

IDENTIFICATION

PRODUCT CODE: AC-C656D-MA
PRODUCT NAME: AJRLADO RL8-A/RL01 DSKLS CTRLR
PRODUCT DATE: SEPT, 1981
MAINTAINER: DIAGNOSTIC ENGINEERING GROUP
AUTHOR: JACK RICH

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

- S.1.1 PROGRAM PURPOSE (ABSTRACT)
 - S.1.1.1 REVISION HISTORY
 - S.1.2 SYSTEM REQUIREMENTS
 - S.1.2.1 HARDWARE REQUIREMENTS
 - S.1.2.2 SOFTWARE REQUIREMENTS
 - S.1.3 RELATED DOCUMENTS AND STANDARDS
 - S.1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
 - S.1.5 ASSUMPTIONS
- S.2.0 OPERATING INSTRUCTIONS
 - S.2.1 LOADING AND STARTING PROCEDURES
 - S.2.2 SPECIAL ENVIRONMENTS
 - S.2.2.1 MULTIPLE OPTION TESTER
 - S.2.2.2 APT/8A
 - S.2.3 PROGRAM OPTIONS
 - S.2.3.1 OPERATOR PROMPTS
 - S.2.3.2 PHASE LOCKED LOOP ADJUSTMENT ROUTINE
 - S.2.3.3 MEANING OF SWITCH REGISTER BITS
 - S.2.3.4 CONSOLE PACKAGE COMMAND SUMMARY
 - S.2.4 EXECUTION TIMES
- S.3.0 ERROR INFORMATION
 - S.3.1 ERROR REPORTING PROCEDURES
 - S.3.2 ERROR HALTS
- S.4.0 PROGRESS REPORTS
- S.5.0 DEVICE INFORMATION TABLES
 - S.5.1 INSTRUCTION SET
 - S.5.2 CONTROLLER REGISTERS
- S.6.0 SUBTEST SUMMARIES
- S.7.0 PROGRAM, SYMBOL TABLE, AND CROSS REFERENCE LISTING

8.1.1 PROGRAM PURPOSE (ABSTRACT)

THIS PROGRAM TESTS AND EXERCISES AN RL8A CONTROLLER WITHOUT REQUIRING THE USE OF A DISK DRIVE. DRIVES MAY BE CONNECTED BUT MUST BE EITHER POWERED OFF OR IN THE LOAD STATE. ALL POSSIBLE REGISTERS AND CONTROLLER FUNCTIONS THAT CAN BE TESTED WITHOUT A DRIVE ARE TESTED. THOSE PORTIONS OF THE LOGIC THAT CANNOT BE TESTED, ALONG WITH THEIR ESTIMATED PERCENTAGE OF THE TOTAL LOGIC, ARE:

DRIVE INTERFACE LOGIC	(5%)
WRITE GATE LOGIC	(.1%)
HEADER FOUND LOGIC	(2%)
BAD CRC DETECTION	(.8%)
DRIVE READY LOGIC	(1.2%)
COUNT FUNCTION OF SA REG	(.2%)

ALL OF THE ABOVE FAILURES CAN BE DETECTED BY THE RL8A/RL01 AND RL8A/RL02 DRIVE FUNCTIONAL TESTS.

8.1.1.1 REVISION HISTORY

VERSION D CONTAINS DOCUMENTATION CHANGES TO INCLUDE THE RL278. THE PROGRAM CHANGES WERE TO THE SOURCE FILE, WHICH WAS UPDATED TO SUPPORT VT278.

8.1.2 SYSTEM REQUIREMENTS

8.1.2.1 HARDWARE REQUIREMENTS

PDP8/E,F,M,OR A WITH AT LEAST 8K MEMORY
 VT278 WITH 16-32K OF MEMORY.
 CONSOLE DEVICE (ASR 33 OR EQUIVALENT)
 RL8A CONTROLLER
 RL278 CONTROLLER.
 DRIVE 3 (OR 7) MUST BE POWERED OFF OR UNLOADED, IF CONNECTED AND READY THE DIAGNOSTIC WILL REPORT ERROR CONDITIONS.
 IF THE DRIVE IS UNLOADED THE FAULT INDICATOR WILL BE TURNED ON AND OFF, THIS IS A NORMAL CONDITION FOR UNIT 3 OR 7.

8.1.2.2 SOFTWARE REQUIREMENTS

THIS PROGRAM IS FULLY COMPATIBLE WITH OS/8 AND APT8A.

8.1.3 RELATED DOCUMENTS AND STANDARDS

THIS PROGRAM ADHERES TO THE DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS DOCUMENT NO. 175-003-009-02.

8.1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THIS PROGRAM IS NORMALLY THE FIRST PROGRAM RUN WHEN CHECKING OUT AN RL8A(RL278)/RL01 OR RL8A(RL278)/RL02 SUBSYSTEM.

S.1.5 ASSUMPTIONS

THE CPU, CONSOLE DEVICE, AND ALL AVAILABLE MEMORY (SPECIFIED BY THE OPERATOR) ARE ASSUMED TO BE FAULT FREE. IF A FAULT IS SUSPECTED IN THESE UNITS, THE APPROPRIATE DIAGNOSTICS SHOULD BE RUN FIRST.

ANY USER ATTEMPTING TO DIAGNOSE THE HARDWARE (I.E. USE THE DIAGNOSTIC FOR OTHER THAN GO/NO-GO TESTING) IS ASSUMED TO HAVE SOME KNOWLEDGE OF THE CONTROLLER AND PDP-8 HARDWARE.

S.2.0 OPERATING INSTRUCTIONS
-----**S.2.1 LOADING AND STARTING PROCEDURES**

THE PROGRAM MAY BE LOADED BY PAPER TAPE, APT, OR STANDARD OS/8 LOADING PROCEDURES. THE STARTING ADDRESS IS 200. THERE IS NO OTHER START OR RESTART ADDRESS AND NO WAY TO SELECT A SUBSET OF THE TESTS TO BE EXECUTED.

S.2.2 SPECIAL ENVIRONMENTS
-----**S2.2.1 MULTIPLE OPTION TESTER**

IF RUNNING ON THE PDP-8 MULTIPLE OPTION TESTER, LOCATION 22 BIT 1 MUST BE SET TO A 1. (I.E. LOC 22 = 2000)
ALL DEVICE CODES MUST BE THE SAME.

IT IS POSSIBLE TO TEST ONLY 8 DEVICES (INSTEAD OF 16) USING THE SWITCH REGISTER. UNDER APT, IT IS POSSIBLE TO EITHER CONTINUE TESTING OR REPORT FAILURES TO APT USING THE SWITCH REGISTER.

EXCEPT WHEN ALSO RUNNING UNDER APT, UP TO 3 ERRORS ARE ALLOWED AND REPORTED BEFORE DROPPING THE UNIT. THE FAILURE LIGHT WILL BE LIT AFTER THE FIRST FAILURE. ERROR MESSAGES WILL BE PRECEDED BY A TWO DIGIT OCTAL (!NOT DECIMAL!) NUMBER, IDENTIFYING THE UNIT.

S.2.2.2 APT/8A

THE PROGRAM MAKES NO ATTEMPT TO USE THE CONSOLE DEVICE WHEN RUNNING UNDER APT (LOCATION 22, BIT 0=1). ALL ERRORS ARE FATAL AND REPORTED TO APT, UNLESS ON MULTIPLE OPTION TESTER. APT IS NOTIFIED THAT THE PROGRAM IS STILL RUNNING IN THE STANDARD MANNER.

THE FOLLOWING BITS ARE RELEVANT IN THE HARDWARE CONFIGURATION WORDS:

LOC 21	BITS 0	=0	USE PSEUDO SWITCH REGISTER (LOC 20)
		=1	USE HARDWARE SWITCH REGISTER
	BITS 7-11		MEMORY SIZE INFORMATION (1 LESS THAN THE AMOUNT OF MEMORY AVAILABLE, E.G.
			0 = 1K
			7 = 8K
			37 = 32K
LOC 22	BIT 0	=0	NOT ON APT
		=1	ON APT 8A SYSTEM
	BIT 1	=0	NOT ON MULTIPLE OPTION TESTER
		=1	THE PROGRAM IS RUNNING ON THE MULTIPLE OPTION TESTER.

S.2.3 PROGRAM OPTIONS

S.2.3.1 OPERATOR PROMPTS

THE CONSOLE PACKAGE (DESCRIBED BELOW) IS ALWAYS ACTIVE AND CANNOT BE DISABLED. THE PROGRAM DEFAULTS TO USE OF THE PSEUDO SWITCH REGISTER (LOCATION 20). THE HARDWARE SWITCH REGISTER CAN BE USED BY CHANGING LOCATION 21 BIT 0 TO A 1. THE PSEUDO SWITCH REGISTER CAN BE MODIFIED BY USING THE CONSOLE PACKAGE.

IF THE PSEUDO SWITCH REGISTER IS BEING USED, THE PROGRAM WILL ALLOW THE OPERATOR TO ENTER THE SWITCH REGISTER VALUE AT PROGRAM STARTUP.

THE OPERATOR WILL NEXT BE PROMPTED FOR MEMORY SIZE INFORMATION. THIS VALUE IS ENTERED IN DECIMAL AND MUST BE TERMINATED WITH EITHER A CARRIAGE RETURN OR A "K". THE DEFAULT VALUE IS 8K (GOTTEN BY TYPING A <CR>). THIS INFORMATION IS USED TO TEST THE DATA BREAK FEATURE TO ALL AVAILABLE FIELDS. IF IT IS DESIRED TO CHANGE THIS VALUE, THE PROGRAM MUST BE RESTARTED.

THE OPERATOR IS THEN ASKED IF HE WISHES TO EXECUTE THE PHASE LOCKED LOOP ADJUSTMENT ROUTINE. THIS QUESTION MUST BE ANSWERED WITH A "Y" FOR YES OR A "<CR>" OR "N" FOR NO. THE ROUTINE WILL ONLY BE EXECUTED IN THE FIRST PASS OF THE PROGRAM.

S.2.3.2 PHASE LOCKED LOOP ADJUSTMENT ROUTINE

THIS ROUTINE LOOPS ON A MAINTENANCE COMMAND (FUNCTION 0), WITH THE DATA PATTERN TAKEN FROM THE CONTENTS OF THE SWITCH REGISTER. THE RESULTS OF THE MAINTENANCE WRITE BREAK ARE DISPLAYED IN THE MQ. THIS ALLOWS THE OPERATOR TO ADJUST THE PHASE LOCKED LOOP POTENTIOMETER WHILE VIEWING THE MQ DISPLAY. THE POT SHOULD BE ADJUSTED TO THE CENTER OF THE RANGE WHERE THE DISPLAY IS STABLE (ASSUMING THE CIRCUIT IS WORKING OTHERWISE). SEVERAL DATA PATTERNS SHOULD BE TRIED, SINCE THE PHASE LOCKED LOOP PICKS UP BITS EASILY.

A <CR> IS TYPED TO EXIT THE TEST. BEFORE EXITING, THE PROGRAM WILL ISSUE THE "SR=XXXX" PROMPT OF THE CONSOLE PACKAGE IN CASE THE OPERATOR HAS FORGOTTEN TO CHANGE THE SWITCH REGISTER BACK TO A DESIRED SETTING.

NOTE: IT IS EASIER TO USE THE HARDWARE SWITCHES ON THIS TEST, BUT THE CONTROL-G IS STILL OPERATIVE IN CASE IT IS DESIRED TO USE THE SOFTWARE SWITCHES. (SEE CONSOLE PACKAGE SUMMARY.) THE NORMAL SWITCH REGISTER BITS HAVE NO MEANING DURING THIS ROUTINE; THE SWITCH REGISTER CONTENTS ARE TREATED STRICTLY AS DATA FOR THE MAINTENANCE COMMAND TRANSFER.

IMPORTANT: THIS ROUTINE SHOULD NOT BE EXECUTED UNTIL THE PREVIOUS TEST (FUNCTION 0 TEST) HAS PASSED AT LEAST TO THE POINT OF THE ERROR MESSAGE "MAINT WRITE DID NOT WRITE BACK WORD READ OR MA BAD" AND IT IS VERIFIED THAT THE MA IS OK (THAT THE WRITE BREAK WORD IS BAD). IF THE ROUTINE IS ENTERED BEFORE THE FUNCTION 0 COMMAND IS WORKING TO THIS POINT, THE ROUTINE MAY HANG OR WILL NOT ACCOMPLISH THE DESIRED PURPOSE.

S.2.3.3 MEANING OF SWITCH REGISTER BITS

 THE SWITCH REGISTER BITS AND THEIR MEANINGS ARE DESCRIBED BELOW:

SR BIT SETTING MEANING

SR0 =0 HALT ON ERROR
 (4000) =1 INHIBIT ERROR HALT

SR1 =0 DO NOT LOOP IF ERROR (OVERRIDDEN IF SR2=1)
 (2000) =1 LOOP ON ERROR (HARD OR SOFT)

SR2 =0 MAKE NORMAL LOOP DECISIONS
 (1000) =1 LOOP ON CURRENT TEST

SR3 =0 CONTINUE AFTER END OF PASS
 (0400) =1 HALT AT END OF PASS

SR4 =0 REPORT ERROR ON CONSOLE DEVICE
 (0200) =1 INHIBIT ERROR REPORT
 NOTE: CARE SHOULD BE TAKEN WHEN USING THE HARDWARE
 SWITCHES THAT THIS BIT IS NOT INADVERTENTLY
 LEFT SET WHEN STARTING THE PROGRAM.

SR5 =0 PROGRAM WILL PROMPT OPERATOR FOR MEMORY SIZE
 (0100) INFORMATION AND PSEUDO SWITCH REGISTER SETTING
 ON START OR RESTART
 =1 PROGRAM WILL USE DEFAULT (8K,SWR=0) VALUES (OR
 PREVIOUS RESPONSES) TO OPERATOR PROMPTS ON
 START (OR RESTART).

SR6 =0 REPORT END OF PASS
 (0040) =1 INHIBIT END OF PASS TYPEOUT

THIS SWITCH IS TESTED ONLY IF RUNNING UNDER APT.

SR7 =0 IF ON MULTIPLE OPTION TESTER, CONTINUE
 (0020) TESTING NEXT UUT ON ERROR.
 =1 IF ON MULTIPLE OPTION TESTER, NOTIFY APT OF FIRST FAILURE

SR8 =0 IF ON MULTIPLE OPTION TESTER, TEST 16 UUT'S.
 (0010) =1 TEST ONLY 8 UUT'S ON MULTIPLE OPTION TESTER.
 UNITS 0,2,4,6,10,12,14,16 (OCTAL) WILL BE TESTED.
 UNITS 1,3,5,7,11,13,15,17 (OCTAL) WILL BE IGNORED.

SR11 =0 USE IOT CODES 60,61
 (0001) =1 USE IOT CODES 62,63
 NOTE: CODES ARE MODIFIED AT THE BEGINNING OF
 EACH PROGRAM PASS.

S.2.3.4 CONSOLE PACKAGE COMMAND SUMMARY

SEQ 8

A MORE COMPLETE DESCRIPTION OF THE CONSOLE PACKAGE MAY BE FOUND IN DOCUMENT NO. 175-003-009-02. BELOW IS A SUMMARY OF AVAILABLE CONTROL FUNCTIONS. A "CONTROL" CHARACTER IS TYPED BY HOLDING DOWN THE "CONTROL" KEY AND HITTING THE CHARACTER.

CNTRL-G OPEN THE PSEUDO SWITCH REGISTER FOR MODIFICATION. THE PROGRAM WILL TYPE "G" FOLLOWED BY:
"SR=XXXX ", WHERE XXXX IS THE SWITCH REGISTER SETTING. THE VALUE MAY BE LEFT UNCHANGED OR A NEW VALUE MAY BE ENTERED. TYPING A <CR> WILL SAVE THE NEW VALUE (IF ONE WAS ENTERED) AND CONTINUE THE PROGRAM. TYPING A LINE FEED WILL SAVE THE NEW VALUE (IF ENTERED) AND RESTART THE PROGRAM AT LOCATION 200. TYPING ANY OTHER CHARACTER WILL CAUSE THE PROMPT TO BE RE-ISSUED.

NOTE: IF THE HARDWARE SWITCHES ARE BEING USED, THEIR VALUE WILL BE TYPED OUT, BUT ANY NEW VALUE TYPED IN WILL BE SAVED IN LOCATION 20. THE HARDWARE SWITCHES WILL STILL CONTROL THE PROGRAM.

CNTRL-C RETURN TO OS/8 MONITOR (AT LOCATION 7600). IF THE MONITOR BOOT IS NOT PRESENT, THE RESULTS ARE INDETERMINATE.

CNTRL-S DISABLE ALL OUTPUT. THE PROGRAM WILL HANG WHEN IT ATTEMPTS TO OUTPUT TO THE CONSOLE DEVICE. AT THIS POINT, THE ONLY CHARACTERS THAT WILL BE RECOGNIZED ARE THE CONTROL-C AND CONTROL-Q.

CNTRL-Q REENABLE ALL OUTPUT (AFTER A ^S).

CNTRL-F THIS IS A NON-STANDARD CONTROL CHARACTER. THE CONSOLE FILL COUNT (LOCATION 23) IS OPENED FOR MODIFICATION. BEHAVIOR IS SIMILAR TO THAT FOR CONTROL-G, EXCEPT THAT A LINE FEED WILL NOT BE RECOGNIZED. THE DEFAULT FILL COUNT IS 1. A FILL COUNT OF 14 IS USUALLY REQUIRED FOR LA30'S.

ALL OTHER CHARACTERS WILL BE ECHOED, FOLLOWED BY A "?" AND <CR>.

S.2.4 EXECUTION TIMES

THE FIRST PASS IS QUICK VERIFY AND TAKES LESS THAN 5 SECONDS. (OPERATOR ENTRY OF PARAMETERS REQUIRES SOME SHORT ADDITIONAL TIME.) THE QUICK VERIFY PASS USES A LIMITED SET OF DATA PATTERNS.

SUBSEQUENT PASSES USE ALL DATA PATTERNS AND ARE A MORE EXHAUSTIVE TEST OF THE HARDWARE. THESE PASSES REQUIRE APPROXIMATELY ONE AND ONE HALF MINUTES FOR AN ERROR FREE PASS.

S.3.0 ERROR INFORMATION -----

S.3.1 ERROR REPORTING PROCEDURES -----

ALL ERRORS HAVE THE FOLLOWING FORMAT:

ERROR MESSAGE
DATA HEADERS
DATA

THE ERROR MESSAGE DESCRIBES THE FAILURE. DATA HEADERS VARY IN NUMBER AND IDENTIFY THE DATA TYPED BELOW EACH HEADER. ALL DATA ARE TYPED IN OCTAL. THE PC OF THE ERROR CALL IS ALWAYS INCLUDED. IN ADDITION, IF ON THE MULTIPLE OPTION TESTER, THERE WILL BE AN OCTAL NUMBER ON THE LINE ABOVE THE ERROR MESSAGE, IDENTIFYING THE UNIT TO WHICH THE ERROR MESSAGE APPLIES.

THE FOLLOWING ABBREVIATIONS ARE USED:

CA COMMAND A REGISTER
CB COMMAND B REGISTER
SA SECTOR ADDRESS REGISTER
WC WORD COUNT REGISTER
ER ERROR REGISTER
MA MEMORY ADDRESS REGISTER
EXPCD EXPECTED (I.E. THE CORRECT VALUE)
ACTUAL THE ACTUAL (INCORRECT) VALUE
HI8-LO8 HIGH AND LOW ORDER 8 BITS OF THE DAR
DAR DISK ADDRESS REGISTER
AC ACCUMULATOR
CALLPC ERRORS REPORTED WITHIN SUBROUTINES INCLUDE THE PC OF THE SUBROUTINE CALL

S.3.2 ERROR HALTS -----

THE ONLY ACTUAL HALT INSTRUCTION IS WITHIN THE POWER FAIL ROUTINE. ALL OTHER "HALTS" ARE CALLS TO THE CONSOLE PACKAGE SWITCH REGISTER MODIFICATION ROUTINE. THIS IS TRUE EVEN IF THE HARDWARE SWITCHES ARE SELECTED. THE ONLY SUCH "HALTS" ARE IN THE END OF PASS ROUTINE AND THE ERROR REPORTING ROUTINE.

S.4.0 PROGRESS REPORTS

ONLY ONE PROGRESS REPORT IS GIVEN; THAT IS GIVEN AT THE END OF PASS. IT HAS THE FOLLOWING FORMAT:

END PASS XXXX

WHERE XXXX IS THE PASS COUNT IN OCTAL. NO TOTAL ERROR COUNT IS REPORTED, SO WHEN RUNNING THE PROGRAM UNATTENDED, IT IS RECOMMENDED THAT EITHER ERROR HALTS, ERROR PRINTOUTS, OR LOOPING ON ERROR BE ENABLED TO AVOID PASSING A DEFECTIVE BOARD.

S.5.0 DEVICE INFORMATION TABLES

S.5.1 INSTRUCTION SET

TRANSFERS FROM THE AC TO REGISTER IN THE CONTROLLER CLEAR THE AC AFTER THE TRANSFER IS COMPLETE. TRANSFERS TO THE AC FROM REGISTERS IN THE CONTROLLER CLEAR THE AC FIRST THEN THE TRANSFER TAKES PLACE.

THE SKIP INSTRUCTIONS IN THIS INSTRUCTION SET ARE SKIP AND THEN CLEAR IOT'S. THIS MEANS THAT IF A GIVEN CONDITION IS TRUE, IE; "FUNCTION DONE" IS TRUE (SET TO A LOGIC ONE) THE FUNCTION DONE FLAG WILL BE CLEARED AT THE COMPLETION OF THE SKIP IOT.

THE DEVICE CODE IS JUMPER SELECTABLE FOR EITHER 60,61 OR 30,31. 60,61 IS STANDARD. IOT CODES 30,31 MAY BE USED BY SETTING SWITCH REGISTER BIT 11 TO A 1.

OCTAL CODE -----	MNEMONIC -----	FUNCTION -----
6600	RLDC	CLEAR DEVICE, ALL REGISTERS, AC AND FLAGS.
6601	RLSD	SKIP ON FUNCTION DONE, THEN CLEAR IF SET TO A ONE.
6602	RLMA	LOAD BREAK MA REGISTER FROM AC 0:11
6603	RLCA	LOAD COMMAND REGISTER "A" FROM AC 0:11
6604	RLCB	LOAD COMMAND REGISTER "B" FROM AC 0:11, EXECUTE COMMAND
6605	RLSA	LOAD SECTOR ADDRESS FROM AC 0:5
6606	----	SPARE (WILL CLEAR THE AC)
6607	RLWC	LOAD WORD COUNT FROM AC 0:11
6610	RRER	READ ERROR REGISTER INTO AC 0:2,11.
6611	RRWC	READ WORD COUNT INTO AC 0:11
6612	RRCA	READ COMMAND REGISTER "A" INTO AC 0:11
6613	RRCB	READ COMMAND REGISTER "B" INTO AC 0:11
6614	RRSA	READ SECTOR ADDRESS INTO AC 0:5
6615	RRSI	READ SILO WORD (8 BIT) INTO AC 4:11
6616	RLSR	SPARE (AC NOT AFFECTED)
6617	RLSE	SKIP ON COMPOSITE ERROR, THEN CLEAR IF SET TO A ONE

S.5.2 CONTROLLER REGISTERS

BREAK MA REGISTER

AC0- MSB

"

AC11-LSB

COMMAND REGISTER "A"

AC0- DIRECTION; ZERO= MOVE HEADS AWAY

FROM SPINDLE (LOWER CYL. ADD.)

ONE= MOVE TOWARDS SPINDLE (HI ADD.)

AC1- HEAD SELECT- ZERO= UPPER HEAD

ONE= LOWER HEAD

AC2- SPARE

RL01

RL02

AC3- RESERVED

CYLINDER ADDRESS/DIFFERENCE WORD;MSB

AC4- CYLINDER ADDRESS/DIFFERENCE WORD;MSB

"

"

AC11- CYLINDER ADDRESS/DIFFERENCE WORD; LSB

COMMAND REGISTER "B"

AC0- MAINT. INHIBIT. PREVENT THE FOLLOWING SIGNALS

FROM GOING TO/FROM THE DRIVE;

WRT GATE, WRT DATA, DRV CMND, DRV STAT., DRV RDY

SEC PLS, READ DATA, STAT. CLK, DRV ERR.

AC1- MAINTENANCE- LOOP DAR TO SILO SERIAL IN

AC2- MODE; ZERO= TRUNCATED (128 12 BIT WORDS/SECTOR)

ONE= BYTE (256 8 BIT WORDS/SECTOR)

MUST BE SET TO A 1 WHEN DOING

A "GET STATUS" OR "READ HEADER" COMMAND.

AC3- INTERRUPT ENABLE

AC4- DRIVE SELECT; MSB

AC5- DRIVE SELECT; LSB

AC6- EMA; MSB

AC7- "

AC8- EMA; LSB

AC9- FUNCTION BIT C

AC10- " " B

AC11-FUNCTION BIT A

FUNCTION BIT DEFINITION

BIT C	BIT B	BIT A	COMMAND
-----	-----	-----	-----
0	0	0	MAINTENANCE
0	0	1	RESET
0	1	0	GET STATUS
0	1	1	SEEK
1	0	0	READ HEADER
1	0	1	WRITE DATA
1	1	0	READ DATA
1	1	1	READ DATA
			WITHOUT HEADER CHECK

SECTOR ADDRESS

AC0- SECTOR ADDRESS; MSB

"

"

AC5- SECTOR ADDRESS; LSB

ERROR REGISTER

AC0- CRC ERROR

AC1- OPERATION INCOMPLETE (OP1)

AC2- DATA LATE/HEADER NOT FOUND

AC10-DRIVE ERROR

AC11-DRIVE READY; A ONE= DRV RDY

A ZERO= DRV RDY(NOT)

NOTE:DRIVE ERROR AND DRIVE READY ARE ASSERTED WHEN
DRIVE INTERFACE IS DISABLED (CBO = 1).

WORD COUNT REGISTER; A 12 BIT WORD COUNT

REGISTER HAS BEEN PROVIDED TO ALLOW UP TO

4,096 DATA BREAKS (DMA'S) TO TAKE PLACE AT

ONE TIME. THIS REGISTER IS LOADED WITH

THE RLWC IOT FROM AC 0:11, WHICH MUST BE

THE 2'S COMPLIMENT OF THE NUMBER OF TRANSFERS

THAT ARE TO TAKE PLACE.

S.6.0 SUBTEST SUMMARIES

TEST 0 VERIFY NON-SKIPPING IOTS DO NOT SKIP.

TEST 1 VERIFY SKIPS DO NOT AFFECT AC.

LOAD AC (ALL 1'S FIRST PASS--ALL COMBINATIONS THEREAFTER).

EXECUTE RLSD,RLSE,RLSR AND VERIFY AC IS UNCHANGED.

TEST 2 VERIFY EACH R/W REGISTER CAN BE WRITTEN, READ, AND CLEARED.

NOTE: CURRENT READ/WRITE REGISTERS ARE: CA, CB, SA, WC.

LOAD AC WITH TEST PATTERN. WRITE REGISTER. VERIFY AC CLEARED.

READ REGISTER BACK AND VERIFY CONTENTS. ISSUE RLDC AND VERIFY

REGISTER NOW CLEAR. REPEAT FOR ALTERNATE PATTERN. REPEAT

ENTIRE SEQUENCE FOR EACH R/W REGISTER. FOR CB, USE ONLY

"RESET" IN FUNCTION BITS AND BIT 0 = 1 (DRIVE INTERFACE OFF).

TEST 3 VERIFY AC CLEARED BY OTHER IOTS.

LOAD AC (ALL 1'S FIRST PASS--ALL COMBINATIONS THEREAFTER).

WRITE TO MA REGISTER AND VERIFY AC WAS CLEARED. ALSO

CHECK AC IS CLEARED BY "SPARE" IOT AND RLDC.

TEST 4 VERIFY NO INTERACTION BETWEEN IOTS.

LOAD CB WITH A TEST PATTERN. ISSUE RLCA, RLSA, AND RLHA AND

VERIFY CB IS UNCHANGED. LOAD WC AND SA WITH ALTERNATE PATTERNS

AND ISSUE SPARE AND RRSI AND VERIFY WC AND SA ARE UNCHANGED.

TEST 6 VERIFY DONE BIT SETS AND CLEARS.

ISSUE "RESET" AND WAIT FOR DONE TO SET. CHECK THAT RLSD SKIPS.

CHECK THAT SECOND RLSD DOES NOT SKIP.

- TEST 7 VERIFY RLDC CLEARS DONE BIT.
ISSUE "RESET" AND WAIT FOR DONE. ISSUE RLDC AND CHECK THAT RLSD DOES NOT SKIP.
- TEST 8 VERIFY LOADING CB CLEARS DONE; RLDC TERMINATES ACTION.
ISSUE "RESET" AND WAIT FOR DONE. ISSUE ANOTHER RESET AND CHECK THAT RLSD DOES NOT SKIP. ISSUE RLDC AND VERIFY DONE DOES NOT SET.
- TEST 9 VERIFY INTERRUPT REQUEST GENERATED BY DONE.
CLEAR DEVICE. VERIFY NO INTERRUPTS ARE ALREADY PENDING. ISSUE "RESET" WITH INTERRUPT ENABLE SET. VERIFY INTERRUPT REQUEST IS ASSERTED ON THE BUS. CLEAR DONE AND VERIFY INTERRUPT REQUEST IS RESET.
- TEST 10 VERIFY INTERRUPT NOT REQUESTED IF INTERRUPT ENABLE NOT SET.
ISSUE "RESET" WITHOUT INTERRUPT ENABLE. VERIFY NO INTERRUPT REQUEST IS GENERATED.
- TEST 11 VERIFY SILO CAN BE CLEARED BY RLDC.
SET UP TO LOAD 1'S INTO FORCIBLE BITS OF DISK ADDRESS REGISTER
LOAD CB WITH "RESET" AND MAINTENANCE BIT 1 SET (DAR TO SILO).
ISSUE RLDC. SET UP TO WRITE 0'S TO DAR AND ISSUE RESET COMMAND (DAR TO SILO). VERIFY THAT FIRST TRANSFER IS NOT THERE.
- TEST 12 VERIFY MAINTENANCE DISK ADDRESS REGISTER TO SILO TRANSFER OK.
CLEAR DEVICE. SET UP FOR DESIRED DAR LOADING (ALTERNATE 1'S AND SET. READ SILO AND VERIFY IT IS AS EXPECTED. REPEAT FOR ALTERNATE PATTERN.
- TEST 13 VERIFY SILO CAN BE CLEARED BY CLK OPI DLY.
SET UP TO LOAD 1'S INTO FORCIBLE BITS OF DISK ADDRESS REGISTER.
LOAD CB WITH "RESET" AND MAINTENANCE BIT 1 SET (DAR TO SILO).
ISSUE NON-MAINTENANCE RESET TO ASSERT CLK OPI DLY TO THE SILO.
DO A DAR TO SILO TRANSFER AND VERIFY THAT THE FIRST TRANSFER IS NOT THERE.
- TEST 14 VERIFY GET STATUS DAR TO SILO TRANSFER; CA LOADS PROPERLY IN DAR TRANSFER TO SILO WAS OK. (SA CLEAR ON THIS TEST.)
- TEST 15 DO GET STATUS TO DRIVE 0 WITH INTERFACE DISABLED (MAINT BIT 0 SET) AND VERIFY OPI SETS. IF A RESPONSE WAS RECEIVED FROM THE DRIVE, PREVENT PROGRAM FROM CONTINUING.
- TEST 16 VERIFY PROPER LOADING OF CA INTO DAR.
CLEAR SA AND USE ALTERNATE PATTERN FROM ONE USED IN TEST 14 TO LOAD CA. ISSUE MAINTENANCE SEEK AND VERIFY TRANSFER TO SILO WAS OK.

0'S IN FORCIBLE BITS) AND LOAD CB WI

CLEAR DEVICE. ISSUE MAINTENANCE "GE

- TEST 17 VERIFY PROPER DAR LOADING OF SA.
CLEAR DEVICE. LOAD TEST PATTERN INTO SA. ISSUE GET STATUS
WITH MAINT BIT 1 SET. VERIFY TRANSFER TO SILO OK. REPEAT FOR
ALTERNATE PATTERN IN SA.
- TEST 18 VERIFY 12 BIT DAR TO SILO TRANSFER.
ISSUE DAR TO SILO RESET IN 12 BIT MODE AND VERIFY TRANSFER TO SILO
WAS OK.
- TEST 19 VERIFY SILO CAN BE FILLED WITHOUT ERRORS.
USE MAINTENANCE MODE TO FILL SILO COMPLETELY. VERIFY NO ERROR
BITS WERE SET. READ BACK ALL SILO WORDS AND VERIFY NO ERRORS.
- TEST 20 VERIFY OVERFILLING SILO SETS READ DATA LATE.
OVERFILL SILO USING MAINTENANCE MODE. VERIFY DATA LATE SET IN
ERROR REGISTER. VERIFY RLDC CLEARS DATA LATE IN ERROR REGISTER.
- TEST 21 VERIFY PROPER EMPTYING OF SILO.
FILL SILO. READ TWO WORDS. INITIATE TRANSFER. VERIFY NO
ERRORS. READ ONE WORD. INITIATE ONE TRANSFER (OVERFILL BY
ONE). VERIFY DATA LATE SET. ISSUE RLDC. REFILL SILO AND
CHECK FOR NO ERRORS TO VERIFY SILO WAS CLEARED.
- TEST 22 VERIFY OPI SET BY SEEK, CLEARED BY RLDC.
ISSUE SEEK COMMAND TO DRIVE. VERIFY DRIVE READY STAYS ASSERTED.
IF NOT, INHIBIT PROGRAM FROM CONTINUING. VERIFY OPI AND DONE SET.
CHECK THAT DRIVE ERROR INHIBITED OPI FROM SETTING ERROR FLAG.
CHECK THAT RLDC CLEARS OPI.
- TEST 23 VERIFY CAF TERMINATES ACTION AND INHIBITS OPI
ISSUE A SEEK TO THE DRIVE. ISSUE CAF AND VERIFY OPI AND DONE
DO NOT SET.
- TEST 24 VERIFY OPI SET BY READ HEADER, CLEARED BY LOADING CB.
ISSUE READ HEADER. VERIFY DONE AND OPI ARE SET. LOAD CB.
VERIFY ERROR BIT CLEARED.

- TEST 25 VERIFY OPI AND DONE SET BY READ DATA.
- TEST 26 VERIFY CLK OPI DLY GETS ASSERTED TO SILO BY READ DATA W/O HEADER CHECK.
DO A DAR TO SILO RESET. ISSUE COMMAND 7. VERIFY ERROR FLAG AND
DONE WERE SET BY COMMAND 7 MICROCODE. VERIFY RLSE CLEARED THE
ERROR FLAG. DO A DAR TO SILO RESET AND VERIFY THE SECOND TRANSFER
IS READ FROM THE SILO.
- TEST 27 VERIFY INTERRUPT REQUEST GENERATED BY ERROR.
VERIFY NO PRIOR INTERRUPT REQUESTS ARE PENDING ON THE BUS.
SET ERROR FLAG BY ISSUING READ DATA W/O HEADER CHECK WITH MAINT
BIT 0 SET. CLEAR DONE BIT AND VERIFY AN INTERRUPT IS PENDING
ON THE BUS. CLEAR ERROR AND VERIFY INTERRUPT IS NO LONGER
REQUESTED.
- TEST 28 VERIFY ERROR FLAG CLEARED BY RLDC.
SET ERROR FLAG BY ISSUING COMMAND 7 W/MAINT BIT 0 SET. ISSUE
RLDC AND VERIFY RLSE DOES NOT SKIP.
- TEST 29 VERIFY MAINTENANCE WRITE BREAK (12 BIT) WORKS.
SET UP BUFFER AT LOCATION 15252. SET WORD COUNT TO READ AND
WRITE BACK TWO WORDS (STANDARD PATTERNS) WITH CRC. ISSUE
MAINTENANCE WRITE. VERIFY WORD COUNT OVERFLOW STOPPED TRANSFER.
VERIFY DATA AND CRC WORDS IN BUFFER.
- PHASE LOCKED LOOP ADJUSTMENT ROUTINE
THIS ROUTINE IS ENTERED IN THE FIRST PASS ONLY AND ONLY IF REQUESTED
BY THE OPERATOR IN RESPONSE TO A QUESTION AT PROGRAM STARTUP.
THE ROUTINE LOOPS ON A MAINT COMMAND WITH THE DATA TAKEN FROM THE
CONTENTS OF THE SWITCH REGISTER AND THE RESULT DISPLAYED IN THE
MQ. THIS ENABLES THE OPERATOR TO ADJUST THE PHASE LOCKED LOOP
POTENTIOMETER WHILE VIEWING THE RESULTS.
- TEST 30 VERIFY WORD COUNT OVERFLOW WILL INHIBIT DMA'S WHEN NOT EXPECTED
BY U-CODE.
ISSUE MAINT COMMAND WITH INITIAL WORD COUNT REGISTER OF -2.
VERIFY DONE SETS AND WORD COUNT DOES NOT GO POSITIVE.
- TEST 31 VERIFY MA CLEARED BY RLDC.
SET UP BUFFER AT LOCATION 0 FIELD 1. LOAD MA (TO NON-ZERO
BUFFER). ISSUE RLDC AND MAINTENANCE WRITE. VERIFY DATA
TRANSFER TOOK PLACE AT LOCATION 0 BUFFER.
- TEST 32 VERIFY CLK OPI DLY GETS ASSERTED BY WRITE COMMAND; VERIFY DATA
BREAKS OCCUR.
SET UP A WORD AT LOCATION 10000. DO A DAR TO SILO TRANSFER.
ISSUE WRITE COMMAND FROM FIELD 1. WAIT LONG ENOUGH FOR DATA BREAKS
TO OCCUR AND READ THE SILO. VERIFY THAT THE CORRECT WORD FROM
FIELD 1 WAS READ.
- TEST 33 VERIFY DATA BREAKS TAKE PLACE ON READ COMMAND.
DO A DAR TO SILO TRANSFER. CLEAR OUT A BUFFER IN FIELD 1.
ISSUE A READ COMMAND TO FIELD 1. VERIFY THE DAR WORDS ARE
WRITTEN TO THE BUFFER.

TEST 34 MAINT WRITE TO FIELD 1 -- ROTATE BUFFERS.
SET UP A BUFFER AT LOCATION 12525. USE ALTERNATE PATTERNS IN
BUFFER THAN ONE USED IN TEST 29. ISSUE MAINT COMMAND AND
VERIFY DATA AND CRC WORDS.

TEST 35 VERIFY MAINTENANCE WRITE BREAK TO FIELD 0.
SET UP A BUFFER IN FIELD 0. VERIFY BREAK TO BUFFER.

TEST 36 VERIFY DATA BREAK FROM ALL OF FIELD 1.
ISSUE WRITE COMMAND FROM FIELD 1. READ 4096
SILO WORDS AND COMPARE WITH WORDS FROM FIELD 1.

TEST 37 VERIFY DATA BREAK TO ALL AVAILABLE FIELDS.
USE EXTENDED MEMORY INFORMATION FROM OPERATOR OR APT TO CHECK
MAINTENANCE WRITE TO ALL OTHER AVAILABLE FIELDS. IF ALL OF
FIELD TWO IS AVAILABLE, THEN THE ENTIRE FIELD WILL BE WRITTEN
WITH DATA AND A WRITE COMMAND FROM FIELD 2 WILL BE EXECUTED.
THE 4K TRANSFER TO THE SILO WILL BE VERIFIED.

