

CD11/CD20

CD DIAGNOSTIC
MD-11-DZCDB-A

EP-DZCDB-A-DL-A
COPYRIGHT © 1976
FICHE 1 OF 2

NOV 1976
digital
MADE IN USA

This microfiche contains 100 frames of technical data for the CD11/CD20 diagnostic system. The frames are arranged in a 10x10 grid. Each frame contains a page of data, which includes:

- Diagnostic test procedures and instructions.
- Tables of test results and parameters.
- Flowcharts and diagrams illustrating the diagnostic process.
- Textual descriptions of system components and their functions.

The data is presented in a structured, tabular format, typical of technical manuals. The text is small and dense, characteristic of microfiche storage.

CD11/CD20

DIAGNOSTIC
AH-0030A-MC

11-DZCUB-A-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 2

MADE IN U.S.A.

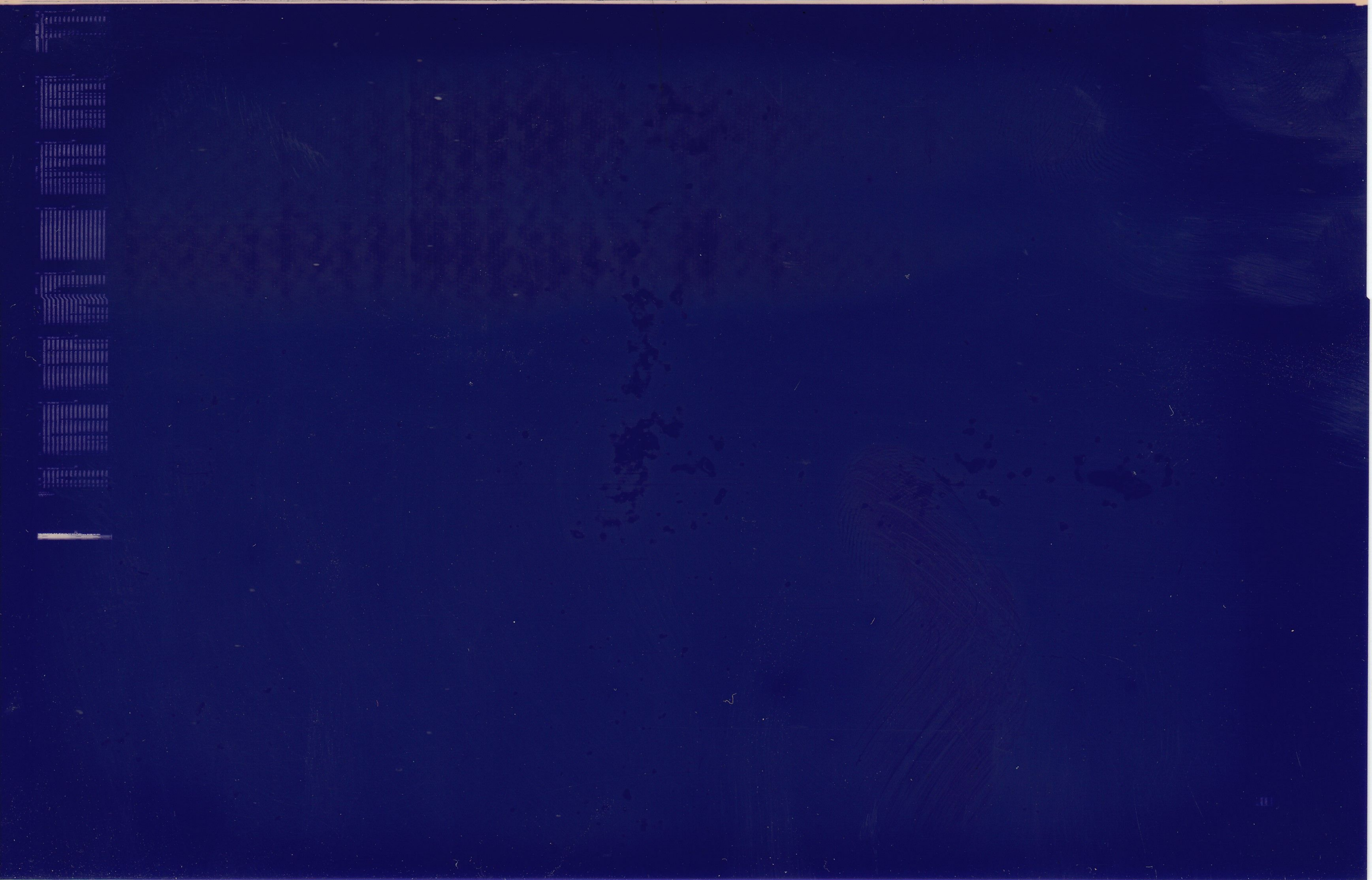
This microfiche card contains a grid of 100 frames of diagnostic data, arranged in 10 rows and 10 columns. Each frame displays a different set of diagnostic information, including alphanumeric codes, numerical values, and graphical representations such as waveforms and histograms. The data is organized into various sections, with some frames featuring headers like 'TEST RESULTS' or 'PARAMETER VALUES'. The overall layout is dense and structured, typical of a diagnostic manual or test report.

