLER SET FOR 4 CHAR
699 /#1/#2
700 1047 0000 FILCNT, 0 /COUNTER FOR FILL
701
702
703
704 //*****

```

```

705 /#1
706 /C8CKPA
707 /THIS ROUTINE WILL CHECK IF A CHARACTER WAS ENTERED FROM THE
708 /TERMINAL. IF THE FLAG IS SET AND THE CONSOLE PACKAGE IS
709 /ACTIVE A CHECK IS MADE TO DETERMINE IF IT IS A CONTROL CHAR.
710 /IF IT WAS A CONTROL CHAR THEN ITS CONTROL FUNCTION IS PERFORMED.
711 /IF NOT A CONTROL CHARACTER OR A CONTROL E-D-L-O- IT WILL DO
712 /THE CONTROL FUNCTION AND RETURN TO CALL PLUS 2.
713 /A NON CONTROL CHARACTER WILL BE PRINTED AND A "?" IT WILL RETURN TO
714 /CALL PLUS 2.
715 /IF NO FLAG IS SET OR THE CONSOLE IS NOT ACTIVE THE RETURN IS TO
716 /CALL PLUS 1.
717
718
719 / C8CKPA= JMS XC9CKP
720
721
722 /EX. C8CKPA /CALL TO CHECK IF CONTROL CHAR SET
723 / ANYTHING(SKIP) /RETURN IF NOT FLAG OR NOT CONSOLE ACTIVE
724 / ANYTHING(JMP EXIT SKIP CHAIN) /RETURN IF NOT CONTROL OR CONTINUE CONTROL
725
726
727 1050 0000 XC8CKP, 0
728 1051 3770 DCA ACSAVE /SAVE THE AC
729 1052 6004 GTF /SAVE THE FLAGS
730 1053 3767 DCA FLSAVE /SAVE THE FLAGS
731
732 /#2
733 1054 7501 /#6 MQA /PUT MQ IN AC
734
735 1055 3766 /#6 DCA MQSAVE /SAVE THE MQ
736
737 /#2
738 1056 6031 KSF /CHECK THE KEYBOARD FLAG
739 1057 5270 JMP C8BY3 /EXIT TO CALL PLUS 1
740 1060 4765 JMS C8CK22 /CHECK IF ON CONSOLE
741 1061 7410 SKP
742 1062 5270 JMP C8BY3 /EXIT TO CALL PLUS 1
743 1063 4764 JMS XC8TTY /GET THE CHAR
744
745 /#2
746 1064 4763 JMS C8GET /GET THE FLAGS
747 1065 4762 JMS XC8CNTR /CHECK IF CONTROL CHAR.
748
749 /#2
750 1066 7000 NOP /RETURN IF A CONTINUE CHAR.
751 1067 2250 ISZ XC8CKP /BUMP RETURN FOR CALL PLUS 2
752 1070 4763 C8BY3, JMS C8GET /GET REGISTERS
753 1071 5650 JMP I XC8CKP /SAY GOOD BY
754
755 /#1
756 //*****
757 /C8ECHO
758 /THIS ROUTINE WILL LOOK FOR A CHAR FROM THE KEYBOARD. STORE IT IN LOCATION CHAR
759 /CHECK IF IT WAS A C8CNTR CHARACTER - SET INMODE = C8PRNT CHARACTER
760 / C8ECHO= JMS XC8ECH
761 /EX. C8ECHO /LOOK FOR CONSOLE CHAR C8PRNT IT

```

```

760 /RETURN CALL PLUS ONE AC = CHAR C0TYPED IN
761
762
763
764 1072 0000 /
765 1073 4764* XC0ECH, 0 JMS XC0TTY /WAIT FOR CHAR FROM KEYBOARD
766 /#1
767 /#1
768 1074 4763* JMS C0GET
769 1075 2305 ISZ INMODE /SET INMODE IDENTIFYING THIS AS A EXPECTED CHAR
770 1076 4762* JMS XC0CNTN /GO CHECK IF IT IS A CONTROL CHAR
771 1077 5672 JMP I XC0ECH /WAS A CONTROL CHAR - CONTINUE RUNNING
772 1100 4306 JMS XC0TYPE /NOT A CONTROL CHAR C0PRNT IT
773 1101 3305 DCA INMODE /CLEAR FLAG THAT CHAR EXPECTED
774 1102 1304 TAD CHAR /GET CHAR IN AC
775 1103 5672 JMP I XC0ECH /EXIT
776 1104 0000 CHAR, 0
777 1105 0000 INMODE, 0
778
779 /*****
780
781 /C0TYPE
782 /THIS ROUTINE WILL C0PRNT ON THE CONSOLE OR THE LPT WITH DEVICE CODE 66.
783 /
784 / C0TYPE= JMS XC0TYP
785
786 /EX. C0TYPE /C0PRNT THE CHAR IN THE AC.
787 / /RETURN CALL PLUS ONE AC =0000
788 / /DO NOT CLEAR THE LINK IN THIS ROUTINE NEEDED BYC0CT
789
790
791 1106 0000 XC0TYP, 0
792 1107 3331 DCA PNTBUF /STORE CHAR
793 1110 1332 TAD TTYLPT /CHECK U=TTY 7777=LPT
794 1111 7640 SZA CLA
795 1112 5321 JMP XDOLPT /DO OUT PUT ON LPT
796 1113 1331 TAD PNTBUF
797 1114 6046 TIS
798 1115 6041 TSF
799 1116 5315 JMP .-1
800 1117 6042 TCF
801 1120 5327 JMP C0BY5
802 1121 1331 XDOLPT, TAD PNTBUF /GET CHAR
803 1122 6666 PSTB PCLF /C0PRNT IT
804
805 1123 4333 JMS C0HANG /CHECK KEYBOARD IF HUNG
806
807 1124 6661 PSKF
808 1125 5323 JMP .-2 /WAIT UNTIL DONE
809 1126 6662 PCLF
810 1127 7200 C0BY5, CLA /CLEAR THE AC
811 1130 5706 JMP I XC0TYP /EXIT
812 1131 0000 PNTBUF, 0
813 1132 0000 TTYLPT, 0
814

```

```

815
816 /#6
817 1133 0000 C0HANG, 0 /WILL CHECK KEYBOARD FOR CNTRL CHAR
818 /WILL NEED IF LPT HANGS TO GET OUT
819 1134 4250 JMS XC0CKPA /SEE IF KEYBOARD INPUT
820 1135 7000 NOP
821 1136 5733 JMP I C0HANG /IF HUNG IN LPT SKIP FLAG NOT SET
822 /ONLY WAY OUT IS CNTRL R - C !!!
823
824 /#6
825 1162 0400
826 1163 0624
827 1164 0306
828 1165 1330
829 1166 1315
830 1167 1316
831 1170 1314
832 1171 0212
833 1172 0215
834 1173 0260
835 1174 0007
836 1175 7774
837 1176 0661
838 1177 0277
839 PAGE
840 /*****
841 /C0BERR
842 /THIS ROUTINE WILL DETERMINE WHAT TO DO WHEN A C0BERR IS ENCOUNTERED
843 /WILL CHECK IF CLASSIC SYSTEM, WILL CHECK C0SWIT REGISTERS.
844 / C0BERR= JMS XC0ERR
845 /EX. C0ERR /GO TO C0BERR CALL IF N0LT/0/
846 / /RETURN IS CALL PLUS ONE AC =0000
847
848
849
850 1200 0000 XC0ERR, 0
851 1201 6002 IOF
852 1202 3314 DCA ACSAVE /SAVE AC
853 1203 6004 GTF
854 1204 3316 DCA FLSAVE /SAVE THE FLAGS
855 1205 7501 MQA
856 1206 3315 DCA MQSAVE /SAVE THE MQ
857 1207 7340 CLA CLL CMA /SUBTRACT A 1 FOR TRUE LOCATION
858 1210 1200 TAD XC0ERR /GET RETURN LOCATION
859 1211 3313 DCA PCSAVE /SAVE ADD OF C0BERR CALL
860 1212 1022 TAD 22 /GET LOCATION 22
861 1213 0377 AND (400 /MASK FOR CLASSIC SYSTEM
862 1214 7650 SNA CLA /SKIP IF CLASSIC BIT IN LOC 22 SET
863 1215 5252 JMP WICLAS /NOT CLASSIC SYSTEM
864 1216 1776* TAD STOPNT /NO ERROR PRINTING
865 1217 7640 SZA CLA
866 1220 5245 JMP C0D010 /DO NOT PRINT
867
868 1221 4775* /#2 JMS XC0CRLF

```

```

869                                     /#2
870 1222 4774* JMS XC8PNT
871 1223 1263 ERRMES /PRINT THE ERROR MESSAGE
872 1224 4774* JMS XC8PNT
873 1225 1273 MESPC /PRINT THE PC STATEMENT
874 1226 1313 TAD PCSAVE
875 1227 4773* JMS XC8OCTA /CONVERT 4 DIGIT PC TO ASCII
876 1230 4774* JMS XC8PNT
877 1231 1276 MESAC /PRINT THE AC MESS
878 1232 1314 TAD ACSAVE
879 1233 4773* JMS XC8OCTA
880 1234 4774* JMS XC8PNT
881 1235 1301 MESMQ /PRINT MQ
882 1236 1315 TAD MQSAVE
883 1237 4773* JMS XC8OCTA
884 1240 4774* JMS XC8PNT
885 1241 1304 MESFL /PRINT FL
886 1242 1316 TAD FLSAVE
887 1243 4773* JMS XC8OCTA
888 1244 4775* JMS XC8CRLF
889 1245 4772* C8D010, JMS XC8SW /CHECK SWITCH REGISTER
890
891 1246 7710 /#1 SPA CLA /SKIP IF BIT 0 SET
892
893 /#1
894 1247 5261 JMP C8BY2 /LEAVE
895 1250 4771* JMS XC8INQU /GO TO THE INQUIRE ROUTINE
896 1251 5261 JMP C8BY2 /LEAVE
897 1252 4772* NTCLAS, JMS XC8SW /CHECK PSEUDO SWITCH REGISTER
898 /CHECK THE C8SWIT REGISTER
899 /#1
900 1253 7710 SPA CLA /SKIP IF HALT
901 /#1
902 1254 5600 JMP I XC8ERR /NO HALT CONTINUE
903 1255 1370 TAD (7402 /CODE FOP HLT
904 1256 3713 DCA I PCSAVE /PUT IT IN CALL LOC.
905 /#5
906 1257 4767* JMS C8GET
907 /#5
908 1260 5713 JMP I PCSAVE /EXIT TO CALL AND HALT
909 1261 4767* C8BY2, JMS C8GET /GET THE REGISTERS
910 /#3
911 /#3
912 1262 5600 JMP I XC8ERR
1263 0411 ERRMES, TEXT "DIRXAC FAILED "
1264 2230
1265 0103
1266 4040
1267 0601
1270 1114
1271 0504
1272 4000
913 1273 4040 MESPC, TEXT " PC:"
1274 2003
1275 7200
914 1276 4040 MESAC, TEXT " AC:"
    
```

```

1277 0103
1300 7200
915 1301 4040 MESMQ, TEXT " MQ:"
1302 1521
1303 7200
916 1304 4040 MESFL, TEXT " FL:"
1305 0614
1306 7200
917 1307 2701 WATMES, TEXT"WAITING" /TEXT FOR INQU ROUTINE NEED SPACE
1310 1124
1311 1116
1312 0700
918 1313 7777 PCSAVE, 7777
919 1314 7777 ACSAVE, 7777
920 1315 7777 MQSAVE, 7777
921 1316 7777 FLSAVE, 7777
922
923
924 /C8PAUS
925 /THIS ROUTINE WILL CHECK IF THE CONSOL PACKAGE IS ACTIVE,IF ACTIVE
926 /IT WILL RETURN TO CALL PLUS ONE AC= 0, AND DO THAT INSTRUCTION.
927 /IF THE CONSOL PACKAGE IS NOT ACTIVE THE CALL WILL BE REPLACED
928 /WITH A7402 HALT AND THEN RETURN TO THE HALT.
929
930 /
931 C8PAUS= JMS XC8PAU
932 /
933 /EX. C8PAUS /CHECK IF ON ACTIVE CONSOL IF NOT HALT HERE
934 / ANYTHING /RETURN HERE IF ON ACTIVE CONSOL
935 /
936 /
937 /
938
939 1317 0000 XC8PAU, 0
940 1320 4330 JMS C8CK22 /CHECK IF CONSOLE
941 1321 5327 JMP C8D03 /GO DO CONSOL PART RETURN CALL +1
942 1322 7040 CMA /PUT HLT IN CALL
943 1323 1317 TAD XC8PAU /GET CORRECT RETURN ADDR
944 1324 3317 DCA XC8PAU /SET UP RETURN
945 1325 1766* TAD HLT /GET CODE FOR HLT
946 1326 3717 DCA I XC8PAU /PUT HALT IN CALL LOCATION
947 1327 5717 C8D03, JMP I XC8PAU /GO TO HALT OR RETURN TO NEXT LOCATION
948 /*****
949 /CHECK LOCATION 22 FIELD 0
950
951
952 1330 0000 C8CK22, 0
953 1331 7200 CLA
954 1332 6201 CDF 00
955 1333 1765 TAD I (22 /GET LOC 22 FIELD 0
956 1334 6211 CDF 10
957 1335 7650 SNA CLA
958 1336 2330 ISZ C8CK22
959 1337 5730 JMP I C8CK22 /EXIT
960
    
```

```

961
962
963
964 1340 0000  XXC8CNTR, 0
965 1341 4777* JMS XC8CNTR /CHECK CHAR STILL IN BUFFER FOR ANY CONTROL FUNCTION
966 1342 7000 NOP
967 1343 6203 CDF CIF 00
968 1344 5740 JMP I XXC8CNTR /RETURN TO FIELD 0
969
970
971
972 /*****
973 /CALL FROM FIELD 0 FOR WAITING MESSAGE WITH RETURN TO FLD 0
974
975 1345 0000  XXC8INQU, 0
976 1346 4771* JMS XC8INQU /PRINT WAITING RETURN TO FLD 00
977 1347 6203 CDF CIF 00
978 1350 5745 JMP I XXC8INQU /EXIT TO FIELD 00
979
980
981 /*****
982 /CALL FROM FIELD 0 FOR PRINT SWIT RG QUESTION RETURN TO FLD 0
983 1351 0000  XXC8PSW, 0
984 1352 4764* JMS XC8PSW /PRINT PSEUDO SW REG
985 1353 6203 CDF CIF 00 /RETURN TO FIELD 00
986 1354 5751 JMP I XXC8PSW /EXIT TO FIELD 00
987
988
989 /*****
990 /CHECK THE PSEUDO OR HARDWRE SWITCHES
991 1355 0000  XXC8SW, 0
992 1356 4772* JMS XC8SW /CHECK THE SWITCHES
993 1357 6203 CDF CIF 00 /RETURN TO FIELD 00
994 1300 5755 JMP I XXC8SW /TO FIELD 00
995
996 /
997 /LOOK IS RIUTINE TO WAIT FOR A TTY CHAR CHECK IT FOR CONTROL VALUE
998
999 1364 0651
1000 1365 0022
1001 1366 4461
1002 1367 0624
1003 1370 7402
1004 1371 0635
1005 1372 0267
1006 1373 1007
1007 1374 0317
1008 1375 1032
1009 1376 0353
1010 1377 0400
1400 1400 PAGE
1011 1400 0000  XC8LOOK, 0
1012 1401 4777* JMS XC8TTY /GET THE CHAR
1013 1402 4776* JMS XC8CNTR /CHECK FOR CONTROL

```

```

1015 1403 7410 SKP /IS A CONTINUE CHAR
1016 1404 5207 JMP REASK1 /NOT A CONTROL CHAR
1017 1405 6203 EXTLOOK, CDF CIF 00 /EXIT LOOK TO FIELD 0
1018 1406 5600 JMP I XC8LOOK /EXIT LOOK
1019 1407 4775* REASK1, JMS XC8INQU
1020 1410 5205 JMP EXTLOOK /GOT CONTINUE CHAR
1021
1022
1023 1575 0635
1024 1576 0400
1025 1577 0306
0000 FIELD 0

```

```

0000
0100

0200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0300 11111111 11111111 11111111 11111111 11111111 11110001 11111111 11111111

0400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0500 11111111 11111111 11111111 11111111 11111000 00000000 11111111 11111111

0600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0700 11111111 11111111 11111111 11111111 11111110 01111111 11111111 11111111

1000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1100 11111111 11111111 11111111 11111110 00000000 00000000 00111111 11111111

1200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1300 11111111 11111111 11111111 11111111 11111111 11111111 10001111 11111111

1400 11111111 10000000 00000000 00000000 00000000 00000000 00000000 00000000
1500 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000111

1600
1700

2000
2100

2200
2300

2400
2500

2600
2700

3000
3100

3200
3300

3400
3500

3600
3700

```

```

4000
4100

4200
4300

4400
4500

4600
4700

5000
5100

5200
5300

5400
5500

5600
5700

6000
6100

6200
6300

6400
6500

6600
6700

7000
7100

7200
7300

7400
7500

7600
7700

```

```

1026
1027
1028
1029
1030 / RX8/RX01 DIAGNOSTIC * MAINDEC-00-DIRXA-C
1031 /
1032 /STARTING ADDRESS 200 = (AC) = STARTUP SWITCHES
1033 /
1034 /STARTING ADDRESS 201 = RESTART (PARAMETERS ALREADY SELECTED AT START 200)
1035 /
1036 /STARTING ADDRESS 202 = CHANGE DEVICE CODES.
1037 /
1038 /START-UP AC SWITCH DEFINITIONS (WHEN THE STARTING ADDRESS IS 200)
1039 /
1040 / AC 0 - (0) TEST DISKETTE DRIVE 0
1041 / AC 1 - (0) TEST DISKETTE DRIVE 1
1042 / AC 2 - (0) RX01 CONTROL CABLED TO RX8
1043 / AC 3 -
1044 / AC 4 -
1045 / AC 5 -
1046 / AC 6 -
1047 / AC 7 - T
1048 / AC 8 - T (FOR NORMAL PROGRAM)
1049 / AC 9 - T (OPERATION, POSITION)
1050 / AC 10- T (AC SWITCHES 7 THRU 11 = 0)
1051 / AC 11- T
1052 /
1053 /THE PRETEST IS NOT SWITCH SELECTABLE - THIS PRETEST IS ALWAYS
1054 /EXECUTED PRIOR TO THE FIRST TTTT SELECTION
1055 /
1056 /PRETEST = VERIFICATION OF INIT [KEY]
1057 /
1058 / (PRETEST) IB = INIT PART I [KEY] / FLAG DETECTION PART I
1059 /
1060 /INTERFACE / CONTROL TESTS
1061 /
1062 /NOTE: * THE KEYBOARD IS " ALIVE " DURING TESTS 6, 7, 10, AND 11, AND 12
1063 /IF A CHARACTER IS STRUCK AN ERROR MAY OCCUR
1064 /
1065 / TTTT
1066 /
1067 / 0 I = FLAG DETECTION PART II / " C " LINES PART I
1068 / 1 IB = DIRECTION OF IOT XDR PART I / IOT DECODING PART I
1069 / = " C " LINES PART II
1070 / 2 IB = FLAG DETECTION PART II / " C " LINES PART III
1071 / 3 IB = IOT DEVICE CODE VERIFICATION
1072 / 4 IB = DIRECTION OF IOT XDR PART II / " C " LINES PART IV
1073 / 5 I = IOT DECODING PART II
1074 / * 6 IB/I = INTERRUPT TESTING PART I / IOT DECODING PART III
1075 / * 7 IB = INTERRUPT TESTING PART II
1076 / * 10 IB = INTERRUPT TESTING PART III
1077 / * 11 I = INTERRUPT TESTING PART IV
1078 / * 12 IB/I = INIT PART II [PROGRAMMED] / INTERRUPT TEST PART V
1079 /RX01 CONTROL TESTS
1080 /

```

```

1081 / 13 C = FILL BUFFER 12-BIT MODE
1082 / 14 C = EMPTY BUFFER 12-BIT MODE
1083 / 15 C = VERIFICATION OF PREVIOUS TEST
1084 / 16 C = FILL BUFFER 8-BIT MODE
1085 / 17 C = EMPTY BUFFER 8-BIT MODE
1086 / 20 C = VERIFICATION OF PREVIOUS TEST
1087 / 21 C = FILL BUFFER 8-BIT MODE (ALL 0'S)
1088 / 22 C = FILL BUFFER 8-BIT MODE (ALL 1'S)
1089 /
1090 /DISKETTE DRIVE TESTS
1091 /
1092 / 23 D = STATUS BIT "DRIVE READY"
1093 / 24 D = B-CODE VERIFICATION (70) PART I
1094 / 25 D = B-CODE VERIFICATION (70) PART II
1095 / 26 D = B-CODE VERIFICATION (70) PART III
1096 / 27 D = B-CODE VERIFICATION (40) PART IV
1097 / 30 D = SEEK AND CRC VERIFICATION
1098 / 31 D = WRITE TEST
1099 / 32 D = INIT PART III [PROGRAMMED] IMPLIED HEAD TRACK 1 SECTOR 1
1100 / 33 D = READ TEST
1101 / 34 D = WRITE-READ-PROGRAM VERIFY 12 BIT MODE
1102 / 35 D = WRITE-READ-PROGRAM VERIFY 8 BIT MODE
1103 / 36 D = WRITE-READ-PROGRAM VERIFY 12 BIT MODE WITH DELETED DATA
1104 / 37 D = WRITE-READ-PROGRAM VERIFY 8 BIT MODE WITH DELETED DATA
1105 /
1106 / I - MEANS RX8 INTERFACE TEST
1107 / B - MEANS RX01 MAY BE CABLED TO RX8
1108 / C - MEANS AN RX01 MUST BE CABLED TO THE RX8
1109 / D - MEANS A DRIVE MUST BE READY
1110 /
1111 /OPERATIONAL AC SWITCH DEFINITIONS
1112 /
1113 / AC 0 - (1) HALT ON DETECTION OF ERROR
1114 / AC 1 - (1) HALT AT END OF PASS
1115 / AC 2 -
1116 / AC 3 - PRINT AN ERROR MESSAGE
1117 / AC 4 - (1) LOCK SCOPE LOOP ON ERROR
1118 / AC 5 - (1) LOCK SCOPE LOOP ON TEST
1119 / AC 6 - (1) DISABLE THE ISSUING OF [INIT]
1120 / AC 7 - (1) HALT AT END OF A TEST
1121 / AC 8 -
1122 / AC 9 -
1123 / AC 10-
1124 / AC 11- (1) DISABLE RINGING OF BELL AT ERROR
1125 /THE FOLLOWING MAP IS A SUMMARY OF ALL ERRORS.
1126 /
1127 /
1128 / ERA TEST BLANK EAC GOOD COMMENT:
1129 /
1130 /
1131 1600 0410 E0PRE / PRETEST - - - UNEXPECTED TRANSFER REQUEST FLAG
1132 1601 0413 E2PRE / - - - UNEXPECTED ERROR FLAG
1133 1602 0423 E3PRE / - - - MISSING DONE FLAG
1134 1603 0443 E1PRE / STATUS STATUS
1135 / FROM MINUS 4, STATUS NOT = INIT DONE, OR

```



```

1136 / / / TR DEL 204 DRIVE READY + INIT DONE
1137 / / / DATA
1138 1604 0446 E4PE / - - - UNEXPECTED DONE FLAG
1139 / / /
1140 / / /
1141 1605 0620 E0 / T0 - X 0 IOT 67X1 DIDN'T CLEAR AC
1142 1606 0624 E1 / - - - UNEXPECTED TRANSFER REQUEST FLAG
1143 1607 0630 E2 / - - - UNEXPECTED ERROR FLAG
1144 1610 0634 E3 / - - - UNEXPECTED DONE FLAG
1145 / / /
1146 / / /
1147 / / /
1148 1611 0660 E11 / T1 - X 0 IOT 67X1 FAILED TO CLEAR AC
1149 E10 / - X 200 TR NOT = 200 (DRIVE READY)
1150 / / /
1151 / / /
1152 1612 0671 E20 / T2 - - - MISSING DONE FLAG
1153 1613 0674 E21 / - - - MISSING TR FLAG
1154 1614 0677 E22 / - - - MISSING ERROR FLAG
1155 1615 0702 E23 / - - - MISSING DONE FLAG
1156 1616 0705 E24 / - - - MISSING TR FLAG
1157 1617 0715 E25 / - - - MISSING ERROR FLAG
1158 1620 0723 E26 / - X 7777 IOT 67X6 CLEARED AC
1159 1621 0731 E27 / - X 7777 IOT 67X3 CLEARED AC
1160 E28 / - X 7777 IOT 67X4 CLEARED AC
1161 / / /
1162 / / /
1163 1622 0755 E30 / T3 - X 67X5 (EAC) = ILLEGAL DEVICE CODE
1164 / / /
1165 / / /
1166 / / /
1167 1623 1021 E42 / T4 7776 X 200 IOT 67X1 CLEARED AC
1168 / / /
1169 / / /
1170 1624 1044 E40 / T0 376
1171 / / / 375 DATA TO TR NOT =
1172 / / / 373 DATA FROM TR
1173 1625 1056 E41 / 7677 367 OR,
1174 / / / 357 DATA FROM TR NOT =
1175 / / / 337 DATA FROM TR PREVIOUSLY
1176 / / / 7677
1177 1626 1076 E56 / T5 - X 0 (TR) NOT = 0
1178 1627 1101 E50 / - - - MISSING DONE FLAG
1179 1630 1105 E53 / - - - UNEXPECTED DONE FLAG
1180 1631 1110 E51 / - - - MISSING TR FLAG
1181 1632 1114 E54 / - - - UNEXPECTED TR FLAG
1182 1633 1117 E52 / - - - MISSING ERROR FLAG
1183 1634 1123 E55 / - - - UNEXPECTED ERROR FLAG
1184 / / /
1185 / / /
1186 / / /
1187 1635 1137 E60 / T6 - - - UNEXPECTED RX#1 IRQ
1188 1636 1147 E61 / - - - MISSING DONE FLAG
1189 1637 1152 E62 / - - - MISSING TR FLAG
1190 1640 1155 E63 / - - - MISSING ERROR FLAG

```

```

1191 / / /
1192 / / /
1193 / / /
1194 1641 1213 E70 / T7 - - - MISSING RX#1 IRQ
1195 / / /
1196 / / /
1197 / / /
1198 1642 1227 E100 / T10 - - - UNEXPECTED RX#1 IRQ#
1199 / / /
1200 / / /
1201 / / /
1202 1643 1253 E110 / T11 - - - UNEXPECTED RX#1 IRQ
1203 / / /
1204 / / /
1205 / / /
1206 1644 1276 E124 / T12 - - - UNEXPECTED RX#1 IRQ
1207 1645 1310 E120 / - - - UNEXPECTED DONE FLAG
1208 1646 1314 E121 / - - - UNEXPECTED TR FLAG
1209 1647 1320 E122 / - - - UNEXPECTED ERROR FLAG
1210 1650 1325 E123 / - X 0 (TR) NOT = 0
1211 1651 1411 EA120 / ALT 12 - - - UNEXPECTED TR FLAG
1212 1652 1415 EA121 / - - - UNEXPECTED DONE FLAG#
1213 1653 1437 EA122 / ACTUAL ACTUAL 4 OR
1214 / / STATUS MINUS 204 STATUS NOT = 4, OR 204
1215 / / DEL DAT
1216 1654 1450 EA123 / - X 0 THE B-CODE NOT = 0
1217 /THE "XRSTB" SUBROUTINE WHICH READS THE B-CODE STATUS BY ISSUING
1218 /COMMAND # 7 IS ENTERED FROM TESTS: *** ALT12, (T24, T25, T26), AND T27
1219 /
1220 /THE CONTENTS OF "BLANK" = THE CONTENTS OF GOOD FROM TEST: ALT12
1221 /
1222 1655 2405 E7000 / 4/204 - - UNEXPECTED TR FLAG
1223 1656 2416 E7001 / 4/204 STATUS - UNEXPECTED ERROR FLAG
1224 1657 2430 E7002 / 4/204 X 7000 SHIFT REGISTER NOT SHIFTING
1225 / / /
1226 / / /
1227 / / /
1228 /THE "XRST" SUBROUTINE TO READ THE STATUS REGISTER BY ISSUING COMMAND # 5
1229 /IS ENTERED FROM TESTS: *** ALT12, AND T23
1230 /
1231 1660 2444 E7003 / 4/204 - - UNEXPECTED TR FLAG
1232 1661 2455 E7004 / 4/204 STATUS - UNEXPECTED ERROR FLAG
1233 1662 2467 E7006 / 4/204 STATUS 5000 SHIFT REGISTER NOT SHIFTING
1234 1663 2505 E7005 / 4/204 STATUS 0/200
1235 / / /
1236 / / /
1237 / / /
1238 /SUBROUTINE "FBEB" TO FILL AND EMPTY THE BUFFER IS ENTERED FROM TESTS:
1239 / T13, T16 (FILL THE BUFFER) / T14, T17 (EMPTY THE BUFFER)
1240 /
1241 1664 1512 E130 / COMMAND - - UNEXPECTED ERROR FLAG
1242 1665 1525 E131 / COMMAND ACTUAL EXPECT # OF TR FLAGS NOT OK
1243 / (EAC) = # OF FLAGS
1244 / NEG. # MEANS NOT ENOUGH
1245 / >0 MEANS TO MANY

```

```

1246 /
1247 /
1248 /
1249 /ERROR # 140 MAY OCCUR WITHIN TESTS T14, T17, T15, AND T20
1250 /
1251 /THE CONTENTS OF " BLANK " IS EQUIVALENT TO THE WORD/BYTE COUNT AT THE ERROR
1252 /
1253 /THE CONTENTS OF THE " EAC " IS EQUIVALENT TO THE ACTUAL DATA FROM THE SECTOR
1254 /BUFFER (8 OR 12-HIT MODE)
1255 /
1256 /THE CONTENTS OF " GOOD " IS EQUIVALENT TO THE EXPECTED CONTENTS OF THE
1257 /SECTOR BUFFER
1258 /
1259 1666 1661 E140 / * # ACTUAL EXPECT DATA COMPARISON ERROR
1260 / " FB128BYTES " IS A SUBROUTINE WHICH FILLS THE SECTOR BUFFER WITH 128 BYTES
1261 /OF DATA (ALL 1'S OR ALL 0'S) AND IS ENTERED FROM TESTS T21, AND T22
1262 /
1263 1667 4557 E210 / * - - - UNEXPECTED ERROR FLAG
1264 /
1265 /
1266 /SUBROUTINE " IX " EMPTIES THE SECTOR BUFFER AND COMPARES THE DATA TO AN
1267 /EXPECTED PATTERN
1268 /
1269 /THIS SUBROUTINE IS ENTERED FROM TESTS: *** T21, T22, (T24, T25, T26), AND T27
1270 /
1271 1670 2320 E211 / # ACAUAL EXPECT DATA COMPARISON ERROR
1272 1671 2332 E212 / - - - UNEXPECTED ERROR FLAG
1273 /
1274 /
1275 /
1276 /ERRORS E240, E245, E241, AND E242 MAY OCCUR WITHIN TESTS: T24, T25, T26
1277 /
1278 /
1279 /
1280 1672 2030 E240 / * CMND X 2 # OF TR FLAGS NOT OK
1281 1673 2033 E245 / CMND - - MISSING ERROR FLAG
1282 1674 2067 E241 / ACTUAL X T24/200,300
1283 / T25/200
1284 / T26/300
1285 1675 2100 E242 / - X 70 B-CODE NOT = 70
1286 /
1287 /
1288 /
1289 1676 2134 E270 / T27 115 # OF 2 # OF TR FLAGS NOT OK
1290 / XFERS
1291 1677 2137 E271 / - - - MISSING ERROR FLAG
1292 1700 2150 E272 / - X 40 B-CODE NOT = 40
1293 0000 0000 *0
1294 0000 0303 303 / C IS THE REVFSION
1295 0001 0001 *1
1296 0001 5402 JMP I IPI
1297 /
1298 /PROGRAM LOCATION 2 CONTAINS THE INTERRUPT RETURN ADDRESS
1299 /
1300 /PROGRAM LOCATION 2 IS MODIFIED WITHIN CERTAIN TESTS

```

```

1301 /
1302 0002 5405 IPI, PI
1303 /
1304 0005 *5
1305 /*****
1306 /CONSOLE
1307 /*****
1308 /
1309 4405 C8LOOK= JMS I .
1310 0005 1400 XC8LOOK /WAIT FOR TTY INPUT AND THEN RETURN
1311 /
1312 /*****
1313 /
1314 /AUTO INDEX REGISTER DEFINITION.
1315 /
1316 0010 *10
1317 0010 0000 A10, 0
1318 0011 0000 A11, 0
1319 0012 0000 A12, 0
1320 0013 0000 A13, 0 /TEST 0
1321 0020 *20
1322 /
1323 /THE FOLLOWING PROGRAM LOCATIONS (20, 21, AND 22, 23) ARE RESERVED FOR ACT8/A
1324 /
1325 0020 0500 0500 /SET FOR DRIVES 0 AND 1 DEVICE CODE 75
1326 0021 4000 4000 /0000=PSEUDO SWITCH REGISTER IF ON ACTIVE CONSOLE
1327 /4000=USE HARDWARE SWITCH REGISTER
1328 /
1329 /
1330 0022 0000 0000 /0000=NOT ACTIVE CONSOLE PACKAGE
1331 /0400= ACTIVE CONSOLE PACKAGE
1332 0024 *24
1333 /
1334 /*****
1335 /
1336 /THE FOLLOWING CALLS ARE USED FOR THE CONSOLE PACKAGE
1337 /
1338 4424 CHECKC0= JMS I .
1339 0024 4516 XCHECK /USED TO CHECK IF THE CONSOLE IS ACTIVE
1340 /
1341 4425 C8PASS= JMS I .
1342 0025 0200 XC8PASS /END OF PASS FOR CONSOL
1343 /
1344 4426 C8SWIT= JMS I .
1345 0026 1351 XXC8PSW /ASK THE PSEUDO SWITCH QUESTION
1346 /
1347 4427 C8INQU= JMS I .
1348 0027 1345 XXC8INQU /PRINT WAITING
1349 /
1350 4430 C8CKSW= JMS I .
1351 0030 1355 XXC8SW /CHECK IF USING HARDWARE SWIT REG
1352 /*****
1353 /
1354 /THE FOLLOWING PROGRAM LOCATIONS "OD", "ID", "FIRST", AND "LAST" MAY BE
1355 /CHANGED BY THE OPERATOR MANUALLY HOWEVER FOLLOWING THESE RESTRICTIONS.

```

```

1356 /
1357 /
1358 / 1. THE CONTENTS OF "OD" (MIN VAL 0) MUST BE <= THE
1359 / CONTENTS OF "ID" (MAX VAL 114).
1360 /
1361 / 2. THE CONTENTS OF "FIRST" (MIN VAL 1) MUST BE <= THE
1362 / CONTENTS OF "LAST" (MAX VAL 32)
1363 /
1364 /THE PROGRAM INITIALLY SETS THESE VALUES AT PROGRAM LOAD TIME
1365 /
1366 / (OD) = 52, AND (ID) = 53
1367 /
1368 /BECAUSE TRACK 53 IS THE TRACK AT WHICH THE RX01 MICROCONTROLLER WILL
1369 /DECREASE THE WRITE CURRENT IN HALF
1370 /
1371 0031 0001 OD, 1 /OUTSIDE DIAMETER (MIN VALUE 0)
1372 0032 0114 ID, 114 /INSIDE DIAMETER (MAX VALUE 114)
1373 0033 0001 FIRST, 1 /FIRST SECTOR TO ACCESS (MIN VAL 1)
1374 0034 0032 LAST, 32 /LAST SECTOR TO ACCESS (MAX VAL 32)
1375 /
1376 /PDP-8/E AUGMENTED INSTRUCTIONS
1377 /
1378 0035 2612 BSW=JMS I . XBSW /BYTE SWAP
1379 6001 ION=6001
1380 6002 IOF=6002
1381 4570 LAS=CKSWIT
1382 7501 MQA=7501 /"OR" (MQ) WITH (AC)
1383 7421 MQL=7421 /MQ=AC (THEN CLEAR AC)
1384 /
1385 /DISKETTE IOT SUBROUTINES
1386 /
1387 4436 TYBOCT=JMS I .
1388 0036 5200 XTYBOCT /TYPE EIGHT OCTAL DIGITS.
1389 4437 LCD=JMS I .
1390 0037 6200 XLCD
1391 4440 LCDA=JMS I .
1392 0040 6205 XLCDA
1393 4441 LCDH=JMS I .
1394 0041 6232 XLCDB
1395 4442 XDRIN=JMS I .
1396 0042 6400 XXDRIN
1397 4443 XDROUT=JMS I .
1398 0043 6403 XXDROUT
1399 4444 STR=JMS I .
1400 0044 6407 XSTR
1401 4445 SER=JMS I .
1402 0045 6414 XSER
1403 4446 SDN=JMS I .
1404 0046 6121 XSDN
1405 4447 INTR=JMS I .
1406 0047 6426 XINTR
1407 4450 INIT=JMS I .
1408 0050 6434 XINIT
1409 4451 INITB=JMS I .
1410 0051 6447 XINITB
    
```

```

1411 /OPERATING SYSTEM SUBROUTINES
1412 /
1413 4452 ERROR=JMS I .
1414 0052 2637 XERROR
1415 5453 EXIT=JMP I .
1416 0053 0450 MORETESTS /EXIT FROM A TEST (IF RX8 ONLY)
1417 4354 DONE=JMS I .
1418 0054 4243 XDONE / FORM: "DONE"; NO; YES"
1419 4455 GETAPATTERN=JMS I .
1420 0055 3035 XGETAPATTERN
1421 4456 GETASECTOR=JMS I .
1422 0056 4607 XGETASECTOR
1423 4457 GETATRACK=JMS I .
1424 0057 4261 XGETATRACK /GET A TRACK FOR IOT LCD-B (TRACK #)
1425 4160 GETUNIT=JMS I .
1426 0060 4200 XGETUNIT /SELECT A DISKETTE DRIVE
1427 4461 HLT=HALT
1428 4461 HALT=JMS I .
1429 0061 4115 XHALT
1430 4462 INITSECTORS=JMS I .
1431 0062 4600 XINITSECTORS
1432 4463 INITTRACKS=JMS I .
1433 0063 4250 XINITTRACKS
1434 4464 LOCKUP=JMS I .
1435 0064 2600 XLOCKUP
1436 5453 NOTEST=EXIT
1437 4465 OK=JMS I .
1438 0065 6123 XOK /SKIP IF NOT ON APT.
1439 4466 TICK=JMS I .
1440 0066 4127 XTICK
1441 4467 AEHROR=JMS I .
1442 0067 4346 XAEHRRO
1443 4470 APT0=JMS I .
1444 0070 1546 XAPT0
1445 4471 WAIT=JMS I .
1446 0071 1722 XWAIT
1447 4472 CHEK22=JMS I .
1448 0072 2541 XCHK22 /CHECK FOR APT SYSTEM.
1449 4473 PRINT=JMS I .
1450 0073 5244 XPRINT /PRINT A MESSAGE; FORM: "PRINT; MESSAGE"
1451 4474 READ=JMS I .
1452 0074 3414 XREAD
1453 4475 READCOMPARE=JMS I .
1454 0075 3410 XREADCOMPARE
1455 4476 RST=JMS I .
1456 0076 2437 XRST /STATUS AFTER "RST" COMMAND (12) / 5 TIMES 2
1457 4477 RSTB=JMS I .
1458 0077 2400 XRSTB /STATUS AFTER READ B-CODES COMMAND (16) 7 X 2
1459 4500 SETUP=JMS I .
1460 0100 4077 XSETUP
1461 4501 SPECIALTYPE=JMS I .
1462 0101 5351 XSPECIALTYPE
1463 4502 SCOPE=JMS I .
1464 0102 1332 XSCOPE
1465 4503 SUBSCOPE=JMS I .
    