S.7.0 PROGRAM, SYMBOL TABLE, AND CROSS REFERENCE LISTING

/AJRLA-D-0 RL8A DISKLESS CONTROLLER DIAGNOSTIC

/ REVISION HISTORY

/ -----

/ MODIFICATIONS BY

/ -----

/ H. POULTER FEBRUARY 1979

/

/ HP 001 -- VERSION "C" CHANGES FOR RL02 UPGRADE.
/ ONE CHANGE TO TEXT FILE "MANDEC" TO PRINT VERSION "C"

/ MODIFICATIONS BY

/ -----

/ M. LETENDRE JANUARY 1981

/

/ ML 001 -- VERSION "D" CHANGES FOR THE VT278 COMPATABILITY UPGRADE./

/ FIELD 0

/ *0

0000 0000

0000 0304

0001 5402

0002 5704

0003 5720

*D
JMP I .+1
PWRFAL
PWRUP/REVISION D
/GO TO POWER FAIL ROUTINE
/POINTER TO POWER FAIL ROUTINE
/POINTER TO POWER UP ROUTINE (A JMP I 3
/INSTRUCTION IS PLACED AT 0)

0010

*10

0010. 0000

AUTO10, 0

0011 0000

AUTO11, 0

0020

*20

0020 0000

PSWR, 0

0021 0000

HCW1, 0

/DEFAULT TO SOFTWARE SWITCHES--4000 FOR HARDWARE SWR
/BITS 7-11 FOR MEMORY SIZE UNDER APT
/BIT 0 FOR APT, BIT 1 FOR MULTIPLE OPTION TESTER
/NUMBER OF FILLER CHARACTERS REQUIRED
/DEVICE CLEAR IOT SUBROUTINE CALL
/POINTERS TO IOT ROUTINES
/SKIP ON DONE IOT SUBROUTINE CALL

0022 0000

HCW2, 0

0023 0001

FILLER, 1

4424

RLDC= JMS I .

0024 5503

XRLDC

4425

RLSD= JMS I .

0025 5511

XRLSD

4426

RLMA= JMS I .

0026 5516

XRLMA

4427

RLCA= JMS I .

0027 5524

XRLCA

4430

RLCB= JMS I .

0030 5532

XRLCB

4431

RLSA= JMS I .

0031 5540

XRLSA

4432

RLWC= JMS I .

0032 5554

XRLWC

4433

RRER= JMS I .

0033 5562

XRRER

4434

RRWC= JMS I .

0034 5600

XRRWC

4435

RRCA= JMS I .

0035 5606

XRRCA

4436

RRCB= JMS I .

0036 5614

XRRCB

4437

RRSA= JMS I .

0037 5622

XRRSA

/READ SECTOR ADDR IOT SUBROUTINE CALL

0040	4440	RRSI=	JMS I	.	/READ SILO WORD IOT SUBROUTINE CALL
	5630		XRRSI		
	4441	RLSR=	JMS I	.	/SKIP ON DRIVE READY IOT SUBROUTINE CALL
0041	5636		XRLSR		
	4442	RLSE=	JMS I	.	/SKIP ON DRIVE ERROR IOT SUBROUTINE CALL
0042	5643		XRLSE		
	4443	SPARE=	JMS I	.	/"SPARE" IOT
0043	5546		XSPARE		
	4444	STOPF=	JMS I	.	/ATTEMPT TO STOP FUNCTION (7 DEVICE CLEARS)
0044	6561		XSTOPF		
	4445	GETSR=	JMS I	.	/READ EITHER HARDWARE OR PSEUDO SWITCHES
0045	4765		XGETSR		
	4446	APTCHK=	JMS I	.	/SKIP IF NOT RUNNING UNDER APT
0046	4564		XAPTCH		
	4447	TICK=	JMS I	.	//GENERATE TIMING FOR APT
0047	5466		XTICK		
	4450	SCOPE=	JMS I	.	/LOOP CONTROL SUBROUTINE
0050	4400		XSCOPE		
	4451	ERROR=	JMS I	.	/ERROR HANDLER
0051	4436		XERROR		
	4452	CONSOL=	JMS I	.	/TELETYPE INPUT HANDLER
0052	4600		XCONSO		
	4453	CNTRLG=	JMS I	.	/CONTROL-G HANDLER--SWITCH REG MODIFY ROUTINE
0053	4704		XCNTRL		
	4454	PAT25=	JMS I	.	/GET PATTERN INTO AC BASED ON 2525
0054	6462		XPAT25		
	4455	PAT52=	JMS I	.	/GET PATTERN INTO AC BASED ON 5252
0055	6472		XPAT52		
	4456	PATCMP=	JMS I	.	/COMPARE AC WITH DATA IN "PATRN"
0056	6506		XPATCH	.	/
	4457	RESET=	JMS I	.	SKIP IF ERROR
					/ISSUE A RESET AND WAIT FOR DONE
0057	5656		XRESET		
	4460	RESETW=	JMS I	.	/ISSUE A RESET AND WAIT LONG ENOUGH FOR DONE
0060	5650		XRESETW		
	4461	RESET3=	JMS I	.	/ISSUE A NON-MAINT. RESET TO DRIVE 3
0061	5665		XRESET3		
	4462	MRSET3=	JMS I	.	/ISSUE MAINT RESET TO DRIVE 3 WITH INTERFACE ENABLED
0062	5674		XMRSET		
	4463	OPIWAT=	JMS I	.	/WAIT FOR OPI AND DONE--RETURN TO CALL+1-4
0063	6031		XOPIWA		
	4464	WAIT=	JMS I	.	/WAIT APPROX 19 USEC.
0064	5766		XWAIT		
	4465	DAREXP=	JMS I	.	/COMPUTE PORTION OF EXPECTED DAR VALUE
0065	5736		XDAREX	.	/
	4466	PATM1=	JMS I	.	AC HAS INFORMATION ON RETURN
					/GET PATTERN INTO AC BASED ON 7777
0066	6502		XPATM1		
	4467	XFRCHK=	JMS I	.	/CHECK IF TRANSFER TO SILO WAS OK AND
0067	6000		XXFRCH	.	/
	4470	CRCCHK=	JMS I	.	SKIP IF ERROR OCCURED
					/GENERATE AND CHECK CRC. REPORT ERROR
0070	6200		XCRCCH	.	/AND SKIP IF NO ERROR. REQUIRES FIELD ARG*10
	4471	ERRCHK=	JMS I	.	/CHECK ERROR FLAG AND REPORT ERROR IF
0071	6335		XERRCH	.	/SET. SKIP IF NOT SET
	4472	MANTEC=	JMS I	.	/CHECK THAT DATA LATE AND ERR FLAG SET OK ON MAINT COMMAND
0072	6523		XMANTE		
	4473	JMPPM1=	JMS I	.	/JMP .-1 EXCEPT ALLOW CONSOLE INPUT

```

0073 6072      XJMPPM
      4474      SLOFIL= JMS I .
0074 6355      XSLOF1
      4475      BRKVFY= JMS I .
0075 6400      XBRKVF
      4476      BUFSET= JMS I .
0076 6453      XBUFSE
      4477      LISN= JMS I .
0077 5101      XLISN
      4500      MESSAGE= JMS I .
0100 5421      MESAGX
      4500      MESSAG= MESSAGE
      4501      PRNT1= JMS I .
0101 5412      XPRNT1
      4502      PRNT2= JMS I .
0102 5000      XPRNT2
      4503      PRNT4= JMS I .
0103 5400      XPRNT4
      4504      SPACE2= JMS I .
0104 5075      SPACX2
      4505      TYPE= JMS I .
0105 5012      XTYPE
      4506      CRLF= JMS I .
0106 5056      XCRLF
      4507      VT278= JMS I .
0107 2135      XVT278
      4510      SCNINT= JMS I .
0110 2114      XSCNIN
      4511      SETINT= JMS I .
0111 2121      XSETIN

0112 0000      ERRPC, 0
0113 0000      DATA1, 0
0114 0000      DATA2, 0
0115 0000      DATA3, 0
0116 0000      DATA4, 0
0117 0000      DATA5, 0
0120 0000      DATA6, 0
0121 0000      LUPCNT, 0
0122 0000      LOOPPT, 0

0123 0000      ERRFLG, 0
0124 0000      PATTRN, 0

0125 0000      TEMP1, 0
0126 0000      TEMP2, 0
0127 0000      NOPRNT, 0
0130 0000      PASCNT, 0
0131 0000      MEMSIZ, 0
0132 0000      PHLKLP, 0
0133 7302      K7302, 7302
0134 5302      K5302, 5302
0135 7700      K7700, 7700
0136 0364      K364, 364
0137 3770      K3770, 3770

```

/FILL SILO USING MAINT RESET

/VERIFY DATA BREAK CAN OCCUR TO A GIVEN
/FIELD (REQUIRES FIELD ARGUMENT E.G. 2)
/SET UP A FOUR WORD MAINT INSTRUCTION BUFFER

/BECAUSE I CAN NEVER REMEMBER WHICH

/PC OF ERROR CALL STORED HERE
/DATA FOR ERROR TYPEOUTS STORED HERE

/NUMBER OF LOOPS MADE BY SCOPE ROUTINE
/POINTER (SET FROM PREVIOUS SCOPE
/STATEMENT) USED FOR LOOPING
/ERROR FLAG USED BY SCOPE AND ERROR
/GOOD DATA STORED HERE FOR COMPARISON
/BY SUBROUTINE "PATCMP"
/TEMPORARY STORAGE ONLY!

/FLAG TO ALLOW OUTPUT (USED W/ CNTRL-S)
/NUMBER OF PASSES OF PROGRAM
/AMOUNT OF MEMORY GETS STORED HERE
/FLAG TO EXECUTE PHASE LOCK LOOP TEST

0140	0100	K100,	100
0141	7301	K7301,	7301
0142	0301	K301,	301
0143	0310	K310,	310
0144	0300	K300,	300
0145	0010	K10,	10
0146	0377	K377,	377
0147	0200	K200,	200
0150	7766	M12,	-12
0151	0020	K20,	20
0152	0400	K400,	400
0153	0003	K3,	3
0154	7600	K7600,	7600
0155	7764	M14,	-14
0156	7760	M20,	-20
0157	0004	K4,	4
0160	0013	K13,	13
0161	4301	K4301,	4301
0162	6301	K6301,	6301
0163	7555	M223,	-223
		M4,	
0164	7774	K7774,	7774
0165	2525	PBUF1B,	BUF1B
0166	7400	PBUF0,	BUF0
0167	0001	K1,	1
0170	7770	M10,	-10
0171	5252	PBUF1A,	BUF1A
0172	7773	M5,	-5
0173	5307	K5307,	5307
0174	5707	K5707,	5707
0175	0037	K37,	37
0176	0177	K177,	177
0177	7701	K7701,	7701

6007	CAF=6007
6003	SRQ=6003
6035	KIE=6035
7002	BSW=7002
6102	SPL=6102
6030	KCF=6030
7421	MQL=7421
6031	KSF=6031
6045	TIE=6045
6032	KCC=6032
6055	ESIA=6055

```

/CLEAR ALL FLAGS
/SKIP ON INTERRUPT REQUEST
/AC11 TO CONSOLE INTERRUPT ENABLE FF

```

```

/SKIP ON POWER LOW
/CLEAR KEYBOARD FLAG
/LOAD MQ FROM AC THEN CLEAR AC
/SKIP ON INPUT FLAG,CLEAR IT ON A VT278
/SET/CLEAR TTY OUTPUT INTRPTS,VIA AC<11>
/CLEAR INPUT FLAG,CLEAR AC ONLY ON VT278
/CLEAR INTERRUPT ENABLE (CRTC)

```

	0200	PAGE	
	0200	*200	
0200	6007	CAF	
0201	3122	DCA	LOOPPT
0202	6035	KIE	
0203	6045	TIE	
0204	3130	DCA	PASCNT
0205	4752	JMS I	MOTINI

```

/CLEAR ALL FLAGS
/FLAG THAT LOOP POINTER IS NOT VALID
/((IN CASE OF A RESTART)
/DISABLE CONSOLE INTERRUPTS INPUT ONLY
/DISABLE CONSOLE INTERRUPTS OUTPUT ONLY
/CLEAR PASS COUNT
/INITIALIZE MULTIPLE OPTION TESTER AND

```

0206	4446	APTCHK		/ERROR TABLE, IF RUNNING ON TESTER
0207	5312	JMP	APTMEM	/SKIP IF NOT ON APT
0210	4500	MESSAG		/SKIP ALL OPENING DIALOG
0211	0340	MANDEC		/TYPE OUT MAINDEC NUMBER
0212	4452	CONSOL		/POINTER TO TEXT (LOCATED IN FIELD 1)
0213	4445	GETSR		/CHECK FOR CONSOLE INPUT
0214	0140	AND	K100	/GET SWITCHES (HARD OR PSEUDO)
0215	7640	SZA CLA		/MASK OUT ALL BUT BIT 5
0216	5316	JMP	IOTCNG	/SKIP IF NOT USING PREVIOUS RESPONSES
0217	1021	TAD	HCW1	/SKIP OVER OPERATOR PROMPTS
0220	7700	SMA CLA		/GET HARDWARE CONFIGURATION WORD 1
0221	4453	CNTRLG		/SKIP IF USING HARDWARE SWITCH REGISTER
0222	4445	GETSR		/LET OPERATOR ENTER SWITCH REGISTER VALUE
0223	0140	AND	K100	/GET SWITCHES (HARD OR PSEUDO)
0224	7640	SZA CLA		/MASK OUT ALL BUT BIT 5
0225	5316	JMP	IOTCNG	/SKIP IF NOT USING PREVIOUS RESPONSES
0226	4500	MEMMES, MESSAG		/SKIP OVER OPERATOR PROMPTS
0227	0362	MEMPRM		/PROMPT OPERATOR FOR MEMORY SIZE
0230	3131	DCA	MEMSIZ	
0231	4477	MEMLIS, LISN		/CLEAR MEMORY SIZE INFORMATION
0232	0001	1		/WAIT FOR INPUT
0233	0253	GOTNUM		/IF OCTAL VALUE INPUT
0234	7510	-*8		/GO TO GOTNUM (GOT NUMBER)
0235	0252	GET8		/IF AN "8" IS INPUT
0236	7507	-*9		/GO TO GET8
0237	0251	GET9		/IF A "9" IS INPUT
0240	7563	-215		/GO TO GET9
0241	0273	Q4PLL		/IF A CARRIAGE RETURN
0242	7465	-*K		/NUMBER IS SAVED--QUESTION FOR PHASE LOCK LOOP TEST
0243	0272	KINP		/IF A "K" IS INPUT
0244	0000	0		/GO TYPE <CR><LF>
0245	0246	.+1		/IF UNMATCHED INPUT
0246	4500	MESSAG		/CONTINUE
0247	0416	QESMRK		/TYPE "?<CR><LF>"
0250	5226	JMP	MEMMES	
0251	7001	GET9, IAC		/GO START OVER
0252	1145	GET8, TAD	K10	/INCREMENT AC (TO GET 9 INTO IT)
0253	3125	GOTNUM, DCA	TEMP1	/ADD 8 INTO AC
0254	1150	TAD	M12	/SAVE NUMBER INPUT
0255	3126	DCA	TEMP2	/GET -10. INTO AC
0256	1131	TAD	MEMSIZ	/SAVE IT IN A COUNTER
0257	2126	ISZ	TEMP2	/
0260	5256	JMP	.-2	/ >MULTIPLY MEMSIZ BY 10.
0261	1125	TAD	TEMP1	//
0262	3131	DCA	MEMSIZ	/ADD IN LSB TO PREVIOUS DIGITS
0263	1131	TAD	MEMSIZ	/SAVE ENTERED MEMORY SIZE SO FAR
0264	1347	TAD	M40	/GET MEMORY SIZE ENTERED
0265	7750	SPA SNA CLA		/SUBTRACT 32 FROM MEMORY SIZE
0266	5231	JMP	MEMLIS	/SKIP IF MEMORY SIZE IS TOO BIG
0267	4500	MESAGE		/GO LISTEN FOR MORE
0270	0420	MEMBIG		/MEMORY SIZE TOO BIG (ENTER 8-J2<CR>):
0271	5230	JMP	MEMLIS-1	
0272	4506	KINP, CRLF		/CLEAR MEMSIZ AND TRY AGAIN
0273	4500	Q4PLL, MESSAGE		/TYPE A <CR><LF> IF A "K" WAS ENTERED
				/EXECUTE PHASE LOCK LOOP ADJUSTMENT ROUTINE?

0274	4464	EXPLL		
0275	4477	LISN		/WAIT FOR RESPONSE
0276	7447	-*Y		/IF "Y" IS TYPED
0277	0306	.*7		/SET THE FLAG TO EXECUTE THE TEST
0300	7462	-*N		/IF "N" IS TYPED,
0301	0307	.*6		/CLEAR THE FLAG
0302	7563	-215		/IF <CR>, THEN HANDLE AS A "NO"
0303	0307	.*4		
0304	0000	0		/ANY OTHER RESPONSE IS INVALID
0305	0273	U4PLL		
0306	7001	IAC		/MAKE THE AC NON-ZERO
0307	3132	DCA	PHLKLP	/SAVE THE FLAG (0 OR 1)
0310	4506	CRLF		
0311	5316	JMP	IOTCNG	/GO CHANGE IOT CODES
0312	1021	APTHM, TAD	HCW1	/GET HARDWARE CONFIGURATION WORD 1
0313	0175	AND	K37	/MASK ALL BUT MEMORY SIZE INFORMATION
0314	3131	DCA	MEMSIZ	/SAVE FOR LATER
0315	3132	DCA	PHLKLP	/DO NOT EXECUTE PHASE LOCK LOOP ADJUSTMENT ROUTINE
0316	4445	IOTCNG, GETSR		/GET SWITCHES
0317	7010	RAR		/GET SW11 INTO LINK
0320	7630	SZL CLA		/SKIP IF USING IOT CODES 60,61
0321	1151	TAD	K20	/ADD IN IOTCODE 2
0322	3126	DCA	TEMP2	/SAVE IOT CODE BIT (0 IF 60,61; 20 IF 62,63)
0323	1346	TAD	PIOTS	/GET POINTER TO TABLE OF IOT POINTERS
0324	3010	DCA	AUTO10	/PUT IN AUTO INDEX REGISTER
0325	6211	IOTLUP, CDF	10	/CHANGE TO FIELD WHERE POINTERS ARE
0326	1410	TAD I	AUTO10	/PICK UP POINTER TO AN IOT FROM TABLE
0327	6201	CDF	0	/BACK TO HERE
0330	7450	SNA		/SKIP IF NOT TABLE TERMINATOR
0331	5340	JMP	SCOPIN	/GO INIT SCOPE ROUTINE
0332	3125	DCA	TEMP1	/SAVE POINTER TO IOT
0333	1525	TAD I	TEMP1	/PICK UP IOT THRU POINTER
0334	0350	AND	K7757	/MASK OUT CHANGABLE PART OF CODE
0335	1126	TAD	TEMP2	/ADD IN DESIRED CODE
0336	3525	DCA I	TEMP1	/SAVE IOT WITH NEW CODE
0337	5325	JMP	IOTLUP	/CONTINUE DOWN IOT POINTER TABLE
0340	3123	SCOPIN, DCA	ERRFLG	/NO ERRORS PRIOR TO TEST1
0341	3121	DCA	LUPCNT	/NO ITERATIONS YET ON TEST 1
0342	1351	TAD	PTEST0	/PICK UP POINTER TO FIRST TEST
0343	3122	DCA	LOOPPT	/MAKE FIRST SCOPE LOOP TO TEST0
0344	4424	RLDC		/CLEAR THE DEVICE
0345	5751	JMP I	PTEST0	/GO TO TEST0
/DATA AREA FOR THIS PAGE				
0346	4523	PIOTS, IOTTAB-1		/POINTER TO TABLE OF IOT POINTERS
0347	7740	M40, -40		
0350	7757	K7757, 7757		
0351	0353	PTEST0, TEST0		
0352	4271	MOTINI, INITMO		/POINTER TO MULTIPLE OPT. TESTER INIT ROUTINE
/EXECUTE ALL NON SKIPPING INSTRUCTIONS				
/IF ANY SKIP, AN ERROR WILL BE REPORTED WITHIN THE IOT SUBROUTINE				
/				
0353	4450	TEST0, SCOPE		
0354	4426	RLMA		

```

0355 4450      SCOPE
0356 4427      RLCA
0357 4450      SCOPE
0360 1161      TAD      K4301      /ISSUE A SAFE COMMAND
0361 4430      RLCB
0362 4464      WAIT
0363 4444      STOPF      /STOP THE FUNCTION
0364 4450      SCOPE
0365 4431      RLSA
0366 4450      SCOPE
0367 4443      SPARE
0370 4450      SCOPE
0371 4432      RLWC
0372 4450      SCOPE
0373 4433      RRER
0374 4450      SCOPE
0375 4434      RRWC
0376 4450      SCOPE
0377 4435      RRCA
0400 4450      SCOPE
0401 4436      RRCB
0402 4450      SCOPE
0403 4437      RRSA
0404 4450      SCOPE
0405 4440      RRSI
0406 4450      SCOPE

/VERIFY SKIPS DO NOT AFFECT AC.
/LOAD AC (ALL 1'S FIRST PASS--ALL COMBINATIONS THEREAFTER).
/EXECUTE RLSD,RLSE,RLSR AND VERIFY AC IS UNCHANGED.
/
0407 4466      TEST1,  PATM1      /GET PATTERN BASED ON ALL 1'S INTO AC
0410 3124      DCA      PATRN      /SAVE PATTERN FOR COMPARISON
0411 1124      TAD      PATRN
0412 4425      RLSD      /SKIP ON DONE
0413 7000      NOP
0414 4442      RLSE      /SKIP ON ERROR
0415 7000      NOP
0416 4441      RLSR      /SKIP ON READY
0417 7000      NOP
0420 4456      PATCMP      /COMPARE AC TO PREVIOUS VALUE
0421 5224      JMP      .+3      /EQUAL-GO TO SCOPE
0422 4451      ERROR      /AC AFFECTED BY A SKIP IOT
0423 0200      E1      /PC.      EXPCTD  ACTUAL
0424 4450      SCOPE

/VERIFY EACH R/W REGISTER CAN BE WRITTEN, READ, AND CLEARED.
/NOTE: CURRENT READ/WRITE REGISTERS ARE: CA, CB, SA, WC.
/LOAD AC WITH TEST PATTERN. WRITE REGISTER. VERIFY AC CLEARED.
/READ REGISTER BACK AND VERIFY CONTENTS. ISSUE RLDC AND VERIFY
/REGISTER NOW CLEAR. REPEAT FOR ALTERNATE PATTERN. REPEAT
/ENTIRE SEQUENCE FOR EACH R/W REGISTER. FOR CB, USE ONLY
/"RESET" IN FUNCTION BITS AND BIT 0 = 1 (DRIVE INTERFACE OFF).
/
0425 4454      TEST2,  PAT25      /GET FIRST PATTERN INTO AC

```

```

0426 4263      JMS      CACHK      /CHECK COMMAND A WITH PATTERN
0427 4450      SCOPE

0430 4455      PAT52
0431 4263      JMS      CACHK      /GET ALTERNATE PATTERN
0432 4450      SCOPE      /CHECK CA WITH PATTERN

/SIMILAR TEST FOR COMMAND B REGISTER
/
0433 4454      PAT25      /GET PATTERN
0434 4370      JMS      SAFECB     /MAKE FUNCTION=1 AND DRIVE INTERFACE OFF
0435 4311      JMS      CBCHK     /CHECK CB WITH PATTERN
0436 4450      SCOPE

0437 4455      PAT52      /GET 2ND PATTERN
0440 4370      JMS      SAFECB     /MAKE FUNCTION=1 AND DRIVE INTERFACE OFF
0441 4311      JMS      CBCHK     /CHECK CB WITH PATTERN
0442 4450      SCOPE

/SIMILAR TEST FOR SECTOR ADDRESS REGISTER
/
0443 4454      PAT25      /GET FIRST PATTERN
0444 7002      BSW        /GET LSB OF PATTERN INTO HIGH BYTE
0445 0135      AND        K7700    /MASK OUT UNUSED BITS
0446 3124      DCA        PATTRN   /SAVE CORRECT PATTERN
0447 1124      TAD        PATTRN
0450 4342      JMS      SACHK     /CHECK SA WITH PATTERN
0451 4450      SCOPE

0452 4455      PAT52      /GET SECOND PATTERN
0453 7002      BSW        /GET LSB OF PATTERN INTO HIGH BYTE
0454 0135      AND        K7700    /MASK OUT UNUSED BITS
0455 3124      DCA        PATTRN   /SAVE CORRECT PATTERN
0456 1124      TAD        PATTRN
0457 4342      JMS      SACHK     /CHECK SA WITH PATTERN
0460 4450      SCOPE
0461 5662      JMP I      .+1
0462 0626      T2WC

/THIS SUBROUTINE LOADS THE CA REGISTER WITH AC CONTENTS WHEN CALLED,
/VERIFIES THAT THE AC WAS CLEARED, AND VERIFIES THAT THE REGISTER READS
/BACK THE DATA THAT WAS WRITTEN.  FINALLY, IT CHECKS THAT AN RLDC IOT
/CLEAR THE REGISTER.
/
/IF AT ANY POINT A FAILURE IS DETECTED, THE REMAINDER OF THE TEST IS SKIPPED.
/
0463 0000      CACHK, 0
0464 4427      RLCA        /LOAD REGISTER WITH PATTERN
0465 7450      SNA        /SKIP IF AC NOT CLEARED
0466 5273      JMP        .+5      /ALL OK--CONTINUE
0467 3113      DCA        DATA1   /SAVE BAD AC FOR TYPEOUT
0470 4451      ERROR      /AC NOT CLEARED BY RLCA
0471 0203      CA2A       /PC ACTUAL
0472 5310      JMP        CARET    /SKIP REMAINDER OF CHECKS
0473 4435      RRCA        /READ BACK REGISTER

```

0474	4456	PATCMP		/COMPARE WITH VALUE IN "PATTERN"
0475	5301	JMP	+.4	/COMPARED EQUAL--SKIP ERROR CALL
0476	4451	ERROR		/CA DID NOT CONTAIN EXPECTED DATA
0477	0206	CA2B		/PC EXPCTD ACTUAL
0500	5310	JMP	CARET	/SKIP REST OF TEST
0501	4424	RLDC		/CLEAR DEVICE
0502	4435	RRCA		/READ REGISTER
0503	7450	SNA		/WAS IT CLEARED??
0504	5310	JMP	CARET	/YES-GO RETURN
0505	3113	DCA	DATA1	/SAVE BAD REGISTER CONTENTS
0506	4451	ERROR		/CA NOT CLEARED BY RLDC
0507	0211	CA2C		/PC ACTUAL
0510	5663	CARET, JMP I	CACHK	/RETURN W/ AC CLEAR

/THIS SUBROUTINE IS IDENTICLE TO CACHK, EXCEPT IT CHECKS THE COMMAND B REG

0511	0000	CBCHK, 0		
0512	4430	RLCB		/LOAD REGISTER WITH PATTERN
0513	7450	SNA		/SKIP IF AC NOT CLEARED
0514	5321	JMP	+.5	/ALL OK--CONTINUE
0515	3113	DCA	DATA1	/SAVE BAD AC FOR TYPEOUT
0516	4451	ERROR		/AC NOT CLEARED BY RLCB
0517	0214	CB2A		/PC ACTUAL
0520	5340	JMP	CBRET	/SKIP REMAINDER OF CHECKS
0521	4436	RRCB		/READ BACK REGISTER
0522	4456	PATCMP		/COMPARE WITH VALUE IN "PATTERN"
0523	5330	JMP	+.5	/COMPARED EQUAL--SKIP ERROR CALL
0524	4444	STOPF		/TERMINATE POSSIBLE ADVERSE ACTIONS
0525	4451	ERROR		/CB DID NOT CONTAIN EXPECTED DATA
0526	0217	CB2B		/PC EXPCTD ACTUAL
0527	5340	JMP	CBRET	/SKIP REST OF TEST
0530	4464	WAIT		
0531	4424	RLDC		/CLEAR DEVICE
0532	4436	RRCB		/READ REGISTER
0533	7450	SNA		/WAS IT CLEARED??
0534	5340	JMP	CBRET	/YES-GO RETURN
0535	3113	DCA	DATA1	/SAVE BAD REGISTER CONTENTS
0536	4451	ERROR		/CB NOT CLEARED BY RLDC
0537	0222	CB2C		/PC ACTUAL
0540	4444	CBRET, STOPF		
0541	5711	JMP I	CBCHK	/RETURN W/ AC CLEAR

/THIS SUBROUTINE IS IDENTICLE TO CACHK, EXCEPT IT CHECKS THE SECTOR ADDRESS REG

0542	0000	SACHK, 0		
0543	4431	RLSA		/LOAD REGISTER WITH PATTERN
0544	7450	SNA		/SKIP IF AC NOT CLEARED
0545	5352	JMP	+.5	/ALL OK--CONTINUE
0546	3113	DCA	DATA1	/SAVE BAD AC FOR TYPEOUT
0547	4451	ERROR		/AC NOT CLEARED BY RLSA
0550	0225	SA2A		/PC ACTUAL
0551	5367	JMP	SARET	/SKIP REMAINDER OF CHECKS
0552	4437	RRSA		/READ BACK REGISTER
0553	4456	PATCMP		/COMPARE WITH VALUE IN "PATTERN"
0554	5360	JMP	+.4	/COMPARED EQUAL--SKIP ERROR CALL

```

0555 4451      ERROR      /SA DID NOT CONTAIN EXPECTED DATA
0556 0230      SA2B       /PC      EXPCTD  ACTUAL
0557 5367      JMP        SARET    /SKIP REST OF TEST
0560 4424      RLDC       /CLEAR DEVICE
0561 4437      RRSA       /READ REGISTER
0562 7450      SNA        /WAS IT CLEARED??
0563 5367      JMP        SARET    /YES-GO RETURN
0564 3113      DCA        DATA1   /SAVE BAD REGISTER CONTENTS
0565 4451      ERROR      /SA NOT CLEARED BY RLDC
0566 0233      SA2C       /PC      ACTUAL
0567 5742      SARET,    JMP I     SACHK  /RETURN W/ AC CLEAR

0570 0000      SAFECB, 0
0571 0137      AND        K3770    /MASK OUT FUNCTION AND DRIVE INTERFACE ENABLE BITS
0572 1161      TAD        K4301    /MAKE FUNCTION 1 (RESET) -- DISABLE INTERFACE
0573 3124      DCA        PATRN    /SAVE NEW CORRECT PATTERN
0574 1124      TAD        PATRN
0575 5770      JMP I     SAFECB    /RETURN WITH SAFE PATTERN IN AC

```

0600 PAGE

/THIS SUBROUTINE IS IDENTICLE TO CACHK, EXCEPT IT CHECKS THE WORD COUNT REG

```

0600 0000      WCCHK, 0
0601 4432      RLWC       /LOAD REGISTER WITH PATTERN
0602 7450      SNA        /SKIP IF AC NOT CLEARED
0603 5210      JMP        .+5      /ALL OK--CONTINUE
0604 3113      DCA        DATA1   /SAVE BAD AC FOR TYPEOUT
0605 4451      ERROR      /AC NOT CLEARED BY RLWC
0606 0236      WC2A       /PC      ACTUAL
0607 5225      JMP        WCRET    /SKIP REMAINDER OF CHECKS
0610 4434      RRWC       /READ BACK REGISTER
0611 4456      PATCMP     /COMPARE WITH VALUE IN "PATTERN"
0612 5216      JMP        .+4      /COMPARED EQUAL--SKIP ERROR CALL
0613 4451      ERROR      /WC DID NOT CONTAIN EXPECTED DATA
0614 0241      WC2B       /PC      EXPCTD  ACTUAL
0615 5225      JMP        WCRET    /SKIP REST OF TEST
0616 4424      RLDC       /CLEAR DEVICE
0617 4434      RRWC       /READ REGISTER
0620 7450      SNA        /WAS IT CLEARED??
0621 5225      JMP        WCRET    /YES-GO RETURN
0622 3113      DCA        DATA1   /SAVE BAD REGISTER CONTENTS
0623 4451      ERROR      /WC NOT CLEARED BY RLDC
0624 0244      WC2C       /PC      ACTUAL
0625 5600      WCRET,    JMP I     WCCHK  /RETURN W/ AC CLEAR

```

/SIMILAR TEST FOR WORD COUNT REGISTER

```

0626 4454      T2WC,    PAT25      /TEST WORD COUNT REGISTER WITH
0627 4200      JMS      WCCHK      /FIRST PATTERN
0630 4450      SCOPE

0631 4455      PAT52      /TEST WORD COUNT REGISTER WITH
0632 4200      JMS      WCCHK      /SECOND PATTERN
0633 4450      SCOPE

```

/VERIFY AC CLEARED BY OTHER IOTS.
 /NOTE: CURRENTLY, MA IS THE ONLY WRITE-ONLY REGISTER.
 /LOAD AC (ALL 1'S FIRST PASS--ALL COMBINATIONS THEREAFTER).
 /WRITE TO WRITE ONLY REGISTERS AND VERIFY AC WAS CLEARED. ALSO
 /CHECK AC IS CLEARED BY "SPARE" IOT AND RLDC.
 /

0634	4466	TEST3, PATM1	/GET PATTERN INTO AC BASED ON MINUS 1
0635	4426	RLMA	/WRITE TO MA REG
0636	7450	SNA	/SKIP IF AC WAS NOT CLEARED
0637	5243	JMP .+4	/OK--GO TO SCOPE
0640	3113	DCA DATA1	/SAVE BAD AC
0641	4451	ERROR	/AC NOT CLEARED BY RLMA
0642	0247	MA3	/PC BAD
0643	4450	SCOPE	

/REPEAT SIMILAR TEST FOR "SPARE" IOT

0644	4466	PATM1	/GET PATTERN INTO AC BASED ON MINUS 1
0645	4443	SPARE	/ISSUE "SPARE" IOT
0646	7450	SNA	/SKIP IF AC WAS NOT CLEARED
0647	5253	JMP .+4	/OK--GO TO SCOPE
0650	3113	DCA DATA1	/SAVE BAD AC
0651	4451	ERROR	/AC NOT CLEARED BY "SPARE" IOT
0652	0255	SPARE3	/PC BAD
0653	4450	SCOPE	

/VERIFY AC CLEARED BY RLDC

0654	4466	PATM1	/GET PATTERN INTO AC BASED ON MINUS 1
0655	4424	RLDC	/ISSUE DEVICE CLEAR
0656	7450	SNA	/SKIP IF AC WAS NOT CLEARED
0657	5263	JMP .+4	/OK--EXIT TEST
0660	3113	DCA DATA1	/SAVE BAD AC
0661	4451	ERROR	/AC NOT CLEARED BY RLDC
0662	0260	DC3	/PC BAD
0663	4450	SCOPE	

/VERIFY NO INTERACTION BETWEEN IOTS.

/LOAD CB WITH A TEST PATTERN. ISSUE RLCA, RLSA, AND RRSA AND
 /VERIFY CB IS UNCHANGED. LOAD WC AND SA WITH ALTERNATE PATTERNS
 /AND ISSUE SPARE AND RRS1 AND VERIFY WC AND SA ARE UNCHANGED.
 /

0664	4454	TEST4, PAT25	/GET 1ST PATTERN
0665	4732	JMS I PSAFECB	/MAKE FUNCTION=1 AND DRIVE INTERFACE OFF
0666	4430	RLCB	/WRITE PATTERN TO CB REGISTER
0667	4427	RLCA	/WRITE TO CA
0670	4431	RLSA	/WRITE SA
0671	4437	RRSA	/READ SA
0672	4436	RRCB	/NOW READ BACK CB
0673	3125	DCA TEMP1	/SAVE AC
0674	4444	STOPF	/CLEAR DEVICE IN CASE CB CHANGED
0675	1125	TAD TEMP1	/RESTORE AC
0676	4456	PATCMP	/COMPARE DATA READ WITH DATA WRITTEN
0677	5302	JMP .+3	/OK--GO TO SCOPE
0700	4451	ERROR	/CB AFFECTED BY RLCA,RLSA OR RRSA


```

0701 0263          CB4          /PC GOOD BAD
0702 4450          SCOPE

0703 4454          PAT25        /GET FIRST PATTERN
0704 4431          RLSA         /LOAD SECTOR ADDRESS
0705 4455          PAT52        /GET ALTERNATE PATTERN
0706 4432          RLWC         /LOAD WORD COUNT
0707 4443          SPARE        /ISSUE SPARE
0710 4440          RRSI         /ISSUE RRSI
0711 4434          RRWC         /READ BACK WORD COUNT
0712 4456          PATCMP       /COMPARE TO PATTERN WRITTEN
0713 5317          JMP          .+4 /OK--CONTINUE
0714 4451          ERROR        /WC AFFECTED BY "SPARE" OR RRSI
0715 0266          WC4          /PC GOOD BAD
0716 5327          JMP          .+11 /EXIT TEST
0717 4454          PAT25        /GET PATTERN THAT WAS WRITTEN TO SA
0720 0135          AND          K7700 /MASK BOTTOM BITS
0721 3124          DCA          PATTRN /SAVE EXPECTED PATTERN
0722 4437          RRSa         /GET SA REG
0723 4456          PATCMP       /COMPARE SA TO EXPECTED SA
0724 5327          JMP          .+3 /OK--EXIT TEST
0725 4451          ERROR        /SA AFFECTED BY RLWC, SPARE, OR RRSI
0726 0271          SA4          /PC GOOD BAD
0727 4450          SCOPE
0730 5731          JMP I       .+1
0731 1000          TEST6

0732 0570          PSAFECB, SAFECB

1000          PAGE

/VERIFY DONE BIT SETS AND CLEARS.
/ISSUE RESET COMMAND AND WAIT FOR DONE TO SET. CHECK THAT RLSD SKIPS.
/CHECK THAT SECOND RLSD DOES NOT SKIP.
/
1000 4460          TEST6, RESETW /ISSUE RESET AND WAIT LONG ENOUGH FOR DONE
1001 4425          RLSD          /SKIP IF DONE FLAG IS SET
1002 5211          JMP          ERR6 /GO REPORT ERROR
1003 4471          ERRCHK        /CHECK FOR ERROR FLAG AND REPORT
1004 4425          RLSD          /DONE FLAG SHOULD HAVE BEEN CLEARED BY IOT
1005 5216          JMP          END6 /OK--EXIT TEST
1006 4451          ERROR        /DONE NOT CLEARED BY RLSD
1007 0274          SD5B          /PC
1010 5216          JMP          END6
1011 4436          ERR6, RRCB     /SAVE CB FOR ERROR TYPEOUT
1012 3113          DCA          DATA1
1013 4444          STOPF         /CLEAR DEVICE
1014 4451          ERROR        /DONE NOT SET
1015 0277          NODONE       /PC CB
1016 4450          END6, SCOPE

/VERIFY RLDC CLEARS DONE BIT.
/ISSUE RESET COMMAND AND WAIT FOR DONE. ISSUE RLDC AND CHECK THAT RLSD
/DOES NOT SKIP.
/

```

1017	4460	TEST7,	RESETW		/ISSUE RESET AND WAIT FOR DONE
1020	4424		RLDC		/CLEAR DEVICE
1021	4425		RLSD		/SKIP ON DONE
1022	5225		JMP	..+3	/OK--DONE WAS CLEARED--GO TO SCOPE
1023	4451		ERROR		/DONE NOT CLEARED BY RLDC
1024	0302		SD6		/PC
1025	4450		SCOPE		
1026	5627		JMP I	..+1	
1027	1200		TEST8		

1200 PAGE

/VERIFY LOADING CB CLEARS DONE.
 /ISSUE RESET COMMAND AND WAIT FOR DONE. ISSUE A RESET AND CHECK
 /THAT RLSD DOES NOT SKIP.

1200	4460	TEST8,	RESETW		/ISSUE A RESET AND WAIT FOR DONE
1201	4777'		JMS	VTCHK	/CHECK IF VT278
1202	7410		SKP		/SKIP IF NOT A VT278
1203	4510		SCNINT		/TURN OFF INTRPT CRTC
1204	1141		TAD	K7301	/SET AC = 7301
1205	4430		RLCB		/EXEC RESET CMD
1206	4425		RLSD		/SKIP ON FUNCTION DONE
1207	5217		JMP	ZEND8	/ALL O.K.---CONTINUE
1210	4777'		JMS	VTCHK	/CHECK IF VT278
1211	7410		SKP		/SKIP IF NOT VT278
1212	4511		SETINT		/TURN ON INTRPT CRTC
1213	4451		ERROR		/DONE NOT CLEARED BY LOAD CB
1214	0305		SD7		/PC
1215	4444	END8,	STOPF		
1216	5222		JMP	..+4	/ALL OK--CONTINUE
1217	4777'	ZEND8,	JMS	VTCHK	/CHECK IF VT278
1220	7410		SKP		/SKIP IF NOT VT278
1221	4511		SETINT		/TURN OFF INTRPT CRTC
1222	4450		SCOPE		

/VERIFY INTERRUPT REQUEST GENERATED BY DONE.
 /CLEAR DEVICE. VERIFY NO INTERRUPTS ARE ALREADY PENDING. ISSUE
 /"RESET" WITH INTERRUPT ENABLE SET. VERIFY INTERRUPT REQUEST IS
 /ASSERTED ON THE BUS. CLEAR DONE AND VERIFY INTERRUPT REQUEST
 /IS RESET.