CD11/CD20

CD DIAGNOSTIC
MD-11-DZCDB-A

EP-DZCDB-A-DL-A
COPYRIGHT © 1976
FICHE 2 OF 2

NOV 1976
digital
MADE IN USA



CD11/CD20

DIAGNOSTIC
AH-0030A-MC

11-DZCDB-A-DL-A
COPYRIGHT © 1976
FICHE 2 OF 2

NOV 1976
digital
MADE IN U.S.A.

IDENTIFICATION

SEQ 0001

PRODUCT CODE: 4INDEC-11-DZCDB-A-D
PRODUCT NAME: CD11/CD20 CARD READER DIAGNOSTIC
DATE CREATED: 21-MAR-1976
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: BRUCE BURGESS

COPYRIGHT (C) 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	PROGRAM PURPOSE
1.2	SYSTEM REQUIREMENTS HARDWARE REQUIREMENTS SOFTWARE REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	LOADING AND STARTING PROCEDURES INSTRUCTION AND DATA RELIABILITY ERROR FUNCTIONS MODELS M1000/M200 ERROR FUNCTIONS MODEL M1200 SINGLE SUBTEST LOOP SINGLE DATA PATTERN TEST
2.2	SPECIAL ENVIRONMENTS
2.3	PROGRAM OPTIONS DEFAULT PARAMETERS CONTROL SWITCH SETTINGS STARTING ADDRESSES
2.4	EXECUTION TIMES
3.0	ERROR INFORMATION
3.1	ERROR REPORTING PROCEDURES INPUT HOPPER EMPTY DATA RELIABILITY TESTING GENERAL PROGRAM OPERATION
3.2	ERROR HALTS
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES STATUS REGISTER (177160) COLUMN COUNT REGISTER (177162) CURRENT ADDRESS REGISTER (177164) DATA BUFFER REGISTER (177166)
6.0	SUB-TEST SUMMARIES
6.1	INSTRUCTION TESTS
6.2	DATA RELIABILITY TEST
6.3	ERROR FUNCTION TESTS FOR M1000/M200/M1200
6.4	READING A SINGLE DATA PATTERN
6.5	LOOPING ON A SELECTED TEST
7.0	FLOW CHARTS
8.0	PROGRAM LISTING

1.0 GENERAL PROGRAM INFORMATION

1.1 PROGRAM PURPOSE

THIS PROGRAM CAN BE USED WITH EITHER A CD11 OR CD20 CARD READER INTERFACE TO A DECIMATION M1000, M200 OR AN M1200 MODEL CARD READER. THE PROGRAM TESTS ALL LOGIC FUNCTIONS OF THE CARD READER AS WELL AS EXERCISING ALPHANUMERIC AND BINARY CODED DATA VIA PUNCHED CARD TEST DECKS. SEPARATE STARTING ADDRESSES ALLOW TESTING OF ERROR FUNCTIONS (E.G. - HOPPER CHECK, STACK CHECK, ETC.) FOR ALL CARD READER MODELS UTILIZING MANUAL TECHNIQUES. TO AID IN DIAGNOSING SPECIAL PATTERNS A SECTION OF THE PROGRAM WILL ALLOW TESTING OF CARD DECKS WITH ALL COLUMNS OF EACH CARD IDENTICALLY PUNCHED.

