

IDENTIFICATION

SEQ 0001

PRODUCT CODE: MAINDEC-11-DZRX8-E-D
PRODUCT NAME: RX11 INTERFACE DIAGNOSTIC
DATE: APRIL 1976
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: DAVID L. ADAMS

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

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

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

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

TABLE OF CONTENTS -----

1.0	GENERAL PROGRAM INFORMATION	
1.1	ABSTRACT	
1.2	SYSTEM REQUIREMENTS	
1.2.1	HARDWARE	
1.2.2	SOFTWARE	
2.0	OPERATING INSTRUCTIONS	
2.0.1	OUTLINE OF OPERATING PROCEDURE	
2.1	LOADING PROCEDURE	
2.2	STARTING ADDRESSES	
2.3	OPERATOR ACTION BEFORE STARTING PROGRAM	
2.3.1	DEVICE ADDRESS SELECTION	
2.3.2	NON-STANDARD DISKETTE ADDRESS SELECTION	
2.3.3	SOFTWARE SWITCH REGISTER (LOC. 176)	
2.3.4	TEST PARAMETER SELECTION ("DTESTP" LOC. 1212)	
2.3.4.1	PREREQUISITES OF TESTS	
2.4	OPERATOR ACTION TO RUN THE PROGRAM	
2.4.1	STARTING THE PROGRAM	
2.4.2	OPERATING CONDITIONS	
2.5	TEST DEFINITIONS	
2.5.1	PRETEST	
2.5.2	TEST 1 - RXCS TEST PART I / INTERRUPT TEST PART I	
2.5.3	TEST 2 - INTERRUPT TEST PART II / VECTOR ADDRESS VERIFICATION	
2.5.4	TEST 3 - INTERRUPT TEST PART III / PRIORITY LEVEL VERIFICATION PART I	
2.5.5	TEST 4 - INTERRUPT TEST PART IV / PRIORITY VERIFICATION PART II	
2.5.6	TEST 5 - INIT (PROGRAMED) / RST	
2.5.7	TEST 6 - FILL BUFFER TRANSFER LENGTH VERIFICATION	
2.5.8	TEST 7 - EMPTY BUFFER TRANSFER LENGTH AND CONTENT VERIFICATION PART I	
2.5.9	TEST 10 - EMPTY BUFFER TRANSFER LENGTH AND CONTENT VERIFICATION PART II	
2.5.10	TEST 11 - FILL / EMPTY BUFFER ALL 0'S	
2.5.11	TEST 12 - FILL / EMPTY BUFFER ALL 1'S	
2.5.12	TEST 13 - DRIVE READY VERIFICATION	
2.5.13	TEST 14 - ERROR FLAG AND B-CODE VERIFICATION PART I	
2.5.14	TEST 15 - ERROR FLAG AND B-CODE VERIFICATION PART II /DELETED DATA BIT SETS	
2.5.15	TEST 16 - ERROR FLAG AND B-CODE VERIFICATION PART III /DELETED DATA BIT CLEARS	

2.5.16 TEST 17 - ILLEGAL TRACK ERROR AND B-CODE VERIFICATION
 2.5.17 TEST 20 - SEEK VERIFICATION VIA READ FUNCTION
 2.5.18 TEST 21 - WRITE TEST
 2.5.19 TEST 22 - INITIALIZE IMPLIED READ
 2.5.20 TEST 23 - READ TEST
 2.5.21 TEST 24 - DATA TRANSFER AND VERIFICATION
 2.5.22 TEST 25 - DATA TRANSFER AND VERIFICATION
 2.5.23 TEST 26 - HEAD "HOME" TEST

3.0 ERRORS

3.1 ERROR HEADING FOR TESTS 1 - 17, 21 - 23
 3.2 ERROR OUTPUT PER TEST
 3.3 ERROR HEADING FOR TEST 20, 24 - 26
 3.3.1 NO ERROR FLAG ERRORS
 3.3.2 ERROR FLAG ERRORS
 3.3.3 ERRORS RESULTING FROM PREVIOUS ERRORS
 3.3.4 DEFINITIVE ERROR CODES

3.4 PROGRAM HUNG

4.0 HALTS

5.0 FLOW CHARTS

1.0 GENERAL PROGRAM INFORMATION

1.1 ABSTRACT

THE RX11 INTERFACE DIAGNOSTIC CONSISTS OF A SERIES OF SELECTABLE TESTS THAT MAY BE RUN INDIVIDUALLY, SEQUENCE THROUGH ALL TESTS, OR START AT A SELECTED TEST AND RUN THROUGH REMAINING TESTS, IN ORDER, THEN GO BACK TO THE SELECTED TEST.

THESE TESTS CHECK OUT THE BASIC FUNCTIONS OF THE RX11 INTERFACE SUCH AS:

- A. DONE FLAG
- B. INTERRUPT LEVEL / ADDRESS
- C. PROGRAM INITIALIZE
- D. READ STATUS REGISTERS
- E. FILL / EMPTY BUFFER TRANSFER VERIFICATION
- F. FILL / EMPTY BUFFER WITH DATA PATTERNS

IT IS NECESSARY TO INSURE THAT THESE FUNCTIONS WORK BEFORE A DATA RELIABILITY TEST IS RUN.

ANY ERRORS ARE REPORTED BY THE PROGRAM, AND IT IS POSSIBLE TO LOOP ON THE ERROR OR A PARTICULAR TEST FOR SCOPE TESTING.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

THE FOLLOWING EQUIPMENT IS REQUIRED:

- A. PDP-11 SERIES COMPUTER WITH MINIMUM OF 8K MEMORY
- B. RX11 FLOPPY DISK SYSTEM, INCLUDING A SINGLE OR DUAL DRIVE RX01 AND A PDP-11 INTERFACE CARD [M7846].
- C. NOTE: A DISKETTE MUST BE INCLUDED WITH EACH DRIVE TESTED.
CONSOLE TELEPRINTER

1.2.2 SOFTWARE REQUIREMENTS

NO PREREQUISITE SOFTWARE

2.0 OPERATING INSTRUCTIONS

2.0.1 OUTLINE OF OPERATING PROCEDURE

THE STANDARD RUNNING PROCEDURE FOR THE DIAGNOSTIC (TO RUN ALL TESTS ON BOTH DRIVES WITH NO OPERATOR INTERVENTION VIA THE SWITCH REGISTER) IS AS FOLLOWS:

- A. LOAD THE PROGRAM INTO MEMORY
1. IF IT IS BEING LOADED FROM A DISKETTE
REPLACE THE "LIBRARY" DISKETTE WITH
A "SCRATCH" DISKETTE.
- NOTE: IF THIS STEP IS FORGOTTEN AND THE PROGRAM
WAS LOADED VIA PXDP (FLOPPY MONITOR) ON
UNIT 0 WITH UNIT 0 SELECTED BY USER TO
UNDERGO TESTING THE PROGRAM WILL FAILSAFE
THE OPERATION AND PROMPT THE USER AS FOLLOWS:
"CAUTION - IF YOU DESIRE TO TEST UNIT 0
REPLACE LOAD MEDIUM WITH A SCRATCH DISKETTE
THEN PRESS CONTINUE"

CAUTION AGAIN, HOWEVER *****
NOTE 1) WHEN RUNNING THIS PROGRAM ON A SMALL 11 (E.G. /04,
LSI 11, ETC.) WHERE THERE IS NO CONSOLE SWITCH
REGISTER IT IS IMPERATIVE TO REMEMBER THIS SETUP.

NOTE 2) BEFORE PROCEEDING TO STEP B, ENSURE THAT THE FOLLOWING
MODIFIABLE LOCATIONS CONTAIN THE PARAMETERS YOU REQUIRE
FOR TESTING. THE FOLLOWING TABLE DESCRIBES EACH LOCATION
WITH RESPECT TO THE DEFAULT PARAMETERS WHICH WILL BE USED
IF LEFT UNMODIFIED BY THE USER:

LOCATION	LABEL	CONTENTS	PROGRAM REACTION
1200	OD:	0	TRACKS 0,52,53,114(8)
1202	FIRST:	015001	SECTORS 1 THRU 32(8)
1204	KRXVEC:	264	ASSUMES PROPER DEVICE VECTOR
1206	RXCS:	177170	ASSUMES PROPER DEVICE STATUS REGISTER
			(CALCULATES 'RXDB' ADDRESS FROM)
1212	DTESTP:	0	TESTS BOTH UNITS AUTOMATICALLY SEQUENCES
			THRU ALL TESTS
1214	BRLEV:	5	ASSUMES PROPER DEVICE 'BR' LEVEL

REFERENCE SECTION 2 OF THIS DOCUMENT FOR A MORE THOROUGH DESCRIPTION
OF EACH OF THESE ITEMS AND HOW TO MODIFY THESE LOCATIONS IF YOU
DESIRE TO CHANGE THE ABOVE MENTIONED DEFAULT TESTING PARAMETERS.

B. START THE PROGRAM AT LOCATION 200

C. THE PROGRAM WILL TYPE OUT MAINDEC NUMBER, A TEST PARAMETER OF 0 (USE BOTH DRIVES AND RUN ALL TESTS), THEN TYPE TRACKS TO BE ACCESSED AND SECTOR LIMITS. THE PROGRAM IS NOW RUNNING ALL TESTS IN SEQUENCE.

D. IF THERE ARE NO ERRORS, AT THE END OF THE PASS (APPROX. 50 SECONDS RUN TIME), A "D" WILL BE TYPED AND IT WILL CONTINUE ON FOR ANOTHER PASS.

E. TO HALT THE TEST AT ANY TIME (AFTER OR BEFORE COMPLETION OF A PASS) JUST HALT THE PROCESSOR.

F. AFTER COMPLETING A PASS OF THE DIAGNOSTIC, THE RX11 RELIABILITY TEST MAY BE RUN.

G. THERE ARE TWO TYPES OF ERROR PRINT OUT FORMATS

1. TESTS PRETEST, 1 - 17, AND 21 - 23 USE THE FORMAT SHOWN IN SECTION 3.1. THE IMPORTANT ADDRESS THERE IS THE "ERADR" (ERROR ADDRESS) GO TO THE LISTING AT THAT LOCATION TO GET MORE INFORMATION ON THE ERROR CONDITION
2. TEST 20, AND 24 - 26 USE THE FORMAT SHOWN IN SECTION 3.3. IN THIS CASE THE "TEST PC" IS THE ADDRESS OF THE TEST BEING RUN WHEN THE ERROR OCCURED. THEN THE VITAL INFORMATION OF THE ERROR IS PRINTED (CONTENTS OF ALL REGISTERS, ADDRESS OF WHERE ON THE DISKETTE THE ERROR OCCURED, AND THE TYPE OF ERROR).

2.1 LOADING THE PROGRAM

LOAD THE PROGRAM INTO MEMORY USING THE STANDARD PROCEDURE FOR BINARY PAPER TAPES. MAKE SURE THE TOTAL SYSTEM IS READY FOR OPERATION. THE DISKETTES INSERTED PROPERLY, DOORS CLOSED ON DRIVES TO BE TESTED ETC.

2.2 STARTING ADDRESSES

THE PROGRAM HAS TWO STARTING ADDRESS LOCATIONS AS FOLLOWS:

2.2.1 INITIAL START [LOC.200]

THIS STARTING ADDRESS TESTS FOR AND SELECTS THE HARDWARE, OR SOFTWARE SWITCH REGISTER, PRINTS MAINDEC NAME AND REVISION, THE TEST AND DRIVE SELECTION, AND TRACKS AND SECTORS BEING USED.

2.2.2 RESTART [LOC.202]

THIS STARTING ADDRESS DIRECTS THE PROGRAM TO CONTINUE RUNNING USING THE DRIVE AND TEST SELECTIONS SPECIFIED IN THE PREVIOUS INITIAL START.

2.3 OPERATOR ACTION BEFORE STARTING THE PROGRAM

2.3.1 DEVICE ADDRESS SELECTION

LIKE MOST OPTIONS ON THE PDP-11 THE RX11 INTERFACE CARD HAS JUMPERABLE REGISTER AND VECTOR ADDRESSES. THIS ALLOWS FOR DEVICES WITH THE SAME STANDARD ADDRESSES TO BE JUMPED TO AN OTHER ADDRESS SO THEY WILL RUN WITHOUT CONFLICT.

THE PROGRAM MUST KNOW WHAT ADDRESSES ARE BEING USED, AS IT IS THROUGH THESE REGISTER AND VECTOR ADDRESSES THAT ALL COMMUNICATION BETWEEN THE PDP-11 AND THE RX11 IS HANDLED.

IF THE RX11 SYSTEM UNDER TEST IS JUMPED FOR REGISTER ADDRESSES OTHER THAN STANDARD, WHICH IS RXCS = 177170 AND RXDB = 177172 PLACE IN THE MEMORY LOCATION CALLED "RXCS" [LOC. 1206] ITS NEW ADDRESS. THE PROGRAM ASSUMES THE NEXT EVEN ADDRESS ABOVE THAT OF RXCS, WILL BE THE ADDRESS OF RXDR, SO SETTING THAT ADDRESS IS NOT NECESSARY. IF THERE IS A NONSTANDARD INTERRUPT VECTOR ADDRESS (STANDARD IS LOC. 264) THEN PLACE IN MEMORY LOCATION CALLED "KRXVEC" [LOC. 1204] ITS NEW ADDRESS.

IF EITHER OF THESE LOCATIONS IS LOADED WITH A WRONG ADDRESS, THE PROGRAM WILL GET UNPREDICTABLE ERRORS AND MAY HALT.

NOTE:

THE PROGRAM EXPECTS THAT THE PRIORITY LEVEL JUMPERS ARE SET FOR A NORMAL 'BR' LEVEL OF 5 (CONTENTS OF PROGRAM LOCATION 'BRLEV;' IS SET TO 5). IF THE PRIORITY LEVEL JUMPERS ARE SET TO ANY OTHER LEVEL TESTS 3 & 4 WILL REPORT ERRORS, UNLESS PROGRAM LOCATION 'BRLEV;' HAS BEEN PATCHED TO CONTAIN THE RELEVANT 'BR' LEVEL BEFORE EXECUTING THE PROGRAM,

IF THIS IS BEING TESTED ON A LSI 11, TESTS 3 AND 4 WILL NOT BE RUN AS THE LSI 11 HAS ONLY 1 LEVEL OF INTERRUPT.

2.3.2 NON-STANDARD DISKETTE ADDRESS SELECTION

IF IT IS DESIRABLE TO TEST THE DISKETTE BETWEEN TRACK AND SECTOR ADDRESS LIMITS OTHER THAN THE PRESELECTED TRACK ADDRESSES, AND/OR MINIMUM (FIRST) AND MAXIMUM (LAST) SECTOR ADDRESSES, THIS IS DONE BY THE OPERATOR MAKING CHANGES TO TWO MEMORY LOCATIONS BEFORE THE PROGRAM IS STARTED. ONE LOCATION IS CALLED "OD" [LOC. 1200] WHICH CONTAINS THE TWO BYTES FOR INNER AND OUTER TRACK ADDRESSES. THE OTHER LOCATION IS CALLED "FIRST" AND IT CONTAINS THE TWO BYTES FOR THE FIRST AND LAST SECTOR ADDRESSES.

A. DEFINITIONS

- OD = ADDRESS OF TRACK AT OUTER DIAMETER (MIN. 0)
- ID = ADDRESS OF TRACK AT INNER DIAMETER (MAX. 114)
- FIRST = ADDRESS OF FIRST SECTOR ON A TRACK (MIN. 1)
- LAST = ADDRESS OF LAST SECTOR ON A TRACK (MAX. 32)

B. LOCATIONS

TRACKS LOCATION 1200	BITS 14	-----8	6	-----0	OD
SECTORS LOCATION 1202	BITS 12	-----8	4	-----0	FIRST

C. RESTRICTIONS

THE VALUE OF "OD" MUST BE LESS THAN OR EQUAL TO THE VALUE OF "ID".
THE VALUE OF "FIRST" MUST BE LESS THAN OR EQUAL TO THE VALUE OF "LAST".

IF THESE LOCATIONS ARE CHANGED TO NEW LIMITS, THEN THE PROGRAM WILL ACCESS ONLY THOSE ADDRESSES INCLUSIVE OF AND BETWEEN THESE LIMITS.
THE EXCEPTION TO THIS IS TEST 26 WHICH ALWAYS USES A SPECIAL TRACK SEQUENCE.

IF THE "OD" LOCATION IS CLEARED OR SET TO ANY ILLEGAL COMBINATION OF TRACKS, THE PROGRAM WILL CLEAR LOCATION "OD". THE TRACK SEQUENCE WILL THEN BE TRACKS 0, 52, 53, AND 114 (OCTAL) ONLY.

IF THE "FIRST" LOCATION IS CLEARED OR SET TO ANY ILLEGAL COMBINATION OF SECTOR ADDRESSES LIMITS THEN THE PROGRAM WILL SET "FIRST" TO 1 AND "LAST" TO 32 (OCTAL).

FOR THE PDP 11 PROCESSORS THAT DO NOT HAVE A HARDWARE SWITCH REGISTER OR IF THE OPERATOR WISHES TO SELECT THE SOFTWARE SWITCH REGISTER, BY PUTTING ALL THE SWITCHES UP TO A "1", (THIS MUST BE DONE EACH TIME THE PROGRAM IS STARTED AT LOCATION 200, OTHERWISE THE PROGRAM WILL USE THE HARDWARE SWR.) LOCATION 176 IS ASSIGNED AS THE SWITCH REGISTER. BITS SET TO A "1" IN THIS LOCATION HAVE THE SAME FUNCTION AS THE CORRESPONDING SWITCH IN THE HARDWARE SWITCH REGISTER. ALL REFERENCES TO THE SWR ARE INDIRECT AND THE PROGRAM ASSIGNS THE CORRECT ADDRESS OF THE SWR AT "INITIAL START". SEE SECTION 2.4.2 FOR THE SELECTION OF OPERATING CONDITIONS.

TO CHANGE THE SOFTWARE SWR. WHILE THE PROGRAM IS RUNNING, TYPE "CONTROL G". EACH TIME THE SWR. IS TO BE TESTED THE PROGRAM WILL CHECK TO SEE IF THE SOFTWARE SWR IS SELECTED, AND THE PROGRAM IS NOT RUNNING IN AUTO MODE OF RXDP/ACT11. IF BOTH CONDITIONS EXIST THEN THE PROGRAM CHECKS FOR THE CTRL G IN THE KEYBOARD BUFFER. IF THE CTRL G IS THERE THE CONTENTS OF THE SOFTWARE SWR. ARE PRINTED AND A "NEW =" IS ASKED FOR. THE OPERATOR MAY NOW TYPE IN THE NEW SWITCH REGISTER CONTENTS, TERMINATED BY A CARRIAGE RETURN (CR), OR IF HE DOESN'T WANT TO CHANGE THE SWR. JUST TERMINATE WITH THE (CR). NOTE SEE THE CHARACTER RESTRICTIONS BELOW.

WHEN THE PROGRAM DETECTS THE (CR) IT WILL REPLACE THE CONTENTS OF THE SOFTWARE SWR., IF A NEW ONE HAS BEEN TYPED IN, AND RETURN TO THE FLOW OF THE PROGRAM.

NOTE: CHARACTER RESTRICTIONS FOR CHANGING THE SOFTWARE SWR.

1. ONLY OCTAL NUMBERS 0 - 7 ARE ACCEPTED. ANY OTHER CHARACTER TYPED WILL BE PRINTED AS A ? AND THE WHOLE SWR MUST BE RETYPED.
2. TO WIPE OUT A "NEW" CONTENTS JUST TYPED IN, TYPE CTRL U. NOW A NEW CONTENTS CAN BE RETYPED.
3. ONLY 6 OCTAL CHARACTERS WILL BE PUT INTO THE SWR. IF MORE THAN 6 CHARACTERS ARE TYPED IN ONLY THE LAST 6 WILL BE PUT INTO THE SWR.

THE DRIVE AND TEST DELECTION MUST BE MADE BEFORE THE PROGRAM STARTS.

LOCATION "DTESTP" (LOC. 1212) IS WHERE THE BITS ARE SET TO TELL THE PROGRAM WHAT DRIVES ARE WANTED AND WHAT TESTS TO RUN AS INDICATED BELOW. WHEN THE PROGRAM STARTS IT WILL PRINT OUT THE CONDITIONS UNDER WHICH IT IS RUNNING.

BIT 15 (1) SELECT DRIVE UNIT 1
BIT 14 (1) SELECT DRIVE UNIT 0

NOTE: IF NEITHER OF THE ABOVE BITS ARE SET TO A 1, THEN THE PROGRAM EXPECTS BOTH DRIVES TO BE READY FOR OPERATION (POWER ON, DISKETTES INSERTED, AND DOORS CLOSED).

THEN SET THE TEST SELECTION IN BITS 4,3,2,1, AND 0 AS FOLLOWS:

"DTESTP" BITS 15 14 13-----5 4 3 2 1 0
U1 U0 NOT USED TESTS

BITS		4 3		2 1		0		TESTS	
0	0	0	0	0	0	0	0	(IF NO TEST SELECTED DEFAULTS TO TEST 1)	
0	0	0	0	0	0	0	1	TEST 1	
0	0	0	0	0	0	1	0	TEST 2	
0	0	0	0	0	1	1	0	TEST 3	
0	0	0	0	1	1	0	0	TEST 4	
0	0	0	1	0	0	0	0	TEST 5	
0	0	0	1	0	1	0	1	TEST 6	
0	0	0	1	1	1	0	0	TEST 7	
0	0	0	1	1	1	1	1	TEST 10	
0	1	0	0	0	0	0	0	TEST 11	
0	1	0	0	0	0	1	1	TEST 12	
0	1	0	1	0	1	0	0	TEST 13	
0	1	0	1	1	1	1	1	TEST 14	
0	1	1	0	0	0	0	0	TEST 15	
0	1	1	0	1	0	1	1	TEST 16	
0	1	1	1	1	1	0	0	TEST 17	
0	1	1	1	1	1	1	1	TEST 20	
1	0	0	0	0	0	0	1	TEST 21	
1	0	0	0	0	1	0	0	TEST 22	
1	0	0	1	1	1	1	1	TEST 23	
1	0	1	0	0	1	0	0	TEST 24	
1	0	1	0	1	0	1	1	TEST 25	
1	0	1	1	1	1	0	0	TEST 26	

NOTE1: SELECTION OF TESTS 27 THROUGH 37 WILL CAUSE THE MESSAGE "ILLEGAL TEST" TO BE PRINTED.

NOTE2: WHEN A SPECIFIED TEST IS SELECTED THE PROGRAM WILL START AT THAT TEST AND THEN RUN THROUGH ALL THE FOLLOWING TESTS UNTIL IT COMPLETES TEST 26, INDICATED BY THE EOP TYPE OUT. THEN IT WILL GO BACK TO THE TEST SELECTED AND START THE NEXT PASS. (IE. IF TEST 24 IS SELECTED THE PROGRAM WILL RUN TEST 24, 25, AND 26, THEN GO BACK TO TEST 24.)

AN EXPANDED DEFINITION OF THE TESTS IS IN SECTION 2.5

2.3.4.1 PREREQUISITE OF TESTS:

THE FOLLOWING TESTS MUST BE RUN IN ORDER, AS ONE TEST SETS UP FOR THE NEXT TEST.

TEST 6 BEFORE TESTS 7 AND TEST 10
TEST 14 BEFORE TEST 15 AND TEST 16
TEST 16 BEFORE TEST 17
TEST 21 BEFORE TEST 22 AND TEST 23

SEE SECTION 2.5 UNDER THE ABOVE TESTS FOR EXPLANATION

2.4 OPERATOR ACTION TO RUN THE PROGRAM

2.4.1 STARTING THE PROGRAM

DEPENDING UPON THE STARTING ADDRESS SELECTED THE PROGRAM WILL DO THE FOLLOWING:

SA200 (INITIAL START)

THE SELECTION OF HARDWARE OR SOFTWARE SWITCH REGISTER IS MADE THEN THE PROGRAM WILL TYPE ITS IDENTIFICATION NUMBER, THE TEST PARAMETERS SELECTED IN LOCATION "DIESIP", AND TRACKS AND SECTORS BEING TESTED. THE PROGRAM THEN PROCEEDS TO RUN UNDER THOSE CONDITIONS.

SA202 (RESTART)

THE PROGRAM WILL TYPE OUT THE TEST PARAMETERS SELECTED BY THE PREVIOUS INITIAL START, PRINTS THE DISKETTE ADDRESS LIMITS, AND STARTS RUNNING THE TESTS. THE ONLY OPERATOR ACTION REQUIRED IS TO SET THE OPERATING CONDITIONS AS DEFINED IN SECTION 2.4.2, AFTER DEPRESSING THE "LOAD ADRS" SWITCH AND BEFORE DEPRESSING THE START SWITCH.

2.4.2 OPERATING CONDITIONS

AFTER THE TEST SELECTION HAS BEEN MADE PRESS THE "CONT" SWITCH. THE PROGRAM WILL THEN ASK FOR OPERATING CONDITIONS. SWITCHES 0 AND 8 THROUGH 15 ARE USED AS INDICATED BELOW. ONCE THEY ARE SET UP AGAIN DEPRESS THE "CONT" SWITCH. THE PROGRAM IS NOW RUNNING UNDER THE SELECTED CONDITIONS.

SW15-SW0 (1) - SELECT SOFTWARE SWITCH REGISTER

NOTE: IF THERE IS A HARDWARE SWITCH REGISTER, AND THE OPERATOR WANTS THE SOFTWARE SWITCH REGISTER. PUT ALL SWITCHES UP (1) BEFORE STARTING THE PROGRAM AT THE INITIAL START ADDRESS.

SW15 (1) - HALT ON ERROR

THE PROGRAM HALTS ON DETECTING AN ERROR, AFTER PRINTING THE ERROR MESSAGE. PRESSING "CONT" RESTORES THE NORMAL OPERATION OF THE PROGRAM.

SW14 (1) - HALT AT END OF PASS

AT "END OF PASS" THE PROGRAM TYPES A BELL THEN AN EOP INDICATOR.

"D" MEANS NO ERRORS DURING THE PASS
"=" MEANS HAD ERRORS DURING THE PASS

IF SW14 IS SET THE PROGRAM WILL HALT, IF SW14 IS OFF THE PROGRAM GOES BACK TO THE TEST SELECTED AND RECYCLES THROUGH TO THE LAST TEST, AT WHICH TIME ANOTHER EOP INDICATOR IS PRINTED. IF THE PROGRAM HALTS DUE TO SW14 THEN PRESS "CONT" WILL RESTORE THE NORMAL FLOW OF THE PROGRAM. IF IT HALTS AT THE END OF A PASS IT WILL TYPE OUT THE NUMBER OF PASSES COMPLETED.

SW13 (1) - INHIBIT ERROR TYPEOUT

AT THE DETECTION OF AN ERROR IF SW13 IS SET NO ERROR PRINT OUT WILL OCCUR. IF SW13 IS OFF THE ERROR INFORMATION IS PRINTED AS DESCRIBED IN SECTION 3.0 ERROR DETECTION

SW12 (1) - LOOP ON TEST

AT THE COMPLETION OF A TEST THE PROGRAM CHECKS SW12. IF SET THE PROGRAM WILL GO BACK TO THE BEGINNING OF THAT TEST AND RERUN IT. THIS PRODUCES A SCOPE LOOP ON A PARTICULAR TEST. THE PROGRAM WILL STAY IN THIS TEST UNTIL;

- A. HALT ON END OF TEST SWITCH IS SET
- B. LOOP ON TEST SWITCH IS TURNED OFF

AT WHICH TIME THE PROGRAM WILL GO ON TO THE NEXT TEST.

NOTE: IF SW12 IS SET AND NO TEST SPECIFIED (0) THE PROGRAM WILL LOOP ON TEST 1.

NOTE: TO LOOP ON A TEST THAT REQUIRES A PREVIOUS TEST TO BE RUN FIRST (SECTION 2.3.4). SELECT THE PREREQUISITE TEST AND SET THE "HALT AT END OF TEST" SWITCH. START THE PROGRAM AND WHEN IT HALTS, SELECT THE DESIRED TEST AND SET THE "LOOP ON

IN SOME TESTS ERRORS CAN OCCUR IN SEVERAL PLACES THROUGH OUT THE TEST. WHEN THE ERROR HAS BEEN REPORTED THE PROGRAM SETS A PC FLAG TO INDICATE WHERE THE ERROR OCCURRED. IF SW11 IS SET THE PROGRAM GOES BACK TO THE BEGINNING OF THE TEST RUNNING, AND GOES THROUGH THE TEST UNTIL:

- A. IT FINDS A DIFFERENT ERROR IN AN EARLIER PART OF THE TEST IN WHICH CASE IT WILL LOCK ONTO THAT ERROR.
- B. IT DETECTS THE PC FLAG INDICATING THIS IS WHERE THE ERROR OCCURRED. IT THEN GOES BACK TO THE BEGINNING OF THE TEST AGAIN. THIS LOOP WILL CONTINUE UNTIL HALT ON ERROR SWITCH IS SET OR THE LOCK ON ERROR SWITCH IS TURNED OFF.

SW10 (1) - HALT AT END OF TEST

WHEN SET IT WILL HALT THE PROGRAM AT THE END OF THE TEST PRESENTLY RUNNING.

SW 9 - LIMIT DATA ERROR PRINT OUTS

(0) - WHEN OFF ONLY THE FIRST 10 DATA BYTE ERRORS WILL BE PRINTED ON A READ CHECK TEST, FOR EACH SECTOR. ANY MORE ERRORS WILL BE TABULATED BUT NOT PRINTED. AN ERROR ON A DIFFERENT SECTOR WILL ALLOW 10 MORE DATA BYTE ERRORS TO BE PRINTED.

(1) - WHEN SET ALL DATA BYTE ERRORS FOR ALL SECTORS WILL BE PRINTED ON AN ERROR.

SW 8 (1) - INHIBIT RECALIBRATION

NO RECALIBRATION OF THE DRIVES WILL OCCUR UPON THE DETECTION OF A SEEK ERROR IF THIS SWITCH IS SET.

SW 0 (1) - INHIBIT BELL AT ERROR

IF SW0 IS OFF THE ERROR ROUTINE WILL RING THE TELEPRINTER BELL AT EACH ERROR DETECTED. WITH SW0 SET NO BELL WILL RING.

2.5 TEST DEFINITIONS

2.5.1 PRETEST - INITIALIZE [KEY] PART I
EACH TIME THE PROGRAM IS STARTED, BY EITHER STARTING ADDRESS, IT RUNS THROUGH A PRETEST.

KEY INITIALIZE SHOULD SET THE DONE FLAG BECAUSE ANY INITIALIZATION OF THE RX01 MICROPROCESSOR IS AN IMPLIED [READ SECTOR] OF TRACK 1 SECTOR 1. THEREFORE ANY ERROR, EXCEPT PARITY, THAT MAY OCCUR FROM A NORMAL [READ SECTOR] COMMAND MAY OCCUR DURING AN INITIALIZE, CAUSING THE ERROR FLAG TO SET.

PRETEST INSURES THAT:

- A. DONE IS SET
- B. ERROR FLAG IS CLEARED
- C. TR FLAG IS CLEARED
- D. INIT DONE IS SET

- 2.5.2 TEST 1 - RXCS TEST PART I / INTERRUPT TEST PART I
THE PURPOSE OF THIS TEST IS TO VERIFY THAT WRITING ALL RXCS WRITABLE BITS TO A 0 ARE NOT WRITTEN TO A 1.

THE PROGRAM WRITES THE RXCS = 0

NO INTERRUPTS SHOULD OCCUR

THE RXCS SHOULD REMAIN UNCHANGED = 40 (DONE)

THE RXDB SHOULD = 0
- 2.5.3 TEST 2 - INTERRUPT TEST PART II / VECTOR ADDRESS VERIFICATION

THE PURPOSE OF THIS TEST IS TO VERIFY THAT WRITING THE RXCS INTERRUPT ENABLE BIT (BIT 6) TO A 1, DOES INDEED WRITE IT TO A 1, THEREFORE BECAUSE DONE IS SET AN INTERRUPT SHOULD OCCUR (THE PDP 11 PRIORITY IS 0)
- 2.5.4 TEST 3 - INTERRUPT TEST PART III / PRIORITY LEVEL TEST PART I

THE PURPOSE OF THIS TEST IS TO VERIFY THE PRIORITY OF THE INTERRUPT REQUEST LINE. THE PROGRAM SETS THE PDP-11 PRIORITY TO 4

AN RX01 INTERRUPT SHOULD OCCUR ON PRIORITY LEVEL 5

IF NO INTERRUPT OCCURS THEN THE PRIORITY LEVEL OF THE RX11 IS NOT 5, BUT MAYBE LEVELS 4,3,2,OR 1
- 2.5.5 TEST 4 - INTERRUPT TEST PART IV / PRIORITY TEST PART II

THE PURPOSE OF THIS TEST IS TO VERIFY THE PRIORITY OF THE RX11 INTERRUPT REQUEST LINE. THE PROGRAM SETS THE PDP-11 PRIORITY TO 5.

NO INTERRUPT SHOULD OCCUR
IF AN INTERRUPT DOES OCCUR THEN THE PRIORITY LEVEL OF THE RX11 IS NOT LEVEL 5, BUT MAYBE LEVEL 6, OR 7.
- 2.5.6 TEST 5 - INIT (PROGRAMMED) B / READ STATUS

THE PURPOSE OF THIS TEST IS TO VERIFY THAT SETTING THE RX11 BIT 14 CAUSES A RX01 PROGRAMMED SUBSYSTEM INITIALIZE

THE RXCS SHOULD = 40 (DONE)
THE RXDB SHOULD = 4, OR 104, OR 204, OR 304

TEST 5 CONT'D - RXCS TEST PART II / RST

THE PURPOSE OF THIS TEST IS TO VERIFY THE READ STATUS COMMAND (FUNCTION #12), AND THAT DONE BIT IS CLEARED BY THE FUNCTION.
- 2.5.7 TEST 6 - FILL BUFFER TRANSFER LENGTH TEST

THE PURPOSE OF THIS TEST IS TO VERIFY THE TRANSFER LENGTH OF THE FUNCTION "FILL BUFFER" OF THE RX01 MICROCONTROLLER

NOTE: THIS TEST LOADS THE SECTOR BUFFER FOR TEST 7 AND 10, AND MUST BE RUN PREVIOUS TO THEM.

- 2.5.8 TEST 7 - EMPTY BUFFER TRANSFER LENGTH AND CONTENT VERIFICATION PART I
THE PURPOSE OF THIS TEST IS TO VERIFY THE TRANSFER LENGTH OF THE FUNCTION "EMPTY BUFFER" AND TO VERIFY THE CONTENTS OF THE SECTOR BUFFER.
- 2.5.9 TEST 10 - EMPTY BUFFER TRANSFER LENGTH AND CONTENT VERIFICATION PART II
THE PURPOSE OF THIS TEST IS TO VERIFY THE PREVIOUS EMPTY BUFFER TEST DID NOT EMPTY AND DESTROY THE CONTENTS OF THE SECTOR BUFFER.
- 2.5.10 TEST 11 - FILL / EMPTY BUFFER WITH ALL 0'S
DURING THE EMPTY BUFFER FUNCTION THIS TEST VERIFIES THAT ALL 0'S ARE IN FACT IN THE SECTOR BUFFER.
- 2.5.11 TEST 12 - FILL / EMPTY BUFFER WITH ALL 1'S
DURING THE EMPTY BUFFER FUNCTION THIS TEST VERIFIES THAT ALL 1'S ARE IN FACT IN THE SECTOR BUFFER.
- 2.5.12 TEST 13 - DRIVE READY VERIFICATION
TESTS THAT THE DRIVE READY (RDY) BIT WILL SET FOR ALL SELECTED DRIVES. THE RDY BIT WILL BE SET AFTER A READ STATUS FUNCTION DIRECTED TO THE SELECTED DRIVE.
- 2.5.13 TEST 14 - ERROR FLAG AND B-CODE VERIFICATION PART I
THE PURPOSE OF THIS TEST IS TO VERIFY THAT TRYING TO READ A NON-EXISTANT SECTOR WILL CAUSE AN ERROR AND THE CORRECT ERROR CODE WILL BE PUT INTO THE RXDB WHEN THE STATUS B IS READ.
NOTE: THIS TEST CHECKS FOR PARITY ERROR ON THE READ STATUS B FUNCTION, THE NEXT TWO TESTS (T15 & T16) DO NOT. THIS TEST MUST BE RUN BEFORE TESTS 15 & 16.
- 2.5.14 TEST 15 - ERROR FLAG AND B-CODE VERIFICATION PART II
THIS TEST VERIFIES THAT TRYING TO WRITE DELETED DATA ON AN ILLEGAL SECTOR WILL PRODUCE AN ERROR AND THE CORRECT H-CODE IS PRODUCED. THE DELETED DATA BIT SHOULD BE SET AFTER THIS TEST.
- 2.5.15 TEST 16 - ERROR FLAG AND B-CODE VERIFICATION PART III
VERIFIES THAT A WRITE FUNCTION TO A NONEXISTANT SECTOR WILL PRODUCE AN ERROR AND THE CORRECT B-CODE IS PRODUCED. THE DELETED DATA BIT WILL ALSO BE CLEARED.
NOTE: TEST 16 MUST BE RUN BEFORE TEST 17 AS TEST 16 CLEARS THE DELETED DATA BIT AND TEST 17 TESTS THAT IT IS CLEARED.