1223	6003	TEST9,	SRQ		/SKIP IF INTERRUPTS PENDING
1224	5230		JMP	..+4	/CONTINUE TEST BELOW ERROR CALL
1225	4451	T9E,	ERROR		/UNKNOWN INTERRUPT REQUESTED
1226	0310		UN8		/PC
1227	5255		JMP	END10	/GO TO SCOPE CALL AT END OF TEST10
1230	1177		TAD	K7701	/ABORT INTERRUPT TESTS
1231	4430		RLCB		/ISSUE RESET W/IE SET
1232	4464		WAIT		/WAIT FOR DONE TO SET
1233	6003		SRQ		/CHECK FOR INTERRUPT REQUEST
1234	7410		SKP		/INTERRUPT NOT REQUESTED--SKIP
1235	5241		JMP	..+4	/INTR REQUESTED--SKIP ERROR CALL
1236	4451		ERROR		/INTERRUPT REQUEST NOT GENERATED BY DONE

```

1237 0313      INT8A      /PC
1240 5247      JMP        .+7      /EXIT TEST
1241 4425      RLSD       /SKIP ON DONE TO CLEAR DONE FLAG
1242 5225      JMP        T9E      /HOW DID I GET HERE? REPORT UNKNOWN
                                   /INTERRUPT ERROR
1243 6003      SRQ        /SKIP IF INTERRUPT REQUESTED
1244 5247      JMP        .+3
1245 4451      ERROR      /INTERRUPT STILL REQUESTED AFTER DONE CLEARED
1246 0316      INT8B      /PC
1247 4450      SCOPE

```

```

/VERIFY INTERRUPT NOT REQUESTED IF INTERRUPT ENABLE NOT SET.
/ISSUE RESET WITHOUT INTERRUPT ENABLE. VERIFY NO INTERRUPT
/REQUEST IS GENERATED.
/

```

```

1250 4460      TEST10, RESETW
1251 6003      SRQ        /SKIP IF INTERRUPT REQUESTED
1252 5255      JMP        .+3      /OK--SKIP ERROR CALL
1253 4451      ERROR      /INTERRUPT REQUESTED WHEN IE NOT SET
1254 0321      INT9       /PC
1255 4450      END10, SCOPE

```

```

/VERIFY SILO CAN BE CLEARED BY RLDC.
/SET UP TO LOAD 1'S INTO FORCABLE BITS OF DISK ADDRESS REGISTER.
/LOAD CB WITH RESET AND MAINTENANCE BIT 1 SET (DAR TO SILO).
/ISSUE RLDC. WRITE 0'S TO DAR AND ISSUE RESET. VERIFY 1ST TRANSFER
/IS NOT THERE.
/

```

```

1256 4466      TEST11, PATM1      /GET PATTERN INTO AC BASED ON MINUS 1
1257 4427      RLCA      /WRITE TO COMMAND A
1260 4466      PATM1      /GET PATTERN INTO AC BASED ON MINUS 1
1261 4431      RLCA
1262 4465      DAREXP      /COMPUTE EXPECTED SILO VALUES
1263 0136      AND        K364      /MASK RESET AND MARKER BITS
1264 1160      TAD        K13      /ADD IN RESET AND MARKER BITS
1265 3114      DCA        DATA2    /SAVE EXPECTED VALUE
1266 4457      RESET      /ISSUE MAINTENANCE RESET AND WAIT FOR DONE
1267 4471      ERRCHK      /CHECK FOR ERROR FLAG AND REPORT
1270 5303      JMP        END11
1271 4424      RLDC      /ISSUE DEVICE CLEAR
1272 4457      RESET      /DO A TRANSFER TO SILO
1273 4467      XFRCHK      /CHECK IF OLD TRANSFER IS STILL THERE
1274 7410      SKP        /STILL THERE--SKIP
1275 5303      JMP        END11      /OLD TRANSFER GONE--EXIT TEST
1276 4465      DAREXP      /CHECK IF OLD TRANSFER = NEW TRANSFER
1277 7650      SNA CLA      /SKIP IF SHOULDN'T BE EQUAL
1300 5303      JMP        END11      /EQUAL SO OK--EXIT TEST
1301 4451      ERROR      /SILO NOT CLEARED BY RLDC
1302 0324      SIERR      /PC
1303 4450      END11, SCOPE

```

```

/VERIFY MAINTENANCE DISK ADDRESS REGISTER TO SILO TRANSFER OK.
/CLEAR DEVICE. SET UP FOR DESIRED DAR LOADING (ALTERNATE 1'S AND
/0'S IN FORCABLE BITS) AND LOAD CB WITH "RESET" WITH MAINT BIT 1
/SET. READ SILO AND VERIFY IT IS AS EXPECTED. REPEAT FOR

```

```

/ALTERNATE PATTERN.
/
1304 4454 TEST12, PAT25 /GET PATTERN TO PUT IN CA FOR DAR LOADING
1305 4427 RLCA /PUT PATTERN INTO CA
1306 1124 TAD PATRN /GET PATTERN AGAIN
1307 4431 RLCA /ALSO LOAD SECTOR ADDRESS WITH PATTERN
1310 4465 DAREXP /FORMULATE PORTION OF EXPECTED DAR WORDS
1311 0136 AND K364 /MASK OUT MARKER AND RESET BITS
1312 1160 TAD K13 /ADD IN MARKER AND RESET BIT TO WORD
/FORMED SO FAR
1313 3114 DCA DATA2 /SAVE FIRST 8 BIT WORD EXPECTED
/IN DAR
1314 4457 RESET /ISSUE MAINT RESET
1315 4467 XFRCHK /CHECK IF TRANSFER WAS OK
1316 5323 JMP .+5 /TRANSFER OK--CALL SCOPE
1317 4436 RRCB /PICK UP COMMAND B
1320 3117 DCA DATA5 /SAVE FOR ERROR TYPEOUT
1321 4451 ERROR /DAR TO SILO TRANSFER (MAINT MODE) FAILED
1322 0327 DAR11 /PC H18-EXPCTD-LO8 H18-ACTUAL-LO8 CB
1323 4450 SCOPE

/REPEAT TEST WITH ALTERNATE PATTERN
1324 4455 PAT52 /GET PATTERN TO PUT IN CA FOR DAR LOADING
1325 4427 RLCA /PUT PATTERN INTO CA
1326 1124 TAD PATRN /GET PATTERN AGAIN
1327 4431 RLCA /ALSO LOAD SECTOR ADDRESS WITH PATTERN
1330 4465 DAREXP /FORMULATE PORTION OF EXPECTED DAR WORDS
1331 0136 AND K364 /MASK OUT MARKER AND RESET BITS
1332 1160 TAD K13 /ADD IN MARKER AND RESET BIT TO WORD
/FORMED SO FAR
1333 3114 DCA DATA2 /SAVE FIRST 8 BIT WORD EXPECTED
/IN DAR
1334 4457 RESET /ISSUE THE DAR TO SILO RESET
1335 4467 XFRCHK /CHECK IF TRANSFER WAS OK
1336 5343 JMP END12 /TRANSFER OK--CALL SCOPE
1337 4436 RRCB /PICK UP COMMAND B
1340 3117 DCA DATA5 /SAVE FOR ERROR TYPEOUT
1341 4451 ERROR /DAR TO SILO TRANSFER (MAINT MODE) FAILED
1342 0327 DAR11 /PC H18-EXPCTD-LO8 H18-ACTUAL-LO8 CB
1343 4450 END12, SCOPE
1344 5745 JMP I .+1
1345 1400 TEST13

1377 2106 PAGE
1400 1400 /VERIFY SILO CAN BE CLEARED BY CLK OPI DLY.
/
1400 4466 TEST13, PATM1 /GET PATTERN INTO AC BASED ON MINUS 1
1401 4427 RLCA /WRITE TO COMMAND A
1402 4466 PATM1 /GET PATTERN INTO AC BASED ON MINUS 1
1403 4431 RLCA
1404 4465 DAREXP /COMPUTE EXPECTED SILO VALUES
1405 0136 AND K364 /MASK RESET AND MARKER BITS
1406 1160 TAD K13 /ADD IN RESET AND MARKER BITS
1407 3114 DCA DATA2 /SAVE EXPECTED VALUE

```

```

1410 4457      RESET
1411 1161      TAD      K4301
1412 4430      RLCB
1413 4425      RLSD
1414 4473      JMPPM1
1415 4427      RLCA
1416 4431      RLCA
1417 4457      RESET
1420 4467      XFRCHK
1421 7410      SKP
1422 5232      JMP      END13
1423 4465      DAREXP
1424 7650      SNA CLA
1425 5232      JMP      END13
1426 4436      RRCB
1427 3113      DCA      DATA1
1430 4451      ERROR
1431 3253      CLOPDL
1432 4450      END13, SCOPE

/ISSUE MAINTENANCE RESET AND WAIT FOR DONE
/ISSUE NON-MAINT RESET

/WAIT FOR DONE

/CLEAR OUT CA AND SA REGISTERS

/DO A TRANSFER TO SILO
/CHECK IF OLD TRANSFER IS STILL THERE
/STILL THERE--SKIP
/OLD TRANSFER GONE--EXIT TEST
/CHECK IF OLD TRANSFER = NEW TRANSFER
/SKIP IF SHOULDN'T BE EQUAL
/EQUAL SO OK--EXIT TEST

/SAVE COMMAND B
/CLK OPI DLY WAS NOT ASSERTED TO SILO (SILO NOT CLEARED)
/PC      CB

/VERIFY GET STATUS DAR TO SILO TRANSFER; CA LOADS PROPERLY.
/CLEAR DEVICE.  ISSUE MAINTENANCE "GET STATUS" AND VERIFY
/TRANSFER TO SILO WAS OK.  (SA CLEAR ON THIS TEST.)
/

1433 4454      TEST14, PAT25
1434 4427      RLCA
1435 4465      DAREXP
1436 1153      TAD      K3

/GET TEST PATTERN
/WRITE PATTERN TO CA
/COMPUTE MOST OF EXPECTED DAR WORDS
/SET GET STATUS AND MARKER BITS (NO
/CHANCE OF CARRY SINCE SA SHOULD BE CLEAR)
/SAVE EXPECTED WORD
/GET MAINT GET STATUS VALUE
/SAVE FOR COMPARISON

1437 3114      DCA      DATA2
1440 1133      TAD      K7302
1441 3124      DCA      PATTRN
1442 1124      TAD      PATTRN
1443 4430      RLCB
1444 4436      RRCB
1445 4456      PATCMP
1446 5253      JMP      .+5
1447 4444      STOPF
1450 4451      ERROR
1451 0217      CB2B
1452 5275      JMP      END14
1453 4464      WAIT
1454 4464      WAIT
1455 4471      ERRCHK
1456 5275      JMP      END14
1457 4425      RLSD
1460 5270      JMP      NODN14
1461 4467      XFRCHK
1462 5275      JMP      END14
1463 4436      RRCB
1464 3117      DCA      DATA5
1465 4451      ERROR
1466 0327      DAR11
1467 5275      JMP      END14
1470 4436      NODN14, RRCB

/ISSUE MAINT GET STATUS
/READ BACK CB
/VERIFY ITS VERACITY
/ALL OK--CONTINUE TEST
/CLEAR DEVICE
/CB DID NOT CONTAIN EXPECTED DATA
/PC      EXPCTD  ACTUAL
/EXIT TEST
/WAIT LONG ENOUGH FOR TRANSFER

/CHECK FOR ERROR FLAG AND REPORT--SKIP IF NO ERROR
/EXIT TEST
/SKIP IF DONE SET
/DONE NOT SET--REPORT ERROR
/CHECK TRANSFER TO SILO
/TRANSFER OK--EXIT TEST
/PICK UP COMMAND B
/SAVE FOR ERROR TIMEOUT
/DAR TO SILO TRANSFER (MAINT MODE) FAILED
/PC      H18-EXPCTD-LOB  H18-ACTUAL-LOB      CB
/EXIT TEST
/READ COMMAND B

```

```

1471 3113          DCA      DATA1      /SAVE FOR ERROR TYPEOUT
1472 4444          STOPF          /CLEAR DEVICE
1473 4451          ERROR          /DONE DID NOT SET
1474 0277          NODONE          /PC      CB
1475 4450          END14, SCOPE

/DO GET STATUS TO DRIVE 0 WITH INTERFACE DISABLED (MAINT BIT 0
/SET) AND VERIFY OPI SETS.
/IF A RESPONSE WAS RECEIVED FROM THE DRIVE, PREVENT PROGRAM FROM
/CONTINUING.
/

1476 1134          TEST15, TAD      K5302      /ISSUE GET STATUS TO DRIVE 0 WITH
1477 4430          RLCB          /DRIVE INTERFACE DISABLED
1500 4463          OPIWAT          /WAIT FOR DONE AND OPI
1501 5305          JMP      .+4      /NEITHER DONE NOR OPI
1502 5313          JMP      NODN15    /OPI BUT NO DONE
1503 5323          JMP      NOPI15    /DONE BUT NO OPI
1504 5332          JMP      END15     /OPI AND DONE--EXIT TEST
1505 4436          RRCB          /READ COMMAND B
1506 3113          DCA      DATA1      /SAVE FOR ERROR TYPEOUT
1507 4444          STOPF          /CLEAR DEVICE
1510 4451          ERROR          /OPI AND DONE DID NOT SET
1511 2075          NDNNOPI          /PC      CB
1512 5332          JMP      END15     /EXIT TEST
1513 4433          NODN15, RRER          /READ ERROR REGISTER
1514 3113          DCA      DATA1      /SAVE FOR ERROR TYPEOUT
1515 4436          RRCB          /READ COMMAND B
1516 3114          DCA      DATA2      /SAVE FOR MESSAGE
1517 4444          STOPF          /CLEAR DEVICE
1520 4451          ERROR          /DONE NOT SET AFTER OPI
1521 2116          OPINDN          /PC      ER      CB
1522 5332          JMP      END15     /EXIT TEST
1523 4471          NOPI15, ERRCHK          /REPORT ERROR IF ERROR FLAG SET
1524 5332          JMP      END15     /EXIT TEST
1525 4451          ERROR          /DRIVE INTERFACE NOT DISABLED (OPI NOT SET ON GET STATUS)
1526 2136          DRINT          /PC
1527 4452          CONSOL          /ALLOW CONSOL INPUT
1530 4424          RLDC
1531 5276          JMP      TEST15     /LOCK PERMANENTLY ON TEST
1532 7240          END15, CLA CMA      /-1 INTO AC
1533 3121          DCA      LUPCNT      /INHIBIT LOOPING ON THIS TEST
1534 4450          SCOPE

/VERIFY PROPER LOADING OF CA INTO DAR.
/CLEAR SA AND USE ALTERNATE PATTERN FROM ONE USED IN TEST 14 TO
/LOAD CA.  ISSUE MAINTENANCE RESET AND VERIFY TRANSFER TO SILO
/WAS OK.
/

1535 4455          TEST16, PAT52      /GET TEST PATTERN
1536 4427          RLCA          /WRITE PATTERN TO CA
1537 4465          DAREXP          /COMPUTE MOST OF EXPECTED DAR WORDS
1540 1160          TAD      K13      /ADD IN MARKER AND RESET BITS
1541 3114          DCA      DATA2      /SAVE EXPECTED WORD
1542 4457          RESET          /INITIATE TRANSFER
1543 4471          ERRCHK          /CHECK FOR ERROR FLAG AND REPORT--SKIP IF NO ERROR

```

```

1544 5353      JMP      END16      /EXIT TEST
1545 4467      XFRCHK     /CHECK TRANSFER TO SILO
1546 5353      JMP      END16      /TRANSFER OK--EXIT TEST
1547 4436      RRCB       /PICK UP COMMAND B
1550 3117      DCA        DATA5  /SAVE FOR ERROR TYPEOUT
1551 4451      ERROR      /DAR TO SILO TRANSFER (MAINT MODE) FAILED
1552 0327      DAR11      /PC      HI8-EXPCTD-LO8 HI8-ACTUAL-LO8 CB
1553 4450      END16, SCOPE
1554 5755      JMP I      .+1      /NEXT TEST ON NEXT PAGE
1555 1600      TEST17

```

1600 PAGE

/VERIFY PROPER DAR LOADING OF SA.
 /CLEAR DEVICE. LOAD TEST PATTERN INTO SA. ISSUE GET STATUS
 /WITH MAINT BIT 1 SET. VERIFY TRANSFER TO SILO OK. REPEAT FOR
 /ALTERNATE PATTERN IN SA.

```

1600 4454      TEST17, PAT25      /GET TEST PATTERN
1601 4431      RLSA       /WRITE PATTERN TO SA
1602 4465      DAREXP     /COMPUTE MOST OF EXPECTED DAR WORDS
1603 0164      AND        K7774  /MASK OUT MARKER AND GET STATUS BITS
1604 1153      TAD        K3     /ADD THEM IN
1605 3114      DCA        DATA2 /SAVE EXPECTED WORD
1606 1133      TAD        K7302  /PICK UP MAINT (DAR TO SILO) GET STATUS
1607 4430      RLCB       /ISSUE COMMAND
1610 4425      RLSD      /WAIT FOR DONE FLAG
1611 4473      JMPPM1
1612 4467      XFRCHK     /CHECK TRANSFER TO SILO
1613 5220      JMP      END17A   /TRANSFER OK--EXIT TEST
1614 4436      RRCB       /PICK UP COMMAND B
1615 3117      DCA        DATA5 /SAVE FOR ERROR TYPEOUT
1616 4451      ERROR      /DAR TO SILO TRANSFER (MAINT MODE) FAILED
1617 0327      DAR11      /PC      HI8-EXPCTD-LO8 HI8-ACTUAL-LO8 CB
1620 4450      END17A, SCOPE

```

/TEST 17B--REPEAT TEST 17A FOR ALTERNATE PATTERN IN SA

```

1621 4455      PAT52      /GET TEST PATTERN
1622 4431      RLSA       /WRITE PATTERN TO SA
1623 4465      DAREXP     /COMPUTE MOST OF EXPECTED DAR WORDS
1624 0164      AND        K7774  /MASK MARKER AND GET STATUS BITS
1625 1153      TAD        K3     /ADD THEM IN
1626 3114      DCA        DATA2 /SAVE EXPECTED WORD
1627 1133      TAD        K7302  /PICK UP MAINT (DAR TO SILO) GET STATUS
1630 4430      RLCB       /ISSUE COMMAND
1631 4425      RLSD      /WAIT FOR DONE FLAG
1632 4473      JMPPM1
1633 4467      XFRCHK     /CHECK TRANSFER TO SILO
1634 5241      JMP      END17B   /TRANSFER OK--EXIT TEST
1635 4436      RRCB       /PICK UP COMMAND B
1636 3117      DCA        DATA5 /SAVE FOR ERROR TYPEOUT
1637 4451      ERROR      /DAR TO SILO TRANSFER (MAINT MODE) FAILED
1640 0327      DAR11      /PC      HI8-EXPCTD-LO8 HI8-ACTUAL-LO8 CB
1641 4450      END17B, SCOPE

```

/VERIFY 12 BIT DAR TO SILO TRANSFER

```

/
1642 4454 TEST18, PAT25 /WRITE PATTERN TO COMMAND A
1643 4427 RLCA
1644 1124 TAD PATRN /WRITE SAME PATTERN TO SA
1645 4431 RLCA
1646 1124 TAD PATRN
1647 7002 BSW /GET CYLINDER ADDRESS DIFFERENCE INTO POSITION
1650 7004 RAL
1651 0154 AND K7600 /MASK OUT ALL BUT CYLINDER ADDRESS DIFF BITS DESIRED
1652 3114 DCA DATA2 /SAVE DAR WORD EXPECTED SO FAR
1653 4437 RRSA /GET SECTOR ADDRESS REG
1654 7002 BSW /PUT INTO POSITION
1655 1114 TAD DATA2 /ADD IN PREVIOUSLY FORMED BITS
1656 3114 DCA DATA2 /SAVE WORD FORMED SO FAR
1657 4435 RRCA
1660 7004 RAL /GET CAO INTO LINK, CA1 INTO SIGN BIT
1661 7710 SPA CLA /SKIP IF CA1 WAS CLEAR
1662 1140 TAD K100 /ADD IN CA1 BIT
1663 1114 TAD DATA2 /ADD IN PORTION OF WORD ALREADY FORMED
1664 7420 SNL /SKIP IF CAO WAS SET
1665 5270 JMP .+3 /DON'T ADD IN CAO BIT
1666 0371 AND K7773A /MASK OUT CAO BIT
1667 1157 TAD K4 /ADD IN CAO BIT
1670 0372 AND K7764 /MASK OUT MARKER AND RESET BITS
1671 1160 TAD K13 /SET MARKER AND RESET BITS
1672 3114 DCA DATA2 /SAVE EXPECTED WORD
1673 1114 TAD DATA2 /PICK UP EXPECTED GARBAGE
1674 3113 DCA DATA1 /SAVE EXPECTED WORD OF GARBAGE
1675 1162 TAD K6301
1676 4430 RLCB /ISSUE 12 BIT DAR TO SILO RESET
1677 4425 RLSD /WAIT FOR DONE FLAG
1700 4473 JMPPM1
1701 4467 XFRCHK /CHECK TRANSFER TO SILO
1702 5307 JMP .+5 /OK--EXIT TEST
1703 4436 RRCB /GET COMMAND B
1704 3117 DCA DATA5 /SAVE FOR TYPEOUT
1705 4451 ERROR /DAR TO SILO TRANSFER FAILED
1706 0327 DAR11 /PC H18-EXPCTD-LO8 H18-ACTUAL-LO8 CB
1707 4450 SCOPE

```

/VERIFY SILO CAN BE FILLED WITHOUT ERRORS.

```

/USE MAINTENANCE MODE TO FILL SILO COMPLETELY. VERIFY NO ERROR
/BITS WERE SET. READ BACK ALL SILO WORDS AND VERIFY NO ERRORS.
/

```

```

1710 4455 TEST19, PAT52
1711 4427 RLCA /WRITE TEST PATTERN TO CA
1712 1124 TAD PATRN /PICK UP PATTERN AGAIN
1713 4431 RLCA /WRITE SAME PATTERN TO SECTOR ADDRESS
1714 4465 DAREXP /COMPUTE EXPECTED DAR VALUE
1715 0136 AND K364 /MASK RESET AND MARKER BITS
1716 1160 TAD K13 /NOW ADD THEM IN
1717 3117 DCA DATA5 /SAVE WORD 2
1720 1113 TAD DATA1 /PICK UP WORD 1 AND SAVE
1721 3116 DCA DATA4 / IN DIFFERENT SPOT

```


1722	4474	SLOFIL	/FILL SILO
1723	4433	RRER	/ERROR OCCURRED--READ ERROR FLAG
1724	0164	AND K7774	/MASK OUT DRIVE READY AND DRIVE ERROR
1725	7450	SNA	/SKIP IF AN ERROR BIT IS SET
1726	5333	JMP .+5	/CONTINUE TEST BELOW
1727	3113	DCA DATA1	/SAVE FOR TYPEOUT
1730	4451	ERROR	/ERROR OCCURRED AFTER FILLING SILO
1731	2210	ERSLOF	/PC ER
1732	5365	JMP END19	/EXIT TEST
1733	1170	TAD M10	/SET UP COUNTER FOR CHECKING WORDS IN SILO
1734	3370	DCA SLOCTA	
1735	3115	DCA DATA3	/CLEAR WORD NUMBER BEING CHECKED
1736	4447	SLORDL, TICK	/GENERATE APT TIMING
1737	2115	ISZ DATA3	/INCREMENT TO NEXT WORD NUMBER
1740	1117	TAD DATA5	/GET EXPECTED VALUE
1741	3124	DCA PATTRN	/SAVE FOR COMPARISON
1742	4440	RRSI	/GET SILO WORD
1743	0146	AND K377	/MASK GARBAGE
1744	4456	PATCMP	/COMPARE TO EXPECTED VALUE
1745	5351	JMP .+4	/COMPARED EQUAL--CONTINUE BELOW
1746	4451	ERROR	/SILO FILL TEST DATA ERROR
1747	2235	SLODAT	/PC EXPCTD ACTUAL WORD NO.
1750	5365	JMP END19	/EXIT TEST
1751	2115	ISZ DATA3	/INCREMENT WORD NUMBER BEING CHECKED
1752	1116	TAD DATA4	/GET EXPECTED VALUE
1753	3124	DCA PATTRN	/SAVE FOR COMPARISON
1754	4440	RRSI	/READ SILO WORD
1755	0146	AND K377	/MASK GARBAGE
1756	4456	PATCMP	/COMPARE TO EXPECTED VALUE
1757	5363	JMP .+4	/OK--CONTINUE BELOW ERROR CALL
1760	4451	ERROR	/SILO FILL TEST DATA ERROR
1761	2235	SLODAT	/PC EXPCTD ACTUAL WORD NO.
1762	5365	JMP END19	/EXIT TEST
1763	2370	ISZ SLOCTA	/CHECKED ALL SILO WORDS?
1764	5336	JMP SLORDL	/NO--DO TWO MORE
1765	4450	END19, SCOPE	/YES--ALL DONE
1766	5767	JMP I .+1	
1767	2000	TEST20	
1770	0000	SLOCTA, 0	
1771	7773	K7773A, 7773	
1772	7764	K7764, 7764	
2000		PAGE	
		/VERIFY OVERFILLING SILO SETS READ DATA LATE.	
		/OVERFILL SILO USING MAINTENANCE MODE. VERIFY DATA LATE SET IN	
		/ERROR REGISTER.	
		/VERIFY RLDC CLEARS DATA LATE IN ERROR REGISTER.	
		/	
2000	4474	TEST20, SLOFIL	/FILL SILO
2001	4462	MRSET3	/INITIATE ONE MORE DAR TO SILO TRANSFER
2002	4433	RRER	/GET ERROR REGISTER
2003	7006	RTL	/GET DATA LATE INTO SIGN BIT
2004	7500	SMA	/SKIP IF DATA LATE SET

2005	5233	JMP	NODALA	/NO DATA LATE--GO REPORT ERROR
2006	4442	RLSE		/SKIP ON ERROR
2007	5226	JMP	NOFL20	/GO REPORT ERROR--FLAG SHOULD HAVE BEEN SET
2010	4442	RLSE		/ERROR FLAG SHOULD HAVE BEEN CLEARED
2011	5215	JMP	.+4	/YES--CONTINUE
2012	4451	ERROR		/ERROR FLAG NOT CLEARED BY RLSE
2013	2256	EFCLR		/PC
2014	5237	JMP	END20	
2015	4424	RLDC		/CLEAR DEVICE
2016	4433	RRER		/READ ERROR REG
2017	0164	AND	K7774	/MASK DRIVE READY BIT
2020	7450	SNA		/SKIP IF NOT CLEARED
2021	5237	JMP	END20	/ALL OK--EXIT TEST
2022	3113	DCA	DATA1	/SAVE ERROR REG FOR TYPEOUT
2023	4451	ERROR		/ER NOT CLEARED BY RLDC
2024	2302	ERDC		/PC ER
2025	5237	JMP	END20	/EXIT TEST
2026	7012	NOFL20, RTR		/RESTORE ERROR REG
2027	3113	DCA	DATA1	/SAVE FOR ERROR TYPEOUT
2030	4451	ERROR		/ERROR FLAG NOT SET BY READ DATA LATE FF
2031	3361	EFNRDL		/PC ER
2032	5237	JMP	END20	
2033	7012	NODALA, RTR		/RESTORE ERROR REG
2034	3113	DCA	DATA1	/SAVE FOR ERROR PRINT
2035	4451	ERROR		/DATA LATE NOT SET BY OVERFILLING SILO
2036	2322	NDL20		/PC ER
2037	4450	END20, SCOPE		

/VERIFY SILO CLEARED BY RLDC.
 /FILL SILO. READ TWO WORDS. INITIATE TRANSFER. VERIFY NO
 /ERRORS. READ ONE WORD. INITIATE ONE TRANSFER (OVERFILL BY
 /ONE). VERIFY DATA LATE SET. ISSUE RLDC. REFILL
 /SILO AND CHECK FOR NO ERRORS TO VERIFY SILO WAS CLEARED.
 /

2040	4474	TEST21, SLOFIL		/FILL SILO
2041	4440	RRSI		/EMPTY TWO WORDS FROM SILO
2042	4440	RRSI		
2043	7200	CLA		
2044	4457	RESET		/INITIATE A TRANSFER
2045	4433	RRER		/READ ERROR REG
2046	0164	AND	K7774	/MASK OUT DRIVE ERROR AND READY
2047	7450	SNA		/SKIP IF AN ERROR BIT IS SET
2050	5255	JMP	.+5	/CONTINUE TEST
2051	3113	DCA	DATA1	/SAVE FOR ERROR TYPEOUT
2052	4451	ERROR		/ERROR OCCURRED AFTER FILLING SILO
2053	2210	ERSLOF		/PC ER
2054	5303	JMP	END21	/EXIT TEST
2055	4440	RRSI		/EMPTY ONE WORD FROM SILO
2056	7200	CLA		
2057	4457	RESET		/INITIATE A TWO WORD TRANSFER
2060	4433	RRER		/GET ERROR REG
2061	7006	RTL		/GET DATA LATE INTO SIGN BIT
2062	7710	SPA CLA		/SKIP IF DATA LATE NOT SET
2063	5271	JMP	.+6	/CONTINUE BELOW
2064	4433	RRER		/SAVE ERROR REG FOR TYPEOUT

2065	3113	DCA	DATA1	
2066	4451	ERROR		/DATA LATE NOT SET BY OVERFILLING SILO
2067	2322	NDL20		/ PC ER
2070	5303	JMP	END21	/EXIT TEST
2071	4447	TICK		/TICK APT TIMER BECAUSE OF LONG TEST
2072	4424	RLDC		/CLEAR DEVICE
2073	4474	SLOFIL		/FILL SILO
2074	4433	RRER		/SAVE ERROR REG FOR TYPEOUT
2075	0164	AND	K7774	/MASK OUT DRIVE READY AND DRIVE ERROR
2076	7450	SNA		/SKIP IF AN ERROR BIT IS SET
2077	5303	JMP	END21	/EXIT TEST
2100	3113	DCA	DATA1	
2101	4451	ERROR		/FULL SILO COULD NOT BE REFILLED AFTER RLDC
2102	2377	NOREFL		/ PC ER
2103	4450	SCOPE		
2104	5705	JMP I	+.1	
2105	2200	TEST22		
2106	0000	VTCHK,	0	
2107	6030	KCF		
2110	6031	KSF		
2111	5706	JMP I	VTCHK	
2112	2306	ISZ	VTCHK	
2113	5706	JMP I	VTCHK	
2114	0000	XSCNIN,	0	
2115	7300	CLA CLL		/CLEAR AC AND LINK
2116	6055	ESIA		/CLEAR INTRPT ENABLE (CRTC)
2117	6115	6115		/CLEAR INTRPT ENABLE (TTY)
2120	5714	JMP I	XSCNIN	/RETURN TO CALING ROUTINE
2121	0000	XSETIN,	0	
2122	6111	6111		
2123	7000	NOP		
2124	6051	6051		
2125	7000	NOP		
2126	6116	6116		
2127	7200	CLA		
2130	7201	CLA IAC		
2131	6115	6115		
2132	6055	6055		
2133	7200	CLA		
2134	5721	JMP I	XSETIN	
2135	0000	XVT278,	0	
2136	6031	KSF		/SKIP IF FLAG SET, CLEAR IT ON VT278
2137	6030	KCF		/SET TTY FLAG
2140	5735	JMP I	XVT278	/RETURN TO CALLING ROUTINE
2200	PAGE			
/VERIFY OPI AND ERROR SET BY SEEK, CLEARED BY RLDC.				
/ISSUE SEEK COMMAND TO DRIVE. VERIFY DRIVE READY REMAINS ASSERTED.				
/IF NOT, INHIBIT PROGRAM FROM CONTINUING. VERIFY OPI, ERROR, AND				
/DONE SET. CHECK THAT RLDC CLEARS OPI.				
/				
2200	1350	TEST22, TAD	K4303	/PICK UP SEEK COMMAND W/DRIVE INTERFACE DISABLED
2201	3124	DCA	PATTRN	/SAVE FOR COMPARISON

2202	1124	TAD	PATRN	
2203	4430	RLCB		/ISSUE COMMAND
2204	4436	RRCB		/READ BACK CB
2205	4456	PATCMP		/VERIFY ITS VERACITY
2206	5213	JMP	.+5	/ALL OK--CONTINUE TEST
2207	4444	STOPF		/CLEAR DEVICE
2210	4451	ERROR		/CB DID NOT CONTAIN EXPECTED DATA
2211	0217	CB2B		/PC EXPCTD ACTUAL
2212	5310	JMP	END22	/EXIT TEST
2213	4464	WAIT		/WAIT LONG ENOUGH FOR DRIVE READY TO DROP
2214	4464	WAIT		/IF DRIVE INTERFACE WERE NOT DISABLED
2215	4433	RRER		/GET ERROR REG
2216	7010	RAR		/ROTATE DRIVE READY BIT INTO LINK
2217	7630	SZL CLA		/SKIP IF DRIVE READY CLEAR
2220	5230	JMP	OPWA22	/GO WAIT FOR OPI
2221	4436	RRCB		/SAVE COMMAND B FOR ERROR TYPEOUT
2222	3113	DCA	DATA1	
2223	4444	STOPF		/CLEAR DEVICE
2224	4451	ERROR		/DRIVE READY BIT CLEAR WITH INTERFAC DISABLED
2225	1762	DRVRODY		/PC CB
2226	4452	CONSOL		/ALLOW CONSOLE INPUT
2227	5200	JMP	TEST22	/LOCK PERMANENTLY ON TEST
2230	4463	OPWA22, OPIWAT		/WAIT FOR DONE AND OPI
2231	5254	JMP	NETH22	/NEITHER DONE NOR OPI
2232	5262	JMP	NDN22	/OPI BUT NO DONE
2233	5272	JMP	NOPI22	/DONE BUT NO OPI
		/BOTH DONE AND OPI		
2234	4442	RLSE		/SKIP ON ERROR
2235	7410	SKP		/ERROR FLAG S/B INHIBITED--CONTINUE TEST
2236	5247	JMP	ERFLST	/ERROR FLAG SET--REPORT ERROR
2237	4424	RLDC		/CLEAR DEVICE
2240	4433	RRER		/PICK UP ERROR REG
2241	0164	AND	K7774	/MASK OUT DRIVE READY BIT
2242	7450	SNA		/SKIP IF ER NOT CLEARED BY RLDC
2243	5310	JMP	END22	/EXIT TEST
2244	4451	ERROR		/ER NOT CLEARED BY RLDC
2245	2302	ERDC		/PC ER
2246	5310	JMP	END22	/EXIT TEST
2247	4433	ERFLST, RRER		/GET ERROR REG AND SAVE FOR TYPEOUT
2250	3113	DCA	DATA1	
2251	4451	ERROR		/ERROR FLAG SET WHEN INTERFACE DISABLED (DRV ERR NOT ASSERTED)
2252	3517	EFNOIN		/PC ER
2253	5310	JMP	END22	/EXIT TEST
2254	4436	NETH22, RRCB		/GET CB AND SAVE FOR ERROR TYPEOUT
2255	3113	DCA	DATA1	
2256	4444	STOPF		/CLEAR DEVICE
2257	4451	ERROR		/OPI AND DONE DID NOT SET
2260	2075	NDNOPI		/PC CB
2261	5310	JMP	END22	/EXIT TEST
2262	4436	NDN22, RRCB		/SAVE CB FOR ERROR TYPEOUT
2263	0114	DATA2		
2264	4433	RRER		/SAVE ER FOR ERROR TYPEOUT
2265	0113	DATA1		
2266	4444	STOPF		/CLEAR DEVICE
2267	4451	ERROR		/DONE NOT SET AFTER OPI

```

2270 2116          OPINDN          / PC   ER   CB
2271 5310          JMP      END22   /EXIT TEST
2272 4442      NOI22, RLSE          /SKIP IF ERROR FLAG SET
2273 5304          JMP      NO22    /REPORT ERROR
2274 4433          RRER            /SAVE ERROR REG
2275 3113          DCA      DATA1
2276 4436          RRCB            /SAVE CB
2277 3114          DCA      DATA2
2300 4424          RLDC            /CLEAR DEVICE
2301 4451          ERROR          /ERROR FLAG SET
2302 0335          ERFLGS        / PC   ER   CB
2303 5310          JMP      END22   /EXIT TEST
2304 4436      NO22,  RRCB            /SAVE CB
2305 3113          DCA      DATA1
2306 4451          ERROR
2307 2756          OPINST
2310 7240      END22, STA
2311 3121          DCA      LUPCNT  /OPI NOT SET (DRIVE INTERFACE NOT DISABLED?)
2312 4450          SCOPE          / PC   CB
                                   /-1 INTO AC
                                   /INHIBIT LOOPING

```

```

/VERIFY CAF TERMINATES ACTION AND INHIBITS OPI
/ISSUE A SEEK TO THE DRIVE.  ISSUE CAF AND VERIFY OPI AND DONE
/DO NOT SET.
/NOTE:  WHEN LOOPING ON THIS TEST, IT MAY BE NECESSARY TO TYPE A CHARACTER
/       SEVERAL TIMES BEFORE IT GETS SEEN BY THE CONSOLE PACKAGE.
/

```

```

2313 1350      TEST23, TAD      K4303  /ISSUE SEEK
2314 4430          RLCB
2315 4464          WAIT            /WAIT A BIT
2316 4452          CONSOL        /CHK FOR CONSOLE INPUT IMMEDIATELY PRIOR TO CAF
2317 6007          CAF            /TERMINATE ACTION
2320 6035          KIE            /DISABLE CONSOLE INTERRUPTS INPUT ONLY
2321 6045          TIE            /DISABLE CONSOLE INTERRUPTS OUTPUT ONLY
2322 4463          OPIWAT        /WAIT FOR OPI OR DONE
2323 5343          JMP      END23   /NEITHER--EXIT TEST
2324 5331          JMP      OPI23   /OPI BUT NO DONE
2325 5334          JMP      DONE23  /DONE BUT NO OPI
2326 4451          ERROR          /DONE AND OPI SET AFTER SEEK AND CAF
2327 2572          SKDO          / PC
2330 5343          JMP      END23   /EXIT TEST
2331 4451      OPI23,  ERROR        /CAF DID NOT INHIBIT OPI
2332 2620          CAFOP1        / PC
2333 5343          JMP      END23   /EXIT TEST
2334 4442      DONE23, RLSE        /SKIP IF ERROR FLAG SET
2335 5341          JMP      .+4
2336 4451          ERROR          /ERROR AND DONE SET AFTER CAF
2337 2640          CAFEDS        / PC
2340 5343          JMP      END23   /EXIT TEST
2341 4451          ERROR          /DONE SET AFTER CAF
2342 2663          DNSAC        / PC
2343 7240      END23,  STA
2344 3121          DCA      LUPCNT  /INHIBIT SUBTEST LOOPING
2345 4450          SCOPE
2346 5747          JMP I      .+1
2347 2400          TEST24

```

2350 4303 K4303, 4303

2400 PAGE

/VERIFY OPI SET BY READ HEADER, CLEARED BY LOADING CB.
 /ISSUE READ HEADER. VERIFY DONE AND OPI ARE
 /SET. LOAD CB. VERIFY ERROR BITS CLEARED. ISSUE CAF TO TERMINATE ACTION.
 /NOTE: WHEN LOOPING ON THIS TEST IT MAY BE NECESSARY TO TYPE A CHARACTER
 / SEVERAL TIMES BEFORE IT IS SEEN BY THE CONSOLE PACKAGE.
 /