```

```

1466 0103 1342 XSSCOF
1467 4504 TAB=JMS I .
1468 0104 5215 XTAB / FORM: " TAB; N "
1469 4505 TY4OCT=JMS I .
1470 0105 5000 XTY4OCT /TYPE (4) OCTAL ; FORM: " TY4OCT; OCTAL "
1471 4506 TYPEIT=JMS I .
1472 0106 5305 XTYPLIT /TYPE 1 8 BIT ASCII.AC =ASCII.
1473 4507 WAITTY=JMS I .
1474 0107 5400 XWAITTY
1475 4510 WRITE=JMS I .
1476 0110 3200 XWRITE
1477 /
1478 /ACCUMULATOR SWITCH REGISTER DEFINITIONS
1479 /
1480 4000 SW0=4000
1481 2000 SW1=2000
1482 1000 SW2=1000
1483 0400 SW3=400
1484 0200 SW4=200
1485 0100 SW5=100
1486 0040 SW6=40
1487 0020 SW7=20
1488 0010 SW8=10
1489 0004 SW9=4
1490 0002 SW10=2
1491 0001 SW11=1
1492 /OPERATING SYSTEM ALLOCATED STORAGE REFERENCES
1493 /
1494 0111 0000 BUSY, 0 / = 1 = PROCESSING AN RX01 PROGRAM INTEPRUPT
1495 0112 0000 COMMAND, 0 /DISKETTE COMMAND ; (AC) AT LCD
1496 0113 0000 COMPERROR, 0 /PROGRAM DATA COMPARE ERRORS
1497 0114 0000 DTESTP, 0 /DIAGNOSTIC TEST PARAMETERS (SELECTED AT L/S )
1498 0115 7777 FIRSTERROR, 7777 / (7777) IF 1ST ERROR ; (0) IF NOT
1499 0116 0000 HANGER, 0 /COUNTER TO DETECT DEVICE TEST HUNG
1500 0117 7777 K7777, -1 /
1501 0120 0000 RDC, 0 / = 0 IF A RDC TEST, = 7777 IF NOT
1502 0121 7765 KRETRY, -13 / 1 ORIGINAL TRY + 10 RETRYS
1503 0122 7746 SECTORS, -32 /NEGATIVE # OF SECTORS PER TRACK (1-32 OCTAL)
1504 0123 0000 SSTART, 0 /SECTOR LAST ACCESSED ( 0 = "HOME" )
1505 0124 0000 STARGET, 0 /TARGET SECTOR OF (UNITX)
1506 0125 0000 START, 0 /TRACK LAST ACCESSED
1507 0126 0000 ASTATUS, 0 /DISKETTE STATUS AT ERROR OR DONE
1508 0127 0000 BSTATUS, 0 /RX01 DEFINITIVE ERROR CODE REGISTER
1509 0130 0000 CSTATUS, 0 /STATUS FROM THE "READ STATUS" COMMAND
1510 0131 0000 TARGET, 0 /TARGET TRACK OF (UNITX)
1511 0132 0000 TESTP, 0 /TEST PARAMETERS (DYNAMIC BY PROGRAM)
1512 0133 0000 XA10, 0
1513 0134 0000 XA11, 0
1514 /
1515 /PROGRAM LOCATION XXX IS A TEMPORARY STORAGE REGISTER FOR DATA
1516 /OR ADDRESSES OF DATA WHICH ARE CALLED WITHIN SUBROUTINES WHICH
1517 /DO NOT CALL SUBROUTINES WHICH CALL THESE STORAGE REGISTERS
1518 /
1519 /XXX IS CALLED WITHIN THE FOLLOWING SUBROUTINES
1520 /

```

```

1521 /D,D/RX-ERROR (2)
1522 /COMPARE (2)
1523 /XGETAPATTERN (6)
1524 /RDORWK (2)
1525 /
1526 0135 0000 XXX, 0
1527 /
1528 0136 7765 RIRETRY, -13 /RERROS /WRITE ERROR RECOVERY
1529 0137 7765 RRETRY, -13 /RERPOS /HEAD ERROR RECOVERY
1530 0140 0000 DNSLOG, 0 /DATAERROS /DATA ERROR WITH CRC STATUS ERROR
1531 0141 0000 DNSLOG, 0 /DNSEPKOS /DATA ERROR BUT NO CRC STATUS ERROR
1532 0142 0000 SNDLOG, 0 /SNDEPKOS /CRC STATUS ERROR BUT NO DATA ERROR
1533 0143 7765 SRETRY, -13 /SERROS /SEFK ERROR RECOVERY
1534 0144 7765 PRETRY, -13 /PERROS /PARITY ERROR RECOVERY
1535 / (TRACKS) ARE SET TO THE NEGATIVE DIFFERENCE BETWEEN (00), AND (10)
1536 / IN THE SUBROUTINE "INITTRACKS "
1537 /
1538 0145 7663 TRACKS, -115 / =# OF TRACKS PER DISKETTE (-115 TO -1 DYNAMIC DECREMENT)
1539 0146 0115 TTRACKS, 115 / # OF TRACKS PER DISKETTE (115 TO 1 STATIC)
1540 0147 0000 XTARGET, 0 / ; (AC) = TRACK+SECTOR AT IOT LCD-H ; DESTINATION
1541 0150 0000 ECOMMAND, 0
1542 0151 0000 HI, 0
1543 0152 0000 GOBIT, 0 / > 0 MEANS EXPECTING AN RX01 PI, <= 0 MEANS NOT EXPECTING
1544 /
1545 /
1546 / (PAT-SUMCHECK) IS A NUMBER GENERATED WITHIN SUBROUTINE " XGETAPATTERN "
1547 /EQUIVALENT TO SUMCHECK OF 60/124 [ 12/8 BIT MODE] DATA WORDS
1548 /
1549 0153 0000 PATSUMCHECK, 0
1550 /
1551 / (WORDX) IS AN ADDRESS WHOSE CONTENTS = : (PAT-SUMCHECK) + (WBUFFER) + (WBUFFER+1)
1552 /
1553 / (WORDY) IS AN ADDRESS FOR WHICH THE CONTENTS REPRESENT THE NEGATIVE-1
1554 /OF 2 TIMES THE CONTENTS OF THE ADDRESS WITHIN PROGRAM LOCATION " WORDX "
1555 /
1556 /THE CONTENTS OF BOTH WORDX AND WORDY ARE GENERATED WITHIN SUBROUTINE " XGETASECTOR "
1557 /
1558 0154 0000 WORDX, 0
1559 0155 0000 WORDY, 0
1560 0156 0000 LSB, 0
1561 0157 0000 MSB, 0
1562 0160 0000 TTYBUSY, 0
1563 0161 0000 PASS, 0
1564 0162 0000 /PASS COUNT TO A MAGNITUDE OF 16777215(10)
1565 0163 0000 RXHERE, 0 / = 1000 IF AN RX01 IS CABLED TO THE RX8
1566 0164 0000 GOOD, 0 /EXPECTED RESULT
1567 0165 0000 EAC, 0 / (AC) AT ERROR "BAD" (ACTUAL RESULT)
1568 0166 0000 BLANK, 0
1569 0167 0611 TEST, T0 /ADDRESS OF STARTING ADDRESS OF TEST
1570 /*****
1571 /ADDITIONAL CALL FOR CONSOLE PACKAGE
1572 /
1573 /
1574 4570 CKSWIT= JMS I .
1575 0170 3400 XCKSWIT /CHECK IF LAS TO USE LOC 22 OR HARDWARE

```

```

1576      4571      C0CNTR= JMS I
1577      0171 1340      XXC0CNTR          /CHECK FOR CONSOLE CONTROL CHAR,
1578      0172 7000      K7000, 7000
1579      0173 0007      K0007, 0007
1580      0174 7777      XCNT, -1
1581      4575      FORCE=JMS I
1582      0175 5513      XXFORCE
1583      /STARTING ADDRESS 200 = (AC) = STARTUP SWITCHES
1584      /
1585      /STARTING ADDRESS 201 = RESTART (PARAMATERS ALREADY SELECTED AT START 200)
1586      /
1587      /
1588      0200      *200
1589      /
1590      /*****
1591      /CONSOLE
1592      /*****
1593      0200 5202      C0STRT, JMP .+2          /NORMAL PROGRAM ACTIVITY.
1594      0201 5203      JMP .+2          /RESTART WITH SAME PARAMETERS.
1595      /THIS SECTION IS NORMAL PROGRAM ACTIVITY.
1596      /DEVICE CODE 75 IS ASSUMED, ANY OTHER WILL CAUSE ERRORS.
1597      0202 7240      STA
1598      0203 3010      DCA A10
1599      0204 3160      DCA TTYBUSY
1600      0205 3111      DCA BUSY
1601      0206 3152      DCA GOBIT
1602      0207 3161      DCA PASS
1603      0210 3162      DCA PASS+1
1604      0211 1377      TAD (-40)
1605      0212 3116      DCA HANGER
1606      0213 3123      DCA SSTART
1607      0214 3131      DCA TARGET
1608      0215 1010      TAD A10
1609      0216 7650      SMA CLA
1610      0217 5230      JMP AROUND          /USE EXISTING PARAMETERS.
1611      /
1612      /*****
1613      0220 4776      JMS PNTID          /PRINT ID AND REMOVE DIAGNOSTIC
1614      /*****
1615      0221 4424      CHECKC0          /CONSOLE ACTIVE
1616      0222 4426      C0SWIT          /ASK SR QUESTION.
1617      0223 5225      JMP .+2
1618      0224 4461      HLT
1619      0225 4570      LAS          /GET PARAMETERS.
1620      0226 3114      DCA DTESTP
1621      0227 4775      JMS CHNDEV          /CHANGE DEVICE CODES.
1622      AROUND, TAD (1000)
1623      AND DTESTP
1624      DCA RXHERE/ = 0 IF RX01 CABLED TO RX8
1625      TAD DTESTP
1626      CMA
1627      AND (S0+SW1)
1628      DCA UNITS          / = UNITS TO TEST
1629      PRINT
1630      0240 6532      MDTESTP          /TEST PARAMATER CONFIRMATION MESSAGE

```

```

1631      0241 4505      TY40CT
1632      0242 0114      DTESTP
1633      /THE PROGRAM WILL VERIFY THAT THE CONTENTS OF PROGRAM LOCATIONS:
1634      /
1635      /
1636      /
1637      /WHICH ARE VARIABLE BY THE USER ARE WITHIN SELECTABLE LIMITS
1638      /
1639      / 0 <= OD <= 114
1640      /
1641      0243 1031      TAD OD
1642      0244 7700      SMA CLA
1643      0245 5250      JMP .+3
1644      0246 1371      TAD (52)
1645      0247 3031      DCA OD
1646      0250 1370      TAD (-114)
1647      0251 1031      TAD OD
1648      0252 7740      SMA SZA CLA
1649      0253 5246      JMP .-5
1650      0254 4473      PRINT
1651      0255 6673      MOD
1652      0256 4505      TY40CT
1653      0257 0031      OD
1654      / 0 <= ID <= OD
1655      /
1656      0260 1032      TAD ID
1657      0261 7700      SMA CLA
1658      0262 5265      JMP .+3
1659      0263 1367      TAD (53)
1660      0264 3032      DCA ID
1661      0265 1032      TAD ID
1662      0266 7041      CIA
1663      0267 1031      TAD OD
1664      0270 7740      SMA SZA CLA
1665      0271 5263      JMP .-6
1666      0272 4473      PRINT
1667      0273 6677      MID
1668      0274 4505      TY40CT
1669      0275 0032      ID
1670      / 0 < (FIRST) <= 32
1671      /
1672      0276 1033      TAD FIRST
1673      0277 7740      SMA SZA CLA
1674      0300 5303      JMP .+3
1675      0301 7301      CLL CLA IAC
1676      0302 3033      DCA FIRST
1677      0303 1033      TAD FIRST
1678      0304 1366      TAD (-32)
1679      0305 7740      SMA SZA CLA
1680      0306 5301      JMP .-5
1681      0307 4473      PRINT
1682      0310 6703      MFIRST
1683      0311 4505      TY40CT
1684      0312 0033      FIRST
1685      0313 5765      JMP TADLAST

```

```

1686 /IF THE TELEPRINTER IS BUSY (TTYBUSY = X), AND
1687 /
1688 /IF A KEYBOARD FLAG HAS OCCURED,
1689 /
1690 /THEN STOP TELEPRINTER OUTPUTS (IF A KRB = <CTRL>S ), OR
1691 /
1692 /THEN RESUME TELEPRINTER OUTPUTS (IF A KRB = <CTRL>Q
1693 /AND A PREVIOUS <CTRL> Q HAD OCCURED)
1694 /
1695 0314 0336 XKCC, KRB /READ THE KEYBOARD BUFFER STATIC
1696 0315 0364 AND (177
1697 0316 1363 TAD (200)
1698 0317 6211 CDF 10 /MAKE IT 8 BIT CODE
1699 0320 3751 DCA I XC8CHAR /STORE IN FIELD ONE CHAR
1700 0321 1751 TAD I XC8CHAR /GET THE CHAR
1701 0322 6201 CDF 0
1702 0323 0364 AND (177)
1703 0324 1362 TAD (-21) / <CTRL>Q
1704 0325 7440 SZA
1705 0326 5335 JMP NOTQ / ?? <CTRL>S
1706 /
1707 /THE KEYBOARD BUFFER STATIC IS <CTRL>Q
1708 /
1709 /RESUME TELEPRINTER OUTPUTS
1710 /
1711 / (IF A PREVIOUS <CTRL>S HAD OCCURED)
1712 /
1713 0327 1160 TAD TTYBUSY / = 4000 IF A PREVIOUS <CTRL>S
1714 0330 7700 SMA CLA
1715 0331 5761* JMP PIEXIT /NO PREVIOUS <CTRL>S
1716 0332 7301 CLL CLA IAC
1717 0333 3160 DCA TTYBUSY / 1
1718 0334 5760* JMP XTCF /RESUME TELEPRINTER OUTPUTS
1719 /
1720 /THE KEYBOARD BUFFER STATIC IS NOT A <CTRL>Q
1721 /
1722 /IF IT IS A <CTRL>S THEN SUSPEND TELEPRINTER OUTPUTS
1723 /
1724 0335 1357 NOTQ, TAD (-2)
1725 0336 7640 SZA CLA
1726 0337 5343 JMP C8TEST /NOT A <CTRL>S EITHER
1727 0340 7330 STL CLA RAR
1728 0341 3160 DCA TTYBUSY / 4000
1729 0342 5761* JMP PIEXIT
1730 /*****
1731 /CONSOLE
1732 /*****
1733 /
1734 0343 4424 C8TEST, CHECKC8 /CONSOLE ACTIVE.
1735 0344 4571 C8CNTP /CHECK CONSOLE CONTROL CHARACTERS
1736 0345 7000 NOP
1737 0346 7200 CLA
1738 0347 3160 DCA TTYBUSY /CLEAR FLAG
1739 0350 5761* JMP PIEXIT /EXIT
1740 0351 1104 XC8CHAR, CHAR /LOC IN FIELD 1

```

```

1741 /*****
1742 /CONSOLE
1743 /*****
1744 /
1745 0357 7776
1746 0360 5343
1747 0361 5476
1748 0362 7757
1749 0363 0200
1750 0364 0177
1751 0365 3273
1752 0366 7746
1753 0367 0053
1754 0370 7664
1755 0371 0052
1756 0372 4235
1757 0373 6000
1758 0374 1000
1759 0375 5067
1760 0376 1530
1761 0377 7740
1762 0400
1763 /
1764 / (A) IF AN RX01 MICROCONTROLLER IS [NOT] CABLED TO THE RX8 INTERFACE,
1765 / THEN ALL FLAGS (DONE, TRANSFER REQUEST, AND ERROR) , AND THE RX8
1766 / INTERFACE TRANSFER REGISTER SHOULD HAVE BEEN CLEARED BY "KEY"
1767 / INITIALIZE (IF THEY WERE EVER SET).
1768 /
1769 / (B) IF AN RX01 MICROCONTROLLER [IS] CABLED TO THE RX8 INTERFACE,
1770 / THEN "KEY" INITIALIZE SHOULD HAVE [SET] THE DONE FLAG BECAUSE
1771 / ANY [INIT] OF THE RX01 MICROCONTROLLER IS AN IMPLIED [READ SECTOR]
1772 / OF TRACK 0 SECTOR 1 (FOR SYSTEMS PROGRAMMING BOOTSTRAP APPLICATIONS).
1773 /
1774 / THEREFORE, ANY ERROR (EXCEPT PARITY) THAT MAY OCCUR FROM A NORMAL
1775 / "READ SECTOR" COMMAND MAY OCCUR HERE CAUSING THE ERROR FLAG TO SET, AND
1776 / DISPLAYING THE ERROR STATUS WITHIN THE TRANSFER REGISTER AT "DONE".
1777 /
1778 / THE TRANSFER REQUEST FLAG SHOULD BE CLEARED.
1779 /
1780 /NOTE:
1781 /
1782 /SCOPE LOOPING IS NOT OFFERED BECAUSE THE "INIT" FUNCTION
1783 /
1784 0400 7240 PRETEST, STA
1785 0401 3115 DCA FIRSTERROR /FOR FIRST ERROR EVER THIS PASS
1786 0402 3777* DCA ERRORS /CLEAR "ERRORS" FOR FIRST "SCOPE" EVER
1787 0403 4502 SCOPE /THIS "SCOPE" TO REFRESH "FAT" IF ERROR
1788 0404 1776* TAD PCSCOPE
1789 0405 3167 DCA TEST /TO REFRESH "FAT" FOR "ERROR"
1790 0406 4444 STR
1791 0407 4465 OK
1792 0410 4452 EOPRE, ERROR /UNEXPECTED TRANSFER REQUEST FLAG
1793 0411 4445 SER
1794 0412 4465 OK

```

```

1795 0413 4452 E2PRE, ERROR /UNEXPECTED ERROR FLAG
1796 /
1797 /*****
1798 /*****
1799 /
1800 /IF AN RX01 MICROCONTROLLER [IS] CABLED TO THE RX8 INTERFACE
1801 /THEN THE DONE FLAG SHOULD BE SET
1802 /
1803 0414 1163 TAD RXHERE
1804 0415 7640 SZA CLA
1805 0416 5227 JMP NORX01 /
1806 0417 4471 WAIT
1807 0420 4446 SDN
1808 0421 5217 JMP .-2 /WAIT FOR DONE FLAG
1809 0422 7410 SKP
1810 0423 4452 E3PRE, ERROR /MISSING DONE FLAG
1811 /
1812 /*****
1813 /*****
1814 /THE ENTIRE STATUS WORD IS DISPLAYED IN THE TRANSFER REGISTER AT ERROR/DONE TIME.
1815 /
1816 /IF AN RX01 CONTROLLER IS CABLED TO THE RX8 INTERFACE
1817 / (AND DRIVE 0 IS READY THEN THE STATUS SHOULD INDICATE " SEL DRV RDY" ), ALSO
1818 /DELETED DATA [MAY] = 1 IF TRACK 0/SECTOR 1 WAS WRITTEN WITH DELETED DATA
1819 /AND "INIT DONE" SHOULD BE SET.
1820 /
1821 0424 5775 JMP TSTUNT /OFF PAGE BECAUSE OF ROOM.
1822 /
1823 /
1824 0425 1374 TAD (40 /PROGRAM EXPECTS DRIVE 0 TO BE READY
1825 /
1826 /
1827 ////////////////////////////////////////////////////////////////////
1828 /
1829 / 4 5 - - 8 9 10 11 /
1830 /
1831 / SEL WRITE INIT PAR /
1832 / DRIVE DD PROTECT [DONE] CRC /
1833 / RDY (N/A) /
1834 /
1835 ////////////////////////////////////////////////////////////////////
1836 /
1837 /
1838 0426 7107 CLL IAC PTL / 4 [INIT] DONE OR 204
1839 0427 3164 NORX01, DCA GOOD
1840 0430 4442 XDRIN
1841 0431 3166 DCA BLANK /ACTUAL STATUS FROM [INIT]
1842 0432 1166 TAD BLANK
1843 0433 0773 AND COMP
1844 0434 3165 DCA EAC /STATUS MINUS DELETED DATA (BIT 5)
1845 0435 1165 TAD EAC
1846 0436 7041 CIA
1847 0437 1164 TAD GOOD /EXPECTED
1848 0440 7650 SNA CLA
1849 0441 5244 JMP .+3 /OK

```

```

1850 0442 1165 TAD EAC
1851 0443 4452 E1PRE, ERROR / [INIT] STATUS NOT = EXPECTED
1852 /
1853 /IF AN RX01 MICROCONTROLLER [IS] CABLED TO THE RX8 INTERFACE
1854 /THEN THE PREVIOUS "SDN" SHOULD HAVE CLEARED THE DONE FLAG, BUT
1855 /
1856 /IF AN RX01 MICROCONTROLLER IS [NOT] CABLED TO THE RX8 INTERFACE
1857 /THEN "KEY" INITIALIZE SHOULD HAVE CLEARED THE DONE FLAG
1858 /
1859 /TECHNICAL NOTE:
1860 /
1861 /IF THE DONE FLAG IS SET, AND IF THE INTERRUPT ENABLE FLIP-FLOP IS SET ILLEGALLY,
1862 /THEN AN "UNEXPECTED RX01 INTERRUPT" WILL OCCUR IN T0 (IF AN RX01 CONTROLLER
1863 / [IS] CABLED TO THE RX8 INTERFACE) OR IN T1 WHEN THE MAINTENANCE FLIP-
1864 /FLOP "SETS ALL FLAGS"
1865 /
1866 0444 4446 SDN
1867 0445 4465 OK
1868 0446 4452 E4PRE, ERROR /UNEXPECTED DONE FLAG
1869 /
1870 /END OF PRE-TEST /END OF PRETEST
1871 /
1872 0447 5345 JMP REBEGIN
1873 /
1874 0450 4404 MORETESTS, LOCKUP
1875 0451 3777 FIRSTTEST, DCA ERRORS
1876 0452 1413 TAD I A13 / FAT (FIRST ADDRESS OF TEST)
1877 0453 3167 DCA TEST
1878 0454 1167 TAD TEST / EQUIVALENT TO " SCOPE "
1879 0455 3776 DCA PCSCOPE /FOR FIRST ENTRY INTO XGETUNIT THIS TEST
1880 0456 3772 DCA WUNITS
1881 0457 5567 JMP I TEST
1882 /
1883 /TESTS
1884 /
1885 0460 0611 TESTS, T0
1886 0461 0637 T1
1887 0462 0663 T2
1888 0463 0734 T3
1889 0464 1000 T4
1890 0465 1065 T5
1891 0466 1126 T6
1892 0467 1200 T7
1893 0470 1216 T10
1894 0471 1232 T11
1895 0472 1261 T12
1896 0473 1455 T13
1897 0474 1603 T14
1898 0475 1601 T15
1899 0476 1454 T16
1900 0477 1602 T17
1901 0500 1600 T20
1902 0501 1675 T21
1903 0502 1674 T22
1904 0503 2265 T23

```

```

1905 0504 1714 T24
1906 0505 1716 T25
1907 0506 1720 T26
1908 0507 2106 T27
1909 0510 2200 T30
1910 0511 2202 T31
1911 0512 2223 T32
1912 0513 2241 T33
1913 0514 2300 T34
1914 0515 2276 T35
1915 0516 2277 T36
1916 0517 2275 T37
1917 0520 0521 NOMORE-1
1918 /THERE ARE NO MORE TESTS
1919 /
1920 /PRINT AN END OF PASS INDICATOR
1921 /
1922 / A - INTERFACE TEST OK (ONLY RX8 TO TEST)
1923 / C - RX8 AND RX01 TEST OK
1924 / D - RX8 AND RX01 AND DRIVE TESTING OK
1925 /
1926 / - - AN ERROR OCCURED (DURING A, B, OR D)
1927 /
1928 0400 XD=0400
1929 0521 1371 TAD (XD)
1930 0522 3356 NOMORETESTS, DCA MX / (XI), (XC), (XD), OR 0
1931 /
1932 /NOTE:IF THE CONTENTS OF PROGRAM LOCATION FIRSTERROR = 0
1933 /THEN AN ERROR HAS OCCURED FOR THIS PASS
1934 /
1935 0523 1115 TAD FIRSTERROR
1936 0524 7640 SZA CLA
1937 0525 5330 JMP .+3
1938 0526 1370 TAD (5500)
1939 0527 3356 DCA MX / -
1940 /*****
1941 /CONSOLE
1942 /*****
1943 0530 4424 CHECKC8
1944 0531 4430 C8CKSW
1945 0532 5767* JMP C8RET2 /IF ACTIVE CONSOL DO C8RET2
1946 /*****
1947 0533 4473 PRINT
1948 0534 0556 MX
1949 0535 2161 ISZ PASS
1950 0536 5341 JMP .+3
1951 0537 2162 ISZ PASS+1
1952 0540 7000 NOP
1953 0541 4570 LAS
1954 0542 0366 AND (SW1)
1955 0543 7640 SZA CLA
1956 0544 4461 HLT
1957 0545 1114 REBEGIN, TAD DTESTP
1958 0546 0365 AND (37)
1959 0547 1364 TAD (TESTS=1)

```

```

1960 0550 3013 DCA A13
1961 0551 7240 STA
1962 0552 3115 DCA FIRSTERROR /FIRST ERROR SWITCH FOR EACH PASS
1963 0553 7340 CLL CLA CMA
1964 0554 3763* DCA CLKCNT /FOR APT TIMING
1965 0555 5251 JMP FIRSTTEST
1966 /
1967 0556 1100 MX, TEXT "I" / I, C, OR D
1968 0563 4145
1969 0564 0457
1970 0565 0037
1971 0566 2000
1972 0567 0600
1973 0570 5500
1974 0571 0400
1975 0572 4236
1976 0573 2756
1977 0574 0040
1978 0575 2744
1979 0576 1364
1980 0577 1363
1981 0600 PAGE
1982 /*****
1983 /ROUTINE FOR CONSOLE PASS
1984
1985 0600 4424 C8RET2,CHECKC8
1986 0601 4425 C8PASS
1987 0602 4461 HLT
1988 0603 6001 ION /CONSOLE PASS
1989 0604 2161 ISZ PASS
1990 0605 5210 JMP .+3
1991 0606 2162 ISZ PASS+1
1992
1993 0607 7000 NOP
1994 0610 5777* JMP REBEGIN /CONTINUE WITH PROGRAM RETURN
1995 /;*****
1996 /TEST 0 - FLAG DETECTION PART II / " C " LINES VERIFICATION PART I
1997 /
1998 /*****
1999 /*****
2000 /
2001 /IF AN RX01 MICROCONTROLLER IS CABLED TO THE RX8 INTERFACE
2002 /THEN DON'T EXECUTE THIS TEST
2003 /BECAUSE ISSUING THE IOT LCD WITH THE AC = 177
2004 /RESEMBLED A COMMAND TO THE RX01
2005 /
2006 0611 1163 T0, TAD RXHERE
2007 0612 7650 SNA CLA
2008 0613 5453 NOTEST
2009 0614 3164 DCA GOOD
2010 /*****
2011 /*****
2012 /
2013 /THE PURPOSE OF THIS TEST IS TO VERIFY THAT THE LCD (LOAD COMMAND REGISTER)

```



```

2014 /IOT 67X1 DOES [NOT] SET THE MAINTENANCE FLIP-FLOP WHEN THE CONTENTS
2015 /OF THE AC = 177 AT THE TIME THE LCD IOT IS ISSUED.
2016 /
2017 /
2018 /TECHNICAL NOTE:
2019 /
2020 /IF AN ERROR OCCURS, THEN IT IS ASSUMED [KEY] INIT FAILED TO CLEAR THE
2021 /MAINTENANCE FLIP-FLOP, OR, THAT THE ISSUING OF THE LCD IOT REALLY
2022 / [SET] THE MAINTENANCE FLIP-FLOP INSTEAD OF [CLEARING] .
2023 /
2024 / " C " LINES VERIFICATION PART I
2025 /
2026 0615 1376 TAD (177)
2027 0616 4437 LCD /MAINTENANCE MODE <OFF>
2028 /THE (AC) SHOULD = 0 BECAUSE IOT LCD 67X1 SHOULD CLEAR THE AC
2029 /
2030 0617 7140 SZA
2031 0620 4452 E0, ERROR / IOT 67X1 DID NOT CLEAR THE AC
2032 0621 4503 SUBSCOPE
2033 /
2034 /FLAG DETECTION PART II
2035 /
2036 /THE PURPOSE OF THIS TEST IS TO VERIFY THAT ISSUING IOT LCD 67X1 WITH
2037 /THE AC = 177 DOES NOT SET THE MAINTENANCE FLIP-FLOP
2038 /WHICH IN TURN WOULD SET ALL FLAGS
2039 /
2040 /THEREFORE ALL FLAGS SHOULD BE CLEARED
2041 /
2042 0622 4444 STR
2043 0623 4465 OK
2044 0624 4452 E1, EPROR /UNEXPECTED TRANSFER REQUEST FLAG
2045 0625 4503 SUBSCOPE
2046 0626 4445 SER
2047 0627 4465 OK
2048 0630 4452 E2, ERROR /UNEXPECTED ERROR FLAG
2049 0631 4503 SUBSCOPE
2050 0632 4446 SDN
2051 0633 4465 OK
2052 0634 4452 E3, ERROR /UNEXPECTED DONE FLAG
2053 0635 4502 SCOPE
2054 0636 5453 EXIT / END OF TEST 0
2055 /
2056 /TEST 1 - DIRECTION OF IOT XDR (67X2) PART I / IOT DECODING PART I
2057 /
2058 / - " C " LINES VERIFICATION PART II
2059 /
2060 /THE PURPOSE OF THIS TEST IS TO VERIFY THAT
2061 /ISSUING THE IOT XDR (TRANSFER DATA REGISTER) 67X2 DOES [NOT] CLEAR
2062 /THE MAINTENANCE FLIP-FLOP
2063 /
2064 /TECHNICAL NOTE:
2065 /
2066 /THE IOT'S SDN (67X5), AND SEP (67X4) ARE NOT TESTED HERE
2067 /BECAUSE IF AN RX01 MICROCONTROLLER IS CABLED TO THE RX8 INTERFACE
2068 /AND IF THE IOT LCD IS ISSUED WITH THE AC = 200

```

```

2069 /REALLY CLEARS THE MAINTENANCE FLIP-FLOP
2070 /THEN THE DONE FLAG, AND THE ERROR FLAG SHOULD BE CLEARED, AND
2071 /TRANSFER REQUEST MAY BE SET
2072 /BECAUSE THE CLEARING OF THE MAINTENANCE FLIP-FLOP WOULD HAVE
2073 /RESEMBLED A COMMAND TO THE RX01 MICROCONTROLLER
2074 /
2075 /TECHNICAL NOTE:
2076 /
2077 /IF THE CONTENTS OF THE TRANSFER REGISTER IS NOT = 200, THEN IS MUST
2078 /BE ASSUMED THAT THE SECOND LCD IOT CLEARED THE MAINTENANCE FLIP-FLOP
2079 /OR THAT IOT XDR CLEARED THE MAINTENANCE FLIP-FLOP
2080 /
2081 0637 3164 T1, DCA GOOD
2082 0640 1375 TAD (200)
2083 0641 4437 LCD / MAINTENANCE MODE <ON>
2084 /THE (AC) SHOULD = 0 AFTER ISSUING IOT LCD 67X1
2085 /
2086 0642 7440 SZA
2087 0643 4452 E11, ERROR / IOT LCD 67X1 FAILED TO CLEAR AC
2088 0644 4503 SUBSCOPE
2089 /
2090 0645 1375 TAD (200)
2091 0646 4437 LCD / MAINTENANCE MODE <ON>, AGAIN
2092 0647 4442 XDRIN /CONTENTS OF TRANSFER REGISTER
2093 0650 3165 DCA EAC /SAVE
2094 0651 1165 TAD EAC
2095 0652 1374 TAD (-200) /COMPARE WITH "EXPECTED"
2096 0653 7650 SNA CLA
2097 0654 5261 JMP +5 / OK
2098 0655 1375 TAD (200)
2099 0656 3164 DCA GOOD / "EXPECTED" RESULT
2100 0657 1165 TAD EAC / "ACTUAL" RESULT
2101 0660 4452 E10, EPROR /TRANSFER REGISTER NOT =200
2102 0661 4502 SCOPE
2103 0662 5453 EXIT / END OF TEST 1
2104 /TEST 2 - FLAG DETECTION PART III / " C " LINES VERIFICATION PART III
2105 /
2106 / (A) THE SETTING OF THE MAINTENANCE FLIP-FLOP SHOULD "DIRECT SET" ALL
2107 / FLAGS (DONE, TRANSFER REQUEST, AND ERROR).
2108 /
2109 / (B) IF AN RX01 MICROCONTROLLER [IS] CABLED TO THE RX8 INTERFACE,
2110 / THEN THE SETTING OF THE MAINTENANCE FLIP-FLOP WILL ASSERT THE "RUN"
2111 / LINE (RESEMBLING A FILL BUFFER COMMAND) THUS CAUSING THE RX01
2112 / CONTROLLER TO SETUP FOR A "FILL BUFFER", BUT, HOWEVER, BECAUSE THE
2113 / MAINTENANCE FLIP-FLOP [IS] SET, THE RX8 INTERFACE RECEIVERS
2114 / SHOULD BE DISABLED AND NOT REACTIVE TO THE RX01 MICROCONTROLLER.
2115 /
2116 /WITH ALL FLAGS SET, THE RX8 INTERFACE IOT'S:
2117 /
2118 / SDN = "SKIP ON DONE" (67X5), AND
2119 / SER = "SKIP ON ERROR" (67X4), AND
2120 / STR = "SKIP ON TRANSFER REQUEST" (67X3) SHOULD SKIP
2121 /
2122 /TECHNICAL NOTE:
2123 /

```

```

2124 /IF FLAGS ARE "MISSING", IS THE MAINTENANCE MODE FLIP-FLOP REALLY SET ?
2125
2126 #663 1375 T2, TAD (200)
2127 #664 4437 LCD / MAINTENANCE <ON>
2128 #665 4446 SDN
2129 #666 4452 E20, ERROR /MISSING DONE FLAG
2130 #667 4503 SUBSCOPE
2131 #670 4444 STR
2132 #671 4452 E21, ERROR /MISSING TRANSFER REQUEST FLAG
2133 #672 4503 SUBSCOPE
2134 #673 4445 SER
2135 #674 4452 E22, ERROR /MISSING ERROR FLAG
2136 #675 4503 SUBSCOPE
2137 /
2138 /ALL FLAGS SHOULD REMAIN " DIRECT SFT "
2139 /BECAUSE THE MAINTENANCE FLIP-FLOP SHOULD STILL BE SET
2140 /
2141 /TECHNICAL NOTE:
2142 /
2143 /IF THE FLAGS ARE "MISSING" THEN IT IS ASSUMED THAT THE PREVIOUS
2144 /FLAG TESTING ACTUALLY (CLEARED) THE FLAGS.
2145 /
2146 #676 4446 SDN
2147 #677 4452 E23, ERROR /MISSING DONE FLAG
2148 #700 4503 SUBSCOPE
2149 #701 4444 STR
2150 #702 4452 E24, ERROR /MISSING TRANSFER REQUEST FLAG
2151 #703 4503 SUBSCOPE
2152 #704 4445 SER
2153 #705 4452 E25, ERROR /MISSING ERROR FLAG
2154 #706 4503 SUBSCOPE
2155 / " C " LINES VERIFICATION PART III
2156 /
2157 /THE FOLLOWING RX8 INTERFACE IOT'S SHOULD NOT CLEAR THE AC:
2158 /
2159 /IOT'S: SDN(67X5), SER(67X4), OR STR(67X3)
2160 /
2161 #707 7240 STA
2162 #710 3164 DCA GOOD
2163 #711 1164 TAD GOOD
2164 #712 6755 K67X5E, 6755
2165 #713 7000 NOP
2166 #714 7650 SNA CLA
2167 #715 4452 E26, ERROR / IOT SDN (67X5) CLEARED THE AC
2168 #716 4503 SUBSCOPE
2169 #717 1164 TAD GOOD
2170 #720 6753 K67X3R, 6753
2171 #721 7000 NOP
2172 #722 7650 SNA CIA
2173 #723 4452 E27, ERROR / IOT STR (67X3) CLEARED THE AC
2174 #724 4503 SUBSCOPE
2175 #725 1164 TAD GOOD
2176 #726 6754 K67X4B, 6754
2177 #727 7000 NOP
2178 #730 7650 SNA CLA
    