THE DISTINCTION BETWEEN THE CD11 AND CD20 CARD READER INTERFACES WILL BE DESCRIBED IN SECTION 1.2, PARAGRAPH A.

THE ALPHANUMERIC AND BINARY TEST DECKS TO BE USED WILL BE DESCRIBED IN SECTION 1.2, PARAGRAPH B.

1.2 SYSTEM REQUIREMENTS

A. HARDWARE REQUIREMENTS

A PDP-11 FAMILY COMPUTER WITH EITHER A CD11 OR CD20 CARD READER INTERFACE TO A DECIMATION MODEL CARD READER (MODELS M200, M1000, M1200).

***** NOTE *****
A MINIMUM OF 12K OF MEMORY IS REQUIRED
FOR LOADING & RUNNING THIS DIAGNOSTIC!

1. THE CD20 INTERFACE IS A CD11 INTERFACE WITH ECO #CD-11-00014 INSTALLED. THE CD20 INTERFACE IS USED WITH THE DECSYSTEM 20 SERIES OF COMPUTERS. THE ECO IMPLEMENTS THE FOLLOWING ADDITIONAL SOFTWARE FEATURES INTO THE NORMAL CD11 CARD READER CONTROLLER:
 - A. DURING DATA TRANSFERS IN UNPACKED MODE BITS 15 THRU 12 IN THE DATA REGISTER (177166) HAVE BEEN REDEFINED TO INDICATE MORE THAN ONE BIT SET IN A ZONE (BIT15) AND WHICH ZONE (BITS 14 THRU 12).
 - B. DURING NON-DATA TRANSFERS PERIODS THE DATA REGISTER (177166) IS USED AS A SECOND STATUS REGISTER WITH BIT DEFINITIONS AS FOLLOWS:

BIT	DEFINITION
---	-----
15	ALWAYS SET IF ECO IS INSTALLED
14	ALWAYS CLEAR IF ECO NOT INSTALLED
14	READ CHECK ! BREAKOUT OF BIT14 OF
13	PICK CHECK ! STATUS REGISTER
12	STACK CHECK ! (177160)

B. SOFTWARE REQUIREMENTS

TWO MAIN CARD TEST DECKS ARE REQUIRED BY THIS PROGRAM:

<u>ALPHANUMERIC DECK</u>	<u>BINARY DECK</u>
--------------------------	--------------------

MAINDEC-89-D181-C	MAINDEC-89-D182-C
-------------------	-------------------

IN ADDITION TO SPARE CARDS FOR ERROR FUNCTION TESTING.

1.3 RELATED DOCUMENTS AND STANDARDS

- A. CD11 ENGINEERING DRAWINGS
- B. PDP11 PERIPHERALS HANDBOOK
- C. PDP11 PROCESSOR HANDBOOK
- D. MAINDEC-11-D20AC-B1
SYSMAC-SPL DIAGNOSTIC UTILITIES PACKAGE
- E. MAINDEC-11-D20XA
'XODP' USER'S GUIDE
- F. DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS
PROGRAMMING PRACTICES
DOC. NO. 175-003-009-20

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

NONE.

1.5 ASSUMPTIONS

- A. IT IS ASSUMED THAT BOTH CARD TEST DECKS ARE IN THEIR PROPER SEQUENCE. IT IS A GOOD IDEA UPON RECEIVING THE TEST DECKS TO NUMBER THEM IN THE TOP RIGHT CORNER IN THE EVENT THAT THEY ARE ACCIDENTLY DROPPED.
- B. IT IS ASSUMED THAT ECO #CD-11-00014 HAS BEEN INSTALLED WHEN THIS PROGRAM IS BEING RUN ON A DECSYSTEM 20 SERIES COMPUTER.