2400	1365	TEST24, TAD	K304	/PICK UP READ HEADER COMMAND TO DRIVE 3
2401	3124	DCA	PATRN	/SAVE FOR COMPARISON
2402	1124	TAD	PATRN	
2403	4430	RLCB		/ISSUE COMMAND
2404	4436	RRCB		/READ BACK CB
2405	4456	PATCMP		/VERIFY ITS VERACITY
2406	5213	JMP	.+5	/ALL OK--CONTINUE TEST
2407	4444	STOPF		/CLEAR DEVICE
2410	4451	ERROR		/CB DID NOT CONTAIN EXPECTED DATA
2411	0217	CB2B		/PC EXPCTD ACTUAL
2412	5302	JMP	END24	/EXIT TEST
2413	4463	OPIWAT		/WAIT FOR DONE AND OPI
2414	5237	JMP	NET24	/NEITHER DONE NOR OPI
2415	5250	JMP	NDN24	/OPI BUT NO DONE
2416	5260	JMP	NOP24	/DONE BUT NO OPI
		/BOTH DONE AND OPI		
2417	4442	RLSE		/VERIFY OPI SET THE ERROR FLAG
2420	5271	JMP	NOFL24	/GO REPORT THE ERROR
2421	1367	TAD	K4303A	/PICK UP A SEEK COMMAND
2422	4430	RLCB		/LOAD CB
2423	4433	RRER		/PICK UP ERROR REG
2424	0164	AND	K7774	/MASK OUT DRIVE READY BIT
2425	7450	SNA		/SKIP IF IT WAS NOT CLEARED
2426	5232	JMP	.+4	/EXIT TEST
2427	3113	DCA	DATA1	/SAVE ER FOR ERROR TYPEOUT
2430	4451	ERROR		/ER NOT CLEARED BY LOADING CB
2431	2772	ERNCLD		/PC ER
2432	4452	CONSOL		/CHECK FOR CONSOLE INPUT
2433	6007	CAF		/CLEAR THE WORLD
2434	6035	KIE		/DISABLE CONSOLE INTERRUPTS INPUT ONLY
2435	6045	TIE		/DISABLE CONSOLE INTERRUPTS OUTPUT ONLY
2436	5302	JMP	END24	/EXIT TEST
2437	4436	NET24, RRCB		/GET CB AND SAVE FOR ERROR TYPEOUT
2440	3113	DCA	DATA1	
2441	4452	CONSOL		/CHECK FOR CONSOLE INPUT
2442	6007	CAF		/CLEAR THE WORLD
2443	6035	KIE		/DISABLE CONSOLE INTERRUPTS INPUT ONLY
2444	6045	TIE		/DISABLE CONSOLE INTERRUPTS OUTPUT ONLY
2445	4451	ERROR		/OPI AND DONE DID NOT SET
2446	2075	NDNOPI		/PC CB
2447	5302	JMP	END24	/EXIT TEST
2450	4436	NDN24, RRCB		/SAVE CB FOR ERROR TYPEOUT
2451	0114	DATA2		
2452	4433	RRER		/SAVE ER FOR ERROR TYPEOUT

2453	0113	DATA1		
2454	4444	STOPF		/CLEAR DEVICE
2455	4451	ERROR		/DONE NOT SET AFTER OPI
2456	2116	OPINDM		/ PC ER CB
2457	5302	JMP	END24	/EXIT TEST
2460	4442	NOP24, RLSE		/SKIP IF ERROR FLAG SET
2461	5276	JMP	NO24	/REPORT ERROR
2462	4433	RRER		/SAVE ERROR REG
2463	3113	DCA	DATA1	
2464	4436	RRCB		/SAVE CB
2465	3114	DCA	DATA2	
2466	4451	ERROR		/ERROR FLAG SET
2467	0335	ERFLGS		/ PC ER CB
2470	5302	JMP	END24	/EXIT TEST
2471	4433	NOFL24, RRER		/SAVE ER FOR ERROR TYPEOUT
2472	3113	DCA	DATA1	
2473	4451	ERROR		/ERROR FLAG NOT SET BY OPI
2474	3411	NEFOPI		/PC ER
2475	5302	JMP	END24	/EXIT TEST
2476	4436	NO24, RRCB		/SAVE CB
2477	3113	DCA	DATA1	
2500	4451	ERROR		/OPI NOT SET
2501	2756	OPINST		/ PC CB
2502	7240	END24, STA		/-1 INTO AC
2503	3121	DCA	LUPCNT	/INHIBIT LOOPING
2504	4450	SCOPE		
/VERIFY OPI SET BY READ.				
/ISSUE READ COMMAND. VERIFY DONE AND OPI ARE SET.				
/				
2505	1366	TEST25, TAD	K4316	/PICK UP READ COMMAND W/DRIVE INTERFACE DISABLED
2506	3124	DCA	PATRN	/SAVE FOR COMPARISON
2507	1124	TAD	PATRN	
2510	4430	RLCB		/ISSUE COMMAND
2511	4436	RRCB		/READ BACK CB
2512	4456	PATCMP		/VERIFY ITS VERACITY
2513	5320	JMP	.+5	/ALL OK--CONTINUE TEST
2514	4444	STOPF		/CLEAR DEVICE
2515	4451	ERROR		/CB DID NOT CONTAIN EXPECTED DATA
2516	0217	CB2B		/PC EXPCTD ACTUAL
2517	5360	JMP	END25	/EXIT TEST
2520	4463	OPIWAT		/WAIT FOR DONE AND OPI
2521	5325	JMP	NET25	/NEITHER DONE NOR OPI
2522	5333	JMP	NDN25	/OPI BUT NO DONE
2523	5343	JMP	NOP25	/DONE BUT NO OPI
2524	5360	JMP	END25	/BOTH DONE AND OPI--EXIT TEST
2525	4436	NET25, RRCB		/GET CB AND SAVE FOR ERROR TYPEOUT
2526	3113	DCA	DATA1	
2527	4444	STOPF		/CLEAR DEVICE
2530	4451	ERROR		/OPI AND DONE DID NOT SET
2531	2075	NDNOPI		/ PC CB
2532	5360	JMP	END25	/EXIT TEST
2533	4436	NDN25, RRCB		/SAVE CB FOR ERROR TYPEOUT
2534	0114	DATA2		
2535	4433	RRER		/SAVE ER FOR ERROR TYPEOUT

2536	0113	DATA1	
2537	4444	STOPF	
2540	4451	ERROR	/CLEAR DEVICE
2541	2116	OPINDN	/DONE NOT SET AFTER OPI
2542	5360	JMP	/ PC ER CB
2543	4442	NOP25, RLSE	/EXIT TEST
2544	5354	JMP	/SKIP IF ERROR FLAG SET
2545	4433	RRER	/REPORT ERROR
2546	3113	DCA	/SAVE ERROR REG
2547	4436	RRCB	
2550	3114	DCA	/SAVE CB
2551	4451	ERROR	/ERROR FLAG SET
2552	0335	ERFLGS	/ PC ER CB
2553	5360	JMP	/EXIT TEST
2554	4436	NO25, RRCB	/SAVE CB
2555	3113	DCA	
2556	4451	ERROR	/OPI NOT SET (DRIVE INTERFACE NOT DISABLED?)
2557	2756	OPINST	/ PC CB
2560	7240	END25, STA	/-1 INTO AC
2561	3121	DCA	/INHIBIT LOOPING
2562	4450	SCOPE	
2563	5764	JMP I	
2564	2600	TEST26	.+1
2565	0304	K304,	304
2566	4316	K4316,	4316
2567	4303	K4303A,	4303
2600		PAGE	
			/VERIFY CLK OPI DLY GETS ASSERTED BY READ W/O HEADER CHECK COMMAND
			/VERIFY READ W/O HEADER CHECK MICRO-CODE SETS ERROR FLAG AND DONE
			/
2600	4466	TEST26, PATM1	/SET UP TO WRITE SOME 1'S INTO DAR
2601	4431	RLSA	
2602	4466	PATM1	/WRITE SAME PATTERN TO COMMAND A
2603	4427	RLCA	
2604	4465	DAREXP	/COMPUTE EXPECTED SILO VALUES
2605	0136	AND	/MASK RESET AND MARKER BITS
2606	1160	TAD	/ADD IN RESET AND MARKER BITS
2607	3114	DCA	/SAVE EXPECTED VALUE
2610	4457	RESET	/DO A DAR TO SILO TRANSFER
2611	1173	TAD	/PICK UP AND ISSUE A READ W/O HEADER CHK CQMMAND
2612	3124	DCA	/SAVE FOR COMPARISON
2613	1124	TAD	
2614	4430	RLCB	
2615	4436	RRCB	/READ BACK CB
2616	4456	PATCMP	/VERIFY ITS VERACITY
2617	5224	JMP	/ALL OK--CONTINUE TEST
2620	4444	STOPF	/CLEAR DEVICE
2621	4451	ERROR	/CB DID NOT CONTAIN EXPECTED DATA
2622	0217	CB2B	/PC EXPCTD ACTUAL
2623	5267	JMP	/EXIT TEST
2624	4464	WAIT	/LET A LITTLE MICROCODE EXECUTE
2625	4464	WAIT	/WAIT A BIT


```

2626 4464      WAIT
2627 4425      RLSD
2630 5257      JMP      NODN26
2631 4442      RLSE
2632 5265      JMP      NOER26
2633 4442      RLSE
2634 5240      JMP      .+4
2635 4451      ERROR
2636 2256      EFCLR
2637 5267      JMP      END26
2640 4427      RLCA
2641 4431      RLCA
2642 4457      RESET
2643 4467      XFRCHK
2644 7410      SKP
2645 5267      JMP      END26
2646 4466      PATM1
2647 0327      AND      K6777A
2650 7650      SNA CLA
2651 5267      JMP      END26
2652 1173      TAD      K5307
2653 3113      DCA      DATA1
2654 4451      ERROR
2655 3253      CLOPDL
2656 5267      JMP      END26
2657 1173      NODN26, TAD      K5307
2660 3113      DCA      DATA1
2661 4444      STOPF
2662 4451      ERROR
2663 0277      NODONE
2664 5267      JMP      END26
2665 4451      NOER26, ERROR
2666 2704      NOFLG7
2667 4450      END26, SCOPE

/VERIFY INTERRUPT REQUEST GENERATED BY ERROR.
/VERIFY NO PRIOR INTERRUPT REQUESTS ARE PENDING ON THE BUS.
/SET ERROR FLAG BY ISSUING A READ DATA W/O HEADER CHECK COMMAND WITH
/MAINT BIT 0 SET. CLEAR DONE BIT AND VERIFY AN INTERRUPT
/IS PENDING ON THE BUS. CLEAR ERROR AND VERIFY INTERRUPT IS NO
/LONGER REQUESTED.
/
2670 6003      TEST27, SRQ
2671 5275      JMP      .+4
2672 4451      UNI27, ERROR
2673 0310      UN8
2674 5312      JMP      END27
2675 1174      TAD      K5707
2676 4430      RLCB
2677 4425      RLSD
2700 4473      JMPPM1
2701 6003      SRQ
2702 5310      JMP      .+6
2703 4442      RLSE
2704 5272      JMP      UNI27

/SKIP ON INTERRUPT REQUEST
/CONTINUE BELOW ERROR CALL
/UNKNOWN INTERRUPT REQUESTED
/PC
/EXIT TEST
/ISSUE MAINT READ DATA W/O HDR CHECK WITH
/ INTERRUPT ENABLE SET
/WAIT FOR DONE

/SKIP IF INTERRUPT REQUESTED
/REPORT ERROR BELOW
/CLEAR ERROR FLAG
/SHOULD BE SKIPPED

```

```

2705 6003      SRQ                /INTERRUPT SHOULD NO LONGER BE REQUESTED
2706 5312      JMP      END27      /EXIT TEST
2707 5272      JMP      UNI27      /REPORT UNKNOWN INTERRUPT
2710 4451      ERROR              /INTERRUPT NOT REQUESTED ON ERROR
2711 2545      INT21              /PC
2712 4450      END27, SCOPE

/VERIFY ERROR FLAG CLEARED BY RLDC
/
2713 1173      TEST28, TAD      K5307      /ISSUE MAINT READ DATA W/O HDR CHK
2714 4430      RLCB
2715 4425      RLSD              /WAIT FOR DONE
2716 4473      JMPPM1
2717 4424      RLDC              /CLEAR THE DEVICE
2720 4442      RLSE              /ERROR FLAG SHOULD HAVE BEEN CLEARED
2721 5324      JMP      .+3        /OK--EXIT TEST
2722 4451      ERROR              /ERROR FLAG NOT CLEARED BY RLDC
2723 2353      EFDC              /PC
2724 4450      SCOPE
2725 5726      JMP I      .+1
2726 3000      TEST29

2727 6777      K6777A, 6777

3000          PAGE

/VERIFY MAINTENANCE WRITE BREAK (12 BIT) WORKS.
/SET UP BUFFER IN FIELD ONE. SET WORD COUNT TO READ AND WRITE
/BACK TWO WORDS (STANDARD PATTERNS) WITH CRC. ISSUE MAINTENANCE
/WRITE. VERIFY WORD COUNT OVERFLOW STOPPED TRANSFER. VERIFY
/DATA AND CRC WORDS IN BUFFER.
/
3000 4461      TEST29, RESET3      /RESET DRIVE 3
3001 7240      STA
3002 1171      TAD      PBUF1A      /PICK UP POINTER TO MEMORY BUFFER-1
3003 3010      DCA      AUTO10      /PLACE IN AUTOINC REG
3004 4454      PAT25              /GET TEST PATTERN
3005 6211      CDF      10          /SET UP BUFFER IN FIELD 1
3006 4476      BUFSET              /SET UP BUFFER AREA (4 WORDS)
3007 1164      TAD      M4          /SET UP WORD COUNT FOR 4 TRANSFERS
3010 4432      RLWC
3011 1171      TAD      PBUF1A      /SET UP MA REG TO POINT TO BUFFER
3012 4426      RLMA
3013 1143      TAD      K310        /PICK UP MAINT WRITE TO FIELD 1 COMMAND
3014 3124      DCA      PATTRN      /SAVE FOR COMPARISON
3015 3367      DCA      TIMER       /INIT A TIMER FOR DONE WAIT LOOP
3016 1124      TAD      PATTRN      /PICK UP COMMAND
3017 4430      RLCB              /ISSUE IT
3020 4436      RRCB              /READ BACK COMMAND B FOR COMPARISON
3021 4456      PATCMP              /COMPARE TO EXPECTED VALUE
3022 5227      JMP      .+5        /EQUAL--CONTINUE
3023 4444      STOPF              /STOP THE FUNCTION
3024 4451      ERROR              /CB DID NOT CONTAIN EXPECTED DATA
3025 0217      CB2B              /PC      EXPTD  ACTUAL
3026 5307      JMP      END29      /EXIT TEST

```

3027	4425	WALP29,	RLSD		/SKIP IF DONE FLAG HAS SET
3030	7410		SKP		/CHECK FOR WORD COUNT OVERFLOW
3031	5252		JMP	OUT29	/GET OUT OF WAIT LOOP FOR DONE FLAG
3032	4434		RRWC		/READ WORD COUNT REG
3033	7740		SMA SZA	CLA	/SKIP IF WORD COUNT IS NEGATIVE OR ZERO
3034	5245		JMP	STOPWR	/WORD COUNT WENT POSITIVE--GO STOP WRITE
3035	2367		ISZ	TIMER	/TICK THE TIMER
3036	5227		JMP	WALP29	/CONTINUE WAITING FOR DONE
3037	1143		TAD	K310	/SAVE COMMAND B FOR ERROR TYPEOUT
3040	3113		DCA	DATA1	
3041	4444		STOPF		/STOP THE FUNCTION
3042	4451		ERROR		/DONE NOT SET
3043	0277		NODONE		/PC CB
3044	5307		JMP	END29	/EXIT TEST
3045	4444	STOPWR,	STOPF		/STOP THE FUNCTION
3046	4451		ERROR		/WORD COUNT OVERFLOW DID NOT STOP. MAINT TRANSFER
3047	3015		NWCOVF		/PC
3050	4452		CONSOL		/CHECK FOR CONSOLE INPUT
3051	5200		JMP	TEST29	/LOCK ON TEST--CONTINUING WILL WIPE
					/OUT PROGRAM
3052	4472	OUT29,	MANTEC		/CHECK FOR ERROR BITS AND REPORT
3053	5307		JMP	END29	/ERROR--EXIT TEST
3054	4454		PAT25		/GET PATTERN INTO LOCATION PATTRN
3055	7240		STA		/SUBTRACT 1 FROM BUFFER ADDRESS
3056	1171		TAD	PBUF1A	/FOR USE IN AUTO INC REG
3057	3010		DCA	AUTO10	
3060	6241		CDF	10	/BUFFER IS IN FIELD 1
3061	1410		TAD I	AUTO10	/GET 1ST WORD OF BUFFER
3062	6201		CDF	0	/BACK TO HERE
3063	4456		PATCMP		/COMPARE WORD TO CORRECT ONE
3064	5272		JMP	.*6	/COMPARED EQUAL
3065	1010		TAD	AUTO10	/SAVE SUPPOSED MA FOR TYPEOUT
3066	3115		DCA	DATA3	
3067	4451		ERROR		/READ DATA BREAK MODIFIED WORD READ
3070	3107		REDMOD		/PC EXPCTD ACTUAL MA(EXPCTD)
3071	5307		JMP	END29	/EXIT TEST
3072	6211		CDF	10	
3073	1410		TAD I	AUTO10	/GET WORD FROM WRITE BREAK
3074	6201		CDF	0	
3075	4456		PATCMP		/COMPARE TO WORD FOR READ BREAK
3076	5304		JMP	.*6	/EQUAL
3077	1010		TAD	AUTO10	/SAVE SUPPOSED MA FOR TYPEOUT
3100	3115		DCA	DATA3	
3101	4451		ERROR		/MAINT WRITE DID NOT WRITE BACK WORD READ OR MA BAD
3102	3051		MWRER		/PC EXPCTD ACTUAL MA(EXPCTD)
					/IF ACTUAL=0, THEN MA BAD; ELSE WROTE BACK BAD DATA
3103	5307		JMP	END29	/EXIT TEST
3104	4470		CRCCHK		/GENERATE AND CHECK CRC WORDS
3105	0010		10		/FIELD ARGUMENT
3106	7000		NOP		/ERROR--EXIT TEST
3107	4450	END29,	SCOPE		

/PHASE LOCKED LOOP ADJUSTMENT TEST

/THIS TEST IS EXECUTED ONLY IF REQUESTED BY THE OPERATOR IN RESPONSE TO
/QUESTION ASKED AT PROGRAM STARTUP. A MAINT COMMAND IS CONTINUALLY

/EXECUTED AND THE WORD WRITTEN BACK TO MEMORY IS DISPLAYED IN THE MQ REG.
 /THE DATA PATTERN USED IS TAKEN FROM THE SWITCH REGISTER. (IT IS RECOMMENDED
 /THAT THE HARDWARE SWITCHES BE USED FOR THIS TEST.) THE PHASE LOCK LOOP
 /POT MAY BE ADJUSTED UNTIL THE MQ DISPLAY IS SEEN STABLEIZED. IT IS
 /RECOMMENDED TO USE SEVERAL DATA PATTERNS. THIS TEST SHOULD NOT BE
 /EXECUTED UNTIL THE PREVIOUS TEST HAS PASSED AT LEAST TO THE POINT OF THE
 /ERROR MESSAGE "MAINT WRITE DID NOT WRITE BACK WORD READ OR MA BAD".
 /THIS TEST IS EXECUTED IN THE FIRST PASS ONLY.

3110	1130	TST29A, TAD	PASCNT	/DON'T EXECUTE TEST IS NOT FIRST PASS
3111	7640		SZA CLA	
3112	5766		JMP I PTST30	/GO TO NEXT TEST
3113	1132		TAD PHLKLP	/PICK UP FLAG TO SEE IF TEST IS REQUESTED
3114	7650		SNA CLA	/SKIP IF IT WAS
3115	5766		JMP I PTST30	/GO TO NEXT TEST
3116	4500		MESSAGE	/PHASE LOCK LOOP ADJUSTMENT ROUTINE
3117	4340		PLLRO1	/SWITCHES SELECT DATA PATTERN FOR MAINT TRANSFER.
3120	4500		MESSAGE	/WORD FORM WRITE BREAK DISPLAYED IN MQ.
3121	4425		PLLRO2	/TYPE <CR> TO EXIT
3122	7240	PLLOOP, STA		/SUBTRACT 1 FROM BUFFER ADDRESS FOR
3123	1171		TAD PBUF1A	/ USE IN AUTO INC REG
3124	3010		DCA AUTO10	
3125	4445		GETSR	/GET DATA PATTERN FROM SWITCHES
3126	6211		CDF 10	/CHANGE TO FIELD OF BUFFER
3127	3410		DCA I AUTO10	/DEPOSIT DATA PATTERN AT FIRST WORD OF BUFFER
3130	6201		CDF 0	
3131	1164		TAD M4	/SET UP A WORD COUNT
3132	4432		RLWC	
3133	1171		TAD PBUF1A	/POINT CONTROLLER TO BUFFER
3134	4426		RLMA	
3135	1143		TAD K310	/ISSUE MAINT COMMAND TO FIELD 1
3136	4430		RLCB	
3137	4425		RLSD	/WAIT FOR DONE
3140	5337		JMP .-1	/(JMPPM1'S CONSOLE CALL MAKES <CR> DISSAPPEAR
3141	6211		CDF 10	/CHANGE TO FIELD OF BUFFER
3142	1410		TAD 1 AUTO10	/PICK UP WORD FROM WRITE BREAK
3143	6201		CDF 0	
3144	7421		MQL	/DISPLAY DATA IN MQ
3145	6031		KSF	/SKIP IF KEYBOARD INPUT
3146	5322		JMP PLLOOP	/CONTINUE WITH ANOTHER MAINT COMMAND
3147	4507		VT278	/GO SET TTY FLAG FOR VT278
3150	4477		LISN	/CHECK OUT THE CHARACTER
3151	7563		-215	/ON <CR>.
3152	3161		END29A	/EXIT THE TEST
3153	7571		-207	/ON CONTROL-G
3154	3157		+.3	/OPEN THE SOFTWARE SWITCH REG
3155	0000		0	/NO OTHER CHARACTER ACCEPTED AT THIS TIME
3156	3122		PLLOOP	/ (EXCEPT CONTROL-C)
3157	4453		CNTRLG	/HANDLE THE CONTROL G
3160	5322		JMP PLLOOP	
3161	4453	END29A, CNTRLG		/LET SWITCH REG VALUE BE ENTERED IN CASE
				/OPERATOR LEFT DATA IN IT
3162	4500		MESSAGE	/EXIT
3163	4335		EXIT	
3164	4450		SCOPE	

```

3165 5766          JMP I   PTST30

3166 3200  PTST30, TEST30
3167 0000  TIMER,  0

          3200  PAGE

          /VERIFY WORD COUNT OVERFLOW WILL STOP TRANSFER WHEN NOT "EXPECTED" BY
          /ROM CONTROL STORE
          /

3200 1171  TEST30, TAD      PBUF1A      /SET UP A BUFFER
3201 4426          RLMA          /POINT CONTROLLER TO BUFFER
3202 7346          CLA CLL CMA RTL /-2 INTO AC
3203 4432          RLWC          /LOAD WORD COUNT (PARTIAL MAINT BLOCK)
3204 1143          TAD      K310      /PICK UP MAINT WRITE TO FIELD 1 COMMAND
3205 4430          RLCB          /ISSUE IT
3206 3125          DCA      TEMP1      /SET TIMER
3207 4425  WTLU30, RLSD          /SKIP ON DONE
3210 7410          SKP            /NOT DONE--SKIP
3211 5230          JMP      END30      /DONE SET--EXIT TEST
3212 4434          RRWC          /PICK UP WORD COUNT
3213 7740          SMA SZA CLA      /SKIP IF WORD COUNT STILL NEGATIVE OR ZERO
3214 5225          JMP      STPWRI     /WORD COUNT WENT BEYOND ZERO--STOP WRITE
3215 2125          ISZ      TEMP1      /TICK TIMER
3216 5207          JMP      WTLU30     /KEEP WAITING
3217 4436          RRCB          /SAVE COMMAND B FOR TYPEOUT
3220 3113          DCA      DATA1
3221 4444          STOPF          /STOP THE FUNCTION
3222 4451          ERROR          /DONE NOT SET
3223 0277          NODONE          /PC      CB
3224 5230          JMP      END30      /EXIT TEST
3225 4424  STPWRI, RLDC          /CLEAR DEVICE
3226 4451          ERROR          /WORD COUNT OVERFLOW DID NOT STOP MAINT TRANSFER
3227 3015          NWCOVF          /PC
3230 4450  END30,  SCOPE

          /VERIFY MA CLEARED BY RLDC.
          /SET UP BUFFER AT LOCATION 0 FIELD 1.  LOAD MA (TO NON-ZERO
          /BUFFER).  ISSUE RLDC AND MAINTENANCE WRITE.  VERIFY DATA
          /TRANSFER TOOK PLACE AT LOCATION 0 BUFFER.
          /

3231 7240  TEST31, STA          /POINT AUTO INC REG TO 1 LESS THAN
3232 3010          DCA      AUTO10     /BUFFER AT LOCATION 0
3233 4466          PATM1          /PATTERN INTO AC
3234 3124          DCA      PATTRN     /SAVE PATTERN FOR LATER COMPARISON
3235 1124          TAD      PATTRN
3236 6211          CDF      10        /SET UP BUFFER IN FIELD 1
3237 4476          BUFSET          /AT LOCATION 0
3240 1171          TAD      PBUF1A     /PICK UP POINTER TO NON-ZERO BUFFER
3241 4426          RLMA          /AND WRITE TO MA
3242 4424          RLDC          /CLEAR DEVICE
3243 1164          TAD      M4        /-4 INTO AC
3244 4432          RLWC          /SET UP WORD COUNT FOR FOUR WORDS
3245 1143          TAD      K310      /ISSUE MAINT WRITE TO FIELD 1
3246 4430          RLCB

```

3247	4425	RLSD		/SKIP ON DONE
3250	4473	JMPPM1		/JMP .-1 EXCEPT ALLOW FOR CONSOL INPUT
3251	6211	CDF	10	/FIELD OF BUFFER
3252	1567	TAD I	K1	/GET WORD FROM WRITE BREAK
3253	6201	CDF	0	
3254	7440	SZA		/SKIP IF BUFFER HAS NOT BEEN CHANGED
3255	5262	JMP	.+5	/OK--EXIT TEST
3256	4456	PATCMP		/COMPARE WITH EXPECTED VALUE
3257	5262	JMP	.+3	/COMPARE OK -- BREAK TOOK PLACE AT 0
3260	4451	ERROR		/MA NOT CLEARED BY RLDC
3261	3135	MADC		/PC
3262	4450	SCOPE		
3263	5664	JMP I	.+1	
3264	3400	TEST32		

3400 PAGE

/VERIFY CLK OPI DLY GETS ASSERTED BY WRITE COMMAND. VERIFY DATA BREAKS
/FROM MEMORY OCCUR.

3400	4457	TEST32, RESET		/DO A DAR TO SILO TRANSFER
3401	4454	PAT25		/GET A TEST PATTERN
3402	6211	CDF	10	
3403	3760	DCA I	K0	/WRITE INTO LOCATION 0--FIELD 1
3404	6201	CDF	0	
3405	1354	TAD	K4315	/PICK UP AND ISSUE A WRITE COMMAND
3406	3124	DCA	PATRN	/SAVE FOR COMPARISON
3407	1124	TAD	PATRN	
3410	4430	RLCB		
3411	4436	RRCB		/READ BACK CB
3412	4456	PATCMP		/VERIFY ITS VERACITY
3413	5220	JMP	.+5	/ALL OK--CONTINUE TEST
3414	4444	STOPF		/CLEAR DEVICE
3415	4451	ERROR		/CB DID NOT CONTAIN EXPECTED DATA
3416	0217	CB2B		/PC EXPCTD ACTUAL
3417	5255	JMP	END32	/EXIT TEST
3420	4464	WAIT		/LET A LITTLE MICROCODE EXECUTE
3421	4464	WAIT		/WAIT A BIT
3422	4464	WAIT		
3423	4471	ERRCHK		/CHECK FOR ERROR FLAG AND REPORT
3424	5255	JMP	END32	/ERROR--EXIT TEST
3425	4454	PAT25		/GET THE TEST PATTERN BACK INTO LOCATION PATRN
3426	4440	RRSI		/READ THE FIRST SILO WORD
3427	4456	PATCMP		/COMPARE TO THE EXPECTED WORD FROM THE READ BREAK
3430	5255	JMP	END32	/OK--EXIT TEST
3431	1114	TAD	DATA2	/PICK UP WORD THAT WAS READ FROM SILO
3432	1356	TAD	M13	/SEE IF IT WAS THE DAR TO SILO WORD
3433	7640	SZA CLA		/SKIP IF IT WAS
3434	5253	JMP	ERR32B	/GO REPORT BAD DATA BREAK ERROR
3435	4440	RRSI		/SEE IF THE SECOND WORD IN THE SILO IS FROM
3436	7450	SNA		/ THE 2ND DAR TO SILO WORD--SKIP IF NOT
3437	5246	JMP	ERR32A	/GO REPORT NO CLK OPI DLY
3440	1356	TAD	M13	/CHECK IF FIRST WORD WAS READ AGAIN
3441	7640	SZA CLA		/SKIP IF IT WAS--CLK OPI DLY WAS ASSERTED
3442	5253	JMP	ERR32B	/GO REPORT BAD DATA BREAK

3443	4451	ERROR		/DATA BREAKS DID NOT OCCUR IN WRITE COMMAND
3444	4120	D8NPBW		/PC
3445	5255	JMP	END32	/EXIT TEST
3446	1354	ERR32A, TAD	K4315	/READ AND SAVE COMMAND B FOR ERROR TYPEOUT
3447	3113	DCA	DATA1	
3450	4451	ERROR		/CLK OPI DLY NOT ASSERTED TO SILO (SILO NOT CLEARED)
3451	3253	CLOPDL		/PC CB
3452	5255	JMP	END32	
3453	4451	ERR32B, ERROR		/DATA BREAK READ BAD WORD ON WRITE COMMAND
3454	4067	DBRBW		/PC EXPCTD ACTUAL
3455	4444	END32, STOPF		/STOP THE FUNCTION
3456	4450	SCOPE		
/VERIFY DATA BREAKS TAKE PLACE ON READ COMMAND				
/				
3457	4455	TEST33, PAT52		/SET UP TO WRITE A TEST PATTERN INTO DAR
3460	4431	RLSA		
3461	4454	PAT25		/GET ALTERNATE TEST PATTERN
3462	4427	RLCA		
3463	4465	DAREXP		/COMPUTE EXPECTED SILO VALUES
3464	0136	AND	K364	/MASK RESET AND MARKER BITS
3465	1160	TAD	K13	/ADD IN RESET AND MARKER BITS
3466	3114	DCA	DATA2	/SAVE EXPECTED VALUE
3467	4457	RESET		/DO A DAR TO SILO TRANSFER
3470	7240	STA		/SET UP AN AUTO INC REG TO POINT TO BUFFER
3471	3010	DCA	AUTO10	
3472	6211	CDF	10	/CLEAR OUT A THREE WORD BUFFER IN FIELD 1
3473	3410	DCA I	AUTO10	
3474	3410	DCA I	AUTO10	
3475	3410	DCA I	AUTO10	
3476	6201	CDF	0	
3477	1355	TAD	K7316	/PICK UP AND ISSUE A READ COMMAND
3500	4430	RLCB		
3501	4464	WAIT		/LET A LITTLE MICROCODE EXECUTE
3502	4464	WAIT		/WAIT A BIT
3503	4464	WAIT		
3504	4471	ERRCHK		/CHECK FOR ERROR FLAG AND REPORT
3505	5350	JMP	END33	/ERROR--EXIT TEST
3506	7240	STA		/SET UP A POINTER IN AUTO INC REG TO THE BUFFER
3507	3010	DCA	AUTO10	
3510	6211	CDF	10	
3511	1410	TAD I	AUTO10	/PICK UP THE WORD FROM THE WRITE BREAK
3512	0146	AND	K377	/SAVE ONLY LOWER 8 BITS
3513	3116	DCA	DATA4	/SAVE IT
3514	1410	TAD I	AUTO10	/GET NEXT WORD FROM BREAK
3515	0146	AND	K377	/SAVE ONLY LOWER 8 BITS
3516	3115	DCA	DATA3	/SAVE IT
3517	1410	TAD I	AUTO10	/GET WORD AFTER LAST BREAK WORD
3520	6201	CDF	0	
3521	3117	DCA	DATA5	/SAVE IT
3522	1113	TAD	DATA1	/COMPARE EXPECTED BREAK WORDS
3523	7041	CIA		/ BY SUBTRACTION
3524	1115	TAD	DATA3	/WITH ACTUAL
3525	7640	SZA CLA		/SKIP IF EQUAL
3526	5344	JMP	ERR33	/GO REPORT ERROR

3527	1114	TAD	DATA2	/COMPARE 2ND WORD
3530	7041	CIA		/WITH THE ACTUAL ONE
3531	1116	TAD	DATA4	
3532	7640	SZA CLA		/SKIP IF EQUAL
3533	5344	JMP	ERR33	/REPORT THE ERROR
3534	1117	TAD	DATA5	/PICK UP THE WORD THAT SHOULD NOT HAVE BEEN CHANGED
3535	7450	SNA		/SKIP IF IT CHANGED
3536	5350	JMP	END33	/OK--EXIT TEST
3537	1355	TAD	K7316	/PICK UP CB AND SAVE FOR ERROR TYPEOUT
3540	3113	DCA	DATA1	
3541	4451	ERROR		/WRITE BREAK OCCURRED WHEN SILO WAS EMPTY
3542	4152	WBWSE		/PC CB
3543	5350	JMP	END33	/EXIT TEST
3544	1355	ERR33, TAD	K7316	/SAVE COMMAND B FOR ERROR TYPEOUT
3545	3117	DCA	DATA5	
3546	4451	ERROR		/WRITE BREAKS WERE BAD OR DID NOT OCCUR
3547	4203	WBBNO		/PC WD2-EXPCTD-WD1 WD2-ACTUAL-WD1 CB
3550	4444	END33, STOPF		/STOP THE FUNCTION
3551	4450	SCOPE		
3552	5753	JMP I	..+1	
3553	3600	TEST34		
3554	4315	K4315,	4315	
3555	7316	K7316,	7316	
3556	7765	M13,	-13	
3557	6777	K6777,	6777	
3560	0000	K0,	0	
3600		PAGE		
		/MAINT WRITE TO FIELD 1 -- ROTATE BUFFERS.		
		/SET UP BUFFER IN FIELD 1 TO TEST ALL MA BITS.		
		/		
3600	7240	TEST34, STA		
3601	1165	TAD	PBUF1B	/PICK UP POINTER TO MEMORY BUFFER-1
3602	3010	DCA	AUTO10	/PLACE IN AUTOINC REG
3603	4455	PAT52		/GET TEST PATTERN
3604	6211	CDF	10	/SET UP BUFFER IN FIELD 1
3605	4476	BUFSET		/SET UP BUFFER AREA (4 WORDS)
3606	1164	TAD	M4	/SET UP WORD COUNT FOR 4 TRANSFERS
3607	4432	RLWC		
3610	1165	TAD	PBUF1B	/SET UP MA REG TO POINT TO BUFFER
3611	4426	RLMA		
3612	1143	TAD	K310	/PICK UP MAINT WRITE TO FIELD 1 COMMAND
3613	4430	RLCB		/ISSUE IT
3614	4425	RLSD		/SKIP ON DONE
3615	4473	JMP PM1		/WAIT FOR DONE
3616	7240	STA		/SUBTRACT 1 FROM BUFFER LOCATION
3617	1165	TAD	PBUF1B	/FOR USE IN AUTO INC REG
3620	3010	DCA	AUTO10	
3621	6211	CDF	10	/BUFFER IS IN FIELD 1
3622	1410	TAD I	AUTO10	/GET 1ST WORD OF BUFFER
3623	6201	CDF	0	/BACK TO HERE
3624	4456	PATCMP		/COMPARE WORD TO CORRECT ONE
3625	5233	JMP	..+6	/COMPARED EQUAL


```

3626 1010      TAD      AUTO10      /SAVE SUPPOSED MA FOR TYPEOUT
3627 3115      DCA      DATA3
3630 4451      ERROR                      /READ DATA BREAK MODIFIED WORD READ
3631 3107      REDMOD                      /PC EXPCTD ACTUAL MA(EXPCTD)
3632 5250      JMP      END34          /EXIT TEST
3633 6211      CDF      10
3634 1410      TAD I    AUTO10          /GET WORD FROM WRITE BREAK
3635 6201      CDF      0              /BACK TO HERE
3636 4456      PATCMP                      /COMPARE TO WORD FOR READ BREAK
3637 5245      JMP      .+6            /EQUAL
3640 1010      TAD      AUTO10          /GET MA
3641 3115      DCA      DATA3
3642 4451      ERROR                      /MAINT WRITE DID NOT WRITE BACK WORD READ OR MA BAD
3643 3051      MWRER                      /PC EXPCTD ACTUAL MA(EXPCTD)
3644 5250      JMP      END34          /EXIT TEST
3645 4470      CRCCHK                      /GENERATE AND CHECK CRC WORDS
3646 0010      10                      /FIELD ARGUMENT
3647 7000      NOP                      /ERROR--EXIT TEST
3650 4450      END34, SCOPE

```

```

/VERIFY MAINTENANCE WRITE BREAK TO FIELD 0.
/SET UP A BUFFER IN FIELD 0. VERIFY BREAK TO BUFFER.
/

```

```

3651 4461      TEST35, RESET3          /RESET DRIVE 3
3652 7240      STA
3653 1166      TAD      PBUF0
3654 3010      DCA      AUTO10          /PICK UP POINTER TO MEMORY BUFFER
3655 4454      PAT25                      /PLACE IN AUTOINC REG
3656 4476      BUFSET                      /GET TEST PATTERN
3657 1164      TAD      M4              /SET UP BUFFER AREA (4 WORDS)
3660 4432      RLWC                      /SET UP WORD COUNT FOR 4 TRANSFERS
3661 1166      TAD      PBUF0          /SET UP MA REG TO POINT TO BUFFER
3662 4426      RLMA
3663 1144      TAD      K300
3664 4430      RLCB                      /ISSUE MAINT WRITE TO FIELD 0 COMMAND
3665 4425      RLSD                      /SKIP ON DONE
3666 4473      JMPPM1                    /WAIT FOR DONE
3667 4472      MANTEC                    /CHECK FOR ERROR BIT AND REPORT
3670 5304      JMP      END35          /ERROR--EXIT TEST
3671 1166      TAD      PBUF0          /FOR USE IN AUTO INC REG
3672 3010      DCA      AUTO10
3673 1410      TAD I    AUTO10          /GET WORD FROM WRITE BREAK
3674 7440      SZA                      /IF WORD IS NON-ZERO THEN BREAK OCCURED
3675 5304      JMP      END35          /BREAK OCCURED--EXIT TEST
3676 4456      PATCMP                      /CHECK IF WORD IS SUPPOSED TO BE ZERO
3677 5304      JMP      END35          /YES--EXIT TEST
3700 1010      TAD      AUTO10          /GET EXPECTED MA
3701 3113      DCA      DATA1          /SAVE FOR ERROR TYPEOUT
3702 4451      ERROR                      /DATA BREAK DID NOT OCCUR TO FIELD 0
3703 3155      NOTF0                      /PC MA(EXPCTD)
3704 4450      END35, SCOPE