```

```

2179 #731 4452 E28, ERROR / IOT SER (67X4) CLEARED THE AC
2180 #732 4502 SCOPE
2181 #733 5453 EXIT / END OF TEST 2
2182 /TEST 3
2183 /
2184 /PRX IOT DEVICE CODE VERIFICATION
2185 /
2186 /THE PURPOSE OF THIS TEST IS TO VERIFY THAT ONLY THE DEVICE CODE SELECTED
2187 /BY THE OPERATOR (AC SWITCHES 3-4-5 AT THE START OF THIS PROGRAM) IS ACTIVE.
2188 /
2189 /FIRST SET THE MAINTENANCE FLIP-FLOP, WHICH HAS PREVIOUSLY BEEN VERIFIED TO
2190 /DIRECT SET ALL FLAGS, THEN SEQUENCE THROUGH ALL DEVICE CODES (EXPECT THE
2191 /DEVICE CODE SELECTED AT THE START OF THIS PROGRAM) BY ISSUING IOT SDN 67X5
2192 / (SKIP ON DONE FLAG), WHICH HAS ALSO PREVIOUSLY BEEN VERIFIED TO "SKIP AND
2193 /CLEAR " SUCCESSFULLY.
2194 /
2195 /NOTE:
2196 /
2197 /THE PROGRAM DOES NOT ISSUE THE DEVICE CODE 67X5 WHERE X = POSITION OF AC
2198 /SWITCHES 3-4-5 AT THE START 200 OF THIS PROGRAM.
2199 /*****
2200 /*****
2201 /THIS TEST HAS BEEN REMOVED
2202 /THIS TEST IS NOT NEEDED IN A SYSTEM ENVIRONMENT.
2203 /TO REPLACE TEST T3 CHANGE LOCATIONS:
2204 / LOCATION FROM TO
2205 / ----- ---- --
2206 / 734 5464 1367
2207 /
2208 / 767 XXXX #200
2209 /REMOVED MAY 16,1975
2210 /
2211 /
2212 #734 5453 T3, EXIT
2213 #735 4437 LCD / MAINTENANCE <ON>
2214 #736 1773 TAD K67X5A
2215 #737 3164 DCA GOOD
2216 #740 1372 TAD (-7)
2217 #741 3010 DCA A10
2218 #742 1371 TAD (6705)
2219 #743 3352 DCACTIVE, DCA ACTIVE
2220 #744 4502 SCOPE /REFRESH PROGRAM LOCATION PCSCOPE
2221 #745 1164 TAD GOOD
2222 #746 7041 CIA
2223 #747 1352 TAD ACTIVE
2224 #750 7650 SNA CLA
2225 #751 5357 JMP NEXTACTIVE
2226 #752 6775 ACTIVE, 6775
2227 #753 5357 JMP NEXTACTIVE
2228 #754 1352 TAD ACTIVE
2229 #755 4452 E30, ERROR
2230 #756 4502 SCOPE / (AC) = ILLEGAL DEVICE CODE
2231 #757 1370 NEXTACTIVE, TAD (10)
2232 #760 1352 TAD ACTIVE
2233 #761 2010 ISZ A10
    
```

2234 0762 5343
2235 0763 5453
2236 0770 0010
2237 0771 6705
2238 0772 7771
2239 0773 6422
2240 0774 7600
2241 0775 0200
2242 0776 0177
2243 0777 0545

JMP UCACTIVE
EXIT / END OF TEST 3

PAGE

2244 /TEST 4 - TRANSFER REGISTER DIRECTION TESTING (PART II)
2245 /
2246 / - " C " LINES VERIFICATION PART IV
2247 /
2248 /WITH THE MAINTENANCE FLIP-FLOP SET THE PROGRAM WILL VERIFY THE DIRECTION
2249 /AND TRANSFER MODE (8-BIT MODE INCLUSIVE "OR", AND 12-BIT MODE "JAM")
2250 /TRANSFERS INTO THE ACCUMULATOR FROM THE RX8 TRANSFER REGISTER BY ISSUING
2251 /IOT "XDM" (TRANSFER DATA REGISTER) 67X2 AFTER PREVIOUSLY [LOADING] THE
2252 /THE TRANSFER REGISTER WITH THE CONTENTS OF THE ACCUMULATOR REPRESENT-
2253 /ACTIVE OF THE FLOWING PATTERNS WHEN THE "LCD" IOT 67X1 IS ISSUED.
2254 /
2255 / (1) 200 - MAINTENANCE MODE <ON>
2256 / (2) 376 -
2257 / (3) 375 -
2258 / (4) 373 - (BYTES 2 THRU 7)
2259 / (5) 367 - (INCLUSIVE "OR")
2260 / (6) 357 -
2261 / (7) 337 -
2262 / (8) 7677 - (WORD 8 - "JAM ")
2263 /
2264 /THE LCD IOT WILL BE ISSUED A TOTAL OF 8 TIMES.
2265 /
2266 /THE 1ST LCD IOT WILL BE ISSUED WITH THE AC = 200 WHICH INITIALLY SETS THE
2267 /MAINTENANCE FLIP-FLOP THEREBY GUARANTEEING THE CONTENTS OF THE TRANSFER
2268 /REGISTER [ADTER] EACH SUCCEEDING LCD IOT.
2269 /
2270 /LCD IOT'S 2 THRU 8 ARE ISSUED WITH THE ACCUMULATOR CONTAINING THE PATTERNS
2271 /DESCRIBED ABOVE.
2272 /
2273 /ALL PATTERNS EXCEPT WORD 8 (7677) TEST THE INCLUSIVE "OR" TRANSFER OF
2274 /THE RX8 INTERFACE TRANSFER REGISTER. WORD 8 TESTS THE 12-BIT "JAM" TRANSFER.
2275 /
2276 1000 1377 T4, TAD (200)
2277 1001 4437 LCD /MAINTENANCE MODE <ON>
2278 1002 7350 CLL STA RAR / 3777
2279 1003 3166 DCA BLANK
2280 1004 1166 T4H, TAD BLANK
2281 1005 7120 STL
2282 1006 7500 SMA
2283 1007 7100 CIL
2284 1010 7004 RAL
2285 1011 3166 DCA BLANK
2286 1012 4502 SCOPE / REFRESH PROGRAM LOCATION PCSCOPE
2287 1013 1166 TAD BLANK / (BLANK) = (AC) BEFORE LCD IOT 67X1

2288 1014 4437
2289 /
2290 / " C " LINES VERIFICATION PART IV
2291 /
2292 /THE PURPOSE OF THIS TEST IS TO VERIFY THAT SUCCEEDING LCD IOT'S(67X1)
2293 /TRANSFER THE (AC) INTO THE DATA REGISTER CLEARING THE ACCUMULATOR
2294 /
2295 1015 3165 DCA EAC / (AC) AFTER ISSUING IOT LCD (67X1)
2296 1016 3164 DCA GOOD / PROGRAM EXPECTS AC = 0
2297 1017 1165 TAD EAC
2298 1020 7440 SZA
2299 1021 4452 E42, EPROR / IOT LCD (67X1) DIDN'T CLEAR THE AC
2300 1022 4503 SUBSCOPE
2301 /
2302 /TRANSFER DIRECTION PART II
2303 /
2304 1023 4442 XDRIN / FROM
2305 1024 3165 DCA EAC
2306 1025 1376 TAD (100)
2307 1026 0166 AND BLANK
2308 1027 7106 CLL RTL
2309 1030 7006 / LINK = 1 FOR 8-BIT MODE
2310 1031 7006 RTL
2311 1032 1166 TAD BLANK
2312 1033 7430 SZL
2313 1034 0375 AND (377) / 8-BIT BYTE "GOOD" MASK
2314 1035 3164 DCA GOOD
2315 1036 1164 TAD GOOD /EXPECTED RESULT
2316 1037 7041 CIA
2317 1040 1165 TAD EAC /ACTUAL RESULT
2318 1041 7650 SNA CLA
2319 1042 5245 JMP .+3 /COMPARED OK
2320 1043 1165 TAD EAC
2321 1044 4452 E40, EPROR / TRANSFER REGISTER NOT = "GOOD"
2322 1045 4503 SUBSCOPE
2323 /
2324 /THE TRANSFER REGISTER SHOULD REMAIN UNCHANGED
2325 /FROM THE PREVIOUS XDR IOT
2326 /
2327 1046 4442 XDRIN / FROM
2328 1047 3165 DCA EAC
2329 1050 1165 TAD EAC /ACTUAL
2330 1051 7041 CIA
2331 1052 1164 TAD GOOD /EXPECTED
2332 1053 7650 SNA CLA
2333 1054 5257 JMP .+3 /COMPARED OK
2334 1055 1165 TAD EAC
2335 1056 4452 E41, EPROR / TRANSFER REGISTER NOT = "GOOD"
2336 1057 4502 SCOPE
2337 1060 1166 TAD BLANK
2338 1061 0376 AND (100)
2339 1062 7640 SZA CLA
2340 1063 5204 JMP T4B /UNTIL (BLANK) = 7677.
2341 1064 5453 EXIT / END OF TEST 4
2342 /TEST 5

```

2343 /
2344 /RX8 IOT DECODING VERIFICATION PART II
2345 /
2346 /*****
2347 /*****
2348 /
2349 /IF AN RX01 CONTROL IS CABLED TO THE RX8 INTERFACE
2350 /THEN DON'T EXECUTE THIS TEST
2351 /BECAUSE THE CLEARING OF THE MAINTENANCE F/F
2352 /RESEMBLES A FILL BUFFER COMMAND
2353 /
2354 1065 1163 T5, TAD RXHERE
2355 1066 7650 SNA CLA
2356 1067 5453 NOTEST
2357 1070 3164 DCA GOOD
2358 /*****
2359 /*****
2360 /
2361 /THE MAINTENANCE FLIP-FLOP HAS PREVIOUSLY BEEN VERIFIED TO SET AND CLEAR.
2362 /THE IOT UNDER TEST SHOULD "SKIP AND CLEAR" [ONLY] ITS RESPECTIVE FLAG.
2363 /ALL OTHER FLAGS SHOULD REMAIN UNCHANGED
2364 /
2365 / (I.E. THE SDN IOT 67X5 SHOULD SKIP AND CLEAR ONLY THE DONE FLAG, ALL
2366 /OTHER FLAGS SHOULD REMAIN SET)
2367 /
2368 1071 1377 TAD (200)
2369 1072 4437 LCD
2370 1073 4437 LCD / MAINTENANCE <ON> / <OFF>
2371 1074 4442 XDRIN
2372 1075 7440 SZA
2373 1076 4452 E56, ERROR /TRANSFER REGISTER NOT = 0
2374 1077 4503 SUBSCOPE
2375 1100 4446 SDN
2376 1101 4452 E50, ERROR /DONE FLAG WASN'T SET, OR
2377 1102 4503 SUBSCOPE /IOT LCD OR XDR CLEARED THE DONE FLAG
2378 1103 4446 SDN
2379 1104 4465 OK
2380 1105 4452 E53, ERROR /IOT SDN DIDN'T "SKIP AND CLEAR"
2381 1106 4503 SUBSCOPE
2382 1107 4444 STF
2383 1110 4452 E51, ERROR /TRANSFER REQUEST FLAG WASN'T EVER SET, OR
2384 1111 4503 SUBSCOPE /IOT LCD, OR SDN OR XDR CLEARED THE TR FLAG
2385 1112 4444 STR
2386 1113 4465 OK
2387 1114 4452 E54, ERROR /IOT STR DIDN'T "SKIP AND CLEAR"
2388 1115 4503 SUBSCOPE
2389 1116 4445 SEP
2390 1117 4452 E52, ERROR /EPROR FLAG WASN'T EVER SET, OR
2391 1120 4503 SUBSCOPE /IOTS LCD OR SDN OR XDR OR STF CLEARED THE ERROR FLAG
2392 1121 4445 SEP
2393 1122 4465 OK
2394 1123 4452 E55, ERROR /IOT SER DIDN'T "SKIP AND CLEAR"
2395 1124 4502 SCOPE / END OF TEST 5
2396 1125 5453 EXIT
2397 /TEST 6 = INTERRUPT TEST PART I / IOT DECODING VERIFICATION PART III
    
```

```

2398 /
2399 /INTERRUPT TEST PART I
2400 /
2401 /THE MAINTENANCE FLIP-FLOP HAS PREVIOUSLY BEEN VERIFIED TO DIRECT
2402 / [SET] ALL FLAGS AND THE INTERFACE IOT "SKIP ON DONE" "SDN" 67X5 WAS
2403 /FOUND TO "SKIP AND CLEAR" SUCCESSFULLY,
2404 /
2405 /FIRST SET THE MAINTENANCE FLIP-FLOP WHICH IN TURN SETS ALL FLAGS.
2406 /
2407 /THEN ISSUE IOT INTR 67X6 WITH THE AC = 0 [CLEARING] THE RX8 INTERRUPT ENABLE
2408 /NO INTERRUPTS SHOULD OCCUR
2409 /
2410 1126 4507 T6, WAITTY
2411 1127 1374 TAD (E60)
2412 1130 3002 DCA IPI
2413 1131 1377 TAD (200)
2414 1132 4437 LCD
2415 1133 4447 INTR /INTERRUPT ENABLE FLIP-FLOP <OFF>
2416 1134 7000 NOP /...WAIT
2417 1135 7000 NOP /...PLENTY
2418 1136 7410 SKP /...OF TIME
2419 1137 4452 E60, ERROR /UNEXPECTED INTERRUPT
2420 1140 4503 SUBSCOPE
2421 /
2422 /*****
2423 /*****
2424 /
2425 /IF AN RX01 MICRO-CONTROLLER IS CABLED TO THE RX8 INTERFACE
2426 /THEN DON'T EXECUTE THE REMAINING PORTION OF THIS TEST
2427 /BECAUSE THE CLEARING OF THE MAINTENANCE FLIP-FLOP RESEMBLES A FILL BUFFER COMMAND
2428 /TO THE RX01 MICROCONTROLLER
2429 /
2430 1141 1163 TAD RXHERE
2431 1142 7650 SNA CLA
2432 1143 5453 NOTEST
2433 /
2434 /*****
2435 /*****
2436 /
2437 /IOT DECODING PART III- IOT INTR 67X6 DECODING VERIFICATION
2438 /
2439 /TECHNICAL NOTE:
2440 /
2441 /ALL FLAGS SHOULD REMAIN SET
2442 /IF ANY FLAG IS MISSING,
2443 /THEN IT IS ASSUMED THAT IOT "INTR" 67X6 CLEARED THE FLAG(S)
2444 /
2445 1144 4437 LCD /MAINTENANCE MODE <OFF>
2446 1145 4447 INTR /DISABLE THE INTERRUPT ENABLE F/F
2447 1146 4446 SDN
2448 1147 4452 E61, ERROR /MISSING DONE FLAG
2449 1150 4503 SUBSCOPE
2450 1151 4444 STR
2451 1152 4452 E62, ERROR /MISSING TRANSFER REQUEST FLAG
2452 1153 4503 SUBSCOPE
    
```

```

2453 1154 4445 SER
2454 1155 4452 E63, ERROR /MISSING ERROR FLAG
2455 1156 4502 SCOPE
2456 1157 5453 EXIT / END OF TEST 6
2457 1174 1137
2458 1175 0377
2459 1176 0100
2460 1177 0200
      1200
      PAGE
2461 /
2462 /TEST 7 - INTERRUPT TEST PART II
2463 /
2464 /INTERUPT TEST PART II
2465 /
2466 /THE MAINTENANCE FLIP-FLOP HAS PREVIOUSLY BEEN VERIFIED TO DIRECT
2467 / [SET] ALL FLAGS AND THE INTERFACE IOT -SKIP ON DONE- "SDN" 67X5 WAS
2468 /FOUND TO "SKIP AND CLEAR" SUCCESSFULLY,
2469 /
2470 /FIRST SET THE MAINTENANCE FLIP-FLOP
2471 /WHICH SHOULD DIRECT SET THE DONE FLAG,
2472 /THEN BY SETTING THE RX01 INTERRUPT ENABLE
2473 /RY ISSUING THE IOT "INTR" 67X6 WITH THE AC = 1.
2474 /
2475 /AN INTERRUPT REQUEST SHOULD BE ASSERTED.
2476 /
2477 /THE PROGRAM IS EXPECTING AN INTERRUPT.
2478 /
2479 /TECHNICAL NOTE:
2480 /
2481 /IF AN INTERRUPT DOES NOT OCCUR, THEN IT IS ASSUMED THAT ISSUING THE IOT
2482 / "INTR" 67X6 DID NOT SET THE RX8 INTERRUPT ENABLE, OR INTERRUPT REQUEST
2483 /
2484 1200 4507 T7, WAITTY
2485 1201 1377 TAD (200)
2486 1202 4437 LCD /MAINTENANCE <ON>
2487 1203 1376 TAD (70K)
2488 1204 3032 DCA IPI
2489 1205 6001 ION /*SEE FOOTNOTE NEXT PAGE
2490 1206 7201 CLA IAC
2491 1207 4447 INTR /RX01 INTERRUPT ENABLE <ON>
2492 1210 7000 NOP
2493 1211 7000 NOP
2494 /PROGRAM NOTE:
2495 /
2496 /CLEAR PROGRAM LOCATION "GOBIT" BECAUSE THE TIME FOR THE INTERRUPT
2497 /TO OCCUR HAS EXPIRED (IF IT WAS EVER GOING TO OCCUR THAT IS)
2498 /
2499 1212 3152 DCA GOBIT
2500 1213 4452 E70, ERROR /MISSING INTERRUPT
2501 1214 4502 T70K, SCOPE
2502 1215 5453 EXIT / END OF TEST 7
2503 /TEST 10
2504 /
2505 /INTERUPT TEST (PART III)
2506 /

```

```

2507 /IOT INTR 67X6 SHOULD CLEAR THE INTERRUPT ENABLE FLIP-FLOP, THEN
2508 /
2509 /WITH ALL FLAGS SET, NO INTERRUPTS SHOULD OCCUR
2510 /
2511 /TECHNICAL NOTE:
2512 /
2513 /IF AN UNEXPECTED PROGRAM INTERRUPT OCCURS FROM APPROXIMATELY THIS PC
2514 /THEN THE RX PROGRAM INTERRUPT REQUEST TOOK TOO LONG TO SET
2515 /FROM THE PREVIOUS TEST.
2516 /
2517 / * FOOTNOTE:
2518 /
2519 /THIS IOT "ION" IS ISSUED HERF BECAUSE - IF AN UNEXPECTED PROGRAM
2520 /INTERRUPT HAD OCCUPIED IN THE PREVIOUS TEST AND AC SW3 = 1 DIRECTING
2521 /THE PROGRAM NOT TO PRINT AN FROR - THEN THE PDP'S INTERRUPT FACILITY
2522 /WOULD BE <OFF> - THEREFORE NEVER EXECUTING THIS TEST PROPERLY
2523 /
2524 1216 4507 T10, WAITTY
2525 1217 4447 INTR /DISABLE RX8 INTERRUPT ENABLE
2526 1220 1375 TAD (E100)
2527 1221 3002 DCA IPI
2528 1222 6001 ION /*SEE FOOTNOTE ABOVE
2529 1223 1377 TAD (200)
2530 1224 4437 LCD
2531 1225 7000 NOP
2532 1226 7410 SKP
2533 1227 4452 E100, ERROR /UNEXPECTED INTERRUPT
2534 1230 4502 SCOPE
2535 1231 5453 EXIT / END OF TEST 10
2536 /TEST 11
2537 /
2538 /INTERUPT TEST (PART IV)
2539 /
2540 /*****
2541 /*****
2542 /
2543 /IF AN RX01 CONTROL IS CABLED TO THE RX8 INTERFACE
2544 /THEN DON'T EXECUTE THIS TEST
2545 /BECAUSE THE CLEARING OF THE MAINTENANCE F/F
2546 /RESEMBLES A FILL BUFFER COMMAND
2547 /
2548 1232 1163 T11, TAD RXHERE
2549 1233 7650 SNA CLA
2550 1234 5453 NOTEST
2551 /*****
2552 /*****
2553 /
2554 /TOGGLING THE MAINTENANCE MODE <ON> / <OFF> SETS ALL FLAGS AND
2555 /
2556 /PERMITS IOT SDN TO CLEAR THE DONE FLAG
2557 /
2558 /THEREFORE NO INTERRUPTS SHOULD OCCUR (ONLY DONE FLAG RAISES AN INTERRUPT REQUEST)
2559 /
2560 / (EVEN THOUGH THE RX01 INTERRUPT ENABLE IS ! )
2561 /

```

```

2562 1235 4507          WAITTY
2563                    /
2564 1236 1377          TAD (200)
2565 1237 4437          LCD
2566 1240 4437          LCD          /MAINTENANCE <ON> / <OFF>
2567 1241 4446          SDN
2568 1242 7000          NOP
2569 1243 7000          NOP          /CLEAR THE DONE FLAG
2570 1244 1374          TAD (E110)
2571 1245 3002          DCA IPI
2572 1246 7201          CLA IAC
2573 1247 4447          INTR          /RX01 INTERRUPT ENABLE <ON>
2574 1250 7000          NOP
2575 1251 7000          NOP
2576 1252 7330          STL CLA RAR
2577 1253 4447          INTR          /RX01 INTERRUPT ENABLE <OFF>
2578 1254 1152          TAD GOBIT
2579 1255 7700          SMA CLA
2580 1256 4452          ERROR          /UNEXPECTED INTERRUPT
2581 1257 4502          SCOPE
2582 1260 5453          EXIT          / END OF TEST 11
2583                    /
2584                    /
2585                    /
2586                    /
2587                    /
2588                    /IF AN RX01 IS CABLED TO THE RX8 THEN DON'T EXECUTE T12
2589                    /
2590                    /BUT EXECUTE ALT12 (THE ALTERNATIVE TEST)
2591                    /
2592 1261 1163          T12,   TAD RXHERE
2593 1262 7650          SNA CLA
2594 1263 5773          JMP ALT12
2595                    /
2596                    /
2597                    /
2598                    /INTERRUPT TEST PART V / INITIALIZE TEST PART II [PROGRAMMED]
2599                    /
2600                    /THE PURPOSE OF THIS TEST IS TO VERIFY THAT IOT INIT CLEARS THE INTERRUPT
2601                    /ENABLE FLIP-FLOP WHEN SET
2602                    /
2603 1264 1372          TAD (E124)
2604 1265 3002          DCA IPI
2605 1266 7201          CLA IAC
2606 1267 4447          INTR          / SET THE RX8 INTERRUPT ENABLE F/F
2607 1270 4451          INTR
2608 1271 3152          DCA GOBIT          / ISSUE INIT IOT 67X7
2609                    /...BUT AN INTERRUPT SHOULD NOT OCCUR
2610                    /
2611                    /IF AN INTERRUPT OCCURS THEN IOT INIT FAILED TO CLEAR
2612                    /
2613                    /THE RX8 INTERRUPT ENABLE FLIP-FLOP
2614                    /
2615 1272 1377          TAD (200)
2616 1273 4437          LCD

```

```

2617 1274 4437          LCD          / MAINTENANCE MODE <ON> / <OFF>
2618                    /
2619                    /THE DONE FLAG SHOULD BE SET, BUT NO INTERRUPTS SHOULD OCCUR
2620                    /
2621 1275 7330          STL CLA RAR
2622                    /
2623                    /RETURN TO HERE IF AN INTERRUPT OCCURED
2624                    /
2625 1276 4447          E124,  INTR          / RX8 INTERRUPT ENABLE <OFF>
2626 1277 1152          TAD GOBIT
2627 1300 7700          SMA CLA
2628 1301 4452          ERROR          / IOT INIT 67X7 DID NOT CLEAR THE IE F/F
2629 1302 4503          SUBSCOPE
2630                    /
2631                    /IOT "INIT" 67X7 SHOULD CLEAR THE RX8 INTERFACE TRANSFER REGISTER, THE
2632                    /MAINTENANCE FLIP-FLOP, AND ALL FLAGS (DONE, TRANSFER REQUEST, AND ERROR).
2633                    /
2634 1303 7240          STA
2635 1304 4437          LCD          /ALL 1'S TO TRANSFER REGISTER
2636 1305 4451          INITB          / IOT 67X7
2637 1306 4446          SDN
2638 1307 4465          OK
2639 1310 4452          E120,  ERROR          /UNEXPECTED DONE FLAG
2640 1311 4503          SUBSCOPE
2641 1312 4444          STR
2642 1313 4465          OK
2643 1314 4452          E121,  ERROR          /UNEXPECTED TRANSFER REQUEST FLAG
2644 1315 4503          SUBSCOPE
2645 1316 4445          SER
2646 1317 4465          OK
2647 1320 4452          E122,  ERROR          /UNEXPECTED ERROR FLAG
2648 1321 4503          SUBSCOPE
2649 1322 3164          DCA GOOD          / PROGRAM EXPECTS TRANSFER REGISTER = 0
2650 1323 4142          XDRIN
2651 1324 7440          SZA
2652 1325 4452          E123,  ERROR          /TRANSFER REGISTER NOT = 0
2653 1326 4502          SCOPE
2654                    /
2655                    /
2656                    /
2657                    /
2658                    /NO MORE RX8 INTERFACE TESTS EXIST
2659                    /
2660                    /IF AN RX01 CONTROL IS CABLED TO THE RX8 INTERFACE
2661                    /
2662                    /THEN CONTINUE WITH THE NORMAL FLOW OF TESTING
2663                    /
2664 1100          XI=1100
2665                    /
2666                    / END OF PASS " I "
2667                    /
2668 1327 4464          LOCKUP
2669 1330 1371          TAD (XI)
2670 1331 5770          JMP NOMORETESTS
2671                    /

```

```

2672 /*****
2673 1332 1332 XSCOPE,
2674 1333 1363 TAD ERRORS
2675 1334 7640 SZA CLA
2676 1335 5355 JMP SCOPING
2677 /NO ERROR HAS BEEN DETECTED HERE
2678 /
2679 /JUST SET (PCSCOPE)=THE FIRST ADDRESS OF THE SCOPE LOOP
2680 /
2681 / (IN CASE ANY ERRORS ARE EVER DETECTED LATER)
2682 /
2683 1336 3363 NOSCOPE, DCA ERRORS
2684 1337 1332 TAD XSCOPE
2685 1340 3364 DCA PCSCOPE
2686 1341 5732 JMP I XSCOPE
2687 / " SUBSCOPE "
2688 /
2689 1342 1342 XSSCOPE,
2690 1343 1342 TAD XSSCOPE
2691 1344 3332 DCA XSCOPE
2692 1345 1363 TAD ERRORS
2693 1346 7650 SNA CLA
2694 1347 5732 JMP I XSCOPE
2695 /ERRORS DO EXIST
2696 /
2697 /IF THIS ERROR IS THE SAME AS THE ADDRESS WITHIN THE PROGRAM LOCATION
2698 /PCSSCOPE, THEN THIS IS A SCOPE LOOP
2699 /
2700 /IF NOT, THEN EXIT
2701 /
2702 1350 1332 TAD XSCOPE
2703 1351 7041 CIA
2704 1352 1362 TAD EPCSCOPE
2705 1353 7640 SZA CLA
2706 1354 5732 JMP I XSCOPE
2707 /THIS IS A SCOPING LOOP
2708 /
2709 1355 4570 SCOPING, LAS
2710 1356 0377 AND (SW4)
2711 1357 7650 SNA CLA
2712 1360 5336 JMP NOSCOPE
2713 1361 5764 JMP I PCSCOPE
2714 1362 0000 EPCSCOPE, 0 /ADDRESS +1 OF "SCOPE" OR "SUBSCOPE"
2715 1363 0000 ERRORS, 0 / > 0 IF AN ERROR HAS BEEN DETECTED (FOR THIS TEST)
2716 1364 0000 PCSCOPE, 0 / FIRST ADDRESS OF SCOPE LOOP
2717 1370 0522
2718 1371 1100
2719 1372 1276
2720 1373 1400
2721 1374 1253
2722 1375 1227
2723 1376 1214
2724 1377 0200
2725 1400 PAGE
/ALTERNATE TEST 12 - VERIFICATION OF [INIT]

```

```

2726 /
2727 /INITIALIZE TEST PART II
2728 /
2729 /THIS TEST IS EXECUTED IN PLACE OF T12
2730 /
2731 /BECAUSE AN RX01 CONTROLLER IS CABLED TO THE RX8 INTERFACE
2732 /
2733 /IOT "INIT" 67X7 WILL PERFORM AN IMPLIED READ OF TRACK 0 SECTOR 1
2734 /
2735 / (IF DRIVE 0 IS READY)
2736 /
2737 /THEREFORE THE DONE FLAG SHOULD SET AT THE END OF THAT IMPLIED READ.
2738 /
2739 /TECHNICAL NOTE:
2740 /
2741 /IF AN ERROR FLAG IS SET (AND DRIVE 0 IS READY) THEN THE ERROR MAY HAVE
2742 /BEEN THE RESULT FROM THE [IMPLIED READ SECTOR 0]
2743 /
2744 1400 4502 ALT12, SCOPE
2745 1401 4451 INITB
2746 1402 4444 ALT12LOOP, STR
2747 1403 7410 SKP
2748 1404 5211 JMP EA120 /UNEXPECTED TRANSFER REQUEST FLAG
2749 1405 4446 SDN
2750 1406 5202 JMP ALT12LOOP /WAIT FOR THE DONE FLAG
2751 1407 4444 STR
2752 1410 4465 OK
2753 1411 4452 EA120, ERROR /UNEXPECTED TRANSFER REQUEST
2754 1412 4503 SUBSCOPE
2755 1413 4445 SER
2756 1414 4465 OK
2757 1415 4452 EA121, ERROR /UNEXPECTED ERROR FLAG
2758 1416 4503 SUBSCOPE
2759 1417 1777 TAD UNITS /UNITS SELECTED BY OPERATOR
2760 1420 7710 SPA CLA
2761 1421 1376 TAD (40) /PROGRAM EXPECTS DRIVE 0 TO BE READY
2762 /
2763 /
2764 /
2765 /
2766 / 4 5 - - 8 9 10 11 /
2767 /
2768 / SEL WRITE INIT PAR /
2769 / DRIVE DD PROTECT [DONE] CRC /
2770 / RDY (N/A) /
2771 /
2772 /
2773 /
2774 /
2775 1422 7107 CLL IAC RTL / 4 [INIT] DONE, OR 204
2776 1423 3164 DCA GOOD
2777 1424 4442 XDRIN / "ACTUAL" STATUS AT DONE
2778 1425 3166 DCA BLANK
2779 1426 1166 TAD BLANK
2780 1427 0375 AND (-100-1)

```

```

2781 1430 3165 DCA EAC / "ACTUAL" MINUS DELETED DATA (IF ANY)
2782 1431 1165 TAD EAC
2783 1432 7041 CIA
2784 1433 1164 TAD GOOD /EXPECTED STATUS
2785 1434 7650 SNA CLA
2786 1435 5240 JMP ,+3 /COMPARED OK
2787 1436 1165 TAD EAC
2788 1437 4452 EA122, ERROR /TRANSFER REGISTER NOT = "GOOD"
2789 1440 4503 SUBSCOPE
2790 /
2791 /READ THE B-CODE STATUS AND EXPECT = 0
2792 /
2793 /PROGRAMMING NOTE:
2794 /
2795 /THE PROGRAM SAVES THE CONTENTS OF "GOOD" FOR REFERENCES WITHIN SUBROUTINE "XRST"
2796 /
2797 /BECAUSE "GOOD" IS REFRESHED WITHIN PROGRAM SUBROUTINE "XRSTB"
2798 /
2799 1441 1164 TAD GOOD
2800 1442 3166 DCA BLANK
2801 1443 4477 RSTB
2802 1444 7650 SNA CLA
2803 1445 5251 JMP ,+4
2804 1446 3164 DCA GOOD /PROGRAM EXPECTS 0
2805 1447 1165 TAD EAC
2806 1450 4452 EA123, ERROR /H-CODE NOT = 0
2807 1451 4503 SUBSCOPE
2808 /
2809 /READ THE CONTENTS OF THE RX01 STATUS REGISTER USING THE COMMAND # 5
2810 /
2811 /THIS STATUS SHOULD = THE STATUS IN THE TRANSFER REGISTER AT ERROR/DONE
2812 /
2813 1452 4476 RST / "READ STATUS" (COMMAND # 5)
2814 /
2815 1453 5453 EXIT / END OF TEST AL112 (TEST 12)
2816 /THE PURPOSE OF THESE TESTS IS TO VERIFY THE TRANSFER LENGTH OF THE FUNCTION
2817 / "FILL BUFFER" AND "EMPTY BUFFER" OF THE RX01 MICROCONTROLLER
2818 /
2819 /64 TRANSFERS SHOULD OCCUR FOR 12-BIT MODE, AND
2820 /128 TRANSFERS SHOULD OCCUR FOR 8-BIT MODE
2821 /
2822 /THE SECTOR BUFFER IS FILLED WITH A COUNT PATTERN
2823 /
2824 /
2825 / WORD/BYTE 0 = 0
2826 /
2827 / WORD 2 = 0202
2828 / BYTE 2 = 2
2829 /
2830 / WORD 77 = 7777
2831 / BYTE 177 = 177
2832 /
2833 /FILL BUFFER 8-BIT MODE
2834 /
2835 1454 1374 T16, TAD (100)

```

```

2836 /FILL BUFFER 12-BIT MODE
2837 /
2838 1455 4437 T13, LCD /ISSUE THE COMMAND
2839 1456 1374 TAD (100)
2840 1457 0112 AND COMMAND
2841 1460 7640 SZA CLA
2842 1461 7307 CLL CLA IAC RTL / 4
2843 1462 1373 TAD (A000)
2844 1463 3132 DCA TESTP
2845 1464 4455 GETAPATTERN
2846 1465 1372 TAD (WBUFFER-1)
2847 1466 3010 DCA A10 /PROGRAMS "WRITE" BUFFER
2848 1467 1371 TAD (WBUFFER+1)
2849 1470 3133 DCA XA10 / -STARTING ADDRESS OF WRITE BUFFER
2850 1471 4276 FB, JMS FBEB
2851 1472 5453 EXIT / ** END OF TESTS 13, OR 16
2852 1473 1410 TAD I A10
2853 1474 4443 XDROUT
2854 1475 5271 JMP FB / AND "FILL THE BUFFER"
2855 / FILL / EMPTY BUFFER SUBROUTINE FOR TESTS: *** 13, 16 / 14, 17 ***
2856 /
2857 1476 1476 FBEB, .
2858 1477 4444 STR
2859 1500 5303 JMP ,+3 /WAIT FOR TRANSFER REQUEST FLAG
2860 1501 2276 ISZ FBEB
2861 1502 5676 JMP I FBEB
2862 1503 4446 SDN
2863 1504 5277 JMP FBEB+1 /WAIT FOR THE DONE FLAG
2864 1505 3006 DCA 6 /FIRST TIME FOR WAIT.
2865 1506 1112 TAD COMMAND
2866 1507 3166 DCA BLANK
2867 1510 4445 SER
2868 1511 4465 OK
2869 1512 4452 E130, ERROR /UNEXPECTED ERROR FLAG
2870 1513 4503 SUBSCOPE
2871 /
2872 / 64 OR 128 BYTES SHOULD HAVE BEEN TRANSFERRED IN OR OUT
2873 /
2874 1514 1374 TAD (100)
2875 1515 0112 AND COMMAND / 200 FOR 8-BIT MODE
2876 1516 1374 TAD (100)
2877 1517 3164 DCA GOOD
2878 1520 1164 TAD GOOD
2879 1521 7041 CIA / 100 FOR 12-BIT MODE
2880 1522 1010 TAD A10
2881 1523 1133 TAD XA10
2882 1524 7440 SZA
2883 1525 4452 E131, ERROR /SKIP IF TRANSFERS OK
2884 1526 4502 SCOPE / (AC) = - # MEANS NOT ENOUGH TRANSFERS
2885 1527 5676 JMP I FBEB / (AC) > 0 MEANS TO MANY TRANSFERS
2886 /RETURN IS TO EXIT
2887 /
2888 /*****
2889 /CONSOLE PACKAGE
2890 /*****

```



```

2891 /*****
2892 /CONSOL
2893 /*****
2894
2895 1530 0000 PNTID, 0 /PRINT BEGIN MESSAGES
2896 1531 6007 CAF /TO REPLACE A CLEAR SWITCH
2897 1532 4470 APT8 /TEST FOR APT SYSTEM.
2898 1533 4473 PRINT
2899 1534 6471 MIDENTIFICATION /ID MESSAGE
2900 1535 4473 PRINT
2901 1536 6452 REMOVE /REMOVE DIAGNOSTIC DISKETTE
2902 1537 4424 CHECKC0
2903 1540 4405 C8LOOK /WAIT FOR CONTROL E TO CONTINUE
2904 1541 7000 NOP /REQUIRED.
2905 1542 4473 PRINT
2906 1543 6504 MSELECT /SELECT PARAMETERS
2907 1544 7200 CLA
2908 1545 5730 JMP I PNTID /EXIT PNTID
2909 /*****
2910 /
2911 /
2912 /
2913 /
2914 /
2915 /ROUTINE TO DETERMINE IF ON APT-8. IF APT-8 IS SELECTED
2916 /THEN CONSOLE AND TEST PARAMETER SELECTION FUNCTIONS ARE NOP.
2917 /IF NOT ROUTINE IS NOP.
2918 /
2919 1546 0000 XAPT8, 0
2920 1547 7300 CLA CLL
2921 1550 4472 CHEK22
2922 1551 7410 SKP /ON APT-8
2923 1552 5746 JMP I XAPT8
2924 1553 1022 TAD 22
2925 1554 0362 AND K7377 /NOP CONSOLE PACKAGE
2926 1555 3022 DCA 22 /RESTORE 22
2927 1556 1172 TAD 17000
2928 1557 3763 DCA I HLTNOP
2929 1560 1020 TAD 20 /GET TEST PARAMETERS.
2930 1561 5770 JMP AROUND-2 /MAIN FLOWOF PROGRAM.
2931 1562 7377 K7377, 7377
2932 1563 3316 HLTNOP, C0RET4-1
2933 1570 0226
2934 1571 0666
2935 1572 7112
2936 1573 6000
2937 1574 0100
2938 1575 7677
2939 1576 0040
2940 1577 4235
2941 1600 PAGE
2942 /SECTOR BUFFER ADDRESSING VERIFICATION TESTS
2943 /
2944 /OPERATIONAL NOTE:

```