2.0 OPERATING INSTRUCTIONS
-----2.1 LOADING AND STARTING PROCEDURES

THERE ARE FIVE DISTINCT SECTIONS TO THE PROGRAM EACH HAVING ITS OWN STARTING ADDRESS, SWITCH SETTINGS, AND MODES OF OPERATION. THEY ARE AS FOLLOWS:

A. INSTRUCTION & DATA RELIABILITY

1. LOAD PROGRAM INTO MEMORY. ON A DECSYSTEM 20 SERIES COMPUTER THIS IS ACCOMPLISHED VIA A 'FLOPPY' DISKETTE ON UNIT 0 BY -

1ST- DEPRESSING THE 'FLOPPY' SWITCH (ON FRONT END PANEL) TO LOAD 'RXDP' (FLOPPY MONITOR)

2ND- TYPING 'DZCDB*' (CARRIAGE RETURN) ON CONSOLE IN RESPONSE TO THE 'RXDP' MONITOR REQUEST FOR INPUT (A DOT.).

WHERE * = LATEST REV. LETTER OF THE PROGRAM

2. LOAD A TEST DECK (ALPHANUMERIC OR BINARY) INTO THE CARD READER INPUT HOPPER
3. PRESS 'RESET' BUTTON ON THE CARD READER
4. SET A STARTING ADDRESS OF 200 INTO SWITCH REGISTER
5. PRESS 'LOAD ADDRESS'
6. SET SWITCHES FOR MODE OF OPERATION:

'ALPHA' DECK (IMAGE MODE) - SET SW<02>

'ALPHA' DECK (PACK MODE) - SET SW<03>

'BINARY' DECK (IMAGE MODE) - SET SW<02> & SW<04>

'BINARY' DECK (PACK MODE) - SET SW<03> & SW<04>

NOTE: WITH ONLY THE ABOVE SWITCHES SET FOR MODE OF OPERATION THE PROGRAM WILL PASS THRU THE INSTRUCTION PORTION ONLY ONCE (1ST PASS). ON ALL SUBSEQUENT PASSES THE PROGRAM WILL LOOP ON THE DATA RELIABILITY PORTION. TO ALTER THIS APPROACH SET OTHER SWITCHES AS INDICATED UNDER SECTION 2.3.

7. PRESS 'START'
8. WITH ONLY THE SWITCHES SET AS INDICATED UNDER ITEM 6., PASS COMPLETION PRINTOUTS SHOULD APPEAR AS FOLLOWS (AN EXAMPLE):

MAINDEC-11-DZCDB REV.A

ENTERING LOGIC TESTS
 ENTERING DATA TESTS
 END PASS #1

* AT THIS POINT THE INPUT HOPPER SHOULD BE EMPTY *
 * AND THE PROGRAM WILL HANG WAITING, AT WHICH *
 * POINT - *
 * A. RELOAD TEST DECK/S INTO INPUT HOPPER, AND *
 * B. PRESS 'RESET' ON CARD READER *
 * PROGRAM SHOULD RESUME OPERATION WITH. *

ENTERING DATA TESTS
 END PASS #2

ETC.

NOTE: THE FORM OF THE PRINTOUT WILL VARY AS OTHER SWITCHES ARE SET.

- B. ERROR FUNCTIONS FOR CARD READER MODELS M200 OR M1000
 1. LOAD PROGRAM INTO MEMORY AS INDICATED BY SECTION 2.1, PARAGRAPH A., SUBPARAGRAPH 1.
 2. LOAD A FEW SPARE CARDS INTO THE INPUT HOPPER.
 NOTE: DO NOT LOAD A TEST DECK HERE AS PORTIONS OF ERROR FUNCTION TESTING DESTROYS CARDS!
 3. PRESS 'RESET' BUTTON ON THE CARD READER
 4. SET A STARTING ADDRESS OF 210 INTO SWITCH REGISTER.
 5. PRESS 'LOAD ADDRESS'
 6. SET DESIRED SWITCHES AS INDICATED BY SECTION 2.3

7. PRESS 'START'
8. FOLLOW THE INSTRUCTIONS AS THEY ARE PRINTED OUT
A FULL PASS OF THIS SECTION SHOULD APPEAR AS FOLLOWS
(AN EXAMPLE):

ENTERING M1000/M200 ERROR FUNCTION TESTS
MAINDEC-11-DZCDB REV.A

* INSTRUCTIONS FOLLOW AND UPON A COMPLETE PASS *
* WILL APPEAR, *

MAINDEC-11-DZCDB REV.A

.
ETC.