2.5.16 TEST 17 - ILLEGAL TRACK ERROR VERIFICATION

THIS TEST VERIFIES THAT IF A TRACK ADDRESS LARGER THAN 114(OCTAL) IS ACCESSED, AN ERROR CONDITION WILL OCCUR, AND THE B-CODE WILL = 40. IT ALSO EXPECTS THE DELETED DATA BIT TO BE CLEARED.

2.5.17 TEST 20 - SEEK VERIFICATION VIA READ FUNCTION

THIS TEST DOES A READ FUNCTION ON THE SELECTED TRACKS TESTING FOR SEEK ERRORS ON VARIOUS SECTIONS OF THE DISKETTE.

2.5.18 TEST 21 - WRITE TEST

THE PURPOSE OF THIS TEST IS TO WRITE ALL ONES ON SECTOR 1, TRACK 1, AND TO VERIFY THAT THE DATA IN THE SECTOR BUFFER IS NOT CHANGED. NOTE: THIS TEST MUST BE RUN BEFORE TESTS 22 & 23 AS THEY CHECK FOR DATA WRITTEN ON TRACK 1 SECTOR 1.

2.5.19 TEST 22 - INITIALIZE IMPLIED READ

AFTER PREVIOUSLY WRITING DATA ON TRACK 1 SECTOR 1, THIS TEST CHANGES THE CONTENTS OF THE SECTOR BUFFER AND DOES A PROGRAMMED INITIALIZE. AT THE END OF AN INIT.(RECAL.) THE SECTOR BUFFER MUST CONTAIN THE DATA FROM TRACK 1 SECTOR 1. NOTE: UNIT 0 MUST BE ON-LINE FOR THIS TEST TO WORK.

2.5.20 TEST 23 - READ TEST

THIS TEST VERIFIES THAT A READ FUNCTION DOES INFACIT LOAD THE SECTOR BUFFER WITH DATA READ FROM THE SELECTED ADDRESS.

2.5.21 TEST 24 - DATA TRANSFER AND VERIFICATION

THE PURPOSE OF THIS TEST IS TO WRITE THEN READ AND CHECK DATA ON ALL SECTORS OF THE SELECTED TRACKS. THE TEST ALTERNATES BETWEEN DRIVES, IF BOTH DRIVES ARE SELECTED, BEFORE CHANGING TRACKS. THE DATA PATTERN USED IS A FLOATING 0 PATTERN.

2.5.22 TEST 25 - DATA VERIFICATION VIA DELETED DATA MODE.

THIS TEST IS THE SAME AS TEST 24 EXCEPT IT CHECKS FOR DELETED DATA INDICATORS AND USES A DATA PATTERN OF FLOATING 1.

2.5.23 TEST 26 - HEAD "HOME" TEST

THIS TEST CHECKS FOR THE "HOME FOUND BEFORE THE DESIRED TRACK WAS REACHED" ERROR CODE. THE HEAD IS MOVED OUT 10 TRACKS THEN DECREMENTED BACK TO TRACK 0. IT TESTS ALL SELECTED DRIVES, AND USES A DATA PATTERN OF RANDOM DATA.

3.0 ERRORS

PRETEST AND TESTS 1 - 17, AND TESTS 21 - 23 HANDLE ERRORS AS INDICATED IN SECTION 3.1. FOR THE MOST PART THESE TESTS DO NOT RELY ON AN INTERRUPT TO INDICATE THE FUNCTION IS COMPLETED. WHEREAS THE OTHER TESTS (TESTS 20, AND 24 - 26) DO READ, WRITE AND READ CHECK FUNCTIONS OVER THE SELECTED TRACK, SECTORS, AND DRIVES. THESE REQUIRE THE INTERRUPT SERVICE AND ERROR DETECTION THAT WAS USED IN THE DATA RELIABILITY TEST. THIS IS DESCRIBED IN SECTION 3.3.

NOTE: IF LOOP ON ERROR SWITCH IS UP THEN THE PROGRAM WILL LOOP ON THE SHORTEST SET OF INSTRUCTIONS THAT WILL KEEP IT IN THE FAILING LOOP. OTHERWISE AFTER REPORTING THE ERROR THE PROGRAM WILL CONTINUE RUNNING THROUGH THE REMAINING ADDRESSES AND TESTS.

3.1 ERROR HEADING FOR TESTS 1 - 17, AND 21 - 23 PLUS PRETEST.

THE ERROR HEADING IS AS FOLLOWS:

ERADR	FAST	FAPT	[BLANK]	GOOD	BAD
-------	------	------	---------	------	-----

UNDER EACH COLUMN THE ERROR ROUTINE PRINTS PERTINENT INFORMATION.

ERADR = ERROR ADDRESS
ADDRESS OF THE ERROR TRAP INSTRUCTION WHERE
THE ERROR WAS DETECTED.

FAST = FIRST ADDRESS OF SELECTED TEST
ADDRESS OF THE TEST SELECTED AND RUNNING

FAPT = FIRST ADDRESS OF PRESENT TEST
ADDRESS OF THE TEST OR SUBTEST PRESENTLY RUNNING, OR
ADDRESS OF THE SCOPE LOOP.

[BLANK] ADDITIONAL GENERAL INFORMATION SUPPLIED BY SOME
TESTS ON AN ERROR.

GOOD = EXPECTED RESULTS OF THE TEST
TEST RESULTS OF WHAT SHOULD HAVE HAPPENED IF
THERE WAS NO ERROR.

BAD = ACTUAL TEST RESULTS
THE DATA THAT WAS RECEIVED FROM THE RX01,
THAT CAUSED THE ERROR.

PASS = NUMBER OF PASSES MADE UP TO THIS ERROR

3.2

ERROR OUTPUT PER TEST

THE FOLLOWING ARE THE TYPES OF PRINT OUTS UNDER THE COLUMNS [BLANK], GOOD, AND BAD FOR THE VARIOUS TESTS, USING THIS ERROR FORMAT.

TEST (SECTION)	[BLANK] (R2)	GOOD (R0)	BAD (R1)
----	-----	-----	-----
PRETEST (1)	N/A	40	(RXCS)
PRETEST (2)	(RXCS) INCL, DD BIT	4 OR 204	(RXCS) NO DD BIT
TEST 1 (1)	N/A	40	(PXCS)
TEST 1 (2)	N/A	0	(RXCS)
TEST 1 (3)	(KRXVEC)	N/A	N/A
TEST 2 (1)	(KRXVEC)	N/A	N/A
TEST 2 (2)	(KRXVEC)	140	(RXCS)
TEST 2 (3)	(KRXVEC)	40	(RXCS)
TEST 2 (4)	(KRXVEC)	40	(RXCS)
TEST 2 (5)	(KRXVEC)	40	(RXCS)
TEST 3 (1)	(KRXVEC)	N/A	N/A
TEST 4 (1)	(KRXVEC)	N/A	N/A
TEST 5 (1)	N/A	40	(RXCS)
TEST 5 (2)	(RXDB) INCL, DD BIT	4 OR 204	(KXDB) NO DD BIT
TEST 5 (3)	N/A	0	(RXCS)
TEST 5 (4)	N/A	40	(RXCS)
TEST 5 (5)	(RXCS) INCL, DD BIT	200	(RXCS) NO DD BIT
TEST 6 (1)	NO. OF XFERS	N/A	N/A
TEST 7 (1)	NO. OF XFERS	EXPEC. DATA	ACTUAL DATA
TEST 10 (1)	NO. OF XFERS	EXPEC. DATA	ACTUAL DATA

TEST 11&12 (1) [USES TEST 6&7 TO FILL / EMPTY BUFFER]

SEQ 0019

TEST 13 (1) (RXDB) 200 (RXDB)
NO DD BIT

TEST 13 (2) (RXDB) 200 (RXDB)
NO DD BIT

TEST 14 (1) NO. OF TR'S 100040 (RXCS)

TEST 14 (2) (RXDB) 0 (RXDB)
NO DD BIT

TEST 14 (3) (RXDB) 40 (RXCS)

TEST 14 (4) N/A 70 (RXDB)
ERROR CODE

TEST 15 (1) NO. OF TR'S 100040 (RXCS)

TEST 15 (2) N/A 100 (RXDB)

TEST 15 (3) N/A 70 (RXDB)
ERROR CODE

TEST 16 (1) NO. OF TR'S 100040 (RXCS)

TEST 16 (2) N/A 0 (RXDB)

TEST 16 (3) N/A 70 (RXDB)
ERROR CODE

TEST 17 (1A) (RXDB) 0 (RXCS)

TEST 17 (1B) N/A 100040 (RXCS)

TEST 17 (2) N/A 0 (RXDB)

TEST 17 (3) (RXDB) 40 (RXCS)

TEST 17 (4) N/A 40 (RXDB)
ERROR CODE

TEST 21 (1) (RXES) NO. OF (RXDB)
STATUS A BYTE STATUS B

TEST 21 (2) [USES TEST 7 TO EMPTY BUFFER]

TEST 22 [USES TEST 6 & 7 TO FILL AND EMPTY BUFFER]

TEST 23 [USES TEST 6 & 21 TO FILL AND CHECK BUFFER]

AS PREVIOUSLY STATED THESE TESTS ACCESS ALL THE SELECTED SECTORS, TRACKS, AND DRIVES, AND RELY ON THE INTERRUPT SERVICE ROUTINE TO INDICATE THAT A FUNCTION IS COMPLETED OR AN ERROR OCCURED. ALL ERRORS, WITH THE EXCEPTIONS WHERE NOTED, WILL TYPE AS ITS FIRST OR SECOND LINE OF THE MESSAGE "ERROR CONDITIONS TEST PC = XXXX PASS = X".

THE TEST PC NUMBER IS THE STARTING ADDRESS OF THE TEST RUNNING, AND THE PASS NUMBER IS THE NUMBER OF PASSES MADE UP TO THE ERROR

ON MOST ERRORS THE PROGRAM WILL TYPE OUT THE CONTENTS OF "STATUS A" AND "STATUS B".

STATUS A IS THE CONTENTS OF THE RXES (ERROR AND STATUS REGISTER) AT THE TIME THE ERROR WAS DETECTED. IT SHOWS THE CRC, PAR, ETC. ERRORS

STATUS B IS THE "DEFINITIVE ERROR CODES" THAT THE RX01 DETECTED, THAT MAY HAVE CAUSED THE ERROR CONDITION. THESE ERROR CODES ARE DEFINED IN SECTION 3.3.4

THERE ARE THREE CATEGORIES OF ERRORS AS LISTED AND DESCRIBED BELOW.

3.3.1 NO ERROR FLAG ERRORS

THESE ARE ERRORS THAT CAN OCCUR BUT THE ERROR FLAG IN THE RXCS WILL NOT BE SET.

A. UNEXPECTED OR MISSING DELETED DATA BIT

THIS ERROR RESULTS WHEN THE PROGRAM EXPECTS AND DOESN'T SEE THE DD BIT ("D D MARK MISSING"), OR DOESN'T EXPECT AND FINDS THE DELETED DATA BIT SET ("UNEXPECTED D D MARK"). THE PROGRAM WILL TYPE OUT AT WHAT DISKETTE ADDRESS THIS OCCURED THEN CONTINUE TESTING.

NOTE: SEE SECTION 3.3.3 FOR OTHER CAUSES OF THIS ERROR.

B. DATA NO STATUS ERROR

THIS ERROR OCCURS DURING A READ CHECK WHEN THE DATA READ DOES NOT MATCH THE DATA IN THE MEMORY DATA BUFFER, AND THERE WAS NO CRC ERROR INDICATED. THIS MEANS THAT THE DATA WAS PROBABLY READ OFF THE DISKETTE CORRECTLY BUT THE TRANSFER BETWEEN THE SECTOR BUFFER AND THE RXDB IN THE RX11 PRODUCED BAD DATA.

THE ERROR MESSAGE WILL INCLUDE THE DISKETTE ADDRESS, "BYTE" NUMBER IN THE SECTOR, THE DATA READ FROM THE SECTOR BUFFER "BAD", AND THE EXPECTED DATA FROM THE MEMORY BUFFER "GOOD".

BYTE # BAD GOOD
(THE DATA PATTERNS ARE FORMATTED AS SHOWN)

0 (TRACK ADDRESS; BITS 6 - 0)
1 (UNIT NUMBER BIT 7)
(SECTOR ADDRESS BITS 4 - 0)

BYTES 2 - 125 CONTAIN THE SELECTED DATA PATTERN.

126 (THE SUM OF ALL BYTES 0 - 125)
127 (THE NEGATIVE OF 2 TIMES BYTE 125)

THE PROGRAM DETECTS A CHECKSUM ERROR BY SUMMING ALL THE DATA READ FROM THE SECTOR BUFFER AND COMPARING THAT SUM TO 0.

AT THE END OF THE DATA ERROR TYPEOUT THE PROGRAM PRINTS IF THE CHECKSUM ACCUMULATED WAS "GOOD" OR "BAD". IF BYTES 0 OR 1 HAVE DATA ERRORS THE OPERATOR MUST CHECK THE RESULTS OF THE CHECKSUM. IF IT IS ALSO BAD, THEN THERE WAS A TRUE DATA ERROR. IF THE CHECKSUM WAS GOOD, THEN IT MIGHT BE THAT THE HEAD IS NOT OVER THE TRACK EXPECTED, AND THERE IS A POSITIONING ERROR.

IF SWITCH 9 IS DOWN THEN ONLY 10 DATA ERRORS WILL BE PRINTED, AND AT THE END OF THE SECTOR THE "TOTAL READ CHECK ERRORS =" WILL BE TYPED. IF SWITCH 9 IS UP THEN ALL THE DATA ERRORS FOR THAT SECTOR WILL BE TYPED OUT.

C. POWER FAILURE

THE PROGRAM TESTS FOR TWO TYPES OF POWER FAILURE, TOTAL SYSTEM POWER LOSS, AND RX11 POWER LOSS RESULTING IN A RECALIBRATION OF THE DRIVES.

THE TOTAL SYSTEM POWER FAILURE IS DETECTED BY "SYSMAC" SUBROUTINE "SPOWER". WHEN THE POWER IS DETECTED TO BE GOING DOWN, THE REGISTERS ARE SAVED, WHEN THE POWER COMES BACK UP THE REGISTERS ARE RESTORED AND THE MESSAGE "POWER" IS PRINTED. THE PROGRAM THEN RESTARTS.

LOSS OF POWER IN THE RX11 CAUSES A RECALIBRATION OF ALL DRIVES. WHEN THIS HAPPENS THE "INIT DONE" BIT IS SET IN THE RXES REGISTER ALONG WITH THE NORMAL DONE FLAG. AT EACH INTERRUPT THE PROGRAM TESTS FOR THE INIT DONE BIT. IF IT IS FOUND SET, THE FUNCTION WAS NOT COMPLETED AND A POWER LOSS MUST HAVE BEEN DETECTED. WHEN THIS HAPPENS THE PROGRAM TYPES OUT "RX11 POWER" AND RESTARTS. THE ERROR HEADING IS NOT TYPED ON THIS ERROR.

D. UNKNOWN INTERRUPT

IF AN INTERRUPT OCCURS THROUGH THE RX11 INTERRUPT VECTOR ADDRESS AND NONE OF THE STATUS BITS ARE SET (DONE, ERROR, ETC.) THE PROGRAM WILL TYPE "UNKNOWN INTERRUPT" AND RETURN BACK TO THE PROGRAM TO CONTINUE THE FUNCTION.
THE ERROR HEADING IS NOT PRINTED.

E. NO INTERRUPT AT DONE

THE PROGRAM EXPECTS AN INTERRUPT AT DONE ON THE FUNCTIONS OF THESE TESTS. IF AN INTERRUPT DOES NOT OCCUR AT DONE TIME THEN THE PROGRAM WILL TYPE OUT "NO INTERRUPT AT DONE ERROR" THEN GO INTO THE INTERRUPT SERVICE ROUTINE AS IF AN INTERRUPT DID OCCUR. AT THIS POINT OTHER ERRORS MAY BE PRINTED IF ANY ARE DETECTED.

3.3.2 ERROR FLAG ERRORS

THESE ERRORS ARE DETECTED AS THE RESULTS OF THE ERROR BIT BEING SET IN THE RXCS AT AN INTERRUPT.

A. PARITY ERROR

A PARITY ERROR RESULTS FROM AN INCORRECT TRANSFER OF A COMMAND WORD FROM THE RX11 INTERFACE TO THE RX01 MICRO-PROCESSOR CONTROLLER. THE PROGRAM WILL TYPE OUT THE CONTENTS OF THE COMMAND STATUS REGISTER (RXCS) SHOWING THE FUNCTION THAT FAILED, THE ADDRESS OF THE ERROR, CONTENTS OF STATUS A (RXES) WITH THE PARITY BIT SET, CONTENTS OF STATUS B (RXDB) WITH THE DEFINITIVE ERROR CODE OF 210 SET. THEN A "READ, WRITE, FILL BUFFER OR EMPTY BUFFER PARITY ERROR" WILL BE PRINTED.
IF A PARITY ERROR OCCURS ON A "READ DEFINITIVE ERROR CODE" FUNCTION, THEN THE CONTENTS OF THE RXCS AND "PARITY ERROR" WILL BE TYPED OUT.

B. CRC ERRORS

ON ALL DATA TRANSFERS BETWEEN THE SECTOR BUFFER AND THE DISKETTE, A CRC WORD IS GENERATED AND CHECKED. IF AN ERROR IS DETECTED BY THE MICRO-PROCESSOR IN THIS CRC WORD THEN A CRC ERROR IS GENERATED.
THE PROGRAM AGAIN TYPES OUT THE CONTENTS OF THE REGISTERS (RXCS CONTAINS FUNCTION, STATUS A WITH "CRC ERR" BIT SET, STATUS B WITH AN ERROR CODE OF 200). THEN IF IT IS A READ ONLY FUNCTION, OR A READ CHECK FUNCTION AND THERE WERE DATA ERRORS IT WILL TYPE OUT "DATA CRC ERRORS" THEN PRINT THE BAD BYTES IF ANY. IF IT WAS A READ CHECK FUNCTION AND THERE WERE NO DATA ERRORS IT WILL PRINT "CRC ERROR NO DATA ERROR".

C. SEEK ERRORS

ANY ERROR THAT PRODUCES A DEFINITIVE ERROR CODE BUT DOES NOT SET AN ERROR BIT IN STATUS A (RXDH AT END OF FUNCTION) IS LABELED A SEEK ERROR. SEE SECTION 3.3.4 FOR ERROR CODES AND MEANINGS.

THE SAME INFORMATION IS PRINTED FOR THESE ERRORS AS IN PARITY, OR CRC ERRORS, EXCEPT IT STATES THAT IT IS A "WRITE OR READ SEEK ERROR".

IF SWITCH 8 IS DOWN THEN AT EACH SEEK ERROR FOUND THE PROGRAM DOES AN INITIALIZE OF THE RX01 SO IT WILL RECALIBRATE TO A KNOWN (HOME) POSITION. THE PROGRAM THEN GOES ON TO THE NEXT SECTOR OR TRACK AND CONTINUES TESTING, IF THE LOOP ON ERROR SWITCH IS OFF. (SEE SECTION 3.3.3 FOR ERRORS CAUSED BY PREVIOUS ERRORS.) IF THE LOOP ON ERROR SWITCH IS UP IT WILL RETRY THE FUNCTION AT THE SAME ADDRESS.

IF SWITCH 8 IS UP THEN NO "INITIALIZE" IS DONE AND THE PROGRAM LOOKS AT THE OTHER SWITCHES FOR OPERATING CONDITIONS. SEEK ERRORS ALSO PRINT THE TRACK ADDRESS THAT THE HEAD MOVED FROM AT THE TIME OF THE ERROR.

D. ERROR FLAG ERROR

IF THE ERROR FLAG IS NOT SET IN THE RXCS AND AN ERROR BIT IS SET IN STATUS A OR AN ERROR CODE IS SET IN STATUS B THEN THERE WAS AN ERROR BUT THE ERROR FLAG WAS NOT SET. THE MESSAGE "ERROR FLAG ERROR" IS PRINTED THEN THE PROGRAM CONTINUES TO TYPE OUT THE TYPE OF ERROR.

3.3.3 ERRORS RESULTING FROM PREVIOUS ERRORS

IF THERE IS A "WRITE SEEK ERROR" THE PROGRAM WILL GO ON TO THE NEXT ADDRESS WITHOUT WRITING ON THE ADDRESS WHERE THE ERROR OCCURED. (UNLESS THE LOOP ON ERROR SWITCH 11 IS UP AND THE SEEK ERROR IS RECOVERED.) IF THE WRITE FUNCTION IS FOLLOWED BY A READ CHECK FUNCTION AND THE READ DOES NOT HAVE A SEEK ERROR AT THE SAME ADDRESS. THEN THERE MAY BE DATA ERRORS, OR UNEXPECTED OR MISSING DELETED DATA BIT ERRORS RESULTING FROM NO DATA BEING WRITTEN ON THAT ADDRESS BY THE PREVIOUS WRITE FUNCTION.

3.3.4 DEFINITIVE ERROR CODES

THE RX01 MICRO-PROCESSOR HAS DEFINED THE ERROR CODES AND MEANINGS WHICH ARE AVAILABLE TO THE PROGRAM BY ISSUING COMMAND #7 "READ DEFINITIVE ERROR CODE" THE FOLLOWING ARE THE CODES AND THEIR MEANINGS

- 10 - DRIVE 0 FAILED TO SEE HOME FROM INITIALIZE
- 20 - DRIVE 1 FAILED TO SEE HOME FROM INITIALIZE
- 30 - HOME FOUND WHEN STEPPING OUT 10 TRACKS FOR INIT.
- 40 - TRIED TO ACCESS A TRACK GREATER THEN 76
- 50 - HOME FOUND BEFORE DESIRED TRACK WAS REACHED
- 60 - SELF DIAGNOSTIC ERROR
- 70 - DESIRED SECTOR NOT FOUND AFTER SAMPLING 52 HEADERS
- 100 - WRITE PROTECT ERROR
- 110 - MORE THEN 40 US AND NO SEP CLOCK DETECTED
- 120 - A PREAMBLE COULD NOT BE FOUND
- 130 - PREAMBLE FOUND BUT NO ID MARK FOUND IN TIME
- 140 - CRC ERROR ON A HEADER, NO ERROR FLAG
- 150 - GOOD HEADER (NO CRC ERROR) BUT TRACK COMPARE ERROR
- 160 - ID ADDRESS MARK NOT FOUND IN TIME
- 170 - DATA MARK NOT FOUND IN TIME
- 200 - DATA CRC ERROR
- 210 - PARITY ERRORS

3.4 PROGRAM HUNG

IF THERE IS NO RESPONSE FROM THE RX11 WHILE WAITING FOR THE TRANSFER REQUEST (TR) FLAG OR THE DONE FLAG, THE PROGRAM WILL TYPE "DEVICE TEST HUNG @ PC" (ONLY IF SW13 IS OFF) AND THEN GO ON TO THE NEXT TEST, OR THE BEGINNING OF THE PRESENT TEST.

4.0 HALTS

THE ONLY HALTS IN THE PROGRAM ARE THE SELECTABLE HALTS (EOP,EOT, AT ERROR), THE ILLEGAL VECTOR HALTS, AND THE ILLEGAL TEST SELECTION HALT.

NOTE: ONE ADDITIONAL 'HALT' EXISTS IN THE PROGRAM, IT OCCURS WHEN THE USER HAS LOADED HIS PROGRAM VIA THE 'RXDP' MONITOR (ON UNIT 0) AND ALSO REQUIRES TESTING OF UNIT 0, A PROMPT MESSAGE IS TYPED REMINDING THE USER TO REPLACE HIS LOAD MEDIUM WITH A SCRATCH DISKETTE BEFORE GOING ON, THE PROGRAM WILL WAIT FOR THE 'CONTINUE' SWITCH TO BE DEPRESSED.

5.0 FLOW CHARTS

84	BASIC DEFINITIONS
249	TEST SELECTION VIA SWITCH REGISTER
269	OPERATIONAL SWITCH REGISTER POSITIONS
295	RXCS (RX COMMAND STATUS REGISTER)
346	RXDB (RX DATA BUFFER REGISTER)
398	START AND RESTART ADDRESSES
420	GET VALUE FOR SOFTWARE SWITCH REGISTER
533	PRETEST - INITIALIZE [KEY] PART I
904	TEST 1 - RXCS TEST PART I / INTERRUPT TEST PART I
1066	TEST 2 - INTERRUPT TEST PART II / VECTOR ADDRESS VERIFICATION
1304	TEST 3 - INTERRUPT TEST PART III / PRIORITY LEVEL VERIFICATION PART I
1362	TEST 4 - INTERRUPT TEST PART IV / PRIORITY VERIFICATION PART II
1422	TEST 5 - INIT [PROGRAMMED] / RST
1608	TEST 6 - FILL BUFFER TRANSFER LENGTH VERIFICATION
1709	TEST 10 - EMPTY BUFFER XFER LENGTH AND CONTENT VERIFICATION PART II
1717	TEST 7 - EMPTY BUFFER XFER LENGTH AND CONTENT VERIFICATION PART I
1798	TEST 12 - FILL/EMPTY BUFFER ALL 1'S
1805	TEST 11 - FILL/EMPTY BUFFER ALL 0'S
1817	TEST 13 DRIVE READY VERIFICATION
1897	TEST 14 - ERROR FLAG AND B-CODE VERIFICATION PART I
2032	TEST 15 - ERROR FLAG AND B-CODE VERIFICATION PART II
2085	TEST 16 - ERROR FLAG AND B-CODE VERIFICATION PART III
2168	TEST 17 - ILLEGAL TRACK ERROR VERIFICATION
2273	TEST 20 - SEEK VERIFICATION VIA READ FUNCTION
2311	TEST 21 - WRITE TEST
2380	TEST 22 - INITIALIZE IMPLIED READ
2402	TEST 23 - READ TEST
2417	TEST 24 - DATA TRANSFER AND VERIFICATION
2431	TEST 25 - DATA TRANSFER AND VERIFICATION VIA DELETED DATA MODE
2439	TEST 26 - HEAD "HOME" TEST
2507	" ERROR " TRAP SERVICE ROUTINE
2582	" SCOPE " TRAP SERVICE ROUTINE
2699	DRIVE TEST SELECTION
2746	WRITE FUNCTION
2885	READ DATA FROM THE DISKETTE
3016	READ AND VERIFY DATA
3156	INTERRUPT SERVICE
3265	PATTERN GENERATOR
3408	UNIT SELECTION
3451	TRACK SEQUENCE SELECTION
3543	SECTOR SELECTION
3576	TYPE ROUTINE
3664	BINARY TO OCTAL (ASCII) AND TYPE
3741	SAVE AND RESTORE R0-R5 ROUTINES
3786	TTY INPUT ROUTINE
3933	TRAP DECODER
3949	TRAP TABLE
3970	POWER DOWN AND UP ROUTINES
4015	SINGLE LENGTH BINARY TO DECIMAL ASCII ROUTINE
4033	DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
4125	MESSAGES


```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
```

```

,NLIST CND,MD,MC
,LIST ME
,ENABL ABS,AMA
,MCALL ,HEADER,,EQUAT,,SETUP,,STYPE,,STYPOCT,,STRAP,TYPOCS
,MCALL SETPRI,,SPOWER,STARS,,SSB2D,,SDB2D,,SSAVE,COMMENT
,MCALL ENDCOMMENT,,SREAD,GETSWR

;TITLE MAINDEC-11-DZRXB-E
;*COPYRIGHT (C) MARCH 21,1976
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY D. ADAMS/B. BURGESS
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZGAC-CN),MAR 21, 1976.
;*
STN=1
SSWR=160000 ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT

;COPYRIGHT (C) 1975,1976
;THIS SOFTWARE IS FURNISHED UNDER LICENCE FOR USE ONLY
;ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
;THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE, THIS
;SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED
;OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON
;EXCEPT FOR USE ON SUCH SYSTEM, AND TO ONE WHO AGREES TO
;THESE LICENCE TERMS, TITLE TO OWNERSHIP OF THE
;SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

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

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

000001
160000

```
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
```

```

;MODIFIED TO REV. D BY B. BURGESS NOV. 10, 1975 AS FOLLOWS:
;
;A) ADDED CAPABILITY OF VARIABLE DEVICE 'BR' LEVEL. ALL RELEVANT TESTS
;CALCULATE 'CPU' LEVEL BASED ON CURRENT CONTENTS OF LOCATION 'BRLEV:'.
;DEFAULT 'BR' LEVEL, FOR THE DEVICE, SET BY THE PROGRAM IS 5. ANY OTHER
;'BR' LEVEL ( E.G. 6 ) WOULD HAVE TO BE PATCHED INTO LOCATION 'BRLEV:'
;BEFORE RUNNING THE PROGRAM.
;
;B) ADDED TWO (2) ROUTINES TO HANDLE 'UNEXPECTED' BUS TIMEOUT AND
;RESERVED INSTRUCTION TRAPS ( TRAPS TO VECTORS 4 & 10, RESPECTIVELY).
;BOTH ROUTINES WILL INDICATE WHICH TRAP OCCURRED, THE 'PC' LOCATION
;OF WHERE THE TRAP OCCURRED, AND ATTEMPT TO RESTART THE PROGRAM.
;
;C) ADDED CODE TO FAILSAFE UNIT 0 UNDERGOING TESTING IF PROGRAM WAS
;LOADED VIA UNIT 0 USING 'RXDP' MONITOR AND USER STARTED RUNNING
;THE PROGRAM WITHOUT HAVING REPLACED HIS LOAD MEDIUM WITH A 'SCRATCH'
;DISKETTE.
;
;D) ADDED MESSAGES TO INDICATE TO USER WHEN HE HAS SELECTED TRACK AND/OR
;SECTOR LIMITS 'OUT OF RANGE' AND CORRESPONDING DEFAULT LIMITS WHEN
;THIS CONDITION ARISES
;
;E) MODIFIED TESTS 1 THRU 4 TO CORRECTLY PRINT OUT THE CONTENTS OF
;'KRXVEC' ( LOCATION HOLDING THE DEVICE VECTOR) AS 264 INSTEAD OF 270.
;
;F) MODIFIED TEST 2 TO HANDLE A 'LOCKED IN INTERRUPT STATE' CONDITION
;ARISING WHEN 'INTERRUPT ENABLE' AND 'DONE' ARE BOTH QUALIFIED AND
;THE 'REQUEST INTERRUPT' FLOP NEVER GETS CLEARED.
;
;G) ADDED EXTENSIVE MAINTENANCE INFORMATION BASED ON FAULT INSERTION
;RESULTS. INFORMATION IS KEYED TO THE 'ERROR' REPORT WITHIN A
;TEST. INFORMATION PROVIDED SHOULD BE SELF-EXPLANATORY BUT SHOULD
;NOT BE MISCONSTRUED AS BEING ALL ENCOMPASSING DUE TO HUMAN ERRORS
;IN STATISTICS GATHERING, INABILITY TO FAULT INSERT SOME CHIPS, AS
;WELL AS ONLY TWO (2) MODULES ABLE TO BE FAULT INSERTED I.E. -
;M7846 (UNIBUS INTERFACE) AND M7727 (READ/WRITE CONTROL).
;
;H) ADDED FLOW CHARTS
;
```



```

83
84      ,SBTTL BASIC DEFINITIONS
85
86      ;*INITIAL ADDRESS OF THE STACK POINTER *** 1200 ***
87      001200 STACK= 1200
88      ,EQUIV EMT,ERROR      ;BASIC DEFINITION OF ERROR CALL
89      ,EQUIV IOT,SCOPE      ;BASIC DEFINITION OF SCOPE CALL
90
91      ;*MISCELLANEOUS DEFINITIONS
92      000011 HT= 11      ;CODE FOR HORIZONTAL TAB
93      000012 LF= 12      ;CODE FOR LINE FEED
94      000015 CR= 15      ;CODE FOR CARRIAGE RETURN
95      000200 CRLF= 200    ;CODE FOR CARRIAGE RETURN-LINE FEED
96      177776 PS= 177776  ;PROCESSOR STATUS WORD
97      ,EQUIV PS,PSW
98      177774 STK LMT= 177774 ;STACK LIMIT REGISTER
99      177772 PIRQ= 177772 ;PROGRAM INTERRUPT REQUEST REGISTER
100     177570 DSWH= 177570 ;HARDWARE SWITCH REGISTER
101     177570 DDISP= 177570 ;HARDWARE DISPLAY REGISTER
102
103     ;*GENERAL PURPOSE REGISTER DEFINITIONS
104     000000 R0= %0      ;GENERAL REGISTER
105     000001 R1= %1      ;GENERAL REGISTER
106     000002 R2= %2      ;GENERAL REGISTER
107     000003 R3= %3      ;GENERAL REGISTER
108     000004 R4= %4      ;GENERAL REGISTER
109     000005 R5= %5      ;GENERAL REGISTER
110     000006 R6= %6      ;GENERAL REGISTER
111     000007 R7= %7      ;GENERAL REGISTER
112     ,EQUIV R6,SP      ;STACK POINTER
113     ,EQUIV R7,PC      ;PROGRAM COUNTER
114
115     ;*PRIORITY LEVEL DEFINITIONS
116     000000 PR0= 0      ;PRIORITY LEVEL 0
117     000040 PR1= 40     ;PRIORITY LEVEL 1
118     000100 PR2= 100    ;PRIORITY LEVEL 2
119     000140 PR3= 140    ;PRIORITY LEVEL 3
120     000200 PR4= 200    ;PRIORITY LEVEL 4
121     000240 PR5= 240    ;PRIORITY LEVEL 5
122     000300 PR6= 300    ;PRIORITY LEVEL 6
123     000340 PR7= 340    ;PRIORITY LEVEL 7
124
125     ;*"SWITCH REGISTER" SWITCH DEFINITIONS
126     100000 SW15= 100000
127     040000 SW14= 400000
128     200000 SW13= 200000
129     010000 SW12= 100000
130     004000 SW11= 400000
131     002000 SW10= 200000
132     001000 SW09= 100000
133     000400 SW08= 400000
134     000200 SW07= 200000
135     000100 SW06= 100000
136     000040 SW05= 400000
137     000020 SW04= 200000
138     000010 SW03= 100000

```

```

139     000004 SW02= 4
140     000002 SW01= 2
141     000001 SW00= 1
142     ,EQUIV SW09,SW9
143     ,EQUIV SW08,SW8
144     ,EQUIV SW07,SW7
145     ,EQUIV SW06,SW6
146     ,EQUIV SW05,SW5
147     ,EQUIV SW04,SW4
148     ,EQUIV SW03,SW3
149     ,EQUIV SW02,SW2
150     ,EQUIV SW01,SW1
151     ,EQUIV SW00,SW0
152
153     ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
154     100000 BIT15= 100000
155     040000 BIT14= 400000
156     200000 BIT13= 200000
157     010000 BIT12= 100000
158     004000 BIT11= 400000
159     002000 BIT10= 200000
160     001000 BIT09= 100000
161     000400 BIT08= 400000
162     000200 BIT07= 200000
163     000100 BIT06= 100000
164     000040 BIT05= 400000
165     000020 BIT04= 200000
166     000010 BIT03= 100000
167     000004 BIT02= 400000
168     000002 BIT01= 200000
169     000001 BIT00= 100000
170     ,EQUIV BIT09,BIT9
171     ,EQUIV BIT08,BIT8
172     ,EQUIV BIT07,BIT7
173     ,EQUIV BIT06,BIT6
174     ,EQUIV BIT05,BIT5
175     ,EQUIV BIT04,BIT4
176     ,EQUIV BIT03,BIT3
177     ,EQUIV BIT02,BIT2
178     ,EQUIV BIT01,BIT1
179     ,EQUIV BIT00,BIT0
180
181     ;*BASIC "CPU" TRAP VECTOR ADDRESSES
182     000004 ERRVEC= 4      ;TIME OUT AND OTHER ERRORS
183     000010 RESVEC= 10     ;RESERVED AND ILLEGAL INSTRUCTIONS
184     000014 TRITVEC= 14    ;"T" BIT
185     000014 TRTVEC= 14     ;TRACE TRAP
186     000014 BPTVEC= 14     ;BREAKPOINT TRAP (BPT)
187     000020 IOTVEC= 20     ;INPUT/OUTPUT TRAP (IOT) **SCOPE**
188     000024 PWRVEC= 24     ;POWER FAIL
189     000030 EMTVEC= 30     ;EMULATOR TRAP (EMT) **ERROR**
190     000034 TRAPVEC= 34    ;"TRAP" TRAP
191     000060 TKVEC= 60      ;TTY KEYBOARD VECTOR
192     000064 TPVEC= 64      ;TTY PRINTER VECTOR
193     000240 PIRQVEC= 240   ;PROGRAM INTERRUPT REQUEST VECTOR
194

```