```

2945 / (1). FOR TEST 14 TO EMPTY THE BUFFER IN 12-BIT MODE SUCCESSFULLY, TEST 13
2946 / MUST HAVE FILLED THE BUFFER IN 12-BIT MODE SUCCESSFULLY.
2947 /
2948 / (2). FOR TEST 17 TO EMPTY THE BUFFER IN 8-BIT MODE SUCCESSFULLY, TEST 16
2949 / MUST HAVE FILLED THE BUFFER IN 8-BIT MODE SUCCESSFULLY
2950 /
2951 /THE PURPOSE OF THESE TESTS IS TO VERIFY THAT THE CONTENTS OF THE SECTOR
2952 /BUFFER REMAIN UNCHANGED AFTER THE PREVIOUS EMPTY BUFFER 8-BIT MODE TEST, AND
2953 /AFTER THE PREVIOUS EMPTY BUFFER 12-BIT MODE TEST
2954 /
2955 1600 7410 T20, SKP /VERIFY EMPTY BUFFER 8-BIT MODE
2956 /
2957 1601 7410 T15, SKP /VERIFY EMPTY BUFFER 12-BIT MODE
2958 /
2959 /THE PURPOSE OF THESE TESTS IS TO VERIFY THE CONTENTS OF THE SECTOR BUFFER AFTER
2960 /THE PREVIOUS FILL BUFFER 8-BIT MODE TEST, AND THE PREVIOUS FILL BUFFER
2961 /12-BIT MODE TEST.
2962 /
2963 /EMPTY BUFFER 8-BIT MODE
2964 /
2965 1602 1377 T17, TAD (40
2966 /
2967 /EMPTY BUFFER 12-BIT MODE
2968 /
2969 1603 7105 T14, CLL IAC RAL
2970 1604 4437 LCD /ISSUE THE COMMAND 2 OR 102
2971 1605 1376 TAD (100)
2972 1606 0112 AND COMMAND
2973 1607 7640 SZA CLA
2974 1610 7307 CLL CLA IAC RTL
2975 1611 1375 TAD (6000)
2976 1612 3132 DCA TESTP
2977 1613 4455 GETAPATTERN / COUNT PATTERN ( PATTERN #6)
2978 1614 1374 TAD (RBUFFER-1)
2979 1615 3010 DCA A10 /PROGRAMS "READ" BUFFER
2980 1616 1373 TAD (-RBUFFER+1)
2981 1617 3133 DCA XA10 / -STARTING ADDRESS OF READ BUFFER
2982 1620 1010 TAD A10
2983 1621 3011 DCA A11 /
2984 1622 4772 EB, JMS FBEB
2985 1623 5230 JMP EBCOMPARE
2986 1624 3410 DCA I A10
2987 1625 4442 XDRIN
2988 1626 3411 DCA I A11 / AND "EMPTY THE BUFFER"
2989 1627 5222 JMP EB /
2990 /COMPARE THE CONTENTS OF THE SECTOR BUFFER
2991 /
2992 /WITH THE GOOD DATA IN "WBUFFER"
2993 /
2994 1630 1371 EBCOMPARE, TAD (WBUFFER-1)
2995 1631 3133 DCA XA10 /EXPECTED
2996 1632 1374 TAD (RBUFFER-1)
2997 1633 3134 DCA XA11 /ACTUAL
2998 1634 3113 DCA COMPERROR / # 1 IF COMPARE ERROR
2999 /

```

```

3000 1635 2133 EBLOOP, ISZ XA10
3001 1636 2134 ISZ XA11
3002 1637 7100 CLL
3003 1640 1376 TAD (100)
3004 1641 0112 AND COMMAND
3005 1642 7640 SZA CLA
3006 1643 7120 STL /SET LINK IF 8-BIT MODE
3007 1644 1533 TAD I XA10
3008 1645 7430 SZL
3009 1646 0370 AND (377)
3010 1647 3164 DCA GOOD
3011 1650 1164 TAD GOOD
3012 1651 7041 CIA
3013 1652 1534 TAD I XA11
3014 1653 7650 SNA CLA
3015 1654 5263 JMP EBOX
3016 /A COMPARE ERROR HAS OCCURED
3017 /
3018 /INCORRECT DATA WAS TRANSFERRED FROM THE RX01 CONTROL SECTOR BUFFER
3019 /TO THE RX8 INTERFACE AND SAVED WITHIN PPROGRAM LOCATIONS BEGINNING WITH "RBUFFER"
3020 /
3021 /THAT DATA, HOWEVER, MAY HAVE BEEN TRANSFERRED INCORRECTLY *TO*
3022 /THE RX01 CONTROL FROM THE RX8 INTERFACE PREVIOUSLY WITHIN T12, OR T15
3023 /
3024 1655 1367 TAD (-WBUFFER)
3025 1656 1133 TAD XA10
3026 1657 3166 DCA BLANK
3027 1660 1534 TAD I XA11 /ACTUAL WORD/BYTE FROM SECTOR BUFFER
3028 1661 4452 E140, ERROR /IS NOT = EXPECTED
3029 1662 4502 SCOPE
3030 1663 1376 EBOX, TAD (100)
3031 1664 0112 AND COMMAND
3032 1665 1376 TAD (100)
3033 1666 7041 CIA
3034 1667 1133 TAD XA10
3035 1670 1366 TAD (-WBUFFER+1)
3036 1671 7640 SZA CLA
3037 1672 5235 JMP EBLOOP
3038 /END OF TESTS 14, 15, 17, OR 20
3039 /
3040 /...ALSO
3041 /
3042 /END OF TESTS 31, 32, OR 33
3043 /
3044 1673 5453 EXIT
3045 /
3046 /SECTOR BUFFER DATA TESTING
3047 /
3048 /THE PURPOSE OF THESE TESTS IS TO VERIFY THAT ALL 1'S AND ALL 0'S CAN BE
3049 /SET INTO THE SECTOR BUFFER
3050 /
3051 /SECTOR BUFFER DATA TESTING
3052 /
3053 /FILL THE SECTOR BUFFER WITH ALL 1'S
3054 /

```

```

3055 1674 1370 T22, TAD (377) / "GOOD" = ALL 1'S
3056 /
3057 /FILL THE SECTOR BUFFER WITH ALL 0'S
3058 /
3059 /FILL THE SECTOR BUFFER WITH 128 BYTES OF "GOOD"
3060 /
3061 1675 4765 T21, JMS FB128BYTES / "GOOD" = ALL 0'S
3062 1676 4764 JMS IX
3063 /*****
3064 /*****
3065 /
3066 /IF THIS IS TEST #22
3067 /
3068 /THEN TEST FOR A DRIVE SELECTION
3069 /
3070 /IF NO DRIVES ARE ENABLED WITHIN PROGRAM LOCATION "DTESTP"
3071 /
3072 /THEN THERE ARE NO MORE RX01 CONTROL TESTS TO EXECUTE
3073 /
3074 1677 1167 TAD TEST
3075 1700 1363 TAD (-T22)
3076 1701 7640 SZA CLA
3077 1702 5453 EXIT / END OF TEST 21
3078 1703 1762 TAD UNITS
3079 1704 7640 SZA CLA
3080 1705 5453 EXIT / END OF TEST 22
3081 0300 XC=0300
3082 /
3083 / END OF PASS " C "
3084 /
3085 1706 7340 CLL CLA CMA
3086 1707 3761 DCA COUNT /INIT TIMING FOR APT IF ONLY
3087 /INTERFACE IS TO BE TESTED.
3088 1710 4760 JMS XTICK
3089 1711 4464 LOCKUP
3090 1712 1357 TAD (XC)
3091 1713 5756 JMP NOMORETESTS
3092 /*****
3093 /*****
3094 1714 7327 T24, CLA STL IAC RTL
3095 1715 5755 JMP IRDWR / 6 (READ)
3096 1716 7307 T25, CLL CLA IAC RTL
3097 1717 5755 JMP IRDWR / 4 (WRITE)
3098 1720 1354 T26, TAD (14)
3099 1721 5755 JMP IRDWR / 14 (WRITE DELETED DATA)
3100 /
3101 /THE PURPOSE OF THESE TESTS IS TO VERIFY
3102 /THE RX01 CONTROL CLOCK SET OF THE ERROR FLAG
3103 /
3104 /BY FORCING A SEEK ERROR TO OCCUR
3105 / (ATTEMPTING TO PERFORM A FUNCTION ON A NON-EXISTANT SECTOR #0 )
3106 /
3107 /NOTE;THE CONTENTS OF THE SECTOR BUFFER SHOULD REMAIN UNCHANGED AND CONTAIN
3108 / THE PATTERN OF ALL 1'S AS FILLED WITHIN TEST 22 BECAUSE THE READ SHOULD
3109 / HAVE NEVER OCCURED.

```

```

3110 /
3111 / 1. THE STATUS WITHIN THE TRANSFER REGISTER AT ERROR SHOULD = 0
3112 / 2. THE R-CODE STATUS SHOULD = 70 (UNABLE TO FIND SECTOR)
3113 / 3. THE STATUS FROM THE RST COMMAND (12) SHOULD = DRIVE READY (200)
3114 /
3115 /THE LENGTH ( # OF TRANSFERS TO THE RX01 CONTROL) SHOULD BE TWO
3116 / (ONE EACH FOR THE SECTOR AND FOR THE TRACK)
3117 /
3118 /GET ONE UNIT ONLY (UNIT 1 IF UNIT 0 IS NOT READY)
3119 /
3120 /ROUTINE TO WAIT FOR SKIP ON AN IOT. IF SKIP DOES NOT OCCUR
3121 /THE ROUTINE WILL PRINT PC POINT IN ERROR AND GO BACK ABOUT ITS
3122 /BUSINESS.
3123 /
3124 1722 0000 XWAIT, 0
3125 1723 1006 TAD 6
3126 1724 7650 SNA CLA
3127 1725 5330 JMP .+3
3128 1726 7240 STA
3129 1727 3006 DCA 6
3130 1730 2151 ISZ H1
3131 1731 5722 JMP I XWAIT
3132 1732 2116 ISZ HANGER
3133 1733 5722 JMP 1 XWAIT
3134 1734 1322 TAL XWAIT /GET ERROR PC
3135 1735 1353 TAD (-E3PRE
3136 1736 7710 SPA CLA /DID CALL COME FROM PRETEST.
3137 1737 5752 JMP E3PRE /YES. REPPORT ERROR.
3138 1740 1322 TAD XWAIT /GET BACK ERROR
3139 1741 5751 JMP HUNGUP
3140 /
3141 1751 3341
3142 1752 0423
3143 1753 7355
3144 1754 0014
3145 1755 2000
3146 1756 0522
3147 1757 0300
3148 1760 4127
3149 1761 4146
3150 1762 4235
3151 1763 6104
3152 1764 2303
3153 1765 4541
3154 1766 0666
3155 1767 0665
3156 1770 0377
3157 1771 7112
3158 1772 1476
3159 1773 0466
3160 1774 7312
3161 1775 6000
3162 1776 0100
3163 1777 0040
3164 2000

```

PAGE

```

3164 2000 3112 IRDWR, DCA COMMAND
3165 2001 3777 DCA WUNITS
3166 2002 4460 GETUNIT
3167 2003 1112 TAD COMMAND
3168 2004 1776 TAD UNIT
3169 2005 3160 DCA BLANK
3170 2006 1166 TAD BLANK
3171 2007 4437 LCD / 6, 4, OR 14
3172 2010 3165 DCA EAC
3173 2011 5214 JMP .+3
3174 2012 2165 ISZ EAC
3175 2013 4443 XDROUT
3176 2014 4444 STR
3177 2015 7410 SKP
3178 2016 5212 JMP .-4 /SECTOR 0 = TRACK 0
3179 2017 4446 SDN
3180 2020 5214 JMP .-4 /WAIT FOR DONE FLAG
3181 2021 7344 CLL STA PAL
3182 2022 1165 TAD EAC
3183 2023 7650 SNA CLA
3184 2024 5231 JMP .+5
3185 2025 7305 CLL CI-A IAC PAL
3186 2026 3164 DCA GOOD / 2 TRANSFERS WERE EXPECTED
3187 2027 1165 TAD EAC
3188 2030 4452 E240, ERROR / # OF TRANSFER REQUEST FLAGS NOT OK
3189 2031 4503 SUBSCOPE / (AC) = # OF TRANSFERS OCCURED
3190 /
3191 /THE ERROR FLAG SHOULD = 1
3192 /
3193 2032 4445 SER
3194 2033 4452 E245, ERROR / MISSING ERROR FLAG
3195 2034 4503 SUBSCOPE
3196 /IF THIS IS T24, THEN DELETED DATA [MAY] BE SET (BUT THAT'S OK FOR NOW),
3197 /THE STATUS AT ERROR SHOULD = X (100 MAYBE DELETED DATA)+200 DRIVE READY
3198 /
3199 /IF THIS IS T25, THEN DELETED DATA [SHOULD NOT] BE SET, THEREFORE
3200 /THE CONTENTS OF THE TRANSFER REGISTER (THE STATUS AT THE ERROR) SHOULD = 200
3201 /
3202 /IF THIS IS T26, THEN DELETED DATA [MUST] BE SET, THEREFORE
3203 /THE STATUS SHOULD = 300 (200 DRIVE READY)+(100 (DELETED DATA)
3204 /
3205 2035 1167 TAD TEST /FOR T #
3206 2036 1375 TAD (-T25)
3207 2037 7650 SNA CLA
3208 2040 5245 JMP .+5 / T25 THEN * JMP .+5 *
3209 2041 1167 TAD TEST
3210 2042 1374 TAD (-T26)
3211 2043 7650 SNA CLA
3212 2044 1373 TAD (100) / T26 MEANS EXPECT DELETED DATA
3213 2045 3164 DCA GOOD / T24
3214 2046 4442 XDRIN /ACTUAL STATUS
3215 2047 3166 DCA BLANK
3216 2050 1166 TAD BLANK
3217 2051 3165 DCA EAC
3218 /

```

```

3219 /IF THIS IS T24, THEN DELETED DATA [MAY] BE SET, (BUT THAT'S OK FOR NOW)
3220 /
3221 TAD TEST / FOR T #
3222 TAD (-T24)
3223 SZA CLA / T24 ?
3224 JMP .+4
3225 TAD (-100-1) / YES
3226 AND BLANK /ACTUAL STATUS MINUS DELETED DATA
3227 DCA EAC
3228 TAD EAC
3229 CIA
3230 TAD GOOD /EXPECTED
3231 SNA CLA
3232 JMP .+3
3233 TAD EAC
3234 E241, ERROR /STATUS NOT = "GOOD"
3235 2070 4503 SUBSCOPE
3236 /
3237 /THE B-CODE SHOULD = 70 (UNABLE TO FIND SECTOR)
3238 /
3239 RSTB /RETURN WITH AC = CODE
3240 TAD (-70)
3241 SNA CLA
3242 JMP .+5
3243 TAD (70)
3244 DCA GOOD
3245 TAD EAC
3246 E242, ERROR /B-CODE STATUS NOT = CODE # 70
3247 2101 4502 SCOPE
3248 /
3249 /THE CONTENTS TO THE SECTOR BUFFER SHOULD REMAIN UNCHANGED
3250 /
3251 /THE CONTENTS OF THE SECTOR BUFFER SHOULD = ALL BYTES OF 1'S
3252 /
3253 TAD (377)
3254 DCA GOOD /EXPECT ALL 1'S
3255 JMS TX /VERIFY SECTOR BUFFER SUBROUTINE
3256 EXIT / END OF TEST 24, 25, 26
3257 /
3258 / TEST 27 - SEEK AND CRC VERIFICATION (FIRST PROGRAMMED HEAD MOVEMENT)
3259 /
3260 /THE PURPOSE OF THIS TEST IS TO VERIFY THAT SUPPLYING THE RX01 WITH A TRACK
3261 /WHOSE VALUE IS GREATER THAN 114 (OCTAL) EXPECTS A B-CODE ERROR OF 40
3262 /
3263 T27, TAD (115)
3264 DCA BLANK
3265 STL CLA IAC RTL / ISSUE READ COMMAND
3266 LCD
3267 DCA EAC
3268 SKP
3269 ISZ EAC / + TO TRANSFER COUNT
3270 TAD BLANK / SECTOR TRACK
3271 XDROUT /TO RX01 CONTROL
3272 STR / WAIT FOR TRANSFER REQUEST FLAG
3273 SKP
3274 JMP .-5

```

```

3274 2122 4446 SDN / WAIT FOR DONE FLAG
3275 2123 5317 JMP .-4
3276 LCA 6 /WAIT POINTER
3277 2125 7344 CLL STA RAL / ONLY 2 TRANSFER REQUESTS WERE EXPECTED
3278 2126 1165 TAD EAC / ACTUAL # OF TRANSFER REQUEST OCCURED
3279 2127 7650 SNA CLA
3280 JMP .+5
3281 2131 7305 CLL CLA IAC RAL
3282 2132 3164 DCA GOOD
3283 2133 1165 TAD EAC
3284 2134 4452 E270, ERROR / # OF TRANSFER REQUEST FLAGS NOT OK
3285 2135 4503 SUBSCOPE / (AC) CONTAINS THE # OF REQUEST OCCURED
3286 /
3287 /THE ERROR FLAG SHOULD = 1
3288 /
3289 2136 4445 SER
3290 2137 4452 E271, ERROR / MISSING ERROR FLAG
3291 2140 4503 SUBSCOPE
3292 /THE B-CODE SHOULD = 40
3293 /
3294 2141 4477 RSTB
3295 2142 1363 TAD (-40)
3296 2143 7650 SNA CLA
3297 2144 5351 JMP .+5
3298 2145 1362 TAD (40)
3299 2146 3164 DCA GOOD
3300 2147 1165 TAD EAC
3301 2150 4452 E272, ERROR / B-CODE NOT = 40
3302 2151 4502 SCOPE
3303 /
3304 /THE CONTENTS OF THE SECTOR BUFFER SHOULD REMAIN UNCHANGED
3305 /
3306 2152 1366 TAD (377)
3307 2153 3164 DCA GOOD / EXPECT ALL 1'S
3308 2154 4765 JMS TX
3309 /
3310 2155 5453 EXIT / END OF TEST 27
3311 2162 0040
3312 2163 7740
3313 2164 0115
3314 2165 2303
3315 2166 0377
3316 2167 0070
3317 2170 7710
3318 2171 7677
3319 2172 6064
3320 2173 0100
3321 2174 6060
3322 2175 6062
3323 2176 4242
3324 2177 4236
3325 2200 PAGE
3326 /TEST 30 - SEEK AND CRC VERIFICATION
3327 /
3328 /READ ALL SECTORS OF ALL TRACKS

```

```

3328 /
3329 /IF THE DATA IS OF KNOWN QUALITY THEN MONITOR FOR CRC ERRORS
3330 /
3331 2200 3132 T30, DCA TESTP
3332 2201 5777* JMP TEST4
3333 /TEST 31 - FIRST WRITE EVER
3334 /
3335 /WRITING TO THE DISK SHOULD NOT DESTROY THE CONTENTS OF THE SECTOR BUFFER
3336 /
3337 /TECHNICAL NOTE:
3338 /
3339 /THIS TEST WRITES ON ONE UNIT ONLY - UNIT 0, BUT IF UNIT 0 WAS
3340 /NOT SELECTED BY THE OPERATOR AT THE START OF THIS PROGRAM
3341 /THEN THIS TEST WRITES ON UNIT 1
3342 /
3343 /PROGRAMMING NOTE:
3344 /
3345 /THIS "DCA WUNITS" IS NECESSARY IF THIS TEST IS LOCKED WITH SWS
3346 /
3347 2202 3776* T31, DCA WUNITS
3348 /
3349 2203 7307 CLL CLA IAC RTL / 4 (WRITE 8-BIT MODE)
3350 2204 1375 TAD (6000)
3351 2205 3132 DCA TESTP /COUNT PATTERN 6
3352 2206 7240 STA
3353 2207 3120 DCA RDC
3354 / "GETATRACK"
3355 /
3356 2210 1031 TAD OD
3357 2211 3131 DCA TARGET / TRACK = (OD)
3358 / "GETASECTOR"
3359 /
3360 2212 1033 TAD FIRST
3361 2213 3124 DCA STARGET / SECTOR = (FIRST)
3362 /GET UNIT 0 (OR UNIT 1 IF UNIT 0 NOT SELECTED)
3363 /
3364 2214 4460 GETUNIT
3365 /
3366 / "INITSECTOR"
3367 /
3368 2215 7240 STA
3369 2216 3122 DCA SECTORS / 1 SECTOR (FIRST) TO WRITE
3370 2217 1374 TAD (+3)
3371 2220 3773* DCA XWRITE / RETURN ADDRESS FROM WRITE SUBROUTINE
3372 2221 5772* JMP REWRITE+1 / JMP TO WRITE SUBROUTINE
3373 /
3374 /RETURN HERE FROM SUBROUTINE " XWRITE "
3375 /
3376 / JMP TO T17 TO VERIFY THE CONTENTS OF THE SECTOR BUFFER
3377 /
3378 2222 5771* JMP T17
3379 /TEST 32 - INIT [PROGRAMMED] PART III / IMPLIED READ OF TRACK 1 SECTOR 1
3380 /
3381 /*****
3382 /*****

```

```

3383 /
3384 /IF THE CONTENTS OF PROGRAM LOCATIONS:
3385 /
3386 / " OD " = 1, AND
3387 / " FIRST " = 1,
3388 /
3389 /AND IF UNIT 0 WAS SELECTED BY THE OPERATOR AT THE START OF THIS PROGRAM
3390 /
3391 /THEN EXECUTE THIS TEST
3392 /
3393 2223 1033 T32, TAD FIRST
3394 2224 7110 CLL RAR
3395 2225 7640 SZA CLA
3396 2226 5453 NOTEST
3397 2227 1031 TAD OD
3398 2230 7110 CLL RAR
3399 2231 7640 SZA CLA
3400 2232 5453 NOTEST
3401 2233 1770* TAD UNITS
3402 2234 7700 SMA CLA
3403 2235 5453 NOTEST
3404 /
3405 /*****
3406 /*****
3407 /
3408 /THE PURPOSE OF THIS TEST IS TO VERIFY THE "IMPLIED READ" OF TRACK 1 SECTOR 1
3409 /AS PART OF THE INITIALIZE FUNCTION
3410 /
3411 /FIRST FILL THE SECTOR BUFFER WITH ALL 0'S, THEN ISSUE IOT INIT 67X7
3412 /
3413 /INIT SHOULD READ THE CONTENTS OF TRACK 1 SECTOR 1 OF UNIT 0
3414 /
3415 /INTO THE SECTOR BUFFER
3416 /
3417 2236 4767* JMS FB120BYTES / FILL THE SECTOR BUFFER
3418 2237 4450 INIT /OF UNIT 0
3419 /
3420 /JMP TO TEST 17 TO VERIFY THE CONTENTS OF THE SECTOR BUFFER
3421 /
3422 2240 5771* JMP T17 / TO TEST 17
3423 /TEST 33 - FIRST READ [PROGRAMMED] EVER
3424 /
3425 /FIRST FILL THE SECTOR BUFFER WITH ALL 0'S
3426 /
3427 /THEN READ FROM THE DISK TRACK # (OD), SECTOR # (FIRST)
3428 /
3429 /THE CONTENTS OF THE SECTOR BUFFER SHOULD BE THAT OF THE PREVIOUS TEST
3430 /
3431 /PROGRAMMING NOTE:
3432 /
3433 /THIS "DCA WUNITS" IMPERATIVE IF T27 WAS EXECUTED PREVIOUSLY THIS PASS
3434 /
3435 2241 3776* T33, DCA WUNITS
3436 2242 4767* JMS FB120BYTES / FILL THE SECTOR BUFFER
3437 2243 7307 CLL CLA IAC RTL

```

```

3438 2244 3132 DCA TESTP / 4 (READ 8-BIT MODE)
3439 /PROGRAMMING NOTE:
3440 /
3441 /THE FOLLOWING CODE TO " JMP T17 " IS IMPERATIVE HOUSEKEEPING PRIMING THE
3442 / " READ " SUBROUTINE
3443 /
3444 2245 7240 STA
3445 2246 3120 DCA RDC
3446 / "GETATRACK"
3447 /
3448 2247 1031 TAD OD
3449 2250 3131 DCA TARGET
3450 / "GETASECTOR"
3451 /
3452 2251 1033 TAD FIRST
3453 2252 3124 DCA STARGET
3454 /GET ONE UNIT ONLY (UNIT 1 IF UNIT 0 IS NOT READY)
3455 /
3456 2253 4460 GETUNIT / SELECT A UNIT
3457 2254 7240 STA
3458 2255 3136 DCA RIRETPY /SO NO "REWRITES" OCCUR
3459 / "INITSECTOR"
3460 /
3461 2256 7240 STA
3462 2257 3122 DCA SFACTORS / 1 SECTOR TO READ (#FIRST)
3463 2260 1366 TAD (.+4)
3464 2261 3765* DCA XREAD /RETURN ADDRESS FROM " READ " SUBROUTINE
3465 2262 5764* JMP RFARETRY / JMP TO READ SUBROUTINE
3466 /THIS "WRITE" IS IMPERATIVE FOR REFERENCES WITHIN "XREAD"
3467 /
3468 2263 4510 WRITE / "WRITE" FOR PROGRAM REFERENCES ONLY
3469 /
3470 /RETURN HERE FROM SUBROUTINE " XREAD "
3471 /
3472 /
3473 /JMP TO TEST 17 TO VERIFY THE CONTENTS OF THE SECTOR BUFFER
3474 /
3475 2264 5771* JMP T17 / TO TEST 17
3476 /TEST 23 - DRIVE READY SELECTION (ALL UNITS SELECTED BY OPERATOR)
3477 /
3478 /THE PURPOSE OF THIS TEST IS TO VERIFY THE "SEL DRV RDY" STATUS BIT 4
3479 /
3480 2265 1363 T23, TAD (200)
3481 2266 3166 DCA BLANK
3482 2267 4460 GETUNIT
3483 2270 1762* TAD UNIT
3484 2271 4476 RST
3485 2272 4454 DONE
3486 2273 5265 JMP T23
3487 2274 5453 EXIT / END OF TEST 27
3488 /
3489 /THE PURPOSE OF THESE TESTS IS TO WRITE-READ-AND PROGRAM VERIFY THE DATA
3490 /ON ALL TRACKS FROM (OD) TO (ID), AND ALL SECTORS FROM (FIRST) TO (LAST),
3491 /EXERCISING 8-BIT MODE, 12-BIT MODE, AND DELETED DATA
3492 /

```

```

3493 /A PATTERN OF ALL 1'S ARE WRITTEN ON THE DISK
3494 /
3495 / WORD/BYTE 1 IS THE TRACK ID (BITS 5 THRU 11)
3496 / WORD/BYTE 2 IS THE SECTOR ID (BITS 7 THRU 11)
3497 /
3498 / WORDS 3 THRU 62 IS THE ALL 1'S DATA
3499 /
3500 / BYTES 3 THRU 126 IS THE ALL 1'S DATA
3501 /
3502 / WORDS 63, AND 64 ARE SPECIAL SUMCHECK WORDS
3503 /
3504 / BYTES 127, AND 128 ARE SPECIAL SUMCHECK WORDS
3505 /
3506 /TEST 37 - 8 BIT MODE WITH DELETED DATA
3507 /
3508 2275 7305 T37, CLL CLA IAC RAL / 6 (DELETED DATA - 8/BIT MODE)
3509 /
3510 /TEST 35 - 8 BIT MODE
3511 /
3512 2276 1361 T35, TAD (2) / 4 (8/BIT MODE)
3513 /
3514 /TEST 36 - 12 BIT MODE WITH DELETED DATA
3515 /
3516 2277 1361 T36, TAD (2) / 2 (DELETED DATA - 12/BIT MODE)
3517 /
3518 /TEST 34 - 12 BIT MODE
3519 /
3520 2300 1360 T34, TAD (1000) /ALL 1'S PATTERN
3521 2301 3132 DCA TESTP / (12/BIT MODE)
3522 2302 5757* JMP THESTP
3523 /THIS SUBROUTINE IS ENTERED FROM TESTS: *** T21, T22, (T24, T25, T26), T27
3524 /
3525 /EMPTY THE BUFFER TO VERIFY THE CONTENTS = ALL 1'S OR ALL 0'S
3526 /
3527 2303 2303 TX,
3528 2304 3166 DCA BLANK / BYTE # 1 TO 128
3529 2305 1356 TAD (102)
3530 2306 4437 LCO /EMPTY BUFFER 8-BIT MODE
3531 2307 5322 JMP T20STR-1
3532 2310 4442 T20XDRIN, XDRIN
3533 2311 3165 DCA EAC /DATA FROM SECTOR BUFFER
3534 2312 1165 TAD EAC
3535 2313 7041 CIA
3536 2314 1164 TAD GOOD /COMPARED WITH EXPECTED DATA
3537 2315 7650 SNA CLA
3538 2316 5321 JMP .+3
3539 2317 1165 TAD EAC
3540 2320 4452 E211, ERROR /DATA "TO" NOT = DATA "FROM"
3541 2321 4503 SUBSCOPE
3542 2322 2166 ISZ BLANK / 1 TO 128
3543 2323 4444 T20STR, STR
3544 2324 7410 SKP
3545 2325 5310 JMP T20XDRIN
3546 2326 4446 SDN
3547 2327 5323 JMP T20STR

```

```

3548 2330 4445 SER
3549 2331 4465 OK
3550 2332 4452 E712, ERROR /UNEXPECTED ERROR FLAG
3551 2333 4502 SCOPE
3552 2334 5703 JMP I TX
3553 /
3554 /
3555 /
3556 /
3557 2356 0102
3558 2357 3000
3559 2360 1000
3560 2361 0002
3561 2362 4242
3562 2363 0200
3563 2364 3423
3564 2365 3414
3565 2366 2264
3566 2367 4541
3567 2370 4235
3568 2371 1602
3569 2372 3207
3570 2373 3200
3571 2374 2222
3572 2375 6000
3573 2376 4236
3574 2377 2510
3575 2400

```

```

PAGE
/READ THE B-CODE STATUS
/
/THIS SUBROUTINE IS ENTERED FROM TESTS: *** ALT12, (T24, T25, T26), T27
/

```

```

3579 2400 2400 XRSTB, .
3580 2401 1377 TAD (16)
3581 2402 4437 LCD /ISSUE COMMAND # 7
3582 2403 4444 STR
3583 2404 4465 OK
3584 2405 4452 E700, ERROR /UNEXPECTED TRANSFER REQUEST FLAG
3585 2406 4503 SUBSCOPE
3586 2407 4446 SDN
3587 2410 5203 JMP .-5
3588 2411 4442 XDRIIN
3589 2412 3165 DCA EAC /ACTUAL STATUS
3590 2413 4145 SER
3591 2414 5217 JMP .+3
3592 2415 1165 TAD EAC
3593 2416 4452 E7001, ERROR /UNEXPECTED ERROR FLAG
3594 2417 4503 SUBSCOPE
3595 2420 1165 TAD EAC
3596 2421 0376 AND (7000) /BITS 0,1,2 ARE RESIDUAL FROM COMMAND SHIFT
3597 2422 1375 TAD (1000).
3598 2423 7650 SNA CLA
3599 2424 5231 JMP .+5
3600 2425 1376 TAD (7000)
3601 2426 3164 DCA GOOD

```

```

3602 2427 1165 TAD EAC
3603 2430 4452 E7002, ERROR /SHIFT REGISTER NOT SHIFTING OK
3604 2431 4503 SUBSCOPE
3605 2432 1165 TAD EAC
3606 2433 0374 AND (377)
3607 2434 3165 DCA EAC
3608 2435 1165 TAD EAC
3609 2436 5600 JMP I XRSTB

```

```

/
/READ STATUS SUBROUTINE
/
/THE CONTENTS OF THE AC AT ENTRY = BIT 7 (UNIT SELECT)
/
/READ THE STATUS AT DONE BY ISSUING COMMAND # 5
/
/THIS SUBROUTINE IS ENTERED FROM TESTS: *** ALT12, AND T27
/

```

```

3610 /
3611 /
3612 /
3613 /
3614 /
3615 /
3616 /
3617 /
3618 /
3619 2437 2437 XRST, .
3620 2440 1373 TAD (12)
3621 2441 4437 LCD /ISSUE COMMAND # 5 (AC AT ENTRY = UNIT)
3622 2442 4444 STR
3623 2443 4465 OK
3624 2444 4452 E7003, ERROR /UNEXPECTED TRANSFER REQUEST FLAG
3625 2445 4503 SUBSCOPE
3626 2446 4446 SDN
3627 2447 5242 JMP .-5 /WAIT FOR DONE FLAG
3628 2450 4442 XDRIIN
3629 2451 3165 DCA EAC /ACTUAL STATUS
3630 2452 4445 SER
3631 2453 5256 JMP .+3
3632 2454 1165 TAD EAC
3633 2455 4452 E7004, ERROR /UNEXPECTED ERROR FLAG
3634 2456 4503 SUBSCOPE

```

```

/
/THE PURPOSE OF THIS SUBTEST IS TO VERIFY THE SHIFTING OF THE RX8 INTERFACE
/
/TRANSFER REGISTER BY THE RX01 MICROCONTROLLER
/
/BITS 0, 1, AND 2 OF THE RX8 INTERFACE TRANSFER REGISTER SHOULD BE REMNANTS
/
/OF THE PREVIOUS COMMAND (COMMAND # 5), THEREFORE BITS 0, 1, AND 2 SHOULD = 101 (BINARY)
/

```

```

3644 2457 1165 TAD EAC
3645 2460 0376 AND (7000)
3646 2461 1372 TAD (3000)
3647 2462 7650 SNA CLA
3648 2463 5270 JMP .+5
3649 2464 1371 TAD (5000)
3650 2465 3164 DCA GOOD
3651 2466 1165 TAD EAC
3652 2467 4452 E7006, ERROR
3653 2470 4503 SUBSCOPE
3654 /
3655 /THE CONTENTS OF THE RST STATUS SHOULD = DRIVE READY (200)
3656 /

```

```

3657 / (IF UNIT-5 WERE SELECTED AT L/S 200 OTHERWISE THE STATUS SHOULD = 0 )
3658 /
3659 2471 1166 TAD BLANK / (GOOD) SAVED PRIOR TO ENTRY INTO " XRST "
3660 2472 0370 AND (-4-1) / MINUS " INIT DONE " BIT
3661 2473 3164 DCA GOOD
3662 2474 1165 TAD EAC / ACTUAL STATUS MINUS " DELETED DATA "
3663 2475 0367 AND (277)
3664 2476 3166 DCA BLANK
3665 2477 1164 TAD GOOD /EXPECTED
3666 2500 7041 CIA
3667 2501 1166 TAD BLANK /ACTUAL (MASSAGED)
3668 2502 7650 SNA CLA
3669 2503 5306 JMP +3
3670 2504 1165 TAD EAC / ACTUAL ACTUAL STATUS
3671 2505 4452 E7005, ERROR /EXPECTED STATUS NOT = (GOOD)
3672 2506 4502 SCOPE
3673 2507 5637 JMP I XRST
3674 /SUB-TEST SELECTIONS
3675 /
3676 /
3677 / TEST 0 = " THE TEST "
3678 / TEST 001 = WRITE - READ (PARITY CHECK)
3679 / TEST 010 = WRITE - READ - READ CHECK (PROGRAM VERIFY)
3680 / TEST 011 = READ - READ CHECK (COMPATABILITY)
3681 / TEST 100 = READ (PARITY CHECK)
3682 / TEST 5 = WRITE ONLY
3683 / TEST 110 = EMPTY BUFFER
3684 / TEST 111 =
3685 /
3686 /
3687 2510 1366 TEST4, TAD (READ)
3688 2511 7410 SKP
3689 2512 1365 TEST3, TAD (READCOMPARE)
3690 2513 3333 DCA DOB
3691 2514 1376 TAD (NOP)
3692 2515 5325 JMP DCAD0A
3693 2516 1365 TEST2, TAD (READCOMPARE)
3694 2517 7410 SKP
3695 2520 1366 TEST1, TAD (READ)
3696 2521 7410 SKP
3697 2522 1376 TEST5, TAD (NOP)
3698 2523 3333 DCA DOB
3699 2524 1364 TAD (WRITE)
3700 2525 3332 DCAD0A, DCA DOA
3701 /
3702 2526 4455 TESTX, GETAPATTERN
3703 2527 4463 INITTRACKS
3704 2530 4460 GETUNIT
3705 2531 4457 TESTXL, GETATRACK
3706 2532 4510 DOA, WRITE
3707 2533 4474 DOB, READ
3708 2534 2145 ISZ TRACKS
3709 2535 5331 JMP ILSXL
3710 2536 4454 DONE
3711 2537 5326 JMP TESTX

```

```

3712 2540 5453 EXIT
3713 /
3714 /ROUTINE TO DETERMINE IF ON APT.
3715 /
3716 2541 0000 XCHK22, 0
3717 2542 1022 TAD 22
3718 2543 0363 AND (4000)
3719 2544 7640 SZA CLA
3720 2545 5741 JMP I XCHK22
3721 2546 2341 ISZ XCHK22
3722 2547 5741 JMP I XCHK22
3723 2563 4000
3724 2564 4510
3725 2565 4475
3726 2566 4474
3727 2567 0277
3728 2570 7773
3729 2571 5000
3730 2572 3000
3731 2573 0012
3732 2574 0377
3733 2575 1000
3734 2576 7000
3735 2577 0016
3736 PAGE
3737 /AC SW 6 = 1 TO HALT AT END OF TEST
3738 /
3739 2600 2600 XLOCKUP, .
3740 2601 4570 LAS
3741 2602 0377 AND (SW7)
3742 2603 7640 SZA CLA
3743 2604 4461 HLT
3744 /AC SW 5 = 1 TO LOCK SCOPE LOOP ON TEST
3745 /
3746 2605 4570 LAS
3747 2606 0376 AND (SW5)
3748 2607 7640 SZA CLA
3749 2610 5567 JMP I TEST
3750 2611 5600 JMP I XLOCKUP
3751 /
3752 /BYTE SWAP SUBROUTINE
3753 /
3754 /THE CONTENTS OF THE AC AT ENTRY WILL BE SWAPPED
3755 /
3756 2612 2612 XBSW, .
3757 2613 3235 DCA BSWAC
3758 2614 7010 RAR
3759 2615 3236 DCA BSWLINK
3760 2616 1375 TAD (-6)
3761 2617 3234 DCA BSWRAL
3762 2620 1235 TAD BSWAC
3763 2621 7100 CLL
3764 2622 7510 SPA
3765 2623 7120 STL
3766 2624 7004 RAL

```