C. ERROR FUNCTIONS FOR CARD READER MODEL M1200

EVERYTHING APPLIES AS INDICATED UNDER SECTION 2.1,
PARAGRAPH B., EXCEPT:

1. START ADDRESS IS 250, AND
2. LEAD-IN MESSAGE UPON EXECUTION IS "ENTERING M1200 ERPOR
FUNCTION TESTS"

D. SINGLE SUBTEST LOOP

1. LOAD PROGRAM INTO MEMORY AS INDICATED BY SECTION 2.1,
PARAGRAPH A., SUBPARAGRAPH 1.
2. LOAD A FEW SPARE CARDS INTO THE INPUT HOPPER

NOTE: SINCE ONLY THE INSTRUCTION PORTION OF THE
PROGRAM IS MEANT TO BE SELECTED UNDER THIS
PHASE OF OPERATION A CARD TEST DECK MAY BE
SUBSTITUTED IN PLACE OF THE SPARE CARDS.

3. PRESS 'RESET' BUTTON ON THE CARD READER
4. SET A STARTING ADDRESS OF 220 INTO SWITCH REGISTER
5. PRESS 'LOAD ADDRESS'
6. AT THE 1ST 'HALT':
LOAD THE STARTING ADDRESS OF THE DESIRED TEST (ADDRESS
OF 'SCOPE' INSTRUCTION AT BEGINNING OF TEST), THEN
PRESS 'CONTINUE'.
7. AT THE 2ND 'HALT':
SET DESIRED SWITCHES AS INDICATED BY SECTION 2.3
(SW<11> MUST NOT BE SET), THEN PRESS 'CONTINUE'
*** HOWEVER *** (SW<14> MUST BE SET... MUST BE SET ...)
8. PROGRAM WILL LOOP ON TEST SELECTED

E. SINGLE DATA PATTERN TEST

1. LOAD PROGRAM INTO MEMORY AS INDICATED BY SECTION 2.1, PARAGRAPH A., SUBPARAGRAPH 1.
2. LOAD A 'PREPARED' DECK INTO THE INPUT HOPPER
NOTE: THE 'PREPARED' DECK CAN CONSIST OF ONE OR MORE CARDS AND HAVE ANY DATA PATTERN BUT THIS PATTERN MUST BE IDENTICAL IN ALL 80 COLUMNS OF EACH AND EVERY CARD MAKING UP THE DECK (E.G. IF COLUMN 1 CONTAINS 1777 SO MUST ALL THE OTHER 79 COLUMNS).
3. PRESS 'RESET' BUTTON ON THE CARD READER
4. SET A STARTING ADDRESS OF 240 INTO SWITCH REGISTER
5. PRESS 'LOAD ADDRESS'
6. PRESS 'START'
7. AT THE 1ST 'HALT'
SET THE DATA PATTERN SELECTED INTO THE SWITCH REGISTER USING SMS<11 THRU 00>, THEN PRESS 'CONTINUE'
8. AT THE 2ND 'HALT'
SET DESIRED SWITCHES AS INDICATED BY SECTION 2.3
9. WHEN THE CARD READER RUNS OUT OF CARDS RELOAD THE 'PREPARED' DECK AND PRESS 'RESET' BUTTON ON THE CARD READER
10. THE PROGRAM SHOULD CONTINUE

2.2 SPECIAL ENVIRONMENTS

THE PROGRAM DOES HAVE THE CAPABILITY TO BE RUN ON THE ACT-11 MANUFACTURING LINE BUT IS NOT IDEALLY SUITED FOR THIS ENVIRONMENT DUE TO THE PROGRAM'S REQUIREMENT OF LOADING CARD TEST DECKS.