```

195
196
197 ;SPECIAL EQUATES
198
199
200 000017 RDER =17 ; READ B CODE
201 000040 DONEBIT =40
202 000101 FBIE =101 ; IE+FILL BUFFER
203 000103 EBLE =103 ; IE+EMPTY BUFFER
204 000105 WRTE =105 ; IE+WRITE SECTOR
205 000107 RDIE =107 ; IE+READ SECTOR
206 000115 WTDIE =115 ; IE+WRITE DD SECTOR
207 040001 RECAL =40001
208 000000 OPEN =0
209
210 000000 ,=0
211 000000 000000 000000 ,WORD 0,0
212
213 000004 ,=4
214 000004 000130 ,WORD BUSEPR ;UNEXPECTED TIMEOUT TRAP PC
215 000006 000340 ,WORD PR7 ;UNEXPECTED TIMEOUT TRAP PS
216 000010 000154 ,WORD RESFRR ;UNEXPECTED RESERVED INSTRUCTION TRAP PC
217 000012 000340 ,WORD PR7 ;UNEXPECTED RESERVED INSTRUCTION TRAP PS
218
219 000020 ,=20
220 000020 000476 XSCOPE
221 000022 000340 PR7
222 000024 015116 SPWRDN
223 000026 000340 PR7
224 000030 000232 XEPROR
225 000032 000340 PR7
226 000034 015046 STHAP ;ADDRESS OF TRAP SERVICE
227 000036 000340 PR7
228
229 000046 ,=46
230 000046 002452 LOGICAL ;ACT 11 EOP HOOKS
231
232
233 000052 ,=52
234 000052 000000 ,WORD 0
235
236 000174 ,=174
237 000174 000000 DISPREG: 0
238 000176 000000 SWREG: 0
239
240
241 000200 ,=200
242 000200 000401 BR 16
243 000202 000402 BR 26
244 000204 000137 001232 1S: JMP SA200 ;OPERATOR SELECTED CONDITIONS
245 000210 000137 001222 2S: JMP SA202 ;RESTART PROGRAM WITH PREVIOUS PARAMETERS
246
247
248

```

```

249 ;SBTTL TEST SELECTION VIA SWITCH REGISTER
250
251 ;*****
252 ;*****
253 ;*****
254
255 ;SET TEST AND DRIVE SELECTION IN " DTEST " LOCATION 1212
256 ;
257 ; BIT 15 = 1 - UNIT 1 SELECTED
258 ; BIT 14 = 1 - UNIT 0 SELECTED
259 ; BIT 15 & BIT 14 = 0 - BOTH DRIVES MUST BE READY
260 ;
261 ; BIT 4 - BIT 0 = OCTAL NUMBER OF DESIRED STARTING TEST
262 ; BIT 4 - BIT 0 = 0 -ALL TESTS WILL BE SEQUENCED THROUGH
263
264 ;*****
265 ;*****
266 ;*****
267
268 ;SBTTL OPERATIONAL SWITCH REGISTER POSITIONS
269
270 ;*****
271 ;*****
272 ;*****
273
274 ; SFT OPERATING CONDITIONS IN THE SWITCH REGISTER (HARDWARE)
275 ; OR SOFTWARE SWITCH REGISTER LOCATION 176
276
277
278 ; 15 = 1 - HALT ON ERROR
279 ; 14 = 1 - HALT AT END OF PASS
280 ; 13 = 1 - INHIBIT ERROR TYPEOUT
281 ; 12 = 1 - LOOP ON TEST
282 ; 11 = 1 - LOCK ON ERROR
283 ; 10 = 1 - HALT AT END OF TEST
284 ; 9 = 1 - PRINT ALL DATA ERRORS
285 ; 9 = 0 - PRINT ONLY FIRST 10 DATA ERRORS PER SECTOR
286 ; 8 = 1 - INHIBIT RECALIBRATION ON SEEK ERRORS,
287 ;
288 ; 0 = 1 - INHIBIT <BELL> AT ERROR
289
290 ; 15-0 = 1 - SELECT SOFTWARE SWITCH REGISTER
291
292 ;*****
293 ;*****
294 ;*****

```



```

295          ,SBTTL RXCS (RX COMMAND STATUS REGISTER)
296
297          ,=STACK
298
299 001200 000000      OD: 0          ;OD/ID = 0 UNLESS SPECIFIC TRACKS SELECTED,
300          001201      ID=OD+1
301 001202 015001      FIRST: 015001      ; FIRST = 1, LAST = 32
302          001203      LAST=FIRST+1
303
304 001204 000264      KRXVEC: 264
305
306 001206 177170      RXCS: 177170
307
308          ; RXCS: STANDARD DEVICE ADDRESS = 177170
309
310          ; TOGGLE INTO PROGRAM LOCATION " RXCS " THE RX11 DEVICE ADDRESS IF NOT = 177170
311
312          ;KEY: R = READ ONLY BIT
313          ;      W = WRITE ONLY BIT
314
315          ;          15 = R = EPROH
316          ;          14 = W = INITIALIZE
317          ;          13 =
318          ;          12 =
319          ;          11 = (BITS 13-8)
320          ;          10 = (NOT USED)
321          ;          9 =
322          ;          8 =
323          ;          7 = R = TRANSFER REQUEST
324          ;          6 = R/W= INTERRUPT ENABLE
325          ;          5 = R = DONE
326          ;          4 = W = UNIT SELECT
327          ;          3 = W = FUNCTION
328          ;          2 = W = FUNCTION
329          ;          1 = W = FUNCTION
330          ;          0 = W = GO !
331
332          ; FUNCTION
333
334          ; 3 2 1 0
335
336          ; - - - GO
337
338          ; 0 + GO = FILL BUFFER
339          ; 2 + GO = EMPTY BUFFER
340          ; 4 + GO = WRITE SECTOR
341          ; 6 + GO = READ SECTOR
342          ;
343          ; 12 + GO = READ STATUS " A "
344          ; 14 + GO = WRITE DELETED DATA
345          ; 16 + GO = READ STATUS " B " (CODES)

```

```

346          ,SBTTL RXDB (RX DATA BUFFER REGISTER)
347
348 001210 177172      RXDB: 177172
349
350          ; RXDB: STANDARD DEVICE ADDRESS = 177172
351
352          ; THE FOLLOWING BIT IDENTIFICATION REPRESENTS THE STATUS AT THE END OF A FUNCTION
353          ; (BUT NOT FUNCTION # 16 TO READ STATUS " B ") DISPLAYED WITHIN THE RX-DATA BUFFER.
354
355          ;          (A)      7 = SELECTED DRIVE READY
356          ;          6 = DELETED DATA
357          ;          5 =
358          ;          4 =
359          ;          3 = WRITE PROTECT ERROR
360          ;          (B)      2 = INITIALIZE DONE
361          ;          1 = PARITY ERROR
362          ;          0 = CRC ERROR
363
364          ; (A) = VISIBLE ONLY IF THE FUNCTION WAS # 12 READ STATUS " A "
365          ; (B) = INIT DONE VISIBLE IF AN INITIALIZAE [KEY] OR [PROGRAMMED] WAS ISSUED
366
367          DTESTP: 0
368 001212 000000
369
370 001214 000005      BRLEV: 5
371
372          ;BRLEV: STANDARD PRIORITY INTERRUPT LEVEL = 5
373
374          ;TOGGLE INTO PROGRAM LOCATION "BRLEV" THE RX11 INTERRUPT PRIORITY
375          ;LEVEL IF NOT = 5
376
377 001216 177570      SWR:          ,WORD DSWR
378 001220 177570      DISPLAY:      ,WORD DDISP
379
380          ; R0 = GOOD          /EXPECTED RESULT OF TEST
381          ; R1 = EAC          /ACTUAL RESULT OF TEST
382          ; R2 = BLANK          /
383          ; R3 = TEST Q
384
385          ;*****
386
387          ;WORD "UNITSEL" HAS THE FOLLOWING BIT DEFFINITIONS
388          ;
389          ;BIT15 = 1 = UNIT 1 SELECTED FOR USE
390          ;BIT14 = 1 = UNIT 1 IN USE
391          ;BIT8 = 1 = THIS PASS HAD AN ERROR
392          ;BIT7 = 1 = UNIT 0 SELECTED FOR USE
393          ;BIT6 = 1 = UNIT 0 IN USE
394          ;BIT4 = 1 = UNIT SELECTION BIT
395
396          ;*****
397

```



```

398                                     ,SBTTL START AND RESTART ADDRESSES
399
400                                     ; THE STARTING ADDRESS WAS 202
401
402 001222 005200 SA202: INC R0
403 001224 012706 MOV #STACK,SP
404 001230 000447 BR RESTART
405
406                                     ; THE STARTING ADDRESS WAS 200
407
408 001232 005000 SA200: CLR R0
409 001234 012737 MOV #177570,SWR ;RESET TO HARDWARE SWR,
410 001242 012706 MOV #STACK,SP
411 001246 104400 TYPE ,MREV ;PRINT THE NAME AND REVISION
412 001252 013746 MOV 4,-(SP) ;SAVE 'BUSERR' TIMEOUT 'PC'
413 001256 012737 MOV #18,4 ;SET UP TIMEOUT VECTOR
414 001264 022777 CMP #177777,ASWR ;IS SOFTWARE SWR SELECTED
415 001272 001402 BEQ 2S ;YES, INSERT IT'S ADDRESS
416 001274 000423 BR 3S ;BR IF NO TIMEOUT TRAP OCCURS
417 001276 022626 1S: CMP (SP)+,(SP)+ ;RESTORE THE STACK
418 001300 012737 000176 001216 2S: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REGISTER
419 001306 012737 000174 001220 MOV #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REG,
420
421 .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
422 001314 005737 000042 TST #42 ;ARE WE RUNNING UNDER XXDP/ACT?
423 001320 001006 BNE 64S ;BRANCH IF YES
424 001322 023727 001216 000176 CMP SWR,#SWREG ;SOFTWARE SWITCH REG SELECTED?
425 001332 104404 BNE 65S ;BRANCH IF NO
426 001334 000403 GTSWR ;GET SOFT-SWR SETTINGS
427 001336 112737 000001 015044 BR 65S ;SET AUTO-MODE INDICATOR
428 001344
429 001344 012637 000004 65S: MOV (SP)+,4 ;RESET TIMEOUT VECTOR TO 'BUSERR'
430 001350 000005 RESTART: RESET ;INITIALIZE THE RX11 SYSTEM
431 001352 012746 MOV #PK7,-(SP)
432 001356 012746 MOV #4S,-(SP)
433 001362 000002 RTI ;LOAD THE PSW
434 001364 013737 001206 001210 4S: MOV RXCS, RXDB ;GET ADDRESS OF RXCS
435 001372 062737 000002 001210 ADD #2, RXDB ;SET UP ADDRESS OF RXDB
436 001400 012737 001234 012626 MOV #001234,RAN1 ; INITIALIZE CONSTANTS OF
437 001406 012737 000765 012630 MOV #000765,RAN2 ; RANDOM NUMBER GENERATOR
438 001414 005037 002526 CLR CCOUNT
439 001420 005037 002530 CLR PASS
440 001424 005037 006756 CLR HANGER
441 001430 012737 177740 MOV #177740,HANGPL
442 001436 005700 TST R0
443 001440 001055 RNE XSA202 ; STARTING ADDRESS WAS 202
444 001442 005037 CLR UNITSEL
445 001446 032737 140000 001212 BIT #140000,DTESTP ;WERE ANY DRIVES SELECTED
446 001454 001004 RNE 1S ;YES GO SET THEIR BITS
447 001456 052737 100200 012724 BIS #100200,UNITSEL ;NO, BOTH UNITS MUST BE READY
448 001464 000415 BR 2S
449 001466 032737 040000 001212 1S: BIT #BIT14,DTESTP ;WAS UNIT 0 SELECTED
450 001474 001434 BFO 3S ;NO, MUST BE UNIT 1
451 001476 052737 000200 012724 RIS #200,UNITSEL ;YES,SET SELECTED BIT
452 001504 005737 001212 TST DTESTP ;WAS UNIT 1 SELECTED
453 001510 100003 BPL 2S ;NO

```

```

454 001512 052737 100000 012724 BIS #BIT15,UNITSEL ;YES,SET THE SELECTED BIT
455 001520 123727 000041 000010 2S: CMPB 41,#10 ;WAS PROGRAM LOADED IN DUMP MODE
456
457 001526 001022 RNE XSA202 ;VIA XXDP?
458 001530 005737 000042 TST #42 ;BRANCH IF NOT
459 001534 001410 BEQ 5S ;CHECK FOR PXDP OPERATION
460 001536 042737 000200 012724 BIC #200,UNITSEL ;IN CHAIN MODE, DESELECT UNIT 0
461 001544 104400 016173 TYPE ,MUNIT1
462 001550 104400 016203 TYPE ,MONLY
463 001554 000407 BR XSA202
464 001556 104400 017226 5S: TYPE, D0LOAD ;AND DO NOT HALT
465 ;INFORM USER TO REMOVE LOAD MEDIUM
466 ;FROM UNIT 0 AND REPLACE WITH
467 ;A 'SCRATCH' DISKETTE IF HE
468 ;WISHES TO TEST UNIT 0
469 ;WAIT FOR USER RESPONSE
470 001562 000000 HALT
471 001564 000403 RP XSA202
472 001566 052737 100000 012724 3S: BIS #BIT15,UNITSEL ;IS SELECTED BIT
473 001574 042737 100200 001200 XSA202: RIC #100200,OD ;CLEAR 1ST TIME BITS FOR BOTH DRIVES
474 001602 104400 015614 TYPE, MDTESTP
475 001606 013746 001212 MOV DTESTP,-(SP) ;SAVE DTESTP FOR TYPEOUT
476 001612 104402 TYPOS ;GO TYPE--OCTAL ASCII
477 001614 0006 .BYTE 6 ;TYPE 6 DIGIT(S)
478 001616 0000 .BYTE 0 ;SUPPRESS LEADING ZEROS
479 001618 104400 016120 TYPE ,MCRLF
480 001622 005737 001200 LIMITS: TST OD
481 001626 001005 BNE TRKLMT
482 001630 005037 013136 CLR SEQUEN
483 001634 104400 016417 TYPE ,MLIMTRK
484 001640 000432 BR SECLMT
485
486 ; 0 <= OD <= ID <= 114
487 001642 123727 001201 000114 TRKLMT: CMPB ID,#114
488 001650 101021 BHI 1S
489 001652 123737 001200 001201 CMPB OD,ID
490 001660 101015 BHI 1S
491 001662 104400 016445 TYPE ,MOD
492 001664 113746 001200 MOVB OD,-(SP)
493 001672 104402 TYPOS
494 001674 0003 .BYTE 3
495 001676 0000 .BYTE 0
496 001678 104400 016451 TYPE ,MID
497 001702 113746 001201 MOVB ID,-(SP)
498 001706 104402 TYPOS
499 001710 0003 .BYTE 3
500 001712 000405 .BYTE 0
501 001714 104400 017053 1S: BR SECLMT
502 ;TYPE MSG. INDICATING ID OR OD
503 ;TOO BIG & DEFAULTING TO TRACKS
504 ;0, 52, 53, 114
505 001720 005037 001200 CLR OD
506 001724 000736 BR LIMITS
507
508 ; 1 <= FIRST <= LAST <= 32
509 001726 105737 001202 SECLMT: TSTB FIRST

```



```
;TYPE MSG. INDICATING THAT
;SECTOR RANGE SELECTED WAS
;INVALID AND DEFAULTING TO A
; 1ST SECTOR VALUE OF 1 AND
;A LAST SECTOR VALUE OF 32
```

[illegible]

1
M7846 (UNIBUS INTERFACE)

```

; NOTE: MAKE SURE THE DRIVES ARE CONNECTED CORRECTLY,THE
; DISKETTES INSERTED,AND THE DOORS OF THE SELECTED DRIVES
; ARE CLOSED. IF THE THESE CONDITIONS ARE NOT SET THERE
; WILL BE AN ERROR AT THIS POINT.

```



```
;IF THE FAULT CANNOT BE FOUND ON THE UNIBUS INTERFACE MODULE
;AND/OR THE FAULT IS NOT INHERENT TO THE UNIBUS INTERFACE MODULE
;M7846 THERE IS A POSSIBILITY OF ITS EXISTENCE ON THE READ/WRITE
;MODULE M7727.
```

IM7727 (READ/WRITE CONTROL)

SIGNAL NAME	REASON	POSSIBLE CHIPS
-----	PIN 15 NOT AT GROUND	E15
SEL TRK 0	STUCK LOW	E15
DK0 TRK 0	STUCK LOW	E15
SEL DK0	STUCK LOW	E13
DC LO	STUCK LOW	E13
OUT	STUCK HIGH	E13
-----	E18 CLOCK LOCKED HIGH/LOW	E14, E18
-----	REPLACE E18	-----
INIT	STUCK LOW	E18, E16
-----	PIN 14 NOT +5V THRU 1K	E16

```

) ----- ) *J,K' INPUTS TO E18 LOCKED      E17
            HIGH/LOW
) ----- ) *J' INPUT TO E16 LOCKED HIGH     E17

```

```

653                                     ; ////////////////////////////////////////////
654                                     ; STATUS A [RXDB] AT "DONE"
655                                     ;      7      6      -      3      2      1      0
656                                     ;
657                                     ;      SEL      WRITE INIT PAR      CRC
658                                     ;      DRIVE DD      PROTECT [DONE]
659                                     ;      RDY      (N/A)
660                                     ;
661                                     ; ////////////////////////////////////////////
662
663
664
665
666 002110 012737 177740 0067b0 15: MOV #177740,HANGPL ;RESET HANG COUNTER
667 002116 012700 000204 MOV #204,R0 ;EXPECT INIT DONE AND UNIT 0 RDY
668 002122 017702 177062 MOV 0RXDB,R2
669 002126 010201 MOV R2,R1
670 002130 042701 000100 BIC #BIT6,R1 ;CLEAR DELETED DATA BIT
671 002134 105737 012724 TSTB UNITSEL ;WAS UNIT 0 SELECTED
672 002140 100404 RMI 2$
673 002142 042701 000200 BIC #BIT7,R1 ;UNIT 0 WAS NOT SELECTED
674 002146 042700 000200 BIC #BIT7,R0 ;CLEAR THE DRIVE 0 RDY BITS
675 002152 020100 28: CMP R1, R0
676 002154 001544 BEQ REBEGIN
677
678 ; (R0) = 4 IF UNIT 0 IS NOT SELECTED, OR 204 IF UNIT 0 IS SELECTED
679 ; (R1) = ACTUAL RXDB MINUS DELETED DATA BIT#6
680 ; (R2) = ACTUAL RXDB
681
682 002156 104000 ERROR ; RXDB NOT = 4, OR 204
683
684

```

[illegible]

```

;THE FOLLOWING IS A PRESENTATION OF POSSIBLE REASONS AS TO WHY
;THIS ERROR REPORT WAS GENERATED.  THE INFORMATION SHOWN IS
;BASED ON FAULT INSERTION RESULTS, AND SHOULD PROVIDE LOGICAL
;AREAS TO CHECK FOR THE RELEVANT FAULT/S,
;
;IF THIS ERROR REPORT WAS THE 1ST GIVEN IN A SERIES OF ERRORS
;ANALYZE THE FOLLOWING AREA/S:

```

1M7846 (UNIBUS INTERFACE)

SIGNAL NAME	REASON	POSSIBLE CHIPS
1		
1		

3

```

;B SHIFT          STUCK LOW/HIGH          E15,E19,E2
;B DONE          STUCK HIGH              E15
;TO RX01         STUCK LOW/HIGH          E15
;LOAD            STUCK HIGH              E17,E18
;SELECT 00       STUCK HIGH              E17,E18
;SELECT 02       STUCK LOW               E17,E18,E34
;IN              STUCK LOW               E17,E21
;CMD             STUCK LOW/HIGH          E21
;BUS D15         STUCK HIGH              E40
;RX INIT        STUCK HIGH              E32
;BUS D00 -> D03  STUCK HIGH              E41,E7
;DATA           STUCK LOW               E18,E11
;-----
;-----          INCORRECT SHIFT OUTPUTS
;-----          CAN'T SELECT 'B' INPUTS
;-----          CAN'T RESET
;LB              STUCK LOW               E34,E8
;RX BUSY        STUCK LOW               E34,E22,E19
;B DONE         STUCK LOW               E19
;BUS D05        STUCK LOW               E4
;B SER DATA    STUCK LOW/HIGH          E9
;INT ENB(1) H   STUCK HIGH              E37
;

```

```

; IF THE FAULT CANNOT BE FOUND ON THE UNIBUS INTERFACE MODULE
; AND/OR THE FAULT IS NOT INHERENT TO THE UNIBUS INTERFACE MODULE
; M7846 THERE IS A POSSIBILITY OF ITS EXISTENCE ON THE READ/WRITE
; MODULE M7727.

```

NOTE: ONLY APPROX. 30% OF THIS MODULE LENT ITSELF CONDUCTIVE TO THE FAULT INSERTION PROCESS; ERGO, THE RESOLUTION FOR FAULT ANALYSIS OBTAINABLE BY THIS MODULE IS NOT VERY HIGH. HOWEVER, ANALYSIS OF THE FOLLOWING AREA/S, IF THIS ERROR REPORT WAS THE 1ST GIVEN IN A SERIES OF ERRORS, SHOULD AT LEAST PLACE YOU WITHIN THE RELEVANT AREA ON THE MODULE.

IM7727 (READ/WRITE CONTROL)

SIGNAL NAME	REASON	POSSIBLE CHIPS
SEL WT PROT	STUCK LOW	E4,E6,E15

1 / 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 10

```

754      002160  000542      BP      REBEGIN
755      002162  004737      002556  MORETESTS: JSR PC, LOCKUP
756      002166  005723      TANOHERE: IST (P3)+, ADJUST P3 FOR NEXT TEST ADDRESS

```

```

757      002170 016337 002214 006556 FIRSTTEST: MOV TESTS(R3), PCSCOPE      ; EQUIVALENT TO " SCOPE "
758      002176 013737 006556 002532      MOV PCSCOPE,FAST      ;SAVE THE FIRST ADDRESS OF THE TEST
759      002204 012706 001200      MOV #STACK,SP
760      002210 000173 002214      JMP #TESTS(R3)
761      002214 002166 002620 002740 TESTS:  T0N0THERE, T1, T2, T3, T4, T5, T6, T7
762      002222 003262 003420 003564
763      002230 004010 004142
764      002234 004140 004226 004216      T10, T11, T12, T13, T14, T15, T16, T17
765      002242 004272 004416 004610
766      002250 004750 005216
767      002254 005462 005506 005672      T20, T21, T22, T23, T24, T25, T26, N0M0RETESTS
768      002262 005750 006016 006050
769      002270 006062 002274
770
771      ; TEST 1 - RXCS TEST PART I / INTERRUPT TEST PART I
772
773      ; TEST 2 - INTERRUPT TEST PART II / VECTOR ADDRESS VERIFICATION
774
775      ; * TEST 3 - INTERRUPT TEST PART III / PRIORITY LEVEL VERIFICATION PART I
776
777      ; * TEST 4 - INTEPRUPT TEST PART IV / PRIORITY VERIFICATION PART II
778
779      ; TEST 5 - INIT [PROGRAMMED] / RST
780
781      ; TEST 6 - FILL BUFFER TRANSFER LENGTH VERIFICATION
782
783      ; TEST 7 - EMPTY BUFFER TRANSFER LENGTH AND CONTENT VERIFICATION PART I
784
785      ; TEST 10 - EMPTY BUFFER TRANSFER LENGTH AND CONTENT VERIFICATION PART II
786
787      ; TEST 11 - FILL/EMPTY BUFFER ALL 0'S
788
789      ; TEST 12 - FILL/EMPTY BUFFER ALL 1'S
790
791      ; TEST 13 - DRIVE READY VERIFICATION FOR SELECTED DRIVES
792
793      ; TEST 14 - ERROR FLAG AND B - CODE VERIFICATION PART I
794
795      ; TEST 15 - ERROR FLAG AND B - CODE VERIFICATION PART II
796      ; /DELETED DATA BIT SETS
797
798      ;TEST 16 - ERROR FLAG AND B - CODE VERIFICATION PART III
799      ; /DELETED DATA BIT CLEARS
800
801      ;TEST 17 - ILLEGAL TRACK ERROR AND B - CODE VERIFICATION
802
803      ;TEST 20 - SEEK VERIFICATION VIA READ FUNCTION
804
805      ;TEST 21 - WRITE TEST
806
807      ;TEST 22 - INITIALIZE IMPLIED READ
808
809      ;TEST 23 - READ TEST
810
811      ;TEST 24 - DATA TRANSFER & VERIFICATION
812

```



```

813 ;TEST 25 - DATA TRANSFER & VERIFICATION VIA DELETED DATA MODE
814
815 ;TEST 26 - HEAD "HOME" TEST
816
817 ;THERE ARE NO MORE TESTS
818
819 ; * NOTE: ON PROCESSORS WITHOUT HARDWARE PROCESSOR STATUS WORDS (PSW)
820 ; THESE TEST WILL NOT BE RUN.

```

```

821 ;PRINT AN END OF PASS INDICATOR
822
823 ; C - RX11/RX01 TEST PASS OK
824 ; D - RX11/RX01 AND DRIVE TESTING OK
825 ; - - AN ERROR OCCURRED (DURING C OR D)
826
827 ; NOTE: IF BIT 0 OF UNITSEL IS A 1
828 ; THEN AN ERROR HAS OCCURRED FOR THIS PASS
829
830 002274 042777 000100 176704 NOMORETESTS: BIC #BIT6,0RXCS ;CLEAR 'IE' BIT BEFORE NEXT PASS
831 002302 005037 006756 CLR HANGER
832 002306 032737 000400 012724 BIT #BIT0,UNITSEL ;"C" OR "D" MEANS ERRORLESS PASS.
833 002314 001403 BEQ 1$
834 002316 012737 000055 002534 MOV #"-",MX ; " - " MEANS UN-ERRORLESS PASS
835 002324 005737 002526 1$: TST CCOUNT
836 002330 001002 BNE 3$
837 002332 104400 016120 TYPE, MCRLF
838 002336 005237 002526 3$: INC CCOUNT
839 002342 022737 000110 002526 CMP #72,,CCOUNT
840 002350 001002 BNE 4$
841 002352 005037 002526 CLR CCOUNT
842 002356 104400 002534 4$: TYPE, MX
843 002362 104400 006470 TYPE, MABELL
844 002366 005237 002530 2$: INC PASS
845 002372 102775 HVS 2$
846 002374 104405 CKSWR
847 002376 032777 040000 176612 BIT #SW14,0SWR ; AC SW 14 = 1 TO HALT AT END OF PASS
848 002404 001413 BEQ 6$
849 002406 104400 016120 TYPE, MCRLF
850 002412 104400 006725 TYPE, MPASS
851 002416 013737 002530 002430 MOV PASS,5$
852 002424 004537 015600 JSR R5,SGLDEC
853 002430 000000 5$: OPEN
854 002432 000000 HALT
855 002434 005237 006756 6$: INC HANGER ;WAIT FOR EOP INDICATOR TO BE PRINTED
856 002440 001375 BNE 6$
857 002442 013705 000042 MOV #42,R5 ;ACT 11 END OF PASS HOOKS
858 002446 001405 BEQ HERE
859 002450 000005 RESET
860 002452 004715 LOGICAL: JSR PC,(R5)
861 002454 000240 NOP
862 002456 000240 NOP
863 002460 000240 NOP
864 002462 000137 002466 HERE: JMP REBEGIN
865
866 002466 042737 000400 012724 REBEGIN: BIC #BIT0,UNITSEL ;CLEAR HARD ERROR INDICATOR
867 002474 013703 001212 MOV DTESTP, R3
868 002500 042703 177740 BIC #177740, R3 ; R3 CONTAINS TEST # 0 TO 26
869 002504 020327 000027 CMP R3,#27
870 002510 103002 BHS 1$
871 002512 006303 ASL R3
872 002514 000625 BR FIRSTTEST
873
874 002516 104400 002536 1$: TYPE, M1LST
875 002522 000137 001232 JMP SA200
876

```



```

877 002526 000000          CCOUNT: 0
878 002530 000000          PASS: 0
879 002532 000000          FAST: 0
880
881 002534 000103          MX:      ,ASCIZ "C"
882
883 002536 046111 042514 040507 MILTST: ,ASCIZ "ILLEGAL TEST"<15><12>
884 002544 020114 042524 052123
885 002552 005015      000
886
887      002556          ,EVEN
888

```

```

889                                     ; DATA SW 10 = 1 TO HALT AT END OF TEST
890
891 002556 104405          LOCKUP: CKSWR
892 002560 032777 002000 176430      RII #SW10,0SWR
893 002566 001401          REQ 15
894 002570 000000          HALT
895
896                                     ; DATA SW 12 = 1 TO LOCK SCOPE LOOP ON TEST OK OR NOT
897
898 002572 032777 010000 176416 15:   BIT #SW12,0SWR          ;15 LOOP ON TEST SWITCH SET
899 002600 001403          BEQ 25          ;IF NOT SET GO ON TO NEXT TEST
900 002602 062716 000002          ADD #2,#SP          ;IF SET RETURN TO FIRSTTEST
901 002606 000207          RTS PC
902 002610 042737 040100 012724 25:   BIC #40100,UNITSEL      ;CLEAR UNIT USED BITS
903 002616 000207          RTS PC

```


MAINDEC-11-DZRXB-E MACY11 27(732) 16-MAR-76 15:55 PAGE 22
DZRXBE.P11 TEST 1 - RXCS TEST PART 1 / INTERRUPT TEST PART 1

SEQ 0047

[illegible]

[illegible]

IF THIS ERROR REPORT WAS THE 1ST GIVEN IN A SERIES OF ERRORS
ANALYZE THE FOLLOWING AREA/S:

[illegible]

```

1064 002732 000004          SS:  SCOPE
1065 002734 000137 004246      JMP CEXIT      ;END OF TEST 1

```

[illegible]

IF THIS ERROR REPORT WAS THE 1ST GIVEN IN A SERIES OF ERRORS
ANALYZE THE FOLLOWING AREA/S:

SIGNAL NAME	REASON	POSSIBLE CHIPS
-----	SACK FLOP LOCKED CLEAR	E13,E31,E36
BBSY FLOP CLOCK LOCKED HIGH		E13,E31,E36,E33
BUS REQUEST	STUCK HIGH	E31
BUS INTR	STUCK HIGH	E31,E35,E8,E38
-----	SACK FLOP CLOCK LOCKED HIGH	E31,E21,E9,E12
-----	SACK FLOP CLOCK LOCKED LOW	E31,E21
BBSY FLOP CLOCK LOCKED LOW		E31,E33,E9
BUS D02/D04/D05	STUCK HIGH	E38
BUS SACK	STUCK HIGH	E32
BG OUT	STUCK HIGH	E32,E25,E28
BUS REQUEST	STUCK LOW	E39
-----	JUMPER (N1) NOT IN	-----
BUS D07	STUCK HIGH	E35,E8
-----	GRANT FLOP CAN'T BE PRESET	E25,E9,E28
B SER DATA	STUCK LOW	E9
-----	GRANT FLOP 'Q' OUTPUT	E28
	STUCK HIGH	

[illegible]

[illegible][illegible]

SEQ 4051

$\frac{1}{\sqrt{2}} \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$

```

1228      003064  104412          25:      SUBSCOPE
1229
1230
1231          , TEST 2 = CONT'D
1232
1233          , THE PURPOSE OF THIS TEST IS TO VERIFY THAT BIT 6 OF THE RXCS (INTERRUPT ENABLE)

```



```

1234 ; CAN BE CLEARED AFTER IT WAS KNOWN TO BE SET
1235
1236 003066 013702 001204 MOV KRXVEC, R2
1237 003072 010246 MOV R2, -(SP) ;SAVE INTERRUPT VECTOR FOR
1238 ;ERROR REPORT
1239 003074 012722 003164 MOV #48, (R2)+ ; RX11 VECTOR ADDRESS
1240 003100 012722 000340 MOV #PR7, (R2)+
1241 003104 012602 MOV (SP)+, R2 ;RESTORE INTERRUPT VECTOR FOR
1242 ;ERROR REPORT
1243 003106 042777 000100 176072 BIC #BIT6, @ RXCS ; CLEAR THE RX11 INTERRUPT ENABLE BIT
1244
1245 ; THE RXCS SHOULD = 40 (DONE)
1246
1247 003114 012700 000040 MOV #40, R0
1248 003120 020077 176062 CMP R0, @ RXCS
1249 003124 001403 BEQ 35
1250 003126 017701 176054 MOV @ RXCS, R1
1251
1252 ; (R0) = 40 ; (P1) = ACTUAL RXCS ; (R2) = N/A
1253
1254 003132 104000 ERROR ; RXCS NOT = 40
1255 003134 104412 35: SUBSCOPE
1256 ; NO RX11 INTERRUPTS SHOULD OCCUR [YET]
1257
1258 003136 005046 CLP -(SP) ; PDP PRIORITY <ON>
1259 003140 012746 003146 MOV #105, -(SP)
1260 003144 000002 RTI
1261 003146 000240 105: NOP
1262 003150 000240 NOP
1263 003152 012746 000340 MOV #PR7, -(SP) ; PDP PRIORITY <OFF> 7
1264 003156 012746 003166 MOV #115, -(SP)
1265 003162 000002 RTI
1266
1267 ; RETURN TO HERE WITH THE PDP PRIORITY = 7 IF AN UNEXPECTED RX11 INTERRUPT
1268 ; WHILE CLEARING THE RX11 INTERRUPT ENABLE BIT 6
1269
1270 ; (R0) = N/A ; (R1) = N/A ; (R2) = N/A
1271
1272 003164 104000 45: ERROR ; UNEXPECTED RX11 INTERRUPT
1273 003166 104412 115: SUBSCOPE
1274
1275 ; AN RX11 INTERRUPT SHOULD OCCUR [NOW]
1276
1277 003170 013702 001204 MOV KRXVEC, R2
1278 003174 010246 MOV R2, -(SP) ;SAVE INTERRUPT VECTOR FOR
1279 ;ERROR REPORT
1280 003176 012722 003246 MOV #56, (R2)+ ; RX11 VECTOR ADDRESS
1281 003202 012722 000340 MOV #PR7, (R2)+
1282 003206 012602 MOV (SP)+, R2 ;RESTORE INTERRUPT VECTOR FOR
1283 ;ERROR REPORT
1284 003210 005046 CLP -(SP) ; PDP PRIORITY <ON>
1285 003212 012746 003220 MOV #125, -(SP)
1286 003216 000002 RTI
1287 003220 052777 000100 175760 125: BIS #BIT6, @ RXCS ; SET RX11 INTERRUPT ENABLE BIT
1288 003226 000240 NOP
1289 003230 000240 NOP

```

```

1290 003232 012746 000340 MOV #PR7, -(SP)
1291 003236 012746 003244 MOV #135, -(SP)
1292 003242 000002 RTI
1293
1294 ; (R0) = N/A ; (R1) = N/A ; (R2) = N/A
1295
1296 003244 104000 135: ERROR ; NO RX11 INTERRUPT OCCURRED
1297
1298 ; RETURN TO HERE WITH THE PDP PRIORITY = 7 IF AN RX01 INTERRUPT
1299
1300 003246 000004 55: SCOPE
1301 003250 042777 000100 175730 BIC #BIT6, @ RXCS ; CLEAR THE RX11 INTERRUPT ENABLE
1302
1303 003256 000137 004246 JMP CEXIT ;END OF TEST 2

```


MAINDEC-11-DZRXB-E MACY11 27(732) 16-MAR-76 15:55 PAGE 30
DZRXB-E.P11 TEST 3 - INTERRUPT TEST PART III / PRIORITY LEVEL VERIFICATION PART I

SEQ 0055