```

3766 2625 2234 ISZ BSWPAL
3767 2626 5221 JMP .-5
3768 2627 3235 DCA BSWAC
3769 2630 1236 TAD BSWLINK
3770 2631 7104 CLL RAL
3771 2632 1235 TAD BSWAC
3772 2633 5612 JMP I XBSW
3773 2634 7772 BSWRAL, -6
3774 2635 0000 BSWAC, 0
3775 2636 0000 BSWLINK, 0
3776 2637 2637 XERROR, .
3777 2640 3165 DCA EAC
3778 2641 1165 TAD EAC
3779 2642 4467 AERROR /GET ERROR PC.
3780 2643 2111 ISZ BUSY
3781 2644 1237 TAD XERROR
3782 2645 7001 IAC
3783 2646 3774* DCA EPCSCOPE / ERROR RETURN ADDRESS + 1
3784 2647 7301 CLL CLA IAC
3785 2650 3773* DCA ERRORS
3786 /IF AC SW 3 = 0 THEN PRINT AN ERROR MESSAGE
3787 /
3788 2651 4570 LAS
3789 2652 0372 AND (SW3)
3790 2653 7640 SZA CLA
3791 2654 5327 JMP XNOPRINT
3792 /IF THIS IS THE FIRST ERROR FOR THIS PASS THEN PRINT THE HEADER LINE
3793 /
3794 2655 2115 ISZ FIRSTERROR
3795 2656 5261 JMP .+3
3796 2657 4473 PRINT
3797 2660 6544 MXEHEADER
3798 2661 4473 PRINT
3799 2662 6001 MCRLF
3800 / EPR FAT FAST EAC GOOD PASS
3801 /
3802 / ERR = EPNR ADDRESS ( E # )
3803 / FAT = FIRST ADDRESS OF TEST
3804 / FAST = FIRST ADDRESS OF SUB-TEST
3805 /
3806 / EAC = CONTENTS OF THE ACCUMULATOR AT THE ERROR
3807 / PASS = PASS IN WHICH ERROR OCCURED
3808 /
3809 2663 7240 STA
3810 2664 1237 TAD XERROR
3811 2665 3156 DCA LSB
3812 2666 4505 TY4OCT
3813 2667 0156 LSB
3814 2670 4504 TAB
3815 2671 0005 5
3816 2672 1167 TAD TEST
3817 2673 3156 DCA LSB
3818 2674 4505 TY4OCT
3819 2675 0156 LSB
3820 2676 4504 TAB
    
```

```

3821 2677 0012 12
3822 2700 4505 TY4OCT
3823 2701 1364 PCSCOPE
3824 2702 4504 TAB
3825 2703 0017 17
3826 2704 4505 TY4OCT
3827 2705 0166 BLANK
3828 2706 4504 TAB
3829 2707 0024 24
3830 2710 4505 TY4OCT
3831 2711 0165 EAC
3832 2712 4504 TAB
3833 2713 0031 31
3834 2714 4505 TY4OCT
3835 2715 0164 GOOD
3836 2716 4504 TAB
3837 2717 0036 36
3838 2720 4436 TY8OCT
3839 2721 0161 PASS
3840 2722 4570 LAS
3841 2723 7700 SMA CLA
3842 2724 5327 JMP .+3
3843 2725 4504 TAB
3844 2726 0043 43
3845 BELL=707
3846 /
3847 2727 4570 XNOPRINT, LAS
3848 2730 0371 AND (SW11)
3849 2731 7640 SZA CLA
3850 2732 5335 JMP .+3
3851 2733 4501 SPECIALTYPEIT
3852 2734 0207 BELL
3853 2735 3115 DCA FIRSTERROR
3854 2736 4570 LAS
3855 2737 7710 SPA CLA
3856 2740 4461 HLT
3857 /
3858 2741 3111 DCA BUSY
3859 2742 6001 ION
3860 2743 5637 JMP I XERROR
3861 /
3862 /
3863 /ROUTINE TO DETERMINE WHETHER TO TEST DRIVE ZERO
3864 /OR IF NO DRIVES ARE TO BE TESTED.
3865 /
3866 2744 1770* TSTUNT, TAD UNITS /UNITS SELECTED BY OPERATOR
3867 2745 0367 AND (4000 /ISOLATE DRIVE ZERO.
3868 2746 7640 SZA CLA /IS DRIVE ZERO THERE.
3869 2747 5353 JMP DRVZERO /YES.
3870 2750 1366 TAD (-300-1) /DON'T TEST DRIVE ZERO.
3871 2751 3356 DCA COMP /IT IS NOT ANERROR IF DRIVE
3872 /ZERO IS OPERATIONAL.
3873 2752 5765* JMP NORX01-1
3874 2753 1364 DRVZERO, TAD (-100-1)
3875 2754 3356 DCA COMP
    
```

```

3876 2755 5763* JMP NORX01-2 /DRIVE ZERO THERE SO INDICATE IT.
3877 /
3878 2756 0000 COMP, 0
3879 /
3880 /
3881 2763 0425
3882 2764 7677
3883 2765 0426
3884 2766 7477
3885 2767 4000
3886 2770 4235
3887 2771 0001
3888 2772 0400
3889 2773 1363
3890 2774 1362
3891 2775 7772
3892 2776 0100
3893 2777 0020
3894 3000 PAGE
3895 3000 4455 /
3896 3001 4463 THETEST, GETAPATTERN
3897 3002 4460 INITTRACKS
3898 GETUNIT
3899 /
3900 /FORCE THE ACTUATOR SEQUENCE = 0 (INCREMENTAL 0-114)
3901 /
3901 3003 1377 THEL, TAD (XTHEL)
3902 3004 3776* DCA XGETATRACK
3903 3005 1131 TAD TARGET
3904 3006 3125 DCA START
3905 3007 1146 TAD TTRACKS
3906 3010 1145 TAD TRACKS
3907 3011 7640 SZA CLA
3908 3012 5775* JMP SEQ000
3909 3013 1031 TAD OD
3910 3014 3131 DCA TARGET
3911 3015 5775* JMP SEQ000
3912 3016 4510 XTHEL, WRITE
3913 3017 4475 HEADCOMPARE
3914 3020 2145 ISZ TRACKS
3915 3021 5203 JMP THEL
3916 3022 4454 DONE
3917 3023 5201 JMP THETEST+1
3918 /ACTUATOR MOVEMENT IS THAT SELECTED OF BITS 6,7,8 OF (TESTP)
3919 /
3920 3024 4463 XXTHEL, INITTPACKS
3921 3025 4460 GETUNIT
3922 3026 4457 XYTHEL, GETATRACK
3923 3027 4475 READCOMPARE
3924 3030 2145 ISZ TRACKS
3925 3031 5226 JMP XYTHEL
3926 3032 4454 DONE
3927 3033 5224 JMP XXTHEL
3928 3034 5453 EXIT
3929 /GET A PATTERN

```

```

3930 /
3931 3035 3035 XGETAPATTERN, .
3932 /
3933 /ONLY GENERATE A PATTERN, HOWEVER, IF THE CONTENTS OF PROGRAM LOCATIONS
3934 /
3935 /ERROFS = 0 (MEANS NO ERRORS),
3936 /
3937 /AND AC SW5 = 0 (NOT TO LOCK ON TEST OK)
3938 /
3939 /AND TEST = T13, T16, T34, T35, T36, T37
3940 /
3941 3036 4570 LAS
3942 3037 0374 AND (SW5)
3943 3040 1773* TAD ERRORS
3944 3041 7640 SZA CLA
3945 3042 5635 JMP I XGETAPATTERN
3946 3043 1372 TAD (GENTESTS-1)
3947 3044 3012 DCA A12
3948 3045 1412 TAD I A12
3949 3046 7450 SNA
3950 3047 5635 JMP I XGETAPATTERN
3951 3050 1167 TAD TEST
3952 3051 7640 SZA CLA
3953 3052 5445 JMP .-5
3954 3053 1371 TAD (WBUFFER+1)
3955 3054 3010 DCA A10
3956 3055 3153 DCA PATSUMCHECK
3957 3056 7307 CLL CLA IAC RTL
3958 3057 0132 AND TESTP / 8/12 MODE MASK
3959 3060 7640 SZA CLA
3960 3061 1370 TAD (-100) /124 DATA WORDS
3961 3062 1367 TAD (-74)
3962 3063 3011 DCA A11 / 60 DATA WORDS
3963 /WORD X AND WORD Y CONTAIN THE ADDRESSES OF THE LAST 2 SUMCHECK WORDS
3964 /
3965 3064 1011 TAD A11
3966 3065 7041 CIA
3967 3066 1366 TAD (WBUFFER+2)
3968 3067 3154 DCA WORDX
3969 3070 7301 CLL CLA IAC
3970 3071 1154 TAD WORDX
3971 3072 3155 DCA WORDY
3972 3073 1365 TAD (7000)
3973 3074 0132 AND TESTP
3974 3075 7106 CLL RTL
3975 3076 7006 RTL
3976 3077 1364 TAD (TAD PATTERNS)
3977 3100 3301 DCA .+1
3978 3101 1315 TAD PATTERNS
3979 3102 3303 DCA XPATTERNS
3980 3103 4763* XPATTERNS, JMS RANGEN
3981 3104 3135 DCA XXX
3982 /DEVELOP A 12 BIT SUMCHECK FOR THE 60/124 DATA WORDS
3983 /
3984 /NOTE:

```

```

3985 /
3986 /THE SUMCHECK WRITTEN IS THIS DEVELOPED SUMCHECK WITH THE FIRST 2 ID WORDS, AND
3987 /THE LAST 2 OVERALL SUMCHECK WORDS APPENDED WITHIN SUBROUTINE XGETASECTOR
3988 /
3989 3105 1153 TAD PATSUMCHECK
3990 3106 1135 TAD XXX
3991 3107 3153 DCA PATSUMCHECK
3992 3110 1135 TAD XXX
3993 3111 3410 DCA I A10
3994 3112 2011 ISZ A11
3995 3113 5303 JMP XPATTERNS
3996 3114 5635 JMP I XGETAPATTERN
3997 /THE FOLLOWING ARE THE ALLOCATED PATTERN POINTERS
3998 /
3999 3115 4763 PATTERNS, JMS RANGEN
4000 3116 7240 STA / 1
4001 3117 1325 TAD PAT2
4002 3120 1326 TAD PAT3
4003 3121 1327 TAD PAT4
4004 3122 1330 TAD PAT5
4005 3123 5340 JMP PAT6
4006 3124 7200 CLA
4007 /
4008 / 12 BIT MODE 8 BIT MODE
4009 /
4010 3125 1463 PAT2, 1463 / 63 / 0011 00110011
4011 3126 6314 PAT3, 6314 / 314 / 1100 11001100
4012 3127 5252 PAT4, 5252 / 252 / 1010 10101010
4013 3130 2525 PAT5, 2525 / 125 / 0101 01010101
4014 /
4015 3131 6323 GENTESTS, -T13
4016 3132 6324 -T16
4017 3133 5500 -T34
4018 3134 5502 -T35
4019 3135 5501 -T36
4020 3136 5503 -T37; 0
4021 3137 0000
4021 /PROGRAMMING NOTE:
4022 /
4023 /PATTERN 6 IS A COUNT PATTERN FROM 0 TO 7777, OR FROM 0 TO 177
4024 /
4025 /THIS COUNT PATTERN IS A " PURE " PATTERN
4026 /
4027 /A CHFKSUM IS NOT GENERATED FOR THIS PATTERN AS FOR ALL OTHER PATTERNS
4028 /
4029 / WORD/BYTE 0 = 1
4030 / WORD/BYTE 1 = 1
4031 /
4032 / WORD 2 = 0202
4033 / BYTE 2 = 2
4034 /
4035 / WORD 77 = 7777
4036 / BYTE 177 = 177
4037 /
4038 3140 7307 PAT6, CLL CLA IAC RTL / 4
    
```

```

4039 3141 0132 AND TESTP
4040 3142 7440 SZA
4041 3143 7120 STL / LINK = 1 FOR 8-BIT MODE (128 BYTES)
4042 3144 5351 JMP ,+5
4043 3145 7001 XPAT6, IAC
4044 3146 7420 SNL
4045 3147 1374 TAD (100)
4046 3150 1135 TAD XXX
4047 3151 3135 DCA XXX
4048 3152 1135 TAD XXX
4049 3153 3410 DCA I A10
4050 3154 2011 ISZ A11
4051 3155 5345 JMP XPAT6
4052 3156 5635 JMP I XGETAPATTERN
4053 3163 4710
4054 3164 1315
4055 3165 7000
4056 3166 7115
4057 3167 7704
4058 3170 7700
4059 3171 7114
4060 3172 3130
4061 3173 1363
4062 3174 0100
4063 3175 4306
4064 3176 4261
4065 3177 3016
4066 3200 PAGE
4067 /WRITE ONLY
4068 /
4069 /WRITE ALL SELECTED SECTORS OF THAT TRACK
4070 3200 3200 XWRITE, .
4071 3201 7330 STL CLA RAR
4072 3202 3120 DCA RDC / THIS IS A WRITE (RDC = 4000)
4073 /
4074 3203 4462 INITSECTORS
4075 3204 4456 WRITEL, GETASECTOR
4076 3205 7410 SKP
4077 3206 1377 REWRITE, TAD (JMP WHICHREAD)
4078 3207 3263 DCA JMPWHICHREAD
4079 /THE PROGRAM WILL ISSUE AN INIT FOR ALL SEEK ERRORS
4080 /
4081 /NOTE:
4082 /
4083 /THE FUNCTION OF THE INIT IS TO SEEK TRACK 0/SECTOR 1, AND
4084 /TRANSFER INTO THE SECTOR BUFFER THE CONTENTS OF SECTOR 1, THEREFORE
4085 /TO RECOVER FROM A SEEK ERROR, THE PROGRAM MUST RE-FILL THE SECTOR BUFFER
4086 /THEN RE-SEEK
4087 /
4088 3210 1121 TAD KRETRY
4089 3211 3143 DCA SRETRY
4090 3212 1121 WRESEEK, TAD KRETRY
4091 3213 3144 DCA PRETRY /PARITY RETRY COUNTER
4092 /FILL RX01 SECTOR BUFFER
    
```

```

4093 /
4094 3214 1376 REFILL, TAD (WBUFFER-1)
4095 3215 3010 DCA A10
4096 3216 4440 LCDA
4097 3217 5237 JMP FILLOK
4098 3220 5226 JMP FILLERROR
4099 /RETURN TO HERE IS FROM SUBROUTINE XLCDA
4100 /
4101 3221 4444 FILL, STR
4102 3222 5221 JMP FILL
4103 3223 1410 TAD I A10
4104 3224 4443 XDROUT
4105 3225 5221 JMP FILL
4106 /AC SW (4) = 0 ; INHIBIT INTERFACE PARITY RECOVERY
4107 /
4108 3226 2144 FILLERROR, ISZ PRETRY
4109 3227 7410 SKP
4110 3230 5600 JMP I XWRITE
4111 3231 4570 LAS
4112 3232 0375 AND (SW4)
4113 3233 7640 SZA CLA
4114 3234 5214 JMP REFILL
4115 3235 3144 DCA PRETRY
4116 3236 5600 JMP I XWRITE
4117 /THE SECTOR BUFFER HAS SUCCESSFULLY BEEN FILLED AND CONTAINS THE
4118 /PATTERN SELECTED - ALL SOFT PARITY ERRORS HAVE BEEN LOGGED
4119 /
4120 3237 1121 FILLOK, TAD KPRETRY
4121 3240 3144 DCA EPRETRY
4122 3241 7305 CIL CLA IAC PAL / 2
4123 3242 0132 AND TESTP /TESTP FOR DELETED DATA SELECTION
4124 3243 7007 IAC RTL / 4 IF WRITE OR 14 IF WRITE DELETED DATA
4125 3244 4441 LCDB
4126 3245 5257 JMP WRITEOK
4127 /RETURN TO HERE IS FROM A PI (IF AN RX01 ERROR FLAG)
4128 /
4129 /AC SW (4) = 0 ; INHIBIT WRITE PRETRY
4130 /
4131 3246 2143 WRITERERROR, ISZ SRETRY
4132 3247 7410 SKP
4133 3250 5261 JMP WNOTOK
4134 3251 4570 LAS
4135 3252 0375 AND (SW4)
4136 3253 7640 SZA CLA
4137 3254 5712 JMP WRESEEK
4138 3255 3143 DCA SRETRY
4139 3256 5264 JMP WNOTOK
4140 /NOTE: THE ONLY ERROR EXPECTED HERE EVER IS A "SEEK" FRPOR
4141 /
4142 /RETURN TO HERE IS FROM A PI (IF ONLY AN RX01 DONE FLAG)
4143 /
4144 3257 1131 WRITEOK, TAD TARGET
4145 3260 3125 DCA START
4146 3261 1124 TAD STARGET
4147 3262 3123 DCA SSTART
    
```

```

4148 3263 5267 JMPWHICHREAD, JMP WHICHREAD /CONTAINS 0 OR MODIFIED TO JMP WHICHREAD
4149 /
4150 3264 2122 WNOTOK, ISZ SECTORS
4151 3265 5204 JMP WRITE
4152 3266 5600 JMP I XWRITE
4153 /IF THIS IS A WRITE AFTER READ, "JMP PEADRETRY", BUT
4154 /
4155 /IF A HARD SEEK ERROR, "JMP NUREAD"
4156 /
4157 3267 1143 WHICHREAD, TAD SRETRY
4158 3270 7650 SMA CLA
4159 3271 5774 JMP NUREAD /HARD SEEK ERROR
4160 3272 5773 JMP READRETRY /SOFT
4161 /
4162 / 0 < (LAST) => (FIRST)
4163 /
4164 3273 1034 TADLAST, TAD LAST
4165 3274 7740 SMA SZA CLA
4166 3275 5300 JMP .+3
4167 3276 1372 TAD (32)
4168 3277 3034 DCA LAST
4169 3300 1034 TAD LAST
4170 3301 7041 CIA
4171 3302 1033 TAD FIRST
4172 3303 7740 SMA SZA CLA
4173 3304 5276 JMP .-6
4174 3305 4473 PRINT
4175 3306 6710 MLAST
4176 3307 4505 TY4OCT
4177 3310 0034 LAST
4178 3311 4504 TAB
4179 3312 0005 TAB 5 /TAB TO MOVE TELEPRINTER HEAD
4180 /*****
4181 /CONSOLE
4182 /*****
4183 /
4184 /
4185 3313 4421 CHECKC0 /IS CONSOLE ACTIVE
4186 3314 4426 C0SWIT /YES PRINT SWITCH REGISTER QUESTION
4187 /FOR RUNNING SWITCHES
4188 3315 5317 JMP C0RET4
4189 3316 4461 HLT
4190 3317 5771 C0RET4, JMP PRETEST
4191 /ENTRY TO HERE IS FROM RX01 PI ERROR SERVICE ROUTINE
4192 /
4193 /CONTROLLER FAILURES
4194 /
4195 3320 1370 SDNUNEXPECTED, TAD (MSDNUNEXPECTED)
4196 3321 7410 SKP
4197 3322 1367 NOSER, TAD (MNOSER)
4198 3323 3334 DCA XMESSAGE
4199 3324 4570 LAS
4200 3325 0371 AND (SW3)
4201 3326 7640 SZA CLA
4202 3327 5335 JMP QUIET
    
```

```

4203 3330 4575 FORCE
4204 3331 1334 TAD XMESSAGE
4205 3332 4467 /POINTS TO ERROR MESSAGE THAT WOULD HAVE BEEN PRINTED,
4206 3333 4473 PRINT /REPORT ERROR TO APT IF REQUIRED.
4207 3334 0000 XMESSAGE, 0
4208 3335 4570 QUIET, LAS
4209 3336 7710 SPA CLA
4210 3337 4461 HLT
4211 3340 5766 JMP ERRETURN
4212 /
4213 /ENTRY TO HERE FROM SUBROUTINES XSER, OR XSDN
4214 /
4215 3341 2111 HUNGUP, ISZ BUSY
4216 3342 3353 DCA HUNGPC
4217 3343 4575 FORCE
4218 /
4219 /THE LABEL "XHUNG" MUST RESIDE HERE BECAUSE OF REFERENCES MADE WITHIN "FORCE"
4220 /
4221 3344 4473 XHUNG, PRINT
4222 3345 6611 MHUNGPC
4223 3346 4505 TX4OCT
4224 3347 3353 HUNGPC
4225 3350 1365 TAD (-40)
4226 3351 3116 DCA HANGER
4227 3352 5453 NOTEST
4228 3353 0000 HUNGPC, 0
4229 /
4230 3365 7740 /
4231 3366 5506 /
4232 3367 7055 /
4233 3370 7042 /
4234 3371 0400 /
4235 3372 0032 /
4236 3373 3423 /
4237 3374 3517 /
4238 3375 0200 /
4239 3376 7112 /
4240 3377 5767 /
4241 3400 PAGE
4242 /
4243 /ROUTINE IS USED INPLACE OF THE LAS INSTRUCTION WILL READ CORRECT SWITCHES
4244 /
4245 3400 0000 XCKSWIT, 0
4246 3401 7200 CLA
4247 3402 1022 TAD 22 /CHECK STATUS WORD
4248 3403 0377 AND (400) /1= ACTIVE CONSOLE
4249 3404 7650 SNA CIA
4250 3405 7614 7614 /LAS AND SKIP
4251 3406 1020 TAD 20 /GET PSEUDO SW REG
4252 3407 5600 JMP I XCKSWIT /EXIT XCKSWIT
4253 /READ VERIFY
4254 /
4255 3410 3410 XREADCOMPARE,
4256 3411 1210 TAD XREADCOMPARE

```

```

4257 3412 3214 DCA XREAD
4258 3413 5216 JMP XRDC
4259 /READ AND HEAD AFTER WRITE SUBROUTINE
4260 /
4261 /*****
4262 /
4263 /IF THIS IS A READ AFTER WRITE, THEN 1 PROGRAM LOCATION PRECEEDING
4264 /THE PROGRAM LOCATION CONTAINING THE "READ" WILL CONTAIN "WRITE"
4265 /
4266 /*****
4267 /
4268 /IF THIS IS A READ ONLY, THEN THE CONTENTS OF " RDC " WILL = 1
4269 /
4270 /IF THIS IS A READ VERIFY, THEN THE CONTENTS OF " RDC " WILL = 0
4271 /
4272 /READ ALL SELECTED SECTOPS OF THAT TRACK
4273 /
4274 3414 3414 XREAD,
4275 3415 7301 CLL CLA IAC
4276 3416 3120 XRDC, DCA RDC
4277 3417 4462 INITSECTORS
4278 3420 1121 READL, TAD KRETRY
4279 3421 3136 DCA RIRETRY
4280 3422 4456 GETASECTOR
4281 3423 1121 READRETRY, TAD KRETRY
4282 3424 3137 DCA R2RETRY
4283 /REFRESH PROGRAM LOCATION SNDLOG BECAUSE THE CONTENTS MIGHT BE RESIDUAL
4284 /IF A PREVIOUS SND (STATUS NO DATA) ERROR EVER OCCURED
4285 /
4286 3425 3142 DCA SNDLOG
4287 3426 3140 DCA DNSLOG
4288 3427 3141 DCA DNSLOG
4289 /
4290 /
4291 3430 1121 REREAD, TAD KRETRY
4292 3431 3143 DCA SRETRY
4293 3432 1121 RESEEK, TAD KRETRY
4294 3433 3144 DCA PRETRY
4295 3434 7327 CLA STL IAC RTL
4296 3435 4441 LCDB
4297 3436 5327 JMP READOK
4298 /RETURN TO HERE IS FROM A PI (IF AN RX01 ERROR FLAG)
4299 /
4300 /AN ERROR HAS BEEN DETECTED
4301 /
4302 /IF NOT A CRC ERROR THEN ASSUME A SEEK ERROR
4303 /
4304 3437 7301 READERROR, CLL CLA IAC
4305 3440 0126 AND ASTATUS /CRC MASK
4306 3441 7640 SZA CLA
4307 3442 5254 JMP CRCERROR
4308 3443 2143 ISZ SRETRY
4309 3444 7410 SKP
4310 3445 5345 JMP RLOGGED
4311 3446 4570 LAS

```

```

4312 3447 0376 AND (SW4)
4313 3450 7640 SZA CLA
4314 3451 5232 JMP RESECK
4315 3452 3143 DCA SRETRY
4316 3453 5345 JMP RLOGGED
4317 /
4318 /PROGRAMMING NOTE:
4319 /
4320 /ANY RECOVERABLE SEEK ERRORS ARE NOTED AT PROGRAM LOCATION " DNS "
4321 /
4322 /AN ERROR HAS OCCURED
4323 /
4324 3506 SND=DNS
4325 3506 DWS=SND
4326 /
4327 /THE SECTOP BUFFER CONTAINS THE DATA READ
4328 /
4329 /ANY PARITY ERRORS WOULD HAVE PREVIOUSLY BEEN DETECTED AND LOGGED
4330 /
4331 /THIS ERROR IS NOT A SEEK ERROR, THEREFORE IT IS ASSUMED TO BE A
4332 /
4333 / CRC ERROR
4334 /
4335 /COMPARE THE DATA WITHIN THE SECTOR BUFFER TO DETECT CRC STATUS
4336 /WITHOUT DATA ERRORS (SND)
4337 /
4338 /IF AT THE END OF THE COMPARE, STATUS NO DATA ERRORS HAVE BEEN DETECTED
4339 /THEN PRINT AN APPROPRIATE MESSAGE
4340 /
4341 3454 1120 CRCERROR, TAD RDC
4342 3455 7640 SZA CLA
4343 3456 5303 JMP XRCERROR
4344 3457 4775 JMS COMPARE
4345 3460 1113 TAD COMPERROR
4346 3461 7450 SNA
4347 3462 1774 TAD INSUMCHECK
4348 3463 7640 SZA CLA
4349 3464 5306 JMP DWS
4350 3465 2142 ISZ SNDLOG
4351 /
4352 /IF AC SW (3) = 1 THEN DO NOT PRINT THE ERROR INFORMATION
4353 /
4354 3466 4570 LAS
4355 3467 0377 AND (SW3)
4356 3470 7640 SZA CLA
4357 3471 5303 JMP XRCERROR
4358 3472 4350 JMS RDORWR
4359 3473 5277 JMP ,+4
4360 3474 4473 PRINT
4361 3475 6715 MWRITE
4362 3476 5301 JMP ,+3
4363 3477 4473 PRINT
4364 3500 6721 MREAD
4365 3501 4473 PRINT
4366 3502 6737 MSHDERROR
    
```

```

4367 3503 4570 XPCERROR, LAS
4368 3504 7710 SPA CLA
4369 3505 4461 HLT7, HLT
4370 /
4371 3506 1121 DNS, TAD RPRETRY
4372 3507 3143 DCA SRETRY
4373 3510 2137 ISZ R2PRETRY
4374 3511 7410 SKP
4375 3512 5322 JMP UREAD
4376 /AC SW (4) = 0 ; INHIBIT READ PRETRY
4377 /
4378 3513 4570 LAS
4379 3514 0376 AND (SW4)
4380 3515 7640 SZA CLA
4381 3516 5230 JMP REREAD
4382 /A HARD FILL BUFFER PARITY ERROR OR A HARD PARITY ERROR ON THE COMMAND/
4383 /SECTOR/TRACK WORDS, OR
4384 /A HARD SEEK ERROR WHICH HAS BEEN LOGGED WITHIN THE WRITE SUBROUTINE, OR
4385 /A HARD CRC ERROR WHICH OCCURED WHILE WITHIN THE READ SUBROUTINE
4386 /
4387 3517 3136 NUREAD, DCA RIRETRY
4388 3520 3137 DCA R2RETRY
4389 3521 5345 JMP RLOGGED
4390 /IF THIS IS A PEAD AFTER WRITE THEN RE-WRITE THE SECTOR IN ERROR
4391 /
4392 /BECAUSE THIS IS A HARD PROGRAM COMPARE DATA ERROR
4393 /
4394 3522 4350 UREAD, JMS RDORWR
4395 3523 5317 JMP NUREAD
4396 3524 2136 ISZ RIRETRY
4397 3525 5773 JMP REWRITE
4398 3526 5345 JMP RLOGGED
4399 /IF THIS IS A READ COMPARE TEST
4400 /
4401 /THEN " JMS COMPARE ", BUT ...
4402 /
4403 /IF A STATUS ERROR WITH NO DATA ERROR HAS PREVIOUSLY BEEN DETECTED
4404 /
4405 /THEN DO NOT RE- " JMS COMPARE ", AND
4406 /
4407 /DO NOT RESET PROGRAM LOCATION " START " WITH " TARGET "
4408 /
4409 /PROGRAMMING NOTE:
4410 /
4411 /ANY RECOVERABLE SEEK ERRORS WHICH OCCURED PRIOR TO CRC OR DATA ERRORS
4412 /
4413 /ARE LOGGED AT PROGRAM LOCATION " DNS "
4414 /
4415 3527 1142 READOK, TAD SNDLOG
4416 3530 1120 TAD RDC
4417 3531 7640 SZA CLA
4418 3532 5341 JMP ROK
4419 3533 4775 JMS COMPARE
4420 3534 1113 TAD COMPERROR
4421 3535 7450 SNA
    
```

```

4422 3536 1774' TAD INSUMCHECK
4423 3537 7640 SZA CLA
4424 3540 5306 JMP DNS
4425 3541 1131 ROK, TAD TARGET
4426 3542 3125 DCA START
4427 3543 1124 TAD STARGET
4428 3544 3123 DCA SSTART
4429 3545 2122 RLOGGED, ISZ SECTORS
4430 3546 5220 JMP READL
4431 3547 5614 JMP I XREAD
4432 /IF THE PROGRAMMED INSTRUCTION IMMEDIATELY BEFORE " READ "
4433 /IS A " WRITE " , THEN THIS IS A WRITE DATA ERROR, IF NOT,
4434 /THEN THIS IS A READ DATA ERROR
4435 /
4436 /FORM: JMS RDORWR; (READ RETURN); (WRITE RETURN)
4437 /
4438 3550 3550 RDORWR,
4439 3551 7344 CLL STA PAL
4440 3552 1214 TAD XREAD
4441 3553 3135 DCA XXX
4442 3554 1535 TAD I XXX
4443 3555 1372 TAD (-WRITE)
4444 3556 7650 SNA CLA
4445 3557 2350 ISZ RDORWR
4446 3560 5750 JMP I RDORWR
4447 3572 3270 /
4448 3573 3206 /
4449 3574 3761 /
4450 3575 3601 /
4451 3576 0200 /
4452 3577 0400 /
4453 3600 3600 PAGE
4454 3600 5601 JMPICOMPARE, JMP I COMPARE
4455 /
4456 /THE FOLLOWING INFORMATION IS ALWAYS PRINTED IF A PROGRAM COMPARE DATA ERROR
4457 /
4458 DATA ERROR
4459 WORD GOOD BAD
4460 /
4461 /WHERE " WORD " IS THE WORD NUMBER (0-127),
4462 /AND " GOOD " IS THE DATA WORD WRITTEN,
4463 /AND " BAD " IS THE DATA WORD READ (IN ERROR)
4464 /
4465 /WORDS 0 AND 1 ARE HEADER WORDS (8 BIT BYTES)
4466 /
4467 / WORD 0 - TRACK # (BITS 5-11)
4468 / WORD 1 - SECTOR (BITS 7-11)
4469 /
4470 /AC SW (4) = 1 ; INHIBIT READ DATA ERROR TYPEOUT
4471 /
4472 /
4473 3601 3601 COMPARE, .
4474 /
4475 /EMPTY BUFFER
    
```

```

4476 /
4477 /SAVE THE PREVIOUS READ'S A-STATUS REGISTER
4478 /
4479 3602 7301 CLL CLA IAC
4480 3603 0126 AND ASTATUS
4481 3604 3360 DCA XASTATUS
4482 3605 1121 TAL KPENTRY
4483 3606 3144 DCA PENTRY
4484 3607 1377 EMPTYL, TAD (WBUFFER)
4485 3610 3134 DCA XA11
4486 3611 7305 CLL CLA IAC RAL
4487 3612 4440 LCDA
4488 3613 5776* JMP EMPTYOK
4489 3614 5775* JMP EMPTYERROR
4490 /RETURN TO HERE FROM SUBROUTINE XLCD
4491 /
4492 3615 3113 DCA COMPREPROR
4493 3616 3361 DCA INSUMCHECK
4494 /
4495 /AC SW (3) = 1 ; INHIBIT FURTHER READ DATA ERROR TYPEOUTS
4496 /
4497 /...BUT STAY IN THIS LOOP UNTIL THE RX01 SECTOR BUFFER IS EMPTIED
4498 /
4499 3617 4444 EMPTY, STR
4500 3620 5217 JMP EMPTY
4501 /
4502 /DISABLE THE PX01 INTERRUPT
4503 /
4504 3621 4447 INTR
4505 /
4506 /TRANSFER DATA FROM THE SECTOR BUFFER INTO THE ACCUMULATOR
4507 /
4508 3622 4442 XDRIN / "ACTUAL" (BAD) DATA
4509 3623 3133 DCA XA10
4510 /
4511 /ASSUME A 12 BIT SUMCHECK EVEN IF 8 BIT MODE FOR NOW
4512 /
4513 /NOTE: THE CONTENTS OF PROGRAM LOCATION " INSUMCHECK " SHOULD = 0
4514 / (AFTER ALL WORDS HAVE COME IN)
4515 /
4516 3624 1133 TAD XA10
4517 3625 1361 TAD INSUMCHECK
4518 3626 3361 DCA INSUMCHECK
4519 3627 7307 CLL CLA IAC RTL / 4 (8/12 BIT MODE MASK)
4520 3630 0132 AND TESTP
4521 3631 7112 CLL RTR
4522 3632 7010 RAR
4523 3633 1534 TAD I XA11
4524 3634 7430 SZL / 0 = 12 BIT, 1 = 8 BIT
4525 3635 0374 AND (377)
4526 3636 7041 CIA
4527 3637 1133 TAD XA10
4528 3640 7650 SNA CLA
4529 3641 5773* JMP ENDCOMPARE
4530 /
    
```

```

4531 /A PROGRAM COMPARE DATA ERROR HAS BEEN DETECTED
4532 /
4533 /NOTATION:
4534 /
4535 /THE CONTENTS OF PROGRAM LOCATION XASTATUS REFLECTS THE STATUS OF THE
4536 /RX01 AT THE COMPLETION OF THE PREVIOUS " READ SECTOR "
4537 /
4538 /IF A CRC ERROR EXISTS THEN THIS IS A DATA ERROR WITH STATUS ERROR
4539 /
4540 /IF A CRC ERROR DOESN'T EXIST THEN THIS IS A DATA ERROR WITH NO ERROR STATUS
4541 /
4542 3642 1360 TAD XASTATUS / A-STATUS OF PREVIOUS READ
4543 3643 7450 SNA
4544 3644 2141 ISZ DNSLOG /NO CRC STATUS
4545 3645 7640 SZA CLA
4546 3646 2140 ISZ DWSLOG /CRC STATUS
4547 /
4548 /AC SW (3) TO INHIBIT ERROR PRINTOUT
4549 /
4550 /IF THIS IS A READ ONLY TEST (NOT PFAD COMPARE) OR IF AC SW 3 = 1
4551 /THEN DO NOT PRINT ERROR INFORMATION
4552 /
4553 /BUT,
4554 /
4555 /IF THIS IS A READ COMPARE TEST AND AC SW 3 = 0
4556 /THEN PRINT ERROR INFORMATION
4557 /
4558 3647 4570 LAS
4559 3650 0372 AND (SW3)
4560 3651 1120 TAD RDC
4561 3652 7640 SZA CLA
4562 3653 5356 JMP ISZCOMPERROR
4563 /IF THIS IS THE FIRST COMPARE ERROR THEN PRINT HEADER INFORMATION
4564 /
4565 3654 1113 TAD COMPERROR
4566 3655 7640 SZA CLA
4567 3656 5317 JMP CNOTFIRST
4568 /
4569 /IF A DATA NO CRC STATUS ERROR (DNS) THEN PRINT ENTIRE ERROR INFORMATION
4570 /
4571 3657 1141 TAD DNSLOG
4572 3660 7640 SZA CLA
4573 3661 4575 FORCE
4574 /
4575 /IF THE PROGRAMMED INSTRUCTION IMMEDIATELY BEFORE " READ "
4576 /IS A " WRITE " , THEN THIS IS A WHITE DATA ERROR, IF NOT,
4577 /THEN THIS IS A READ DATA ERROR
4578 /
4579 3662 4771 XCOMPARE, JMS RDORWR
4580 3663 5267 JMP .+4
4581 3664 4473 PRINT
4582 3665 6715 MWRITE
4583 3666 5271 JMP .+3
4584 3667 4473 PRINT
4585 3670 6721 MPEAD
4586 /NOTATION:

```

```

4586 /
4587 /THE CONTENTS OF PROGRAM LOCATION XASTATUS REFLECTS THE STATUS OF THE
4588 /RX01 AT THE COMPLETION OF THE PREVIOUS " READ SECTOR "
4589 /
4590 /IF A CRC ERROR EXISTS THEN THIS IS A DATA ERROR WITH STATUS ERROR
4591 /
4592 /IF A CRC ERROR DOESN'T EXIST THEN THIS IS A DATA ERROR WITH NO ERROR STATUS
4593 /
4594 3671 1360 TAD XASTATUS
4595 3672 7640 SZA CLA
4596 3673 5277 JMP DWESERROR
4597 3674 4473 PRINT
4598 3675 6725 MDNSERROR
4599 3676 5301 JMP .+3
4600 3677 4473 DWESERROR, PRINT
4601 3700 6733 MDWESERROR
4602 3701 4473 PRINT
4603 3702 6752 MDATAERROR
4604 /PRINT " WORD " IF 12-BIT MODE, OR PRINT " BYTE " IF 8-BIT MODE
4605 /
4606 3703 1370 TAD (MWORD)
4607 3704 3314 DCA XLENGTH
4608 3705 7307 CLL CLA IAC RTL
4609 3706 0132 AND TESTP
4610 3707 7650 SNA CLA
4611 3710 5313 JMP .+3
4612 3711 1367 TAD (MBYTE)
4613 3712 3314 DCA XLENGTH
4614 3713 4473 PRINT
4615 3714 6761 XLENGTH, MWORD
4616 3715 4473 PRINT
4617 3716 6767 MCB
4618 / AC 2 = 0 - PRINT ONLY FIRST 3 COMPARE ERRORS
4619 / AC 2 = 1 - PRINT ALL COMPARE ERRORS
4620 /
4621 3717 4570 CNOTFIRST, LAS
4622 3720 7106 CLL RTL
4623 3721 7710 SPA CLA
4624 3722 5327 JMP .+5
4625 3723 7346 CLL STA RTL
4626 3724 1113 TAD COMPERROR
4627 3725 7700 SMA CLA
4628 3726 5356 JMP ISZCOMPERROR
4629 3727 4473 PRINT
4630 3730 6601 MCRLF
4631 3731 1366 TAD (=WBUFFER)
4632 3732 1134 TAD XA11
4633 3733 3156 DCA LSB
4634 3734 4505 TY4OCT
4635 3735 0156 LSB
4636 3736 4504 TAB
4637 3737 0005 5
4638 /IF 8-BIT MODE THEN MASK THE 12-BIT "GOOD" WORDS (377)
4639 /
4640 3740 7307 CLL CLA IAC RTL

```