```

```

/VERIFY DATA BREAK FROM THE ENTIRE 4K OF FIELD 1.
/

```

```

3705 1353      TEST36, TAD      K4315A      /PICK UP WRITE FROM FIELD 1 COMMAND

```

```

3706 4430      RLCB      /ISSUE IT
3707 4464      WAIT      /LET DMA'S HAVE TIME TO START
3710 4464      WAIT
3711 3125      DCA        TEMP1
3712 6211      CDF        10      /CLEAR OUT COMBINATION POINTER AND COUNTER
3713 6615      IOT15B, 6615      /CHANGE TO FIELD OF DATA
3714 7041      CIA        /READ THE SILO
3715 1525      TAD I      TEMP1      /NEGATE IT
3716 7440      SZA        /ADD IN CORRECT VALUE
3717 5324      JMP        .+5      /SKIP IF CORRECT
3720 2125      ISZ        TEMP1      /ERROR--REPORT IT BELOW
3721 5313      JMP        IOT15B      /BUMP POINTER--SKIP IF DONE WITH ALL 4K
3722 6201      CDF        0      /CONTINUE READING SILO
3723 5337      JMP        TSTWC      /DONE--CHANGE BACK TO THIS FIELD
3724 7041      CIA        /ALL DONE--GO CHECK THAT WORD COUNT REG CONTAINS 0
3725 1525      TAD I      TEMP1      /NEGATE THE VALUE IN THE AC -(SILO-CORRECTVAL)
3726 3114      DCA        DATA2      /ADD IN CORRECT VALUE (DATA FIELD IS STILL 2)
3727 1525      TAD I      TEMP1      /SAVE VALUE READ FROM SILO; (CORRECTVAL-(SILO-CORRECTVAL))=SILO
3730 6201      CDF        0      /GET CORRECT VALUE AND SAVE IT FOR ERROR TYPEOUT
3731 3113      DCA        DATA1      /BACK TO THIS FIELD FINALLY
3732 1125      TAD        TEMP1      /SAVE THE CORRECT EXPECTED VALUE
3733 3115      DCA        DATA3      /GET THE BREAK ADDRESS
3734 4451      ERROR      /SAVE FOR ERROR TYPEOUT
3735 4307      DER4K1      /DATA ERROR IN 4K WRITE FROM FIELD 1
3736 5345      JMP        END36      /PC      EXPCTD  ACTUAL  MA(EXPCTD)
3737 4434      TSTWC, RRWC      /READ THE WORD COUNT REG
3740 7450      SNA        /SKIP IF NOT ZERO
3741 5345      JMP        END36      /OK--EXIT TEST
3742 3113      DCA        DATA1      /SAVE WORD COUNT REG FOR ERROR TYPEOUT
3743 4451      ERROR      /WORD COUNT REGISTER NOT BACK TO 0 AFTER 4K DATA BREAK TRANSFER
3744 1010      WCBAD      /PC      WC
3745 4444      END36, STOPF      /CLEAR DEVICE TO TERMINATE DATA BREAKS
3746 7240      STA        /INHIBIT LOOPING ON THIS TEST
3747 3121      DCA        LUPCNT
3750 4450      SCOPE
3751 5752      JMP I      .+1
3752 4000      TEST37

3753 4315      K4315A, 4315

4000          PAGE

/VERIFY DATA BREAK TO ALL AVAILABLE FIELDS.
/USE EXTENDED MEMORY INFORMATION FROM OPERATOR OR APT (LOCATION MEMSIZ)
/TO CHECK MAINTENANCE WRITE TO ALL OTHER AVAILABLE FIELDS.
/

4000 1131      TEST37, TAD      MEMSIZ      /PICK UP MEMORY SIZE
4001 1170      TAD        M10      /SUBTRACT 8K FROM MEMORY SIZE
4002 7550      SPA SNA      /SKIP IF MORE THAN 8K AVAILABLE
4003 5736      JMP I      PPEOP      /NO MEMORY AVAILABLE--GO TO END OF PASS
4004 4475      BRKVFY      /VERIFY BREAK TO BEGINNING OF FIELD 2
4005 0002      2          /FIELD ARGUMENT
4006 4450      SCOPE

```

/IF ALL OF FIELD 2 IS AVAILABLE, FILL FIELD 2 WITH DATA AND DO READ BREAKS

```

/INTO THE SILO FROM THE ENTIRE 4K
/
4007 1131      TAD      MEMSIZ      /GET MEMORY SIZE INFORMATION
4010 1155      TAD      M14        /SUBTRACT 12K
4011 7710      SPA CLA              /SKIP IF ALL OF FIELD 2 IS AVAILABLE
4012 5736      JMP I   PPEOP        /NO MORE MEMORY--GO TO END OF PASS
4013 7240      STA                      /POINT AUTO INC REG 11 TO BEGINNING OF FIELD
4014 3011      DCA      AUTO11
4015 1135      TAD      K7700        /SET UP A COUNTER FOR THE OUTER LOOP
4016 3125      DCA      TEMP1        / NUMBER OF TIMES TO WRITE DATA BLOCK (64*64=4096)
4017 1340      OLOOP, TAD      P4KDAT /PICK UP POINTER TO THE 4K DATA (64 WORDS)
4020 3010      DCA      AUTO10      /SAVE IN AUTO INC REG 10
4021 1135      TAD      K7700        /SET UP COUNTER FOR THE INNER LOOP
4022 3126      DCA      TEMP2        / NUMBER OF WORDS IN DATA BLOCK
4023 6211      ILOOP, CDF          /CHANGE TO FIELD OF DATA PATTERN
4024 1410      TAD I   AUTO10      /PICK UP A DATUM
4025 6221      CDF      20          /FIELD OF TEST
4026 3411      DCA I   AUTO11      /DEPOSIT DATUM
4027 2126      ISZ      TEMP2        /DONE WITH 64 WORD DATA BLOCK?
4030 5223      JMP      ILOOP       /NO--GO BACK
4031 2125      ISZ      TEMP1        /DONE 64 DATA BLOCKS FOR ENTIRE 4K?
4032 5217      JMP      OLOOP       /NO--GO BACK
4033 6201      CDF      0           /CHANGE DATA FIELD BACK TO HERE

/THE 4K BUFFER IN FIELD 2 HAS NOW BEEN SET UP
/ISSUE WRITE COMMAND TO START DATA BREAKS FROM MEMORY INTO THE SILO
4034 4424      RLDC                      /CLEAR THE DEVICE--MA,WC=0
4035 1337      TAD      K4325        /PICK UP WRITE FROM FIELD 2 COMMAND
4036 4430      RLCB                      /ISSUE IT
4037 4464      WAIT                      /LET DMA'S HAVE TIME TO START
4040 4464      WAIT
4041 3125      DCA      TEMP1        /CLEAR OUT COMBINATION POINTER AND COUNTER
4042 6221      CDF      20          /CHANGE TO FIELD OF DATA
4043 6615      IOT15A, 6615          /READ THE SILO
4044 7041      CIA                      /NEGATE IT
4045 1525      TAD I   TEMP1        /ADD IN VALUE IT SHOULD BE
4046 7440      SZA                      /SKIP IF CORRECT
4047 5254      JMP      .+5          /ERROR--REPORT IT BELOW
4050 2125      ISZ      TEMP1        /BUMP POINTER--SKIP IF DONE WITH ALL 4K
4051 5243      JMP      IOT15A      /CONTINUE READING SILO
4052 6201      CDF      0           /DONE--CHANGE BACK TO THIS FIELD
4053 5266      JMP      END4KT      /GO TO END OF 4K TRANSFER TEST
4054 7041      CIA                      /NEGATE THE VALUE IN THE AC -(SILO-CORRECTVAL)
4055 1525      TAD I   TEMP1        /ADD IN CORRECT VALUE (DATA FIELD IS STILL 2)
4056 3114      DCA      DATA2      /SAVE VALUE READ FROM SILO; (CORRECTVAL-(SILO-CORRECTVAL))=SILO
4057 1525      TAD I   TEMP1        /GET CORRECT VALUE AND SAVE IT FOR ERROR TYPEOUT
4060 6201      CDF      0           /BACK TO THIS FIELD FINALLY
4061 3113      DCA      DATA1      /SAVE THE CORRECT EXPECTED VALUE
4062 1125      TAD      TEMP1        /GET THE BREAK ADDRESS
4063 3115      DCA      DATA3      /SAVE FOR ERROR TYPEOUT
4064 4451      ERROR                      /DATA ERROR IN 4K WRITE FROM FIELD 2
4065 4261      DER4KW                      /PC      EXPCTD  ACTUAL  MA(EXPCTD)
4066 4444      END4KT, STOPF          /CLEAR DEVICE TO TERMINATE DATA BREAKS
4067 7240      STA                      /INHIBIT LOOPING ON THIS TEST
4070 3121      DCA      LUPCNT
4071 4450      SCOPE

```

```

/ DATA BREAK TEST TO FIELD 3
/
4072 1131      TAD      MEMSIZ      /GET MEMORY SIZE
4073 1341      TAD      M15        /SUBTRACT 13K
4074 7710      SPA CLA      /SKIP IF ANY OF FIELD 3 IS AVAILABLE
4075 5736      JMP I  PPEOP      /GO TO END OF PASS
4076 4475      BRKVFY      /VERIFY DATA BREAK TO FIELD
4077 0003      3            /FIELD ARGUMENT
4100 4450      SCOPE

/ DATA BREAK TEST TO FIELD 4
/
4101 1131      TAD      MEMSIZ      /GET MEMORY SIZE
4102 1342      TAD      M21        /SUBTRACT 17K
4103 7710      SPA CLA      /SKIP IF ANY OF FIELD 4 IS AVAILABLE
4104 5736      JMP I  PPEOP      /GO TO END OF PASS
4105 4475      BRKVFY      /VERIFY DATA BREAK TO FIELD
4106 0004      4            /FIELD ARGUMENT
4107 4450      SCOPE

/ DATA BREAK TEST TO FIELD 5
/
4110 1131      TAD      MEMSIZ      /GET MEMORY SIZE
4111 1343      TAD      M25        /SUBTRACT 21K
4112 7710      SPA CLA      /SKIP IF ANY OF FIELD 5 IS AVAILABLE
4113 5736      JMP I  PPEOP      /GO TO END OF PASS
4114 4475      BRKVFY      /VERIFY DATA BREAK TO FIELD
4115 0005      5            /FIELD ARGUMENT
4116 4450      SCOPE

/ DATA BREAK TEST TO FIELD 6
/
4117 1131      TAD      MEMSIZ      /GET MEMORY SIZE
4120 1344      TAD      M31        /SUBTRACT 25K
4121 7710      SPA CLA      /SKIP IF ANY OF FIELD 6 IS AVAILABLE
4122 5736      JMP I  PPEOP      /GO TO END OF PASS
4123 4475      BRKVFY      /VERIFY DATA BREAK TO FIELD
4124 0006      6            /FIELD ARGUMENT
4125 4450      SCOPE

/ DATA BREAK TEST TO FIELD 7
/
4126 1131      TAD      MEMSIZ      /GET MEMORY SIZE
4127 1345      TAD      M35        /SUBTRACT 29K
4130 7710      SPA CLA      /SKIP IF ANY OF FIELD 7 IS AVAILABLE
4131 5736      JMP I  PPEOP      /GO TO END OF PASS
4132 4475      BRKVFY      /VERIFY DATA BREAK TO FIELD
4133 0007      7            /FIELD ARGUMENT
4134 4450      END37, SCOPE
4135 5736      JMP I  PPEOP

4136 4200      PPEOP, EOP
4137 4325      K4325, 4325
4140 4556      P4KDAT, DATA4K

```

4141 7763 M15, -15
 4142 7757 M21, -21
 4143 7753 M25, -25
 4144 7747 M31, -31
 4145 7743 M35, -35

4200 PAGE

```

/END OF PASS ROUTINE.
/IF ON MULTIPLE OPTION TESTER, SELECT NEXT OPTION THAT HAS NOT FAILED
/PREVIOUSLY. (HANG UP IF ALL HAVE FAILED.) IF ALL HAVE BEEN TESTED,
/SELECT FIRST ONE AND REPORT END OF PASS. OTHERWISE, SKIP END OF PASS
/AND BEGIN TESTING NEXT OPTION.
/THE REMAINDER OF THE ROUTINE TYPES OUT END PASS MESSAGE AND
/TRANSFERS CONTROL BACK TO THE IOT CODE CHANGE ROUTINE IN PREPARATION
/FOR THE NEXT PASS. SWITCH REGISTER BITS AFFECTING THIS ROUTINE ARE:
/
/   SR3   =0   LOOP PROGRAM
/         =1   HALT AT END OF PASS
/   SR6   =0   REPORT END OF PASS
/         =1   INHIBIT EOP MESSAGE
/   SR8   =0   TEST 16 UNITS IF ON MULTIPLE OPTION TESTER
/         =1   TEST 8 UNITS ON MULTIPLE OPTION TESTER
/               (UNITS 0,2,4,6,10,12,14,16 OCTAL WILL BE TESTED)
/
4200 7332 EOP,   CLA STL RTR      /GET MASK FOR MULTI OPTION BIT IN THE
4201 0022      AND   HCW2      /  HARDWARE CONFIGURATION WORD 2
4202 7650      SNA CLA      /SKIP IF ON MULTI-OPTION TESTER
4203 5227      JMP   INCPAS    /IGNORE MULTI-OPTION GARBAGE; GO INC PASS COUNT
4204 4445      GETSR          /GET SWITCHES
4205 0145      AND   K10      /SAVE SR8
4206 7640      SZA CLA      /SKIP IF USING 16 UUT'S
4207 6371      6371          /INCREMENT SELECT COUNTER (TWICE)
4210 6371      6371          /INCREMENT SELECT COUNTER FOR NEXT UUT
4211 6375      6375          /READ SELECT COUNTER BACK
4212 7640      SZA CLA      /SKIP IF AM BACK TO FIRST OPTION
4213 5224      JMP   NXTOPT    /GO CHECK THE FAILURE FLAG FOR THIS OPTION
4214 4255 FALCHK, JMS   MULFAL  /CHECK FAILURE FLAG FOR THIS OPTION
4215 5227      JMP   INCPAS    /FAILURE FLAG OK--GO REPORT END OF PASS
4216 4445      GETSR          /GET SWITCHES
4217 0145      AND   K10      /SAVE SR8
4220 7640      SZA CLA      /SKIP IF USING 16 UUT'S
4221 6371      6371          /INCREMENT SELECT COUNTER (TWICE)
4222 6371      6371          /INCREMENT SELECT COUNTER FOR NEXT UUT
4223 5214      JMP   FALCHK    /CHECK IF FAILURE FLAG SET FOR THIS UUT (BUT REMEMBER THAT
/      EOP MUST BE REPORTED WHEN A GOOD ONE IS FOUND)
4224 4255 NXTOPT, JMS   MULFAL  /SKIP IF FAILURE FLAG IS SET FOR THIS UUT
4225 5654      JMP I   PSCOPI   /UUT OK--DON'T REPORT EOP; GO TEST THIS DEVICE
4226 5204      JMP   EOP+4      /TRY NEXT HIGHER UUT
4227 2130 INCPAS, ISZ   PASCNT   /INCREMENT PASS COUNT
4230 5233      JMP   .+3
4231 7240      STA          /FREEZE PASS COUNT AT
4232 3130      DCA   PASCNT    /  7777 AFTER 4097 PASSES
4233 4446      APTCHK          /SKIP IF NOT ON APT
4234 5653      JMP I   NEWPAS   /DONT TYPE OR HALT IF ON APT
4235 4445      GETSR          /GET SWITCH REGISTER

```

```

4236 7002      BSW          /GET SR6 INTO SIGN BIT
4237 7510      SPA          /SKIP IF BIT IS CLEAR
4240 5246      JMP          .+6 /INHIBIT TYPEOUT
4241 4500      MESSAGE      /TYPE "END PASS "
4242 0445      EOPMES
4243 1130      TAD          PASCNT /GET PASS COUNT
4244 4503      PRNT4        /TYPE PASS NUMBER
4245 4506      CRLF         /CARRIAGE RETURN, LINE FEED
4246 4445      GETSR       /GET SWITCHES
4247 0152      AND          K400  /MASK ALL BUT SWITCH 3
4250 7640      SZA CLA      /SKIP IF NO HALT DESIRED
4251 4453      CNTRLG      /"HALT"--GO TO CONTROL-G ROUTINE
4252 5653      JMP I       NEWPAS /TRANSFER CONTROL BACK TO BEGINNING

/
4253 0316      NEWPAS, IOTCNG
4254 0340      PSCOPI, SCOPIN

/JMS MULFAL
/
/ON MULTIPLE OPTION TESTER, MAKE NORMAL RETURN IF FAILURE FLAG IS NOT SET.
/IF IT IS SET, CHECK IF THE ERROR COUNT FOR THIS UNIT IS LESS THAN 3.
/IF SO, MAKE NORMAL RETURN (SELECT UNIT), IF NOT, SKIP ON RETURN (REJECT
/UNIT).
/
4255 0000      MULFAL, 0
4256 6374      6374        /SKIP IF FAILURE FLAG IS SET
4257 5267      JMP          MFRET /MAKE NORMAL RETURN
4260 6375      6375        /READ SELECT COUNTER
4261 1270      TAD          PERTB2 /ADD IN AS INDEX TO MAKE POINTER TO ERROR TABLE ENTRY
4262 3125      DCA          TEMP1  /SAVE POINTER
4263 7346      STA CLL      RTL    /-3
4264 1525      TAD I        TEMP1  /COMPARE ERROR COUNT TO 3
4265 7700      SMA CLA      /SKIP IF ERROR COUNT LESS THAN 3 (MAKE OK RETURN)
4266 2255      ISZ          MULFAL /SKIP ON RETURN
4267 5655      MFRET, JMP I  MULFAL
/
4270 6572      PERTB2, ERRTAB

/JMS I MOTINI
/
/SUBROUTINE INITIALIZES MULTIPLE OPTION TESTER, IF PRESENT, AND CLEARS
/ERROR COUNT TABLE (IF ON MULT. OPT. TESTER).
/
4271 0000      INITMO, 0
4272 7332      CLA STL      RTR    /GET MASK FOR MULTIPLE OPTION TESTER BIT
4273 0022      AND          HCW2   /FROM HARDWARE CONFIGURATION WORD 2
4274 7650      SNA CLA      /SKIP IF ON MULTIPLE OPTION TESTER
4275 5307      JMP          IMORET /RETURN
4276 6373      6373        /CLEAR SELECT COUNTER AND ALL FAIL FLAGS
4277 7240      STA
4300 1270      TAD          PERTB2 /SET UP POINTER TO ERROR TABLE
4301 3010      DCA          AUTO10
4302 1156      TAD          M20    /SET UP COUNTER FOR TABLE ENTRIES
4303 3125      DCA          TEMP1
4304 3410      DCA I        AUTO10 /CLEAR AN ENTRY

```

```

4305 2125          ISZ      TEMP1      /DONE 16?
4306 5304          JMP      .-2        /NO
4307 5671  IMORET, JMP I  INITMO

```

4400

PAGE

/SCOPE

/

```

/THIS ROUTINE IS CALLED AT THE END OF EACH SUBTEST. APT TIMING
/IS GENERATED AND CONSOLE INPUT IS PROCESSED. A DECISION IS THEN
/MADE WHETHER TO LOOP ON THE SUBTEST OR CONTINUE AT THE NEXT ONE.
/SWITCH REGISTER BITS AFFECTING THIS ROUTINE ARE (IN ORDER OF PRIORITY):

```

```

/      SR2      =1      LOOP ON CURRENT TEST
/      SR1      =1      LOOP ON ERROR (HARD OR SOFT)
/      =0        DO NOT LOOP IF ERROR (OVERRIDEN IF SR2=1)

```

```

/UNDER NORMAL CONDITIONS (NO ERROR,SR2=0) 4096 SUBTEST ITERATIONS ARE
/PERFORMED. THESE ITERATIONS ARE INHIBITED ON THE FIRST PASS. THE SUBTESTS
/MAY ALSO INHIBIT THIS BY SETTING "LUPCNT" TO -1. NOTE, HOWEVER, THAT IN THIS CASE,
/IF LOOPING ON ERROR OR ON TEST, LUPCNT WILL = -1 ON SUBSEQUENT LOOPS. IF USING
/A PATTERN GENERATOR IN SUCH A TEST, LUPCNT SHOULD BE CLEARED AT THE BEGINNING OF
/THE TEST.

```

/CALL TICK;

/IF CONSOLE-INPUT THEN CALL CONSOL;

/IF SR2 = 1

/ THEN RETURN POINTER = LOOPPT;

/ ELSE IF ERRFLG NOT= 0

/ THEN IF SR1 = 1

/ THEN RETURN POINTER = LOOPPT;

/ ELSE DO;

/ LOOPPT = RETURN POINTER;

/ ERRFLG = 0;

/ END;

/ ELSE IF PASCNT NOT= 0

/ THEN DO;

/ LUPCNT=LUPCNT+1;

/ IF LUPCNT = 0

/ THEN DO;

/ LOOPPT = RETURN POINTER;

/ ERRFLG = 0;

/ END;

/ ELSE RETURN POINTER = LOOPPT;

/ END;

/ ELSE DO;

/ LOOPPT = RETURN POINTER;

/ ERRFLG = 0;

/ END;

/RETURN VIA RETURN POINTER;

/

4400 0000

XSCOPE, 0

4401 4447

TICK

/GENERATE APT TIMING

4402 6031

KSF

/CONSOLE INPUT?

4403 5206

JMP

.+3

4404 4507

VT278

/SET TTY FLAG FOR VT278

4405 4452

CONSOL

/YES--PROCESS IT

4406 4445

GETSR

/GET SWITCHES (HARD OR PSEUDO)

```

4407 7006          RTL          /GET SR1 INTO LINK, SR2 INTO SIGN BIT
4410 7710          SPA CLA      /SKIP IF SR2 = 0
4411 5232          JMP          /LOOP ON TEST--GO SET RETURN POINTER
4412 1123          TAD          /GET ERROR FLAG
4413 7650          SNA CLA      /SKIP IF ERROR OCCURRED
4414 5220          JMP          /NO ERROR--CHECK WHICH PASS IT IS
4415 7430          SZL          /SKIP IF SR1 = 0
4416 5232          JMP          /LOOP ON ERROR--SET RETURN POINTER
4417 5225          JMP          /DO NOT LOOP--SET LOOP POINTER
4420 1130          PASCHK, TAD  /GET PASS COUNT
4421 7650          SNA CLA      /SKIP IF NOT FIRST PASS
4422 5225          JMP          /INHIBIT FIRST PASS ITERATIONS--GO SET LOOP POINTER
4423 2121          ISZ          /INCREMENT LOOPCOUNTER
4424 5232          JMP          /NOT THROUGH LOOPING--SET RETURN POINTER
4425 1200          LUPSET, TAD  /PICK UP RETURN POINTER
4426 3122          DCA          /SAVE IT AT LOOP POINTER
4427 3121          DCA          /CLEAR LOOP COUNTER
4430 3123          DCA          /CLEAR ERROR FLAG
4431 5234          JMP          .+3
4432 1122          RETSET, TAD  /PICK UP LOOP POINTER
4433 3200          DCA          /PUT IT IN RETURN POINTER
4434 4424          RLDC         /ISSUE A DEVICE CLEAR
4435 5600          JMP I       /RETURN VIA RETURN POINTER

```

```

/ERROR
/

```

```

/THIS ROUTINE IS CALLED BY A SUBTEST WHEN AN ERROR HAS BEEN DETECTED.
/THE PC OF THE ERROR CALL IS STORED AT "ERRPC" FOR TYPE-OUT. IF ON
/MULTIPLE OPTION TESTER, SET FAILURE FLAG AND EITHER TEST NEXT OPTION OR
/TELL APT, DEPENDING ON STATE OF SR7. IF ON APT, NOTIFY APT
/OF THE ERROR. THE ARGUMENT IN THE LOCATION FOLLOWING THE ERROR
/CALL IS A POINTER TO AN ENTRY IN THE ERROR TABLE. EACH ENTRY IN THE ERROR
/TABLE CONSISTS OF THREE WORDS: 1) POINTER TO ERROR MESSAGE, 2) POINTER
/TO DATA HEADER, AND 3) MINUS THE NUMBER OF DATA ITEMS TO TYPE. THE
/DATA ITEMS START AT LOCATION "ERRPC". THE ERROR PC IS ALWAYS TYPED.
/

```

```

/SWITCH REGISTER BITS AFFECTING THIS ROUTINE ARE:
/

```

```

/      SRO      =0      HALT ON ERROR
/              =1      INHIBIT ERROR HALT
/      SR4      =0      TYPEOUT ERROR
/              =1      INHIBIT ERROR REPORT
/      SR7      =0      IF ON MULTI-OPTION TESTER, SELECT NEXT UUT AND CONTINUE TESTING
/              =1      IF ON MULTI-OPTION TESTER AND ON APT, NOTIFY APT OF ERROR
/

```

```

/GET ERROR PC;

```

```

/IF ON MULTIPLE OPTION TESTER

```

```

/      THEN DO;

```

```

/          SET FAIL FLAG
/          IF ON APT

```

```

/          THEN IF SR7=0

```

```

/              THEN      SET ERROR COUNT FOR UNIT TO 3 (DROP UNIT)

```

```

/              GO TO EOP

```

```

/              ELSE      GO TO UVPRM
/

```



```

/
/      ELSE      INCREMENT ERROR COUNT FOR UNIT
/      ELSE TYPE UNIT #
/
/      END;
/IF ON-APT THEN GO TO UVPROM;
/IF SR4=0
/      THEN DO;
/
/      GET POINTER TO ERROR TABLE ENTRY;
/      GET ENTRY FROM ERROR TABLE;
/      IF ENTRY NOT= 0 THEN TYPE MESSAGE;
/      UPDATE POINTER TO TABLE;
/      GET ENTRY FROM ERROR TABLE;
/      IF ENTRY NOT= 0 THEN TYPE MESSAGE;
/      UPDATE POINTER TO TABLE;
/      GET ENTRY FROM ERROR TABLE;
/      SET POINTER TO DATA AREA (ERRPC-1);
/      DO I=(ENTRY FROM TABLE) TO -1;
/          INC POINTER TO DATA AREA;
/          GET DATUM;
/          TYPE IT (PRNT4);
/
/      END;
/
/      END;
/IF SRO=0 THEN HALT;
/SET ERRFLG;
/IF (ON MULTIPLE OPTION TESTER) AND (UNIT FAIL COUNT > 3) THEN GO TO EOP(DROP UNIT)
/RETURN;
/
4436 0000      XERROR, 0
4437 7240      CLA CMA      /SUBTRACT 1 FROM RETURN
4440 1236      TAD      XERROR      /ADDRESS TO GET ERROR PC
4441 3112      DCA      ERRPC      /SAVE ERROR PC
4442 1022      TAD      HCW2      /GET HARDWARE CONFIGURATION WORD 2 AND PUT
4443 7004      RAL      /APT INTO LINK, MULTI-OPT TESTER INTO ACO
4444 7700      SMA CLA      /SKIP IF ON MULTIPLE OPTION TESTER
4445 5267      JMP      ERRCHA      /AM NOT--GO CHECK IF ON APT
4446 6372      6372      /SET THE FAILURE FLAG
/
/      /THE NEXT 3 INSTRUCTIONS WILL NOT AFFECT LINK (APT INFO STILL IN IT)
4447 6375      6375      /GET UNIT THAT FAILED
4450 1350      TAD      PERTAB      /INDEX INTO TABLE
4451 3347      DCA      ETABPT      /SAVE POINTER TO ERROR COUNT FOR UNIT
4452 7420      SNL      /SKIP IF ON APT
4453 5263      JMP      MULTER      /HANDLE ERROR COUNT STUFF
4454 4445      GETSR      /GET SWITCHES
4455 0151      AND      K20      /CHECK SR7
4456 7640      SZA CLA      /SKIP IF NOT TO NOTIFY APT
4457 5337      JMP      APTERR      /GO NOTIFY APT
4460 7325      CLA IAC STL RAL      /3 INTO AC
4461 3747      DCA I      ETABPT      /FLAG TO DROP UNIT
4462 5746      JMP I      PEOP      /CONTINUE WITH NEXT UUT
4463 2747      MULTER, ISZ I      ETABPT      /INCREMENT ERROR COUNT FOR FAILED UNIT
4464 4506      CRLF      /TYPE <CR><LF>
4465 6375      6375      /GET SELECT COUNTER
4466 4502      PRNT2      /AND TYPE IT
4467 4446      ERRCHA, APTCHK      /SKIP IF NOT ON APT
4470 5337      JMP      APTERR      /GO TELL APT
4471 4445      GETSR      /GET (HARD OR PSEUDO) SWITCH REG

```

4472	0147	AND	K200	/MASK ALL BUT SR4
4473	7640	SZA CLA		/SKIP IF OK TO TYPE
4474	5315	JMP	HLTCHK	/INHIBIT ERROR REPORT
4475	1636	TAD I	XERROR	/GET ARGUMENT
4476	3342	DCA	TABENT	/SAVE POINTER TO TABLE ENTRY
4477	4351	JMS	ERRTYP	/TYPE MESSAGE FROM TABLE ENTRY
4500	4351	JMS	ERRTYP	/TYPE DATA HEADER FROM TABLE ENTRY
4501	6211	CDF	10	/DATA FIELD OF ERROR TABLE
4502	1742	TAD I	TABENT	/GET NUMBER OF DATA ITEMS
4503	6201	CDF	0	/BACK TO HERE
4504	3343	DCA	DATCNT	/SAVE COUNT
4505	1344	TAD	DATTAB	/PICK UP POINTER TO DATA TABLE
4506	3010	DCA	AUTO10	/PUT IN AUTO-INC REGISTER
4507	1410	DATLUP, TAD I	AUTO10	/PICK UP DATUM
4510	4503	PRNT4		/TYPE DATA
4511	4504	SPACE2		/FORMAT TO NEXT TAB STOP
4512	2343	ISZ	DATCNT	/TYPED ALL NUMBERS?
4513	5307	JMP	DATLUP	/NO--GET NEXT DATUM
4514	4506	CRLF		/YES--CARRIAGE RETURN
4515	4445	HLTCHK, GETSR		/GET SWITCHES
4516	7700	SMA CLA		/SRO=0
4517	4453	CNTRLG		/YES--"HALT" (GO TO CONTROL-G ROUTINE)
4520	7001	IAC		/SET AC NON-ZERO
4521	3123	DCA	ERRFLG	/SET ERROR FLAG
4522	1022	TAD	HCW2	/GET HARDWARE CONFIGURATION WORD 2 AND PUT
4523	7004	RAL		/ MULTI-OPT TESTER INTO ACO
4524	7700	SMA CLA		/SKIP IF ON MULTIPLE OPTION TESTER
4525	5335	JMP	NOTMOT	
4526	7346	STA CLL	RTL	/-3 INTO AC
4527	1747	TAD I	ETABPT	/COMPARE ERROR COUNT TO 3
4530	7710	SPA CLA		/SKIP IF TOO MANY ERRORS
4531	5335	JMP	NOTMOT	/CONTINUE TESTING
4532	4500	MESSAGE		/TYPE DROPPING UNIT MESSAGE
4533	4514	DRPUNI		
4534	5746	JMP I	PEOP	/CONTINUE WITH NEXT UNIT
4535	2236	NOTMOT, ISZ	XERROR	/SKIP OVER ARGUMENT
4536	5636	JMP I	XERROR	/AND RETURN W/AC CLEAR
4537	1112	APTERR, TAD	ERRPC	/GET ERROR PC
4540	6272	CIF	70	/INSTRUCTION FIELD OF UVPRM
4541	5745	JMP I	PAPTER	/GO TELL APT
4542	0000	TABENT, 0		/USED AS POINTER TO ERROR TABLE ENTRY
4543	0000	DATCNT, 0		/USED TO COUNT NUMBER OF DATA ITEMS TYPED
4544	0111	DATTAB, ERRPC-1		/POINTER TO DATA ENTRIES
4545	6520	PAPTER, 6520		/POINTER TO APT'S ERROR ROUTINE
4546	4200	PEOP, EOP		
4547	0000	ETABPT, 0		/INDEXED POINTER INTO ERROR TABLE STORED HERE
4550	6572	PERTAB, ERRTAB		/POINTER TO ERROR TABLE (USED ONLY ON
		/		/MULTIPLE OPTION TESTER)
4551	0000	ERRTYP, 0		
4552	6211	CDF	10	/FIELD OF ERROR TABLE
4553	1742	TAD I	TABENT	/PICK UP TABLE ENTRY
4554	6201	CDF	0	/BACK TO HERE

4555	7450	SNA		/SKIP IF VALID POINTER
4556	5362	JMP	..+4	/NULL ENTRY, IGNORE IT
4557	3361	DCA	..+2	/PLACE POINTER AS MESSAGE ARGUMENT
4560	4500	MESSAGE		/TYPE MESSAGE VIA POINTER FROM ERROR TABLE
4561	0000	0		/POINTER GETS PLACED HERE
4562	2342	ISZ	TABENT	/POINT TO NEXT TABLE ENTRY
4563	5751	JMP I	ERRTYP	/RETURN TO W/AC CLEAR

/SUBROUTINE SKIPS IF NOT RUNNING UNDER APT

4564	0000	XAPTCH, 0		
4565	7200	CLA		
4566	1022	TAD	HCW2	/GET HARDWARE CONFIGURATION WORD 2
4567	7700	SMA CLA		/SKIP IF ON APT (NORMAL RETURN)
4570	2364	ISZ	XAPTCH	/SKIP ON RETURN (NOT ON APT)
4571	5764	JMP I	XAPTCH	/RETURN W/AC CLEAR

4600

PAGE

/CONSOL

/THIS ROUTINE IS USED TO PROCESS KEYBOARD INPUT. IF OUTPUT IS DISABLED, THEN
/ONLY CONTROL-Q AND CONTROL-C ARE RECOGNIZED, IF OUTPUT IS ENABLED, THEN
/IN ADDITION, CONTROL-G AND CONTROL-S ARE RECOGNIZED. THESE CHARACTERS
/HAVE THE FOLLOWING CONTROL FEATURES:

/ CNTRL-G -- ENTER SWITCH REGISTER MODIFICATION ROUTINE
/ CNTRL-C -- RETURN TO OS/8 MONITOR
/ CNTRL-S -- INHIBIT ALL FURTHER OUTPUT. ONLY CNTRL-Q OR CNTRL-C
/ WILL BE RECOGNIZED. (IF THE PROGRAM ATTEMPTS TO
/ OUTPUT, IT WILL HANG WAITING FOR A CNTRL-Q OR C.)

/ CNTRL-Q -- ENABLE OUTPUT
/ CNTRL-F -- ADJUST FILLER CHARACTER COUNT
/ALL OTHER CHARACTERS ARE ECHOED, FOLLOWED BY "?" AND CR LF.

/IF NOT ON-APT OR NO INPUT
/ THEN IF OUTPUT-OFF

THEN DO;

GET CHARACTER;

IF CHARACTER = ^Q

THEN TURN OUTPUT ON;

ELSE IF CHARACTER = ^C

THEN DO;

TURN OUTPUT ON;

TYPE ^C;

GOTO OS/8;

END;

END;

ELSE DO;

GET CHARACTER;

IF CHARACTER = ^G

THEN DO;

TYPE ^G;

CALL CNTRLG;

END;

ELSE IF CHARACTER = ^Q

```

    THEN;
ELSE IF CHARACTER = ^C
THEN DO;
        TYPE "^C";
GOTO OS/8;
END;
ELSE IF CHARACTER = ^S
THEN TURN OUTPUT OFF;
ELSE IF CHARACTER = ^F
THEN CALL CNTRLF;
ELSE TYPE "?<CR><LF>";
END;

/RETURN;
XCONSO, 0

4600 0000 XCONSO, 0
4601 6031 KSF /SKIP IF KEYBOARD INPUT
4602 5244 JMP CONRET /NO INPUT--RETURN
4603 4446 APTCHK /SKIP IF NOT ON APT
4604 5244 JMP CONRET /ON APT--IGNORE KEYBOARD
4605 4507 VI278 /SET TTY FLAG FOR VT278
4606 1127 TAD NOPRNT /PICK UP OUTPUT DISABLE FLAG
4607 7650 SNA CLA /OUTPUT DISABLED?
4610 5215 JMP ONLISN /NO---GOTO ONLISN
4611 4477 LISN /YES--GET CHARACTER FROM KEYBOARD
4612 4643 CNTRLQ /IF ^Q GO TO CNTRLQ
4613 0000 0 /IF NOT ^Q OR ^C THEN
4614 4644 CONRET / RETURN
4615 4477 ONLISN, LISN /GET CHARACTER
4616 7571 -207 /CNTRL-G
4617 4635 CTLG
4620 7555 -223 /CNTRL-S
4621 4641 CNTRLS /GO DISABLE OUTPUT
4622 7557 -221 /CNTRL-Q
4623 4644 CONRET /OUTPUT IS ALREADY ON, SO RETURN
4624 7572 -206 /IF CONTROL-F GO TO
4625 4633 CTLF / CTLF
4626 0000 0 /IF INVALID CHARACTER THEN
4627 4630 .+1 / TYPE OUT
4630 4500 MESSAGE / "?" FOLLOWED BY
4631 0416 QESMRK / <CR><LF>
4632 5244 JMP CONRET / AND RETURN
4633 4245 CRLF, JMS CNTRLF /CALL CNTRL-F HANDLER
4634 5244 JMP CONRET /RETURN
4635 4500 CTLG, MESSAGE /TYPE "G<CR><LF>"
4636 0455 UPARRG
4637 4453 CNTRLG /GO TO CONTROL G HANDLER
4640 5244 JMP CONRET /RETURN
4641 2127 CNTRLS, ISZ NOPRNT /SET NOPRINT FLAG
4642 5244 JMP CONRET /RETURN
4643 3127 CNTRLQ, DCA NOPRNT /CLEAR NOPRINT FLAG
4644 5600 CONRET, JMP I XCONSO /RETURN W/AC CLEAR

/CNTRLF

```

/CONTROL-F HANDLER. CALLED WITH "JMS CNTRLF" WHEN CNTRL-F IS RECEIVED
/FROM CONSOL.

```

4645 0000 CNTRLF, 0
4646 4500 MESSAGE /TYPE "FILL = "
4647 0457 FILLEQ
4650 1023 TAD FILLER /GET FILL COUNT
4651 4502 PRNT2 /PRINT TWO OCTAL DIGIT FILL COUNT
4652 4504 SPACE2 /SPACE OVER FOR INPUT WAIT
4653 3363 DCA DIGITS /CLEAR LOCATION FOR STORING OCTAL INPUT
4654 7346 CLA CLL CMA RTL /-3 INTO AC
4655 3364 DCA DIGCNT /SAVE IN COUNTER FOR NUMBER OF DIGITS
4656 4477 FILLIN, LISN /WAIT FOR INPUT
4657 7572 -206 /IF CNTRL-F, THEN
4660 4646 CNTRLF+1 / REISSUE PROMPT
4661 7563 -215 /IF CARRIAGE RETURN
4662 4675 FIXFIL / GO TO FIXFIL
4663 0001 1 /IF OCTAL INPUT
4664 4672 FILOCT /GO HANDLE IT
4665 0000 0 /UNMATCHED INPUT
4666 4667 .+1
4667 4500 MESSAGE /TYPE "?<CR>"
4670 0416 QESMRK
4671 5246 JMP CNTRLF+1 /GO START OVER
4672 4347 FILOCT, JMS NEWOCT /GO HANDLE OCTAL NUMBER JUST INPUT
4673 5267 JMP .-4 /TOO MANY DIGITS HAVE BEEN INPUT
4674 5256 JMP FILLIN /DIGIT HAS BEEN SAVED--WAIT FOR MORE INPUT
4675 7325 FIXFIL, CLA CLL CML IAC RAL /3 INTO AC
4676 1364 TAD DIGCNT /WERE ANY DIGITS INPUT?
4677 7650 SNA CLA /SKIP IF YES
4700 5303 JMP .+3 /NO--LEAVE FILL UNCHANGED
4701 1363 TAD DIGITS /PICK UP VALUE ENTERED
4702 3023 DCA FILLER /SAVE NEW FILL COUNT
4703 5645 JMP I CNTRLF /RETURN W/ AC CLEAR

```

/CNTRLG

/

/CONTROL-G HANDLER. CALLED WITH "CNTRLG" WHENEVER CONTROL-G IS
/RECEIVED FROM CONSOLE, OR ON PROGRAM "HALT", OR AT BEGINNING OF PROGRAM
/IF SOFTWARE SWITCHES USED.

```

4704 0000 XCNTRL, 0
4705 4500 MESSAGE /TYPE "SR="
4706 0465 SWRMMSG
4707 4445 GETSR /GET SWITCH REG VALUE INTO AC
4710 4503 PRNT4 /TYPE OUT OCTAL VALUE
4711 3363 DCA DIGITS /CLEAR LOCATION FOR STORING OCTAL INPUT
4712 1172 TAD M5 /SET UP COUNTER FOR FOUR OCTAL
4713 3364 DCA DIGCNT / DIGITS
4714 4477 INPUT, LISN /WAIT FOR KEYBOARD INPUT
4715 7571 -207 /IF CONTROL-G, THEN
4716 4705 XCNTRL+1 / REISSUE PROMPT
4717 7563 -215 /ON <CR>
4720 4737 RESTSR /GO TO RESTSR (RESTORE SWR AND RETURN)