```

1360 003414 000137 004246 4S: JMP CEXIT ;END OF TEST 3
1361
1362 ,SHLTL TEST 4 - INTERRUPT TEST PART IV / PRIORITY VERIFICATION PART II
1363
1364 ; THE PURPOSE OF THIS TEST IS TO VERIFY THE PRIORITY OF THE RX11 INTERRUPT REQUEST LINE
1365 ; THE PROGRAM SETS THE PDP PRIORITY = THE DEVICE LEVEL, (NORMALLY 5 OR THE CONTENTS OF L
1366 ; NO RX01 INTERRUPTS SHOULD OCCUR
1367 ; IF AN INTERRUPT DOES OCCUR THEN THE PRIORITY LEVEL OF THE RX11 IS [NOT]
1368 ; = THE NORMAL DEVICE LEVEL OF 5, OR WHATEVER IS THE VALUE IN LOCATION "BRLEV:"
1369 ; BUT MAYBE SOME VALUE GREATER THAN THE CONTENTS OF LOC. "BRLEV:"
1370 ; NOTE: IF THERE IS NO HARDWARE "PSW" THIS TEST WILL BE SKIPPED.
1371
1372 003420 005401 T4: CLR R1 ;INDICATOR TO CPU PRIORITY ROUTINE
1373 ;TO DROP CPU PRIORITY 1 LEVEL
1374 ;LESS THAN THE DEVICE LEVEL
1375 003422 013702 001204 MOV RXXVEC, R2
1376 003426 010246 MOV R2, -(SP) ;SAVE INTERRUPT VECTOR FOR
1377 ;ERROR REPORT
1378 003430 012722 003546 MOV #16, (R2)+ ; RX01 VECTOR ADDRESS
1379 003434 012722 000340 MOV #PR7, (R2)+
1380 003440 012602 MOV (SP)+, R2 ;RESTORE INTERRUPT VECTOR FOR
1381 ;ERROR REPORT
1382 003442 052701 000200 BIS #BIT7, R1 ;SET INDICATOR TO CPU PRIORITY
1383 ;ROUTINE TO SET CPU PRIORITY LEVEL
1384 ;TO THE SAME LEVEL AS THE DEVICE
1385 003446 013746 000004 MOV 4, -(SP) ;SAVE "BUSERR" TIMEOUT "PC"
1386 003452 012737 003470 000004 MOV #35, 4 ;SET TIMEOUT VECTOR
1387 003460 012737 000240 177776 MOV #PRS, PSW ;SET LEVEL TO 5 IF "PSW" EXISTS
1388 003466 000404 BR 4S ;GO ON TO RESET VECTOR & DO TEST
1389 003470 022626 3S: CMP (SP)+, (SP)+ ;CORRECT STACK FROM BUS TIMEOUT
1390 003472 012637 000004 MOV (SP)+, 4 ;RESTORE TIMEOUT VECTOR TO "BUSERR"
1391 003476 000430 BR 5S ;NO HARDWARE PSW - SKIP THIS TEST
1392 003500 012637 000004 4S: MOV (SP)+, 4 ;RESET TIMEOUT VECTOR TO BUSERR
1393 003504 004737 006200 JSR PC, CPUPRI ;CALCULATE CPU PRIORITY LEVEL TO
1394 ;BE THE SAME AS THE DEVICE LEVEL
1395 ;I.E. - SAME AS CONTENTS OF LOC.
1396 ;"BRLEV"
1397 003510 010046 MOV R0, -(SP) ;;PUT NEW PS ON STACK
1398 003512 012746 003520 MOV #64S, -(SP) ;;PUT NEW PC ON STACK
1399 003516 000002 RTI ;;POP NEW PC AND PS
1400 003520
1401 003520 052777 000100 175460 64S: BIS #BIT6, @RXCS ;SET RX01 INTERRUPT ENABLE
1402 003526 000240 NOP
1403 003530 000240 NOP
1404 003532 013746 000340 MOV PR7, -(SP) ;;PUT NEW PS ON STACK
1405 003536 012746 003544 MOV #65S, -(SP) ;;PUT NEW PC ON STACK
1406 003542 000002 RTI ;;POP NEW PC AND PS
1407 003544
1408 003544 000401 65S: BR 2S
1409
1410 ; RETURN TO HERE WITH THE PDP PRIORITY = 7 IF AN RX01 INTERRUPT
1411
1412 ; (R0) = N/A ; (R1) = N/A ; (R2) = N/A
1413
1414 003546 104000 1S: ERROR ;PRIORITY LEVEL NOT = TO CONTENTS
1415 ;OF LOCATION "BRLEV:" (NORMALLY 5)

```



```

1416                                     ;BUT MAYBE SOME VALUE GREATER THAN
1417                                     ;THE CONTENTS SPECIFIED BY LOC.
1418                                     ;'BRLEV;'
1419 003550 000004                26:  SCOPE
1420 003552 042777 000100 175426 58:  BIC #BIT6, @ RXCS
1421 003560 000137 004246                JMP CEXIT
                                     ; CLEAR THE RX01 INTERRUPT ENABLE
                                     ;END OF TEST 4

```

MAINDEC-11-DZRXB-E MACY11 27(732) 16-MAR-76 15:55 PAGE 32
DZRXBE.P11 TEST 5 - INIT [PROGRAMMED] / RST

SEQ 0057

```

1422                                ,SBTTL TEST 5 - INIT [PROGRAMMED] / RST
1423
1424                                ; THE PURPOSE OF THIS TEST IS TO VERIFY THAT SETTING THE RXCS BIT 14
1425                                ; CAUSES AN RX01 PROGRAMMED SUBSYSTEM INITIALIZE
1426
1427                                ; THE RXCS SHOULD = 40 (DONE)
1428
1429                                ; THE RXDB SHOULD = 4, OR 104, OR 204, OR 304
1430
1431    003564    052777    040000    175414    T5:      BIS #BIT14, @ RXCS                ; RX01 PROGRAMMED INITIALIZE
1432    003572    004737    006574                1S:      JSR PC, SDN                    ; WAIT FOR THE DONE BIT
1433                                BR 1S
1434    003600    012700    000040                MOV #40, R0
1435    003604    017701    175376                MOV @ RXCS, R1                ; PROGRAM EXPECTS RXCS = 40 (DONE)
1436    003610    020100                CMP R1, R0                ; ACTUAL RXCS
1437    003612    001401                BEQ 2S
1438
1439                                ; (R0) = 40 ; (R1) = ACTUAL RXCS ; (R2) = N/A
1440
1441    003614    104000                ERROR
1442                                ; RXCS NOT = 40

```

[illegible]

```

;IF THE FAULT CANNOT BE FOUND ON THE UNIBUS INTERFACE MODULE
;AND/OR THE FAULT IS NOT INHERENT TO THE UNIBUS INTERFACE MODULE
;M7846 THERE IS A POSSIBILITY OF ITS EXISTENCE ON THE READ/WRITE
;MODULE M7727.

```

NOTE: ONLY APPROX. 30% OF THIS MODULE WENT ITSELF CONDUCTIVE TO THE FAULT INSERTION PROCESS; ERGO, THE RESOLUTION FOR FAULT ANALYSIS OBTAINABLE BY THIS MODULE IS NOT VERY HIGH, HOWEVER, ANALYSIS OF THE FOLLOWING AREA/S, IF THIS ERROR REPORT WAS THE 1ST GIVEN IN A SERIES OF ERRORS, SHOULD AT LEAST PLACE YOU WITHIN THE RELEVANT AREA ON THE MODULE.

1
1M7727 (READ/WRITE CONTROL)

SIGNAL NAME	REASON	POSSIBLE CHIPS
'J' INPUT LOCKED LOW		E16

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

```

1470 003616 104412          28:      SUBSCOPE
1471                                     , ///////////////////////////////////////////////////
1472                                     ,
1473                                     ,
1474                                     ,      7      6      -      -      3      2      1      0
1475                                     ,
1476                                     ,      SEL      WRITE      INIT      PAR
1477                                     ,      DRIVE DD      PROTECT [DONE]      CRC

```


[illegible][illegible]

1532 003662 104412 49: SUBSCOPE

[illegible]

```

;M7846 (UNIBUS INTERFACE)
;
;          SIGNAL NAME              REASON                POSSIBLE CHIPS
; -----
;          PARITY FLOP CLOCK LOCKED HIGH      E2
;          PARITY FLOP "Q" OUTPUT             E2
;          LOCKED HIGH                        E2
;RX DATA                                     STUCK LOW      E6
;
```

1585 003740 104412 1061 SUBSCOPE

```

1586
1587 ; THE PXOB SHOULD = 200 (IF DRIVE 0 IS READY), OR 0 IF UNIT 0 IS NOT SELECTED
1588 ; MAYBE 300 (IF DRIVE 0 IS READY AND SECTOR 1 WAS WRITTEN WITH DELETED DATA)

```


MAINDEC-11-DZRXB-E MACY11 27(732) 16-MAR-76 15:55 PAGE 36
DZRXBE,P11 TEST 6 - FILL BUFFER TRANSFER LENGTH VERIFICATION

SEW 0061

[illegible][illegible]

```

;THE FOLLOWING IS A PRESENTATION OF POSSIBLE REASONS AS TO WHY
;THIS ERROR REPORT WAS GENERATED.  THE INFORMATION SHOWN IS
;BASED ON FAULT INSERTION RESULTS, AND SHOULD PROVIDE LOGICAL
;AREAS TO CHECK FOR THE RELEVANT FAULT/S.
;
;IF THIS ERROR REPORT WAS THE 1ST GIVEN IN A SERIES OF ERRORS
;ANALYZE THE FOLLOWING AREA/S:

```

```

;
;M7846 (UNIBUS INTERFACE)
;

```


[illegible]

```

; IF THIS ERROR REPORT WAS THE 1ST GIVEN IN A SERIES OF ERRORS
; ANALYZE THE FOLLOWING AREA/S:

```

[illegible]

SEQ 0063

[illegible]

```

; IF THIS ERROR REPORT WAS THE 1ST GIVEN IN A SERIES OF ERRORS
; ANALYZE THE FOLLOWING AREA/S:

```

SIGNAL NAME	REASON	POSSIBLE CHIPS
JRX RUN	STUCK LOW	E24


```

AINDEC-11-DZRXB-E          MACY11 7(732) 16-MAR-76 15:55 PAGE 40
ZRXBE,P11          TEST 11  - FILL/EMPTY BUFFER ALL 0'S

1811 004246 012737 000103 002534 CEXIT:  MOV #*C,MX          ;EOP INDICATOR FOR END OF CONTROL TESTS
1812 004254 000137 002162          JMP MORETESTS
1813
1814 004260 012737 000104 002534 DEXIT:  MOV #*D,MX          ;EOP INDICATOR FOR END OF DRIVE TESTS
1815 004266 000137 002162          JMP MORETESTS
1816
1817          ,SBTTL TEST 13 DRIVE READY VERIFICATION
1818
1819          ;THIS TEST VERIFIES THAT DRIVE READY WILL SET FOR ALL SELECTED DRIVES.
1820
1821          ;THIS SECTION OF TEST FOR UNIT 0 DRY NO A READ STATUS A FUNCTION
1822
1823 004272 005737 012724          T13:    TSTB UNITSEL          ;WAS UNIT 0 SELECTED
1824 004276 100020          BPL 25          ;NO, GO TEST UNIT 1
1825 004300 012777 000013 174700 MOV #13,BRXCS          ;READ STATUS A UNIT 0
1826 004306 004737 006574          15:    JSR PC,SDN          ;WAIT FOR DONE FLAG
1827 004312 000775          BR 16
1828 004314 012700 000200          MOV #200,R0          ;EXPECT DRIVE READY TO BE SET
1829 004320 017702 174664          MOV 0RXDB,R2          ;ACTUAL RXDB
1830 004324 010201          MOV R2,R1
1831 004326 042701 000100          BIC #BIT0,R1          ;CLEAR DD BIT IF SET
1832 004332 020001          CMP R0,R1          ;WAS " DRY " SET AND INITDONE CLEARED
1833 004334 001401          BEQ 25
1834
1835          ;R0 = 200 ; P1 = RXDB MINUS DD BIT ;R2 = ACTUAL RXDB
1836
1837 004336 104000          EPROR
1838 004340 104412          25:    SUBSCOPE
1839
1840          ;TEST FOR UNIT 1 DRIVE READY ON A READ STATUS A FUNCTION
1841
1842 004342 005737 012724          TST UNITSEL          ;WAS UNIT 1 SELECTED
1843 004346 100020          BPL 35          ;NO, GO TO END OF TEST
1844 004350 012777 000033 174630 MOV #33,BRXCS          ;READ STATUS A FOR DRIVE 1
1845 004356 004737 006574          48:    JSR PC,SDN          ;WAIT FOR DONE FLAG
1846 004362 000775          BR 48
1847 004364 012700 000200          MOV #200,R0          ;EXPECT " DRY " TO BE SET
1848 004370 017702 174614          MOV 0RXDB,R2
1849 004374 010201          MOV R2,R1
1850 004376 042701 000100          BIC #BIT0,R1          ;CLEAR DD BIT IF ANY
1851 004402 020100          CMP R1,R0          ;IS DRY SET AND INITDONE CLEARED
1852 004404 001401          BEQ 35
1853
1854          ;R0 = 200 ; R1 = RXDB MINUS DD BIT ; P2 = ACTUAL RXDB
1855
1856 004406 104000          EPROR
1857

```

```

; IF THE FAULT CANNOT BE FOUND ON THE UNIBUS INTERFACE MODULE
; AND/OR THE FAULT IS NOT INHERENT TO THE UNIBUS INTERFACE MODULE
; M7846 THERE IS A POSSIBILITY OF ITS EXISTENCE ON THE READ/WRITE
; MODULE M7727.

```


M7727 (READ/WRITE CONTROL)

SIGNAL NAME	REASON	POSSIBLE CHIPS

```

; NOTE: MAKE SURE THE DRIVES ARE CONNECTED CORRECTLY,THE
; DISKETTES INSERTED,AND THE DOORS OF THE SELECTED DRIVES
; ARE CLOSED, IF THE THESE CONDITIONS ARE NOT SET THERE
; WILL BE AN ERROR AT THIS POINT.

```

```

;SEL DK1          STUCK LOW          E13,E14
;DK1 INDX         STUCK HIGH/LOW     E15
;SEL INDX         STUCK HIGH/LOW     E15
;-----         'A2' INPUT STUCK LOW E15

```

[illegible]

```

1895 004410 000004          3S:  SCOPE
1896 004412 000137 004260      JMP DEXIT          ;END OF TEST 13

```

SEQ 0067

```

1897                                .SRTTL TEST 14 - ERROR FLAG AND B-CODE VERIFICATION PART 1
1898
1899                                ;THE PURPOSE OF THIS TEST IS TO VERIFY THAT TRYING TO READ A NON-EXISTANT
1900                                ;SECTOR WILL CAUSE AN ERROR, AND THE CORRECT ERROR CODE WILL BE PUT
1901                                ;INTO THE R4DB WHEN THE B STATUS IS READ.
1902
1903                                ;THIS SECTION INSUPES THAT ONLY 2 TR FLAGS WERE REQUIRED TO TAKE THE
1904                                ;DISKETTE ADDRESS, AND THAT AN ERROR DOES EXIST.
1905
1906                                004416    005002                                T14:      CLR R2
1907                                004420    005000                                CLR R0
1908                                004422    105737    012724                                TSTR UNITSEL                                ;IS UNIT 0 SELECTED
1909                                004426    100004                                RPL 10S
1910                                004430    012777    000007    174550                                MOV #7,0RXCS                                ;SET READ SECTOR FUNCTION AND GO
1911                                004436    000003                                BR 11S
1912                                004440    012777    000027    174540    10S:      MOV #27,0RXCS                                ;SEND READ FUNCTION TO UNIT 1
1913                                004446    004737    005106                                11S:      JSR PC,ILLADR                                ;SEND THE BAD SECTOR ADDRESS
1914                                004452    000006                                BR 1S                                           ;PREMATURE ERROR CONDITION
1915                                004454    012700    100040                                MOV #100040,R0                                ;EXPECT ERROR AND DONE BITS
1916                                004460    017701    174522                                MOV 0RXCS,R1                                ;SAVE THE RXCS
1917                                004464    020001                                CMP R0,R1
1918                                004466    001401                                BEQ 2S
1919
1920                                ;R0 = 100040 ; R1 = ACTUAL RXCS ; R2 = # OF TR FLAGS
1921
1922                                004470    104000                                1S:      ERROR
1923

```

[illegible]

```

;THE FOLLOWING IS A PRESENTATION OF POSSIBLE REASONS AS TO WHY
;THIS ERROR REPORT WAS GENERATED.  THE INFORMATION SHOWN IS
;BASED ON FAULT INSERTION RESULTS, AND SHOULD PROVIDE LOGICAL
;AREAS TO CHECK FOR THE RELEVANT FAULT/S.

```

IF THIS ERROR REPORT WAS THE 1ST GIVEN IN A SERIES OF ERRORS
ANALYZE THE FOLLOWING AREA/S:

IM7846 (UNIBUS INTERFACE)

	SIGNAL NAME	REASON	POSSIBLE CHIPS
1			
2			
3	BUS D15	STUCK HIGH	E14,E9,E40,E24
4	FX RUN	STUCK LOW	E41

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

1948 004472 104412
1949
1950
1951
1952

28: SUBSCOPE

```

;T14 CONT. - THIS SECTION TESTS THAT NO PARITY OR CRC ERROR OCCURRED
;ON PREVIOUS FUNCTION.

```


SEQ 2069

[illegible]


```

2032      ,SBTTL TEST 15 - ERROR FLAG AND B-CODE VERIFICATION PART II
2033
2034      ;THIS TEST VERIFIES THAT TRYING TO WRITE,USING DELETED DATA MODE, ON A
2035      ;NON-EXISTANT SECTOR WILL PRODUCE AN ERROR AND THE CORRECT B-CODE IS PRODUCED
2036      ;THIS SECTION SENDS OUT AN ILLEGAL SECTOR ADDRESS AND EXPECTS AN ERROR
2037      ; NOTE TEST 14 MUST BE RUN BEFORE THIS TEST
2038
2039      004610 005000      T15:   CLR R0
2040      004612 005002      CLR R2
2041      004614 105737      TSTB UNITSEL      ;WAS UNIT 0 SELECTED
2042      004620 100004      BPL 10S
2043      004622 012777      MOV #15,0RXCS      ;SET WRITE DELETED DATA FUNCTION
2044      004630 000403      BR 11S
2045      004632 012777      MOV #35,0RXCS      ;SEND WTR DD FUNCTION TO UNIT 1
2046      004640 004737      JSR PC,ILLADR      ;SEND THE ILLEGAL SECTOR ADDRESS
2047      004644 000406      BR 1S              ;PREMATURE ERROR CONDITION
2048      004646 012700      MOV #100040,R0      ;EXPECT ERROR AND DONE FLAGS
2049      004652 017701      MOV 0RXCS,R1
2050      004656 020001      CMP R0,R1
2051      004660 001401      BEQ 2S
2052
2053      ;R0 = 100040 ; R1 = ACTUAL RXCS ; R2 = # OF TR FLAGS
2054
2055      004662 104000      1S:   ERROR      ;RXCS NOT = 100040
2056      004664 104412      2S:   SUBSCOPE
2057
2058      ;T15 CONT. - THIS SECTION TESTS THAT THERE IS NO PARITY, CRC ERROR
2059      ;AND THAT THE DELETED DATA BIT IS SET.
2060
2061      004666 005002      CLR R2
2062      004670 012700      MOV #100,R0      ;EXPECT DELETED DATA BIT TO BE SET
2063      004674 017701      MOV 0RXDB,R1
2064      004700 020001      CMP R0,R1
2065      004702 001401      BEQ 3S
2066
2067      ; R0 = 100 ; R1 = ACTUAL RXDB ; R2 = N/A
2068      004704 104000      ERROR      ;DELETED DATA NOT SET OR OTHER ERRORS
2069      004706 104412      3S:   SUBSCOPE
2070
2071      ;T15 CONT. - THIS SECTION TESTS FOR THE B-CODE FOR ILLEGAL SECTOR.
2072
2073      004710 012777      000017 174270      MOV #17,0RXCS      ;SET READ STATUS B FUNCTION
2074      004716 004737      000574      JSR PC,SDN      ;WAIT FOR DONE FLAG
2075      004722 000775      BR 4S
2076      004724 012700      000070      MOV #70,R0
2077      004730 017701      174254      MOV 0RXDB,R1
2078      004734 020001      CMP R0,R1
2079      004736 001401      BEQ 5S
2080
2081      ; R0 = 70 ; R1 = ACTUAL B-CODE ; R2 = N/A
2082      004740 104000      ERROR      ;RXDB NOT = 70
2083      004742 000004      5S:   SCOPE
2084      004744 000137 004260      JMP DEXIT      ;END OF TEST 15
  
```

```

2085      ,SBTTL TEST 16 - ERROR FLAG AND B-CODE VERIFICATION PART III
2086
2087      ;THIS TEST VERIFIES THAT A WRITE FUNCTION TO A NON-EXISTANT SECTOR WILL
2088      ;PRODUCE AN ERROR AND A B-CODE OF 70. THE DELETED DATA BIT SHOULD ALSO BE CLEARED
2089      ;THIS SECTION TRANSFERS AN ILLEGAL SECTOR ADDRESS FOR A WRITE FUNCTION
2090      ; NOTE TEST 14 MUST BE RUN BEFORE THIS TEST
2091
2092      004750 005000      T16:   CLR R0
2093      004752 005002      CLR R2
2094      004754 105737      TSTB UNITSEL      ;WAS UNIT 0 SELECTED
2095      004760 100004      BPL 10S
2096      004762 012777      MOV #5,0RXCS      ;SET THE WRITE FUNCTION
2097      004770 000403      BR 11S
2098      004772 012777      MOV #25,0RXCS      ;SEND WRITE FUNCTION TO UNIT 1
2099      005000 004737      JSR PC,ILLADR      ;SEND THE ILLEGAL ADDRESS
2100      005004 000406      BR 1S              ;PREMATURE ERROR CONDITION
2101      005006 012700      MOV #100040,R0      ;EXPECT ERROR AND DONE BITS SET
2102      005012 017701      MOV 0RXCS,R1
2103      005016 020001      CMP R0,R1
2104      005020 001401      BEQ 2S
2105
2106      ; R0 = 100040 ; R1 = ACTUAL RXCS ; R2 = # OF TR FLAGS
2107
2108      005022 104000      1S:   ERROR
2109      005024 104412      2S:   SUBSCOPE
2110
2111      ;T16 CONT. - TESTS FOR NO PARITY, CRC ERRORS, AND NO DELETED DATA BIT
2112
2113      005026 005002      CLR R2
2114      005030 005000      CLR R0      ;NO BITS SHOULD BE SET IN THE RXDB
2115      005032 017701      MOV 0RXDB,R1
2116      005036 020001      CMP R0,R1
2117      005040 001401      BEQ 3S
2118
2119      ; R0 = 0 ; R1 = ACTUAL RXDB ; R2 = N/A
2120      005042 104000      ERROR      ;SOME BIT IS SET IN THE RXDB
2121      005044 104412      3S:   SUBSCOPE
2122
2123      ;T16 CONT. - TEST FOR CORRECT B-CODE FOR ILLEGAL SECTOR ADDRESS
2124
2125      005046 012777      000017 174132      MOV #17,0RXCS      ;SET READ STATUS B FUNCTION
2126      005054 004737      000574      JSR PC,SDN      ;WAIT FOR DONE FLAG
2127      005060 000775      BR 4S
2128      005062 012700      000070      MOV #70,R0
2129      005066 017701      174116      MOV 0RXDB,R1
2130      005072 020001      CMP R0,R1
2131      005074 001401      BEQ 5S              ;IS B-CODE = 70
2132
2133      ; R0 = 70 ; R1 = ACTUAL RXDB ; R2 = N/A
2134      005076 104000      ERROR
2135      005100 000004      5S:   SCOPE
2136      005102 000137 004260      JMP DEXIT      ;END OF TEST 16
  
```



```

2137
2138 ;GENERATE AN ILLEGAL SECTOR ADDRESS
2139
2140 005106 004737 006560 ILLADR: JSR PC,STR ;LOOK FOR A TR FLAG
2141 005112 000402 BR 26 ;NO TR FLAG, IS DONE SET
2142 005114 005202 INC R2 ;TR FLAG COUNTER
2143 005116 000404 BR 36
2144 005120 004737 006574 25: JSR PC,SDN ;LOOK FOR DONE FLAG
2145 005124 000770 BR ILLADR ;NOT DONE RECHECK TR FLAG
2146 005126 000430 BR 16 ;DONE IS SET TOO EARLY GO TO ERROR
2147 005130 005077 174054 36: CLR 0RXDB ;0 SECTOR ADDRESS (ILLEGAL)
2148 005134 004737 006560 78: JSR PC,STR ;LOOK FOR SECOND TR FLAG
2149 005140 000402 BR 56 ;NOT TR, IS IT DONE
2150 005142 005202 INC R2
2151 005144 000404 BR 66 ;TR FLAG SEND TRACK ADDRESS
2152 005146 004737 006574 56: JSR PC,SDN ;LOOK FOR DONE FLAG
2153 005152 000770 BR 76 ;NOT DONE, RECHECK TR FLAG
2154 005154 000415 BR 16 ;DONE TOO SOON GO TO ERROR
2155 005156 005077 174026 66: CLR 0RXDB ;SEND TRACK ADDRESS OF 0
2156 005162 004737 006560 116: JSR PC,STR ;ARE THERE ANY MORE TR FLAGS
2157 005166 000402 BR 106 ;NO, LOOK FOR DONE
2158 005170 005202 INC R2 ;YES
2159 005172 000406 BR 16 ;TOO MANY TR FLAGS OR MICROCONTROLLER
2160 ;DID NOT DETECT THE ERROR
2161 005174 004737 006574 106: JSR PC,SDN ;LOOK FOR DONE FLAG
2162 005200 000770 BR 116 ;NOT DONE RETEST TR FLAG
2163 005202 062716 000002 ADD #2,#SP
2164 005206 000207 46: RTS PC
2165 005210 017701 173772 16: MOV 0RXCS,R1
2166 005214 000774 BR 46

```

```

2167
2168 ;SBTTL TEST 17 - ILLEGAL TRACK ERROR VERIFICATION
2169
2170 ;THIS TEST VERIFIES THAT IF A TRACK ADDRESS LARGER THAN 114 (OCTAL) IS
2171 ;ACCESSED, AN ERROR CONDITION WILL EXIST, AND A B-CODE WILL = 40
2172
2173 005216 005002 T17: CLR R2
2174 005220 005000 CLR R0
2175 005222 012777 173756 MOV #7,0RXCS ;SET READ FUNCTION
2176 005230 004737 006560 36: JSR PC,STR ;LOOK FOR TR FLAG
2177 005234 000401 BR 16 ;NO TR FLAG CHECK DONE
2178 005236 000410 BR 26
2179 005240 004737 006574 16: JSR PC,SDN
2180 005244 000771 BR 36 ;DONE OCCURRED TOO SOON SET UP FOR ERROR
2181 005246 017701 173734 MOV 0RXCS,R1
2182 005252 017702 173732 MOV 0RXDB,R2
2183 005256 000433 BR 46
2184 005260 012777 173722 26: MOV #1,0RXDB ;SEND LEGAL SECTOR ADDRESS
2185 005266 004737 006560 56: JSR PC,STR ;LOOK FOR TR FLAG
2186 005272 000401 BR 66
2187 005274 000410 BR 76
2188 005276 004737 006574 66: JSR PC,SDN
2189 005302 000771 BR 56
2190 005304 017701 173676 MOV 0RXCS,R1 ;DONE SET TOO EARLY
2191 005310 017702 173674 MOV 0RXDB,R2
2192 005314 000414 BR 46
2193 005316 012777 000115 173664 76: MOV #115,0RXDB ;SEND ILLEGAL TRACK ADDRESS
2194 005324 004737 006574 106: JSR PC,SDN ;WAIT FOR DONE ON THE ERROR
2195 005330 000775 BR 106
2196 005332 012700 100040 MOV #100040,R0 ;EXPECT ERROR AND DONE SET
2197 005336 017701 173644 MOV 0RXCS,R1
2198 005342 020001 CMP R0,R1
2199 005344 001401 BEQ 116
2200
2201 ;TWO ERROR CONDITIONS TO REPORT
2202 ;IF R0 = 0 THEN R1 = RXCS ;R2 = RXDB ON A DONE TOO SOON ERROR
2203 ;IF R0 = 100040 THEN R1 = ACTUAL RXCS ; R2 = N/A
2204
2205 005346 104000 46: ERROR ;DONE SET TOO SOON OR NO ERROR OCCURRED
2206

```

```

;THE FOLLOWING IS A PRESENTATION OF POSSIBLE REASONS AS TO WHY
;THIS ERROR REPORT WAS GENERATED. THE INFORMATION SHOWN IS
;BASED ON FAULT INSERTION RESULTS, AND SHOULD PROVIDE LOGICAL
;AREAS TO CHECK FOR THE RELEVANT FAULTS.
;
;IF THIS ERROR REPORT WAS THE 1ST GIVEN IN A SERIES OF ERRORS
;ANALYZE THE FOLLOWING AREA/S:
;
;

```

17846 (UNIBUS INTERFACE)

SIGNAL NAME	REASON	POSSIBLE CHIPS
-------------	--------	----------------

MAINDEC-11-DZRXB-E MACY11 27(732) 16-MAR-76 15:55 PAGE 50
DZRXBE,P11 TEST 17 - ILLEGAL TRACK ERROR VERIFICATION

SEQ 0075

[illegible]


```

2328 005570 000771      BR 16
2329 005572 012737 000005 007702 28:  MOV #5,FUNCTION
2330 005600 004737 005610      JSR PC,T21X
2331 005604 000137 004260      JMP DEXT
;SET WRITE FUNCTION
;GO ISSUE THE COMMAND
;END OF TEST 21

```

```

2332 005610 004737 007634      T21X: JSR PC,COMMWORD
2333 005614 004737 006574      38:  JSR PC,SDN
2334 005620 000775      BR 38
2335 005622 005777 173360      TST @RXCS
2336 005626 100013      BPL 48
2337 005630 017701 173354      MOV @RXDB,R1
2338 005634 012777 000017 173344      MOV #17,@RXCS
2339 005642 004737 006574      58:  JSR PC,SDN
2340 005646 000775      BR 58
2341 005650 017702 173334      MOV @RXDB,R2
2342
2343      ;R0 = ADDR OF LAST DATA BYTE ; R1 = STATUS A ; R2 = STATUS B
2344
2345 005654 104000      ERROR
2346

```

```

;*****
;*****
;*****

```

IF THE FAULT CANNOT BE FOUND ON THE UNIBUS INTERFACE MODULE
AND/OR THE FAULT IS NOT INHERENT TO THE UNIBUS INTERFACE MODULE
M7746 THERE IS A POSSIBILITY OF ITS EXISTENCE ON THE READ/WRITE
MODULE M7727.

NOTE: ONLY APPROX. 30% OF THIS MODULE LENT ITSELF CONDUCTIVE TO
THE FAULT INSERTION PROCESS; ERGO, THE RESOLUTION FOR FAULT
ANALYSIS OBTAINABLE BY THIS MODULE IS NOT VERY HIGH,
HOWEVER, ANALYSIS OF THE FOLLOWING AREA/S, IF THIS ERROR
REPORT WAS THE 1ST GIVEN IN A SERIES OF ERRORS, SHOULD AT
LEAST PLACE YOU WITHIN THE RELEVANT AREA ON THE MODULE.

M7727 (READ/WRITE CONTROL)

SIGNAL NAME	REASON	POSSIBLE CHIPS
WT DATA	STUCK LOW/HIGH	E5
WT GATE	STUCK LOW	E13

```

;*****
;*****
;*****

```

```

2375 005656 104412      48:  SUBSCOPE
2376 005660 005000      CLR R0
2377 005662 005001      CLR R1
2378 005664 004737 004152      JSR PC,T7EMPTY
2379 005670 000207      RTS PC
;EMPTY BUFFER AND CHECK CONTENTS

```



```

2380 .SRTTL TEST 22 - INITIALIZE IMPLIED READ
2381 ;AFTER PREVIOUSLY WRITING A PATTERN ON SECTOR 1 TRACK 1, THIS TEST
2382 ;CHANGES THE CONTENTS OF THE SECTOR BUFFER AND DOES A PROGRAMMED INITIALIZE,
2383 ;AFTER WHICH THE SECTOR BUFFER MUST AGAIN CONTAIN THE DATA PREVIOUSLY
2384 ;WRITTEN ON THAT SECTOR AND TRACK.
2385 ;NOTE: THIS TEST WILL ONLY WORK IF UNIT 0 IS SELECTED AND ON LINE.
2386
2387
2388 005672 105737 012724 T22: TSTR UNITSEL ;IF UNIT 0 IS NOT SELECTED SKIP THIS TEST
2389 005676 100022 BPL ZS
2390 005700 005037 012324 CLR PAT
2391 005704 004737 004026 JSR PC,T6FILL ;LOAD THE SECTOR BUFFER WITH 0
2392 005710 005237 012324 INC PAT ;RELOAD CORE BUFFER WITH 1'S
2393 005714 004737 012260 JSR PC,GETPATTERN
2394 005720 004737 010766 JSR PC,ADJUSUM
2395 005724 052777 040001 173254 BIS #RECAL,#RXCS ;SET THE INIT. BIT
2396 005732 004737 006574 JSR PC,SDN
2397 005736 000775 BR 16
2398 005740 004737 004152 JSR PC,I7EMPTY ;EMPTY THE SECTOR BUFFER AND CHECK IT.
2399 005744 000137 004260 JMP DEXIT ;END OF TEST 22
2400
2401
2402 .SRTTL TEST 23 - READ TEST
2403
2404 ;THIS TEST VERIFIES THAT A READ FUNCTION DOES IN FACT LOAD THE SECTOR
2405 ;BUFFER WITH DATA READ FROM THE SELECTED ADDRESS.
2406
2407 005750 005037 012324 T23: CLR PAT
2408 005754 004737 004026 JSR PC,T6FILL ;LOAD SECTOR BUFFER WITH 0'S
2409 005760 005237 012324 INC PAT
2410 005764 004737 012634 JSR PC,GETUNIT
2411 005770 004737 012260 JSR PC,GETPATTERN ;RELOAD CORE BUFFER WITH 1'S
2412 005774 004737 010766 JSR PC,ADJUSUM ;SET UP FOR CHECK SUM
2413 006000 012737 000007 007702 MOV #7,FUNCTION ;SET READ FUNCTION AND GO
2414 006006 004737 005010 JSR PC,T2IX ;ISSUE COMMAND, WAIT FOR DONE, & TEST DATA
2415 006012 000137 004260 JMP DEXIT ;END OF TEST 23

```

```

2416
2417 .SRTTL TEST 24 - DATA TRANSFER AND VERIFICATION
2418
2419 ;THE PURPOSE OF THIS TEST IS TO WRITE THEN READ AND VERIFY DATA
2420 ;ON ALL SECTORS OF THE SELECTED TRACKS. THE TEST ALTERNATES BETWEEN
2421 ;DRIVES ON THE SELECTED TRACKS. DATA PATTERN IS A FLOATING 0.
2422
2423 006016 012737 000002 012324 T24: MOV #2,PAT ;SET DATA PATTERN TO FLOATING 0
2424 006024 013702 001204 T24X: MOV KRXVEC,R2 ;SET INTERRUPT ADDRESSES
2425 006030 012722 011526 MOV #INTSERV,(R2)+
2426 006034 012712 000340 MOV #PR7,(R2)
2427 006040 004737 007012 JSR PC,DRVSWP ;GO TRANSFER THE DATA
2428 006044 000137 004260 JMP DEXIT ;END OF TEST 24 OR 25
2429
2430
2431 .SRTTL TEST 25 - DATA TRANSFER AND VERIFICATION VIA DELETED DATA MODE
2432
2433 ;THIS TEST TRANSFERS DATA JUST LIKE TEST 24 EXCEPT IT USES THE
2434 ;DELETED DATA FORMAT AND A DATA PATTERN OF FLOATING 1.
2435
2436 006050 012737 000003 012324 T25: MOV #3,PAT ;SET DATA PATTERN TO FLOATING 1
2437 006056 000137 006024 JMP T24X ;GO TRANSFER THE DATA
2438
2439 .SRTTL TEST 26 - HEAD "HOME" TEST
2440
2441 ;THIS TEST MOVES THE HEAD OUT TO TRACK 12 (OCTAL) AND THEN WRITES/READ CHECKS
2442 ;ALL SECTORS (RANDOM DATA) ON EACH TRACK. THE TRACK SEQUENCE
2443 ;IS DECREMENTED BACK TO TRACK 0 (HOME). AFTER COMPLETING
2444 ;DRIVE 0 IT SWITCHES OVER TO DRIVE 1 DOING THE SAME TEST.
2445
2446
2447 006062 052737 000200 013136 T26: BIS #BIT7,SEQUEN ;SPECIAL DECREMENT SEQUENCE
2448 006070 012737 000007 012324 MOV #7,PAT ;SELECT RANDOM DATA
2449 006076 013702 001204 MOV KRXVEC,R2
2450 006102 012722 011526 MOV #INTSERV,(R2)+
2451 006106 012712 000340 MOV #PR7,(R2)
2452 006112 004737 007066 JSR PC,WTRDCK
2453 006116 042737 000200 013136 BIC #BIT7,SEQUEN
2454 006124 000137 004260 JMP DEXIT ;END OF TEST 26

```