```

4641 3741 0132 AND TESTP
4642 3742 7112 CLL RTR
4643 3743 7010 RAR
4644 3744 1534 TAD I XA11
4645 3745 7430 SZL
4646 3746 0374 AND (377)
4647 3747 3156 DCA LSB
4648 3750 4505 TY4OCT
4649 3751 0156 LSB
4650 3752 4504 TAB
4651 3753 0012 12
4652 3754 4505 TY4OCT
4653 3755 0133 XA10
4654 3756 2113 ISZCOMPERROR, ISZ COMPERROR
4655 3757 5773* JMP ENDCOMPARE
4656 /THE CONTENTS OF THE XASTATUS PROGRAM LOCATION REFLECTS THE STATUS AT THE
4657 /COMPLETION OF THE PREVIOUS READ FUNCTION
4658 /
4659 /THE PREVIOUS STATUS IS SAVED BECAUSE A SUCCEEDING ERROR
4660 / (PARITY ERROR ON THE COMMAND WORD) MAY OCCUR
4661 /
4662 3760 0000 XASTATUS, 0
4663 /
4664 /THE CONTENTS OF "INSUMCHECK" IS THE SUMCHECK DERIVED FROM THE SUM OF ALL
4665 /DATA WORDS COMING IN (THE SUM OF ALL "BAD"), AND
4666 /SHOULD BE EQUIVALENT TO 0 AT THE END OF THE EMPTY BUFFER
4667 /
4668 3761 0000 INSUMCHECK, 0
4669 3766 0665
4670 3767 6764
4671 3770 6761
4672 3771 3550
4673 3772 0400
4674 3773 4000
4675 3774 0377
4676 3775 4005
4677 3776 4016
4678 3777 7113
4679 / PAGE
4680 /
4681 /RE-ENABLE THE RX01 INTERRUPT
4682 /
4683 4000 7301 ENDCOMPARE, CLL CLA IAC
4684 4001 4447 INTR
4685 4002 2134 ISZ XA11
4686 4003 6001 ION
4687 4004 5777* JMP EMPTY
4688 /
4689 /THIS INTERFACE PARITY ERROR MUST BE ON THE COMMAND WORD TO "EMPTY BUFFER"
4690 /
4691 /NOTE: IT CAN'T BE ANY OTHER ERROR
4692 /
4693 4005 2144 EMPTYERROR, ISZ PRETRY
4694 4006 7410 SKP
4695 4007 5266 JMP XEMPTYOKNOTOK
    
```

```

4695 4010 4570 LAS
4696 4011 0376 AND (SW4)
4697 4012 7640 SZA CLA
4698 4013 5775* JMP EMPTY
4699 4014 3144 DCA PRETRY
4700 4015 5266 JMP XEMPTYOKNOTOK
4701 /IF AC SW (3) = 0
4702 /
4703 /PRINT A VALUE SYMBOLIC OF THE TOTAL # OF COMPARE ERRORS DETECTED
4704 /
4705 /IF 8-BIT MODE THEN THE "SUMCHECK" WILL OVERFLOW INTO BITS 0 TO 3
4706 /
4707 /THEREFORE MASK THE CONTENTS OF "INSUMCHECK"
4708 /
4709 /THE RESULT OF THE SUBTRACTION SHOULD = 0
4710 /
4711 /NO MASK IS NEEDED FOR 12-BIT MODE
4712 /
4713 4016 7307 EMPTYOK, CLL CLA IAC RTL
4714 4017 0132 AND TESTP
4715 4020 7112 CLL RTR
4716 4021 7010 RAR
4717 4022 1774* TAD INSUMCHECK
4718 4023 7430 SZL
4719 4024 0373 AND (377)
4720 4025 3774* DCA INSUMCHECK
4721 4026 4570 LAS
4722 4027 0372 AND (SW3)
4723 4030 7640 SZA CLA
4724 4031 5266 JMP XEMPTYOK
4725 /IF A SUMCHECK ERROR EXISTS WITHOUT A "COMPERROR" THEN "FORCE" A TYPEOUT
4726 /
4727 4032 1113 TAD COMPERROR
4728 4033 7640 SZA CLA
4729 4034 5241 JMP .+5
4730 4035 1774* TAD INSUMCHECK
4731 4036 7650 SNA CLA
4732 4037 5266 JMP XEMPTYOK
4733 4040 4575 FORCE
4734 4041 4473 PRINT
4735 4042 6774 MSUMCHECK
4736 4043 1774* TAD INSUMCHECK
4737 4044 7640 SZA CLA
4738 4045 5251 JMP .+4
4739 4046 4473 PRINT
4740 4047 7004 NOK
4741 4050 5254 JMP .+4
4742 4051 4505 TY4OCT
4743 4052 3761 INSUMCHECK
4744 4053 5257 JMP .+4
4745 4054 1113 TAD COMPERROR
4746 4055 7650 SNA CLA
4747 4056 5266 JMP XEMPTYOK
4748 4057 4473 PRINT
4749 4060 7006 MDESUMMARY
    
```

```

4750 4061 4505 TY40CT
4751 4062 0113 COMPERROR
4752 4063 4504 TAB
4753 4064 0005 5
4754 4065 5266 JMP XEMPTYOK
4755 /
4756 /AC SW 0 = 1 ; HALT ON ERROR
4757 /
4758 4066 1113 XEMPTYOK, TAD COMPERROR
4759 4067 7450 SNA
4760 4070 1774* TAD INSUMCHECK
4761 4071 7650 SNA CLA
4762 4072 5771* JMP JMPICOMPARE
4763 4073 4570 LAS
4764 4074 7710 SPA CLA
4765 4075 4461 HLT6, HLT
4766 4076 5771* JMP JMPICOMPARE
4767 4077 4077 XSETUP, .
4768 4100 4507 WAITTY
4769 4101 1370 TAD (ANDRETURN)
4770 4102 3767* DCA XPRINT
4771 4103 1677 TAD I XSETUP
4772 4104 3314 DCA XANDRETURN
4773 4105 2277 ISZ XSETUP
4774 4106 5677 JMP I XSETUP
4775 /
4776 /AND RETURN TO HEPE WAITING FOR ENTIRE NUMERICAL OUTPUT
4777 /
4778 4107 4507 ANDRETURN, WAITTY
4779 /
4780 /THEN EXIT FROM HERE FOR MAIN LINE CODE
4781 /
4782 4110 1714 THENEXIT, TAD I XANDRETURN
4783 4111 3314 DCA XANDRETURN
4784 4112 3160 DCA TTYBUSY
4785 4113 5714 JMP I XANDRETURN
4786 4114 4114 XANDRETURN, .
4787 /
4788 4115 4115 XHALT, .
4789 4116 4507 WAITTY
4790 /*****
4791 /CONSOLE
4792 /*****
4793 4117 4424 CHECKCB
4794 4120 4427 C0INGU
4795 4121 5325 JMP C0RET3 /PRINT WAITING IF ON CONSOLE
4796 /*****
4797 4122 7240 STA
4798 4123 1315 TAD XHALT
4799 4124 7402 7402
4800 4125 7200 C0RET3,CLA
4801 4126 5715 JMP I XHALT
4802 /
4803 /ROUTINE TO NOTIFY OF OF RUNNING IF NEED BE DONE
4804 /
    
```

```

4805 4127 0000 XTICK, 0
4806 4130 4472 CHEK22
4807 4131 7410 SKP /ON APT.
4808 4132 5727 JMP I XTICK
4809 4133 2345 ISZ CLKCNT
4810 4134 5727 JMP I XTICK
4811 4135 1346 TAD COUNT
4812 4136 3345 DCA CLKCNT /INIT CLOCK COUNTER
4813 4137 6002 IOF
4814 4140 6201 CDF 00
4815 4141 6272 CIF 70
4816 4142 4744 JMS I K6500 /NOTIFY APT-8
4817 4143 5727 JMP I XTICK /EXIT.
4818 /
4819 4144 6500 K6500, 6500
4820 4145 7777 CLKCNT, -1
4821 4146 7763 COUNT, -15
4822 4167 5244
4823 4170 4107
4824 4171 3600
4825 4172 0400
4826 4173 0377
4827 4174 3761
4828 4175 3607
4829 4176 0200
4830 4177 3617
4831 /
4832 /SEQUENCE TO THE NEXT AVAILABLE DISKETTE
4833 /
4833 4200 4200 XGETUNIT, .
4834 4201 3240 DCA UNITZ /CLEAR FOR A NEW DISKETTE
4835 4202 1236 TAD WUNITS /WORKING UNIT COUNTER
4836 4203 7450 SNA
4837 4204 1235 TAD UNITS /EXHAUSTED ALL DISKETTES ; RESET
4838 4205 3236 DCA WUNITS
4839 4206 3237 DCA POLL /CLEAR POLLER
4840 4207 7120 STL /START
4841 4210 1237 NEXT, TAD POLL
4842 4211 7010 RAR /NEXT
4843 4212 3237 DCA POLL
4844 4213 1237 TAD POLL
4845 4214 7430 SZL /DISKETTE
4846 4215 4461 HLT /NO UNITS AVAILABLE ; CATASTROPHIC
4847 4216 0236 AND WUNITS
4848 4217 7450 SNA
4849 4220 2240 ISZ UNITZ /ACTIVE DISKETTE
4850 4221 7450 SNA
4851 4222 5210 JMP NEXT /TRY AGAIN
4852 /A DISKETTE IS AVAILABLE AND SELECTED FOR OPERATIONS
4853 4223 3241 DCA UNITX /A CODED VERSION OF UNIT
4854 4224 1241 TAD UNITX
4855 4225 7040 CMA /...DELETE FROM
4856 4226 0236 AND WUNITS /...AVAILABLE UNIT LIST (WUNITS)
4857 4227 3236 DCA WUNITS /...AND CREATE NEW LIST
4858 4230 1240 TAD UNITZ
    
```

```

4859 4231 7106          CLL RTL
4860 4232 7006          RIL /BIT 7 OF COMMAND REGISTER
4861 4233 3242          DCA UNIT /FOR COMMAND REGISTER LOAD LATER
4862 4234 5600          JMP I XGETUNIT
4863 4235 0000          UNITS, 0000 /AVAILABLE UNIT LIST (MAX SYS CONFIGURATION)
4864 4236 0000          WUNITS, 0 /CODED WORKING UNIT LIST (UNITS YET TO BE EXERCISED)
4865 4237 0000          POLL, 0 /DISKETTE POLL
4866 /
4867 /UNITZ ; UNIT 1 LOOKS LIKE 0001
4868 /UNITX ; UNIT 1 LOOKS LIKE 2000
4869 /UNIT ; UNIT 1 LOOKS LIKE 0020 (RX01 COMMAND WORD BIT 7)
4870 /
4871 4240 0000          UNITZ, 0 /ACTIVE DISKETTE
4872 4241 0000          UNITX, 0 /ACTIVE DISKETTE IN CODED FORM
4873 4242 0000          UNIT, 0 /A CODED VERSION OF UNIT Z (FOR COMMAND REGISTER LOAD)
4874 /
4875 /IF (WUNITS) = 0 ; ALL SELECTED DISKETTE DRIVES HAVE SEQUENCED
4876 /
4877 4243 4243          XDONE, .
4878 4244 1236          TAD WUNITS
4879 4245 7650          SNA CIA
4880 4246 2243          ISZ XDONE / DONE ; (WUNITS) = 0
4881 4247 5643          JMP I XDONE
4882 /INITIALIZE THE NUMBER OF TRACKS ACCESSED VIA THE DIFFERENCE BETWEEN
4883 /THE CONTENTS OF PROGRAM LOCATIONS " OD " AND " ID ".
4884 /
4885 4250 4250          XINITTRACKS, .
4886 4251 1032          TAD ID
4887 4252 7040          CMA
4888 4253 1031          TAD OD
4889 4254 3145          DCA TRACKS
4890 4255 1145          TAD TRACKS
4891 4256 7041          CIA
4892 4257 3146          DCA TTRACKS
4893 4260 5650          JMP I XINITTRACKS
4894 / OD (OUTSIDE DIAMETER) ACTUATOR POSITION INITIALIZED TO (0)
4895 /
4896 / ID (INSIDE DIAMETER) ACTUATOR POSITION INITIALIZED TO 114 (76 DECIMAL)
4897 /
4898 / " XGETATRACK " WILL GET A TRACK VALUE BETWEEN THE LIMITS OF THE CONTENTS OF
4899 /PROGRAM LOCATION OD (MIN 0), AND THE CONTENTS OF ID (MAX 114).
4900 /
4901 /GET A DISKETTE TRACK TO BE (AC) WITHIN IOT LCD-B
4902 /
4903 / IF THIS IS THE " FIRSTMOVE ", (START) IS NOT APPLICABLE HERE, BUT
4904 / WILL BECOME APPLICABLE WITHIN SUBROUTINE " XGETASECTOR ".
4905 /
4906 4261 4261          XGETATRACK, .
4907 4262 1131          TAD TARGET /PRESENT ACTUATOR POSITION (FROM PREVIOUS LCD-B)
4908 4263 3125          DCA START /BECOMES STARTING ACTUATOR POSITION
4909 4264 1377          TAD (70)
4910 4265 0132          AND TESTP
4911 4266 7110          CLL RAR
4912 4267 7012          RTR
4913 4270 1376          TAD (TAD SEQ)

```

```

4914 4271 3272          DCA ,+1
4915 4272 1276          TAD SEQ
4916 4273 3275          DCA ,+2
4917 4274 5675          RESEQUENCE, JMP I ,+1
4918 4275 4306          SEQ000
4919 /TRACK ACCESS SEQUENCE IS SELECTED VIA AC SWITCHES 6,7,8 AT L/S 200
4920 /
4921 / 0 - INCREMENTAL (1-114-0)
4922 / 1 - DECREMENTAL (114-0)
4923 / 2 - 1-114, 113-0
4924 / 3 - BOUNCE ID TO OD ONLY
4925 / 4 - BOUNCE: (114, 0 ; 113, 1 ; 112, 2 ; ...ETC TO 47, 45)
4926 / 5 -
4927 / 6 - STROBE: (77, 0 ; 76, 0 ; 75, 0 ; ...ETC TO 1, 0)
4928 / 7 - RANDOM
4929 /
4930 4276 4306          SEQ, SEQ000
4931 4277 4317          SEQ001
4932 4300 4400          SEQ010
4933 4301 4422          SEQ3
4934 4302 4444          SEQ100
4935 4303 5453          NOTEST
4936 4304 4465          SEQ6
4937 4305 4330          SEQ111
4938 /
4939 /INCREMENTAL ACTUATOR ACCESS (OD INCREMENTALLY TO ID)
4940 /
4941 4306 1146          SEQ000, TAD TTRACKS
4942 4307 1145          TAD TRACKS
4943 4310 7640          SZA CLA
4944 4311 5314          JMP ,+3
4945 4312 1031          TAD OD
4946 4313 5332          JMP DCATARGET
4947 4314 1131          TAD TARGET
4948 4315 7001          IAC
4949 4316 5332          JMP DCATARGET
4950 /DECREMENTAL ACTUATOR ACCESS (ID TO OD)
4951 /
4952 4317 1146          SEQ001, TAD TTRACKS
4953 4320 1145          TAD TRACKS
4954 4321 7640          SZA CLA
4955 4322 5325          JMP ,+3
4956 4323 1032          TAD ID
4957 4324 5332          JMP DCATARGET
4958 4325 1131          TAD TARGET
4959 4326 1117          TAD K7777
4960 4327 5332          JMP DCATARGET
4961 /RANDOM ACTUATOR ACCESS
4962 /
4963 4330 4775          SEQ111, JMS RANGEN
4964 4331 0374          AND (177)
4965 4332 3131          DCATARGET, DCA TARGET
4966 /PREVIOUSLY AT L/S THE FOLLOWING CONDITION WAS TRUE
4967 /
4968 /

```

```

4969 /
4970 /THEREFORE TEST FOR THE CONDITION
4971 /
4972 / (TARGET) <= (ID)
4973 /
4974 4333 1032 XSEQ, TAD ID
4975 4334 7041 CIA
4976 4335 1131 TAD TARGET
4977 4336 7740 SMA SZA CLA
4978 4337 5274 JMP RESEQUENCE
4979 /AND TEST FOR THE CONDITION
4980 /
4981 / (OD) <= (TARGET)
4982 /
4983 4340 1131 TAD TARGET
4984 4341 7041 CIA
4985 4342 1031 TAD OD
4986 4343 7740 SMA SZA CLA
4987 4344 5274 JMP RESEQUENCE
4988 4345 5661 JMP I XGETATRACK
4989 /
4990 /ERROR REPORTED FOR APT. INDICATES TEST PC IN ERROR. THE ONLY EXCEPTIONS
4991 /ARE FOR AN UNEXPECTED RX01 INTERRUPT,
4992 /AND A MISSING ERROR FLAG.
4993 /
4994 4346 0000 XAERRO, 0
4995 4347 3135 DCA XXX /STORE ERROR PC.
4996 4350 4472 CHEK22
4997 4351 7410 SKP
4998 4352 5746 JMP I XAERRO
4999 4353 0002 IOF
5000 4354 1135 TAD XXX /GET BACK ERROR PC.
5001 4355 6201 CDF 00
5002 4356 6272 CIF 70
5003 4357 5761 JMP I K6520
5004 4360 4461 HLT /SOMETHING WENT WRONG ON
5005 /ON REPORTING ERROR TO APT
5006 4361 6520 K6520, 6520
5007 4374 0177
5008 4375 4710
5009 4376 1276
5010 4377 0070
5011 / PAGE
5012 / OD+1 INCREMENTALLY TO ID; ID-1 DECREMENTALLY TO OD
5013 /
5013 4400 1146 SEQ010, TAD TTRACKS
5014 4401 1145 TAD TRACKS
5015 4402 7640 SZA CLA
5016 4403 5213 JMP XSEQ2
5017 4404 1145 TAD TRACKS
5018 4405 7104 CLL RAL
5019 4406 1377 TAD (2)
5020 4407 3145 DCA TRACKS / (TRACKS X 2)-2 = 152(MAX VAL)
5021 4410 1145 TAD TRACKS
5022 4411 7041 CIA

```

```

5023 4412 3146 DCA TTRACKS
5024 /INCREMENT IF (TRACKS) < [ (ID)-(OD) ]
5025 /DECREMENT IF (TRACKS) => [ (ID)-(OD) ]
5026 4413 1031 XSEQ2, TAD OD
5027 4414 7041 CIA
5028 4415 1032 TAD ID
5029 4416 1145 TAD TRACKS
5030 4417 7700 SMA CLA
5031 4420 5776* JMP SEQ001 / ID-1 TO OD
5032 4421 5775* JMP SEQ000 / OD+1 TO ID
5033 /
5034 /BOUNCE; ID,OD ONLY
5035 /
5036 4422 1146 SEQ3, TAD TTRACKS
5037 4423 1145 TAD TRACKS
5038 4424 7640 SZA CLA
5039 4425 5233 JMP XSEQ3
5040 4426 7344 CLL STA RAL
5041 4427 3145 DCA TRACKS
5042 4430 1145 TAD TRACKS
5043 4431 7041 CIA
5044 4432 3146 DCA TTRACKS
5045 4433 1131 XSEQ3, TAD TARGET
5046 4434 7041 CIA
5047 4435 1031 TAD OD
5048 4436 7650 SMA CLA
5049 4437 1032 TAD ID
5050 4440 7450 SMA
5051 4441 1031 TAD OD
5052 4442 3131 DCA TARGET
5053 4443 5774* JMP XSEQ
5054 /BOUNCE ; ID TO OD
5055 /
5056 /THEREFORE 76 (NOT 77) TRACKS ARE TESTED
5057 /
5058 4444 1146 SEQ100, TAD TTRACKS
5059 4445 1145 TAD TRACKS
5060 4446 7640 SZA CLA
5061 4447 5254 JMP Q40D
5062 /FIRST ENTRY INTO SEQUENCE # 4
5063 /
5064 /IF (TRACKS) = [ (ID)-(OD) ] IS A NEGATIVE ODD THEN INCREMENT (TRACKS)
5065 /
5066 4450 7201 CLA IAC
5067 4451 0145 AND TRACKS
5068 4452 7640 SZA CLA
5069 4453 2145 ISZ TRACKS
5070 /IF (TRACKS) = -ODD ; BOUNCE TO OUTSIDE DIAMETER, BUT
5071 /
5072 /IF (TRACKS) = -EVEN ; BOUNCE TO INSIDE DIAMETER
5073 /
5074 4454 7201 Q40D, CLA IAC
5075 4455 0145 AND TRACKS
5076 4456 7650 SMA CLA
5077 4457 5305 JMP Q46ID

```

```

5078 4460 7240 STA
5079 4461 1146 TAD TTRACKS
5080 4462 1145 TAD TRACKS
5081 4463 7110 CLL RAR
5082 4464 5313 JMP Q460D
5083 // (TRACKS) = -EVEN ; THEREFORE THE BOUNCE IS TO THE INSIDE DIAMETER
5084 //
5085 /Q4ID, TAD TTRACKS; TAD TRACKS; CLL RAR; CIA; TAD ID; DCA TARGET; JMP XSEQ
5086 /STROBE ID, OD ; ID=1, OD ; ID=2, OD ; ...ETC...
5087 /
5088 4465 1146 SEQ6, TAD TTRACKS
5089 4466 1145 TAD TRACKS
5090 4467 7640 SZA CLA
5091 4470 5300 JMP Q60D
5092 /FIRST ENTRY INTO SEQUENCE # 6
5093 /
5094 4471 1145 TAD TRACKS
5095 4472 7105 CLL IAC RAL
5096 4473 3145 DCA TRACKS
5097 4474 1145 TAD TRACKS
5098 4475 7041 CIA
5099 4476 3146 DCA TTRACKS
5100 4477 5305 JMP Q46ID
5101 /
5102 // IF (TARGFT) = (0D), THEN STROBE TO INSIDE DIAMETER
5103 /
5104 4500 1131 Q60D, TAD TARGET
5105 4501 7041 CIA
5106 4502 1031 TAD OD
5107 4503 7640 SZA CLA
5108 4504 5313 JMP Q460D
5109 /
5110 /
5111 /
5112 4505 1146 Q46ID, TAD TTRACKS
5113 4506 1145 TAD TRACKS
5114 4507 7110 CLL RAR
5115 4510 7041 CIA
5116 4511 1032 TAD ID
5117 4512 7410 SKP
5118 4513 1031 Q460D, TAD OD
5119 4514 3131 DCA TARGET
5120 4515 5774* JMP XSEQ
5121 /*****
5122 /CONSOLE
5123 /*****
5124
5125
5126 4516 0000 XCHECKCB, 0
5127 4517 7700 CLA /CHECK IF CONSOL IS ACTIVE
5128 4520 1422 TAD 22
5129 4521 0373 AND (400
5130 4522 7650 SNA CLA
5131 4523 5337 JMP NOTCLB /NOT ON ACTIVE CONSOLE
5132 4524 1716 TAD I XCHECKCB /GET CONSOLE CALL

```

```

5133 4525 3331 DCA PACKDO /STORE IT IN THIS LOC
5134 4526 4507 WAITTY /WAIT FOR PRINTING TO FONISH
5135 4527 6002 IOF /KILL INTERRUPT
5136 4530 6213 CDF CIF 10 /TO FIELD ONE
5137 4531 0000 PACKDO, 0000 /DO CONSOLE CALL
5138 4532 7000 NOP
5139 4533 6001 ION /RETURN FROM CALL
5140 4534 2316 EXITCK, ISZ XCHECKCB /BUMP RETURN
5141 4535 7200 CLA
5142 4536 5716 JMP I XCHECKCB /EXIT XCHECK
5143 4537 2316 NOTCLB, ISZ XCHECKCB /BUMP FOR NOT CONSOL CALL +2
5144 4540 5334 JMP EXITCK /LEAVE BY THIS MEANS
5145
5146
5147 /*****
5148 /CONSOLE
5149 /*****
5150 /THIS SUBROUTINE IS ENTERED FROM TESTS: T21, T22, T32, T33
5151 /
5152 /THE CONTENTS OF THE AC AT ENTRY IS THE PATTERN TO FILL THE BUFFER WITH
5153 /
5154 4541 4541 FR128BYTES, .
5155 4542 3164 DCA GOOD /THE CONTENTS OF GOOD IS THE PURE PATTERN
5156 4543 1372 TAD (100)
5157 4544 4437 LCD /FILL BUFFER (0-BIT MODE)
5158 4545 5350 JMP +3 /START BY WAITING FOR TRANSFER REQUEST
5159 4546 1164 TAD GOOD
5160 4547 4443 XDROUT /TRANSFER OUT TO SECTOR BUFFER
5161 4550 4444 STR
5162 4551 7410 SKP
5163 4552 5346 JMF -4 /WAIT FOR TRANSFER REQUEST FLAG
5164 4553 4446 SDN
5165 4554 5350 JMP -4 /WAIT FOR DONE FLAG
5166 4555 4445 SER
5167 4556 4465 OK
5168 4557 4452 E210, ERROR /UNEXPECTED ERROR FLAG
5169 4560 4502 SCOPE
5170 4561 5741 JMP I FR128BYTES
5171 4572 0100
5172 4573 0400
5173 4574 4333
5174 4575 4306
5175 4576 4317
5176 4577 0002
4600
PAGE
5177 /INITIALIZE THE NUMBER OF SECTORS AVAILABLE TO ACCESS (PER TRACK) VIA THE
5178 /DIFFERENCE BETWEEN THE CONTENTS OF PROGRAM LOCATIONS " FIRST " AND " LAST ".
5179 /
5180 4600 4600 XINITSECTORS, .
5181 4601 1034 TAD LAST
5182 4602 7040 CMA
5183 4603 1033 TAD FIRST
5184 4604 3122 DCA SECTORS
5185 4605 3307 DCA XSTARGET
5186 4606 5600 JMP I XINITSECTORS

```

```

5187 /
5188 /GET A SECTOR
5189 /
5190 /SECTOR ACCESS 1-32 (OCTAL)
5191 /
5192 4607 4607 XGETASECTOR, .
5193 /
5194 /SET (SSTART)
5195 /
5196 4610 7301 XXGETASECTOR, CLL CLA IAC / 1
5197 4611 3306 DCA IF
5198 4612 1120 TAD RDC
5199 4613 7450 SNA
5200 4614 2306 ISZ IF / + 1 RDC
5201 4615 7710 SPA CLA
5202 4616 5222 JMP .+4 / + 1 WRITE
5203 4617 7307 CLL CLA IAC RTL
5204 4620 0132 AND TESTP
5205 4621 7640 SZA CLA
5206 4622 2306 ISZ IF / + 1 8-BIT MODE
5207 /
5208 / (IF) IS:
5209 /
5210 / 1 IF READ
5211 / 2 IF WRITE
5212 / 2 IF WRITE 8-BIT MODE
5213 / 2 IF READ AND PROGRAM VERIFY
5214 / 3 IF READ AND PROGRAM VERIFY 8-BIT MODE
5215 / 2 IF READ 8-BIT MODE
5216 /
5217 4623 1307 NEXTSECTOR, TAD XSTARGET
5218 4624 7440 SZA
5219 4625 1306 TAD IF /INTERLEAVE FACTOR 1, 2, 3
5220 4626 7001 IAC /NOW INTERLEAVE FACTOR IS 2, 3, 4
5221 4627 3307 DCA XSTARGET
5222 4630 1307 TAD XSTARGET
5223 4631 1377 TAD (-33)
5224 4632 7510 SPA
5225 4633 5241 JMP OKSTARGET / < 33
5226 4634 7640 SZA CLA /SKIP IF = 33
5227 4635 7240 STA
5228 4636 1376 TAD (-31)
5229 4637 1307 TAD XSTARGET
5230 4640 3307 DCA XSTARGET
5231 4641 7200 OKSTARGET, CLA
5232 4642 1307 TAD XSTARGET
5233 4643 3124 DCA STARGET
5234 /
5235 /PREVIOUSLY AT L/S THE FOLLOWING CONDITION WAS TRUE
5236 /
5237 / 0 < (FIRST) <= (LAST)
5238 /
5239 /THEREFORE TEST FOR THE CONDITION
5240 /
5241 / (STARGET) <= (LAST)

```

```

5242 /
5243 4644 1034 TAD LAST
5244 4645 7041 CIA
5245 4646 1124 TAD STARGET
5246 4647 7740 SMA SZA CLA
5247 4650 5223 JMP NEXTSECTOR
5248 /
5249 /AND FOR THE CONDITION
5250 /
5251 / (FIRST) <= (STARGET)
5252 /
5253 4651 1124 TAD STARGET
5254 4652 7041 CIA
5255 4653 1033 TAD FIRST
5256 4654 7740 SMA SZA CLA
5257 4655 5223 JMP NEXTSECTOR
5258 /FORMAT (XTARGET) BITS 0-6 TRACK ; AND BITS 7-11 SECTOR
5259 /
5260 4656 1131 TAD TARGET
5261 4657 7104 CLL RAL
5262 4660 7006 RTL
5263 4661 7006 RTL
5264 4662 1124 TAD STARGET
5265 4663 3147 DCA XTARGET
5266 /AND SET THE HEADER WORDS 0, AND 1 OF THE WRITE BUFFER FOR RDC I.D.
5267 /
5268 /WORD 0 IS THE TRACK
5269 /WORD 1 IS THE UNIT (BIT 4) AND THE SECTOR (BITS 7-11)
5270 /
5271 4664 1131 TAD TARGET
5272 4665 3775 DCA WBUFFER
5273 4666 1774 TAD UNITZ
5274 4667 7106 CLL RTL
5275 4670 7006 RTL
5276 4671 7006 RTL
5277 4672 7004 RAL
5278 4673 1124 TAD STARGET
5279 4674 3773 DCA WBUFFER+1
5280 /WORDS 63 AND 64 OR 127 AND 128 ARE OUT-SUMCHECK WORDS
5281 /
5282 4675 1153 TAD PATSUMCHECK
5283 4676 1775 TAD WBUFFER
5284 4677 1773 TAD WBUFFER+1
5285 4700 3554 DCA I WORDX
5286 4701 1554 TAD I WORDX
5287 4702 1554 TAD I WORDX
5288 4703 7041 CIA
5289 4704 3555 DCA I WORDY
5290 4705 5607 JMP I XGETASECTOR
5291 /
5292 4706 0000 IF, 0
5293 4707 0000 XSTARGET, 0 / 1 TO 32
5294 /
5295 /RANDOM NUMBER GENERATOR
5296 / (EXIT IS WITH THE RANDOM # IN THE ACCUMULATOR)

```

```

5297 4710 4710  RANGEN, .
5298 4711 7301    CLL CLA IAC
5299 4712 1324    TAD R1
5300 4713 1325    TAD R2
5301 4714 7106    CLL RTL
5302 4715 3324    DCA R1
5303 4716 1325    TAD R2
5304 4717 7012    RTR
5305 4720 1324    TAD R1
5306 4721 3325    DCA R2
5307 4722 1325    TAD R2
5308 4723 5710    JMP I RANGEN
5309 4724 1234    R1, 1234
5310 4725 0765    R2, 0765
5311 4773 7114
5312 4774 4240
5313 4775 7113
5314 4776 7747
5315 4777 7745
5316 5000
5317 / PAGE
5318 /TYPE 4 OCTAL
5319 /
5319 5000 5000  XTY4OCT, .
5320 5001 7410    SKP
5321 5002 3160    DCA TTYBUSY /INITIALIZE TTYBUSY INDICAO.
5322 5003 4500    SETUP
5323 5004 5000    XTY4OCT
5324 5005 1600    TAD I XTY4OCT
5325 5006 2200    ISZ XTY4OCT
5326 5007 3262    DCA XOCTAL /FOR " OCTAL " ADDRESS
5327 5010 1662    TAD I XOCTAL
5328 5011 3262    DCA XOCTAL /OCTAL
5329 5012 7346    CLL STA RTL / -3
5330 5013 1117    TAD K7777 / -1
5331 5014 3263    DCA DIGITS
5332 5015 7346    SHIFT, CLL STA RTL / -3
5333 5016 3265    DCA SHIFTS
5334 5017 1262    TAD XOCTAL
5335 5020 7100    XSHIFT, CLL
5336 5021 7510    SPA
5337 5022 7020    CML
5338 5023 7004    RAL
5339 5024 2265    ISZ SHIFTS
5340 5025 5220    JMP XSHIFT
5341 5026 3262    DCA XOCTAL /NEW
5342 5027 1262    TAD XOCTAL
5343 5030 0173    AND K0007 /OCTAL MASK
5344 5031 7450    SNA
5345 5032 5242    JMPDIG, JMP ISZDIG /DO NOT PRINT LEADING ZEROS.
5346 5033 3264    DCA XXDMP /STORE NUMBER TO BE PRINTED
5347 5034 1172    TAD K7000
5348 5035 3232    DCA .-3 /NOP JMP ISZDIG, LEADING ZEROS WILL NOT BE PRINTED
5349 5036 1264    TAD XXDMP /RETURN VALUE TO BE PRINTED.
5350 5037 1377    TAD (260) /FOR ASCII COMPONENT

```

```

5351 5040 4506    TYPEIT
5352 5041 2266    ISZ CCNT /INDICATES A CHARACTER HAS BEEN PRINTED
5353 5042 2263    ISZDIG, ISZ DIGITS /INDEX DIGIT COUNT
5354 5043 5215    JMP SHIFT
5355 5044 2174    ISZ XCNT /TY8OCT?
5356 5045 5202    JMP XTY4OCT+2 /YES
5357 5046 1266    TAD CCNT
5358 5047 7650    SNA CLA /PRINT ONLY A ZERO??
5359 5050 5257    JMP ZERO /YES
5360 5051 7340    CLL CLA CMA
5361 5052 3174    DCA XCNT /INIT COUNTER
5362 5053 3266    DCA CCNT
5363 5054 1376    TAD (JMP ISZDIG) /ESTABLISH FIRST TIME SWITCH.
5364 5055 3232    DCA JMPDIG
5365 5056 5775    JMP THENEXIT
5366 5057 1377    ZERO, TAD (260)
5367 5060 4506    TYPEIT
5368 5061 5251    JMP .-10
5369 5062 0000    XOCTAL, 0 / " OCTAL " FOR TYPEOUT
5370 5063 7775    DIGITS, -3
5371 5064 0000    XXDMP, 0
5372 5065 7775    SHIFTS, -3
5373 5066 0000    CCNT, 0
5374 /THIS ROUTINE WILL DETERMINE DEVICE CODE TO USE IN PLACE
5375 /OF THE STANDARD 75 CODE.
5376 /
5377 /*****
5378 5067 0000    CHNDEV, 0
5379 5070 1114    TAD DIESTP /GET TESTING PARAMETERS.
5380 5071 0374    AND (700) /ISOLATE DEVICE CODE TO USE.
5381 5072 7450    SNA /USE 75??
5382 5073 5667    JMP I CHNDEV /YES.
5383 5074 7112    CLL RTR
5384 5075 7010    RAR /MOVE TO BIT POSITION 6-8.
5385 5076 1373    TAD (6701)
5386 /PROPER DEVICE CODE SHOULD NOW BE ESTABLISHED.
5387 /IF THE OPERATOR HAS MADE AN ERROR THE PROGRAM HAS NO WAY
5388 /OF KNOWING IT.
5389 /
5390 5077 3010    DCA A10 /SET DEVICE CODE = 67X-
5391 5100 1372    TAD (XDEVIC-1)
5392 5101 3011    DCA A11 /INTO APPLICABLE PROGRAM
5393 5102 1411    TAD I A11
5394 5103 7450    SNA
5395 5104 5312    JMP ,+6 /DEVICE CODE LOCATIONS
5396 5105 3164    DCA GOOD
5397 5106 1010    TAD A10
5398 5107 3564    DCA I GOOD /SPECIFIED AT
5399 5110 2010    ISZ A10
5400 5111 5302    JMP .-7 /PROGRAM LOCATION "DEVICE"
5401 5112 1726    TAD I XK67X2A
5402 5113 3771    DCA K67X2B /DUPLICATE IOT CODE 67X2
5403 5114 1727    TAD I XK67X3A
5404 5115 3770    DCA K67X3B /DUPLICATE IOT CODE 67X3
5405 5116 1730    TAD I XK67X4A

```

```

5406 5117 3767* DCA K67X4B /DUPLICATE IOT CODE 67X4
5407 5120 1731 TAD I XK67X5A
5408 5121 3766* DCA K67X5B /DUPLICATE IOT CODE 67X5
5409 5122 1733 TAD I XK67X7A
5410 5123 3765* DCA K67X7B /DUPLICATE IOT CODE 67X7
5411 5124 5667 JMP I CHNDEV /EXIT AND RUN PROGRAM.
5412 /PROGRAM DEVICE CODE TABLE
5413 /
5414 5125 6203 XDEVICE, K67X1
5415 5126 6401 XK67X2A, K67X2A /XDR TRANSFER DATA REGISTER
5416 5127 6410 XK67X3A, K67X3A /STR (SKIP ON TRANSFER REQUEST FLAG)
5417 5130 6415 XK67X4A, K67X4A /SER (SKIP ON ERROR FLAG)
5418 5131 6422 XK67X5A, K67X5A /SDN (SKIP ON DONE FLAG)
5419 5132 6431 K67X6 /CLEAR (AC = 0) /SET (AC = 1) INTERRUPT ENABLE
5420 5133 6435 XK67X7A, K67X7A /INIT (INITIALIZE / IMPLIED READ TRACK 0 SECTOR 1)
5421 5134 0000 0 /
5422 /
5423 5165 6450
5424 5166 0712
5425 5167 0726
5426 5170 0720
5427 5171 6404
5428 5172 5124
5429 5173 6701
5430 5174 0700
5431 5175 4110
5432 5176 5242
5433 5177 0260
5434 PAGE
5435 /
5436 /ROUTINE TO TYPE 8 OCTAL DIGITS.
5437 /FORMAT TY8OCT; MSB(STARTING ADDRESS OF DOUBLE LOG TO BE PRINTED).
5438 /
5439 5200 0000 XTY8OCT, 0
5440 5201 1600 TAD I XTY8OCT /GET STARTING ADDRESS
5441 5202 3212 DCA .+10
5442 5203 1600 TAD I XTY8OCT
5443 5204 7001 LAC /SECOND ADDRESS
5444 5205 3211 DCA .+4
5445 5206 7344 CLL CLA CMA RAL /-2
5446 5207 3174 DCA XCNT
5447 5210 4505 TY4OCT
5448 5211 7000 NOP /REPLACED WITH POINTER TO BE
5449 5212 7000 NOP /PRINTED
5450 5213 2700 ISZ XTY8OCT /UPDATE RETURN
5451 5214 5600 JMP I XTY8OCT /EXIT
5452 /SUBROUTINE ; TAB
5453 /ENTRY ; TAB; +N
5454 /COMMENT ; PRINT " N " SPACES WHERE N IS VIA INDEXED XTAB
5455 XTAB, .
5456 5215 5215 SETUP
5457 5216 4500 XTAB
5458 5217 5215 DCA XXTAB /FOR COUNT
5459 5221 7200 XTABL, CLA

```