```

```

4721 7566      -212      /ON LINE FEED
4722 4735      LNFEED    /GO TO LNFEED
4723 0001      1        /ON OCTAL INPUT
4724 4732      SROCT     /GO TO SROCT
4725 0000      0        /ON UNMATCHED INPUT
4726 4727      .+1      /CONTINUE
4727 4500      MESSAGE   /TYPE "?" <CR><LF>
4730 0416      QESMRK
4731 5305      JMP      XCNTL+1  /START OVER WITH PROMPT
4732 4347      SROCT, JMS     NEWOCT /GO HANDLE OCTAL INPUT
4733 5327      JMP      .-4      /TOO MANY DIGITS ENTERED
4734 5314      JMP      INPUT    /GO GET MORE INPUT
4735 1147      LNFEED, TAD     K200 /PICK UP STARTING ADDRESS OF
4736 3304      DCA      XCNTL    /PROGRAM AND PLACE FOR STARTUP
4737 1346      RESTSR, TAD     K5  /FIND OUT IF NEW DIGITS WERE
4740 1364      TAD      DIGCNT   /TYPED IN
4741 7650      SNA CLA        /SKIP IF NEW VALUE TYPED IN
4742 5345      JMP      .+3      /RETURN WITHOUT CHANGING VALUE
4743 1363      TAD      DIGITS   /PICK UP NEW SR VALUE
4744 3020      DCA      PSWR     /PUT IT IN PSEUDO SWITCH REG
4745 5704      JMP I  XCNTL     /RETURN W/AC CLEAR
/
4746 0005      K5,      5

/NEWOCT
/
/THIS SUBROUTINE TAKES THE OCTAL VALUE IN THE AC AND PLACES IT IN
/WITH PREVIOUS DIGITS ENTERED. IF TOO MANY DIGITS HAVE BEEN ENTERED
/SO FAR (AS SPECIFIED BY DIGCNT), THEN A NORMAL RETURN IS MADE. IF
/THE VALUE IS VALID, THE NUMBER IS PLACED AND THE SUBROUTINE SKIPS
/ON RETURN.
4747 0000      NEWOCT, 0
4750 2364      ISZ      DIGCNT    /SKIP IF TOO MANY DIGITS
4751 7410      SKP      /OK--CONTINUE
4752 5362      JMP      NEWRET    /MAKE RETURN
4753 3125      DCA      TEMP1     /SAVE OCTAL DIGIT
4754 1363      TAD      DIGITS    /GET PREVIOUS DIGITS INTO AC.
4755 7104      CLL RAL        /ROTATE DIGITS TO THE
4756 7006      RTL        /LEFT
4757 1125      TAD      TEMP1     /ADD IN NEW DIGIT
4760 3363      DCA      DIGITS    /RESTORE DIGITS
4761 2347      ISZ      NEWOCT    /SKIP ON RETURN
4762 5747      NEWRET, JMP I  NEWOCT /RETURN W/ AC CLEAR
/
4763 0000      DIGITS, 0
4764 0000      DIGCNT, 0      /COUNTER FOR NUMBER OF DIGITS TYPED IN

/GETSR
/
/SUBROUTINE TO READ SWITCH REG OR PSEUDO-SR INTO AC, DEPENDING ON
/STATE OF BIT 0 IN LOCATION 21
/
4765 0000      XGETSR, 0
4766 7200      CLA
4767 1021      TAD      HCW1      /PICK UP HARDWARE CONFIGURATION WORD 1

```

4770	7710	SPA CLA		/SKIP IF USING SOFTWARE SWITCHES
4771	7614	LAS SKP		/READ HARDWARE SWITCHES
4772	1020	TAD	PSWR	/READ PSEUDO SWR
4773	5765	JMP I	XGETSR	/RETURN W/SR VALUE IN AC

5000 PAGE
/PRINT THE TWO OCTAL NUMBERS IN THE AC 6 THRU 11

5000	0000	XPRNT2, 0		/CALL BY "PRNT2"
5001	3211	DCA	P2SAVE	
5002	1211	TAD	P2SAVE	
5003	7012	RTR		
5004	7010	RAR		
5005	4501	PRNT1		
5006	1211	TAD	P2SAVE	
5007	4501	PRNT1		
5010	5600	JMP I	XPRNT2	
5011	0000	P2SAVE, 0		

/TYPE THE ASCII CHARACTER IN THE AC

5012	0000	XTYPE, 0		/CALL BY "TYPE"
5013	3255	DCA	CHARSV	/SAVE CHARACTER TO BE TYPED
5014	1127	CHKOK, TAD	NOPRNT	
5015	7650	SNA CLA		/OK TO PRINT?
5016	5221	JMP	+.3	/YES
5017	4452	CONSOL		/WAIT FOR INPUT
5020	5214	JMP	.-4	/CHECK IF OUTPUT REENABLED
5021	6031	KSF		/SKIP IF READER FLAG SET
5022	5242	JMP	OK2TYP	/NOT SET, TYPE DATA
5023	4507	VT278		/SET TTY FLAG FOR VT278
5024	6034	KRS		/READ KEYBOARD--LEAVE FLAG SET
5025	0176	AND	K177	/MASK OUT PARITY BIT
5026	1147	TAD	K200	/ADD IN PARITY BIT
5027	1163	TAD	M223	/ADD IN MINUS CONTROL-S
5030	7440	SZA		/SKIP IF A CONTROL-S
5031	5235	JMP	+.4	/GO CHECK FOR CNTRL-C
5032	2127	ISZ	NOPRNT	/SET NO PRINT FLAG
5033	6032	KCC		/CLEAR THE FLAG SINCE CHARACTER HAS BEEN PROCESSED
5034	5214	JMP	CHKOK	/GO GET IN CONTROL-Q WAIT LOOP
5035	1151	TAD	K20	/CHECK FOR CONTROL-C
5036	7640	SZA CLA		/SKIP IF CONTROL-C
5037	5242	JMP	OK2TYP	/GO TYPE THE CHARACTER
5040	6032	KCC		/CLEAR THE KEYBOARD FLAG SINCE THE CHARACTER WAS PROCESSED
5041	5371	JMP	CNTRL C	/HANDLE CONTROL-C
5042	1255	OK2TYP, TAD	CHARSV	/GET CHARACTER
5043	6046	TL S		
5044	7200	CLA		
5045	6041	TSF		
5046	7410	SKP		/GO CHECK FOR POWER LOW
5047	5253	JMP	+.4	/CHARACTER IS FINISHED
5050	6102	SPL		/SKIP ON POWER LOW
5051	5245	JMP	.-4	/GO WAIT FOR FLAG
5052	4000	JMS	0	/POWER FAIL INTERRUPT

```

5053 6042      TCF
5054 5612      JMP I   XTYPE
5055 0000      /
                CHARSV, 0

```

```

/TYPE A CR AND LF WITH NUMBER OF FILLERS
/AS DETERMINED BY LOCATION "FILLER"

```

```

5056 0000      XCRLF, 0                      /CALL BY "CRLF"
5057 7200      CLA
5060 1272      TAD      K215
5061 4505      TYPE
5062 1023      TAD      FILLER
5063 7040      CMA
5064 3274      DCA      CRLFSV
5065 1273      TAD      K212
5066 4505      TYPE
5067 2274      ISZ      CRLFSV
5070 5266      JMP      -2
5071 5656      JMP I   XCRLF
5072 0215      K215, 0215
5073 0212      K212, 0212
5074 0000      CRLFSV, 0

```

```

/PRINT 2 SPACES

```

```

5075 0000      SPACX2, 0                    /CALL BY "SPACE2"
5076 4500      MESSAGE
5077 0471      TWOSPA
5100 5675      JMP I   SPACX2

```

```

/COMPARE INPUT TO LIST FOLLOWING CALL
/INPUT ONE CHARACTER IF AC=0
/USE LAST INPUT IF AC NON ZERO

```

```

5101 0000      XLISN, 0                      /CALL BY "LISN"
5102 7640      SZA CLA
5103 5334      JMP      LISN1                /USE LAST INPUT SINCE AC NOT ZERO
5104 6031      KSF
5105 5304      JMP      -1
5106 6036      KRB
5107 0176      AND      K177
5110 1147      TAD      K200
5111 3370      DCA      LASTIN
5112 1367      TAD      M203
5113 1370      TAD      LASTIN
5114 7650      SNA CLA
5115 5371      JMP      CNTRLC
5116 1127      TAD      NOPRNT
5117 7640      SZA CLA
5120 5334      JMP      LISN1
5121 1370      TAD      LASTIN

```

```

/CHECK IF CHARACTER WAS A CONTROL-C

```

```

/SKIP IF NOT
/GO HANDLE CONTROL-C
/CHECK OUTPUT ENABLE FLAG
/SKIP IF OK TO OUTPUT
/DON'T PRINT CHARACTER

```


/PRINT FOUR OCTAL NUMBERS IN AC 11 THRU 0 FOLLOWED
/BY TWO SPACES

5400	0000	XPRNT4, 0		/CALL BY "PRNT4"
5401	3211	DCA	P4SAVE	
5402	1211	TAD	P4SAVE	
5403	7002	BSW		
5404	4502	PRNT2		
5405	1211	TAD	P4SAVE	
5406	4502	PRNT2		
5407	4504	SPACE2		
5410	5600	JMP I	XPRNT4	
5411	0000	P4SAVE, 0		

/PRINT THE OCTAL NUMBER IN AC 9 THRU 11

5412	0000	XPRNT1, 0		/CALL BY "JMS XPRNT1"
5413	0217	AND	K7	
5414	1220	TAD	K260	
5415	4505	TYPE		
5416	5612	JMP I	XPRNT1	
5417	0007	K7,	0007	
5420	0260	K260,	0260	

/PRINT PACKED ASCII TEXT TERMINATED BY
/SIX-BIT 00

5421	0000	MESAGX, 0		/CALL BY "MESSAGE"
5422	7344	CLA CLL CMA RAL		/-2 INTO AC
5423	3264	DCA MESCNT		/SET UP CHARACTER COUNTER
5424	1621	TAD I MESAGX		
5425	3265	DCA MESSAV		
5426	2221	ISZ MESAGX		/SET UP RETURN
5427	6211	MESLUP, CDF 10		/FIELD OF TEXT
5430	1665	TAD I MESSAV		
5431	6201	CDF 0		/BACK TO HERE
5432	2264	ISZ MESCNT		/FIRST OR SECOND CHARACTER OF WORD
5433	5242	JMP MESBSW		/FIRST-PUT CHARACTER INTO LOW BYTE
5434	2265	ISZ MESSAV		/2ND--INCREMENT POINTER FOR NEXT TIME
5435	3126	DCA TEMP2		/SAVE AC
5436	7344	CLA CLL CMA RAL		/RESTORE CHARACTER COUNTER
5437	3264	DCA MESCNT		
5440	1126	TAD TEMP2		
5441	7410	SKP		
5442	7002	MESBSW, BSW		/GET FIRST CHARACTER INTO POSITION
5443	0261	AND K77		
5444	7450	SNA		/TERMINATOR (00)?
5445	5621	JMP I MESAGX		/YES
5446	1262	TAD M43		
5447	7450	SNA		/CRLF?
5450	5257	JMP .+7		/YES
5451	1153	TAD K3		
5452	7510	SPA		/200 OR 300
5453	1140	TAD K100		/300
5454	1263	TAD K240		/200
5455	4505	TYPE		

```

5456 7410      SKP
5457 4506      CRLF
5460 5227      JMP      MESLUP
5461 0077      K77,    0077
5462 7735      M43,    7735
5463 0240      K240,   0240
5464 0000      MESCNT, 0
5465 0000      MESSAV, 0

```

/TIMING GENERATOR FOR NOTIFYING APT SYSTEM. TICKS SHOULD BE
/SPACED AT LEAST ONCE EVERY 1 MS AND NOT MORE OFTEN THAN EVERY 40 US
/WHEN NOT CALLING SCOPE

```

5466 0000      XTICK,  0
5467 4446      APTCHK
5470 7410      SKP
5471 5276      JMP      TIKRET
5472 2301      ISZ      TOCK
5473 5276      JMP      TIKRET
5474 6272      CIF      70
5475 4702      JMS I    PAPTIM
5476 6102      TIKRET, SPL
5477 5666      JMP I    XTICK
5500 4000      JMS      0

5501 0000      TOCK,   0
5502 6500      PAPTIM, 6500

```

/SKIP IF NOT ON APT
/ON APT -- GENERATE TIMING
/DON'T GENERATE TIMING
/COUNT A TIME UNIT
/NOT READY FOR APT NOTIFICATION
/FIELD OF UV-PROM
/NOTIFY APT
/SKIP ON POWER LOW
/RETURN
/POWER FAIL INTERRUPT

/TIME UNIT COUNTER
/POINTER TO APT'S TIMING ROUTINE

/IOT SUBROUTINES

/CLEAR DEVICE, ALL REGISTERS, AC, AND FLAGS

```

5503 0000      XRLDC,  0
5504 6600      IOT0,   6600
5505 5703      JMP I    XRLDC
5506 4451      ERROR
5507 3572      RLDCSK
5510 5703      JMP I    XRLDC

```

/ISSUE DEVICE CLEAR
/RETURN (AC CLEAR)
/RLDC CAUSED SKIP
/PC

/SKIP ON FUNCTION DONE, THEN CLEAR IF SET TO A ONE

```

5511 0000      XRLSD,  0
5512 6601      IOT1,   6601
5513 7410      SKP
5514 2311      ISZ      XRLSD
5515 5711      JMP I    XRLSD

```

/SKIP IF DEVICE DONE
/NORMAL RETURN IF NO SKIP
/INCREMENT RETURN ADDR FOR SKIP
/RETURN (AC UNCHANGED)

/LOAD BREAK MA REG FROM AC

```

5516 0000      XRLMA,  0
5517 6602      IOT2,   6602
5520 5716      JMP I    XRLMA
5521 4451      ERROR
5522 3607      RLMAK

```

/LOAD MA REGISTER
/RETURN (AC CLEAR)
/RLMA CAUSED SKIP
/PC

```

5523 5716          JMP I   XRLMA
                    /LOAD COMMAND REGISTER A FROM AC
                    /
5524 0000          XRLCA, 0
5525 6603          IOT3, 6603          /LOAD COMMAND REG A
5526 5724          JMP I   XRLCA          /RETURN (AC CLEAR)
5527 4451          ERROR          /RLCA CAUSED SKIP
5530 3624          RLCASK          /PC
5531 5724          JMP I   XRLCA

                    /LOAD COMMAND REGISTER B FROM AC
                    /
5532 0000          XRLCB, 0
5533 6604          IOT4, 6604          /LOAD COMMAND REG B
5534 5732          JMP I   XRLCB          /RETURN (AC CLEAR)
5535 4451          ERROR          /RLCB CAUSED SKIP
5536 3641          RLCBSK          /PC
5537 5732          JMP I   XRLCB

                    /LOAD SECTOR ADDRESS REGISTER FROM AC 0:5
                    /
5540 0000          XRLSA, 0
5541 6605          IOT5, 6605          /LOAD SECTOR ADDRESS REG
5542 5740          JMP I   XRLSA          /RETURN (AC CLEAR)
5543 4451          ERROR          /RLSA CAUSED SKIP
5544 3656          RLSASK          /PC
5545 5740          JMP I   XRLSA

                    /"SPARE" IOT
                    /
5546 0000          XSPARE, 0
5547 6606          IOT6, 6606          /ISSUE SPARE IOT
5550 5746          JMP I   XSPARE          /RETURN (AC CLEAR)
5551 4451          ERROR          /SPARE IOT CAUSED SKIP
5552 3710          SPARSK          /PC
5553 5746          JMP I   XSPARE

                    /LOAD WORD COUNT FROM AC
                    /
5554 0000          XRLWC, 0
5555 6607          IOT7, 6607          /LOAD WORD COUNT
5556 5754          JMP I   XRLWC          /RETURN (AC CLEAR)
5557 4451          ERROR          /RLWC CAUSED SKIP
5560 3673          RLWCSK          /PC
5561 5754          JMP I   XRLWC

                    /READ ERROR REGISTER INTO AC 0:2
                    /
5562 0000          XRRER, 0
5563 6610          IOT10, 6610          /READ ERROR REGISTER
5564 5762          JMP I   XRRER          /RETURN (AC=E000)
5565 4451          ERROR          /RRER CAUSED SKIP
5566 3727          RRERSK          /PC
5567 5762          JMP I   XRRER

```

```

5600 PAGE
/READ WORD COUNT REG INTO AC
/
5600 0000 XRRWC, 0
5601 6611 IOT11, 6611 /READ WORD COUNT
5602 5600 JMP I XRRWC /RETURN W/WC IN AC
5603 4451 ERROR /RRWC CAUSED SKIP
5604 3744 RRWSCSK /PC
5605 5600 JMP I XRRWC

/READ COMMAND REG A INTO AC
/
5606 0000 XRRCA, 0
5607 6612 IOT12, 6612 /READ COMMAND REG A INTO AC
5610 5606 JMP I XRRCA /RETURN W/CA IN AC
5611 4451 ERROR /RRCA CAUSED SKIP
5612 3761 RRCASK /PC
5613 5606 JMP I XRRCA

/READ COMMAND REG B INTO AC
/
5614 0000 XRRCB, 0
5615 6613 IOT13, 6613 /READ COMMAND REG B
5616 5614 JMP I XRRCB /RETURN W/ INFO IN AC
5617 4451 ERROR /RRCB CAUSED SKIP
5620 3776 RRCBSK /PC
5621 5614 JMP I XRRCB

/READ SECTOR ADDRESS REGISTER INTO AC 0:5
/
5622 0000 XRRSA, 0
5623 6614 IOT14, 6614 /READ SECTOR ADDR
5624 5622 JMP I XRRSA /RETURN (AC=SA00)
5625 4451 ERROR /RRSA CAUSED SKIP
5626 4013 RRSASK /PC
5627 5622 JMP I XRRSA

/READ SILO WORD INTO AC 4:11
/
5630 0000 XRRSI, 0
5631 6615 IOT15, 6615 /READ SILO
5632 5630 JMP I XRRSI /RETURN (AC 0:3 CLEAR; 4:11=SILO)
5633 4451 ERROR /RRSI CAUSED SKIP
5634 4030 RRSISK /PC
5635 5630 JMP I XRRSI

/SKIP ON DRIVE READY, THEN CLEAR IF SET TO A ONE
/
5636 0000 XRLSR, 0
5637 6616 IOT16, 6616 /SKIP ON DRIVE READY
5640 7410 SKP /NORMAL (NO SKIP) RETURN
5641 2236 ISZ XRLSR /INC RETURN ADDR FOR SKIP
5642 5636 JMP I XRLSR /RETURN (AC UNCHANGED)

```

```

/SKIP ON DRIVE ERROR, THEN CLEAR IF SET TO A ONE
/
5643 0000 XRLSE, 0
5644 6617 IOT17, 6617 /SKIP ON DRIVE ERROR
5645 7410 SKP /NORMAL RETURN (NO SKIP)
5646 2243 ISZ XRLSE /INC RETURN ADDR FOR SKIP
5647 5643 JMP I XRLSE /RETURN (AC UNCHANGED)

/RESETW
/
/THIS SUBROUTINE ISSUES A "RESET" COMMAND TO THE CONTROLLER AND WAITS
/LONG ENOUGH FOR "DONE" TO SET. (THE DONE FLAG IS NOT ACTUALLY CHECKED.)
/
5650 0000 XESETW, 0
5651 7200 CLA
5652 1141 TAD K7301 /PICK UP MAINTENANCE RESET COMMAND
5653 4430 RLCB /ISSUE RESET
5654 4464 WAIT /WAIT A BIT
5655 5650 JMP I XESETW /RETURN W/ AC CLEAR

/RESET
/
/THIS SUBROUTINE ISSUES A "RESET" COMMAND TO THE CONTROLLER AND WAITS
/FOR THE DONE FLAG TO SET.
/
5656 0000 XRESET, 0
5657 7200 CLA
5660 1141 TAD K7301 /PICK UP MAINTENANCE RESET COMMAND
5661 4430 RLCB /ISSUE RESET
5662 4425 RLSD /WAIT FOR DONE
5663 4473 JMPPM1
5664 5656 JMP I XRESET /RETURN W/ AC CLEAR

/RESET3
/
/THIS SUBROUTINE ISSUES A "RESET" COMMAND TO DRIVE 3 AND WAITS
/FOR THE DONE FLAG TO SET.
/
5665 0000 XESET3, 0
5666 7200 CLA
5667 1142 TAD K301 /PICK UP RESET COMMAND
5670 4430 RLCB /ISSUE RESET
5671 4425 RLSD /WAIT FOR DONE
5672 4473 JMPPM1
5673 5665 JMP I XESET3 /RETURN W/ AC CLEAR

/MRSET3
/
/THIS SUBROUTINE ISSUES A MAINTENANCE RESET COMMAND TO DRIVE 3 WITH THE
/INTERFACE ENABLED AND WAITS FOR THE DONE FLAG TO SET.
/
5674 0000 XMRSET, 0
5675 7200 CLA
5676 1303 TAD K3301 /PICK UP MAINTENANCE RESET COMMAND

```

```

5677 4430          RLCB          /ISSUE RESET
5700 4425          RLSD          /WAIT FOR DONE
5701 4473          JMPPM1
5702 5674          JMP I   XMRSET /RETURN W/ AC CLEAR
/
5703 3301          K3301, 3301
/POWER FAIL ROUTINES
/
5704 7200          PWRFAL, CLA
5705 1000          TAD          0          /GET PC
5706 3113          DCA          DATA1      /SAVE FOR REFERENCE OR ERROR TYPE OUT
5707 6102          SPL
5710 5314          JMP          IMPINT      /SKIP ON POWER LOW
5711 1317          TAD          JMPUP      /IMPOSSIBLE INTERRUPT--GO REPORT IT
5712 3000          DCA          0          /GET RESTART INSTRUCTION
5713 7402          HLT
/
5714 4451          IMPINT, ERROR          /IMPOSSIBLE INTERRUPT (INTERRUPT SYSTEM IS OFF)!
5715 0252          FATAL
5716 5547          JMP I   K200          / PC   INTPC
/RESTART PROGRAM
/
5717 5403          JMPUP, JMP I   3          /JUMPS TO POWER UP ROUTINE
/
5720 6035          PWRUP, KIE
5721 6045          TIE
5722 3000          DCA          0          /DISABLE KEYBOARD INTERRUPTS INPUT ONLY
5723 4446          APTCHK
5724 5332          JMP          NOPMES      /DISABLE CONSOLE INTERRUPTS OUTPUT ONLY
5725 4500          MESSAGE
5726 1627          POWER
5727 1113          TAD          DATA1      /CLEAR THE JMP I 3 INSTRUCTION
5730 4503          PRNT4
5731 4506          CRLF
5732 1122          NOPMES, TAD          LOOPPT /SKIP IF NOT ON APT
5733 7650          SNA CLA
5734 5547          JMP I   K200          /SKIP THE POWER FAIL MESSAGE
5735 5522          JMP I   LOOPPT          /POWER FAIL PC=
/GET PC OF POWER FAIL
/PRINT IT
/GET LOOP POINTER
/SKIP IF POINTER IS VALID
/RESTART PROGRAM FROM BEGINNING
/RESTART INTERRUPTED TEST
/
/DAREXP
/
/THIS SUBROUTINE COMPUTES THAT PORTION OF THE DAR THAT DOES NOT DEPEND ON THE
/COMMAND BEING ISSUED (8 BIT MODE IS ASSUMED). THE TWO WORDS ARE COMPUTED AS
/THEY ARE EXPECTED IN THE SILO. RETURN IS MADE WITH PORTION OF SECOND WORD
/IN AC. IF MARKER & OTHER BITS ARE EXPECTED TO BE SET (BASED ON COMMAND),
/THEN THEY MUST BE MASKED AND ADDED BACK IN ON RETURN.
/
5736 0000          XDAREX, 0
5737 4435          RRCA
5740 0365          AND          K777          /GET PATTERN BACK
5741 7110          CLL RAR
5742 3113          DCA          DATA1      /MASK ALL BUT CYLINDER ADDRESS DIFFERENCE
5743 7006          RTL
5744 7002          BSW
/ROTATE RIGHT ONCE--LINK NOW HAS CA11
/PLACE 8 BIT WORD EXPECTED IN DAR
/GET CA11 INTO AC10
/PUT LSB OF CYL ADDR DIFF (CA11) INTO POSITION

```

```

5745 3114      DCA      DATA2      /SAVE 8 BIT WORD FORMED SO FAR
5746 4437      RRSA                      /GET SECTOR ADDRESS REG
5747 7002      BSW                      /GET SECTOR ADDRESS INTO POSITION
5750 1114      TAD      DATA2      /ADD IN TO WORD FORMED SO FAR
5751 3114      DCA      DATA2      /SAVE WORD SO FAR
5752 4435      RRCA                      /GET CA REG
5753 7004      RAL                      /PUT DIRECTION BIT (CA0) INTO LINK, HEAD
                                   /SELECT (CA1) INTO SIGN BIT
                                   /SKIP IF HEAD SELECT=0
5754 7710      SPA CLA                      /SET BIT TO BE LOADED WITH CA1
5755 1140      TAD      K100              /GET WORD FORMED SO FAR INTO AC
5756 1114      TAD      DATA2          /SKIP IF DIRECTION BIT WAS 1
5757 7420      SNL                      /DON'T SET SIGN BIT
5760 5363      JMP      .+3              /MASK OUT SIGN BIT FROM DAR WORD
5761 0364      AND      K7773          /ADD IN SIGN BIT TO WORD
5762 1157      TAD      K4              /RETURN WITH PORTION OF SECOND WORD IN AC
5763 5736      JMP I    XDAREX

```

```

5764 7773      K7773, 7773
5765 0777      K777, 777

```

```

/ WAIT -- THIS SUBROUTINE TAKES APPROX. 19 USEC TO EXECUTE ON AN 8/E.
/ IT CAN BE USED TO WAIT FOR DONE ON MAINT RESET OR FOR TIMING PURPOSES
/ (SUCH AS FOR LETTING CONTROLLER MICROCODE RUN OR FOR APT TIMING).
/

```

```

5766 0000      XWAIT, 0
5767 1400      TAD I    0                /EXECUTE ANY OLD INSTRUCTION
5770 0400      AND I    0                /THAT TAKES A FEW U-SECS.
5771 0400      AND I    0
5772 7200      CLA
5773 5766      JMP I    XWAIT            /RETURN WITH AC CLEAR

```

6000 PAGE

/XFRCHK

```

/ THIS SUBROUTINE COMPARES THE VALUE OF THE SILO WORDS (2) WITH THE
/ WORDS STORED AT DATA1 AND DATA2. IF EQUAL, A NORMAL RETURN IS MADE,
/ ELSE THE ROUTINE SKIPS ON RETURN. IF COMMAND 8 INDICATES TRANSFER WAS IN
/ 8 BIT MODE, THE GARBAGE BITS ARE MASKED OUT.
/

```

```

6000 0000      XXFRCH, 0
6001 4436      RRCB                      /GET MODE BIT INTO
6002 7006      RTL                      /SIGN BIT OF AC
6003 7710      SPA CLA                      /SKIP IF TWELVE BIT MODE
6004 1230      TAD      K7400          /GET COMPLEMENT OF GARBAGE MASK
6005 7040      CMA                      /MASK IS EITHER 7777 OR 0377
6006 3126      DCA      TEMP2          /SAVE MASK
6007 4440      RRSI                      /READ SILO WORD
6010 0126      AND      TEMP2          /MASK WORD
6011 3116      DCA      DATA4          /SAVE ACTUAL WORD1
6012 4440      RRSI                      /GET SECOND SILO WORD
6013 0126      AND      TEMP2          /MASK WORD
6014 3115      DCA      DATA3          /SAVE ACTUAL WORD 2
6015 1113      TAD      DATA1          /SUBTRACT EXPECTED WORD1
6016 7041      CIA                      /FROM ACTUAL WORD1

```



```

6017 1115      TAD      DATA3      /IF RESULT=0
6020 7440      SZA              /THEN OK--SO SKIP
6021 5226      JMP      .+5          /MAKE ERROR RETURN
6022 1114      TAD      DATA2      /COMPARE SECOND WORD IN
6023 7041      CIA              /SAME FASHION
6024 1116      TAD      DATA4
6025 7640      SZA CLA          /SKIP IF EQUAL--GOOD RETURN
6026 2200      ISZ      XXFRCHK      /TRANSFER FAILED--SKIP ON RETURN
6027 5600      JMP I      XXFRCHK

```

6030 7400 K7400, 7400

/OPIWAT

```

/THIS SUBROUTINE WAITS FOR OPI OR DONE TO OCCUR.  A PROGRAM TIMING LOOP
/IS USED TO PREVENT THE PROGRAM FROM HANGING SHOULD NEITHER OCCUR.
/THE WAIT IS APPROXIMATELY 552 MS ON AN 8/E. (SLIGHTLY LONGER IF RUNNING
/UNDER APT.) ON OPI, THE RETURN ADDRESS IS INCREMENTED BY ONE. ON DONE,
/THE RETURN ADDRESS IS INCREMENTED BY TWO. THIS CAUSES THE FOLLOWING RETURN PATTERN:
/

```

```

/      OPIWAT      /SUBROUTINE CALL
/      CALL+1      /NEITHER OPI NOR DONE
/      CALL+2      /OPI BUT NOT DONE
/      CALL+3      /DONE BUT NOT OPI
/      CALL+4      /DONE AND OPI

```

6031	0000	XOPIWA,	0		
6032	7200		CLA		
6033	3271		DCA	OPICNT	/INITIALIZE TIMER
6034	4464	OPIUP,	WAIT		/WAIT A BIT
6035	4464		WAIT		/AND A BIT MORE
6036	4464		WAIT		
6037	4464		WAIT		
6040	4464		WAIT		
6041	4425		RLSD		/HAS DONE SET?
6042	7410		SKP		/NO--CONTINUE
6043	5254		JMP	DNOUT	/YES--GET OUT OF WAIT LOOP
6044	4433		RRER		/GET ERROR REGISTER
6045	7004		RAL		/ROTATE OPI INTO SIGN BIT
6046	7710		SPA CLA		/SKIP IF OPI NOT SET
6047	5263		JMP	OPIOUT	/GET OUT OF WAIT LOOP
6050	4447		TICK		/GENERATE APT TIMING
6051	2271		ISZ	OPICNT	/COUNT OPI TIMER
6052	5234		JMP	OPIUP	/WAIT SOME MORE
6053	5270		JMP	OPIRET	/TIME'S UP--RETURN
6054	2231	DNOUT,	ISZ	XOPIWA	/ADD TWO TO RETURN PC
6055	2231		ISZ	XOPIWA	
6056	4433		RRER		/GET ERROR REG
6057	7004		RAL		/PUT OPI INTO SIGN BIT
6060	7710		SPA CLA		/SKIP IF OPI NOT SET
6061	2231		ISZ	XOPIWA	/INCREMENT RETURN ADDRESS
6062	5270		JMP	OPIRET	/RETURN
6063	2231	OPIOUT,	ISZ	XOPIWA	/INCREMENT RETURN ADDRESS
6064	4425		RLSD		/SKIP IF DONE
6065	5270		JMP	OPIRET	/RETURN
6066	2231		ISZ	XOPIWA	/ADD TWO TO RETURN PC

```

6067 2231      ISZ      XOPiWA
6070 5631      OPIRET, JMP I  XOPiWA      /RETURN TO CALL PLUS 1-4
/
6071 0000      OPICNT, 0      /COUNTER FOR TIMING THE OPI WAIT LOOP

/JMPPM1
/
/THIS SUBROUTINE ACTS AS A "JMP .-1" INSTRUCTION EXCEPT THAT THE AC IS
/CLEARED AND CONSOL INPUT CAN BE PROCESSED.
/
6072 0000      XJMPPM, 0
6073 4452      CONSOL      /CHECK FOR CONSOLE INPUT
6074 7344      CLA CLL CMA RAL      /-2 INTO AC
6075 1272      TAD      XJMPPM      /SUBTRACT 2 FROM RETURN PC
6076 3272      DCA      XJMPPM      /SO RETURN IS TO CALL-1
6077 6102      SPL      /SKIP ON POWER LOW
6100 5672      JMP I      XJMPPM      /RETURN TO CALL-1 W/AC CLEAR
6101 4000      JMS      0      /POWER FAIL INTERRUPT

6200          PAGE

/CRCCHK
/FIELD ARGUMENT*10 (E.G. 20)
/
/SUBROUTINE GENERATES CRC BASED ON DATA IN PATTRN. CRC IS GENERATED USING
/16 BITS, STARTING FROM LSB. 0'S ARE FORCED INTO UNUSED MSB POSITIONS.
/THE RESULT IS COMPARED TO THE ACTUAL CRC (POINTED TO BY
/AUTO10 AND THE FIELD ARGUMENT) AND AN ERROR IS REPORTED IF NOT CORRECT.
/IF AN ERROR WAS FOUND, A NORMAL RETURN IS MADE, ELSE THE PROGRAM SKIPS
/ON RETURN (IN ADDITION TO SKIP OVER FIELD ARGUMENT).
/
6200 0000      XCRCH, 0
6201 7200      CLA      /INITIALIZE ALL CRC BITS TO 0
6202 3325      DCA      CRC0      /BITS 0,1,14,15 ARE KEPT SEPERATELY. THEY
6203 3326      DCA      CRC1      /HAVE VALUES OF EITHER 0 OR 4000 (FOR 1).
6204 3327      DCA      CRC14      /THE BIT BEING CHECKED FOR CRC GENERATION
6205 3330      DCA      CRC15      /ALSO HAS A VALUE OF EITHER 0 OR 4000.
6206 3331      DCA      CRC213      /THIS ENABLES ALL XORS TO BE DONE WITH ADDS.
6207 1156      TAD      M20      /SET UP COUNTER FOR 16 BITS
6210 3332      DCA      CRCCNT      /AT LOCATION CRCCNT
6211 1124      TAD      PATTRN      /GET DATA TO GENERATE CRC FROM
6212 3125      DCA      TEMP1      /SAVE FOR ROTATING
6213 4447      CRCLUP, TICK      /GENERATE APT TIMING
6214 7201      CLA IAC      /1 INTO AC FOR MASK
6215 0125      AND      TEMP1      /GET ONLY BIT BEING CHECKED
6216 7112      CLL RTR      /GET BIT INTO SIGN BIT
6217 1330      TAD      CRC15      /XOR BIT BEING CHECKED WITH CRC15
6220 3126      DCA      TEMP2      /SAVE THAT VALUE
6221 1327      TAD      CRC14      /GET OLD CRC BIT 14
6222 1126      TAD      TEMP2      /XOR WITH SAVED VALUE
6223 3330      DCA      CRC15      /THIS BECOMES NEW CRC15
6224 1326      TAD      CRC1      /XOR OLD BIT 1 AND
6225 1126      TAD      TEMP2      /SAVED VALUE
6226 7104      CLL RAL      /PUT RESULT IN LINK, LEAVE AC CLEAR
6227 1331      TAD      CRC213      /GET CRC BITS 2 THRU 13

```

6230	7010	RAR		/ROTATE LINK INTO BIT 2 POSITION, OLD BIT 13 INTO LINK
6231	3331	DCA	CRC213	/SAVE NEW BITS 2-13
6232	7010	RAR		/GET OLD BIT 13 INTO SIGN BIT
6233	3327	DCA	CRC14	/IT BECOMES NEW BIT 14
6234	1325	TAD	CRC0	/GET OLD CRC BIT 0
6235	3326	DCA	CRC1	/IT BECOMES NEW CRC1
6236	1126	TAD	TEMP2	/XOR OF OLD BIT 15 AND INPUT BIT
6237	3325	DCA	CRC0	/BECOMES NEW BIT 0. EASY, HUH?
6240	1125	TAD	TEMP1	/GET DATA WORD
6241	7110	CLL RAR		/ROTATE NEXT BIT TO CHECK INTO POSITION
6242	3125	DCA	TEMP1	/SAVE WORD
6243	2332	ISZ	CRCCNT	/DONE ALL TWELVE BITS?
6244	5213	JMP	CRCLUP	/NO--DO IT ALL AGAIN
		/YES--NOW TAKE BITS AND FORM INTO TWO 8 BIT WORDS		
6245	1326	TAD	CRC1	/PUT BIT 1 INTO LINK
6246	7104	CLL RAL		/AND LEAVE AC CLEAR
6247	1331	TAD	CRC213	/GET BITS 2-13 INTO AC
6250	0135	AND	K7700	/SAVE ONLY BITS 2-7 (BIT 1 STILL IN LINK)
6251	7012	RTR		/PUT INTO POSITION
6252	1325	TAD	CRC0	/ADD IN BIT 0 NEXT TO BIT 1
6253	7012	RTR		/MOVE OVER INTO LSB OF AC
6254	7012	RTR		
6255	3114	DCA	DATA2	/SAVE 8 LSB'S OF CRC
6256	1327	TAD	CRC14	/PUT BIT 14 INTO LINK,
6257	7104	CLL RAL		/LEAVING AC CLEAR
6260	1331	TAD	CRC213	/GET BITS 2-13 INTO AC
6261	0333	AND	K77A	/SAVE ONLY BITS 8-13 (14 STILL IN LINK)
6262	1330	TAD	CRC15	/PUT BIT 15 IN SIGN BIT
6263	7006	RTL		/MOVE OVER
6264	7006	RTL		/PUT IN TOP PART OF WORD
6265	7006	RTL		
6266	3115	DCA	DATA3	/SAVE CRC IN 12 BIT MODE FORMAT
		/NOW COMPARE COMPUTED CRC WITH ACTUAL CRC		
6267	1600	CRCCMP, TAD I	XCRCCH	/GET FIELD ARGUMENT
6270	2200	ISZ	XCRCCH	/SKIP OVER ARGUMENT ON RETURN
6271	1334	TAD	KCDF	/ADD CDF INSTRUCTION INTO FIELD ARGUMENT
6272	3273	DCA	.+1	/SAVE FOR EXECUTION
6273	7000	NOP		/EXECUTE CDF NO
6274	1410	TAD I	AUTO10	/GET ACTUAL CRC WORD
6275	3117	DCA	DATA5	/SAVE IT
6276	1410	TAD I	AUTO10	/GET 2ND CRC WORD
6277	6201	CDF	0	/BACK TO THIS FIELD
6300	3116	DCA	DATA4	/SAVE CRC WORD
6301	1114	TAD	DATA2	/GET EXPECTED CRC
6302	7041	CIA		/SUBTRACT IT FROM
6303	1116	TAD	DATA4	/ACTUAL CRC WORD
6304	7640	SZA CLA		/SKIP IF THEY WERE EQUAL
6305	5313	JMP	.+6	/GO REPORT ERROR
6306	1115	TAD	DATA3	/COMPARE MSB'S OF EXPECTED CRC
6307	7041	CIA		/WITH ACTUAL BY SUBTRACTING
6310	1117	TAD	DATA5	
6311	7650	SNA CLA		/SKIP IF NOT EQUAL
6312	5323	JMP	GDCRRT	/EXIT TEST--SKIP ON RETURN
6313	1124	TAD	PATRN	/GET DATA WORD
6314	3113	DCA	DATA1	/SAVE FOR TYPEOUT

```

6315 7344      CLA CLL CMA RAL      /-2 INTO AC
6316 1200      TAD      XCRCCH      /GET PC OF SUBROUTINE CALL
6317 3120      DCA      DATA6      /SAVE FOR TYPEOUT
6320 4451      ERROR      /CRC ERROR
6321 3203      CRCER      /PC      DATA      MSB-EXPCTD-LSB      MSB-ACTUAL-LSB      CALLPC
6322 7410      SKP      /ERROR RETURN
6323 2200      GDCRRT, ISZ      XCRCCH      /SKIP ON RETURN
6324 5600      JMP I      XCRCCH      /RETURN W/AC CLEAR
/
6325 0000      CRC0, 0
6326 0000      CRC1, 0
6327 0000      CRC14, 0
6330 0000      CRC15, 0
6331 0000      CRC213, 0
6332 0000      CRCCNT, 0
6333 0077      K77A, 77
6334 6201      KCDF, CDF

/ERRCHK
/
/CHECK ERROR FLAG. REPORT ERROR IF SET. SKIP ON RETURN IF NOT SET.
/
6335 0000      XERRCH, 0
6336 4442      RLSE
6337 5352      JMP      NERRET      /SKIP ON ERROR
6340 4433      RRER      /MAKE NO ERROR RETURN
6341 3113      DCA      DATA1      /READ ERROR REG
6342 4436      RRCB      /SAVE FOR ERROR TYPEOUT
6343 3114      DCA      DATA2      /READ COMMAND B
6344 7240      STA      /SAVE FOR ERROR TYPEOUT
6345 1335      TAD      XERRCH      /SUBTRACT 1 FROM RETURN
6346 3115      DCA      DATA3      /PC TO GET CALL PC
6347 4451      ERROR      /SAVE FOR ERROR TYPEOUT
6350 0332      ERFLSS      /ERROR FLAG SET
6351 5354      JMP      .+3      /PC      ER      CB      CALLPC
6352 7200      NERRET, CLA      /MAKE ERROR RETURN
6353 2335      ISZ      XERRCH      /CLEAR AC PRIOR TO RETURN
6354 5735      JMP I      XERRCH      /SKIP ON RETURN
/RETURN W/AC CLEAR

/SLOFIL
/
/SUBROUTINE FILLS THE SILO BY ISSUEING 8 DAR TO SILO RESETS
/
6355 0000      XSLOFI, 0
6356 1170      TAD      M10
6357 3364      DCA      SLOCNT      /SET UP COUNTER FOR
6360 4457      RESET      /EIGHT TRANSFERS
6361 2364      ISZ      SLOCNT      /DO TWO WORD DAR TO SILO TRANSFER
6362 5360      JMP      .-2      /DONE EIGHT TRANSFERS?
6363 5755      JMP I      XSLOFI      /NO--DO ANOTHER
/RETURN W/ AC CLEAR

/
6364 0000      SLOCNT, 0
6400      PAGE

```

```

/BRKVfy
/FIELD ARG
/
/SUBROUTINE VERIFYS DATA BREAK CAN OCCUR TO FIELD SPECIFIED BY ARGUMENT
/
6400 0000 XBRKVF, 0
6401 7240 STA /SET AUTO-INC REG TO ZERO BUFFER
6402 3010 DCA AUTO10
6403 1600 TAD I XBRKVF /GET FIELD ARGUMENT
6404 2200 ISZ XBRKVF /SKIP OVER ARGUMENT ON RETURN
6405 3250 DCA BRKFLD /SAVE ARGUMENT
6406 1250 TAD BRKFLD /GET IT BACK
6407 7106 CLL RTL /ROTATE IT INTO CORRECT POSITION
6410 7004 RAL /FOR CDF AND EMA BITS
6411 3251 DCA FLDEMA /SAVE IT
6412 1251 TAD FLDEMA /FORM A CDF INSTRUCTION TO THE CORRECT
6413 1252 TAD KCDFA /FIELD
6414 3221 DCA CDFINS /SAVE FOR EXECUTION
6415 4454 PAT25 /GET A PATTERN
6416 1250 TAD BRKFLD /ADD FIELD OFFSET
6417 3124 DCA PATTRN /SAVE NEW PATTERN
6420 1124 TAD PATTRN
6421 7000 CDFINS, NOP /EXECUTE CDF TO FIELD OF BREAK
6422 4476 BUFSET /SET UP BUFFER IN THAT FIELD
6423 1164 TAD M4 /SET UP WORD COUNT
6424 4432 RLWC
6425 1144 TAD K300 /SELECT DRIVE 3
6426 1251 TAD FLDEMA /ADD IN EMA BITS
6427 4430 RLCB /EXECUTE FUNCTION 0
6430 4425 RLSD /SKIP ON DONE
6431 4473 JMPPM1 /WAIT FOR DONE
6432 1221 TAD CDFINS /GET CDF INSTRUCTION TO FIELD OF BREAK
6433 3234 DCA .+1 /EXECUTE IT NEXT
6434 7000 NOP /CDF IS PLACED HERE
6435 1567 TAD I K1 /PICK UP WORD FROM WRITE BREAK
6436 6201 CDF 0 /BACK TO HERE
6437 7640 SZA CLA /SKIP IF WORD IS ZERO
6440 5247 JMP BRKRET /BREAK MUST HAVE OCCURED IF WORD IS NON-ZERO
6441 4456 PATCMP /CHECK IF WORD IS SUPPOSED TO BE ZERO
6442 5247 JMP BRKRET /OK SO EXIT
6443 4436 RRCB /READ COMMAND B AND SAVE
6444 3113 DCA DATA1 / FOR ERROR TYPEOUT
6445 4451 ERROR /DATA BREAK DID NOT OCCUR TO CORRECT FIELD
6446 3222 WRNGFL /PC CB
6447 5600 BRKRET, JMP I XBRKVF
/
6450 0000 BRKFLD, 0
6451 0000 FLDEMA, 0
6452 6201 KCDFA, CDF

/BUFSET
/
/SUBROUTINE IS ENTERED WITH DATA FIELD SET TO DESIRED BREAK FIELD, THE PATTERN
/IN THE AC, AND THE ADDRESS OF THE BUFFER-1 IN AUTO10. A FOUR WORD BUFFER IS
/SET UP AND RETURN IS MADE WITH DATA FIELD EQUAL TO 0.