```

2455
2456 ;THE FOLLOWING SECTION OF CODE WILL ALLOW PROVIDING INFORMATION
2457 ;TO THE USER WHEN AN 'UNEXPECTED' BUS TIMEOUT TO LOCATION 4 OCCURS
2458
2459 006130 104400 016734 BUSERR: TYPE, LOC4M ;TYPE MESSAGE INDICATING AN
2460 ;UNEXPECTED BUS TIMEOUT OCCURRED
2461 006134 012646 MOV (SP)+,-(SP) ;SAVE (SP)+ FOR TYPEOUT
2462 ;SETUP TO TYPE PC WHERE TIMEOUT OCCURRED
2463 006136 104402 TYPOS ;GO TYPE--OCTAL ASCII
2464 006140 006 .BYTE 6 ;TYPE 6 DIGITS
2465 006141 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
2466 006142 104400 017047 TYPE, PCM ;TYPE MESSAGE '=PC'
2467 006146 012716 002466 MOV #REBGIN,(SP) ;SET RETURN 'PC' TO START THE
;PROGRAM OVER AGAIN
2468 RTI ;RETURN TO BEGINNING OF PROGRAM
2469 006152 000002
2470
2471 ;THE FOLLOWING SECTION OF CODE WILL ALLOW PROVIDING INFORMATION
2472 ;TO THE USER WHEN AN 'UNEXPECTED' RESERVED INSTRUCTION TRAP TO LOCATION
2473 ;10 OCCURS
2474
2475 006154 104400 017001 RESERR: TYPE, LOC10M ;TYPE MESSAGE INDICATING AN
2476 ;UNEXPECTED RESERVED INSTRUCTION
2477 ;TRAP OCCURRED
2478 006160 012646 MOV (SP)+,-(SP) ;SAVE (SP)+ FOR TYPEOUT
2479 ;SETUP TO TYPE PC WHERE RESERVED TRAP OCCURRED
2480 006162 104402 TYPOS ;GO TYPE--OCTAL ASCII
2481 006164 006 .BYTE 6 ;TYPE 6 DIGITS
2482 006165 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
2483 006166 104400 017047 TYPE, PCM ;TYPE MESSAGE '=PC'
2484 006172 012716 002466 MOV #REBGIN,(SP) ;SET RETURN 'PC' TO START THE
;PROGRAM OVER AGAIN
2485 RTI ;RETURN TO BEGINNING OF PROGRAM
2486 006176 000002
2487
2488 ;THIS ROUTINE WILL CALCULATE THE PRIORITY LEVEL FOR THE PROCESSOR
2489 ;BASED ON THE CURRENT PRIORITY LEVEL OF THE DEVICE (CONTENTS OF 'BRLEV:')
2490
2491 006200 013700 001214 CPUPRI: MOV BPLEV,R0 ;GET THE PROPOSED RX11 DEVICE
2492 ;INTERRUPT PRIORITY LEVEL VALUE
2493 006204 105701 TSTR R1 ;IS CPU LEVEL TO BE THE SAME AS
2494 ;THE DEVICE LEVEL OR 1 LESS?
2495 006206 100401 BMI 1S ;BRANCH IF SAME AS 1
2496 006210 005300 DEC R0 ;DROP DEVICE LEVEL PRIORITY
2497 ;BY 1 LEVEL FOR PSW
2498 006212 006300 1S: ASL R0 ;FORM BITS <7-5> FOR PSW
2499 006214 006300 ASL R0 ;
2500 006216 006300 ASL R0 ;
2501 006220 006300 ASL R0 ;
2502 006222 006300 ASL R0 ;
2503 006224 042700 000037 BIC #37,R0 ;ENSURE THAT T,N,Z,V, & C BITS
2504 ;FOR THE PROCESSOR ARE CLEAR
2505 006230 000207 RTS PC ;RETURN TO MAINLINE CODE
2506

```

```

2507 ;SBTIL " ERROR " TRAP SERVICE ROUTINE
2508
2509 ;*****
2510 ;*****
2511 ; " ERROR "
2512 ;*****
2513 ;*****
2514 ;*****
2515 ;*****
2516
2517 006232 011637 006474 XERROR: MOV # SP, EPCSCOPE ; RETURN ADDRESS FROM " ERROR"
2518 006236 062737 000002 006474 ADD #2, EPCSCOPE ; NOW (EPCSCOPE) = SUBSCOPE+2, OK SCOPE+2
2519 006244 005237 006472 INCERRORS: INC ERRORS
2520 006250 001775 BEQ INCERRORS
2521
2522 ; DATA SW 13 = 0 TO PRINT APPROPRIATE ERROR MESSAGE
2523
2524 006252 104405 CKSWR
2525 006254 032777 020000 172734 BIT #SW13,SWR
2526 006262 001056 BNE NOPRINT
2527 006264 005037 002526 CLP CCOUNT
2528 006270 032737 000400 012724 BIT #BIT0,UNITSEL ;WAS PREVIOUS ERROR REPORTED ON THIS PASS
2529 006276 001002 BNE 1S
2530 006300 104400 015636 TYPE, MXEHEADER
2531
2532 006304 104400 016120 1S: TYPE, MCRLF
2533 006310 011604 MOV # SP, R4 ; ERADR
2534 006312 162704 000002 SUB #2, R4
2535 006316 010446 MOV R4, -(SP)
2536 006320 104402 TYPOS
2537 006322 006 .BYTE 6
2538 006323 001 .BYTE 1
2539 006324 104400 016641 TYPE, SPACE
2540 006330 013746 002532 MOV FAST, -(SP) ; FAST (FIRST ADDRESS OF SELECTED TEST)
2541 006334 104403 TYPON
2542 006336 104400 016641 TYPE, SPACE
2543 006342 013746 006556 MOV PCSCOPE, -(SP) ; FAPT (FIRST ADDRESS OF PRESENT TEST)
2544 006346 104403 TYPON
2545 006350 104400 016641 TYPE, SPACE
2546 006354 010246 MOV R2, -(SP) ; BLANK
2547 006356 104403 TYPON
2548 006360 104400 016641 TYPE, SPACE
2549 006364 010046 MOV R0, -(SP) ; EXPECTED (GOOD) RESULT OF TEST
2550 006366 104403 TYPON
2551 006370 104400 016641 TYPE, SPACE
2552 006374 010146 MOV R1, -(SP) ; ACTUAL (BAD) RESULT OF TEST
2553 006376 104403 TYPON
2554 006400 104400 016641 TYPE, SPACE
2555 006404 013737 002530 006416 MOV PASS,28
2556 006412 004537 015600 JSR R5,SGLDEC
2557 006416 000000 2S: OPEN

```



```

2558 ; DATA SW 0 = 0 TO RING BELL AT ERROR
2559
2560 006420 052737 000400 012724 NOPRINT: BIS #BIT0,UNITSEL ;SET HARD ERROR FLAG
2561 006426 004737 006450 JSR PC,DING
2562
2563 ; DATA SW 15 = 1 TO HALT AT ERROR
2564
2565 006432 104405 1S: CKSWR
2566 006434 032777 100000 172554 BIT #SW15,0SWR
2567 006442 001401 BEQ 2S
2568 006444 000000 HALT
2569 006446 000002 2S: RTI
2570
2571 006450 104405 DING: CKSWR
2572 006452 032777 000001 172536 BIT #SW0,0SWR
2573 006460 001002 BNE 1S
2574 006462 104400 006470 TYPE ,MABELL
2575 006466 000207 1S: RTS PC
2576
2577 006470 000007 MABELL: ,ASCIZ <07> ; DING = A = LING
2578 ,EVEN
2579
2580 006472 000000 ERRORS: 0
2581 006474 000000 EPCSCOPE: 0

```

```

2582 ,SBTTL " SCOPE " TRAP SERVICE ROUTINE
2583
2584 ; " SCOPE "
2585
2586 006476 005737 006472 XSCOPE: TST ERRORS
2587 006502 001015 BNE SCOPING
2588
2589 ; NO ERRORS HAVE BEEN DETECTED
2590
2591 ; JUST SET (PCSCOPE) = FIRST ADDRESS OF THE SCOPE LOOP
2592
2593 ; (IN CASE ERRORS ARE DETECTED LATER)
2594
2595 006504 005037 006472 NOSCOPE: CLR ERRORS
2596 006510 011637 006556 MOV @ SP, PCSCOPE
2597 006514 000002 RTI
2598
2599 ; " SUBSCOPE "
2600
2601 006516 005737 006472 XSUBSCOPE: TST ERRORS
2602 006522 001001 BNE 1S
2603 006524 000002 RTI ; NO ERRORS EXIST
2604
2605 ; ERRORS DO EXIST
2606
2607 ; IF THIS ERROR ADDRESS IS THE SAME ADDRESS WITHIN PROGRAM LOCATION " EPCSCOPE"
2608
2609 ; THEN THIS IS A SCOPING LOOP
2610
2611 ; IF NOT - THEN EXIT
2612
2613 006526 021637 006474 1S: CMP @ SP, EPCSCOPE
2614 006532 001401 BEQ SCOPING
2615 006534 000002 RTI
2616
2617 ; SW 11 = 1 TO LOCK ON SCOPING LOOP
2618
2619 ; THIS IS A SCOPING LOOP
2620
2621 006536 104405 SCOPING: CKSWR
2622 006540 032777 004000 172450 BIT #SW11,0SWR
2623 006546 001756 BEQ NOSCOPE ;DO NOT LOOP ON ERROR
2624 006550 013716 006556 MOV PCSCOPE, @ SP
2625 006554 000002 RTI ; LOCK FOR SCOPE LOOP
2626 006556 000000 PCSCOPE: 0

```


AINDEC-11-DZRXB-E MACY11 27(732) 16-MAR-76 15:55 PAGE 60
ZRXBE,P11 " SCOPE " TRAP SERVICE ROUTINE

SEQ 0085

```

2693
2694 006674 005015 042504 044526 MHUNGPC: ,ASCIZ <15><12>"DEVICE TEST HUNG @ PC "
2695 006702 042503 052040 051505
2696 006710 020124 052510 043516
2697 006716 040040 050040 020103
2698 006724 0000
2699 006725 040 040520 051523 MPASS: ,ASCIZ " PASS ="
2700 006732 036440 0000
2701 006736 ,EVEN
2702
2703 006736 005037 006756 XSDN: CLR HANGER
2704 006742 012737 177740 MOV #177740,HANGPL
2705 006750 062716 000002 ADD #2, @ SP , UPDATE FOR EXIT
2706 006754 000207 RTS PC
2707 006756 000000 HANGER: 0
2708 006760 177740 HANGPL: 177740

```



```

2699          ,SBTTL DRIVE TEST SELECTION
2700
2701          ;DO A READ ONLY FUNCTION ON ALL SECTORS.
2702          ;THIS DOES NOT VERIFY THE DATA, ONLY TESTS FOR CRC ERRORS.
2703
2704          RDONLY:      JSR PC,INITTRACKS
2705                      JSR PC,GETUNIT
2706          1S:         JSR PC,GETTRACK
2707                      JSR PC,READ
2708                      DEC TRKNCTR
2709                      BNE 1S
2710                      RTS PC
2711
2712          ;*****
2713          ;WRITE AND READ DATA ON SPECIFIED TRACK AND ALTERNATE
2714          ;DRIVES BEFORE GOING TO THE NEXT TRACK,
2715
2716          DRVSWP:      JSR PC,GETPATTERN
2717                      JSR PC,INITTRACKS
2718          1S:         JSR PC,GETUNIT
2719                      JSR PC,GETTRACK
2720                      JSR PC,WRITE
2721                      JSR PC,HEADCHK
2722                      JSR PC,GETUNIT
2723                      JSR PC,WRITE
2724                      JSR PC,READCHK
2725                      DEC TRKNCTR
2726                      BNE 1S
2727                      RTS PC
2728
2729          ;*****
2730          ;WRITE ALL SECTORS AND READ/VERIFY ALL SECTORS
2731
2732          WTRDCK:      JSR PC,GETPATTERN
2733                      JSR PC,INITTRACKS
2734          XWTRDCK:     JSR PC,GETUNIT
2735                      JSR PC,GETTRACK
2736                      JSR PC,WRITE
2737                      JSR PC,READCHK
2738                      DEC TRKNCTR
2739                      BNE 1S
2740                      JSR PC,DONE
2741                      RTS PC
2742                      BR XWTRDCK
2743
2744          ;HAVE BOTH DRIVES BEEN TESTED
2745          ;YES
2746          ;NO, GO TO OTHER UNIT

```

```

2746          ,SBTTL WRITE FUNCTION
2747
2748          WRITE:      JSR PC,INITSECTOR
2749                      JSR PC,GETSECTOR
2750          XWRITE:     JSR PC,ADJUSUM
2751          FILLBUF:    MOV #FILLDONE,-(SP)
2752                      MOV #FILLER,-(SP)
2753                      CLR BYTECNTR
2754                      -(SP)
2755                      ;LOWER "CPU" LEVEL
2756                      ;SET RETURN "PC"
2757                      ;GET "CPU" LEVEL INTO "PSW"
2758                      ;EXECUTE FILLBUFER COMMAND
2759                      ;TEST FOR TRANSFER REQUEST FLAG
2760          1S:         MOV #FBIE,@RXCS
2761                      TSTB @RXCS
2762                      BPL FILLFLAG
2763                      MOV (R0)+,@RXDB
2764                      INC BYTECNTR
2765                      BR FILLFLAG
2766
2767          CLR
2768          MOV
2769          RTI
2770
2771          XFRYTE:     MOV #FILLER,@RXCS
2772                      TSTB @RXCS
2773                      BPL FILLFLAG
2774                      MOV (R0)+,@RXDB
2775                      INC BYTECNTR
2776                      BR FILLFLAG
2777
2778          FILLER:     TST (SP)+
2779                      MOV #FILL,PTYP1+2
2780                      MOV #XWRITE,PCONT+2
2781                      MOV #FILLBUF,PLOOP+2
2782                      JMP PARTEST
2783
2784          PARTEST:    CKSWR
2785                      BIT #SW13,@SWR
2786                      BNE CONT4
2787                      TYPE ,OPEN
2788                      TYPE ,MPAR
2789                      TYPE ,MCRLF
2790                      CKSWR
2791                      TST @SWR
2792                      BPL CONT13
2793                      HALT
2794                      ;HALT ON ERROR
2795                      ;TEST LOOP ON ERROR SWITCH
2796                      ;IF NOT SET GO TO NEXT SECTOR
2797                      ;RETURN TO LOOP ON TEST THROUGH HERE
2798                      ;GO TO NEXT SECTOR THROUGH HERE
2799
2800          CONT4:      CKSWR
2801                      TST @SWR
2802                      BPL CONT13
2803                      HALT
2804                      ;HALT ON ERROR
2805                      ;TEST LOOP ON ERROR SWITCH
2806                      ;IF NOT SET GO TO NEXT SECTOR
2807                      ;RETURN TO LOOP ON TEST THROUGH HERE
2808                      ;GO TO NEXT SECTOR THROUGH HERE
2809
2810          FILLDONE:   CLR HANGER
2811                      MOV #WRTDONE,-(SP)
2812                      MOV #WRTIE,-(SP)
2813                      MOV #WRTIE,FUNCTION
2814                      CMP #T25,PCSCOPE
2815                      BNE 1S
2816                      MOV #WTDIE,FUNCTION
2817          1S:         JSR PC,COMMWORD
2818                      -(SP)
2819                      #25,-(SP)
2820                      ;TRANSFER COMMAND TO DRIVE
2821                      ;LOWER "CPU" LEVEL
2822                      ;SET RETURN "PC"
2823                      ;GET "CPU" LEVEL INTO "PSW"
2824                      ;WAIT FOR DONE
2825
2826          CLR
2827          MOV
2828          RTI
2829
2830          2S:         BIT #DONEBIT,@RXCS
2831                      BEQ 2S
2832                      INC HANGER
2833                      BNE 3S
2834                      JMP NOINTER
2835
2836          3S:         ;WAIT FOR INTERRUPT
2837                      ;NO INTERRUPT ERROR

```



```

2801 007430 005337 013414 WRTDONE: DEC SECCNTR ;TEST SECTOR COUNTER
2802 007434 001001 BNE 28 ;NOT LAST SECTOR GO TO NEXT ONE
2803 007436 000207 RTS PC
2804 007440 000137 007140 28: JMP XWRITE
2805
2806 007444 005726 WRTER: TST (SP)+ ;REMOVE THE DONE RETURN FROM THE STACK
2807 007446 032737 000002 012146 BIT #BIT1,ASTAT ;IS THIS A PARITY ERROR
2808 007454 001413 BEO WRTSEK ;NO, IT MUST BE A SEEK ERROR
2809 ;PARITY ERROR DURING A WRITE FUNCTION
2810 007456 012737 016570 007266 MOV #MWRITE,PTYP1+2 ;PUT ADDR OF WRITE MESSAGE IN PAR ER TYP0UT 1
2811 007464 012737 007430 007330 MOV #WRTDONE,PCONT+2 ;IF NO LOOP GO TO NEXT SECTOR
2812 007472 012737 007332 007324 MOV #FILLDONE,PL0OP+2 ;IF LOOP RETURN THROUGH PLOOP TO REWRITE
2813 007500 000137 007252 JMP PARTEST ;GO INC LOG AND TEST FOR RETRY
2814
2815 ;SEEK ERROR DURING A WRITE FUNCTION
2816 007504 012737 007144 007604 WRTSEK: MOV #FILLBUF,SEKRTY+2 ;SETUP FOR WRT RETRY ON SEEK ERROR
2817 ;(AFTER A RECAL, THE CONTENTS OF SECTOR 1,
2818 ;TRACK 1 ARE LOADED INTO THE SECTOR BUFFER,
2819 ;TO REWRITE THE CORRECT DATA THE PROGRAM
2820 ;MUST REFILL THE SECTOR BUFFER,
2821 007512 012737 016570 007542 MOV #MWRITE,STYP1+2 ;PUT ADDR OF WRITE MESSAGE IN SEEK ER TYP0UT 1
2822 007520 004737 007526 JSR PC,SEEKER ;RECORD SEEK ERROR
2823 007524 000741 BR WRTDONE ;GO TO NEXT SECTOR CAN'T FIND THIS ONE
2824
2825 007526 104405 SEEKER: CKSWR
2826 007530 032777 020000 171460 BIT #SW13,0SWR ;CHECK DON'T PRINT ERROR SWITCH
2827 007536 001004 BNE SWHLT1
2828 007540 104400 016570 STYP1: TYPE ,MWRITE ;PRINT WRITE (READ) SEEK ERROR
2829 007544 004737 007606 JSR PC,SEKTPY
2830 007550 104405 SWHLT1: CKSWR
2831 007552 005777 171440 TST 0SWR ;TEST THE HALT ON ERROR SWITCH
2832 007556 100001 BPL CONT14
2833 007560 000000 HLT7: HALT ;HALT ON THE ERROR
2834 007562 004737 007706 CONT14: JSR PC,HOME ;RECALIBRATE ON SEEK ERRORS
2835 007566 104405 CKSWR
2836 007570 032777 004000 171420 BIT #SW11,0SWR ;CHECK THE LOOP ON ERROR SWITCH
2837 007576 001001 BNE SEKRTY ;IF SET LOOP ON THE ERROR THROUGH SEEK RETRY,
2838 007600 000207 RTS PC
2839 007602 000137 007144 SEKRTY: JMP FILLBUF ;RETRY WRITE COMMAND (READ COMAND)
2840
2841 007606 104400 016555 SEKTPY: TYPE ,MSEEK ;TYPE SEEK ERROR
2842 007612 104400 016032 TYPE ,MPRES ;TYPE ADDRESS OF TRACK MOVED FROM
2843 007616 013746 013130 PRESTRK, -(SP) ;SAVE PRESTRK FOR TYPEOUT
2844 007622 104402 MOV TYP0S ;GO TYPE--OCTAL ASCII
2845 007624 003 ,BYTE 3 ;TYPE 3 DIGIT(S)
2846 007625 000 ,BYTE 0 ;SUPPRESS LEADING ZEROS
2847 007626 104400 016120 TYPE ,MCRLF
2848 007632 000207 RTS PC
2849

```

```

2850 007634 153737 012724 007702 COMMWORD: BISB UNITSEL,FUNCTION ;SET UNIT SELECTION BIT IN COMMAND WORD
2851 007642 013777 007702 171336 MOV FUNCTION,0RXCS ;SEND OUT COMMAND TO DRIVE
2852 007650 004737 006560 18: JSR PC,STR ;WAIT FOR TR FLAG
2853 007654 000775 BR 18
2854 007656 113777 013416 171324 MOV B TSECTOR,0RXDB ;SEND OUT TARGET SECTOR
2855 007664 004737 006560 28: JSR PC,STR ;WAIT FOR TR FLAG
2856 007670 000775 BR 28
2857 007672 113777 013126 171310 MOV B TARGET,0RXDB ;SEND OUT TARGET TRACK
2858 007700 000207 RTS PC
2859
2860 007702 000000 FUNCTION: 0
2861 007704 000000 DATAK: 0 ;DATA CHECK ON CRC ERROR FLAG
2862
2863 007706 104405 HOME: CKSWR
2864 007710 032777 000400 171300 BIT #SW8,0SWR ;TEST NO RECAL SWITCH
2865 007716 001035 BNE RTN
2866 007720 012777 040001 171260 MOV #RECAL,0RXCS ;ISSUE RECAL FUNCTION
2867 007726 004737 006574 28: JSR PC,SDN
2868 007732 000775 BR 28
2869 007734 005777 171246 TST 0RXCS ;WAS THERE AN ERROR
2870 007740 100021 BPL XHOME ;NO
2871 007742 104405 CKSWR
2872 007744 032777 020000 171244 BIT #BIT13,0SWR ;YES, SHOULD IT BE PRINTED
2873 007752 001002 BNE 18 ;NO
2874 007754 004737 012152 JSR PC,RDCODE
2875 007760 104405 18: CKSWR
2876 007762 005777 171230 TST 0SWR ;TEST HALT ON ERROR SWITCH
2877 007766 100001 BPL 38
2878 007770 000000 HALT
2879 007772 032777 004000 171216 38: BIT #SW11,0SWR ;TEST LOOP ON ERROR SWITCH
2880 010000 001342 BNE HOME
2881 010002 000207 RTS PC
2882 010004 012737 000001 013130 XHOME: MOV #1,PRESTRK ;SET THE PRESENT TRACK TO THACK 1
2883 010012 000207 RTN: RTS PC
2884

```



```

2885                      ,SHITL  READ DATA FROM THE DISKETTE
2886
2887
2888 010014 004737 013324      READ:      JSR PC,INITSECTOR
2889 010020 004737 013422      XREAD:     JSR PC,GETSECTOR
2890 010024 005037 007704      HREAD:     CLR DATABK          ;CLEAR CRC DATA CHECK FLAG
2891 010030 005037 006756      CLR HANGER
2892 010034 012746 010110      MOV #RDONE,-(SP)      ;SET GOOD RETURN ON STACK
2893 010040 012746 010142      MOV #RDERR,-(SP)      ;SET READ ERROR RETURN ON STACK
2894 010044 112737 000107      MOVB #RDIE,FUNCTION
2895 010052 004737 007634      JSR PC,COMMWORD
2896 010056 005046      CLR          ;LOWER 'CPU' LEVEL
2897 010060 012746 010066      MOV          ;SET RETURN 'PC'
2898 010064 000002      RTI          ;GET 'CPU' LEVEL INTO 'PSW'
2899 010066 032777 000040 171112 1$:      BIT #DONEBIT,0RXC5      ;WAIT FOR DONE BIT
2900 010074 001774      BEQ 1$
2901 010076 005237 006756      2$:      INC HANGER          ;WAIT FOR INTEKRUPT
2902 010102 001375      BNE 2$
2903 010104 000137 011462      JMP NOINTER          ;NO INTERRUPT ON DONE
2904
2905 010110 022737 005462 006556 RDDONE:     CMP #T20,PCSCOPE      ;IS THIS THE READ ONLY TEST (T20)
2906 010116 001405      BEQ NEXTRD      ;YES,DON'T CHECK FOR DELETED DATA
2907 010120 004737 010410      JSR PC,DDCHK      ;CHECK FOR DELETED DATA INDICATION
2908 010124 005701      TST R1          ;BIT 15 OF R1 IS READ 1 SECTOR FLAG
2909 010126 100001      BPL NEXTRD
2910 010130 000207      RTS PC          ;IF SET,GO VERIFY DATA JUST READ
2911 010132 005337 013414      NEXTRD:     DEC SECCNTR
2912 010136 001330      BNE XREAD
2913 010140 000207      RTS PC          ;READ FUNCTION IS DONE
2914
2915 010142 005726      RDEPR:      TST (SP)+          ;REMOVE THE DONE RETURN FROM THE STACK
2916 010144 032737 000002 012146      BIT #BIT1,ASTAT      ;IS THIS A PARITY ERROR
2917 010152 001413      BEQ 1$          ;NO, SEE IF ITS A CRC ERROR
2918                      ;PARITY ERROR DURING A READ FUNCTION
2919 010154 012737 016530 007266      MOV #RREAD,PTYPI+2      ;PUT ADDR OF READ MESSAGE IN PAR EPR TYPEOUT 1
2920 010162 012737 010024 007324      MOV #RREAD,PLLOOP+2      ;IF LOOP ON ERROR LOOP THROUGH PLOOP
2921 010170 012737 010132 007330      MOV #NEXTRD,PCONT+2      ;IF NO LOOP GO TO NEXT READ
2922 010176 000137 007252      JMP PARTEST      ;RECORD PARITY ERROR AND RETRY FUNCTION
2923 010202 032737 000001 012146 1$:      BIT #BIT0,ASTAT      ;IS THIS A CRC ERROR
2924 010210 001011      BNE CRCER          ;YES GO TEST AND LOG IT
2925                      ;SEEK ERROR DURING A READ FUNCTION
2926 010212 012737 010024 007604      MOV #RREAD,SEKRTY+2      ;SET SEEK CONTINUE FOR READ RETRY
2927 010220 012737 016530 007542      MOV #RREAD,STYPI+2      ;SET ADDR OF READ MESSAGE IN SEEK ER TYPEOUT 1
2928 010226 004737 007526      JSR PC,SEEKER
2929 010232 000737      BR NEXTRD      ;GO TO NEXT SECTOR,CAN'T READ THIS ONE

```

```

2930                      ;CRC ERROR DETECTED WHILE READING
2931
2932 010234 005701      CRCER:      TST R1          ;IF READ ONLY, REPORT DATA CRC ERROR
2933 010236 100034      BPL DATACRC
2934 010240 005237 007704      INC DATABK      ;SET DATA CHECK FLAG
2935 010244 004737 010644      JSR PC,EMPBUFF      ;CHECK FOR A DATA ERROR
2936 010250 005737 011460      TST ERCNTR      ;WAS THERE A DATA ERROR
2937 010254 001025      BNE DATACRC      ;YES, REPORT IT
2938 010256 104405      CKSWR
2939 010260 032777 020000 170730      BIT #SW13,0SWR      ;TEST DON'T PRINT SWITCH
2940 010266 001004      BNE 2$
2941 010270 104400 016500      TYPE ,MBADCRC      ;TYPE CRC GENERATOR ERROR
2942 010274 104400 016120      TYPE ,MCRLF
2943 010300 104405      2$:      CKSWR
2944 010302 005777 170710      TST 0SWR      ;TEST HALT ON ERROR SWITCH
2945 010306 100001      BPL CONT15
2946 010310 000000      HALT          ;HALT ON ERROR
2947 010312 032777 004000 170676      BIT #SW11,0SWR      ;CHECK LOOP ON ERROR SWITCH
2948 010320 001001      BNE 3$
2949 010322 000703      BR NEXTRD      ;DON'T LOOP GO TO NEXT SECTOR
2950 010324 000137 010024      3$:      JMP REREAD      ;LOOP ON TEST.
2951
2952                      ;DATA CRC EPROR
2953
2954 010330 104405      DATACRC:     CKSWR
2955 010332 032777 020000 170656      BIT #SW13,0SWR      ;TEST DON'T PRINT ERROR SWITCH
2956 010340 001004      BNE 4$
2957 010342 104400 016536      TYPE ,MCRC      ;TYPE DATA CRC ERROR
2958 010346 104400 016120      TYPE ,MCRLF
2959 010352 104405      4$:      CKSWR
2960 010354 005777 170636      TST 0SWR      ;TEST HALT ON ERROR SWITCH
2961 010360 100001      BPL CONT16
2962 010362 000000      HALT          ;HALT ON ERROR
2963 010364 032777 004000 170624      BIT #SW11,0SWR      ;TEST LOOP ON ERROR
2964 010372 001004      BNE 5$      ;IF SET LOOP ON THE TEST
2965 010374 062706 000002      ADD #2,SP      ;REMOVE READ DONE ADDRESS FROM STACK
2966 010400 000137 010132      JMP NEXTRD      ;READ NEXT SECTOR CAN'T READ THIS ONE
2967 010404 000137 010024      5$:      JMP REREAD      ;NO,GO REREAD THIS SECTOR
2968
2969

```



```

2970 010410 022737 006050 006556 DDCHK: CMP #T25,PCSCOPE ;IS THIS TEST 25
2971 010416 001041 BNE CONT10 ;THIS IS TEST 25
2972 010420 132737 000100 012146 BITB #BIT6,ASTAT ;DD BIT SHOULD BE SET
2973 010426 001056 BNE RETURN ;TEST DON'T PRINT ERROR SWITCH
2974 010430 104405 CKSWR
2975 010432 032777 020000 170556 BIT #SW13,@SWR ;TEST DON'T PRINT ERROR SWITCH
2976 010440 001013 BNE CONT11
2977 010442 004737 010566 JSR PC,ERMSG ;TYPE MISSING DELETED DATA BIT
2978 010446 104400 015747 TYPE ,MDDMIS ;SET HARD ERROR FLAG
2979 010452 052737 000400 012724 DDERR: BIS #BIT8,UNITSEL ;TYPE ADDRESS OF ERROR
2980 010460 004737 012050 JSR PC,TYPADR
2981 010464 104400 016120 TYPE ,MCRLF
2982 010470 104405 CKSWR
2983 010472 005777 170520 TST @SWR ;TEST HALT ON ERROR SWITCH
2984 010476 100001 BPL CONT17
2985 010500 000000 HLT13: HALT ;HALT ON DELETED DATA ERROR
2986 010502 032777 004000 170506 CONT17: BIT #SW11,@SWR ;TEST LOOP ON ERROR
2987 010510 001402 BEQ 48
2988 010512 000137 010024 JMP REREAD ;LOOP ON TEST
2989 010516 000137 010132 JSR NEXTRD ;READ NEXT SECTOR
2990 010522 032737 000100 012146 CONT10: BIT #BIT6,ASTAT ;THIS IS NOT A DELETED DATA TRANSFER
2991 010530 001415 BEQ RETURN
2992 010532 052737 000400 012724 BIS #BIT8,UNITSEL ;SET HARD ERROR FLAG
2993 010540 104405 CKSWR
2994 010542 032777 020000 170446 BIT #SW13,@SWR ;TEST DON'T PRINT ERROR SWITCH
2995 010550 001347 BNE CONT11
2996 010552 004737 010566 JSR PC,ERMSG
2997 010556 104400 015721 TYPE ,MUNXDD ;TYPE UNEXPECTED DELETED DATA BIT
2998 010562 000733 BR DDERR
2999 010564 000207 RETURN: RTS PC
3000
3001
3002 010566 104400 016123 ERMSG: TYPE ,MERHEADER
3003 010572 013746 006556 MOV PCSCOPE,-(SP) ;SAVE PCSCOPE FOR TYPEOUT
3004 010576 104402 TYP0S ;GO TYPE--OCTAL ASCII
3005 010600 006 ;TYPE 6 DIGITS
3006 010601 000 ;SUPPRESS LEADING ZEROS
3007 010602 104400 006725 TYPE ,MPASS
3008 010606 013737 002530 MOV PASS,IS
3009 010614 004537 015600 JSR R5,SGLDEC
3010 010620 000000 OPFN
3011 010622 104400 016120 IS: TYPE ,MCRLF
3012 010626 004737 006450 JSR PC,DING
3013 010632 000207 RTS PC
3014
3015

```

```

3016 ,SBTTL READ AND VERIFY DATA
3017
3018 ;READ A SECTOR,EMPTY THE SECTOR BUFFER AND VERIFY
3019 ;THE DATA READ AGAINST CORE DATA BUFFER
3020
3021 010634 052701 100000 READCHK: BIS #BIT15,R1 ;SET READ ONE SECTOR FLAG
3022 010640 004737 010014 JSR PC,READ ;GO READ ONE SECTOR
3023 010644 005737 013414 EMPBUFF: TST SECCNTH ;IF CLEARED NO SECTORS WERE FOUND
3024 010650 001002 BNE IS
3025 010652 000137 011446 JMP EXIT ;GO TO NEXT TRACK
3026 010656 005037 011452 IS: CLR BYTECNTR ;CLEAR THE BYTE AND ERROR COUNTERS
3027 010662 005037 011460 CLR ERNCNTR
3028 010666 052701 000200 BIS #BIT7,R1 ;R1 BIT 7 IS USED AS FIRST ERROR FLAG
3029 010672 004737 010766 JSR PC,ADJSUM ;ADJUST DATA AND CK SUM FOR ADDRESSES
3030 010676 005037 011056 CLR CKSUM ;SET UP FOR CHECK SUM ACCUMULATION
3031 010702 012746 011306 MOV #EMPDONE,-(SP) ;SET UP RETURN ADDRESSES
3032 010706 012746 011060 MOV #EMPER,-(SP)
3033 010712 005046 CLP ;LOWER 'CPU' LEVEL
3034 010714 012746 010722 MOV #26,-(SP) ;SET RETURN 'PC'
3035 010720 000002 RTI ;GET 'CPU' LEVEL INTO 'PSW'
3036 010722 012777 000103 170256 28: MOV #EBIE,@RXCS ;LOAD EMPTY BUFFER FUNCTION
3037 010730 105777 170252 EMPFLAG: TST @RXCS ;TEST FOR TR FLAG
3038 010734 100375 BPL EMPFLAG
3039
3040 010736 117737 170246 011454 CKBYTE: MOV @RXDB,BADBYTE ;SAVE BYTE FROM DISKETTE
3041 010744 063737 011454 011056 ADD BADBYTE,CKSUM ;ACCUMULATE CHECK SUM
3042 010752 123720 011454 CMPB BADBYTE,(R0)+ ;COMPARE AGAINST GOOD BYTE
3043 010756 001054 BNE DATAER ;IF NOT EQUAL GO TO DATAER
3044 010760 005237 011452 INC BYTECNTR
3045 010764 000761 BR EMPFLAG ;GET NEXT BYTE
3046
3047 010766 113737 013126 017404 ADJSUM: MOVB TARGET,BUFADR ;SET FIRST AND SECOND BYTES WITH ADDRESSES
3048 010774 113737 013416 017405 MOVB TSECTOR,BUFADR+1
3049 011002 013737 012536 011056 MOV SUM,CKSUM ;GET THE PATTERN SUM
3050 011010 063737 013126 011056 ADD TARGET,CKSUM ;ADD TRACK ADDRESS TO CHECK SUM
3051 011016 063737 013416 011056 ADD TSECTOR,CKSUM ;ADD SECTOR ADDRESS TO CHECK SUM
3052 011024 113737 011056 017602 INC CKSUM,BUFADR+176 ;INSERT CHECK SUM TO DATA BUFFER
3053 011032 106337 011056 ASLB CKSUM ;GENERATE NEGATIVE CHECK SUM
3054 011036 105437 011056 NEGB CKSUM
3055 011042 113737 011056 017603 MOVB CKSUM,BUFADR+177 ;INSERT NEG,SUM INTO DATA BUFFER
3056 011050 012700 017404 MOV #BUFADR,R0 ;SET ADDRESS OF BYTE IN R0
3057 011054 000207 RTS PC ;RETURN
3058
3059 011056 000000 CKSUM: 0
3060
3061 011060 005726 EMPEM: TST (SP)+ ;REMOVE THE DONE RETURN FROM THE STACK
3062 011062 012737 016402 007266 MOV #EMPTYP,PTYP+2 ;PUT ADDR OF EMPTYBUF MESSAGE IN PAR ER TYP0UT 1
3063 011070 012737 010644 007324 MOV #EMPBUFF,PL00P+2 ;RETURN THROUGH HERE TO LOOP ON ERROR
3064 011076 012737 011430 007330 MOV #NXREAD,PCONT+2 ;IF NO LOOP ON ERROR GO TO NEXT SECTOR
3065 011104 000137 007252 JMP PARTEST ;REPORT PARITY ERROR
3066

```