```

5460 5222 1243 TAD XXTAB
5461 5223 1615 TAD I XTAB /FOR " N "
5462 5224 3243 DCA XXTAB
5463 5225 1320 TAD CHARLINE / # OF CHARACTERS ALREADY TYPED ON THIS LINE
5464 5226 1377 TAD (110) / 72 CHARACTER LINE STANDARD
5465 5227 7041 CIA
5466 5230 1243 TAD XXTAB
5467 5231 7550 SFA SNA
5468 5232 5221 JMP XTABL
5469 5233 7041 CIA
5470 5234 3243 DCA XXTAB
5471 5235 2215 ISZ XTAB
5472 5236 1376 TAD (240)
5473 5237 4506 TYPEIT
5474 5240 2243 ISZ XXTAB
5475 5241 5236 JMP .-3
5476 5242 5775* JMP THENEXIT
5477 5243 0000 XXTAB, 0
5478 /
5479 /SUBROUTINE ; PRINT
5480 /ENTRY ;
5481 /COMMENT ; PRINT A " MESSAGE ", AND A <CR><LF> AT EACH _
5482 /
5483 / CALL SYNTAX FOR PRINT
5484 /
5485 / 1. PRINT; MTEXT
5486 /
5487 5244 5244 XPRINT, .
5488 5245 4472 CHEK22
5489 5246 5340 JMP PNTEXT
5490 5247 4507 WAITTY
5491 5250 1644 TAD I XPRINT / " TEXT"
5492 5251 2244 ISZ XPRINT
5493 5252 3262 DCA MESSAGE /ADDRESS
5494 5253 1662 NUWORD, TAD I MESSAGE
5495 5254 4263 OUTPUT /LEFT BYTE
5496 5255 1662 TAD I MESSAGE
5497 5256 2262 ISZ MESSAGE
5498 5257 4435 BSW
5499 5260 4263 OUTPUT /RIGHT BYTE
5500 5261 5253 JMP NUWORD
5501 5262 0000 MESSAGE,0
5502 5263 4263 OUTPUT=JMS .
5503 5263 5263 XOUTPUT, .
5504 5264 0374 AND (-100) /MASK MS BITS 0-5
5505 5265 7440 SZA
5506 5266 5271 JMP .+3
5507 5267 3160 DCA TTYBUSY
5508 5270 5773* JMP PIEXIT
5509 5271 1372 TAD (4100)
5510 5272 7450 SNA
5511 5273 5302 JMP NULINE / -
5512 5274 1371 TAD (-4100+2) /CODE 200
5513 5275 7500 SNA
5514 5276 7001 IAC /CODE 300

```



```

5515 5277 4435      BSW
5516 5300 4506      TYPEIT
5517 5301 5063      JMP I XOUTPUT
5518 5302 1263      NULINE, TAD XOUTPUT
5519 5303 3305      DCA XTYPEIT
5520 5304 5311      JMP XNULINE
5521 5305 5305      XTYPEIT, .
5522 5306 4321      JMS TYIASC
5523 5307 2320      ISZ CHARLINE
5524 5310 5705      JMP I XTYPEIT
5525 5311 1370      XNULINE, TAD (15) / <CR>
5526 5312 4321      JMS TYIASC
5527 5313 1367      TAD (12) / <LF>
5528 5314 4321      JMS TYIASC
5529 5315 1366      TAD (-110)
5530 5316 3320      DCA CHARLINE / 72 CHARACTER LINE (NEGATIVE NOTATION)
5531 5317 5705      JMP I XTYPEIT
5532 5320 7670      CHARLINE, -110 / # CHARACTERS PER LINE ARE COUNTED HERE
5533 /
5534 0014 A14=14
5535 /
5536 5321 5321      TYIASC, .
5537 5322 6002      IOF
5538 5323 3014      DCA A14
5539 5324 4472      CHKZ2
5540 5325 5340      JMP PNTEXT
5541 5326 1014      TAD A14
5542 5327 6046      TLS
5543 5330 7200      CLA
5544 5331 1160      TAD TTYBUSY
5545 5332 7040      SZA CLA
5546 5333 5773      JMP PIEXIT
5547 5334 2100      ISZ TTYBUSY
5548 5335 1365      TAD (PI)
5549 5336 3002      DCA IPI
5550 5337 7410      SKP
5551 5340 2244      PNTEXT, ISZ XPRINT /UPDATE RETURN.
5552 5341 6001      ION
5553 5342 5644      JMP I XPRINT
5554 /
5555 /
5556 5343 6042      XTCF, TCF
5557 5344 1014      TAD A14
5558 5345 1364      TAD (-207)
5559 5346 7050      SMA CLA
5560 5347 5705      JMP I XTYPEIT
5561 5350 5721      JMP I TYIASC
5562 /
5563 /
5564 /
5565 5351 5351      XSPECIALTYPEIT, .
5566 5352 4500      SETUP
5567 5353 5351      XSPECIALTYPEIT
5568 5354 1751      TAD I XSPECIALTYPEIT
5569 5355 7351      ISZ XSPECIALTYPEIT

```

```

5570 5356 4506      TYPEIT
5571 5357 5775      JMP THENEXIT
5572 5364 7571      /
5573 5365 5405      /
5574 5366 7670      /
5575 5367 0012      /
5576 5370 0015      /
5577 5371 3702      /
5578 5372 4100      /
5579 5373 5476      /
5580 5374 7700      /
5581 5375 4110      /
5582 5376 0240      /
5583 5377 0110      /
5584 /
5585 /
5586 /
5587 /
5588 5400 5400      XWAITY, .
5589 5401 1160      TAD TTYBUSY
5590 5402 7040      SZA CLA
5591 5403 5201      JMP ,-2
5592 5404 5600      JMP I XWAITY
5593 /ENTRY TO THIS POINT WAS CAUSED BY A PROGRAM INTERRUPT REQUEST
5594 /
5595 5405 3303      PI, DCA XAC
5596 5406 7010      RAR
5597 5407 3304      DCA XLINK /SAVE (AC) AND (LINK)
5598 5410 6031      KSF
5599 5411 7410      SKP
5600 5412 5777      JMP XKCC /IGNORE KEYBOARD IRQ
5601 5413 6041      TSF
5602 5414 5222      JMP PISDN
5603 /IF THIS TELEPRINTER FLAG IS EXPECTED (TTYBUSY) = 1
5604 /THEN "JMP XTCF"
5605 /IF NOT THEN "JMP PIEXIT"
5606 /
5607 5415 1160      TAD TTYBUSY
5608 5416 7740      SMA SZA CLA
5609 5417 5776      JMP XTCF
5610 5420 6042      TCF
5611 5421 5276      JMP PIEXIT
5612 /IF (RUSY) = 1, THEN AN RX01 PI IS ALREADY BEING PROCESSED
5613 /
5614 /IF (GOBIT) = 0, THEN THIS DISKETTE IRQ IS UNEXPECTED
5615 /
5616 /DISABLE RX01 INTERRUPT
5617 /
5618 /READ RX01 STATUS REGISTER
5619 /
5620 5422 1160      PISDN, TAD TTYBUSY
5621 5423 1111      TAD BUSY
5622 5424 7040      SZA CLA
5623 5425 5276      JMP PIEXIT

```

```

5624 5426 2111 ISZ BUSY
5625 /REFRESH PROGRAM LOCATION * FORCE *
5626 /
5627 5427 3313 DCA XXFORCE
5628 /
5629 5430 4446 SDN
5630 5431 5273 JMP UNKNOWN
5631 5432 1152 TAD GOBIT
5632 5433 7750 SPA SNA CLA
5633 5434 5775* JMP SDNUNEXPECTED
5634 5435 4447 INTR
5635 5436 4442 XDRIN
5636 5437 0374 AND (377)
5637 5440 3126 DCA ASTATUS
5638 /IF THIS IS TEST # 30 THEN IGNORE DELETED DATA MARKS (IF ANY)
5639 /
5640 5441 1167 TAD TEST
5641 5442 1373 TAD (-T30)
5642 5443 7650 SNA CLA
5643 5444 5263 JMP DDIGNOPE
5644 /TECHNICAL NOTE:
5645 /
5646 /THE COMMANDS "FILL BUFFER" (0), AND "EMPTY BUFFER" (2)
5647 /NEVER SHOULD ATTRACT THE "DELETED DATA" STATUS (100)
5648 /
5649 5445 1372 TAD (16) /COMMAND MASK
5650 5446 0112 AND COMMAND /FB (0), OR EB (2)
5651 5447 7440 SZA /SKIP IF FILL BUFFER COMMAND (0)
5652 5450 7112 CLL RTR
5653 5451 7640 SZA CLA /SKIP IF EMPTY BUFFER COMMAND (2)
5654 5452 7305 CLL CLA IAC RAL / 2
5655 5453 0132 AND TESTP
5656 5454 7112 CLL RTR /PUT TO LINK
5657 5455 1371 TAL (100) / 100
5658 5456 0126 AND ASTATUS / A STATUS D.D. MASK
5659 /
5660 /IF (L) = 0 AND (AC) = 0, O.K. - NO D.D. MARK
5661 /
5662 /IF (L) = 0 AND (AC) > 0 (=100), UNEXPECTED D.D.
5663 /
5664 /IF (L) = 1 AND (AC) = 0, D.D. MARK EXPECTED DIDN'T OCCUR
5665 /
5666 /IF (L) = 1 AND (AC) > 0 (=100), O.K. - D.D. MARK OCCURED
5667 /
5668 5457 7430 SZL
5669 5460 7640 SZA CLA / (L) = 1
5670 5461 7440 SZA / (L) = 0, OR (L) = 1 AND (AC) > 0
5671 5462 5325 JMP DERROR / (L) = 0 AND (AC) > 0 (=100) , OR (L) = 1 AND (AC) = 0
5672 5463 4445 DDIGNOPE, SEP
5673 5464 5266 JMP VERIFY /RX01 OK - RETURN TO INLINE CODE
5674 5465 5773* JMP RXERROR
5675 /
5676 /VERIFY THAT THE CONTENTS OF THE A-STATUS REGISTER = 0
5677 /
5678 /WHEN NO RX01 ERROR FLAG EXISTS
    
```

```

5679 /
5680 /MASK BITS 4 (DRIVE READY); AND 5 (DELETED DATA)
5681 /
5682 5466 1126 VERIFY, TAD ASTATUS
5683 5467 0370 AND (73)
5684 5470 7640 SZA CLA
5685 5471 5767* JMP NOSEK
5686 5472 5307 JMP XRETURN
5687 /
5688 /AN UNKNOWN PROGRAM INTEPRUPT OCCURED
5689 /
5690 5473 4473 UNKNOWN, PRINT
5691 5474 7067 MUNKNOVN
5692 5475 3111 DCA BUSY
5693 /
5694 5476 1304 PIEXIT, TAD XLINK
5695 5477 7104 CLL RAL
5696 5500 1303 TAD XAC
5697 5501 6001 ION
5698 5502 5400 JMP I 0
5699 5503 0000 XAC, 0
5700 5504 0000 XLINK, 0
5701 /THE CONTENTS OF RETURN ARE SETUP WITHIN THE SUBROUTINES "LCD-A" AND "LCD-B"
5702 /TO REPRESENT THE RETURN ADDRESS OF THE INLINE TESTING
5703 /
5704 5505 0000 RETURN, 0
5705 /
5706 5506 2305 ERETURN, ISZ RETURN /INCREMENT FOR ERROR RETURN ADDRESS
5707 /
5708 /ENTRY TO HERE FROM PI SERVICE
5709 /
5710 /NO RX01 ERROR FLAG EXISTS
5711 /
5712 5507 3111 XRETURN, DCA BUSY
5713 5510 4466 TICK /TIMING FOR APT IF NEEDED.
5714 5511 6001 ION
5715 5512 5705 JMP I RETURN
5716 /...ENTRY TO THIS POINT MAY HAVE BEEN FROM WITHIN THE SUBROUTINE "COMPARE"
5717 /WHICH DETECTED A DATA NO STATUS ERROR (DNS), OR
5718 /...ENTRY TO THIS POINT MAY HAVE BEEN FROM ROUTINES "XHUNGUP" OR "HUNGUP"
5719 /THEREBY FORCING AN ERROR INFORMATION PRINTOUT
5720 /
5721 5513 0000 XXFORCE, 0
5722 5514 2111 ISZ BUSY
5723 5515 5773* JMP RXERROR
5724 5516 7240 XFORCE, STA
5725 5517 1111 TAD BUSY
5726 5520 3111 DCA BUSY
5727 5521 6001 ION
5728 5522 5713 JMP I XXFORCE
5729 5523 7017 DTYPE, MEDDDIDNOT
5730 5524 7030 MUDDID
5731 /A DISKETTE DELETED DATA MALFUNCTION HAS BEEN DETECTED
5732 /
5733 / IF (AC) = 0 - EXPECTED D.D. DIDN'T OCCUR
    
```

```

5734 / IF (AC) = 100 - UNEXPECTED D.D. OCCURED
5735 /
5736 5525 7640 DDERROR, SZA CLA / 0 OR 1
5737 5526 7001 IAC
5738 5527 1366 TAD (DTYPE)
5739 5530 3135 DCA XXX
5740 /PROGRAMMING NOTE: "SER" TO CLEAR ACCOMPANYING ERROR FLAG (IF ANY)
5741 /
5742 5531 4445 SER
5743 5532 7000 NOP
5744 5533 1535 TAD I XXX
5745 5534 5773* JMP RXERROR
5746 5566 5523
5747 5567 3322
5748 5570 0073
5749 5571 0100
5750 5572 0016
5751 5573 5600
5752 5574 0377
5753 5575 3320
5754 5576 5343
5755 5577 0314
5600 PAGE
5756 /A DISKETTE ERROR HAS BEEN DETECTED
5757 /
5758 / (DTYPE) NOT = 0 IF A D.D. ERROR EXISTS
5759 / (DTYPE) = 0 IF NO D.D. ERROR EXISTS
5760 /
5761 5600 3777* RXERROR, DCA DTYPE
5762 /
5763 5601 1112 TAD COMMAND
5764 5602 3150 DCA ECOMMAND
5765 5603 1776* TAD XXFORCE
5766 5604 1375 TAD (-XHING)
5767 5605 7650 SNA CLA
5768 5606 5246 JMP EERROR
5769 5607 1776* TAD XXFORCE
5770 5610 1374 TAD (-XCOMPARE)
5771 5611 7650 SNA CLA
5772 5612 5246 JMP EERROR
5773 5613 1373 SAVEBSTATUS, TAD (16)
5774 5614 4437 LCD
5775 5615 4471 WAIT
5776 5616 4416 SDN
5777 5617 5215 JMP ,=-2
5778 5620 3006 DCA 6
5779 5621 4445 SER /WAIT POINTER
5780 5622 7000 NOP
5781 5623 4442 XDRIN
5782 5624 0372 AND (377)
5783 5625 3127 DCA BSTATUS
5784 5626 1771* SAVECSTATUS, TAD UNIT
5785 5627 1370 TAD (12)
5786 5630 4437 LCD
5787 5631 4471 WAIT

```

```

5788 5632 4446 SDN
5789 5633 5231 JMP ,=-2
5790 5634 3006 DCA 0 /WAIT POINTER
5791 5635 4445 SER
5792 5636 7000 NOP
5793 5637 4442 XDRIN
5794 5640 0372 AND (377)
5795 5641 3130 DCA CSTATUS
5796 /PRINT AN ERROR MESSAGE IF AC SW 3 = 0
5797 /
5798 5642 4570 LASSW3, LAS
5799 5643 0367 AND (SW3)
5800 5644 7640 SZA CLA
5801 5645 5766* JMP NOPRINT
5802 /THE FOLLOWING INFORMATION IS PRINTED FOR ALL ERRORS DETECTED
5803 /
5804 /THE ERROR HEADER TEXT IS INHIBITED IF THE ERROR IS NOT THE FIRST ERROR EVER
5805 /
5806 /
5807 /
5808 / FAT CMND XDR CODE RST START TARGET XXXX PASS
5809 /
5810 /
5811 /
5812 5646 1167 EERROR, TAD TEST /GET TEST IN ERROR
5813 5647 4467 AERROR /REPORT ERROR TO APT.
5814 5650 2115 ISZ FIRSTERROR
5815 5651 5255 JMP NOHEADER
5816 5652 4473 PRINT
5817 5653 6626 MEHEADER
5818 5654 5271 JMP ONECRLF
5819 /IF THIS IS =NOT= A FORCED TYPEOUT, AND IF THERE ARE NO DATA COMPARE
5820 /ERRORS (COMPRERROR=0), THEN PRINT ONLY 1-CRLF
5821 /BECAUSE
5822 /
5823 /THIS ERROR MUST BE AN ERROR AT THE END OF THE EMPTY BUFFER DONE FLAG
5824 /WHICH WOULD BE ASSOCIATED TO ANY PREVIOUS FORCED TYPEOUT OF DATA ERRORS
5825 /
5826 5655 1776* NOHEADER, TAD XXFORCE
5827 5656 7650 SNA CLA
5828 5657 5267 JMP TWOCRLF
5829 5660 1113 TAD COMPRERROR
5830 5661 7640 SZA CLA
5831 5662 5271 JMP ONECRLF
5832 /IF THIS IS A FORCED TYPEOUT FROM " XXINIT " THEN PRINT 1 CRLF
5833 /
5834 5663 1776* TAD XXFORCE
5835 5664 1365 TAD (-XXINIT)
5836 5665 7650 SNA CLA
5837 5666 5271 JMP ONECRLF
5838 5667 4473 TWOCRLF, PRINT
5839 5670 6601 MCRLF
5840 5671 4473 ONECRLF, PRINT
5841 5672 6601 MCRLF
5842 5673 4505 TY4OCT

```

```

5843 5674 0167 TEST
5844 5675 4504 TAB
5845 5676 0005 5
5846 5677 4764 JMS INITSWITCH
5847 5700 5305 JMP .+5
5848
5849 /IF AN ERROR FROM THE RECAL THEN PRINT [INIT] FOR THE COMAND
5849 /
5850 5701 1776 TAD XXFORCE
5851 5702 1365 TAD (-XXINIT)
5852 5703 7640 SZA CLA
5853 5704 5310 JMP .+4
5854 5705 4473 PRINT
5855 5706 6670 MINIT
5856 5707 5312 JMP TAB12
5857 5710 4505 TY4OCT
5858 5711 0150 ECOMMAND
5859 5712 4504 TAB12, TAB
5860 5713 0012 12
5861 /
5862 /IF THE DEVICE TEST IS HUNG, THEN THE A-, B-, AND C- STATUS IS NOT APPLICABLE
5863 /
5864 5714 1776 TAD XXFORCE
5865 5715 1375 TAD (-XHUNG)
5866 5716 7650 SNA CLA
5867 5717 5325 JMP DASHALL
5868
5869 /IF THIS IS A " FORCED " TYPEOUT THEN THE B- AND C-STATUS REGISTERS
5870 /ARE NOT APPLICABLE TO THIS TYPEOUT BECAUSE THEY ARE RESIDUAL FROM THE
5871 /PREVIOUS COMMAND WHICH WOULD HAVE HAD A PREVIOUS ERROR TYPE OUT
5872 /RELATING TO THE B- AND C-STATUS REGISTERS IF AN ERROR HAD OCCURED
5873 /
5874 /A DATA NO ERROR STATUS HAS BEEN DETECTED PRIOR TO THE COMPLETION OF
5875 /THE EMPTY BUFFER FUNCTION
5876 /
5876 5720 1776 TAD XXFORCE
5877 5721 1374 TAD (-XCOMPARE)
5878 5722 7640 SZA CLA
5879 5723 5340 JMP TYASTATUS
5880 5724 5331 JMP DASHBC
5881 5725 4473 DASHALL, PRINT
5882 5726 7016 MDASH
5883 5727 4504 TAB
5884 5730 0017 17
5885 5731 4473 DASHBC, PRINT
5886 5732 7016 MDASH
5887 5733 4504 TAP
5888 5734 0024 24
5889 5735 4473 PRINT
5890 5736 7016 MDASH; JMP TAB31
5891 5737 5763 TYASTATUS, TY4OCT
5892 5740 4505 ASTATUS
5893 5741 0126 TAB
5894 5742 4504 17
5895 5743 0017 TY4OCT
5896 5744 4505 bSTATUS
5897 5745 0127

```

```

5898 5746 4504 TAB
5899 5747 0024 24
5900 5750 5762 JMP TYCSTATUS
5901 5762 0000
5902 5763 0002
5903 5764 0115
5904 5765 1333
5905 5766 0061
5906 5767 0400
5907 5770 0012
5908 5771 4242
5909 5772 0377
5910 5773 0016
5911 5774 4116
5912 5775 4434
5913 5776 5513
5914 5777 0060
5915 6000 4505 PAGE
5916 6001 0130 TYCSTATUS, TY4OCT
5917 6002 4504 CSTATUS
5918 6003 0031 TAB31, TAB
5919 6004 4315 31
5920 6005 5211 JMS INITSWITCH
5921 JMP .+4
5922 /IF (SSTART) = 0 THEN PRINT "HOME" BECAUSE A RECAL HAS TAKEN PLACE
5923 /THEREFORE THE ACTUATOR IS AT TRACK 0 (HOME)
5924 /
5924 6006 1123 TAD SSTART
5925 6007 7640 SZA CLA
5926 6010 5214 JMP .+4
5927 6011 4473 PRINT
5928 6012 6664 MHOME
5929 6013 5222 JMP TAB43
5930 6014 4505 TY4OCT
5931 6015 0125 START
5932 6016 4504 TAB
5933 6017 0036 36
5934 6020 4505 TY4OCT
5935 6021 0123 SSTART
5936 6022 4504 TAB43, TAB
5937 6023 0043 43
5938 6024 4315 JMS INITSWITCH
5939 6025 5232 JMP PHOME
5940 /IF (FORCE) = THE ADDRESS OF "XXINIT" THEN ALSO PRINT [HOME] FOR THE TARGET
5941 /
5942 6026 1777 TAD XXFORCE
5943 6027 1376 TAD (-XXINIT)
5944 6030 7640 SZA CLA
5945 6031 5235 JMP .+4
5946 6032 4473 PHOME, PRINT
5947 6033 6664 MHOME
5948 6034 5243 JMP .+7
5949 6035 4505 TY4OCT
5950 6036 0131 TARGET
5951 6037 4504 TAB

```

```

5952 6040 0050 50
5953 6041 4505 TY4OCT
5954 6042 0124 STARGET
5955 6043 4504 TAB
5956 6044 0055 55
5957 6045 4505 TY4OCT
5958 6046 0132 TFSTP
5959 6047 4504 TAB
5960 6050 0002 2
5961 6051 4436 TY8OCT
5962 6052 0161 PASS
5963 6053 0162 PASS+1
5964 6054 1260 TAD DMITYPE
5965 6055 7650 SNA CLA
5966 6056 5261 JMP ,+3
5967 6057 4473 PRINT
5968 6060 0000 DMITYPE, 0
5969 /
5970 /AC SW 11 TO INHIBIT RINGING OF BELL AT ERROR
5971 /
5972 6061 3115 NOPPINT, DCA FIRSTERROR
5973 6062 4570 LAS
5974 6063 0375 AND (SW11)
5975 6064 7640 SZA CLA
5976 6065 5270 JMP ,+3
5977 6066 4501 SPECIALTYPEIT
5978 6067 0207 BELL
5979 /IF ENTPY WAS FROM A "JMS FORCE" THEN EXIT BY A "JMP I FORCE"
5980 /
5981 6070 1777 TAD XXFORCE
5982 6071 7640 SZA CLA
5983 6072 5774 JMP XFORCE
5984 /
5985 /DEFINITIVE ERROR CODES AND MEANINGS
5986 /
5987 / 0 /NO ERROR
5988 / 10 /DRIVE 0 FAILED TO SEE HOME ON INITIALIZE
5989 / 20 /DRIVE 1 FAILED TO SEE HOME ON INITIALIZE
5990 / 30 /FOUND HOME WHEN STEPPING OUT 10 TRACKS FOR INIT
5991 / 40 /TRIED TO ACCESS A TRACK GREATER THAN 77
5992 / 50 /HOME WAS FOUND BEFORE DESIRED TRACK WAS REACHED
5993 / 60 /SELF DIAGNOSTIC ERROR
5994 / 70 /DESIRED SECTOR COULD NOT BE FOUND AFTER LOOKING AT 52 HEADERS
5995 / 100 /WRITE PROTECT ERROR
5996 / 110 /MORE THAN 40US AND NO SEP CLOCK SEEN
5997 / 120 /A PREAMBLE COULD NOT BE FOUND
5998 / 130 /PREAMBLE FOUND BUT NO ID MARK FOUND WITHIN ALLOWABLE TIME
5999 / 140 /HEADER CRC ERKOP
6000 / 150 /THE HEADER TRACK ADDRESS OF A GOOD HEADER DOES NOT COMPARE
6001 / /WITH THE DESIRED TRACK
6002 / 160 /TO MANY TRIES FOR A IDAM
6003 / 170 /DATA AM NOT FOUNT WITHIN ALLOTTED TIME
6004 / 200 /DATA CRC ERROR
6005 / 210 /ALL PARITY ERRORS
6006 /

```

```

6007 /RECAL IF DEFINITIVE ERROR CODE IS A SEEK EPROR
6008 /
6009 / (NOT CODES 140, 200, OR 210)
6010 /
6011 6073 4570 RECALIF, LAS
6012 6074 0373 AND (SW6)
6013 6075 7640 SZA CLA
6014 6076 5311 JMP LASSW0
6015 6077 1127 TAD BSTATUS
6016 6100 1372 TAD (-140)
6017 6101 7450 SNA
6018 6102 5311 JMP LASSW0
6019 6103 1371 TAD (-40)
6020 6104 7450 SNA
6021 6105 5311 JMP LASSW0
6022 6106 1370 TAD (-10)
6023 6107 7640 SZA CLA
6024 6110 4450 INIT
6025 6111 4570 LASSW0, LAS
6026 6112 7710 SPA CLA
6027 6113 4461 HLT10, HLT /AC SW 0 = 1 (HALT ON ERROR)
6028 6114 5767 JMP EPETURN
6029 /
6030 6115 6115 INITSWITCH, .
6031 6116 1766 TAD XWAIT
6032 6117 1365 TAD (-SDNSECOND)
6033 6120 7640 SZA CLA
6034 6121 2315 ISZ INITSWITCH
6035 6122 5715 JMP I INITSWITCH
6036 /
6037 /GENERATES TIMING FOR APT IF NEEDED.
6038 /
6039 6123 0000 XOK, 0
6040 6124 7000 NOP
6041 6125 2323 ISZ XOK /SKIP.
6042 6126 5723 JMP I XOK
6043 /
6044 6165 1341
6045 6166 1722
6046 6167 5506
6047 6170 7770
6048 6171 7740
6049 6172 7640
6050 6173 0040
6051 6174 5516
6052 6175 0001
6053 6176 1333
6054 6177 5513
6055 6200 PAGE
6056 /THE CONTENTS OF THE AC AT ENTRY ARE THE CONTENTS OF PROGRAM LOCATION "TCOMMAND"
6057 6200 6200 XLCD, .
6058 6201 3112 DCA COMMAND
6059 6202 1112 TAD COMMAND
6060 6203 6751 K67X1, 6751

```

```

6061      6204 5600      JMP I XLCD
6062      /LOAD THE COMMAND FOR: FILL BUFFER, AND EMPTY BUFFER
6063      /WITH THE RX01 INTERRUPT ENABLED
6064      /
6065      /FORM: (AC) IS COMMAND; LCDA; NORMAL RETURN; ERROR RETURN
6066      /
6067      6205 6205      XLCD,      .
6068      6206 1777*      TAD UNIT
6069      6207 3112      DCA COMMAND
6070      /THE CONTENTS OF THE AC WILL = 100 IF 8 BIT MODE
6071      /
6072      6210 7307      CLL CIA IAC RTL
6073      6211 0132      AND TESTP
6074      6212 7640      SZA CIA
6075      6213 1376      TAD (100)
6076      6214 1112      TAD COMMAND
6077      6215 3112      DCA COMMAND
6078      6216 1112      TAD COMMAND
6079      6217 4437      LCD
6080      6220 1205      TAD XLCD
6081      6221 3775*      DCA RETURN
6082      6222 2205      ISZ XLCD
6083      6223 2205      ISZ XLCD
6084      6224 1374      TAD (PI)
6085      6225 3002      DCA IPI
6086      6226 7201      CLA IAC
6087      6227 4447      INTR
6088      6230 6001      ION
6089      6231 5605      JMP I XLCD
6090      /
6091      /LOAD THE COMMAND AND THE TRACK AND SECTOR ADDRESSES AND GO WITH INTERRUPT ENABLE !
6092      /
6093      /FORM: (AC) IS COMMAND; LCDB; NORMAL RETURN; ERROR
6094      /
6095      6232 6232      XLCD,      .
6096      6233 1777*      TAD UNIT      / 0 OR 20
6097      6234 3112      DCA COMMAND      /TEMPORARY STORAGE
6098      /
6099      /WHEN THE CONTENTS OF " GOBIT " ARE = 0 ; NO PROGRAM IRQ IS EXPECTED FROM THE DISK
6100      /
6101      6235 7307      CLL CIA IAC RTL      / 4
6102      6236 0132      AND TESTP      /TESTP FOR 8/12 BIT MODE SELECTION
6103      6237 7640      SZA CIA
6104      6240 1376      TAD (100)      / 100 FOR 8 BIT MODE SELECTION
6105      6241 1112      TAD COMMAND      / 4, 14 OR 6
6106      6242 3112      DCA COMMAND
6107      6243 1112      LCDBL, TAD COMMAND
6108      6244 4437      LCD
6109      /
6110      /LOAD THE TRACK AND SECTOR ADDRESSES FOR THE COMMANDS:
6111      /
6112      /WRITE, OR WRITE DELETED DATA, OR READ SECTOR
6113      /
6114      6245 1373      TAD (LCDBRETURN)
6115      6246 3775*      DCA RETURN
    
```

```

6116      6247 1372      TAD (NOP)
6117      6250 3312      DCA XLCDRETURN
6118      6251 7201      CLA IAC
6119      6252 4447      INTR      /ENABLE RX01 INTERRUPT
6120      6253 4444      STR
6121      6254 5253      JMP ,-1      /SKIP ON TRANSFER REQUEST FLAG
6122      6255 1124      TAD TARGET
6123      6256 4443      XDROUT      / SECTOR
6124      6257 4444      STR
6125      6260 5257      JMP ,-1      /SKIP ON TRANSFER REQUEST FLAG
6126      6261 1131      TAD TARGET
6127      6262 4443      XDROUT      / TRACK
6128      6263 1374      TAD (PI)
6129      6264 3002      DCA IPI
6130      /
6131      /WAIT FOR A PROGRAM INTERRUPT REQUEST
6132      /
6133      / ; BUT WHILE WAITING, DISPLAY IN THE MQ THE CODED INFORMATION
6134      /AS INDICATED BY ACCUMULATOR SWITCHES 9, 10, AND 11 AT RUN-TIME
6135      /
6136      /          0 = TARGET TRACK AND SECTOR
6137      /          1 = TEST PARAMETERS (SELECTED FROM SA 200)
6138      /          2 = " A " STATUS
6139      /          3 = " COMMAND " WORD TO RX01
6140      /
6141      6265 6001      XPI,      ION
6142      6266 1371      TAD (-40)
6143      6267 3116      DCA HANGER
6144      6270 4570      LAS
6145      6271 0370      AND (SW10+SW11)
6146      6272 1367      TAD (TAD I DISPLAY)
6147      6273 3274      DCA ,+1
6148      6274 1705      TAD I DISPLAY
6149      6275 7421      MQL
6150      6276 2151      ISZ H1
6151      6277 5270      JMP XPI+3
6152      6300 2116      ISZ HANGER
6153      6301 5270      JMP XPI+3
6154      6302 7200      CLA
6155      6303 1366      TAD (XPI)
6156      6304 5765*      JMP HUNGUP
6157      6305 0147      DISPLAY, XTARGET
6158      6306 0132      TESTP
6159      6307 0126      ASTATUS
6160      6310 0112      COMMAND
6161      /RETURN TO HERE IF ANY ERROR OCCURS (OF IF "DONE" FLAG OCCURS)
6162      /
6163      /IF A PARITY ERROR OCCURS THEN RETRY TO LOAD THE COMMAND
6164      /
6165      /IF NOT A PARITY ERROR THEN THIS MUST BE THAT "DONE" FLAG I MENTIONED
6166      /
6167      6311 5323      LCDBRETURN, JMP OTHERRORS      / "JMP OTHERRORS" IF RETURN IS OK
6168      6312 7000      XLCDRETURN, NOP
6169      6313 7305      CLL CIA IAC RAL
6170      6314 0126      AND ASTATUS
    
```

```

6171 6315 7650 SNA CLA
6172 6316 5323 JMP OTHERRORS
6173 /PARITY ERROR - RETRY
6174 /
6175 6317 2144 ISZ PRETRY
6176 6320 5243 JMP LCDBL
6177 6321 4502 SCOPE
6178 6322 5453 EXIT
6179 /
6180 /THESE ARE ALL OTHER ERRORS WHICH MAY OCCUR
6181 /
6182 6323 1364 OTHERRORS, TAD (JMP OTHERRORS)
6183 6324 3312 DCA XLCDBRETURN
6184 /
6185 6325 1373 TAD (LCDBRETURN)
6186 6326 7041 CIA
6187 6327 1775 TAD RETURN
6188 6330 1232 TAD XLCDB
6189 6331 3232 DCA XLCDB
6190 6332 5632 JMP I XLCDB
6191 6364 5323 /
6192 6365 3341 /
6193 6366 6265 /
6194 6367 1705 /
6195 6370 0003 /
6196 6371 7740 /
6197 6372 7000 /
6198 6373 6311 /
6199 6374 5405 /
6200 6375 5505 /
6201 6376 0100 /
6202 6377 4242 /
        PAGE
6203 /TRANSFER DATA REGISTER (FROM) THE RX01 CONTROL
6204 /
6205 6400 6400 XXDRIN, .
6206 6401 6752 K67X2A, 6752
6207 6402 5600 JMP I XXDRIN
6208 /TRANSFER DATA REGISTER (TO) THE RX01 CONTROL
6209 /
6210 6403 6403 XXDROUT, .
6211 6404 6752 K67X2B, 6752
6212 6405 7200 CLA
6213 6406 5603 JMP I XXDROUT
6214 /SKIP ON TRANSFER REQUEST
6215 /
6216 6407 6407 XSTR, .
6217 6410 6753 K67X3A, 6753
6218 6411 5607 JMP I XSTR
6219 6412 2207 ISZ XSTR
6220 6413 5607 JMP I XSTR
6221 /
6222 6414 6414 XSER, .
6223 6415 6754 K67X4A, 6754
6224 6416 5614 JMP I XSEP
    
```

```

6225 6417 2214 ISZ XSER
6226 6420 5614 JMP I XSER
6227 /SKIP ON RX01 DONE FLAG
6228 /
6229 6421 6421 XSDN, .
6230 6422 6755 K67X5A, 6755
6231 6423 5621 JMP I XSDN
6232 6424 2221 ISZ XSDN
6233 6425 5621 JMP I XSDN
6234 /ENABLE / DISABLE RX01 INTERRUPT ENABLE
6235 /
6236 / AC = 1 AT ENTRY TO ENABLE INTERRUPT
6237 /
6238 /FORM: (AC = 0, 0R 1); INTR
6239 /
6240 / (GOBIT) = 0, NO RX01 PI IS EXPECTED
6241 /
6242 / (GOBIT) = 1, AN RX01 PI IS EXPECTED
6243 /
6244 6426 6426 XINTR, .
6245 6427 3152 DCA GOBIT
6246 6430 1152 TAD GOBIT
6247 6431 6756 K67X6, 6756
6248 6432 7200 CLA
6249 6433 5626 JMP I XINTR
6250 /INITIALIZE (POWER CLEAR) THE RX01 SUBSYSTEM
6251 /
6252 6434 6434 XINIT, .
6253 6435 6757 K67X7A, 6757
6254 /THE LABEL " SDNSECOND " MUST RESIDE HERE BECAUSE OF REFERENCES WITHIN " ERROR "
6255 /
6256 6436 4471 WAIT
6257 6437 4446 SDNSECOND, SDN
6258 6440 5236 JMP .-2
6259 6441 3006 DCA 6 /WAIT POINTER
6260 6442 4445 SER
6261 6443 5245 JMP XXINIT
6262 /AN ERROR HAS OCCURED FROM THE "INIT"
6263 /
6264 / [HOME] WAS THE TARGET
6265 /
6266 6444 4575 FORCE
6267 6445 3123 XXINIT, DCA SSTART
6268 6446 5634 JMP I XINIT
6269 /
6270 /
6271 /
6272 6447 6447 XINITB, .
6273 6450 6757 K67X7B, 6757
6274 6451 5647 JMP I XINITB
6275 6452 3737 REMOVE, TEXT " _REMOVE DIAGNOSTIC DISKETTE"
        6453 2205
        6454 1517
        6455 2605
        6456 4004
    