```

```

/
6453 0000 XBUFSE, 0
6454 3410 DCA I AUTO10 /SET PATTERN
6455 3410 DCA I AUTO10 /ZERO OUT WORD FOR WRITE BREAK
6456 3410 DCA I AUTO10 /CLEAR TWO CRC WORDS
6457 3410 DCA I AUTO10
6460 6201 CDF 0 /BACK TO FIELD 0
6461 5653 JMP I XBUFSE

/PAT25
/
/THIS SUBROUTINE PUTS A PATTERN INTO THE AC. THE PATTERN IS 2525 PLUS
/THE SUBTEST LOOP COUNT (CONTROLLED BY SCOPE). THE FIRST PASS THE PATTERN
/WILL BE 2525, AND ON SUBSEQUENT PASSES IT WILL COUNT UP FROM 2525 TO
/2524, DEPENDING ON WHICH ITERATION OF THE SUBTEST THE SUBROUTINE IS
/CALLED FROM.
/
6462 0000 XPAT25, 0
6463 7200 CLA
6464 1271 TAD K2525 /LOAD AC WITH 2525
6465 1121 TAD LUPCNT /ADD LOOP COUNT FROM SCOPE
6466 3124 DCA PATRN /SAVE TEST PATTERN (FOR USE WITH
6467 1124 TAD PATRN / PATCMP)
6470 5662 JMP I XPAT25 /RETURN WITH PATTERN IN AC

/
6471 2525 K2525, 2525

/PAT52
/
/THIS SUBROUTINE IS THE SAME AS PAT25 EXCEPT ITS BASE PATTERN IS 5252.
/
6472 0000 XPAT52, 0
6473 7200 CLA
6474 1301 TAD K5252 /GET PATTERN
6475 1121 TAD LUPCNT /ADD LOOP COUNTER VALUE
6476 3124 DCA PATRN /SAVE PATTERN FOR COMPARISON
6477 1124 TAD PATRN
6500 5672 JMP I XPAT52 /RETURN WITH PATTERN IN AC

/
6501 5252 K5252, 5252

/PATM1
/
/THIS SUBROUTINE LOADS THE AC WITH A TEST PATTERN BASED ON 7777 IN THE
/SAME WAY AS "PAT25". HOWEVER, THE PATTERN IS NOT SAVED IN LOCATION
/"PATRN".
/
6502 0000 XPATM1, 0
6503 7240 CLA CMA /-1 INTO AC
6504 1121 TAD LUPCNT /ADD ITERATION COUNT FOR PATTERN
6505 5702 JMP I XPATM1 /RETURN WITH PATTERN IN AC

/PATCMP
/
/THIS SUBROUTINE COMPARES THE VALUE IN THE AC WITH THE VALUE STORED AT

```

/LOCATION PATRN. IF THE TWO VALUES ARE EQUAL, A NORMAL RETURN IS MADE,
 /ELSE THE VALUE IN PATRN IS PUT AT DATA1 AND THE VALUE IN THE AC AT
 /DATA2 (FOR SUBSEQUENT ERROR TYPEOUT) AND THE ROUTINE SKIPS ON ITS RETURN.
 /

6506	0000	XPATCH, 0		
6507	3126	DCA	TEMP2	/SAVE AC VALUE
6510	1124	TAD	PATRN	/GET COMPARISON VALUE
6511	7041	CIA		/SUBTRACT PATTERN FROM ORIGINAL AC
6512	1126	TAD	TEMP2	
6513	7650	SNA CLA		/SKIP IF THEY WERE NOT EQUAL
6514	5322	JMP	+.6	/MAKE NORMAL RETURN--DO NO SAVE VALUES
6515	1126	TAD	TEMP2	/PICK UP ORIGINAL AC VALUE
6516	3114	DCA	DATA2	/SAVE FOR ERROR TYPEOUT
6517	1124	TAD	PATRN	/PICK UP COMPARISON VALUE
6520	3113	DCA	DATA1	/SAVE FOR ERROR TYPEOUT
6521	2306	ISZ	XPATCH	/SKIP ON RETURN
6522	5706	JMP I	XPATCH	/RETURN WITH AC CLEAR

/MANTEC
 /

/THIS SUBROUTINE CHECKS THAT WRITE DATA LATE IS SET AND THAT THE ERROR FLAG IS
 /SET. IF NOT, AN ERROR IS REPORTED AND THE SUBROUTINE MAKES A NORMAL RETURN.
 /IF BOTH ARE SET, THE SUBROUTINE SKIPS ON RETURN. THIS IS USED TO CHECK FOR THE
 /PROPER CONTROLLER STATE AFTER A MAINT COMMAND.
 /

6523	0000	XMANTE, 0		
6524	4433	RRER		/GET ERROR REG
6525	7006	RTL		/GET DATA LATE INTO SIGN BIT
6526	7710	SPA CLA		/SKIP IF NOT SET
6527	5344	JMP	XEFCB	/GO CHECK THAT ERROR FLAG IS SET
6530	4471	ERRCHK		/CHECK IF ERROR FLAG IS SET AND REPORT
6531	5360	JMP	MANRET	/RETURN--ERROR OCCURED
6532	4433	RRER		/GET ERROR REG AND SAVE FOR TYPEOUT
6533	3113	DCA	DATA1	
6534	4436	RRCB		/SAVE COMMAND B ALSO
6535	3114	DCA	DATA2	
6536	7240	STA		/SUBTRACT 1 FROM RETURN
6537	1323	TAD	XMANTE	/PC TO GET CALL PC
6540	3115	DCA	DATA3	/SAVE FOR ERROR TYPEOUT
6541	4451	ERROR		/DATA LATE DID NOT SET ON MAINT TRANSFER
6542	3432	NOL28		/PC ER CB CALLPC
6543	5360	JMP	MANRET	/RETURN
6544	4442	XEFCB, RLSE		/SKIP ON ERROR FLAG
6545	7410	SKP		/NOT SET--REPORT ERROR
6546	5357	JMP	MANRET-1	/ALL OK--SKIP ON RETURN
6547	4433	RRER		/SAVE ERROR REG FOR TYPEOUT
6550	3113	DCA	DATA1	
6551	7240	STA		/SUBTRACT 1 FROM RETURN
6552	1323	TAD	XMANTE	/PC TO GET CALL PC
6553	3114	DCA	DATA2	/SAVE FOR ERROR TYPEOUT
6554	4451	ERROR		/ERROR FLAG NOT SET BY WRITE DATA LATE FF
6555	3466	EFNSDL		/PC ER CALLPC
6556	7410	SKP		/MAKE ERROR RETURN
6557	2323	ISZ	XMANTE	/SKIP ON RETURN

```

6560 5723  MANRET, JMP I  XMANTE          /RETURN
        /STOPF
        /
        /STOP FUNCTION BEING PERFORMED BY CONTROLLER BY ISSUING 7 DEVICE CLEARS.
        /(A DEVICE CLEAR IS NOT GUARANTEED TO TERMINATE A FUNCTION DUS TO ASYNCHRONOUS
        /NATURE OF CONTROLLER AND NARROWNESS OF DEVICE CLEAR PULSE.) A "CAF" WOULD
        /STOP THE MICRO-SEQUENCER BUT SCREWS UP THE CONSOLE PACKAGE.
        /
6561 0000  XSTOPF, 0
6562 6600  IOT0A, 6600
6563 6600  IOT0B, 6600
6564 6600  IOT0C, 6600
6565 6600  IOT0D, 6600
6566 6600  IOT0E, 6600
6567 6600  IOT0F, 6600
6570 6600  IOT0G, 6600
6571 5761          JMP I  XSTOPF
6572 0000  ERRTAB, ZBLOCK 20              /16 WORDS FOR ERROR COUNTS FOR EACH
                                          /UNIT ON MULT. OPT. TESTER ONLY. UNIT IS
                                          /ALLOWED 3 ERRORS BEFORE BEING DROPPED.
                                          /FAIL LIGHT WILL BE LIT AFTER FIRST FAILURE.

        7400  *7400
7400 0000  BUF0,  ZBLOCK 10

        0001  FIELD 1

```


0000 0000 *0
0000 0000 ZBLOCK 4
0200 *200

/ERROR TABLE

/

/EACH ENTRY IN THE ERROR TABLE CONSISTS OF THREE WORDS:

- / 1) POINTER TO ERROR MESSAGE (MAY BE 0 FOR NULL)
- / 2) POINTER TO DATA HEADER (MAY BE 0 FOR NULL)
- / 3) MINUS NUMBER OF DATA ITEMS (AT LEAST -1 (FOR PC))

/THE ERROR CALL CONSISTS OF THE "ERROR" STATEMENT FOLLOWED BY A

/POINTER TO AN ENTRY IN THE ERROR TABLE

/SEPERATE ENTRIES (3 WORD BLOCKS) IN THE ERROR TABLE ARE NOT REQUIRED

/TO BE GROUPED TOGETHER. THIS HAS BEEN DONE STRICTLY FOR FUNCTIONAL

/CLARITY. ERROR TABLE ENTRIES AND ALL TEXT MUST RESIDE IN FIELD 1.

0200	0603	E1,	ACSKIP	/AC AFFECTED BY A SKIP IOT
0201	0540		EXPACT	/PC EXPCTD ACTUAL
0202	7775		-3	
0203	0621	CA2A,	ACRLCA	/AC NOT CLEARED BY RLCA
0204	0530		PCACT	/PC ACTUAL
0205	7776		-2	
0206	0722	CA2B,	CABAD	/CA DID NOT CONTAIN EXPECTED DATA
0207	0540		EXPACT	/PC EXPCTD ACTUAL
0210	7775		-3	
0211	1032	CA2C,	CACLR	/CA NOT CLEARED BY RLDC
0212	0530		PCACT	/PC ACTUAL
0213	7776		-2	
0214	0636	CB2A,	ACRLCB	/AC NOT CLEARED BY RLCA
0215	0530		PCACT	/PC ACTUAL
0216	7776		-2	
0217	0744	CB2B,	CBBAD	/CB DID NOT CONTAIN EXPECTED DATA
0220	0540		EXPACT	/PC EXPCTD ACTUAL
0221	7775		-3	
0222	1047	CB2C,	CBCLR	/CB NOT CLEARED BY RLDC
0223	0530		PCACT	/PC ACTUAL
0224	7776		-2	
0225	0653	SA2A,	ACRLSA	/AC NOT CLEARED BY RLCA
0226	0530		PCACT	/PC ACTUAL
0227	7776		-2	
0230	0766	SA2B,	SABAD	/SA DID NOT CONTAIN EXPECTED DATA
0231	0540		EXPACT	/PC EXPCTD ACTUAL
0232	7775		-3	
0233	1064	SA2C,	SACLR	/SA NOT CLEARED BY RLDC
0234	0530		PCACT	/PC ACTUAL
0235	7776		-2	
0236	0670	WC2A,	ACRLWC	/AC NOT CLEARED BY RLWC
0237	0530		PCACT	/PC ACTUAL
0240	7776		-2	
0241	1010	WC2B,	WCBAD	/WC DID NOT CONTAIN EXPECTED DATA
0242	0540		EXPACT	/PC EXPCTD ACTUAL
0243	7775		-3	
0244	1101	WC2C,	WCCLR	/WC NOT CLEARED BY RLDC
0245	0530		PCACT	/PC ACTUAL
0246	7776		-2	

0247	0705	MA3,	ACRLMA	/AC NOT CLEARED BY RLMA
0250	0530		PCACT	/PC ACTUAL
0251	7776		-2	
0252	1476	FATAL,	INTOFF	/SOFTWARE ERROR (JMP TO 0 OR IMPOSSIBLE INTERRUPT
0253	1543		PCINT	/ PC INTPC
0254	7776		-2	
0255	1116	SPARE3,	ACSPAR	/AC NOT CLEARED BY "SPARE" IOT
0256	0530		PCACT	/PC ACTUAL
0257	7776		-2	
0260	1136	DC3,	ACRLDC	/AC NOT CLEARED BY RLDC
0261	0530		PCACT	/PC ACTUAL
0262	7776		-2	
0263	1153	CB4,	CBAFF	/CB AFFECTED BY RLCA,RLSA,OR RRSA
0264	0540		EXPACT	/PC EXPCTD ACTUAL
0265	7775		-3	
0266	1176	WC4,	WCAFF	/WC AFFECTED BY "SPARE", OR RRSI
0267	0540		EXPACT	/PC EXPCTD ACTUAL
0270	7775		-3	
0271	1217	SA4,	SAAFF	/SA AFFECTED BY RLWC, "SPARE", OR RRSI
0272	0540		EXPACT	/PC EXPCTD ACTUAL
0273	7775		-3	
0274	1243	SD5B,	DNCLR	/DONE NOT CLEARED BY RLSD
0275	0473		PC	/PC
0276	7777		-1	
0277	1436	NODONE,	DNNSET	/DONE DID NOT SET
0300	0521		PCCB	/PC CB
0301	7776		-2	
0302	1261	SD6,	DNRLDC	/DONE NOT CLEARED BY RLDC
0303	0473		PC	/PC
0304	7777		-1	
0305	1553	SD7,	DNRLCB	/DONE NOT CLEARED BY LOADING CB
0306	0473		PC	/PC
0307	7777		-1	
0310	1277	UN8,	UNKINT	/UNKNOWN INTERRUPT REQUESTED
0311	0473		PC	/PC
0312	7777		-1	
0313	1316	INT8A,	NOINTD	/INTERRUPT REQUEST NOT GENERATED BY DONE
0314	0473		PC	
0315	7777		-1	
0316	1344	INT8B,	INTNDN	/INTERRUPT STILL REQUESTED AFTER DONE CLEARED
0317	0473		PC	
0320	7777		-1	
0321	1375	INT9,	INTNIE	/INTERRUPT REQUESTED WHEN IE NOT SET
0322	0473		PC	
0323	7777		-1	
0324	1420	SIERR,	SICLR	/SILO NOT CLEARED BY RLDC
0325	0473		PC	/PC
0326	7777		-1	
0327	1450	DAR11,	BADDAR	/DAR TO SILO TRANSFER (MAINT MODE) FAILED
0330	0554		PCDAR	/PC H18-EXPCTD-LO8 H18-ACTUAL-LO8 CB
0331	7772		-6	
0332	1641	ERFLSS,	ERRSET	/ERROR FLAG SET
0333	1574		ERCBCL	/ PC ER CB CALLPC
0334	7774		-4	
0335	1641	ERFLGS,	ERRSET	/ERROR FLAG SET

0336 1614 PCERCB
0337 7775 -3

/MORE ERROR TABLE ENTRIES AFTER TEXT BELOW

0340	4301	MANDEC, TEXT	"#AJRLA-D RL8-A DISKLESS CONTROLLER#" /	HP 001
0341	1222			
0342	1401			
0343	5504			
0344	4022			
0345	1470			
0346	5501			
0347	4004			
0350	1123			
0351	1314			
0352	0523			
0353	2340			
0354	0317			
0355	1624			
0356	2217			
0357	1414			
0360	0522			
0361	4300			
0362	4305	MEMPRM, TEXT	"#ENTER AMOUNT OF CONTIGUOUS MEMORY AVAILABLE FOR USE: "	
0363	1624			
0364	0522			
0365	4001			
0366	1517			
0367	2516			
0370	2440			
0371	1706			
0372	4003			
0373	1716			
0374	2411			
0375	0725			
0376	1725			
0377	2340			
0400	1505			
0401	1517			
0402	2231			
0403	4001			
0404	2601			
0405	1114			
0406	0102			
0407	1405			
0410	4006			
0411	1722			
0412	4025			
0413	2305			
0414	7240			
0415	0000			
0416	7743	QESMRK, TEXT	"?#"	

0417	0000		
0420	4315	MEMBIG, TEXT	"#MEMORY SIZE TOO BIG (ENTER 8-32 <CR>): "
0421	0515		
0422	1722		
0423	3140		
0424	2311		
0425	3205		
0426	4024		
0427	1717		
0430	4002		
0431	1107		
0432	4050		
0433	0516		
0434	2405		
0435	2240		
0436	7055		
0437	6362		
0440	4074		
0441	0322		
0442	7651		
0443	7240		
0444	0000		
0445	4305	EOPMES, TEXT	"#END PASS "
0446	1604		
0447	4020		
0450	0123		
0451	2340		
0452	0000		
0453	3603	UPARRC, TEXT	"^C"
0454	4300		
0455	3607	UPARRG, TEXT	"^G"
0456	0000		
0457	3606	FILLEQ, TEXT	"^F#FILL = "
0460	4306		
0461	1114		
0462	1440		
0463	7540		
0464	0000		
0465	4323	SWRMSG, TEXT	"#SR = "
0466	2240		
0467	7540		
0470	0000		
0471	4040	TWOSPA, TEXT	" . "
0472	0000		
0473	4020	PC, TEXT	" PC"
0474	0343		
0475	0000		
0476	4020	PCERCL, TEXT	" PC ER CALLPC"
0477	0340		
0500	4040		
0501	4040		
0502	4005		
0503	2240		
0504	4040		
0505	4003		

0506	0114				
0507	1420				
0510	0343				
0511	0000				
0512	4020	PCER,	TEXT	" PC	ER#"
0513	0340				
0514	4040				
0515	4040				
0516	4005				
0517	2243				
0520	0000				
0521	4020	PCCB,	TEXT	" PC	CB#"
0522	0340				
0523	4040				
0524	4040				
0525	4003				
0526	0243				
0527	0000				
0530	4020	PCACT,	TEXT	" PC	ACTUAL#"
0531	0340				
0532	4040				
0533	4001				
0534	0324				
0535	2501				
0536	1443				
0537	0000				
0540	4020	EXPACT,	TEXT	" PC	EXPCTD ACTUAL#"
0541	0340				
0542	4040				
0543	4005				
0544	3020				
0545	0324				
0546	0440				
0547	4001				
0550	0324				
0551	2501				
0552	1443				
0553	0000				
0554	4020	PCDAR,	TEXT	" PC	H18-EXPCTD-L08 H18-ACTUAL-L08 CB#"
0555	0340				
0556	4040				
0557	4010				
0560	1170				
0561	5505				
0562	3020				
0563	0324				
0564	0455				
0565	1417				
0566	7040				
0567	4010				
0570	1170				
0571	5501				
0572	0324				
0573	2501				
0574	1455				

0575	1417		
0576	7040		
0577	4040		
0600	4003		
0601	0243		
0602	0000		
0603	4301	ACSKIP, TEXT	"#AC AFFECTED BY A SKIP IOT#"
0604	0340		
0605	0106		
0606	0605		
0607	0324		
0610	0504		
0611	4002		
0612	3140		
0613	0140		
0614	2313		
0615	1120		
0616	4011		
0617	1724		
0620	4300		
0621	4301	ACRLCA, TEXT	"#AC NOT CLEARED BY RLCA#"
0622	0340		
0623	1617		
0624	2440		
0625	0314		
0626	0501		
0627	2205		
0630	0440		
0631	0231		
0632	4022		
0633	1403		
0634	0143		
0635	0000		
0636	4301	ACRLCB, TEXT	"#AC NOT CLEARED BY RLCA#"
0637	0340		
0640	1617		
0641	2440		
0642	0314		
0643	0501		
0644	2205		
0645	0440		
0646	0231		
0647	4022		
0650	1403		
0651	0243		
0652	0000		
0653	4301	ACRLSA, TEXT	"#AC NOT CLEARED BY RLCA#"
0654	0340		
0655	1617		
0656	2440		
0657	0314		
0660	0501		
0661	2205		
0662	0440		
0663	0231		

0664	4022		
0665	1423		
0666	0143		
0667	0000		
0670	4301	ACRLWC, TEXT	"#AC NOT CLEARED BY RLWC#"
0671	0340		
0672	1617		
0673	2440		
0674	0314		
0675	0501		
0676	2205		
0677	0440		
0700	0231		
0701	4022		
0702	1427		
0703	0343		
0704	0000		
0705	4301	ACRLMA, TEXT	"#AC NOT CLEARED BY RLMA#"
0706	0340		
0707	1617		
0710	2440		
0711	0314		
0712	0501		
0713	2205		
0714	0440		
0715	0231		
0716	4022		
0717	1415		
0720	0143		
0721	0000		
0722	4303	CABAD, TEXT	"#CA DID NOT CONTAIN EXPECTED DATA#"
0723	0140		
0724	0411		
0725	0440		
0726	1617		
0727	2440		
0730	0317		
0731	1624		
0732	0111		
0733	1640		
0734	0530		
0735	2005		
0736	0324		
0737	0504		
0740	4004		
0741	0124		
0742	0143		
0743	0000		
0744	4303	CBBAD, TEXT	"#CB DID NOT CONTAIN EXPECTED DATA#"
0745	0240		
0746	0411		
0747	0440		
0750	1617		
0751	2440		
0752	0317		

0753	1624	
0754	0111	
0755	1640	
0756	0530	
0757	2005	
0760	0324	
0761	0504	
0762	4004	
0763	0124	
0764	0143	
0765	0000	
0766	4323	SABAD, TEXT "#SA DID NOT CONTAIN EXPECTED DATA#"
0767	0140	
0770	0411	
0771	0440	
0772	1617	
0773	2440	
0774	0317	
0775	1624	
0776	0111	
0777	1640	
1000	0530	
1001	2005	
1002	0324	
1003	0504	
1004	4004	
1005	0124	
1006	0143	
1007	0000	
1010	4327	WCBAD, TEXT "#WC DID NOT CONTAIN EXPECTED DATA#"
1011	0340	
1012	0411	
1013	0440	
1014	1617	
1015	2440	
1016	0317	
1017	1624	
1020	0111	
1021	1640	
1022	0530	
1023	2005	
1024	0324	
1025	0504	
1026	4004	
1027	0124	
1030	0143	
1031	0000	
1032	4303	CACLR, TEXT "#CA NOT CLEARED BY RLDC#"
1033	0140	
1034	1617	
1035	2440	
1036	0314	
1037	0501	
1040	2205	
1041	0440	

1042	0231	
1043	4022	
1044	1404	
1045	0343	
1046	0000	
1047	4303	CBCLR, TEXT "#CB NOT CLEARED BY RLDC#"
1050	0240	
1051	1617	
1052	2440	
1053	0314	
1054	0501	
1055	2205	
1056	0440	
1057	0231	
1060	4022	
1061	1404	
1062	0343	
1063	0000	
1064	4323	SACLR, TEXT "#SA NOT CLEARED BY RLDC#"
1065	0140	
1066	1617	
1067	2440	
1070	0314	
1071	0501	
1072	2205	
1073	0440	
1074	0231	
1075	4022	
1076	1404	
1077	0343	
1100	0000	
1101	4327	WCCLR, TEXT "#WC NOT CLEARED BY RLDC#"
1102	0340	
1103	1617	
1104	2440	
1105	0314	
1106	0501	
1107	2205	
1110	0440	
1111	0231	
1112	4022	
1113	1404	
1114	0343	
1115	0000	
1116	4301	ACSPAR, TEXT /#AC NOT CLEARED BY "SPARE" IOT#/"
1117	0340	
1120	1617	
1121	2440	
1122	0314	
1123	0501	
1124	2205	
1125	0440	
1126	0231	
1127	4042	
1130	2320	

1131	0122		
1132	0542		
1133	4011		
1134	1724		
1135	4300		
1136	4301	ACRLDC, TEXT	"#AC NOT CLEARED BY RLDC#"
1137	0340		
1140	1617		
1141	2440		
1142	0314		
1143	0501		
1144	2205		
1145	0440		
1146	0231		
1147	4022		
1150	1404		
1151	0343		
1152	0000		
1153	4303	CBAFF, TEXT	"#CB AFFECTED BY RLCA, RLSA, OR RRSA#"
1154	0240		
1155	0106		
1156	0605		
1157	0324		
1160	0504		
1161	4002		
1162	3140		
1163	2214		
1164	0301		
1165	5440		
1166	2214		
1167	2301		
1170	5440		
1171	1722		
1172	4022		
1173	2223		
1174	0143		
1175	0000		
1176	4327	WCAFF, TEXT	"/#WC AFFECTED BY "SPARE" OR RRSI#/"
1177	0340		
1200	0106		
1201	0605		
1202	0324		
1203	0504		
1204	4002		
1205	3140		
1206	4223		
1207	2001		
1210	2205		
1211	4240		
1212	1722		
1213	4022		
1214	2223		
1215	1143		
1216	0000		
1217	4323	SAAFF, TEXT	"/#SA AFFECTED BY RLWC, "SPARE", OR RRSI#/"

1220 0140
1221 0106
1222 0605
1223 0324
1224 0504
1225 4002
1226 3140
1227 2214
1230 2703
1231 5440
1232 4223
1233 2001
1234 2205
1235 4254
1236 4017
1237 2240
1240 2222
1241 2311
1242 4300
1243 4304
1244 1716
1245 0540
1246 1617
1247 2440
1250 0314
1251 0501
1252 2205
1253 0440
1254 0231
1255 4022
1256 1423
1257 0443
1260 0000
1261 4304
1262 1716
1263 0540
1264 1617
1265 2440
1266 0314
1267 0501
1270 2205
1271 0440
1272 0231
1273 4022
1274 1404
1275 0343
1276 0000
1277 4325
1300 1613
1301 1617
1302 2716
1303 4011
1304 1624
1305 0522
1306 2225

DNCLR, TEXT "#DONE NOT CLEARED BY RLSD#"

DNRLDC, TEXT "#DONE NOT CLEARED BY RLDC#"

UNKINT, TEXT "#UNKNOWN INTERRUPT REQUESTED#"

1307	2024	
1310	4022	
1311	0521	
1312	2505	
1313	2324	
1314	0504	
1315	4300	
1316	4311	NOINTD, TEXT /*INTERRUPT REQUEST NOT GENERATED BY "DONE"*/
1317	1624	
1320	0522	
1321	2225	
1322	2024	
1323	4022	
1324	0521	
1325	2505	
1326	2324	
1327	4016	
1330	1724	
1331	4007	
1332	0516	
1333	0522	
1334	0124	
1335	0504	
1336	4002	
1337	3140	
1340	4204	
1341	1716	
1342	0542	
1343	4300	
1344	4311	INTNDN, TEXT /*INTERRUPT STILL REQUESTED AFTER "DONE" CLEARED*/
1345	1624	
1346	0522	
1347	2225	
1350	2024	
1351	4023	
1352	2411	
1353	1414	
1354	4022	
1355	0521	
1356	2505	
1357	2324	
1360	0504	
1361	4001	
1362	0624	
1363	0522	
1364	4042	
1365	0417	
1366	1605	
1367	4240	
1370	0314	
1371	0501	
1372	2205	
1373	0443	
1374	0000	
1375	4311	INTNIE, TEXT /*INTERRUPT REQUESTED WHEN IE NOT SET*/

1376 1624
1377 0522
1400 2225
1401 2024
1402 4022
1403 0521
1404 2505
1405 2324
1406 0504
1407 4027
1410 1005
1411 1640
1412 1105
1413 4016
1414 1724
1415 4023
1416 0524
1417 4300
1420 4323
1421 1114
1422 1740
1423 1617
1424 2440
1425 0314
1426 0501
1427 2205
1430 0440
1431 0231
1432 4022
1433 1404
1434 0343
1435 0000
1436 4304
1437 1716
1440 0540
1441 0411
1442 0440
1443 1617
1444 2440
1445 2305
1446 2443
1447 0000
1450 4304
1451 0122
1452 4024
1453 1740
1454 2311
1455 1417
1456 4024
1457 2201
1460 1623
1461 0605
1462 2240
1463 5015
1464 0111

SICLR, TEXT "#SILO NOT CLEARED BY RLDC!"

DNNSET, TEXT "#DONE DID NOT SET!"

BADDAR, TEXT "#DAR TO SILO TRANSFER (MAINT MODE) FAILED!"

1465	1624	
1466	4015	
1467	1704	
1470	0551	
1471	4006	
1472	0111	
1473	1405	
1474	0443	
1475	0000	
1476	4377	INTOFF, TEXT
1477	4023	"#? SOFTWARE ERROR (JMP TO 0 OR INTERRUPT WHEN INTERRUPT SYSTEM S/B OFF)#"
1500	1706	
1501	2427	
1502	0122	
1503	0540	
1504	0522	
1505	2217	
1506	2240	
1507	5012	
1510	1520	
1511	4024	
1512	1740	
1513	6040	
1514	1722	
1515	4011	
1516	1624	
1517	0522	
1520	2225	
1521	2024	
1522	4027	
1523	1005	
1524	1640	
1525	1116	
1526	2405	
1527	2222	
1530	2520	
1531	2440	
1532	2331	
1533	2324	
1534	0515	
1535	4023	
1536	5702	
1537	4017	
1540	0606	
1541	5143	
1542	0000	
1543	4020	PCINT, TEXT
1544	0340	" PC INTPC#"
1545	4040	
1546	4040	
1547	1116	
1550	2420	
1551	0343	
1552	0000	
1553	4304	DNRLCB, TEXT
		"#DONE NOT CLEARED BY LOADING CB#"

1554	1716				
1555	0540				
1556	1617				
1557	2440				
1560	0314				
1561	0501				
1562	2205				
1563	0440				
1564	0231				
1565	4014				
1566	1701				
1567	0411				
1570	1607				
1571	4003				
1572	0243				
1573	0000				
1574	4020	ERCBCL, TEXT	" PC	ER	CB CALLPC#"
1575	0340				
1576	4040				
1577	4040				
1600	4005				
1601	2240				
1602	4040				
1603	4040				
1604	4003				
1605	0240				
1606	4040				
1607	4003				
1610	0114				
1611	1420				
1612	0343				
1613	0000				
1614	4020	PCERCB, TEXT	" PC	ER	CB#"
1615	0340				
1616	4040				
1617	4040				
1620	4005				
1621	2240				
1622	4040				
1623	4040				
1624	4003				
1625	0243				
1626	0000				
1627	4320	POWER, TEXT	"#POWER FAIL	PC = "	
1630	1727				
1631	0522				
1632	4006				
1633	0111				
1634	1440				
1635	4020				
1636	0340				
1637	7540				
1640	0000				
1641	4305	ERRSET, TEXT	"#ERROR FLAG SET#"		
1642	2222				

1643	1722						
1644	4006						
1645	1401						
1646	0740						
1647	2305						
1650	2443						
1651	0000						
1652	4020	EXACWN, TEXT	" PC	EXPCTD	ACTUAL	WORD NO. #"	
1653	0340						
1654	4040						
1655	4005						
1656	3020						
1657	0324						
1660	0440						
1661	4001						
1662	0324						
1663	2501						
1664	1440						
1665	4027						
1666	1722						
1667	0440						
1670	1617						
1671	5643						
1672	0000						
1673	4020	EXACMA, TEXT	" PC	EXPCTD	ACTUAL	MA(EXPCTD) #"	
1674	0340						
1675	4040						
1676	4005						
1677	3020						
1700	0324						
1701	0440						
1702	4001						
1703	0324						
1704	2501						
1705	1440						
1706	4015						
1707	0150						
1710	0530						
1711	2003						
1712	2404						
1713	5143						
1714	0000						
1715	4020	PCMA, TEXT	" PC	MA(EXPCTD) #"			
1716	0340						
1717	4040						
1720	1501						
1721	5005						
1722	3020						
1723	0324						
1724	0451						
1725	4300						
1726	4020	DALSB, TEXT	" PC	DATA	LSB-EXPCTD-MSB	LSB-ACTUAL-MSB	CALLPC #"
1727	0340						
1730	4040						
1731	4040						

1732 0401
1733 2401
1734 4040
1735 4014
1736 2302
1737 5505
1740 3020
1741 0324
1742 0455
1743 1523
1744 0240
1745 4014
1746 2302
1747 5501
1750 0324
1751 2501
1752 1455
1753 1523
1754 0240
1755 4003
1756 0114
1757 1420
1760 0343
1761 0000

/MORE ERROR TABLE ENTRIES FOLLOWED BY TEXT

1762 1765
1763 0521
1764 7776
1765 4304
1766 2211
1767 2605
1770 4022
1771 0501
1772 0431
1773 4002
1774 1124
1775 4023
1776 0524
1777 4006
2000 1722
2001 4022
2002 1440
2003 0422
2004 1126
2005 0540
2006 6340
2007 4300
2010 2013
2011 0521
2012 7776
2013 4304
2014 2211

DRVRDY, .+3
PCCB
-2
TEXT "#DRIVE READY BIT SET FOR RL DRIVE 3 #"

DRVERR, .+3
PCCB
-2
TEXT "#DRIVE ERROR BIT CLEAR WHEN INTERFACE DISABLED#"

2015 2605
2016 4005
2017 2222
2020 1722
2021 4002
2022 1124
2023 4003
2024 1405
2025 0122
2026 4027
2027 1005
2030 1640
2031 1116
2032 2405
2033 2206
2034 0103
2035 0540
2036 0411
2037 2301
2040 0214
2041 0504
2042 4300
2043 2046
2044 0473
2045 7777
2046 4322
2047 1423
2050 2240
2051 2313
2052 1120
2053 2005
2054 0440
2055 2710
2056 0516
2057 4004
2060 2211
2061 2605
2062 4011
2063 1624
2064 0522
2065 0601
2066 0305
2067 4004
2070 1123
2071 0102
2072 1405
2073 0443
2074 0000
2075 2100
2076 0521
2077 7776
2100 4317
2101 2011
2102 4001
2103 1604

SRERR, .+3
PC
-1
TEXT

"#RLSR SKIPPED WHEN DRIVE INTERFACE DISABLED#"

NDNOPI, .+3
PCCB
-2
TEXT

"#OPI AND DONE DID NOT SET#"

2104 4004
2105 1716
2106 0540
2107 0411
2110 0440
2111 1617
2112 2440
2113 2305
2114 2443
2115 0000
2116 2121
2117 1614
2120 7775
2121 4304
2122 1716
2123 0540
2124 1617
2125 2440
2126 2305
2127 2440
2130 0106
2131 2405
2132 2240
2133 1720
2134 1143
2135 0000
2136 2141
2137 0473
2140 7777
2141 4304
2142 2211
2143 2605
2144 4011
2145 1624
2146 0522
2147 0601
2150 0305
2151 4016
2152 1724
2153 4004
2154 1123
2155 0102
2156 1405
2157 0440
2160 1722
2161 4017
2162 2011
2163 4017
2164 1605
2165 2310
2166 1724
2167 4002
2170 0104
2171 4050
2172 1720

OPINDN, .+3
PCERCB
-3
TEXT "#DONE NOT SET AFTER OPI#"

DRINT, .+3
PC
-1
TEXT "#DRIVE INTERFACE NOT DISABLED OR OPI ONESHOT BAD (OPI NOT SET ON GET STATUS)#"

2173 1140
 2174 1617
 2175 2440
 2176 2305
 2177 2440
 2200 1716
 2201 4007
 2202 0524
 2203 4023
 2204 2401
 2205 2425
 2206 2351
 2207 4300
 2210 2213
 2211 0512
 2212 7776
 2213 4305
 2214 2222
 2215 1722
 2216 4017
 2217 0303
 2220 2522
 2221 2205
 2222 0440
 2223 0106
 2224 2405
 2225 2240
 2226 0611
 2227 1414
 2230 1116
 2231 0740
 2232 2311
 2233 1417
 2234 4300
 2235 2240
 2236 1652
 2237 7774
 2240 4323
 2241 1114
 2242 1740
 2243 0611
 2244 1414
 2245 4024
 2246 0523
 2247 2440
 2250 0401
 2251 2401
 2252 4005
 2253 2222
 2254 1722
 2255 4300
 2256 2261
 2257 0473
 2260 7777
 2261 4305