A. DEFAULT PARAMETERS

LOCATION LABEL	CONTENTS	USE
CDST:	177160	STATUS
CDCC:	177162	COLUMN COUNT
CDBA:	177164	BUS ADDRESS
CDDB:	177166	DATA/(STATUS ON CD20)
INTVEC:	230	INTERRUPT PC
INTVEC+2:	232	INTERRUPT PS

IF THE CD11/CD20 IS EVER CONFIGURED SO THAT THE ABOVE STANDARD ADDRESSES AND VECTORS ARE NOT RELEVANT THEN JUST PATCH THE ABOVE PROGRAM LOCATIONS TO THE CORRECT VALUES BEFORE ATTEMPTING TO EXECUTE THE PROGRAM.

B. CONTROL SWITCH SETTINGS

SWITCH	USE
SW<15>=1	HALT ON ERROR
SW<15>=0	CONTINUE ON ERROR
SW<14>=1	LOOP ON CURRENT TEST
SW<14>=0	CONTINUE TO NEXT TEST
SW<13>=1	INHIBIT ERROR PRINTOUT
SW<13>=0	PRINT ERROR REPORTS
SW<12>=1	INHIBIT TRACE TRAPPING
SW<12>=0	ALLOW TRACE TRAPPING
SW<11>=1	INHIBIT SUB-TEST ITERATIONS
SW<11>=0	ALLOW SUB-TEST ITERATIONS
SW<07>=1	LOOP ON INSTRUCTION TESTS ONLY
SW<07>=0	NOT APPLICABLE
SW<06>=1	LOOP ON INSTRUCTION & DATA RELIABILITY TEST WHEN CONTINUING FROM ONE CARD TEST DECK TO ANOTHER.
SW<06>=0	INSTRUCTION & DATA RELIABILITY TEST COVERED ON 1ST CARD TEST DECK LOAD. ONLY DATA RELIABILITY TEST COVERED ON ALL SUBSEQUENT CARD TEST DECK LOADS.

SW<05>=1 WHEN MORE THAN ONE CARD TEST DECK IS
LOADED 'HALT' AT COMPLETION OF EACH DECK.
NOTE: PRESSING 'CONTINUE' WILL RESUME
PROGRAM OPERATION AFTER THE 'HALT'.
SW<05>=0 WHEN MORE THAN ONE CARD TEST DECK IS
LOADED RUN THE DATA RELIABILITY TEST
AUTOMATICALLY FROM DECK TO DECK.

SW<04>=1 INDICATOR FOR BINARY TEST DECK
SW<04>=0 NOT BINARY TEST DECK

SW<03>=1 INDICATOR FOR PACK MODE
SW<03>=0 NOT PACK MODE

SW<02>=1 INDICATOR FOR IMAGE MODE
SW<02>=0 NOT IMAGE MODE

C. STARTING ADDRESSES

ADDRESS	USE
200	INSTRUCTION & DATA RELIABILITY TESTING
210	ERROR FUNCTION TESTING OF CARD READER MODELS M200 OR M1000
220	LOOPING ON A SINGLE INSTRUCTION TEST
240	READING A SINGLE DATA PATTERN CONTINUOUSLY
250	ERROR FUNCTION TESTING OF CARD READER MODEL M1200

2.4 EXECUTION TIMES

(TO BE DETERMINED)

3.0 ERROR INFORMATION

3.1 ERROR REPORTING PROCEDURES

THREE TYPES OF ERROR REPORTING TECHNIQUES ARE USED AS FOLLOWS:

A. WHEN INPUT HOPPER GOES EMPTY DURING TESTING

THE FOLLOWING MESSAGE IS TYPED:

CARD READER IS OFF-LINE
REMEDY THE CONDITION BY RELOADING INPUT HOPPER
WITH CARD DECK - PRESS 'RESET' BUTTON ON CARD READER
AND 'CONTINUE' SWITCH ON CPU PANEL

NOTE: ALLOW A FEW SECONDS TO TRANSPIRE BETWEEN PRESSING THE
'RESET' BUTTON AND THE 'CONTINUE' SWITCH AS IF THIS
OPERATION IS DONE TOO QUICKLY THE CARD READER WILL
NOT HAVE HAD A CHANCE TO RESET ITSELF AND THE ABOVE
REPORT WILL ONLY BE REITERATED

B. DURING DATA RELIABILITY TESTING

THE FOLLOWING FORMAT IS TYPED WHEN A DATA ERROR OCCURS:

DECK CARD NUM CARD COL SHB WAS

WHERE: 'DECK' REPRESENTS EITHER 'ALPHA' OR 'BINARY'

'CARD NUM' REPRESENTS WHICH CARD OUT OF THE 80.
CARDS ON WHICH THE ERROR WAS FOUND. 'CARD NUM'
VALUE IS PRINTED IN DECIMAL.