```

	6457	1101		
	6460	0716		
	6461	1723		
	6462	2411		
	6463	0340		
	6464	0411		
	6465	2313		
	6466	0524		
	6467	2405		
	6470	0000		
6276	6471	3737	MIDENTIFICATION,TEXT	"__MAINDEC-08-DIRXA-C"
	6472	1501		
	6473	1116		
	6474	0405		
	6475	0355		
	6476	6070		
	6477	5504		
	6500	1122		
	6501	3001		
	6502	5503		
	6503	0000		
6277	6504	3737	MSELECT, TEXT	"__SELECT PARAMATERS (INCLUDING DEVICE CODE)"
	6505	2305		
	6506	1405		
	6507	0324		
	6510	4020		
	6511	0122		
	6512	0115		
	6513	0124		
	6514	0522		
	6515	2340		
	6516	5011		
	6517	1603		
	6520	1425		
	6521	0411		
	6522	1607		
	6523	4004		
	6524	0526		
	6525	1103		
	6526	0540		
	6527	0317		
	6530	0405		
	6531	5100		
6278	6532	3737	MTESTP, TEXT	"__TEST PARAMATERS: "
	6533	2405		
	6534	2324		
	6535	4020		
	6536	0122		
	6537	0115		
	6540	0124		
	6541	0522		
	6542	2372		
	6543	4000		
6279	6544	3737	MXEHEADER, TEXT	"__EPR FAT FAST EAC GOOD PASS"
	6545	0522		

	6546	2240		
	6547	4006		
	6550	0124		
	6551	4040		
	6552	0601		
	6553	2324		
	6554	4040		
	6555	4040		
	6556	4040		
	6557	0501		
	6560	0340		
	6561	4007		
	6562	1717		
	6563	0440		
	6564	2001		
	6565	2323		
	6566	0000		
6280	6567	3727	MX2HEADER, TEXT	"_WORD GOOD BAD"
	6570	1722		
	6571	0440		
	6572	0717		
	6573	1704		
	6574	4002		
	6575	0104		
	6576	0000		
6281	6577	5440	MCOMMA, TEXT	", "
	6600	0000		
6282	6601	3700	MCRLF, TEXT	"_"
6283	6602	3705	MEOT, TEXT	"_END OF TEST "
	6603	1604		
	6604	4017		
	6605	0640		
	6606	2405		
	6607	2324		
	6610	4000		
6284	6611	3704	MHUNGPC, TEXT	"_DEVICE TEST HUNG AT PC "
	6612	0526		
	6613	1103		
	6614	0540		
	6615	2405		
	6616	2324		
	6617	4010		
	6620	2516		
	6621	0740		
	6622	0124		
	6623	4020		
	6624	0340		
	6625	0000		
6285	6626	3737	MEHEADER, TEXT	"_FAT CMND XDR CODE RSTA START TARGET TEST PASS"
	6627	0601		
	6630	2440		
	6631	4003		
	6632	1516		
	6633	0440		
	6634	3004		

	6635	2240		
	6636	4003		
	6637	1704		
	6640	0540		
	6641	4022		
	6642	2324		
	6643	0140		
	6644	4023		
	6645	2401		
	6646	2224		
	6647	4040		
	6650	4040		
	6651	4024		
	6652	0122		
	6653	0705		
	6654	2440		
	6655	4040		
	6656	402		
	6657	0523		
	6660	2440		
	6661	2001		
	6662	2323		
	6663	0000		
6286	6664	3310	MHOME,	TEXT "[HOME]"
	6665	1715		
	6666	0535		
	6667	0000		
6287	6670	1116	MINIT,	TEXT "INIT"
	6671	1124		
	6672	0000		
6288	6673	3737	MOD,	TEXT "__OD = "
	6674	1704		
	6675	4075		
	6676	4000		
6289	6677	4011	MID,	TEXT " ID = "
	6700	0440		
	6701	7540		
	6702	0000		
6290	6703	4006	MFIRST,	TEXT " FIRST = "
	6704	1122		
	6705	2324		
	6706	4075		
	6707	4000		
6291	6710	4013	MLAST,	TEXT " LAST = "
	6711	0123		
	6712	2440		
	6713	7540		
	6714	0000		
6292	6715	3727	MWRITE,	TEXT "_WRITE-"
	6716	2211		
	6717	2405		
	6720	5500		
6293	6721	3722	MREAD,	TEXT "_READ-"
	6722	0501		
	6723	0455		

	6724	0000		
6294	6725	1617	MDNERROR,	TEXT "NO CPC BUT"
	6726	4003		
	6727	2203		
	6730	4002		
	6731	2524		
	6732	0000		
6295	6733	0322	MDWESERROP,	TEXT "CRC AND"
	6734	0340		
	6735	0116		
	6736	0400		
6296	6737	0322	MSNDEERROR,	TEXT "CRC BUT NO DATA ERROR"
	6740	0340		
	6741	0225		
	6742	2440		
	6743	1617		
	6744	4004		
	6745	0124		
	6746	0140		
	6747	0522		
	6750	2217		
	6751	2200		
6297	6752	4004	MDATAERROR,	TEXT " DATA ERROR_"
	6753	0124		
	6754	0140		
	6755	0522		
	6756	2217		
	6757	2237		
	6760	0000		
6298	6761	2717	MWORD,	TEXT "WORD"
	6762	2204		
	6763	0000		
6299	6764	0231	MBYTE,	TEXT "BYTE"
	6765	2405		
	6766	0000		
6300	6767	4007	MGB,	TEXT " GOOD BAD"
	6770	1717		
	6771	0440		
	6772	0201		
	6773	0400		
6301	6774	3723	MSUMCHECK,	TEXT "_SUM-CHECK IS "
	6775	2515		
	6776	5503		
	6777	1005		
	7000	0313		
	7001	4011		
	7002	2340		
	7003	0000		
6302	7004	1713	MOK,	TEXT "OK"
	7005	0000		
6303	7006	3724	MDESUMMARY,	TEXT "_TOTAL BAD = "
	7007	1724		
	7010	0114		
	7011	4002		
	7012	0104		

```

7013 4075
7014 4000
6304 7015 7200 MCOLON, TEXT ":"
6305 7016 5500 MDASH, TEXT "-"
6306 7017 3715 MEDDDIDNOT, TEXT "_MISSING DD MARK"
7020 1123
7021 2311
7022 1607
7023 4004
7024 0440
7025 1501
7026 2213
7027 0000
6307 7030 3725 MUDDID, TEXT "_UNEXPECTED DD MARK"
7031 1605
7032 3020
7033 0503
7034 2405
7035 0440
7036 0404
7037 4015
7040 0122
7041 1300
6308 7042 3725 MSDNUNEXPECTED, TEXT "_UNEXPECTED RX01 IRQ"
7043 1605
7044 3020
7045 0503
7046 2405
7047 0440
7050 2230
7051 0001
7052 4011
7053 2221
7054 0000
6309 7055 3715 HNOSEP, TEXT "_MISSING FRPDR FLAG"
7056 1123
7057 2311
7060 1607
7061 4005
7062 2222
7063 1722
7064 4006
7065 1401
7066 0700
6310 7067 3737 MUNKNOWN, TEXT "__UNKNOWN IRQ"
7070 2516
7071 1316
7072 1727
7073 1640
7074 1122
7075 2100
6311 7076 3704 MDEV, TEXT "_DEVICE CODE TO BE USED "
7077 0526
7100 1103
7101 0540

```

```

7102 0317
7103 0405
7104 4024
7105 1740
7106 0205
7107 4025
7110 2305
7111 0440
7112 4000
6312 /THE FOLLOWING IS THE WRITE BUFFER ALLOCATED STORAGE
6313 /
6314 7113 WBUFFER=
6315 7313 RBUFFER=WBUFFER+200
6316 7513 *RBUFFER+200
6317
6318
6319
6320
6321 0200 *200 /AUTO START BINARY
6322
6323 $$$

```


A10	0010	C0RETD	0614	DOSET	0256	E51	1110
A11	0011	C0RETR	0541	DRVZRO	2753	E52	1117
A12	0012	C0SETD	0613	DTESTP	0114	E53	1105
A13	0013	C0SETS	0540	DTYPE	5523	E54	1114
A14	0014	C0STRT	0200	DWESER	3677	E55	1123
ACL	7701	C0SWIT	4426	DWS	3506	E56	1076
ACSAVE	1314	C0SWST	0745	DWSLOG	0140	E60	1137
ACTIVE	0752	C0TEST	0343	E0	0620	E61	1147
AERROR	4467	C0TMP1	1030	E0PRE	0410	E62	1152
ALT12	1400	CAF	6007	E1	0624	E63	1155
ALT12L	1402	CCNT	5066	E10	0660	E70	1213
ANDRET	4107	CHAR	1104	E100	1227	E7000	2405
APT8	4470	CHARLI	5320	E11	0643	E7001	2416
AROUND	0230	CHECKC	4424	E110	1253	E7002	2430
ASTATU	0126	CHEK22	4472	E120	1310	E7003	2444
BELL	0207	CHNDEV	5067	F121	1314	E7004	2455
BLANK	0166	CKCOUT	0236	E122	1320	E7005	2505
BSTATU	0127	CKSWIT	4570	E123	1325	E7006	2467
BSW	4435	CLKCNT	4145	E124	1276	EA120	1411
BSWAC	2635	CNOTFI	3717	E130	1512	EA121	1415
BSWLIN	2636	CNTRIC	0476	E131	1525	EA122	1437
BSWRAL	2634	CNTRLD	0600	E140	1661	EA123	1450
BUSY	0111	CNTRLE	0542	E1PRE	0443	EAC	0165
BYRETR	0511	CNTRLQ	0503	E2	0630	EB	1622
C0BY1	0233	CNTRLR	0514	E20	0666	EBCOMP	1630
C0BY2	1261	CNTRLS	0524	E21	0671	EBLOOP	1635
C0BY3	1070	CNTVAL	0257	E210	4557	EBOK	1663
C0BY4	0520	COMMAN	0112	E211	2320	ECOMMA	0150
C0BY5	1127	COMP	2756	E212	2332	EERROR	5646
C0CK22	1330	COMPAR	3601	E22	0674	EMPTY	3617
C0CKP	1031	COMPRE	0113	E23	0677	EMPTYE	4005
C0CKSW	4430	COUNT	4146	E24	0702	EMPTYL	3607
C0CNTR	4571	CRCERR	3454	E240	2030	EMPTYO	4016
C0D01	0324	CSTATU	0130	E241	2067	ENDCOM	4000
C0D010	1245	DASHAL	5725	E242	2100	ENDIT	0742
C0D011	0607	DASHBC	5731	E245	2033	EPSCSO	1362
C0D02	1042	DCACTI	0743	E25	0705	ERETUR	5506
C0D03	1327	DCAD0A	2525	E26	0715	ERR1	1000
C0D04	1015	DCATAR	4332	E27	0723	ERRMES	1263
C0D07	0532	DDERRO	5525	E270	2134	ERROR	4452
C0EXT1	0234	DDIGNO	5463	E271	2137	ERRORS	1363
C0EXT2	0305	DIGITS	5063	E272	2150	EXIT	5453
C0FILL	1046	DISPLA	6305	E28	0731	EXITA	0440
C0GET	0624	DMTYPE	6060	E2PRE	0413	EXITCK	4534
C0GET1	0302	DNS	3506	E3	0634	EXTLOO	1405
C0HANG	1133	DNSLOG	0141	E30	0755	FB	1471
C0INQU	4427	DOA	2532	E3PRE	0423	FB120B	4541
C0LOOK	4405	DOB	2533	E40	1044	FBEB	1476
C0PASS	4425	DOCNT	0254	E41	1056	FILCNT	1047
C0RET2	0600	DONE	4454	E42	1021	FILL	3221
C0RET3	4125	DONEA	0426	E4PRE	0446	FILLER	3226
C0RET4	3317	DOPACK	0210	E50	1101	FILLOK	3237

FIRST	0033	K67X2B	6404	MINIT	6670	PAT6	3140
FIRSTE	0115	K67X3A	6410	MLAST	6710	PATSUM	0153
FIRSTT	0451	K67X3B	0720	MNOSER	7055	PATTER	3115
FLSAVE	1316	K67X4A	6415	MOD	6673	PCLF	6662
FORCE	4575	K67X4B	0726	MOK	7004	PCSAVE	1313
GENTES	3131	K67X5A	6422	MORETE	0450	PCSCOP	1364
GETAPA	4455	K67X5B	0712	MOA	7501	PHOME	6032
GETASE	4456	K67X6	6431	NOL	7421	PI	5405
GETATR	4457	K67X7A	6435	MQSAYE	1315	PIEXIT	5476
GETCHI	0703	K67X7B	6450	MREAD	6721	PISDN	5422
GETDAT	0456	K7000	0172	MSB	0157	PNTBUF	1131
GETUNI	4460	K7377	1562	MSDNUN	7042	PNTBAT	5340
GOBIT	0152	K7777	0117	MSELEC	6504	PNTID	1530
GOITA	0443	KRETRY	0121	MSNDER	6737	POLL	4237
GOOD	0164	LAS	4570	MSUMCH	6774	PRETES	0400
GOTOA	0454	LASSW0	6111	MUDDDI	7030	PRETRY	0144
GTF	6004	LASSW3	5642	MUNKNO	7067	PRINT	4473
H1	0151	LAST	0034	MWORD	6761	PSIE	6665
HALT	4461	LCD	4437	MWRITE	6715	PSKE	6663
HANGEP	0116	LCDA	4440	MX	0556	PSKF	6661
HLT	4461	Lcdb	4441	MX2HEA	6567	PSIB	6664
HLT16	6113	LCDBL	6243	MXEHEA	6544	PTSTOR	0352
HLT6	4075	LCDBRE	6311	NEXT	4210	Q46ID	4505
HLT7	3505	LOCKUP	4464	NEXTAC	0757	Q460D	4513
HLTN0P	1563	LSB	0156	NEXTSE	4623	Q40D	4454
HUNGPC	3353	MBYTE	6764	NOHEAD	5655	Q60D	4500
HUNGUP	3341	MCOLON	7015	NOMORE	0522	QUIET	3335
ID	0032	MCOMMA	6577	NOPRIN	6061	R1	4724
IF	4706	MCRFL	6601	NORX01	0427	R1RETR	0136
INDEXA	0455	MDASH	7016	NOSCOPI	1336	R2	4725
INIT	4450	MDATEE	6752	NOSER	3322	R2RETR	0137
INITB	4451	MDESUM	7006	NOSET	0246	RANGEN	4710
INITSE	4462	MDEV	7076	NOTCLB	4537	RBUFFE	7313
INITSW	6115	MDSER	6725	NOTEST	5453	RDC	0120
INITTR	4463	MDETEST	6532	NOTO	0335	RDRWR	3550
INMODE	1105	MDEESE	6733	NTCLAS	1252	READ	4474
INSUMC	3761	MEDDDI	7017	NULINE	5302	READCO	4475
INTR	4447	MEHEAD	6626	NUREAD	3517	READER	3437
IOF	6002	MEOT	6602	NUWORD	5253	READL	3420
ION	6001	MESA	1004	OD	0031	READOK	3527
IPI	0002	MESAC	1276	OK	4465	READR	3423
IRDWR	2000	MESFL	1304	OKSTAR	4641	REASK1	1407
ISZCOM	3756	MESMQ	1301	ONECRL	5671	REBEG1	0545
ISZDIG	5042	MESPAS	0260	OTHERR	6323	RECALI	6073
JMPDIG	5032	MESPC	1273	OUTPUT	4263	RED01	0661
JMPICO	3600	MESSAG	5262	PACKDO	4531	REDOA	0415
JMPWHI	3263	MFIRST	6703	PASCNT	0255	REFILL	3214
K0007	0173	MGB	6767	PASS	0161	REMOVE	6452
K6500	4144	MHOME	6664	PAT2	3125	REREAD	3430
K6520	4361	MHUNGP	6611	PAT3	3126	RESEK	3432
K67X1	6203	MID	6677	PAT4	3127	RESEQU	4274
K67X2A	6401	MIDENT	6471	PAT5	3130	RETURN	5505

REWRIT	3206	T11	1232	THENEX	4110	XC8CRL	1032
RLOGGE	3545	T12	1261	THETES	3000	XC8ECH	1072
ROK	3541	T13	1455	TICK	4466	XC8ERR	1200
RST	4476	T14	1603	TPCNT	0746	XC8INQ	0635
RSTB	4477	T15	1601	TPACKS	0145	XC8LOO	1400
RXERRO	5600	T16	1454	TSTCHA	0721	XC8OCT	1007
RXHERE	0163	T17	1602	TSTUNT	2744	XC8PAS	0200
SAVERS	5613	T2	0663	TTRACK	0146	XC8PAU	1317
SAVECS	5626	T20	1600	TTYBUS	0160	XC8PNT	0317
SCOPE	4502	T20STR	2323	TTYLPT	1132	XC8PSW	0651
SCOPIN	1355	T20XDR	2310	THOCRL	5667	XC8SW	0267
SDN	4446	T21	1675	TX	2303	XC8TTY	0306
SDNSEC	6437	T22	1674	TY1ASC	5321	XC8TYP	1106
SDNUNE	3320	T23	2265	TY4OCT	4505	XC8HECK	4516
SECTOR	0122	T24	1714	TY8OCT	4436	XCHK22	2541
SEQ	4276	T25	1716	TYASTA	5740	XCKSWI	3400
SEQ000	4306	T26	1720	TYCSTA	6000	XCNT	0174
SEQ001	4317	T27	2106	TYPEIT	4506	XCOMPA	3662
SEQ010	4400	T3	0734	UNIT	4242	XCR CER	3503
SEQ100	4444	T30	2200	UNITS	4235	XD	0400
SEQ111	4330	T31	2202	UNITX	4241	XDEVIC	5125
SEQ3	4422	T32	2223	UNITZ	4240	XDOLPT	1121
SEQ6	4465	T33	2241	UNKNOW	5473	XDONE	4243
SER	4445	T34	2300	UPAROW	0615	XDOSW	0523
SETUP	4500	T35	2276	UREAD	3522	XDRIN	4442
SHIFT	5015	T36	2277	VERIFY	5466	XDROUT	4443
SHIFTS	5065	T37	2275	WAIT	4471	XEMPTY	0066
SND	3506	T4	1000	WAITY	4507	XERROR	2637
SNDLOC	0142	T4H	1004	WATHES	1307	XFLENG	3714
SPLCIA	4501	T5	1065	WBUFPE	7113	XFORCE	5516
SRETRY	0143	T6	1126	WHICHR	3267	XGETAP	3035
SSIART	0123	T7	1200	WNOK	3264	XGETAS	4607
STARGE	0124	T7OK	1214	WORDX	0154	XGETAT	4261
START	0125	TAB	4504	WORDY	0155	XGETUN	4200
STOPNT	0353	TAB12	5712	WRESEE	3212	XHALT	4115
STR	4444	TAB31	6002	WRITE	4510	XHUNG	3344
SUBSCO	4503	TAB43	6022	WRITEL	3204	XI	1100
SW0	4000	TABLA	0461	WRITEO	3257	XINIT	6434
SW1	2000	TABLB	0470	WRITER	3246	XINITB	6447
SW10	0002	TADLAS	3273	WUNITS	4236	XINIT5	4600
SW11	0001	TARGET	0131	XAI0	0133	XINIT7	4250
SW2	1000	TEST	0167	XAI1	0134	XINTR	6426
SW3	0400	TEST1	2520	XAC	5503	XK67X2	5126
SW4	0200	TEST2	2516	XAERRO	4346	XK67X3	5127
SW5	0100	TEST3	2512	XANDRE	4114	XK67X4	5130
SW6	0040	TEST4	2510	XAPT6	1546	XK67X5	5131
SW7	0020	TEST5	2522	XASTAT	3760	XK67X7	5133
SW8	0610	TLSTP	0132	XBSW	2612	XKCC	0314
SW9	0004	TESTS	0460	XC	0300	XLCD	6200
T0	0611	TESTX	2526	XC8CHA	0351	XLCD A	6205
T1	0637	TESTXL	2531	XC8CKP	1050	XLCD B	6232
T10	1216	THEL	3003	XC8CNT	0400	XLCD BR	6312

XLINK	5504	XXFORC	5513
XLOCKU	2600	XXGETA	4610
XMESSA	3334	XXINIT	6445
XXM	0253	XXTAB	5243
XNOPRI	2727	XXTHEL	3024
XNULIN	5311	XXX	0135
XOCTAL	5062	XYTHEL	3026
XOK	6123	ZERO	5057
XOUTPU	5263		
XPAT6	3145		
XPATTE	3103		
XPI	6265		
XPRINT	5244		
XRDC	3416		
XREAD	3414		
XREADC	3410		
XRETUR	5507		
XRST	2437		
XRSTB	2400		
XSCOPE	1332		
XSDN	6421		
XSEQ	4333		
XSEQ2	4413		
XSEQ3	4433		
XSER	6414		
XSETUP	4077		
XSHIFT	5020		
XSPECI	5351		
XSSCOP	1342		
XSTARG	4707		
XSTR	6407		
XTAB	5215		
XTABL	5221		
XTABLA	0457		
XTABLB	0460		
XTARGE	0147		
XTCF	5343		
XTHEL	3016		
XTICK	4127		
XTY4OC	5000		
XTY8OC	5200		
XTYPEI	5305		
XWAIT	1722		
XWAITT	5400		
XWRITE	3200		
XXC8CN	1340		
XXC8IN	1345		
XXC8PS	1351		
XXC8SW	1355		
XXDMP	5064		
XXDRIN	6400		
XXDROU	6403		

IPI 1296 1302# 2412 2488 2527 2571 2604 5549 6085 6129
 IRDWR 3095 3097 3099 3164#
 ISZCOM 4562 4628 4654#
 ISZDIG 5345 5353# 5363
 JMPDIG 5345# 5364
 JMPICO 4153# 4762 4766
 JMPWHI 4078 4148#
 K0007 1579# 5343
 K6500 4816 4819#
 K6520 5003 5006#
 K67X1 5414 6060#
 K67X2A 5415 6206#
 K67X2B 5402 6211#
 K67XJA 5416 6217#
 K67XJB 2170# 5404
 K67X4A 5417 6223#
 K67X4B 2176# 5406
 K67X5A 2214 5418 6230#
 K67X5B 2164# 5418
 K67X6 5419 6247#
 K67X7A 5420 6253#
 K67X7B 5410 6273#
 K7000 1578# 2927 5347
 K7377 2925 2931#
 K7777 1500# 4959 5330
 KPENTRY 1502# 4088 4090 4120 4270 4281 4291 4293 4371 4482
 LAS 1301# 1619 1953 2709 3739 3745 3788 3840 3847 3854 3941 4111 4134 4199
 420R 4311 4354 4367 4378 4558 4621 4695 4721 4763 5798 5973 6011 6025
 6144
 LASSWJ 6014 6018 6021 6025#
 LASSW3 5798#
 LAST 1373# 1164 4168 4169 4177 5181 5243
 LCD 1389# 2027 2083 2091 2127 2213 2277 2288 2369 2370 2414 2445 2486 2530
 2565 2566 2616 2617 2635 2838 2970 3171 3265 3530 3581 3621 5157 5774
 5786 6079 6108
 LCDA 1391# 4096 4487
 LCDB 1393# 4125 4296
 LCDBL 6107# 6176
 LCDBRE 6114 6167# 6185
 LOCKUP 1434# 1874 2660 3089
 LSB 1560# 3811 3813 3817 3819 4633 4635 4647 4649
 MBYTE 4612 6299#
 MCOLON 6304#
 MCOMMA 6281#
 MCRLF 3799 4630 5839 5841 6282#
 MDASH 5882 5886 5890 6305#
 MDATA# 4603 6297#
 MDESUM 4749 6303#
 MDEV 6311#
 MDNSER 4598 6294#
 MDTEST 1630 6278#
 MDWESE 4601 6295#
 MEDDDT 5729 6306#

MEHEAD 5817 6285#
 MEOT 6283#
 MESA 563 643#
 MESAC 877 914#
 MESFL 885 916#
 MESMQ 981 915#
 MESPAS 86 124#
 MESPC 873 913#
 MESSAG 5493 5494 5496 5497 5501#
 MFIRST 1682 6290#
 MGB 4617 6300#
 MHOME 5928 5947 6286#
 MHUNGP 4222 6284#
 MID 1667 6289#
 MIDENT 2899 6276#
 MINIT 5855 6287#
 MLAST 4175 6291#
 MNOSER 4197 6309#
 MOD 1651 6288#
 MOK 4740 6302#
 NOPETE 1416 1874#
 MOA 54# 279 733 855 1382#
 MOG 51# 488 1383# 6149
 MOSAVE 281 487 735 856 882 920#
 MREAD 4364 4584 6293#
 MSB 1561#
 MSDNUN 4195 6308#
 MSELECT 2906 6277#
 MSNDER 4366 6296#
 MSUMCH 4735 6301#
 MUDDDI 5730 6307#
 MUNKNO 5691 6310#
 MWORD 4606 4615 6298#
 MWRITE 4361 4581 6292#
 MX 119 1930 1939 1948 1967#
 MXZHEA 6280#
 MXZHEA 3797 6279#
 NEXT 4841# 4851
 NEXTAC 2225 2227 2231#
 NEXTSE 5217# 5247 5257
 NOHEAD 5815 5826#
 NOMORE 1917 1930# 2670 3091
 NOPRIN 5801 5972#
 NORX01 1805 1839# 3873 3876
 NOSCOP 2683# 2712
 NOSER 4197# 5685
 NOSET 107 113#
 NOTCL8 5131 5143#
 NOTEST 1436# 2008 2356 2432 2550 3396 3400 3403 4227 4935
 NOTQ 1705 1724#
 NTCLAS 863 896#
 NULINE 5511 5518#
 NUREAD 4159 4387# 4395

UREAD 4375 4394#
 VERIFY 5673 5682#
 WAIT 1445# 1806 5775 5787 6256
 WAITTY 1473# 2410 2484 2524 2562 4768 4778 4789 5134 5490
 WAITES 520 917#
 WBUFFE 2846 2848 2994 3024 3035 3954 3967 4094 4484 4631 5272 5279 5283 5284
 6314# 6315
 WHICHR 4077 4148 4157#
 WNOTOK 4133 4139 4150#
 WORDX 1558# 3968 3970# 5285 5286 5287
 WORDY 1559# 3971 5289
 WRESEE 4090# 4137
 WRITE 1475# 3468 3699 3706 3912 4443
 WRITEL 4075# 4151
 WRITEO 4126 4144#
 WRITER 4131#
 WUNITS 1800 3165 3347 3435 4835 4836 4847 4856 4857 4864# 4878
 XA10 1512# 2849 2881 2981 2995 3000 3007 3025 3034 4509 4516 4527 4653
 XA11 1513# 2997 3001 3013 3027 4485 4523 4632 4644 4684
 XAC 5595 5696 5699#
 XAERKO 1442 4994# 4998
 XANDRE 4772 4782 4783 4785 4786#
 XAPTB 1444 2919# 2923
 XASTAT 4481 4542 4594 4662#
 XBSW 1378 3755# 3772
 XC 3081# 3090
 XC8CHA 1699 1700 1740#
 XC8CKP 142 727# 748 750 819
 XC8CNT 270# 275 303 305 308 310 358 392 401 426 456 525 745 770
 965 1014
 XC8CKL 83 95 301 477 518 561 611 641 685# 696 868 888
 XC8ECH 571 579 764# 771 775
 XC8ERR 850# 858 901 911
 XC8INO 99 513# 516 527 528 894 976 1019
 XC8LOO 1012# 1018 1310
 XC8OCT 88 567 655# 671 875 879 883 887
 XC8PAS 71# 100 102 1342
 XC8PAU 939# 943 944 946 947
 XC8PNT 85 93 200# 202 204 208 219 519 562 870 872 876 880 884
 XC8PSW 459 552# 555 613 984
 XC8SW 74 96 139# 157 889 896 922
 XC8TTY 170# 178 397 521 742 765 1013
 XC8TYP 215 224 298 300 474 476 569 640 664 688 693 772 791# 811
 XC8ECK 1339 5126# 5132 5140 5142 5143
 XCHK22 1448 3716# 3720 3721 3722
 XCKSWI 1575 4245# 4252
 XCNT 1580# 5355 5361 5445
 XCOMP8 4578# 5770 5877
 XCPCEP 4343 4357 4367#
 XD 1928# 1929
 XDEVIC 5391 5414#
 XDOLPT 795 802#
 XDONE 1418 1877# 4880 4881

SEQ 0170

XDOSW 375 377#
 XDFLN 1395# 1840 2092 2304 2327 2371 2650 2777 2987 3214 3532 3588 3628 4508
 5635 5781 5793
 XDROUT 1397# 2853 3175 3270 4104 5160 6123 6127
 XFEMPT 4694 4700 4724 4732 4747 4754 4758#
 XFRROP 1414 3776# 3781 3810 3860
 XFLFNG 4607 4613 4615#
 XFORCE 5724# 5983
 XGETAP 1420 3931# 3945 3950 3996 4052
 XGETAS 1422 5192# 5290
 XGETAT 1424 3902 4906# 4988
 XGETUN 1426 4833# 4962
 XHALI 1429 4788# 4798 4801
 XHUNG 4221# 5766 5865
 XI 2664# 2669
 XINIT 1408 6252# 6268
 XINITB 1410 6272# 6274
 XINITs 1431 5180# 5186
 XINIIT 1433 4885# 4893
 XINTH 1406 6244# 6249
 XK67X2 5401 5415#
 XK67X3 5403 5416#
 XK67X4 5405 5417#
 XK67X5 5407 5418#
 XK67X7 5409 5420#
 XKCC 1695# 5600
 XLCD 1390 6057# 6061
 XLCD8 1392 6067# 6080 6082 6083 6089
 XLCD8 1394 6095# 6188 6189 6190
 XLCD8R 6117 6168# 6183
 XLINK 5597 5694 5700#
 XLOCKU 1435 3738# 3749
 XMESSA 4198 4204 4207#
 XMA 90 119#
 XNOPRI 3791 3847#
 XNULIN 5520 5525#
 XOCTAL 5326 5327 5328 5334 5341 5342 5369#
 XOK 1438 6039# 6041 6042
 XOUTPU 5503# 5517 5518
 XPAT6 4043# 4051
 XPATTE 3979 3980# 3995
 XPI 6141# 6151 6153 6155
 XPPINT 1450 4770 5487# 5491 5492 5551 5553
 XRDC 4258 4276#
 XREAD 1452 3464 4257 4274# 4431 4440
 XREADC 1454 4255# 4256
 XRETUR 5686 5712#
 XRST 1456 3619# 3673
 XRSTB 1458 3579# 3609
 XSCOPE 1464 2673# 2684 2686 2691 2694 2702 2706
 XSDN 1404 6229# 6231 6232 6233
 XSEQ 4974# 5053 5120
 XSEQ2 5016 5026#

SEQ 0171

.L1177	639	838#	2276	2368	2413	2460#				
.L1364	984	999#								
.L1365	955	1000#								
.L1366	945	1001#								
.L1367	905	908	1002#							
.L1370	902	1003#	2670	2717#						
.L1371	894	976	1004#	2669	2718#					
.L1372	889	896	992	1005#	2603	2719#				
.L1373	875	879	883	887	1006#	2594	2720#			
.L1374	870	872	876	880	884	1007#	2570	2721#		
.L1375	868	888	1008#	2526	2722#					
.L1376	864	1009#	2487	2723#						
.L1377	861	965	1010#	2485	2529	2564	2615	2710	2724#	
.L1570	2930	2933#								
.L1571	2848	2934#								
.L1572	2846	2935#								
.L1573	2843	2936#								
.L1574	2834	2839	2874	2876	2937#					
.L1575	1019	1023#	2780	2938#						
.L1576	1014	1024#	2761	2939#						
.L1577	1013	1025#	2759	2940#						
.L1751	3139	3141#								
.L1752	3137	3142#								
.L1753	3135	3143#								
.L1754	3098	3144#								
.L1755	3095	3097	3099	3145#						
.L1756	3091	3146#								
.L1757	3090	3147#								
.L1760	3088	3148#								
.L1761	3086	3149#								
.L1762	3078	3150#								
.L1763	3075	3151#								
.L1764	3062	3152#								
.L1765	3061	3153#								
.L1766	3035	3154#								
.L1767	3024	3155#								
.L1770	3009	3055	3156#							
.L1771	2994	3157#								
.L1772	2984	3158#								
.L1773	2980	3159#								
.L1774	2978	2996	3160#							
.L1775	2975	3161#								
.L1776	2971	3003	3030	3032	3162#					
.L1777	2965	3163#								
.L2162	3298	3311#								
.L2163	3295	3312#								
.L2164	3262	3313#								
.L2165	3254	3308	3314#							
.L2166	3252	3306	3315#							
.L2167	3243	3316#								
.L2170	3240	3317#								
.L2171	3225	3318#								
.L2172	3222	3319#								

.L2173	3212	3320#								
.L2174	3210	3321#								
.L2175	3206	3322#								
.L2176	3168	3323#								
.L2177	3165	3324#								
.L2356	3529	3557#								
.L2357	3522	3558#								
.L2360	3520	3559#								
.L2361	3512	3516	3560#							
.L2362	3483	3561#								
.L2363	3480	3562#								
.L2364	3465	3563#								
.L2365	3464	3564#								
.L2366	3463	3565#								
.L2367	3417	3436	3566#							
.L2370	3401	3567#								
.L2371	3378	3422	3475	3568#						
.L2372	3372	3569#								
.L2373	3371	3570#								
.L2374	3370	3571#								
.L2375	3350	3572#								
.L2376	3347	3435	3573#							
.L2377	3332	3574#								
.L2563	3718	3723#								
.L2564	3699	3724#								
.L2565	3689	3693	3725#							
.L2566	3687	3695	3726#							
.L2567	3663	3727#								
.L2570	3660	3728#								
.L2571	3649	3729#								
.L2572	3646	3730#								
.L2573	3620	3731#								
.L2574	3606	3732#								
.L2575	3597	3733#								
.L2576	3596	3600	3645	3691	3697	3734#				
.L2577	3580	3735#								
.L2763	3876	3881#								
.L2764	3874	3882#								
.L2765	3873	3883#								
.L2766	3870	3884#								
.L2767	3867	3885#								
.L2770	3866	3886#								
.L2771	3848	3887#								
.L2772	3789	3888#								
.L2773	3785	3889#								
.L2774	3783	3890#								
.L2775	3759	3891#								
.L2776	3746	3892#								
.L2777	3740	3893#								
.L3163	3980	3999	4053#							
.L3164	3976	4054#								
.L3165	3972	4055#								
.L3166	3967	4056#								

.L3167	3961	4057#				
.L3170	3960	4058#				
.L3171	3954	4059#				
.L3172	3946	4060#				
.L3173	3943	4061#				
.L3174	3942	4045	4062#			
.L3175	3908	3911	4063#			
.L3176	3902	4064#				
.L3177	3901	4065#				
.L3365	4225	4230#				
.L3366	4211	4231#				
.L3367	4197	4232#				
.L3370	4195	4233#				
.L3371	4190	4200	4234#			
.L3372	4167	4235#				
.L3373	4160	4236#				
.L3374	4159	4237#				
.L3375	4112	4135	4238#			
.L3376	4094	4239#				
.L3377	4077	4240#				
.L3572	4443	4447#				
.L3573	4397	4448#				
.L3574	4347	4422	4449#			
.L3575	4344	4419	4450#			
.L3576	4312	4379	4451#			
.L3577	4248	4355	4452#			
.L3766	4631	4669#				
.L3767	4612	4670#				
.L3770	4606	4671#				
.L3771	4578	4672#				
.L3772	4559	4673#				
.L3773	4529	4655	4674#			
.L3774	4525	4646	4675#			
.L3775	4489	4676#				
.L3776	4488	4677#				
.L3777	4484	4678#				
.L4167	4770	4822#				
.L4170	4769	4823#				
.L4171	4762	4766	4824#			
.L4172	4722	4825#				
.L4173	4719	4826#				
.L4174	4717	4720	4730	4736	4760	4827#
.L4175	4698	4828#				
.L4176	4696	4829#				
.L4177	4686	4830#				
.L4174	4964	5007#				
.L4175	4963	5008#				
.L4176	4913	5009#				
.L4177	4909	5010#				
.L4572	5156	5171#				
.L4573	5129	5172#				
.L4574	5053	5120	5173#			
.L4575	5032	5171#				

.L4576	5031	5175#				
.L4577	5019	5176#				
.L4773	5279	5284	5311#			
.L4774	5273	5312#				
.L4775	5272	5283	5313#			
.L4776	5228	5314#				
.L4777	5223	5315#				
.L5165	5410	5423#				
.L5166	5408	5424#				
.L5167	5406	5425#				
.L5170	5404	5426#				
.L5171	5402	5427#				
.L5172	5391	5428#				
.L5173	5385	5429#				
.L5174	5384	5430#				
.L5175	5365	5431#				
.L5176	5363	5432#				
.L5177	5350	5366	5433#			
.L5364	5558	5572#				
.L5365	5548	5573#				
.L5366	5529	5574#				
.L5367	5527	5575#				
.L5370	5525	5576#				
.L5371	5512	5577#				
.L5372	5509	5578#				
.L5373	5508	5546	5579#			
.L5374	5504	5580#				
.L5375	5476	5571	5581#			
.L5376	5472	5582#				
.L5377	5464	5583#				
.L5566	5738	5746#				
.L5567	5685	5747#				
.L5570	5683	5748#				
.L5571	5657	5749#				
.L5572	5649	5750#				
.L5573	5641	5674	5723	5745	5751#	
.L5574	5636	5752#				
.L5575	5633	5753#				
.L5576	5609	5754#				
.L5577	5608	5755#				
.L5762	5900	5901#				
.L5763	5891	5902#				
.L5764	5846	5903#				
.L5765	5835	5851	5904#			
.L5766	5801	5905#				
.L5767	5799	5906#				
.L5770	5785	5907#				
.L5771	5784	5908#				
.L5772	5782	5794	5909#			
.L5773	5773	5910#				
.L5774	5770	5877	5911#			
.L5775	5766	5865	5912#			
.L5776	5765	5769	5826	5834	5850	5864 5876 5913#

Table of numerical data with multiple columns. Row 1: .L5777 5761 5914#. Row 100: .V0100 222 235#. Row 1000: .V1214 2407 2723#.

Table of numerical data with multiple columns. Row 1: .V0212 692 832#. Row 100: .V0400 75 97. Row 1000: .V1214 2407 2723#.