```

3067 011110 052737 000400 012724 DATAER: BIS #BIT8,UNITSEL ;SET THE HAD ERROR FLAG
3068 011116 005237 011460 INC ERCNTR ;INC THE BYTE ERROR COUNTER
3069 011122 104405 CKSWR
3070 011124 032777 020000 170064 BIT #SW13,0SWR ;TEST PRINT ERROR SW IN SWR
3071 011132 001054 BNE NOERTYP ;DON'T PRINT THE ERROR
3072 011134 032777 001000 170054 BIT #SW9,0SWR ;TEST PRINT 10 ERRORS SWITCH
3073 011142 001004 BNE 15 ;IF SET PRINT ALL ERRORS
3074 011144 023727 011460 000012 CMP ERCNTR,#10. ;HAVE 10 ERRORS BEEN TYPED
3075 011152 003044 BGT NOERTYP ;YES,DON'T PRINT ANY MORE
3076 011154 105701 15: TSTB R1 ;TEST FIRST ERROR FLAG
3077 011156 100014 BPL TYPERR
3078 011160 004737 010566 JSR PC,ERMSG ;PRINT ADDRESS OF TEST
3079 011164 104400 015772 TYPE ,MDERHDR ;FIRST ERROR, PRINT ERROR HEADER
3080 011170 104400 016120 TYPE ,MCRLF
3081 011174 004737 012050 JSR PC,TYPADR ;PRINT TRACK AND SECTOR LOCATIONS
3082 011200 104400 016061 TYPE ,MCMUN ;SET UP CMUN HEADINGS
3083 011204 042701 000200 BIC #BIT7,R1 ;CLEAR FIRST ERROR FLAG
3084 011210 013737 011452 011222 TYPERR: MOV BYTECNTR,15 ;PRINT BYTE NUMBER
3085 011216 004537 015600 JSR R5,SGLDEC
3086 011222 000000 15: OPEN
3087 011224 104400 016115 TYPE ,DBLSP
3088 011230 013746 011454 MOV BADBYTE,-(SP) ;PRINT BYTE READ FROM DISKETTE
3089 011234 104402 TYPOS
3090 011236 000003 ,WORD 3
3091 011240 104400 016115 TYPE ,DBLSP
3092 011244 114037 011456 MOVR -(R0),GOODBYTE ;GET GOOD BYTE
3093 011250 005200 INC R0 ;RETURN R0 TO NEXT BYTE IN BUFFER
3094 011252 013746 011456 MOV GOODBYTE,-(SP)
3095 011256 104403 TYPON ;PRINT GOOD DATA
3096 011260 104400 016120 TYPE ,MCRLF
3097 011264 104405 CKSWR
3098 011266 005777 167724 TST 0SWR ;TEST HALT ON ERROR SWITCH
3099 011272 100001 BPL CONT20
3100 011274 000000 HLT14: HALT
3101 011276 005237 011452 CONT20: INC BYTECNTR
3102 011302 000137 010730 JMP EMPFLAG
3103

```

```

3104 011306 005737 007704 EMPDONE: TST DATAK ;WAS THIS READ CHECK CAUSED BY A CRC ERROR
3105 011312 001401 REQ 15 ;NO
3106 011314 000207 RTS PC ;YES,RETURN TO CRC HANDLER
3107 011316 005737 011460 15: TST ERCNTR ;WAS THERE ERRORS
3108 011322 001442 BEQ NXREAD ;NO ERRORS
3109 011324 104405 CKSWR
3110 011326 032777 020000 167602 BIT #SW13,0SWR ;YES,TEST DON'T PRINT SWITCH
3111 011334 001024 BNE 25 ;DON'T PRINT THE ERROR
3112 011336 104400 016333 TYPE ,MERCT ;PRINT THE TOTAL DATA ERROR COUNT
3113 011342 013737 011460 011354 MOV ERCNTR,35
3114 011350 001537 015600 JSR R5,SGLDEC
3115 011354 000000 35: OPEN
3116 011356 104400 016650 TYPE ,MSUM ;INDICATE IF CHECK SUM WAS GOOD OR HAD ERRORS
3117 011362 105737 011056 TSTB CKSUM
3118 011366 001403 BEQ 45
3119 011370 104400 016635 TYPE ,MBAD
3120 011374 000402 BR 55
3121 011376 104400 016643 TYPE ,MG00D
3122 011402 104400 016120 45: TYPE ,MCRLF
3123 011406 104405 25: CKSWR
3124 011410 032777 000400 167600 BIT #SW11,0SWR ;TEST LOOP ON ERROR SWITCH
3125 011416 001404 BEQ NXREAD ;IF NOT SET GO TO NEXT SECTOR
3126 011420 004737 010024 JSR PC,RERead ;YES,GO REREAD THE DATA
3127 011424 000137 010644 JMP EMPRUFF ;GO RECHECK THE DATA
3128 011430 005337 013414 NXREAD: DEC SECCNTR
3129 011434 001404 BEQ EXIT
3130 011436 004737 010020 JSR PC,XREAD ;READ THE NEXT SECTOR
3131 011442 000137 010644 JMP EMPBUFF
3132 011446 005001 EXIT: CLR R1 ;CLEAR THE ONE READ FLAG
3133 011450 000207 RTS PC
3134
3135 011452 000000 BYTECNTR: 0
3136 011454 000000 BADBYTE: 0
3137 011456 000000 GOODBYTE: 0
3138 011460 000000 ERCNTR: 0
3139
3140 ;*****
3141 ;AN INTERRUPT DID NOT OCCURE AT A FUNCTION DONE FLAG.
3142
3143
3144 011462 104405 NOINTER: CKSWR
3145 011464 032777 020000 167524 BIT #SW13,0SWR ;TEST DON'T PRINT ERROR SWITCH
3146 011472 001006 BNE 15
3147 011474 004737 010566 JSR PC,ERMSG ;TYPE NO INTERRUPT ON DONE ERROR
3148 011500 104400 016256 TYPE ,MINTER
3149 011504 104400 016120 TYPE ,MCRLF
3150 011510 104405 15: CKSWR
3151 011512 005777 167500 TST 0SWR ;TEST HALT ON ERROR SWITCH
3152 011516 100001 BPL CONT21
3153 011520 000000 HLT15: HALT ;HALT ON ERROR
3154 011522 004737 011526 CONT21: JSR PC,INTSERV ;JSR TO INTSERV AS IF IT WAS AN INTERRUPT
3155

```



```

3156 ,SHTTL INTERRUPT SERVICE
3157
3158 011526 117737 167456 012146 INTSERV: MOVB @RXDB,ASTAT ;SAVE THE ERROR AND STATUS WORD
3159 011534 005777 167446 TST @RXCS ;TEST THE ERROR FLAG
3160 011540 100444 BMI RXERROR ;THERE WAS AN ERROR GO REPORT IT
3161 011542 032737 000004 012146 BIT #BIT2,ASTAT ;IS INIT DONE SET
3162 011550 001402 BEQ 2S ;NO,CONTINUE
3163 011552 000137 012006 JMP RXPWR ;YES,REPORT POWER FAILED AND RESTART
3164 011556 032737 000003 012146 2S: BIT #3,ASTAT ;ARE PAR OR CRC BITS SET
3165 011564 001021 BNE 1S ;YES GO REPORT ERROR
3166 011566 132777 000040 167412 BITB #DONEBIT,@RXCS ;IS DONE SET
3167 011574 001012 BNE 3S ;IF SET RETURN TO TEST
3168 011576 104405 CКСWR
3169 011600 032777 020000 167410 BIT #SW13,@SWR ;TEST DON'T PRINT ERROR SWITCH
3170 011606 001004 BNE 4S ;DON'T PRINT
3171 011610 104400 016311 TYPE ,MUKNINT ;TYPE UNKNOWN INTERRUPT
3172 011614 104400 016120 TYPE ,MCRLF
3173 011620 000002 RTI ;RETURN FROM THE INTERRUPT
3174 011622 062706 000006 3S: ADD #6,SP ;BYPASS INTERRUPT POINTERS ON STACK
3175 011626 000207 RTS PC ;RETURN TO PROGRAM
3176 011630 104405 1S: CКСWR
3177 011632 032777 020000 167356 BIT #SW13,@SWR ;TEST DON'T PRINT ERROR SWITCH
3178 011640 001004 BNE RXERROR
3179 011642 104400 016614 TYPE ,MNOFLAG ;TYPE NO STATUS ERROR ERROR
3180 011646 104400 016120 TYPE ,MCRLF
3181 011652 005237 006472 RXERROR: INC ERRORS ;AN ERROR INDICATOR
3182 011656 001775 BEQ RXERROR
3183 011660 052737 000400 012724 BIS #BIT8,UNITSEL ;SET HARD ERROR FLAG
3184 011666 012777 000017 167312 MOV #RDER,@RXCS ;GET THE ERROR CODE
3185 011674 004737 006574 3S: JSR PC,SDN ;TEST FOR DONE FLAG
3186 011700 000775 BR 3S
3187 011702 032777 000002 167300 BIT #2,@RXDB ;WAS THERE A PARITY ERROR
3188 011710 001403 BEQ 1S ;NO,CONTINUE
3189 011712 004737 012022 JSR PC,PARTYP ;YES,GO REPORT THE PARITY ERROR
3190 011716 000763 BR 2S ;REISSUE THE FUNCTION
3191 011720 117737 167264 012150 1S: MOVB @RXDB,ASTAT ;SAVE THE ERROR CODE IN B STATUS
3192 011726 104405 CКСWR
3193 011730 032777 020000 167260 BIT #SW13,@SWR ;TEST PRINT ERROR SWITCH IN SWR
3194 011736 001020 BNE 2S
3195 011740 104400 016120 TYPE ,MCRLF ;TYPE ERROR AND MESSAGES
3196 011744 004737 010566 JSR PC,ERMMSG ;TYPE COMMAND STATUS REGISTER
3197 011750 104400 016212 TYPE ,MRXCS ;TYPE COMMAND STATUS REGISTER
3198 011754 013746 007702 FUNCTION, -(SP) ;SAVE FUNCTION FOR TYPEOUT
3199 011760 104402 ;GO TYPE--OCTAL ASCII
3200 011762 306 ;TYPE 6 DIGIT(S)
3201 011763 000 ;SUPPRESS LEADING ZEROS
3202 011764 004737 012050 JSR PC,TYPADR ;TYPE ADDRESSES AND RUN CONDITIONS
3203 011770 104400 016120 TYPE ,MCRLF
3204 011774 004737 012220 JSR PC,TYPCODE ;PRINT THE STATUS REGISTERS
3205 012000 062706 000004 2S: ADD #4,SP ;MOVE ERROR RETURN TO TOP OF STACK
3206 012004 000207 RTS PC
3207
3208 012006 104400 016665 RXPWR: TYPE ,MRX11 ;ONLY THE RX11 POWER HAS FAILED
3209 012012 104400 015270 TYPE ,SPOWER ;PRINT POWER FAILED
3210 012016 000137 001350 JMP RESTART ;GO TO RESTART

```

```

3211 012022 104400 016212 PARTYP: TYPE ,MRXCS ;SAVE @RXCS FOR TYPEOUT
3212 012026 017746 167154 MOV @RXCS, -(SP) ;GO TYPE--OCTAL ASCII
3213 012032 104402 TYPPOS ;TYPE 6 DIGIT(S)
3214 012034 006 .BYTE 6 ;SUPPRESS LEADING ZEROS
3215 012035 000 .BYTE 0
3216 012036 104400 016577 TYPE ,MPAR
3217 012042 104400 016120 TYPE ,MCRLF
3218 012046 000207 RTS PC
3219
3220 012050 104400 016020 TYPADR: TYPE ,MTRK ;TYPE TRACK ADDRESS
3221 012054 013746 013126 TARGET, -(SP) ;SAVE TARGET FOR TYPEOUT
3222 012060 104402 TYPPOS ;GO TYPE--OCTAL ASCII
3223 012062 003 .BYTE 3 ;TYPE 3 DIGIT(S)
3224 012063 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
3225 012064 104400 016047 TYPE ,MSECT ;TYPE SECTOR ADDRESS
3226 012070 013737 013416 012144 MOV TSECTOR, 2S
3227 012076 042737 177740 012144 BIC #177740, 2S ;CLEAR ALL BUT SECTOR ADDRESS
3228 012104 013746 012144 2S, -(SP) ;SAVE 2S FOR TYPEOUT
3229 012110 104400 TYPPOS ;GO TYPE--OCTAL ASCII
3230 012112 002 .BYTE 2 ;TYPE 2 DIGIT(S)
3231 012113 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
3232 012114 104400 016115 TYPE ,DBLSP
3233 012120 032737 000020 012724 BIT #BIT4,UNITSEL ;WHICH DRIVE IS BEING USED
3234 012126 001003 BNE 1S
3235 012130 104400 016163 TYPE ,MUNIT0 ;TYPE UNIT 0
3236 012134 000402 BR 4S
3237 012136 104400 016173 TYPE ,MUNIT1 ;TYPE UNIT 1
3238 012142 000207 1S:
3239 012144 000000 4S:
2S:
RTS PC
OPEN
3240
3241 012146 000000 ASTAT: 0
3242 012150 000000 RSTAT: 0
3243
3244
3245 012152 117737 167032 012146 RDCODE: MOVB @RXDB,ASTAT ;SAVE THE A STATUS
3246 012160 012777 000017 167020 2S: MOV #RDER,@RXCS ;READ THE B STATUS REGISTER
3247 012166 004737 006574 3S: JSR PC,SDN ;WAIT FOR DONE FLAG
3248 012172 000775 BR 3S
3249 012174 032777 000002 167006 BIT #2,@RXDB ;WAS THERE A PARITY ERROR
3250 012202 001403 BEQ 1S ;NO,CONTINUE
3251 012204 004737 012022 JSR PC,PARTYP ;YES,REPORT THE PARITY ERROR
3252 012210 000763 BR 2S ;RETRY READING STATUS B
3253 012212 117737 166772 012150 1S: MOVB @RXDB,ASTAT ;SAVE THE B STATUS CODES
3254 012220 104400 016222 TYPE ,MASTAT ;TYPE THE CONTENTS OF THE TWO STATUS REGISTERS
3255 012224 013746 012146 ASTAT, -(SP) ;SAVE ASTAT FOR TYPEOUT
3256 012230 104402 TYPPOS ;GO TYPE--OCTAL ASCII
3257 012232 003 .BYTE 3 ;TYPE 3 DIGIT(S)
3258 012233 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
3259 012234 104400 016106 TYPE ,TAB
3260 012240 104400 016236 TYPE ,MBSTAT
3261 012244 013746 012150 MOV BSTAT, -(SP)
3262 012250 104403 TYPON
3263 012252 104400 016120 TYPE ,MCRLF
3264 012256 000207 RTS PC

```



```

3265 ,SBTTL PATTERN GENERATOR
3266
3267 ;NOTE: ALL DATA PATTERNS WILL BE MODIFIED SO THE FIRST BYTE WILL
3268 ;CONTAIN THE TRACK ADDRESS, THE SECOND BYTE WILL CONTAIN THE UNIT
3269 ;NUMBER AND SECTOR ADDRESS IN WHICH THE DATA IS WRITTEN, THE MOST
3270 ;SIGNIFICANT BIT OF THIS SECOND BYTE INDICATES THE UNIT, UNIT 0
3271 ;IF "0" UNIT 1 IF "1", THE LAST TWO BYTES CONTAIN THE CHECK SUM,
3272 ;*****
3273
3274
3275
3276 012260 012704 017404 GETPATTERN: MOV #BUFADR,R4 ;SET ADDRESS OF FIRST DATA BYTE
3277 012264 005037 012536 CLR SUM ;SET UP FOR ACCUMULATION OF CHECK SUM
3278 012270 013705 012324 MOV PAT,R5 ;GET PATTERN BITS
3279 012274 006305 ASL R5
3280 012276 000175 012302 JMP @PATTERNS(R5)
3281 012302 012326 PATTERNS: DATA0 ;000 DATA BYTE
3282 012304 012340 DATA1 ;377 DATA BYTE
3283 012306 012350 FLOAT0 ;FLOAT A 0 THROUGH ALL 1'S
3284 012310 012412 FLOAT1 ;FLOAT A 1 THROUGH ALL 0'S
3285 012312 012420 PAT125 ;125/052 DATA WORD
3286 012314 012440 PAT314 ;314/063 DATA WORD
3287 012316 012450 COUNT ;INCREMENT DATA PATTERN
3288 012320 012470 RANDATA ;RANDOM DATA BYTE
3289
3290
3291 012322 000000 DATABYTE: 0
3292 012324 000000 PAT: 0
3293
3294
3295 ;*****
3296
3297 ;LOAD SOFTWARE BUFFER WITH ALL ZEROS
3298 ; PAT = 0
3299
3300 012326 005037 012322 DATAB: CLR DATABYTE
3301 012332 004737 012510 PATGEN: JSR PC,LOAD ;GO LOAD THE DATA BUFFER
3302 012336 000775 BR PATGEN
3303
3304 ;*****
3305
3306 ;LOAD SOFTWARE BUFFER WITH ALL ONES
3307 ; PAT = 1
3308
3309
3310 012340 112737 000377 012322 DATA1: MOVH #377,DATABYTE
3311 012346 000771 BR PATGEN
3312

```

```

3313 ;FLOAT A 0 THROUGH ONES IN SOFTWARE BUFFER
3314 ; PAT = 2
3315
3316 012350 112737 000376 012322 FLOAT0: MOVH #376,DATABYTE ;SET UP A ONES FIELD
3317 012356 000261 XPATGEN: SEC ;SET THE C BIT TO ROTATE THROUGH THE DATA
3318 012360 012702 000000 1S: MOV #0,R2 ;CLR R2 (CAN'T USE "CLR" IT CLEARS "C" BIT)
3319 012364 103001 BCC 2S ;BR IF "C" BIT IS CLEARED
3320 012366 005202 INC R2 ;SET R2 IF "C" BIT IS SET
3321 012370 004737 012510 2S: JSR PC,LOAD ;GO LOAD THE DATA BUFFER
3322 012374 000241 CLC ;CLEAR THE "C" BIT
3323 012376 005702 TST R2 ;IS R2 NONZERO
3324 012400 001401 BEQ 3S
3325 012402 000261 SEC ;YES, SET THE "C" BIT
3326 012404 106137 012322 3S: ROLB DATABYTE
3327 012410 000763 BR 1S
3328
3329 ;*****
3330
3331 ;FLOAT A 1 THROUGH ALL ZEROS IN SOFTWARE BUFFER
3332 ; PAT = 3
3333
3334 012412 005037 012322 FLOAT1: CLR DATABYTE
3335 012416 000757 BR XPATGEN
3336
3337 ;*****
3338
3339 ;LOAD SOFTWARE BUFFER WITH ALTERNATING 1 AND 0 FOR
3340 ;ONE BYTE AND THE COMPLIMENT INTO THE NEXT
3341 ; PAT = 4
3342
3343 012420 112737 000125 012322 PAT125: MOVH #125,DATABYTE
3344 012426 004737 012510 XPATGEN: JSR PC,LOAD
3345 012432 105137 012322 COMB DATABYTE
3346 012436 000773 BR XPATGEN
3347
3348 ;*****
3349
3350 ;LOAD SOFTWARE BUFFER WITH ALTERNATING PAIRS OF 1 AND 0 AND
3351 ;COMPLIMENT INTO THE NEXT
3352 ; PAT = 5
3353
3354 012440 112737 000314 012322 PAT314: MOVH #314,DATABYTE
3355 012446 000767 BR XPATGEN
3356
3357 ;*****
3358
3359 ;LOAD SOFTWARE BUFFER WITH COUNT PATTERN
3360 ; PAT = 6
3361
3362 012450 012737 000377 012322 COUNT: MOV #377,DATABYTE
3363 012456 005237 012322 1S: INC DATABYTE
3364 012462 004737 012510 JSR PC,LOAD
3365 012466 000773 BR 1S

```



```

3366 ;*****
3367
3368 ;LOAD SOFTWARE BUFFER WITH RANDOM DATA PATTERN
3369 ; PAT = 7
3370
3371 012470 004737 012540 RANDATA: JSR PC,RANGEN ;GET RANDOM NUMBER
3372 012474 113737 012632 012322 MOV B RANUM,DATABYTE
3373 012502 004737 012510 JSR PC,LOAD
3374 012506 000770 BR RANDATA
3375
3376 012510 063737 012322 012536 LOAD: ADD DATABYTE,SUM ;ACCUMULATE THE PATTERN CHECK SUM
3377 012516 113724 012322 MOV B DATABYTE,(R4)+ ;LOAD THE DATA BUFFER
3378 012522 022704 017604 CMP #BUFADR+200,R4 ;HAVE 128 BYTES BEEN GENERATED
3379 012526 001401 BEQ 1$ ;IF YES,RETURN TO TEST
3380 012530 000207 RTS PC ;IF NO,RETURN TO PATTERN GENERATOR
3381 012532 005726 1$: TST (SP)+ ;TAKE PATTERN RETURN ADDRESS OF STACK
3382 012534 000207 RTS PC ;RETURN TO TEST
3383
3384 012536 000000 SUM: 0
3385
3386 012540 012700 000001 RANGEN: MOV #1,R0
3387 012544 063700 012626 ADD RAN1,R0
3388 012550 063700 012630 ADD RAN2,R0
3389 012554 042700 170000 BIC #170000,R0
3390 012560 000241 CLC
3391 012562 006100 ROL R0
3392 012564 006100 ROL R0
3393 012566 010037 012626 MOV R0,RAN1
3394 012572 005000 CLR R0
3395 012574 013700 012630 MOV RAN2,R0
3396 012600 006000 ROR R0
3397 012602 006000 ROR R0
3398 012604 063700 012626 ADD RAN1,R0
3399 012610 042700 170000 BIC #170000,R0
3400 012614 010037 012630 MOV R0,RAN2
3401 012620 010037 012632 MOV R0,RANUM
3402 012624 000207 RTS PC
3403
3404 012626 001234 RAN1: 001234
3405 012630 000765 RAN2: 000765
3406 012632 000000 RANUM: 0
3407

```

```

3408 ;$BTTL UNIT SELECTION
3409
3410 ;TEST FOR SELECTED UNITS,DRIVE READY,AND USED CONDITIONS
3411 ;ALSO CONTAINS A "HAD ERROR" FLAG TO BE TESTED AT EOP.
3412 ;THE BITS IN UNITSEL ARE USED AS FOLLOWS
3413 ;
3414 ;BIT15 =UNIT 1 SELECTED VIA SWR
3415 ;BIT14 =UNIT 1 USED BIT
3416 ;BIT8 =THIS PASS HAD AN ERROR
3417 ;BIT7 =UNIT 0 SELECTED VIA SWR
3418 ;BIT6 =UNIT 0 USED BIT
3419 ;BIT4 =UNIT SELECTION FOR FUNCTION WORD
3420
3421 ;*****
3422
3423 012634 032737 000100 012724 GETUNIT: BIT #BIT6,UNITSEL ;WAS UNIT 0 JUST USED
3424 012642 001012 BNE 1$ ;UNIT 0 USED CHECK UNIT 1
3425 012644 105737 012724 TSTB UNITSEL ;WAS UNIT 0 SELECTED
3426 012650 100007 BPL 1$ ;NO GO TO UNIT 1
3427 012652 042737 040020 012724 BIC #40020,UNITSEL ;CLEAR UNIT 1 USED BIT AND FUNCTION UNIT BIT
3428 012660 052737 000100 012724 RIS #BIT6,UNITSEL ;SET UNIT 0 USED BIT
3429 012666 000207 RTS PC
3430 012670 005737 012724 1$: TST UNITSEL ;WAS UNIT 1 SELECTED
3431 012674 100012 BPL 2$ ;NO RETURN
3432 012676 032737 040000 012724 BIT #BIT14,UNITSEL ;HAS UNIT 1 BEEN USED
3433 012704 001006 BNE 2$ ;YES RETURN
3434 012706 042737 000100 012724 BIC #BIT6,UNITSEL ;CLEAR UNIT 0 USED BIT
3435 012714 052737 040020 012724 BIS #40020,UNITSEL ;SET UNIT 1 USED BIT AND FUNCTION UNIT BIT
3436 012722 000207 RTS PC
3437
3438
3439
3440 012724 000000 UNITSEL: 0
3441
3442 ;TEST THAT ALL UNITS HAVE BEEN ACCESSED
3443
3444 012726 005737 012724 DONE: TST UNITSEL ;IS UNIT 1 SELECTED
3445 012732 100006 BPL 1$ ;NO RETURN
3446 012734 032737 040000 012724 BIT #BIT14,UNITSEL ;YES HAS IT BEEN USED
3447 012742 001002 BNE 1$ ;YES RETURN
3448 012744 062716 000002 ADD #2,0SP ;BYPASS NOT DONE RETURN ON STACK
3449 012750 000207 RTS PC
3450

```



```

3451          ,SBTTL TRACK SEQUENCE SELECTION
3452
3453          ,INITIALIZE TRACK SEQUENCE
3454
3455          ;NOTE: IF WORD SEQUEN IS CLEARED THEN TRACK SEQUENCE IS FROM 0-52-53-114 ONLY
3456          ;IF BIT 15 OF SEQUEN IS "1" THEN TRACK SELECTION IS INC. BETWEEN SELECTED OD/ID LIMITS,
3457          ;IF BIT 7 IS "1" THEN TEST 25 DECREMENT SEQUENCE IS REQUIRED,
3458
3459
3460
3461          012752 105737 013136      INITTRACK:      TSTB SEQUEN          ;IS THIS TEST 26 SPECIAL SEQUENCE
3462          012756 100442              BMI 25          ;YES, DEC FROM TRACK 12 TO 0
3463          012760 042737 100200 001200      BIC #100200,OD      ;CLEAR FIRST USED BITS
3464          012766 005737 001200      TST OD          ;TEST CONTENTS OF ID,OD FOR 0
3465          012772 001440              BEQ 35          ;SEQUENCE WILL BE FROM "HOME"-52-53-114-0
3466          012774 052737 100000 013136      BIS #BIT15,SEQUEN      ;LIMITS WERE SELECTED, INC FROM OD TO ID,
3467          013002 113737 001200 013126      MOVB OD,TARGET      ;INIT OD AS PRESENT TRACK
3468          013010 005037 013134      CLR XID          ;INIT WORKING ID AND OD LOCATIONS
3469          013014 113737 001201 013134      MOVB ID,XID
3470          013022 005037 013132      CLR XOD
3471          013026 113737 001200 013132      MOVB OD,XOD
3472          013034 013737 013134 013124      MOV XID,TRKCNTR      ;SET UP NUMBER OF TRACK MOVEMENTS
3473          013042 163737 013132 013124      SUB XOD,TRKCNTR
3474          013050 005237 013124      INC TRKCNTR
3475          013054 052737 100200 001200 15:      BIS #100200,OD      ;SET FIRST TIME BITS IN ID,OD
3476          013062 000207              RTS PC
3477          013064 012737 000013 013124 25:      MOV #13,TPKCNTR      ;SET TRACK COUNTER
3478          013072 000770              BR 15
3479          013074 012737 000004 013124 35:      MOV #4,TPKCNTR      ;SET THE TRACK COUNTER
3480          013102 000764              BR 15
3481
3482
3483          ;*****
3484
3485          013104 113737 013126 013130  GETTRACK:      MOVB TARGET,PRESTRK      ;RESET TO PRESENT TRACK
3486          013112 005737 013136      TST SEQUEN      ;IS THIS THE LIMITED SEQUENCE
3487          013116 001410              BEQ LIMTRK      ;YES, DOING ONLY 0-52-53-114
3488          013120 100446              BMI SEQ1      ;NO,SEQUENCE IS BETWEEN SELECTED LIMITS
3489          013122 000463              BR SEQ2      ;NO,THIS IS TEST 26 DEC SEQUENCE
3490
3491          013124 000000      TRKCNTR:      0
3492          013126 000000      TARGET:      0
3493          013130 000000      PRESTRK:      0
3494          013132 000000      XOD:      0
3495          013134 000000      ID:      0
3496          013136 000000      SEQUEN:      0
3497

```

```

3498          ;*****
3499
3500          ;LIMITED SEQUENCE, ACCESS TRACKS 52 TO 53 TO 114 BACK TO 0
3501          013140 005737 001200      LIMTRK:      TST OD          ;TEST HIGH ORDER FIRST TIME BIT
3502          013144 100007              BPL 15          ;NOT SET, ON TRACK 52
3503          013146 012737 000052 013126      MOV #52,TARGET      ;GO TO TRACK 52
3504          013154 042737 100000 001200      BIC #BIT15,OD      ;CLEAR FIRST TIME BIT
3505          013162 000207              RTS PC
3506          013164 105737 001200 15:      TSTB OD          ;TEST LOW ORDER FIRST TIME BIT
3507          013170 100007              BPL 25          ;NOT SET, ON TRACK 53
3508          013172 012737 000053 013126      MOV #53,TARGET      ;GO TO TRACK 53
3509          013200 042737 000200 001200      BIC #BIT7,OD
3510          013206 000207              RTS PC
3511          013210 023727 013126 000114 25:      CMP TARGET,#114      ;IS IT ON TRACK 114
3512          013216 001404              BEQ 35          ;YES,GO TO TRACK 0
3513          013220 012737 000114 013126      MOV #114,TARGET      ;NO, GO TO TRACK 114
3514          013226 000207              RTS PC
3515          013230 005037 013126 35:      CLR TARGET      ;GO TO TRACK 0
3516          013234 000207              RTS PC
3517
3518          ;*****
3519
3520          ;INCREMENT FROM OD+1 TO ID AND RETURN TO OD
3521          ; USED WHEN TRACK LIMITS ARE SELECTED
3522
3523          013236 042737 100200 001200  SEQ1:      BIC #100200,OD      ;CLEAR FIRST TIME BITS
3524          013244 123737 013134 013130      CMPB XID,PRESTRK      ;PRESENT TRACK EQUAL TO ID
3525          013252 001404              BNE 15          ;NO GET NEW TRACK
3526          013254 113737 001200 013126      MOVB OD,TARGET      ;YES RETURN TO OD
3527          013262 000207              RTS PC
3528          013264 005237 013126 15:      INC TARGET      ;ADD 1 TO TARGET TRACK
3529          013270 000207              RTS PC
3530
3531          ;*****
3532
3533          ;DECREMENT FROM ID = 12 TO OD = 0
3534          ;USED IN TEST 26 ONLY
3535
3536          013272 005737 001200      SEQ2:      TST OD          ;FIRST TIME BIT SET
3537          013276 100007              BPL 15          ;NO GET NEXT TRACK
3538          013300 042737 100200 001200      BIC #100200,OD      ;YES CLEAR FIRST TIME BITS
3539          013306 012737 000012 013126      MOV #12,TARGET      ;MOVE OUT 10 TRACKS
3540          013314 000207              RTS PC
3541          013316 005337 013126 15:      DEC TARGET      ;MOVE TO NEXT TRACK
3542          013322 000207              RTS PC

```



```

3543          ,SBTTL SECTOR SELECTION
3544
3545          ,SECTOR INITIALIZATION AND SELECTION
3546
3547 013324 005737 001202 INITSECTOR: TST FIRST          ;TEST FIRST AND LAST FOR 0
3548 013330 001005          BNE 15          ;SECTORS SPECIFIED USE THEM
3549 013332 005237 001202          INC FIRST          ;NONE SPECIFIED SET FIRST TO 1
3550 013336 112737 000032 001203          MOVB #32,LAST          ;SET LAST TO MAXIMUM
3551 013344 113737 001203 013414 1s: MOVB LAST,SECCNTR          ;SET UP SECTOR COUNTER
3552 013352 163737 001202 013414          SUB FIRST,SECCNTR
3553 013360 005237 013414          INC SECCNTR
3554 013364 105037 013415          CLRB SECCNTR+1
3555 013370 113737 001202 013416          MOVB FIRST,TSECTOR          ;PUT FIRST SECTOR IN TARGET SECTOR
3556 013376 162737 000003 013416          SUB #3,TSECTOR          ;SUB 3 FROM TSECTOR AS FIRST TIME THROUGH
3557          ;IT GETS ADDED BACK ON,
3558 013404 012737 000001 013420          MOV #1,INTLEAV          ;SET INTERLEAVE OFFSET
3559 013412 000207          RTS PC
3560
3561 013414 000000          SECCNTR: 0
3562 013416 000000          TSECTOR: 0
3563 013420 000000          INTLEAV: 0
3564
3565 013422 042737 000200 013416 GETSECTOR: RLC #200,TSECTOR          ;CLEAR THE UNIT BIT BEFORE TESTING
3566 013430 062737 000003 013416          ADD #3,TSECTOR          ;ADD 3 FOR INTERLEAVING
3567 013436 123737 001203 013416          CMPB LAST,TSECTOR
3568 013444 002010          BGE 15          ;NEW SECTOR IS WITHIN LIMITS
3569 013446 113737 001202 013416          MOVB FIRST,TSECTOR          ;RESET TARGET SECTOR TO INTERLEAVE
3570 013454 063737 013420 013416          ADD INTLEAV,TSECTOR          ;ADD ON INTERLEAVE OFFSET VALUE
3571 013462 005237 013420          INC INTLEAV          ;UP DATE THE OFFSET VALUE
3572 013466 032737 000020 012724 1s: BIT #BIT4,UNITSEL          ;IS THIS UNIT 0
3573 013474 001403          BEQ 25
3574 013476 052737 000200 013416          BIS #BIT7,TSECTOR          ;NO, SET UNIT IDENTIFIER IN TARGET SECTOR
3575 013504 000207          RTS PC

```

```

3576          ,SBTTL TYPE ROUTINE
3577
3578          ;*****
3579          ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
3580          ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
3581          ;*NOTE1: SNULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
3582          ;*NOTE2: SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
3583          ;*NOTE3: SFILLC CONTAINS THE CHARACTER TO FILL AFTER.
3584          ;*
3585          ;*CALL:
3586          ;*1) USING A TRAP INSTRUCTION
3587          ;* TYPE ,MESADR          ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
3588          ;*OR
3589          ;* TYPE
3590          ;* MESADR
3591          ;*
3592
3593 013506 105737 013735          $TYPE: TSTB $TPFLG          ;IS THERE A TERMINAL?
3594 013512 100002          BPL 15          ;BR IF YES
3595 013514 000000          HALT          ;HALT HERE IF NO TERMINAL
3596 013516 000407          BR 35          ;LEAVE
3597 013520 010046          1s: MOV R0,-(SP)          ;SAVE R0
3598 013522 017600 000002          MOV 02(SP),R0          ;GET ADDRESS OF ASCIZ STRING
3599 013526 112046          2s: MOVB (R0)+,-(SP)          ;PUSH CHARACTER TO BE TYPED ONTO STACK
3600 013530 001005          BNE 45          ;BR IF IT ISN'T THE TERMINATOR
3601 013532 005726          TST (SP)+          ;IF TERMINATOR POP IT OFF THE STACK
3602 013534 012600          60s: MOV (SP)+,R0          ;RESTORE R0
3603 013536 062716 000002          3s: ADD #2,(SP)          ;ADJUST RETURN PC
3604 013542 000002          RTI          ;RETURN
3605 013544 122716 000011          4s: CMPB #HT,(SP)          ;BRANCH IF <HT>
3606 013550 001430          BEQ 85          ;
3607 013552 122716 000200          CMPB #CRLF,(SP)          ;BRANCH IF NOT <CRLF>
3608 013556 001006          BNE 55          ;
3609 013560 005726          TST (SP)+          ;POP <CR><LF> EQUIV
3610 013562 104400          TYPE          ;TYPE A CR AND LF
3611 013564 013737          SCRLF
3612 013566 105037 013722          CLRB          ;CLEAR CHARACTER COUNT
3613 013572 000755          BR 25          ;GET NEXT CHARACTER
3614 013574 004737 013656          5s: JSR PC,STYPEC          ;GO TYPE THIS CHARACTER
3615 013600 123726 013734          6s: CMPB $FILLC,(SP)+          ;IS IT TIME FOR FILLER CHARS.?
3616 013604 001350          BNE 25          ;IF NO GO GET NEXT CHAR.
3617 013606 013746 013732          MOV $NULL,-(SP)          ;GET # OF FILLER CHARS. NEEDED
3618          ;AND THE NULL CHAR.
3619 013612 105366 000001          7s: DECB 1(SP)          ;DOES A NULL NEED TO BE TYPED?
3620 013616 002770          BLT 65          ;BR IF NO--GO POP THE NULL OFF OF STACK
3621 013620 004737 013656          JSR PC,STYPEC          ;GO TYPE A NULL
3622 013624 105337 013722          DECB $CHARCNT          ;DO NOT COUNT AS A COUNT
3623 013630 000770          BR 75          ;LOOP
3624
3625          ;HORIZONTAL TAB PROCESSOR
3626
3627 013632 112716 000040          8s: MOVB #* ,(SP)          ;REPLACE TAB WITH SPACE
3628 013636 004737 013656          9s: JSR PC,STYPEC          ;TYPE A SPACE
3629 013642 123737 000007 013722          BITB #7,$CHARCNT          ;BRANCH IF NOT AT
3630 013650 001372          BNE 95          ;TAB STOP
3631 013652 005726          TST (SP)+          ;POP SPACE OFF STACK

```