ERSLOF, .+3
 PCER
 -2
 TEXT "ERROR OCCURRED AFTER FILLING SILO#"

SLODAT, .+3
 EXACWN /PC EXPCTD ACTUAL WORD NO.
 -4
 TEXT "SILO FILL TEST DATA ERROR#"

EFCLR, .+3
 PC
 -1
 TEXT "ERROR FLAG NOT CLEARED BY RLSE#"

2262 2222
2263 1722
2264 4006
2265 1401
2266 0740
2267 1617
2270 2440
2271 0314
2272 0501
2273 2205
2274 0440
2275 0231
2276 4022
2277 1423
2300 0543
2301 0000
2302 2305
2303 0512
2304 7776
2305 4305
2306 2240
2307 1617
2310 2440
2311 0314
2312 0501
2313 2205
2314 0440
2315 0231
2316 4022
2317 1404
2320 0343
2321 0000
2322 2325
2323 0512
2324 7776
2325 4304
2326 0124
2327 0140
2330 1401
2331 2405
2332 5710
2333 1606
2334 4016
2335 1724
2336 4023
2337 0524
2340 4002
2341 3140
2342 1726
2343 0522
2344 0611
2345 1414
2346 1116
2347 0740
2350 2311

ERDC, .+3
PCER
-2
TEXT "ER NOT CLEARED BY RLDC#"

NDL20, .+3
PCER
-2
TEXT "DATA LATE/HNF NOT SET BY OVERFILLING SILO#"

2351	1417		
2352	4300		
2353	2356	EFDC,	.+3
2354	0473		PC
2355	7777		-1
2356	4305	TEXT	"#ERROR FLAG NOT CLEARED BY RLDC#"
2357	2222		
2360	1722		
2361	4006		
2362	1401		
2363	0740		
2364	1617		
2365	2440		
2366	0314		
2367	0501		
2370	2205		
2371	0440		
2372	0231		
2373	4022		
2374	1404		
2375	0343		
2376	0000		
2377	2402	NOREFL,	.+3
2400	0512		PCER
2401	7776		-2
2402	4306	TEXT	"#FULL SILO COULD NOT BE REFILLED AFTER RLDC#"
2403	2514		
2404	1440		
2405	2311		
2406	1417		
2407	4003		
2410	1725		
2411	1404		
2412	4016		
2413	1724		
2414	4002		
2415	0540		
2416	2205		
2417	0611		
2420	1414		
2421	0504		
2422	4001		
2423	0624		
2424	0522		
2425	4022		
2426	1404		
2427	0343		
2430	0000		
2525	*2525		
2525	0000	BUF1B,	ZBLOCK 20
2545	2550	INT21,	.+3
2546	0473		PC

2547	7777		
2550	4311	-1	
2551	1624	TEXT	"#INTERRUPT NOT REQUESTED ON ERROR#"
2552	0522		
2553	2225		
2554	2024		
2555	4016		
2556	1724		
2557	4022		
2560	0521		
2561	2505		
2562	2324		
2563	0504		
2564	4017		
2565	1640		
2566	0522		
2567	2217		
2570	2243		
2571	0000		
2572	2575	SKDO, .+3	
2573	0473	PC	
2574	7777	-1	
2575	4304	TEXT	"#DONE AND OPI SET AFTER ISSUING RLDC#"
2576	1716		
2577	0540		
2600	0116		
2601	0440		
2602	1720		
2603	1140		
2604	2305		
2605	2440		
2606	0106		
2607	2405		
2610	2240		
2611	1123		
2612	2325		
2613	1116		
2614	0740		
2615	2214		
2616	0403		
2617	4300		
2620	2623	CAFOPI, .+3	
2621	0473	PC	
2622	7777	-1	
2623	4303	TEXT	"#CAF DID NOT INHIBIT OPI#"
2624	0106		
2625	4004		
2626	1104		
2627	4016		
2630	1724		
2631	4011		
2632	1610		
2633	1102		
2634	1124		
2635	4017		

2636	2011		
2637	4300		
2640	2643	CAFEDS, .+3	
2641	0473	PC	
2642	7777	-1	
2643	4305	TEXT	"#ERROR AND DONE SET AFTER CAF#"
2644	2222		
2645	1722		
2646	4001		
2647	1604		
2650	4004		
2651	1716		
2652	0540		
2653	2305		
2654	2440		
2655	0106		
2656	2405		
2657	2240		
2660	0301		
2661	0643		
2662	0000		
2663	2666	DNSAC, .+3	
2664	0473	PC	
2665	7777	-1	
2666	4303	TEXT	"#CAF DID NOT INHIBIT DONE#"
2667	0106		
2670	4004		
2671	1104		
2672	4016		
2673	1724		
2674	4011		
2675	1610		
2676	1102		
2677	1124		
2700	4004		
2701	1716		
2702	0543		
2703	0000		
2704	2707	NOFLG7, .+3	
2705	0473	PC	
2706	7777	-1	
2707	4305	TEXT	"#ERROR FLAG NOT SET BY COMMAND 7 MICROCODE#"
2710	2222		
2711	1722		
2712	4006		
2713	1401		
2714	0740		
2715	1617		
2716	2440		
2717	2305		
2720	2440		
2721	0231		
2722	4003		
2723	1715		
2724	1501		

2725 1604
2726 4067
2727 4015
2730 1103
2731 2217
2732 0317
2733 0405
2734 4300
2735 2740
2736 0473
2737 7777
2740 4303
2741 0106
2742 4004
2743 1705
2744 2340
2745 1617
2746 2440
2747 0314
2750 0501
2751 2240
2752 0405
2753 2611
2754 0305
2755 4300
2756 2761
2757 0521
2760 7776
2761 4317
2762 2011
2763 4004
2764 1104
2765 4016
2766 1724
2767 4023
2770 0524
2771 4300
2772 2775
2773 0512
2774 7776
2775 4305
2776 2240
2777 1617
3000 2440
3001 0314
3002 0501
3003 2205
3004 0440
3005 0231
3006 4014
3007 1701
3010 0411
3011 1607
3012 4003
3013 0243

CAFCLR, .+3
PC
-1
TEXT "#CAF DOES NOT CLEAR DEVICE#"

OPINST, .+3
PCCB
-2
TEXT "#OPI DID NOT SET#"

ERNCLD, .+3
PCER
-2
TEXT "#ER NOT CLEARED BY LOADING CB#"

3014	0000				
3015	3020	NWCOVF, .+3			
3016	0473	PC			
3017	7777	-1			
3020	4327	TEXT	"#WORD COUNT OVERFLOW DID NOT STOP MAINT TRANSFER#"		
3021	1722				
3022	0440				
3023	0317				
3024	2516				
3025	2440				
3026	1726				
3027	0522				
3030	0614				
3031	1727				
3032	4004				
3033	1104				
3034	4016				
3035	1724				
3036	4023				
3037	2417				
3040	2040				
3041	1501				
3042	1116				
3043	2440				
3044	2422				
3045	0116				
3046	2306				
3047	0522				
3050	4300				
3051	3054	MWRER, .+3			
3052	1673	EXACMA	/PC EXPCTD ACTUAL MA(EXPCTD)		
3053	7774	-4			
3054	4315	TEXT	"#MAINT WRITE DID NOT WRITE BACK WORD READ OR MA BAD#"		
3055	0111				
3056	1624				
3057	4027				
3060	2211				
3061	2405				
3062	4004				
3063	1104				
3064	4016				
3065	1724				
3066	4027				
3067	2211				
3070	2405				
3071	4002				
3072	0103				
3073	1340				
3074	2717				
3075	2204				
3076	4022				
3077	0501				
3100	0440				
3101	1722				
3102	4015				

3103 0140
 3104 0201
 3105 0443
 3106 0000
 3107 3112
 3110 1673
 3111 7774
 3112 4322
 3113 0501
 3114 0440
 3115 0401
 3116 2401
 3117 4002
 3120 2205
 3121 0113
 3122 4015
 3123 1704
 3124 1106
 3125 1105
 3126 0440
 3127 2717
 3130 2204
 3131 4022
 3132 0501
 3133 0443
 3134 0000
 3135 3140
 3136 0473
 3137 7777
 3140 4315
 3141 0140
 3142 1617
 3143 2440
 3144 0314
 3145 0501
 3146 2205
 3147 0440
 3150 0231
 3151 4022
 3152 1404
 3153 0343
 3154 0000
 3155 3160
 3156 1715
 3157 7776
 3160 4304
 3161 0124
 3162 0140
 3163 0222
 3164 0501
 3165 1340
 3166 0411
 3167 0440
 3170 1617
 3171 2440

REDMOD, .+3
 EXACMA
 -4
 TEXT "#READ DATA BREAK MODIFIED WORD READ#"

MADC, .+3
 PC
 -1
 TEXT "#MA NOT CLEARED BY RLDC#"

NOTFO, .+3
 PCMA /PC MA(EXPCTD)
 -2
 TEXT "#DATA BREAK DID NOT OCCUR TO FIELD 0#"

3172 1703
 3173 0325
 3174 2240
 3175 2417
 3176 4006
 3177 1105
 3200 1404
 3201 4060
 3202 4300
 3203 3206
 3204 1726
 3205 7771
 3206 4303
 3207 2203
 3210 4007
 3211 0516
 3212 0522
 3213 0124
 3214 1117
 3215 1640
 3216 0522
 3217 2217
 3220 2243
 3221 0000
 3222 3225
 3223 0521
 3224 7776
 3225 4304
 3226 0124
 3227 0140
 3230 0222
 3231 0501
 3232 1340
 3233 0411
 3234 0440
 3235 1617
 3236 2440
 3237 1703
 3240 0325
 3241 2240
 3242 2417
 3243 4003
 3244 1722
 3245 2205
 3246 0324
 3247 4006
 3250 1105
 3251 1404
 3252 4300
 3253 3256
 3254 0521
 3255 7776
 3256 4303
 3257 1413
 3260 4017

CRCER, .+3
 DALSB /PC DATA LSB-EXPCTD-MSB LSB-ACTUAL-MSB CALLPC
 -7
 TEXT "#CRC GENERATION ERROR#"

WRNGFL, .+3
 PCCB
 -2
 TEXT "#DATA BREAK DID NOT OCCUR TO CORRECT FIELD#"

CLOPDL, .+3
 PCCB
 -2
 TEXT "#CLK OPI DLY WAS NOT ASSERTED TO SILO (SILO NOT CLEARED)#"

3261 2011
3262 4004
3263 1431
3264 4027
3265 0123
3266 4016
3267 1724
3270 4001
3271 2323
3272 0522
3273 2405
3274 0440
3275 2417
3276 4023
3277 1114
3300 1740
3301 5023
3302 1114
3303 1740
3304 1617
3305 2440
3306 0314
3307 0501
3310 2205
3311 0451
3312 4300
3313 3316
3314 1614
3315 7775
3316 4304
3317 2211
3320 2605
3321 4022
3322 0501
3323 0431
3324 4001
3325 2323
3326 0522
3327 2405
3330 0440
3331 0231
3332 4004
3333 2211
3334 2605
3335 4300
3336 3341
3337 1614
3340 7775
3341 4304
3342 2211
3343 2605
3344 4005
3345 2222
3346 1722
3347 4001

DRDY3, .+3
PCERCB
-3
TEXT "#DRIVE READY ASSERTED BY DRIVE#"

DERR3, .+3
PCERCB
-3
TEXT "#DRIVE ERROR ASSERTED BY DRIVE#"

3350 2323
3351 0522
3352 2405
3353 0440
3354 0231
3355 4004
3356 2211
3357 2605
3360 4300
3361 3364
3362 0512
3363 7776
3364 4305
3365 2222
3366 1722
3367 4006
3370 1401
3371 0740
3372 1617
3373 2440
3374 2305
3375 2440
3376 0231
3377 4022
3400 0501
3401 0440
3402 0401
3403 2401
3404 4014
3405 0124
3406 0540
3407 0606
3410 4300
3411 3414
3412 0512
3413 7776
3414 4305
3415 2222
3416 1722
3417 4006
3420 1401
3421 0740
3422 1617
3423 2440
3424 2305
3425 2440
3426 0231
3427 4017
3430 2011
3431 4300
3432 3435
3433 1574
3434 7774
3435 4304
3436 0124

EFNRDL, .+3
PCER
-2
TEXT "#ERROR FLAG NOT SET BY READ DATA LATE FF#"

NEFOPI, .+3
PCER
-2
TEXT "#ERROR FLAG NOT SET BY OPI#"

NDL28, .+3
ERCBCL
-4
TEXT "#DATA LATE DID NOT SET ON MAINT COMMAND TRANSFER#"

3437 0140
3440 1401
3441 2405
3442 4004
3443 1104
3444 4016
3445 1724
3446 4023
3447 0524
3450 4017
3451 1640
3452 1501
3453 1116
3454 2440
3455 0317
3456 1515
3457 0116
3460 0440
3461 2422
3462 0116
3463 2306
3464 0522
3465 4300
3466 3471
3467 0476
3470 7775
3471 4305
3472 2222
3473 1722
3474 4006
3475 1401
3476 0740
3477 1617
3500 2440
3501 2305
3502 2440
3503 0231
3504 4027
3505 2211
3506 2405
3507 4004
3510 0124
3511 0140
3512 1401
3513 2405
3514 4006
3515 0643
3516 0000
3517 3522
3520 0512
3521 7776
3522 4305
3523 2222
3524 1722
3525 4006

EFNSDL, .+3
PCERCL /PC ER CALLPC
-3
TEXT "ERROR FLAG NOT SET BY WRITE DATA LATE FF!"

EFNOIN, .+3
PCER
-2
TEXT "ERROR FLAG SET WHEN INTERFACE DISABLED (DRV ERR NOT ASSERTED TO FF CLK INPUT)!"

3526 1401
3527 0740
3530 2305
3531 2440
3532 2710
3533 0516
3534 4011
3535 1624
3536 0522
3537 0601
3540 0305
3541 4004
3542 1123
3543 0102
3544 1405
3545 0440
3546 5004
3547 2226
3550 4005
3551 2222
3552 4016
3553 1724
3554 4001
3555 2323
3556 0522
3557 2405
3560 0440
3561 2417
3562 4006
3563 0640
3564 0314
3565 1340
3566 1116
3567 2025
3570 2451
3571 4300
3572 3575
3573 0473
3574 7777
3575 4322
3576 1404
3577 0340
3600 0301
3601 2523
3602 0504
3603 4023
3604 1311
3605 2043
3606 0000
3607 3612
3610 0473
3611 7777
3612 4322
3613 1415
3614 0140

RLDCSK, .+3
PC
-1
TEXT "RLDC CAUSED SKIP#"

RLMASK, .+3
PC
-1
TEXT "RLMA CAUSED SKIP#"

3615	0301		
3616	2523		
3617	0504		
3620	4023		
3621	1311		
3622	2043		
3623	0000		
3624	3627	RLCASK, .+3	
3625	0473	PC	
3626	7777	-1	
3627	4322	TEXT	"#RLCA CAUSED SKIP#"
3630	1403		
3631	0140		
3632	0301		
3633	2523		
3634	0504		
3635	4023		
3636	1311		
3637	2043		
3640	0000		
3641	3644	RLCBSK, .+3	
3642	0473	PC	
3643	7777	-1	
3644	4322	TEXT	"#RLCB CAUSED SKIP#"
3645	1403		
3646	0240		
3647	0301		
3650	2523		
3651	0504		
3652	4023		
3653	1311		
3654	2043		
3655	0000		
3656	3661	RLSASK, .+3	
3657	0473	PC	
3660	7777	-1	
3661	4322	TEXT	"#RLSA CAUSED SKIP#"
3662	1423		
3663	0140		
3664	0301		
3665	2523		
3666	0504		
3667	4023		
3670	1311		
3671	2043		
3672	0000		
3673	3676	RLWCSK, .+3	
3674	0473	PC	
3675	7777	-1	
3676	4322	TEXT	"#RLWC CAUSED SKIP#"
3677	1427		
3700	0340		
3701	0301		
3702	2523		
3703	0504		

3704	4023		
3705	1311		
3706	2043		
3707	0000		
3710	3713	SPARSK, .+3	
3711	0473	PC	
3712	7777	-1	
3713	4323	TEXT	"#SPARE IOT CAUSED SKIP#"
3714	2001		
3715	2205		
3716	4011		
3717	1724		
3720	4003		
3721	0125		
3722	2305		
3723	0440		
3724	2313		
3725	1120		
3726	4300		
3727	3732	RRERSK, .+3	
3730	0473	PC	
3731	7777	-1	
3732	4322	TEXT	"#RRER CAUSED SKIP#"
3733	2205		
3734	2240		
3735	0301		
3736	2523		
3737	0504		
3740	4023		
3741	1311		
3742	2043		
3743	0000		
3744	3747	RRWCSK, .+3	
3745	0473	PC	
3746	7777	-1	
3747	4322	TEXT	"#RRWC CAUSED SKIP#"
3750	2227		
3751	0340		
3752	0301		
3753	2523		
3754	0504		
3755	4023		
3756	1311		
3757	2043		
3760	0000		
3761	3764	RRCASK, .+3	
3762	0473	PC	
3763	7777	-1	
3764	4322	TEXT	"#RRCA CAUSED SKIP#"
3765	2203		
3766	0140		
3767	0301		
3770	2523		
3771	0504		
3772	4023		

3773	1311		
3774	2043		
3775	0000		
3776	4001	RRCBSK, .+3	
3777	0473	PC	
4000	7777	-1	
4001	4322	TEXT	"#RRCB CAUSED SKIP#"
4002	2203		
4003	0240		
4004	0301		
4005	2523		
4006	0504		
4007	4023		
4010	1311		
4011	2043		
4012	0000		
4013	4016	RRSASK, .+3	
4014	0473	PC	
4015	7777	-1	
4016	4322	TEXT	"#RRSA CAUSED SKIP#"
4017	2223		
4020	0140		
4021	0301		
4022	2523		
4023	0504		
4024	4023		
4025	1311		
4026	2043		
4027	0000		
4030	4033	RRSISK, .+3	
4031	0473	PC	
4032	7777	-1	
4033	4322	TEXT	"#RRSI CAUSED SKIP#"
4034	2223		
4035	1140		
4036	0301		
4037	2523		
4040	0504		
4041	4023		
4042	1311		
4043	2043		
4044	0000		
4045	4050	HNFS, .+3	
4046	1614	PCERCB	
4047	7775	-3	
4050	4310	TEXT	"#HEADER NOT FOUND FF NOT SET#"
4051	0501		
4052	0405		
4053	2240		
4054	1617		
4055	2440		
4056	0617		
4057	2516		
4060	0440		
4061	0606		

4062 4016
4063 1724
4064 4023
4065 0524
4066 4300
4067 4072
4070 0540
4071 7775
4072 4304
4073 0124
4074 0140
4075 0222
4076 0501
4077 1340
4100 2205
4101 0104
4102 4002
4103 0104
4104 4027
4105 1722
4106 0440
4107 1716
4110 4027
4111 2211
4112 2405
4113 4003
4114 1715
4115 1501
4116 1604
4117 4300
4120 4123
4121 0473
4122 7777
4123 4304
4124 0124
4125 0140
4126 0222
4127 0501
4130 1323
4131 4004
4132 1104
4133 4016
4134 1724
4135 4017
4136 0303
4137 2522
4140 4017
4141 1640
4142 2722
4143 1124
4144 0540
4145 0317
4146 1515
4147 0116
4150 0443

DBRBWW, .+3
EXPACT
-3
TEXT "#DATA BREAK READ BAD WORD ON WRITE COMMAND#"

DBNPBW, .+3
PC
-1
TEXT "#DATA BREAKS DID NOT OCCUR ON WRITE COMMAND#"

4151	0000				
4152	4155	WBWSE,	+.3		
4153	0521		PCCB		
4154	7776		-2		
4155	4327	TEXT		"#WRITE BREAK OCCURRED WHEN SILO WAS EMPTY#"	
4156	2211				
4157	2405				
4160	4002				
4161	2205				
4162	0113				
4163	4017				
4164	0303				
4165	2522				
4166	2205				
4167	0440				
4170	2710				
4171	0516				
4172	4023				
4173	1114				
4174	1740				
4175	2701				
4176	2340				
4177	0515				
4200	2024				
4201	3143				
4202	0000				
4203	4206	WBBDNO,	+.3		
4204	4232		WD1WD2		
4205	7772		-6		
4206	4327	TEXT		"#WRITE BREAKS FAILED OR DID NOT OCCUR#"	
4207	2211				
4210	2405				
4211	4002				
4212	2205				
4213	0113				
4214	2340				
4215	0601				
4216	1114				
4217	0504				
4220	4017				
4221	2240				
4222	0411				
4223	0440				
4224	1617				
4225	2440				
4226	1703				
4227	0325				
4230	2243				
4231	0000				
4232	4020	WD1WD2, TEXT	" PC	WD2-EXPCD-WD1	WD2-ACTUAL-WD1 CB#"
4233	0340				
4234	4040				
4235	4027				
4236	0462				
4237	5505				

4240 3020
4241 0324
4242 0455
4243 2704
4244 6140
4245 4027
4246 0462
4247 5501
4250 0324
4251 2501
4252 1455
4253 2704
4254 6140
4255 4040
4256 4003
4257 0243
4260 0000
4261 4264
4262 1673
4263 7774
4264 4304
4265 0124
4266 0140
4267 0522
4270 2217
4271 2240
4272 1116
4273 4064
4274 1340
4275 2722
4276 1124
4277 0540
4300 0622
4301 1715
4302 4006
4303 1105
4304 1404
4305 4062
4306 4300
4307 4312
4310 1673
4311 7774
4312 4304
4313 0124
4314 0140
4315 0522
4316 2217
4317 2240
4320 1116
4321 4064
4322 1340
4323 2722
4324 1124
4325 0540
4326 0622

DER4KW, .+3
EXACMA
-4
TEXT "#DATA ERROR IN 4K WRITE FROM FIELD 2#"

DER4K1, .+3
EXACMA
-4
TEXT "#DATA ERROR IN 4K WRITE FROM FIELD 1#"

4327 1715
4330 4006
4331 1105
4332 1404
4333 4061
4334 4300
4335 0530
4336 1124
4337 4300
4340 4320
4341 1001
4342 2305
4343 4014
4344 1703
4345 1305
4346 0440
4347 1417
4350 1720
4351 4001
4352 0412
4353 2523
4354 2415
4355 0516
4356 2440
4357 2217
4360 2524
4361 1116
4362 0543
4363 2327
4364 1124
4365 0310
4366 4022
4367 0507
4370 1123
4371 2405
4372 2240
4373 0317
4374 1624
4375 0516
4376 2423
4377 4001
4400 2205
4401 4025
4402 2305
4403 0440
4404 0123
4405 4004
4406 0124
4407 0140
4410 2001
4411 2424
4412 0522
4413 1640
4414 1116
4415 4015

EXIT, TEXT "EXIT!"

PLLRO1, TEXT "PHASE LOCKED LOOP ADJUSTMENT ROUTINE; SWITCH REGISTER CONTENTS ARE USED AS DATA PATTERN IN

4416 0111
4417 1624
4420 4003
4421 1715
4422 1501
4423 1604
4424 0000
4425 4320
4426 0124
4427 2405
4430 2216
4431 4006
4432 2217
4433 1540
4434 2722
4435 1124
4436 0540
4437 0222
4440 0501
4441 1340
4442 0411
4443 2320
4444 1401
4445 3105
4446 0440
4447 1116
4450 4015
4451 2143
4452 2431
4453 2005
4454 4074
4455 0322
4456 7640
4457 2417
4460 4005
4461 3011
4462 2440
4463 0000
4464 4305
4465 3005
4466 0325
4467 2405
4470 4020
4471 1001
4472 2305
4473 4014
4474 1703
4475 1305
4476 0440
4477 1417
4500 1720
4501 4001
4502 0412
4503 2523
4504 2415

PLLR02, TEXT "#PATTERN FROM WRITE BREAK DISPLAYED IN MQTYPE <CR> TO EXIT "

EXPLL, TEXT "#EXECUTE PHASE LOCKED LOOP ADJUSTMENT ROUTINE? "

4505 0516
 4506 2440
 4507 2217
 4510 2524
 4511 1116
 4512 0577
 4513 4000
 4514 0422
 4515 1720
 4516 2011
 4517 1607
 4520 4025
 4521 1611
 4522 2443
 4523 0000

DRPUNI, TEXT "DROPRING UNIT#"

4524 5504 IOTTAB, IOT0
 4525 6562 IOT0A
 4526 6563 IOT0B
 4527 6564 IOT0C
 4530 6565 IOT0D
 4531 6566 IOT0E
 4532 6567 IOT0F
 4533 6570 IOT0G
 4534 5512 IOT1
 4535 5517 IOT2
 4536 5525 IOT3
 4537 5533 IOT4
 4540 5541 IOT5
 4541 5547 IOT6
 4542 5555 IOT7
 4543 5563 IOT10
 4544 5601 IOT11
 4545 5607 IOT12
 4546 5615 IOT13
 4547 5623 IOT14
 4550 5631 IOT15
 4551 4043 IOT15A
 4552 3713 IOT15B
 4553 5637 IOT16
 4554 5644 IOT17
 4555 0000 0

/TABLE TERMINATOR

4556 0000 DATA4K, 0000
 4557 7777 7777
 4560 0000 0000
 4561 7777 7777
 4562 0707 0707
 4563 7070 7070
 4564 2525 2525
 4565 5252 5252
 4566 2525 2525
 4567 5252 5252
 4570 0000 0000
 4571 0001 0001

/64 WORDS OF DATA PATTERNS USED TO FILL FIELD 2
 /WHEN AVAILABLE FOR DATA BREAK TEST

4572	0002	0002
4573	0004	0004
4574	0010	0010
4575	0020	0020
4576	0040	0040
4577	0100	0100
4600	0200	0200
4601	0400	0400
4602	1000	1000
4603	2000	2000
4604	4000	4000
4605	0000	0000
4606	4000	4000
4607	2000	2000
4610	1000	1000
4611	0400	0400
4612	0200	0200
4613	0100	0100
4614	0040	0040
4615	0020	0020
4616	0010	0010
4617	0004	0004
4620	0002	0002
4621	0001	0001
4622	7777	7777
4623	7776	7776
4624	7775	7775
4625	7773	7773
4626	7767	7767
4627	7757	7757
4630	7737	7737
4631	7677	7677
4632	7577	7577
4633	7377	7377
4634	6777	6777
4635	5777	5777
4636	3777	3777
4637	7777	7777
4640	3777	3777
4641	5777	5777
4642	6777	6777
4643	7377	7377
4644	7577	7577
4645	7677	7677
4646	7737	7737
4647	7757	7757
4650	7767	7767
4651	7773	7773
4652	7775	7775
4653	7776	7776
4654	3775	3775
4655	5552	5552

5252 *5252

5252 0000 BUF1A, ZBLOCK 10

0000 FIELD 0

SEQ 129

[illegible]

5000
5100

[illegible]

5400
5500

5600
5700

6000
6100

6200
6300

6400
6500

6600
6700

7000
7100

7200
7300

7400
7500

7600.
7700

0200 *200

8

ACRLCA	0621	CONSOL	4452	EFNRDL	3361	ERSLOF	2210
ACRLCB	0636	CRC0	6325	EFNSDL	3466	ESIA	6055
ACRLDC	1136	CRC1	6326	END10	1255	ETABPT	4547
ACRLMA	0705	CRC14	6327	END11	1303	EXACMA	1673
ACRLSA	0653	CRC15	6330	END12	1343	EXACWN	1652
ACRLWC	0670	CRC213	6331	END13	1432	EXIT	4335
ACSKIP	0603	CRCCHK	4470	END14	1475	EXPACT	0540
ACSPAR	1116	CRCCMP	6267	END15	1532	EXPLL	4464
APTCHK	4446	CRCCNT	6332	END16	1553	FALCHK	4214
APTERR	4537	CR CER	3203	END17A	1620	FATAL	0252
APTMEM	0312	CRCLUP	6213	END17B	1641	FILLEQ	0457
AUTO10	0010	CRLF	4506	END19	1765	FILLER	0023
AUTO11	0011	CRLFSV	5074	END20	2037	FILLIN	4656
BADDAR	1450	CTLF	4633	END21	2103	FILOCT	4672
BRKFLD	6450	CTLG	4635	END22	2310	FIXFIL	4675
BRKRET	6447	DALSB	1726	END23	2343	FLEMA	6451
BRKVFY	4475	DAR11	0327	END24	2502	GDCRRT	6323
BSW	7002	DAREXP	4465	END25	2560	GET8	0252
BUFO	7400	DATA1	0113	END26	2667	GET9	0251
BUF1A	5252	DATA2	0114	END27	2712	GETSR	4445
BUF1B	2525	DATA3	0115	END29	3107	GOTNUM	0253
BUFSET	4476	DATA4	0116	END29A	3161	HCM1	0021
CA2A	0203	DATA4K	4556	END30	3230	HCW2	0022
CA2B	0206	DATA5	0117	END32	3455	HLTCHK	4515
CA2C	0211	DATA6	0120	END33	3550	HNFNS	4045
CABAD	0722	DATCNT	4543	END34	3650	ILOOP	4023
CACHK	0463	DATLUP	4507	END35	3704	IMORET	4307
CACLR	1032	DATTAB	4544	END36	3745	IMPINT	5714
CAF	6007	DBNPBW	4120	END37	4134	INCPAS	4227
CAFCLR	2735	DBRBW	4067	END4KT	4066	INITMO	4271
CAFEDS	2640	DC3	0260	END6	1016	INPUT	4714
CAFOPI	2620	DER4K1	4307	END8	1215	INT21	2545
CARET	0510	DER4KW	4261	EOP	4200	INT8A	0313
CB2A	0214	DERR3	3336	EOPMES	0445	INT8B	0316
CB2B	0217	DIGCNT	4764	ERCBCL	1574	INT9	0321
CB2C	0222	DIGITS	4763	ERDC	2302	INTNDN	1344
CB4	0263	DNCLR	1243	ERFLGS	0335	INTNIE	1375
CBAFF	1153	DNNSET	1436	ERFLSS	0332	INTOFF	1476
CBBAD	0744	DNOUT	6054	ERFLST	2247	IOT0	5504
CBCHK	0511	DNRLCB	1553	ERNCLD	2772	IOT0A	6562
CBCLR	1047	DNRLDC	1261	ERR32A	3446	IOT0B	6563
CBRET	0540	DNSAC	2663	ERR32B	3453	IOT0C	6564
CDFINS	6421	DONE23	2334	ERR33	3544	IOT0D	6565
CHARSV	5055	DRDY3	3313	ERR6	1011	IOT0E	6566
CHKOK	5014	DRINT	2136	ERRCHA	4467	IOT0F	6567
CLOPDL	3253	DRPUNI	4514	ERRCHK	4471	IOT0G	6570
CNTRLC	5171	DRVERR	2010	ERRFLG	0123	IOT1	5512
CNTRLF	4645	DRVRDY	1762	ERROR	4451	IOT10	5563
CNTRLG	4453	E1	0200	ERRPC	0112	IOT11	5601
CNTRLQ	4643	EFCLR	2256	ERRSET	1641	IOT12	5607
CNTRLS	4641	EFDC	2353	ERRTAB	6572	IOT13	5615
CONRET	4644	EFNOIN	3517	ERRTYP	4551	IOT14	5623

IOT15 5631
 IOT15A 4043
 IOT15B 3713
 IOT16 5637
 IOT17 5644
 IOT2 5517
 IOT3 5525
 IOT4 5533
 IOT5 5541
 IOT6 5547
 IOT7 5555
 IOTCNG 0316
 IOTLUP 0325
 IOTTAB 4524
 JMPPM1 4473
 JMPUP 5717
 K0 3560
 K1 0167
 K10 0145
 K100 0140
 K13 0160
 K177 0176
 K20 0151
 K200 0147
 K212 5073
 K215 5072
 K240 5463
 K2525 6471
 K260 5420
 K3 0153
 K300 0144
 K301 0142
 K304 2565
 K310 0143
 K3301 5703
 K364 0136
 K37 0175
 K377 0146
 K3770 0137
 K4 0157
 K400 0152
 K4301 0161
 K4303 2350
 K4303A 2567
 K4315 3554
 K4315A 3753
 K4316 2566
 K4325 4137
 K5 4746
 K5252 6501
 K5302 0134
 K5307 0173

K5707 0174
 K6301 0162
 K6777 3557
 K6777A 2727
 K7 5417
 K7301 0141
 K7302 0133
 K7316 3555
 K7400 6030
 K7600 0154
 K77 5461
 K7700 0135
 K7701 0177
 K7757 0350
 K7764 1772
 K777 5765
 K7773 5764
 K7773A 1771
 K7774 0164
 K77A 6333
 KCC 6032
 KCDF 6334
 KC DFA 6452
 KCF 6030
 KIE 6035
 KINP 0272
 KSF 6031
 LASTIN 5170
 LISN 4477
 LISN1 5134
 LISN2 5145
 LISN3 5160
 LISNUM 5150
 LNFEED 4735
 LOOPPT 0122
 LUPCNT 0121
 LUPSET 4425
 M10 0170
 M12 0150
 M13 3556
 M14 0155
 M15 4141
 M20 0156
 M203 5167
 M21 4142
 M212 5165
 M223 0163
 M25 4143
 M270 5156
 M3 5166
 M31 4144
 M35 4145

M4 0164
 M40 0347
 M43 5462
 M5 0172
 MA3 0247
 MADC 3135
 MANDEC 0340
 MANRET 6560
 MANTEC 4472
 MEMBIG 0420
 MEMLIS 0231
 MEMMES 0226
 MEMPRM 0362
 MEMSIZ 0131
 MESSAGE 4500
 MESAGX 5421
 MESBSW 5442
 MESCNT 5464
 MESLUP 5427
 MESSAG 4500
 MESSAV 5465
 MFRET 4267
 MOTINI 0352
 MQL 7421
 MRSET3 4462
 MULFAL 4255
 MULTER 4463
 MWRRER 3051
 NDL20 2322
 NDL28 3432
 NDN22 2262
 NDN24 2450
 NDN25 2533
 NDNOP1 2075
 NEFOPI 3411
 NERRET 6352
 NET24 2437
 NET25 2525
 NETH22 2254
 NEWOCT 4747
 NEWPAS 4253
 NEWRET 4762
 NO22 2304
 NO24 2476
 NO25 2554
 NODALA 2033
 NUDN14 1470
 NODN15 1513
 NODN26 2657
 NODONE 0277
 NUER26 2665
 NOFL20 2026

NOFL24 2471
 NOFLG7 2704
 NOINTD 1316
 NOP24 2460
 NOP25 2543
 NOPI15 1523
 NOPI22 2272
 NOPMES 5732
 NOPRNT 0127
 NOREFL 2377
 NOTFO 3155
 NOTMOT 4535
 NWCOVF 3015
 NXTOPT 4224
 OK2TYP 5042
 OLOOP 4017
 ONLISN 4615
 OPI23 2331
 OPICNT 6071
 OPILUP 6034
 OPINDN 2116
 OPINST 2756
 OPIOUT 6063
 OPIRET 6070
 OPIWAT 4463
 OPWA22 2230
 OS8 5176
 OUT29 3052
 P2SAVE 5011
 P4KDAT 4140
 P4SAVE 5411
 PAPTER 4545
 PAPTIN 5502
 PASCHK 4420
 PASCNT 0130
 PAT25 4454
 PAT52 4455
 PATCMP 4456
 PATH1 4466
 PATTRN 0124
 PBUF0 0166
 PBUF1A 0171
 PBUF1B 0165
 PC 0473
 PRACT 0530
 PCCB 0521
 PCDAR 0554
 PCER 0512
 PCERCB 1614
 PCERCL 0476
 PCINT 1543
 PCMA 1715

PEOP	4546	RRSISK	4030	TEST14	1433	WC2B	0241
PERTAB	4550	RRWC	4434	TEST15	1476	WC2C	0244
PERTB2	4270	RRWCSK	3744	TEST16	1535	WC4	0266
PHLKLP	0132	SA2A	0225	TEST17	1600	WCAFF	1176
PIOTS	0346	SA2B	0230	TEST18	1642	WCBAD	1010
PLLOOP	3122	SA2C	0233	TEST19	1710	WCCHK	0600
PLLR01	4340	SA4	0271	TEST2	0425	WCCLR	1101
PLLR02	4425	SAAFF	1217	TEST20	2000	WCRET	0625
POWER	1627	SABAD	0766	TEST21	2040	WD1WD2	4232
PPEUP	4136	SACHK	0542	TEST22	2200	WRNGFL	3222
PRNT1	4501	SACLR	1064	TEST23	2313	WTLU30	3207
PRNT2	4502	SAFECB	0570	TEST24	2400	XAPTCH	4564
PRNT4	4503	SARET	0567	TEST25	2505	XBRKVF	6400
PSAFEC	0732	SCNINT	4510	TEST26	2600	XBUFSE	6453
PSCOPI	4254	SCOPE	4450	TEST27	2670	XCNTL	4704
PSWR	0020	SCOPIN	0340	TEST28	2713	XCONSO	4600
PTESTO	0351	SD5B	0274	TEST29	3000	XCRCCH	6200
PTST30	3166	SD6	0302	TEST3	0634	XCRLF	5056
PWRFAL	5704	SD7	0305	TEST30	3200	XDAREX	5736
PWRUP	5720	SETINT	4511	TEST31	3231	XEFCH	6544
Q4PLL	0273	SICLR	1420	TEST32	3400	XERRCH	6335
QESMRK	0416	SIERR	0324	TEST33	3457	XERROR	4436
REDMOD	3107	SKDO	2572	TEST34	3600	XESET3	5665
RESET	4457	SLOCNT	6364	TEST35	3651	XESETW	5650
RESET3	4461	SLOCTA	1770	TEST36	3705	XFRCHK	4467
RESETW	4460	SLODAT	2235	TEST37	4000	XGETSR	4765
RESTSR	4737	SLOFIL	4474	TEST4	0664	XJMPPM	6072
RETSET	4432	SLORDL	1736	TEST6	1000	XLISN	5101
RLCA	4427	SPACE2	4504	TEST7	1017	XMANTE	6523
RLCASK	3624	SPACK2	5075	TEST8	1200	XMRSET	5674
RLCB	4430	SPARE	4443	TEST9	1223	XOPIWA	6031
RLCBSK	3641	SPARE3	0255	TICK	4447	XPAT25	6462
RLDC	4424	SPARSK	3710	TIE	6045	XPAT52	6472
RLDCSK	3572	SPL	6102	TIKRET	5476	XPATCM	6506
RLMA	4426	SRERR	2043	TIMER	3167	XPATM1	6502
RLMASK	3607	SROCT	4732	TOCK	5501	XPRNT1	5412
RLSA	4431	SRQ	6003	TST29A	3110	XPRNT2	5000
RLSASK	3656	STOPF	4444	TSTWC	3737	XPRNT4	5400
RLSD	4425	STOPWR	3045	TWOSPA	0471	XRESET	5656
RLSE	4442	STPWRI	3225	TYPE	4505	XRLCA	5524
RLSR	4441	SWRMMSG	0465	UN8	0310	XRLCB	5532
RLWC	4432	T2WC	0626	UNI27	2672	XRLDC	5503
RLWCSK	3673	T9E	1225	UNKINT	1277	XRLMA	5516
RRCA	4435	TABENT	4542	UPARRC	0453	XRLSA	5540
RRCASK	3761	TEMP1	0125	UPARRG	0455	XRLSD	5511
RRCB	4436	TEMP2	0126	VT278	4507	XRLSE	5643
RRCBSK	3776	TEST0	0353	VTCHK	2106	XRLSR	5636
RRER	4433	TEST1	0407	WAIT	4464	XRLWC	5554
RRERSK	3727	TEST10	1250	WALP29	3027	XRRCA	5606
RRSA	4437	TEST11	1256	WBBNO	4203	XRRCB	5614
RRSASK	4013	TEST12	1304	WBWSE	4152	XRRER	5562
RRSI	4440	TEST13	1400	WC2A	0236	XRRSA	5622

XRRSI	5630
XRRWC	5600
XSCNIN	2114
XSCOPE	4400
XSETIN	2121
XSLOFI	6355
XSPARE	5546
XSTOPF	6561
XTICK	5466
XTYPE	5012
XVT278	2135
XWAIT	5766
XXFRCH	6000
ZEND8	1217

ERRORS DETECTED: 0

LINKS GENERATED: 3

RUN-TIME: 8 SECONDS

3K CORE USED