```

3632 013654 000724      BP      28      ;GET NEXT CHARACTER
3633 013656 105777      STYPEC: TSTB 08TPS ;WAIT UNTIL PRINTER IS READY
3634 013662 100375      BPL      STYPEC
3635 013664 116677      MOVW 2(SP),08TPB ;LOAD CHAR TO BE TYPED INTO DATA REG.
3636 013672 122766      CMPB 000002 000036 ;IS CHARACTER A CARRIAGE RETURN?
3637 013700 001003      BNE 15 ;BRANCH IF NO
3638 013702 105037      CLRB 013722 ;YES--CLEAR CHARACTER COUNT
3639 013706 000406      BR      STYPEX ;EXIT
3640 013710 122766      15: CMPB 0LF,2(SP) ;IS CHARACTER A LINE FEED?
3641 013716 001402      BEQ      STYPEX ;BRANCH IF YES
3642 013720 105227      INCB (PC)+ ;COUNT THE CHARACTER
3643 013722 000000      SCHARCNT: WORD 0 ;CHARACTER COUNT STORAGE
3644 013724 000207      STYPEX: RTS PC
3645
3646 013726 177564      STPS: ,WORD 177564 ;TTY PRINTER STATUS REG. ADDRESS
3647 013730 177566      STPB: ,WORD 177566 ;TTY PRINTER BUFFER REG. ADDRESS
3648 013732 000000      $NULL: ,BYTE 0 ;CONTAINS NULL CHARACTER FOR FILLS
3649 013733 000000      $FILLC: ,BYTE 2 ;CONTAINS # OF FILLER CHARACTERS REQUIRED
3650 013734 000000      $TPFLG: ,BYTE 12 ;INSERT FILL CHARS. AFTER A "LINE FEED"
3651 013735 000000      $QUES: ,ASCII "?" ;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
3652 013736 000000      $CRLF: ,ASCII "<15>" ;QUESTION MARK
3653 013737 000000      $LF: ,ASCII "<12>" ;CARRIAGE RETURN
3654 013740 000012      ;LINEFEED

```

```

3655      .SHTTL BINARY TO OCTAL (ASCII) AND TYPE
3656
3657      ;*****
3658      ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3659      ;OCTAL (ASCII) NUMBER AND TYPE IT.
3660      ;STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3661      ;CALL:
3662      ;*      MOV      NUM,-(SP)      ;NUMBER TO BE TYPED
3663      ;*      TYPPOS      ;CALL FOR TYPEOUT
3664      ;*      ,BYTE      N      ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3665      ;*      ,BYTE      M      ;M=1 OR 0
3666      ;*      ;1=TYPE LEADING ZEROS
3667      ;*      ;0=SUPPRESS LEADING ZEROS
3668
3669      ;STYPOC---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3670      ;STYPOS OR STYPOC
3671      ;CALL:
3672      ;*      MOV      NUM,-(SP)      ;NUMBER TO BE TYPED
3673      ;*      TYPON      ;CALL FOR TYPEOUT
3674      ;*
3675      ;STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3676      ;CALL:
3677      ;*      MOV      NUM,-(SP)      ;NUMBER TO BE TYPED
3678      ;*      TYPOC      ;CALL FOR TYPEOUT
3679
3680 013742 017646 000000 STYPOS: MOV 0(SP),-(SP) ;PICKUP THE MODE
3681 013746 116637 000001 014165 MOVW 1(SP),00FILL ;LOAD ZERO FILL SWITCH
3682 013754 112637 014167 MOVW (SP)+,0MODE+1 ;NUMBER OF DIGITS TO TYPE
3683 013760 062716 000002 ADD #2,(SP) ;ADJUST RETURN ADDRESS
3684 013764 000406 BR STYPOC
3685 013766 112737 000001 014165 STYPOC: MOVW #1,00FILL ;SET THE ZERO FILL SWITCH
3686 013774 112737 000006 014167 MOVW #6,0MODE+1 ;SET FOR SIX(6) DIGITS
3687 014002 112737 000005 014164 STYPOC: MOVW #5,0CNT ;SET THE ITERATION COUNT
3688 014010 010346 MOV R3,-(SP) ;SAVE R3
3689 014012 010446 MOV R4,-(SP) ;SAVE R4
3690 014014 010546 MOV R5,-(SP) ;SAVE R5
3691 014016 113704 014167 MOVW 0MODE+1,R4 ;GET THE NUMBER OF DIGITS TO TYPE
3692 014022 005404 NEG R4
3693 014024 062704 000006 ADD #6,R4 ;SUBTRACT IT FOR MAX. ALLOWED
3694 014030 110437 014166 MOVW R4,0MODE ;SAVE IT FOR USE
3695 014034 113704 014165 MOVW 00FILL,R4 ;GET THE ZERO FILL SWITCH
3696 014040 016605 000012 MOV 12(SP),R5 ;PICKUP THE INPUT NUMBER
3697 014044 005003 CLR R3 ;CLEAR THE OUTPUT WORD
3698 014046 006105 15: ROL R5 ;ROTATE MSB INTO "C"
3699 014050 000404 BR 35 ;GO DO MSB
3700 014052 006105 25: ROL R5 ;FORM THIS DIGIT
3701 014054 006105 ROL R5
3702 014056 006105 ROL R5
3703 014060 010503 MOV R5,R3
3704 014062 006103 35: ROL R3 ;GET LSB OF THIS DIGIT
3705 014064 105337 014166 DECB 0MODE ;TYPE THIS DIGIT?
3706 014070 100016 BPL 75 ;BR IF NO
3707 014072 042703 177770 BIC #177770,R3 ;GET RID OF JUNK
3708 014076 001002 BNE 45 ;TEST FOR 0
3709 014100 005704 TST R4 ;SUPPRESS THIS 0?
3710 014102 001403 BEQ 55 ;BR IF YES

```



```

3711 014104 005204          46: INC R4          ;DON'T SUPPRESS ANYMORE 0'S
3712 014106 052703 000060    51: BIS #'0,R3      ;MAKE THIS DIGIT ASCII
3713 014112 052703 000040    51: BIS #' ,R3      ;MAKE ASCII IF NOT ALREADY
3714 014116 110337 014162      MOV# R3,R6      ;SAVE FOR TYPING
3715 014122 104400 014162      TYPE ,R6        ;GO TYPE THIS DIGIT
3716 014126 105337 014164      75: DECB SOCNT      ;COUNT BY 1
3717 014132 003347            8GT 28          ;BR IF MORE TO DO
3718 014134 002402            INC R4          ;BR IF DONE
3719 014136 005204            BR 28          ;INSURE LAST DIGIT ISN'T A BLANK
3720 014140 000744            68: MOV (SP)+,R5      ;GO DO THE LAST DIGIT
3721 014142 012605            MOV (SP)+,R4      ;RESTORE R5
3722 014144 012604            MOV (SP)+,R3      ;RESTORE R4
3723 014146 012603            MOV 2(SP),R3      ;RESTORE R3
3724 014150 016666 000002 000004 MOV (SP)+,(SP) ;SET THE STACK FOR RETURNING
3725 014156 012616            RTI              ;RETURN
3726 014160 000002            86: .BYTE 0        ;STORAGE FOR ASCII DIGIT
3727 014162 000          .BYTE 0        ;TERMINATOR FOR TYPE ROUTINE
3728 014163 000          SOCNT: .BYTE 0    ;OCTAL DIGIT COUNTER
3729 014164 000          SUFFILL: .BYTE 0  ;ZERO FILL SWITCH
3730 014165 000          SOMODE: .WORD 0   ;NUMBER OF DIGITS TO TYPE
3731 014166 000000

```

```

3732          ,SBTTL SAVE AND RESTORE R0-R5 ROUTINES
3733
3734          ;*****
3735          ;*SAVE R0-R5
3736          ;*CALL:
3737          ;* SAVREG
3738          ;*UPON RETURN FROM SSAVREG THE STACK WILL LOOK LIKE:
3739          ;*
3740          ;*TOP---(+16)
3741          ;* +2---(+18)
3742          ;* +4---R5
3743          ;* +6---R4
3744          ;* +8---R3
3745          ;*+10---R2
3746          ;*+12---R1
3747          ;*+14---R0
3748
3749          SSAVREG:
3750 014170 010046      MOV R0,-(SP)      ;PUSH R0 ON STACK
3751 014172 010146      MOV R1,-(SP)      ;PUSH R1 ON STACK
3752 014174 010246      MOV R2,-(SP)      ;PUSH R2 ON STACK
3753 014176 010346      MOV R3,-(SP)      ;PUSH R3 ON STACK
3754 014200 010446      MOV R4,-(SP)      ;PUSH R4 ON STACK
3755 014202 010546      MOV R5,-(SP)      ;PUSH R5 ON STACK
3756 014204 016646 000022 MOV 22(SP),-(SP) ;SAVE PS OF MAIN FLOW
3757 014210 016646 000022 MOV 22(SP),-(SP) ;SAVE PC OF MAIN FLOW
3758 014214 016646 000022 MOV 22(SP),-(SP) ;SAVE PS OF CALL
3759 014220 016646 000022 MOV 22(SP),-(SP) ;SAVE PC OF CALL
3760 014224 000002      RTI
3761
3762          ;*RESTORE R0-R5
3763          ;*CALL:
3764          ;* RESREG
3765          ;*RESREG:
3766 014226 012666 000022 MOV (SP)+,22(SP) ;RESTORE PC OF CALL
3767 014232 012666 000022 MOV (SP)+,22(SP) ;RESTORE PS OF CALL
3768 014236 012666 000022 MOV (SP)+,22(SP) ;RESTORE PC OF MAIN FLOW
3769 014242 012666 000022 MOV (SP)+,22(SP) ;RESTORE PS OF MAIN FLOW
3770 014246 012605      MOV (SP)+,R5      ;POP STACK INTO R5
3771 014250 012604      MOV (SP)+,R4      ;POP STACK INTO R4
3772 014252 012603      MOV (SP)+,R3      ;POP STACK INTO R3
3773 014254 012602      MOV (SP)+,R2      ;POP STACK INTO R2
3774 014256 012601      MOV (SP)+,R1      ;POP STACK INTO R1
3775 014260 012600      MOV (SP)+,R0      ;POP STACK INTO R0
3776 014262 000002      RTI

```


MAINDEC-11-DZRXB-E MACY11 27(732) 16-MAR-76 15:55 PAGE 86
DZRXBE.P11 TTY INPUT ROUTINE

SEO 0111

```

3833 014472 004737 013656 16s: JSR PC,STYPEC ;,ECHO CHAR
3834 014476 021627 000060 CMP (SP),#60 ;,CHAR < 0?
3835 014502 002420 BLT 18s ;,BRANCH IF YES
3836 014504 021627 000067 CMP (SP),#67 ;,CHAR > 7?
3837 014510 003015 BGT 18s ;,BRANCH IF YES
3838 014512 042726 000060 BIC #60,(SP)+ ;,STRIP-OFF ASCII
3839 014516 005766 000002 TST 2(SP) ;,IS THIS THE FIRST CHAR
3840 014522 001403 BEQ 17s ;,BRANCH IF YES
3841 014524 006316 ASL (SP) ;,NO, SHIFT PRESENT
3842 014526 006316 ASL (SP) ;,CHAR OVER TO MAKE
3843 014530 006316 ASL (SP) ;,ROOM FOR NEW ONE.
3844 014532 025266 000002 17s: INC 2(SP) ;,KEEP COUNT OF CHAR
3845 014536 056616 177776 BIS ~2(SP),(SP) ;,SET IN NEW CHAR
3846 014542 000707 BR 7s ;,GET THE NEXT ONE
3847 014544 104400 013736 18s: TYPE $QUES ;,TYPE ?<CR><LF>
3848 014550 000720 BR 20s ;,SIMULATE CONTROL-U
3849
3850
3851
3852
3853 ;*****
3854 ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
3855 ;*CALL:
3856 ;* R0CHR ;,INPUT A SINGLE CHARACTER FROM THE TTY
3857 ;* RETURN HERE ;,CHARACTER IS ON THE STACK
3858 ;* ;,WITH PARITY BIT STRIPPED OFF
3859 ;
3860 014552 011646 SRDCHR: MOV (SP),-(SP) ;,PUSH DOWN THE PC
3861 014554 016666 000004 000002 MOV 4(SP),2(SP) ;,SAVE THE PS
3862 014562 105777 177476 1s: TSTB 0STKS ;,WAIT FOR
3863 014566 100375 BPL 1s ;,A CHARACTER
3864 014570 117766 177472 000004 MOVb 0STKB,4(SP) ;,READ THE TTY
3865 014576 042766 177600 000004 BIC #<'C'177>,4(SP) ;,GET RID OF JUNK IF ANY
3866 014604 026627 000004 000023 CMP 4(SP),#23 ;,IS IT A CONTROL-S?
3867 014612 001013 BNE 3s ;,BRANCH IF NO
3868 014614 105777 177444 2s: TSTB 0STKS ;,WAIT FOR A CHARACTER
3869 014620 100375 BPL 2s ;,LOOP UNTIL ITS THERE
3870 014622 117746 177440 MOVb 0STKB,-(SP) ;,GET CHARACTER
3871 014626 042716 177600 BIC #<'C'177>,(SP) ;,MAKE IT 7-BIT ASCII
3872 014632 022627 000021 CMP (SP)+,#21 ;,IS IT A CONTROL-Q?
3873 014636 001366 BNE 2s ;,IF NOT DISCARD IT
3874 014640 000750 BR 1s ;,YES, RESUME
3875 014642 026627 000004 000140 3s: CMP 4(SP),#140 ;,IS IT UPPER CASE?
3876 014650 002407 BLT 4s ;,BRANCH IF YES
3877 014652 026627 000004 000175 CMP 4(SP),#175 ;,IS IT A SPECIAL CHAR?
3878 014660 003003 BGT 4s ;,BRANCH IF YES
3879 014662 042766 000040 000004 BIC #40,4(SP) ;,MAKE IT UPPER CASE
3880 014670 000002 4s: RTI ;,GO BACK TO USER
3881 ;*****
3882 ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
3883 ;*CALL:
3884 ;* RDLIN ;,INPUT A STRING FROM THE TTY
3885 ;* RETURN HERE ;,ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
3886 ;* ;,TERMINATOR WILL BE A BYTE OF ALL 0's
3887 ;
3888 014672 010346 SRDLIN: MOV R3,-(SP) ;,SAVE R3

```



```

3889 014674 012703 015000 16: MOV #STTYIN,R3 ;GET ADDRESS
3890 014700 022703 015010 26: CMP #STTYIN+0,,P3 ;BUFFER FULL?
3891 014704 101405 BLOS 46 ;BR IF YES
3892 014706 104406 RDCHR ;GO READ ONE CHARACTER FROM THE TTY
3893 014710 112613 MOVB (SP)+,(R3) ;GET CHARACTER
3894 014712 122713 000177 106: CMPB #177,(R3) ;IS IT A RUBOUT
3895 014716 001003 BNE 38 ;SKIP IF NOT
3896 014720 104400 013736 46: TYPE ,SQUES ;TYPE A '?'
3897 014724 000763 BR 16 ;CLEAR THE BUFFER AND LOOP
3898 014726 111337 014776 36: MOVB (R3),96 ;ECHO THE CHARACTER
3899 014732 104400 014776 TYPE ,96
3900 014736 122723 000015 CMPB #15,(R3)+ ;CHECK FOR RETURN
3901 014742 001356 BNE 26 ;LOOP IF NOT RETURN
3902 014744 105063 177777 CLRB -1(R3) ;CLEAR RETURN (THE 15)
3903 014750 104400 013740 TYPE ,SLF ;TYPE A LINE FEED
3904 014754 012603 MOV (SP)+,R3 ;RESTORE R3
3905 014756 011646 MOV (SP),-(SP) ;ADJUST THE STACK AND PUT ADDRESS OF THE
3906 014760 016666 000004 000002 MOV 4(SP),2(SP) ; FIRST ASCII CHARACTER ON IT
3907 014766 012766 015000 000004 MOV #STTYIN,4(SP)
3908 014774 000002 RTI ;RETURN
3909 014776 000 96: .BYTE 0 ;STORAGE FOR ASCII CHAR. TO TYPE
3910 014777 000 .BYTE 0 ;TERMINATOR
3911 015000 000010 STTYIN: .BLKB 8 ;RESERVE 8 BYTES FOR TTY INPUT
3912 015010 052536 005015 000 SCNTLU: .ASCIZ /"U"<15><12> ;CONTROL "U"
3913 015015 136 006507 000012 SCNTLG: .ASCIZ /"G"<15><12> ;CONTROL "G"
3914 015022 005015 053523 020122 SMSWR: .ASCIZ <15><12>/SWR = /
3915 015030 020075 000 SMNEW: .ASCIZ / NEW = /
3916 015033 040 047040 053505 SAUTOB: .BYTE 0 ;AUTO MODE FLAG
3917 015040 036440 000040 SINTAG: .BYTE 0 ;INTERRUPT MODE FLAG
3918 015044 000
3919 015045 000

```

```

3920 .SBTTL TRAP DECODER
3921
3922 ;*****
3923 ;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3924 ;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3925 ;OF THE DESIRED ROUTINE, THEN USING THE ADDRESS OBTAINED IT WILL
3926 ;GO TO THAT ROUTINE.
3927
3928 015046 010046 STRAP: MOV R0,-(SP) ;SAVE R0
3929 015050 016600 000002 MOV 2(SP),R0 ;GET TRAP ADDRESS
3930 015054 005740 TST -(R0) ;BACKUP BY 2
3931 015056 111000 MOVB (R0),R0 ;GET RIGHT BYTE OF TRAP
3932 015060 006300 ASL R0 ;POSITION FOR INDEXING
3933 015062 016000 015070 MOV STRPAD(R0),R0 ;INDEX TO TABLE
3934 015066 000200 RTS R0 ;GO TO ROUTINE
3935
3936 .SBTTL TRAP TABLE
3937
3938 ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3939 ;*BY THE "TRAP" INSTRUCTION.
3940
3941 ; ROUTINE
3942 ; -----
3943 STRPAD:
3944 015070 STYPE ;CALL=TYPE TRAP+0(104400) TTY TYPEOUT ROUTINE
3945 015072 STYPOC ;CALL=TYPOC TRAP+1(104401) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3946 015074 STYPOS ;CALL=TYPOS TRAP+2(104402) TYPE OCTAL NUMBER (NO LEADING ZEROS)
3947 015076 STYPON ;CALL=TYPON TRAP+3(104403) TYPE OCTAL NUMBER (AS PER LAST CALL)
3948
3949 015100 $GTSWR ;CALL=GTSWR TRAP+4(104404) GET SOFT-SWR SETTING
3950
3951 015102 $CKSWR ;CALL=CKSWR TRAP+5(104405) TEST FOR CHANGE IN SOFT-SWR
3952 015104 $RDCHR ;CALL=RDCHR TRAP+6(104406) TTY TYPEIN CHARACTER ROUTINE
3953 015106 $RDLIN ;CALL=RDLIN TRAP+7(104407) TTY TYPEIN STRING ROUTINE
3954 015110 $SAVREG ;CALL=SAVREG TRAP+10(104410) SAVE R0-R5 ROUTINE
3955 015112 $RESREG ;CALL=RESREG TRAP+11(104411) RESTORE R0-R5 ROUTINE
3956 015114 $SUBSCOPE ;CALL=SUBSCOPE TRAP+12(104412)

```



```

3957 ,SBTTL POWER DOWN AND UP ROUTINES
3958
3959 ;*****
3960 ;POWER DOWN ROUTINE
3961 015116 012737 015262 000024 SPWRDN: MOV #ILLUP,0#PWRVEC ;;SET FOR FAST UP
3962 015124 012737 000340 000026 MOV #340,0#PWRVEC+2 ;;PRIO:7
3963 015132 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
3964 015134 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
3965 015136 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
3966 015140 010346 MOV R3,-(SP) ;;PUSH R3 ON STACK
3967 015142 010446 MOV R4,-(SP) ;;PUSH R4 ON STACK
3968 015144 010546 MOV R5,-(SP) ;;PUSH R5 ON STACK
3969 015146 017746 164044 MOV @SWR,-(SP) ;;PUSH @SWR ON STACK
3970 015152 010637 015266 MOV SP,$SAVR6 ;;SAVE SP
3971 015156 012737 015170 000024 MOV $SPWRUP,0#PWRVEC ;;SET UP VECTOR
3972 015164 000000 HALT
3973 015166 000776 BR ,-2 ;;HANG UP
3974
3975 ;*****
3976 ;POWER UP ROUTINE
3977 015170 012737 015262 000024 SPWRUP: MOV #ILLUP,0#PWRVEC ;;SET FOR FAST DOWN
3978 015176 013706 015266 MOV $SAVR6,SP ;;GET SP
3979 015202 005037 015266 CLR $SAVR6 ;;WAIT LOOP FOR THE ITT
3980 015206 005237 015266 1S: INC $SAVR6 ;;WAIT FOR THE INC
3981 015212 001375 MOV 1S ;;OF WORD
3982 015214 012677 163776 MOV (SP)+,@SWR ;;POP STACK INTO @SWR
3983 015220 012605 MOV (SP)+,R5 ;;POP STACK INTO R5
3984 015222 012604 MOV (SP)+,R4 ;;POP STACK INTO R4
3985 015224 012603 MOV (SP)+,R3 ;;POP STACK INTO R3
3986 015226 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
3987 015230 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
3988 015232 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
3989 015234 012737 015116 000024 MOV $SPWRDN,0#PWRVEC ;;SET UP THE POWER DOWN VECTOR
3990 015242 012737 000340 000026 MOV #340,0#PWRVEC+2 ;;PRIO:7
3991 015250 104400 TYPE REPORT THE POWER FAILURE
3992 015252 015270 SPWRMG: ,WORD $POWER ;;POWER FAIL MESSAGE POINTER
3993 015254 012716 MOV (PC)+,(SP) ;;RESTART AT RESTART
3994 015256 001350 SPWRAD: ,WORD RESTART ;;RESTART ADDRESS
3995 015260 000002 RTI
3996 015262 000000 SILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
3997 015264 000776 BR ,-2 ;;BEFORE THE POWER DOWN WAS COMPLETE
3998 015266 000000 $SAVR6: 0 ;;PUT THE SP HERE
3999 015270 005015 047520 042527 $POWER: ,ASCIZ <15><12>"POWER"
4000 015276 000122 ,EVEN
4001

```

```

4002 ,SBTTL SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
4003
4004 ;*****
4005 ;THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
4006 ;UNSIGNED DECIMAL ASCIZ NUMBER.
4007 ;CALL
4008 ;* MOV NUMBER,-(SP) ;;PUT BINARY NUMBER ON THE STACK
4009 ;* JSR PC,@$SR2D ;;CALL
4010 ;* RETURN ;;ADDRESS OF THE 1ST ASCIZ CHAR,IS ON THE STACK
4011
4012
4013 015300 016637 000002 015330 $SR2D: MOV 2(SP),1S ;;SAVE BINARY NUMBER
4014 015306 012746 015330 MOV #1S,-(SP) ;;SET POINTER
4015 015312 004737 015334 JSR PC,@$SDH2D ;;CALL DOUBLE LENGTH CONVERT
4016 015316 002716 000005 ADD #5,(SP) ;;ONLY ALLOW FIVE CHARACTERS
4017 015322 012606 000002 MOV (SP)+,2(SP) ;;PICKUP POINTER
4018 015326 000207 RTS PC ;;RETURN
4019 015330 000000 000000 1S: ,WORD 0,0
4020 ,SBTTL DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
4021
4022 ;*****
4023 ;THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
4024 ;DECIMAL (ASCII) NUMBER, THE SIGN OF THE BINARY NUMBER MUST BE
4025 ;POSITIVE.
4026 ;CALL
4027 ;* MOV #PNTR,-(SP) ;;POINTER TO LOW WORD OF BINARY NUMBER
4028 ;* JSR PC,@$SDH2D ;;CALL
4029 ;* RETURN ;;THE FIRST ADDRESS OF ASCIZ
4030 ;;IS ON THE STACK
4031
4032
4033 015334 104410 $DR2D: SAVREG ;;SAVE REGISTERS
4034 015336 016602 000002 MOV 2(SP),R2 ;;PICKUP THE DATA POINTER
4035 015342 012700 015514 MOV $DECVL,R0 ;;GET ADDRESS OF "$DECVL" STRING
4036 015346 010066 000002 MOV R0,2(SP) ;;PUT ADDRESS OF ASCII STRING ON STACK
4037 015352 012201 MOV (R2)+,R1 ;;PICKUP THE BINARY NUMBER
4038 015354 012202 MOV (R2)+,R2
4039 015356 012737 000012 015432 MOV #10,,4S ;;SET UP TO DO 10 CONVERSIONS
4040 015364 012704 015444 MOV $TNPWR,R4 ;;ADDRESS OF TEN POWER
4041 015370 012705 015446 MOV $TNPWR+2,R5
4042 015374 005003 1S: CLR R3 ;;CLEAR PARTIAL
4043 015376 161401 2S: SUB (R4),R1 ;;SUBTRACT TEN POWER
4044 015400 005602 SBC R2
4045 015402 161502 SUB (R5),R2
4046 015404 002402 BLT 3S ;;BR IF TEN POWER TO LARGE
4047 015406 005203 INC R3 ;;ADD 1 TO PARTIAL
4048 015410 000772 BR 2S ;;LOOP
4049 015412 062401 3S: ADD (R4)+,R1 ;;RESTORE SUBTRACTED VALUE
4050 015414 005502 ADC R2
4051 015416 062402 ADD (R4)+,R2
4052 015420 022525 CMP (R5)+,(R5)+ ;;MOVE TO NEXT TEN POWER
4053 015422 052703 000060 BIS #0,R3 ;;CHANGE PARTIAL TO ASCII
4054 015426 110320 MOVB R3,(R0)+ ;;SAVE IT
4055 015430 005327 DEC (PC)+ ;;DONE?
4056 015432 000000 4S: ,WORD 0
4057 015434 001357 BNE 1S ;;BR IF NO

```


4058	015436	105020	CLPR	(R0)+	;;TERMINATOR
4059	015440	104411	RESREG		;;RESTORE REGISTERS
4060	015442	000207	RTS	PC	;;RETURN
4061	015444	145000	STNPWR:	145000	;;1,0E09
4062	015446	035632		35632	
4063	015450	160400		160400	;;1,0E08
4064	015452	002765		2765	
4065	015454	113200		113200	;;1,0E07
4066	015456	000230		230	
4067	015460	041100		041100	;;1,0E06
4068	015462	000017		17	
4069	015464	103240		103240	;;1,0E05
4070	015466	000001		1	
4071	015470	023420		23420	;;1,0E04
4072	015472	000000		0	
4073	015474	001750		1750	;;1,0E03
4074	015476	000000		0	
4075	015500	000144		144	;;1,0E02
4076	015502	000000		0	
4077	015504	000012		12	;;1,0E01
4078	015506	000000		0	
4079	015510	000001		1	;;1,0E00
4080	015512	000000		0	
4081	015514	000014	SDECVL:	,BLKB 12,	;;RESERVE STORAGE FOR ASCII STRING
4082					

4083			;	*****	
4084					
4085					
4086					
4087					
4088					
4089					
4090	015530	010046	RTJUST:	MOV R0,-(SP)	;;SAVE R0
4091	015532	016600		MOV 4(SP),R0	;;PICK UP ADDRESS OF ASCII STRING
4092	015536	010037		MOV R0,38	;;SAVE ADDRESS FOR TYPE OUT
4093	015542	105710	1S:	TSTB (R0)	;;IS THIS THE TERMINATOR
4094	015544	001406		BEO 2S	;;IF YES TYPE IT OUT
4095	015546	122710		CMPE #'0,(R0)	;;IS IT A ZERO
4096	015552	001005		BNE 4S	;;IF NO GO PRINT IT
4097	015554	112720		MOVB #'',(R0)+	;;IF YES REPLACE IT WITH A SPACE
4098	015560	000770		BR 1S	;;TEST NEXT CHAR,
4099	015562	112740	2S:	MOVB #'0,-(R0)	;;STRING OFF ALL ZEROS,PUT BACK THE LAST ONE
4100	015566	104400	4S:	TYPE	;;TYPE THE STRING
4101	015570	000000	3S:	OPEN	
4102	015572	012600		MOV (SP)+,R0	;;RESTORE R0
4103	015574	012616		MOV (SP)+,(SP)	;;RESTORE THE STACK
4104	015576	000207		RTS PC	;;RETURN
4105					
4106					
4107					
4108	015600	012546			
4109	015602	004737	SGLDEC:	MOV (R5)+,-(SP)	;;PUT NUMBER TO BE TYPED ON STACK
4110	015606	004737		JSR PC,@#SB2D	;;CONVERT NUMBER TO DECIMAL
4111	015612	000205		JSR PC,RTJUST	;;TYPE THE DECIMAL NUMBER
				RTS R5	


```

4112                                     ,SBTTL MESSAGES
4113
4114
4115 015614 042524 052123 050040 MDTESTP:      ,ASCIZ "TEST PARAMETERS: "
4116 015622 051101 046501 052105
4117 015630 051105 035123 000040
4118
4119 015636 005015 042412 040522 MXEHEADER:    ,ASCIZ <15><12><12> "ERADR FAST FAPT      GOOD  BAD  PASS"
4120 015644 051104 020040 040506
4121 015652 052123 020040 043040
4122 015660 050101 020124 020040
4123 015666 020040 020040 020040
4124 015674 043440 047517 020104
4125 015702 020040 040502 020104
4126 015710 020040 020040 040520
4127 015716 051523 0000
4128 015721 125 042516 050130 MUNXDD:      ,ASCIZ "UNEXPECTED D D MARK"<15><12>
4129 015726 041505 042524 020104
4130 015734 020104 020104 040515
4131 015742 045522 005015 0000
4132
4133 015747 104 042040 046440 MDDMIS:      ,ASCIZ "D D MARK MISSING"<15><12>
4134 015754 051101 020113 044515
4135 015762 051523 047111 006507
4136 015770 000012
4137
4138
4139 015772 040504 040524 020054 MDERHDR:      ,ASCIZ "DATA, NO STATUS ERROR"
4140 016000 047516 051440 040524
4141 016006 052524 020123 051105
4142 016014 047522 000122
4143
4144 016020 047440 020116 051124 MTPK:          ,ASCIZ " ON TRACK"
4145 016026 041501 000113
4146
4147 016032 020040 051106 046517 MPRES:          ,ASCIZ " FROM TRACK"
4148 016040 052040 040522 045503
4149 016046 0000
4150
4151 016047 040 020057 042523 MSECT:          ,ASCIZ " / SECTOR"
4152 016054 052103 051117 0000
4153
4154 016061 015 020012 054502 MCOLMUN:        ,ASCIZ <15><12>" BYTE BAD GOOD"<15><12>
4155 016066 042524 020040 040502
4156 016074 020104 043440 047517
4157 016102 006504 000012
4158
4159 016106 020040 020040 020040 TAR:          ,ASCIZ <40><40><40><40><40><40>
4160 016114 0000
4161
4162 016115 040 000040 DRLSP:          ,ASCIZ <40><40>
4163
4164 016120 005015 0000 MCRLF:          ,ASCIZ <15><12>
4165
4166 016123 015 042412 051122 MERHEADER:      ,ASCIZ <15><12>"ERROR CONDITIONS; TEST PC = "
4167 016130 051117 041440 047117

```

```

4168 016136 044504 044524 047117
4169 016144 035123 020040 042524
4170 016152 052123 050040 020103
4171 016160 020075 0000
4172
4173 016163 125 044516 020124 MUNIT0:      ,ASCIZ "UNIT 0 "
4174 016170 020060 0000
4175
4176 016173 125 044516 020124 MUNIT1:      ,ASCIZ "UNIT 1 "
4177 016200 020061 0000
4178
4179 016203 117 046116 020131 MONLY:      ,ASCIZ "ONLY "
4180 016210 000040
4181
4182 016212 054122 051503 036440 MPXCS:      ,ASCIZ "RXCS = "
4183 016220 000040
4184
4185 016222 052123 052101 051525 MASTAT:      ,ASCIZ "STATUS A = "
4186 016230 040440 036440 000040
4187
4188 016236 052123 052101 051525 MRSTAT:      ,ASCIZ "STATUS B = "
4189 016244 041040 036440 000040
4190
4191 016252 005015 000012 DBLLF:          ,ASCIZ <15><12><12>
4192
4193 016256 047516 044440 052116 MINTER:      ,ASCIZ "NO INTERRUPT AT DONE ERROR"
4194 016264 051105 052522 052120
4195 016272 040440 020124 047504
4196 016300 042516 042440 051122
4197 016306 051117 0000
4198
4199 016311 125 045516 047516 MUKNINT:      ,ASCIZ "UNKNOWN INTERRUPT"
4200 016316 047127 044440 052116
4201 016324 051105 052522 052120
4202 016332 0000
4203
4204 016333 124 052117 046101 MERCT:          ,ASCIZ "TOTAL READ CHECK ERRORS = "
4205 016340 051040 040505 020104
4206 016346 044103 041505 020113
4207 016354 051105 047522 051522
4208 016362 036440 000040
4209
4210 016366 044506 046114 052502 MFIL:          ,ASCIZ "FILLBUFFER "
4211 016374 043106 051105 000040
4212
4213 016402 046505 052120 041131 MEMPTY:      ,ASCIZ "EMPTYBUFFER "
4214 016410 043125 042506 020122
4215 016416 0000
4216
4217 016417 040 051124 041501 MLIMTRK:      ,ASCIZ " TRACKS 52,53,114,0 "
4218 016424 051513 032440 026062
4219 016432 031465 030454 032061
4220 016440 030054 020040 0000
4221
4222 016445 117 036504 0000 MOD:          ,ASCIZ "OD="
4223

```



```

4224 016451 040 044440 036504 MID: ,ASCIZ " ID="
4225 016456 000
4226
4227 016457 040 043040 051111 MFIRST: ,ASCIZ " FIRST="
4228 016464 052123 000075
4229
4230 016470 020040 040514 052123 MLAST: ,ASCIZ " LAST="
4231 016476 000075
4232
4233 016500 051103 020103 051105 MBADCRC: ,ASCIZ "CRC ERROR NO DATA ERROR"
4234 016506 047522 020122 047516
4235 016514 042040 052101 020101
4236 016522 051105 047522 000122
4237
4238 016530 042522 042101 000040 MREAD: ,ASCIZ "READ "
4239
4240 016536 040504 040524 041440 MCRC: ,ASCIZ "DATA CRC ERROR"
4241 016544 041522 042440 051122
4242 016552 051117 000
4243
4244 016555 123 042505 020113 MSEEK: ,ASCIZ "SEEK ERROR"
4245 016562 051105 047522 000122
4246
4247 016570 051127 052111 020105 MWRITE: ,ASCIZ "WRITE "
4248 016576 000
4249
4250 016577 120 051101 052111 MPAR: ,ASCIZ "PARITY ERROR"
4251 016604 020131 051105 047522
4252 016612 000122
4253
4254 016614 051105 047522 020122 MNOFLAG: ,ASCIZ "ERROR FLAG ERROR"
4255 016622 046106 043501 042440
4256 016630 051122 051117 000
4257
4258 016635 102 042101 000 MBAD: ,ASCIZ "BAD"
4259
4260 016641 040 000 SPACE: ,ASCIZ "<40>"
4261
4262 016643 107 047517 000104 MGOOD: ,ASCIZ "GOOD"
4263
4264 016650 020040 044103 041505 MSUM: ,ASCIZ " CHECK SUM "
4265 016656 020113 052523 020115
4266 016664 000
4267
4268 016665 015 051012 030530 MRX11: ,ASCIZ "<15><12>"RX11 / RXV11"
4269 016672 020061 020057 054122
4270 016700 030526 000061
4271
4272 016704 005015 046412 044501 MKEV: ,ASCIZ "<15><12><12>"MAINDEC-11-DZRXB-E" <15><12>"
4273 016712 042116 041505 030455
4274 016720 026461 055104 054122
4275 016726 026502 006505 000012
4276
4277 016734 005015 047125 054105 LOC4M: ,ASCIZ "<15><12>"UNEXPECTED TRAP TO LOC. 4 OCCURRED"
4278 016742 042520 052103 042105
4279 016750 052040 040522 020120

```

```

4280 016756 047524 046040 041517
4281 016764 020056 020064 041517
4282 016772 052503 051122 042105
4283 017000 000
4284
4285 017001 015 052412 042516 LOC10M: ,ASCIZ "<15><12>"UNEXPECTED TRAP TO LOC. 10 OCCURRED"
4286 017006 050130 041505 042524
4287 017014 020104 051124 050101
4288 017022 052040 020117 047514
4289 017030 027103 030440 020060
4290 017036 041517 052503 051122
4291 017044 042105 000
4292
4293 017047 075 041520 000 PCM: ,ASCIZ "=PC"
4294
4295 017053 015 052012 040522 OD2BIG: ,ASCII "<15><12>"TRACK LIMITS SELECTED OUT OF RANGE"
4296 017060 045503 046040 046511
4297 017066 052111 020123 042523
4298 017074 042514 052103 042105
4299 017102 047440 052125 047440
4300 017110 020106 040522 043516
4301 017116 105
4302 017117 015 042012 043105 ,ASCIZ "<15><12>"DEFAULTING TO "
4303 017124 052501 052114 047111
4304 017132 020107 047524 000040
4305
4306 017140 005015 042523 052103 S2BIG: ,ASCII "<15><12>"SECTOR LIMITS SELECTED OUT OF RANGE"
4307 017146 051117 046040 046511
4308 017154 052111 020123 042523
4309 017162 042514 052103 042105
4310 017170 047440 052125 047440
4311 017176 020106 040522 043516
4312 017204 105
4313 017205 015 042012 043105 ,ASCIZ "<15><12>"DEFAULTING TO "
4314 017212 052501 052114 047111
4315 017220 020107 047524 000040
4316
4317 017226 005015 040503 052125 D0LOAD: ,ASCII "<15><12>"CAUTION - IF YOU DESIRE TO TEST UNIT 0"
4318 017234 047511 020116 020055
4319 017242 043111 054440 052517
4320 017250 042040 051505 051111
4321 017256 020105 047524 052040
4322 017264 051505 020124 047125
4323 017272 052111 030040
4324 017276 005015 042522 046120 ,ASCII "<15><12>"REPLACE LOAD MEDIUM WITH A SCRATCH DISKETTE"
4325 017304 041501 020105 047514
4326 017312 042101 046440 042105
4327 017320 052511 020115 044527
4328 017326 044124 040440 051440
4329 017334 051103 052101 044103
4330 017342 042040 051511 042513
4331 017350 052124 105
4332 017353 015 052012 042510 ,ASCIZ "<15><12>"THEN PRESS CONTINUE"<15><12>"
4333 017360 020116 051120 051505
4334 017366 020123 047503 052116
4335 017374 047111 042525 005015

```



```

      .EVEN
;*****
      ;THE FOLLOWING LOCATIONS ARE USED FOR DATA STORAGE,RETRY COUNTERS
      ;ACCESS COUNTERS ETC,
BUFADR:      .BLKB 200
      .END

```

[illegible]

CUPPRI	006200		1334	1393	2491*														
CR	= 000015		94#	3636	3646														
CR CER	010234		2924	2932#															
CR LF	= 000200		95#	3607	3646														
DATABY	012322		3291#	3300#	3310*	3316*	3326*	3334*	3343*	3345*	3354*	3362*	3363*	3372*	3376*				
			3377																
DATA CK	007704		2861#	2890*	2934*	3104													
DATA CR	010330		2933	2937	2954*														
DATA ER	011110		3043	3067#															
DATA O	012326		3281	3300#															
DATA I	012340		3282	3310*															
DB L F	016252		4191#																
DB L SP	016115		3087	3091	3232	4162#													
DD CH K	010410		2907	2970#															
DD ERR	010452		2979#	2998															
DD IS P	= 177570		101#	378															
DEATH	003034		1174	1184*															
DE XIT	004260		1014#	1896	2030	2084	2136	2271	2308	2331	2399	2415	2428	2454					
DING	006450		2561	2571#	3012														
DIS PL A	001220		378#	419*															
DIS PRE	000174		237#	419															
DONE	012726		2743	3444*															
DONE BI =	000040		201#	2796	2899	3166													
DRVS WP	007012		2427	2717#															
DS WR	= 177570		100#	377															
DTESTP	001212		368#	445	449	452	473	867											
D0 LOAD	017226		464	4317#															
EBIE	= 000103		203#	3036															
EMP BU F	010644		2935	3023*	3063	3127	3131												
EMPD ON	011306		3031	3104*															
EMPER	011060		3032	3061*															
EMPF LA	010730		3037#	3038	3045	3102													
EMTVEC=	000030		189#																
EPCSCO	006474		2517#	2518*	2581#	2613													
ERCNTR	011460		2936	3027*	3068*	3074	3107	3113	3138#										
EPMSG	010566		2977	2996	3002*	3078	3147	3196											
ERRORS	006472		549*	2519*	2580#														

[illegible]

RXDB	001210	348#	434*	435*	668	973	1486	1590	1628*	1643	1734	1829	1848	1954
		1975	1985	2063	2077	2115	2129	2147*	2155*	2182	2184*	2191	2193*	2237
		2255	2263	2326*	2337	2341	2760*	2854*	2857*	3040	3158	3187	3191	3245
		3249	3253											
RXERRO	011652	3160	3178	3181#	3182									
RXPWR	012006	3163	3208#											
R0	%000000	104#	402*	408*	442	552*	554	667*	674*	675	918*	920	972*	974
		1195*	1196	1247*	1248	1337	1397	1434*	1436	1485*	1492*	1493	1542*	1550*
		1552	1593*	1597*	1598	1733*	1736	1828*	1832	1847*	1851	1907*	1915*	1917
		1953*	1957	1973*	1984*	1987	2039*	2048*	2050	2062*	2064	2076*	2078	2092*
		2101*	2103	2114*	2116	2128*	2130	2174*	2196*	2198	2235*	2238	2253*	2262*
		2264	2326	2376*	2491*	2496*	2498*	2499*	2500*	2501*	2502*	2503*	2549	2760
		3042	3056*	3092	3093*	3386*	3387*	3388*	3389*	3391*	3392*	3393	3394*	3395*
		3396*	3397*	3398*	3399*	3400	3401	3597	3598*	3599	3602*	3750	3775*	3928
		3929*	3930	3931*	3932*	3933*	3934*	3963	3988*	4035*	4036	4054*	4058*	4090
		4091*	4092	4093	4095	4097*	4099*	4102*						
R1	%000001	105#	553*	554	669*	670*	673*	675	919*	920	973*	974	1198*	1250*
		1316*	1372*	1382*	1435*	1436	1487*	1488*	1491*	1493	1541*	1551*	1552	1591*
		1592*	1596*	1598	1643*	1734	1736	1830*	1831*	1832	1849*	1850*	1851	1916*
		1917	1955*	1956*	1957	1974*	1985*	1987	2049*	2050	2063*	2064	2077*	2078
		2102*	2103	2115*	2116	2129*	2130	2165*	2181*	2190*	2197*	2198	2237*	2238
		2254*	2263	2264	2337*	2377*	2493	2552	2508	2932	3021*	3028*	3076	3083*
		3132*	3751	3774*	3964	3987*	4037*	4043*	4049*					
R2	%000002	106#	668*	669	1018*	1019	1021*	1022*	1023*	1072*	1073	1075*	1076*	1077*
		1236*	1237	1239*	1240*	1241*	1277*	1278	1280*	1281*	1282*	1319*	1320	1322*
		1323*	1324*	1375*	1376	1378*	1379*	1380*	1408*	1487	1590*	1591	1621*	1629*
		1676	1725*	1735*	1829*	1830	1848*	1849	1906*	1954*	1955	1975*	1986*	2040*
		2061*	2093*	2113*	2142*	2150*	2158*	2173*	2182*	2191*	2236*	2255*	2261*	2304*
		2305*	2306*	2322*	2327*	2341*	2424*	2425*	2426*	2449*	2450*	2451*	2546	3318*
		3320*	3323	3752	3773*	3965	3986*	4034*	4037	4038*	4044*	4045*	4050*	4051*
R3	%000003	107#	756	757	760	867*	868*	869	871*	3688	3697*	3703*	3704*	3707*
		3712*	3713*	3714	3723*	3753	3772*	3888	3889*	3890	3893*	3894	3898	3900
		3902*	3904*	3966	3985*	4042*	4047*	4053*	4054					
		108#	1624*	1628	1727*	1733	2533*	2534*	2535	3276*	3377*	3378	3689	3691*
		3692*	3693*	3694	3695*	3709	3711*	3719*	3722*	3754	3771*	3967	3984*	4040*
		4043	4049	4051										
R5	%000005	109#	852*	857*	860	2556*	3009*	3085*	3114*	3278*	3279*	3280	3690	3696*
		3698*	3700*	3701*	3702*	3703	3721*	3755	3770*	3968	3983*	4041*	4045	4052
		4108	4111*											
R6	%000006	110#	112											
R7	%000007	111#	113											
SAVREG=	104410	3954#	4033											
SA200	001232	244	408*	875										
SA202	001222	245	402*											
SCOPIN	006536	2567	2614	2621#										
SDN	006574	1432	1548	1639	1826	1845	1969	2074	2126	2144	2152	2161	2179	2188
		2194	2249	2333	2339	2396	2636#	2867	3185	3247				
SECCNT	013414	2801*	2911*	3023	3128*	3551*	3552*	3553*	3554*	3561#				
SECLMT	001726	482	500	509*										
SEEKER	007526	2822	2825*	2928										
SEKRTY	007602	2816*	2837	2839*	2926*									
SEKTYP	007606	2829	2841*											
SEQUEN	013136	480*	2447*	2453*	3461	3466*	3486	3496*						
SEQ1	013236	3488	3523*											
SEQ2	013272	3489	3536*											
SGLDEC	015600	852	2556	3009	3085	3114	4108*							

	SP	%000006	112*	403*	410*	412*	417	429	431*	432*	473*	491*	496*	517*	522*
			759*	900*	1019*	1023	1025*	1026*	1030*	1031*	1073*	1077	1079*	1080*	1125*
			1126*	1237*	1241	1258*	1253*	1263*	1264*	1278*	1282	1284*	1285*	1290*	1291*
			1320*	1324	1326*	1330	1331	1333	1337*	1338*	1344*	1345*	1376*	1380	1385*
			1389	1390	1392	1397*	1398*	1404*	1405*	1636*	2163*	2461*	2467*	2478*	2464*
			2517	2533	2535*	2540*	2543*	2546*	2549*	2552*	2596	2613	2624*	2631*	2678*
			2695*	2751*	2752*	2754*	2755*	2764	2780*	2787*	2793*	2794*	2806	2843*	2892*
			2893*	2896*	2897*	2915	2965*	3003*	3031*	3032*	3033*	3034*	3061	3088*	3074*
			3174*	3198*	3205*	3212*	3221*	3228*	3255*	3261*	3381	3448*	3597*	3598	3599*
			3601	3602	3603*	3605	3607	3609	3615	3617*	3619*	3627*	3631	3635	3636
			3640	3680*	3681	3682	3683*	3688*	3689*	3690*	3696	3721	3722	3723	3724*
			3725*	3750*	3751*	3752*	3753*	3754*	3755*	3756*	3757*	3758*	3759*	3766*	3767*
			3768*	3769*	3770	3771	3772	3773	3774	3775	3793*	3794*	3795	3802*	3805*
			3806*	3810*	3811*	3815	3818*	3822	3824	3826	3827*	3834	3836	3838*	3839
			3841*	3842*	3843*	3844*	3845*	3860*	3861*	3862*	3865*	3866	3670*	3871*	3872
			3875	3877	3879*	3888*	3893	3904	3905*	3906*	3907*	3928*	3929	3963*	3964*
			3965*	3966*	3967*	3968*	3969*	3970	3978*	3982	3983	3984	3985	3986	3987
			3988	3993*	4013	4014*	4016*	4017*	4034	4036*	4090*	4091	4102	4103*	4108*
	SPACE	016641	2539	2542	2545	2548	2551	2554	4260*						
	STACK =	001200	87#	297	403	410	759								
	STKLMT=	177774	98#												
	STR	006560	1634	2140*	2148	2156	2176	2185	2629*	2852	2855				
	STYP1	007540	2821*	2828*	2927*										
	SUBSC0=	104412	957	1015	1189	1228	1255	1273	1470	1532	1547	1585	1672	1838	1948
	SUM	012536	1963	1980	2056	2069	2109	2121	2230	2244	2260	2375	3956*		
	SWHLT1	007550	3049	3277*	3376*	3384*									
	SWR	001216	2827	2830*											
			377#	409*	414	418*	423	847	892	898	2525	2566	2572	2622	2671
			2771	2777	2780	2826	2831	2836	2864	2872	2876	2879	2939	2944	2947
			2955	2960*	2963	2975	2983	2986	2994	3070	3072	3098	3110	3124	3145
			3151	3169	3177	3193	3789	3826*	3969	3982*					
	SWREG	000176	238#	418	423	3789	3802								
	SW0	= 000001	151#	2572											
	SW00	= 000001	141#	151											
	SW01	= 000002	140#	150											
	SW02	= 000004	139#	149											
	SW03	= 000010	138#	148											
	SW04	= 000020	137#	147											
	SW05	= 000040	136#	146											
	SW06	= 000100	135#	145											
	SW07	= 000200	134#	144											
	SW08	= 000400	133#	143											
	SW09	= 001000	132#	142											
	SW1	= 000002	150#												
	SW10	= 002000	131#	892											
	SW11	= 004000	130#	2622	2780	2836	2879	2947	2963	2986	3124				
	SW12	= 010000	129#	898											
	SW13	= 020000	128#	2525	2671	2771	2826	2939	2955	2975	2994	3070	3110	3145	3109
			3177	3193											
	SW14	= 040000	127#	847											
	SW15	= 100000	126#	2566											
	SW2	= 000004	149#												
	SW3	= 000010	148#												
	SW4	= 000020	147#												
	SW5	= 000040	146#												
	SW6	= 000100	145#												

MAINDEC-11-DZRXB-E MACY11 27(732) 16-MAR-76 15:55 PAGE 107
DZRXBE,P11 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0131

T6FILL	004026	1616	1621*	1808	2391	2408											
T7	004142	761	1723*														
T7EMPT	004152	1723	1725*	1809	2378	2398											
UNITSE	012724	444*	447*	451*	454*	460*	470*	548*	671	832	866*	902*	1489	1594			
		1823	1842	1908	2041	2094	2388	2528	2560*	2850	2979*	2992*	3067*	3103*			
		3233	3423	4425	3427*	3478*	3430	3432	3434*	3435*	3440#	3444	3446	3572			
		2721	2724	2739	2748*												
WRITE	007134	2786	2801*	2811	2823												
WRTDON	007430	2787	2806*														
WPTER	007444	204*	2788														
WPTIE =	000105	2808	2816*														
WRTSEK	007504	206*	2791														
TDDIF=	000115	2452	2735*														
WTROCK	007066	224	2517*														
XERROR	006232	2760*															
XFRBIT	007210	2870	2882*														
XHOME	010004	3468*	3469*	3472	3495*	3524											
XID	013134	3470*	3471*	3473	3494*												
XOD	013132	3317*	3335														
XPATGE	012356	2889*	2912	3130													
XREAD	010020	443	457	463	469	471*											
XSA202	001574	220	2586*														
XSCOPE	006476	2637	2693*														
XSDN	006736	2601*	3956														
XSUBSC	006516	2749*	2766	2804													
XWRITE	007140	2736*	2745														
XWTRDC	007072	3344*	3346	3355													
XXPATG	012426	427*	3797	3916*													
XAUTOB	015044	3612*	3622*	3629	3638*	3643*											
XSHARC	013722	3789*	3951														
XCKSWR	014270	3800	3913*														
XCNTLC	015015	3817	3912*														
XCNTLU	015010	3611	3653*	3828	3912												
XCRLF	013737	4015	4033*														
XDB2D	015334	4035	4081*														
XDECVL	015514	3615	3650*														
FILLC	013734	3649*															
FILLS	013733	3801*	3949														
GTSWR	014340	21	22														
SHD =	000003	3961	3977	3996*			</										

MAINDEC-11-DZRXB-E MACY11 27(732) 16-MAR-76 15:55 PAGE 110
DZRXB-E-P11 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0133

COMMENT	6*	194#	564	685	927	981	1041	1085	1133	1204	1443	1502	1559	1649	1683
ENDCOM	1747	1858	1924	1994	2207	2283	2347	2648							
	7*	194#	649	750	953	1011	1060	1120	1163	1224	1466	1528	1581	1668	1702
ERROR	1791	1891	1944	2025	2226	2300	2371	2663							
	8#	562	682	925	979	1039	1131	1202	1254	1272	1296	1351	1414	1441	1500
	1546	1557	1605	1647	1681	1745	1837	1856	1922	1962	1979	1992	2055	2068	2082
	2108	2120	2134	2205	2243	2259	2269	2345	2647						
ESCAPE	194#														
GETPRI	194#														
GETSWR	7*														
MULT	194#	194#	420												
M7727	248#	618	731	1447	1772	1862	1998	2351							
M7846	246#	563	684	926	980	1040	1084	1132	1203	1501	1558	1648	1682	1746	1923
	2206														
NEWTST	194#														
POP	194#														
PUSH	194#	3770	3982	3983											
REPORT	194#	3750	3963	3969											
SCOPE	8#	550	1064	1300	1358	1419	1606	1706	1795	1895	2029	2083	2135	2270	
SETPRI	6*	194#	1337	1344	1397	1404									
SETTRA	3936#	3945	3946	3947	3949	3951	3952	3953	3954	3955	3956				
SETUP	194#														
SKIP	194#														
SLASH	194#														
SPACE	194#														
STARS	6*	194#	251	252	253	264	265	266	271	272	273	292	293	294	305
	396	2509	2510	2514	2515	2712	2730	3140	3272	3295	3304	3329	3337	3348	3357
	3366	3421	3483	3498	3518	3531	3578	3657	3734	3779	3784	3852	3881	3922	3959
	3975	4004	4022	4083	4340										
SWRSU	194#														
TRMTRP	3936#														
TYPBIN	194#														
TYPDEC	194#														
TYPNAM	194#														
TYPNUM	194#														
TYPPCS	5*	194#	473	2461	2478	2843	3003	3198	3212	3221	3228	3255			
TYP OCT	194#	3802													
TYPTXT	194#														
\$\$ESCA	194#														
\$\$NEWIT	194#														
\$\$SET	3936#	3945	3946	3947	3949	3951	3								

ADC	557	2639	4050														
ADD	435	556	900	1636	2163	2518	2631	2638	2695	2965	3041	3050	3051	3174	3205		
ASL	3376	3387	3388	3398	3448	3566	3570	3603	3683	3693	3818	3827	4016	4049	4051		
ASLB	871	2498	2499	2500	2501	2502	3279	3841	3842	3843	3932						
BCC	3053																
BEQ	3319																
	415	450	459	555	676	833	848	858	893	899	921	975	1197	1249	1437		
	1494	1540	1553	1599	1677	1737	1833	1852	1918	1958	1988	2051	2065	2079	2104		
	2117	2131	2199	2239	2265	2520	2567	2614	2623	2640	2672	2781	2797	2808	2900		
	2906	2917	2987	2991	3105	3108	3118	3125	3129	3162	3182	3188	3250	3324	3379		
	3465	3487	3512	3573	3606	3641	3710	3798	3825	3840	4094						
BGE	3568																
BGT	3075	3717	3837	3878													
BHI	487	489	513	515													
BHIS	870																
BIC	460	471	548	670	673	674	830	866	868	902	1243	1301	1359	1420	1488		
	1491	1492	1592	1596	1831	1850	1956	2453	2503	3083	3227	3389	3399	3427	3434		
	3463	3504	3509	3538	3565	3565	3707	3794	3811	3838	3865	3871	3879	3879			
BIS	447	451	454	470	1082	1287	1341	1382	1401	1431	2395	2447	2560	2979	2992		
	3021	3028	3067	3193	3428	3435	3466	3475	3574	3712	3713	3845	4053				
BISB	2850																
BIT	445	449	832	847	892	898	1539	2525	2528	2566	2572	2622	2636	2671	2771		
	2780	2796	2807	2826	2836	2864	2872	2879	2899	2916	2923	2939	2947	2955	2963		
	2975	2986	2990	2994	3070	3072	3110	3124	3145	3161	3164	3169	3177	3187	3193		
	3233	3249	3423	3432	3446	3572											
BITB	2972	3166	3629														
BLOS	3891																
BLT	3620	3718	3835	3876	4046												
BMI	672	1490	1595	2495	3160	3462	3488										
BNE	422	424	443	446	457	479	510	558	836	840	856	1174	2526	2529	2573		
	2587	2602	2637	2709	2727	2742	2772	2790	2799	2802	2827	2837	2865	2873	2880		
	2902	2912	2924	2937	2940	2948	2956	2964	2971	2973	2976	2995	3024	3043	3071		
	3073	3111	3146	3165	3167	3170	3178	3194	3234	3424	3433	3447	3525	3548	3600		
	3608	3616	3630	3637	3708	3790	3796	3816	3823	3830	3867	3873	3895	3901	3901		
	4057	4096															
BPL	453	1642	1824	1843	1909	1972	2042	2095	2252	2336	2389	2630	2759	2778	2832		
	2870	2877	2909	2933	2945	2961	2984	3038	3077	3099	3152	3426	3431	3445	3502		
	3507	3537	3594	3634	3706	3792	3808	3863	3869								
BR	242	243	404	416	426	448	463	469	482	500	535	527	532	754	872		
	1177	1329	1332	1388	1391	1408	1433	1549	1630	1635	1640	1796	1803	1827	1846		
	1911	1914	1970	2044	2047	2075	2097	2100	2127	2141	2143	2145	2146	2149	2151		
	2153	2154	2157	2159	2162	2166	2177	2178	2180	2183	2186	2187	2189	2192	2195		
	2250	2325	2328	2334	2340	2397	2745	2762	2823	2853	2856	2868	2929	2949	2998		
	3045	3120	3186	3190	3236	3248	3252	3302	3311	3327	3335	3346	3355	3365	3374		
	3478	3480	3489	3596	3613	3623	3632	3639	3684	3699	3720	3819	3846	3848	3874		
	3897	3973	3997	4048	4098												
BVS	845																
CLC	3322	3390															
CLR	408	438	439	440	444	480	504	549	831	841	917	972	1025	1079	1190		
	1258	1284	1316	1372	1542	1597	1621	1725	1807	1906	1907	1953	1986	2039	2040		
	2061	2092	2093	2113	2114	2147	2155	2173	2174	2235	2236	2261	2322	2376	2377		
	2390	2407	2527	2595	2668	2693	2753	2754	2785	2793	2890	2891	2896	3026	3027		
	3030	3033	3132	3277	3300	3334	3394	3460	3470	3515	3697	3805	3806	3979	4042		
CLRB	3554	3612	3638	3902	4058												
CMP	414	417	423	554	675	839	869	920	974	1196	1248	1330	1389	1436	1493		
	1552	1598	1676	1736	1832	1851	1917	1957	1987	2050	2064	2078	2103	2116	2130		

	2198	2238	2264	2613	2789	2905	2970	3074	3378	3511	3789	3795	3815	3822	3834		
CMPB	3836	3866	3872	3875	3877	3890	4052										
	455	486	488	512	514	3042	3524	3567	3605	3607	3615	3636	3640	3797	3829		
	3894	3900	4095														
COMB	3345																
DEC	2496	2708	2726	2741	2801	2911	3128	3541	4055								
DECB	3619	3622	3745	3716													
EMT	88																
HALT	212	468	854	894	1184	2568	2779	2833	2878	2946	2962	2985	3100	3153	3595		
	3972	3996															
INC	402	838	844	855	1629	1735	2142	2150	2158	2327	2392	2409	2519	2761	2798		
	2901	2934	3044	3068	3093	3101	3181	3320	3363	3474	3528	3549	3553	3571	3711		
	3719	3844	3980	4047													
INCB	1175	3642															
IoT	89																
JMP	244	245	760	864	875	1065	1303	1360	1421	1607	1617	1724	1810	1812	1815		
	1896	2030	2084	2136	2271	2308	2331	2399	2415	2428	2437	2454	2673	2682	2768		
	2782	2783	2800	2804	2813	2839	2903	2922	2950	2966	2967	2988	2989	3025	3065		
	3102	3127	3131	3163	3210	3280											
JSR	755	852	860	1334	1393	1432	1548	1616	1623	1625	1634	1639	1723	1728	1808		
	1809	1826	1845	1913	1969	2046	2074	2099	2126	2140	2148	2152	2156	2161			
	2176	2179	2185	2188	2194	2249	2307	2318	2320	2321	2324	2330	2332	2333	2339		
	2378	2391	2393	2394	2396	2398	2408	2410	2411	2412	2414	2427	2452	2556	2561		
	2704	2705	2706	2707	2717	2718	2719	2720	2721	2722	2723	2724	2725	2735	2736		
	2737	2738	2739	2740	2743	2748	2749	2750	2792	2822	2829	2834	2852	2855	2867		
	2874	2888	2889	2895	2907	2928	2935	2977	2980	2996	3009	3012	3022	3029	3078		
	3081	3085	3114	3126	3130	3147	3154	3185	3189	3196	3202	3204	3247	3251	3301		
	3321	3344	3364	3371	3373	3614	3621	3628	3833	4015	4109	4110					
MOV	403	409	410	412	413	418	419	429	431	432	434	436	437	441	473		
	531	551	552	553	666	667	668	669	757	758	759	834	851	857	867		
	918	919	973	1018	1019	1021	1022	1023	1026	1030	1031	1072	1073	1075	1076		
	1077	1080	1125	1126	1195	1198	1236	1237	1239	1240	1241	1247	1250	1259	1263		
	1264	1277	1278	1280	1281	1282	1285	1290	1291	1319	1320	1322	1323	1324	1326		
	1327	1328	1331	1333	1337	1338	1344	1345	1375	1376	1378	1379	1380	1385	1386		
	1387	1390	1392	1397	1398	1404	1405	1434	1435	1485	1486	1487	1538	1541	1550		
	1551	1590	1591	1593	1615	1622	1624	1643	1726	1727	1802	1811	1814	1825	1828		
	1829	1830	1844	1847	1848	1849	1910	1912	1915	1916	1954	1955	1968	1973	1974		
	1975	1984	1985	2043	2045	2048	2049	2062	2063	2073	2086	2077	2096	2098	2101		
	2102	2115	2125	2128	2129	2165	2175	2181	2182	2184	2190	2191	2193	2196	2197		
	2237	2248	2253	2254	2255	2262	2263	2304	2305	2306	2316	2317	2319	2323	2329		
	2337	2338	2341	2413	2423	2424	2425	2426	2436	2448	2449	2450	2451	2461	2467		
	2478	2484	2491	2517	2533	2535	2540	2543	2546	2549	2552	2555	2596	2624	2669		
	2694	2751	2752	2755	2757	2765	2766	2767	2786	2787	2794	2810	2811	2812	2816		
	2821	2843	2851	2866	2882	2892	2893	2897	2919	2920	2921	2926	2927	3003	3008		
	3031	3032	3034	3036	3049	3056	3062	3063	3064	3084	3088	3094	3113	3184	3198		
	3212	3221	3226	3228	3246	3255	3261	3276	3278	3318	3362	3386	3393	3395	3400		
	3401	3472	3477	3479	3503	3508	3513	3539	3558	3597	3598	3602	3617	3680	3688		
	3689	3690	3696	3703	3721	3722	3723	3724	3725	3750	3751	3752	3753	3754	3755		
	3756	3757	3758	3759	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775	3802		
	3826	3831	3860	3861	3886	3889	3904	3905	3906	3907	3928	3929	3933	3961	3962		
	3963	3964	3965	3966	3967	3968	3969	3970	3971	3977	3978	3982	3983	3984	3985		
	3986	3987	3988	3989	3990	3993	4013	4014	4017	4034	4035	4036	4037	4038	4039		
	4040	4041	4090	4091	4092	4102	4103	4108									
MOVB	427	491	496	511	517	522	1628	1733	1734	2326	2700	2788	2791	2854	2857		
	2894	3040	3047	3048	3052	3055	3092	3158	3191	3245	3253	3310	3316	3343	3354		
	3372	3377	3467	3469	3471	3485	3526	3550	3551	3555	3569	3599	3627	3635	3681		

	3682	3685	3686	3687	3691	3694	3695	3714	3793	3810	3864	3870	3893	3898	3931
	4054	4097	4099												
NEG	3692														
NEGB	3054														
NOP	861	862	863	1028	1029	1083	1124	1261	1262	1280	1289	1342	1343	1402	1403
	1715														
RESET	430	859													
ROL	3391	3392	3698	3700	3701	3702	3704								
ROLB	3326														
ROR	3396	3397													
RTI	433	1027	1032	1081	1127	1260	1265	1286	1292	1339	1346	1399	1406	2469	2486
	2569	2597	2603	2615	2625	2756	2795	2898	3035	3173	3604	3726	3760	3776	3832
	3880	3908	3995												
RTS	901	903	1626	1637	1708	1729	2164	2379	2505	2575	2632	2641	2696	2710	2728
	2744	2803	2838	2848	2858	2881	2883	2910	2913	2999	3013	3057	3106	3133	3175
	3206	3218	3238	3264	3380	3382	3402	3429	3436	3449	3476	3505	3510	3514	3516
	3527	3529	3540	3542	3559	3575	3644	3934	4018	4060	4104	4111			
SBC	4044														
SEC	3317	3325													
SUB	2534	2678	3473	3552	3556	4043	4045								
TRAP	3936	3945	3946	3947	3949	3951	3952	3953	3954	3955	3956				
TST	421	442	452	458	478	756	835	1170	1641	1842	1971	2251	2335	2586	2601
	2764	2777	2806	2831	2869	2876	2908	2915	2932	2936	2944	2960	2983	3023	3061
	3098	3104	3107	3151	3159	3323	3381	3430	3444	3464	3486	3501	3536	3547	3601
	3609	3631	3709	3824	3839	3930									
TSTB	509	671	1489	1594	1823	1908	2041	2094	2388	2493	2629	2758	3037	3076	3117
	3425	3461	3506	3593	3633	3791	3807	3862	3868	4093					
,ASCII	3652	3653	4295	4306	4317	4324									
,ASCIZ	881	883	2577	2684	2689	3654	3912	3913	3914	3916	3999	4115	4119	4128	4133
	4139	4144	4147	4151	4154	4159	4162	4164	4166	4173	4176	4179	4182	4185	4188
	4191	4193	4199	4204	4210	4213	4217	4222	4224	4227	4230	4233	4238	4240	4244
	4247	4250	4254	4258	4260	4262	4264	4268	4272	4277	4285	4293	4302	4313	4332
,BLKB	3911	4081	4345												
,BYTE	475	476	493	494	498	499	519	520	524	525	2464	2465	2481	2482	2537
	2538	2680	2681	2845	2846	3005	3006	3200	3201	3214	3215	3223	3224	3230	3231
	3257	3258	3648	3649	3650	3651	3727	3728	3729	3730	3909	3910	3918	3919	
,DSABL	3849														
,ENABL	4	3782													
,END	4347														
,ENDC	16	88	180	194	240	252	253	254	265	266	267	272	273	274	293
	294	295	386	397	423	429	476	477	568	652	689	753	931	956	985
	1014	1045	1063	1089	1123	1137	1166	1208	1227	1338	1345	1398	1405	1447	1469
	1506	1531	1563	1584	1653	1671	1687	1705	1751	1794	1862	1894	1928	1947	1998
	2028	2211	2229	2287	2303	2351	2374	2465	2466	2482	2483	2510	2511	2515	2516
	2652	2666	2713	2731	2846	2847	3006	3007	3141	3201	3202	3215	3216	3224	3225
	3231	3232	3258	3259	3273	3296	3305	3330	3338	3349	3358	3367	3422	3484	3499
	3519	3532	3579	3599	3658	3735	3750	3783	3785	3813	3849	3853	3861	3862	3889
	3891	3894	3896	3912	3920	3923	3929	3932	3944	3945	3946	3947	3948	3949	3950
	3951	3952	3953	3954	3955	3956	3960	3969	3970	3976	3982	3983	3993	3995	4002
,EQUIV	4005	4023	4084	4341											
	88	89	97	112	113	142	143	144	145	146	147	148	149	150	151
	170	171	172	173	174	175	176	177	178	179					
,EVEN	887	2578	2691	4001	4338										
,IF	12	86	152	180	240	251	252	253	264	265	266	271	272	273	292
	293	294	385	396	420	423	475	476	565	649	686	750	928	953	982
	1011	1042	1060	1086	1120	1134	1163	1205	1224	1337	1344	1397	1404	1444	1460

	1503	1528	1560	1581	1650	1668	1684	1702	1748	1791	1859	1891	1925	1944	1995
	2025	2208	2226	2284	2300	2348	2371	2464	2465	2481	2482	2509	2510	2514	2515
	2649	2663	2712	2730	2845	2846	3005	3006	3140	3200	3201	3214	3215	3223	3224
	3230	3231	3257	3258	3272	3295	3304	3329	3337	3348	3357	3366	3421	3483	3496
	3518	3531	3578	3599	3657	3734	3779	3783	3784	3813	3852	3853	3861	3862	3889
	3890	3894	3895	3911	3912	3918	3922	3928	3932	3936	3945	3946	3947	3948	3949
	3951	3952	3953	3954	3955	3956	3959	3969	3970	3975	3982	3983	3991	3993	3995
,IFF	3999	4004	4022	4083	4340										
	88	252	253	254	265	266	267	272	273	274	293	294	295	386	397
	476	1337	1344	1397	1404	2464	2465	2481	2482	2510	2511	2515	2516	2713	2731
	2846	3005	3006	3141	3201	3215	3224	3231	3258	3273	3296	3305	3330	3338	3349
	3358	3367	3422	3484	3499	3519	3532	3579	3658	3735	3780	3785	3853	3855	3860
	3881	3882	3891	3895	3912	3923	3929	3960	3976	3993	4005	4023	4084	4341	
,IFT	3855	3860													
,IFTF	3800	3853	3856												
,IIF	11	16	21	22	474	2462	2479	2844	3004	3199	3213	3222	3229	3256	3646
	3647	3648	3649	3650	3651	3652	3653	3654	3655	3700	3781	3803	3904	3912	3918
	3919	3920	3944	3945	3946	3947	3949	3951	3952	3953	3954	3955	3956		
,IRP	240	3750	3770	3963	3969	3982	3983								
,LIST	3	194	212	240	564	653	685	754	927	957	981	1015	1041	1064	1085
	1124	1133	1167	1204	1228	1443	1470	1502	1532	1559	1585	1649	1672	1683	1706
	1747	1795	1858	1895	1924	1948	1994	2029	2207	2230	2283	2304	2347	2375	2648
	2667	3881	3936	3944	3945	3946	3947	3948	3949	3950	3951	3952	3953	3954	3955
	3956	3957													
,MACRO	246	248	3936												
,MCALL	5	6	7	194											
,NLIST	2	194	212	240	564	653	685	754	927	957	981	1015	1041	1064	1085
	1124	1133	1167	1204	1228	1443	1470	1502	1532	1559	1585	1649	1672	1683	1706
	1747	1795	1858	1895	1924	1948	1994	2029	2207	2230	2283	2304	2347	2375	2648
	2667	3881	3936	3944	3945	3946	3947	3948	3949	3950	3951	3952	3953	3954	3955
	3956	3957													
,PAGE	43	83	2507												
,REPT	212	565	649	686	750	928	953	982	1011	1042	1060	1086	1120	1134	1163
	1205	1224	1444	1466	1503	1528	1560	1581	1650	1668	1684	1702	1748	1791	1859
	1891	1925	1944	1995	2025	2208	2226	2284	2300	2348	2371	2649	2663		
,SBTTL	84	249	269	295	346	398	420	533	904	1066	1304	1362	1422	1608	1709
	1717	1798	1805	1817	1897	2032	2085	2168	2273	2311	2380	2402	2417	2431	2439
	2507	2582	2699	2746	2885	3016	3156	3265	3408	3451	3543	3576	3655	3732	3777
	3920	3936	3957	4002	4020	4112									
,TITLE	11														
,WORD	211	214	215	216	217	234	377	378	1178	3090	3643	3646	3647	3731	3780
	3781	3992	3994	4019	4056										

ERRORS DETECTED: 0
DEFAULT GLOBALS