'CARD COL' REPRESENTS THE COLUMN ON THE CARD
(SPECIFIED BY 'CARD NUM') WHICH CONTAINS THE
ERROR. 'CARD COL' VALUE IS PRINTED IN DECIMAL.

'SHB' REPRESENTS THE ENCODED VALUE THAT SHOULD
BE INTERPRETED.

'WAS' REPRESENTS THE VALUE THAT WAS INTERPRETED.

C. GENERAL ERRORS DURING PROGRAM OPERATION

THERE ARE SEVEN GENERAL TYPES OF ERROR REPORTS IN USE
THROUGHOUT THE PROGRAM AS FOLLOWS:

TYPE 1

(PC) (SP) (CDS) (CDD) (PS)

TYPE 2

(PC) (SP) (CDS) (CDD) (CDC) (PS)

TYPE 3

(PC) (SP) (CDS) (CDD) (CDA) (PS)

TYPE 4

(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)

TYPE 5

(PC) (SP) (CDS) (CDD) (CDC) (CDA) (CDA) (PS)
WAS SHB

TYPE 6

(PC) (SP) (CDS) (CDD) (CDC) (CDA) (CDC) (PS)
WAS SHB

TYPE 7

(PC) (SP) (CDS) (CDD) (CDD) (PS)
WAS SHB

- D. IF THERE IS NO TERMINAL TO OUTPUT INFORMATION
- E. DURING A POWER FAILURE IF THE POWER UP SEQUENCE WAS STARTED BEFORE THE POWER DOWN SEQUENCE HAD COMPLETED.

4.0 PERFORMANCE AND PROGRESS REPORTS

NOT APPLICABLE

5.0 DEVICE INFORMATION TABLES

A. STATUS REGISTER (177160) BIT DESIGNATIONS

BIT	DESIGNATION	MODE
---	-----	----
0	READ	WRITE
1	DATA PACKING	READ/WRITE
2	HOPPER EMPTY	READ
3	READER TRANSITION TO ON-LINE	READ
4	EXTENDED BUS ADDRESS (BIT16)	READ/WRITE
5	EXTENDED BUS ADDRESS (BIT17)	READ/WRITE
6	INTERRUPT ENABLE	READ/WRITE
7	CONTROLLER READY	READ
8	POWER CLEAR	WRITE
9	NON-EXISTANT MEMORY	READ
10	DATA LATE	READ
11	DATA ERROR	READ
12	OFF-LINE	READ
13	END OF FILE (M1200 ONLY)	READ
14	CARD READER ERROR (READ, STACK OR PICK)	READ
15	ERROR	READ

B. COLUMN COUNT REGISTER (177162) BIT DESIGNATIONS

BITS <15:0> CONTAIN THE 2'S COMPLEMENT OF THE NUMBER OF COLUMNS TO BE TRANSFERRED TO MEMORY WHEN CARDS ARE BEING READ. THE CONTENTS OF THIS REGISTER IS INCREMENTED BY 1 EACH TIME A COLUMN TRANSFER OCCURS AND ALL TRANSFERS ARE INHIBITED WHEN THE CONTENTS OF THIS REGISTER IS EQUAL TO 0.

ALL BITS ARE READ/WRITE.

C. CURRENT ADDRESS REGISTER (177164) BIT DESIGNATIONS

BITS <15:0> CONTAIN THE MEMORY ADDRESS INTO WHICH THE NEXT COLUMN OF DATA IS TO BE STORED. THIS REGISTER IS INITIALLY SET TO THE MEMORY LOCATION OF THE 1ST COLUMN TO BE READ. IT THEN INCREMENTS BY 1 FOR TRANSFERS IN PACK MODE; BY 2 FOR TRANSFERS IN NON-PACK MODE. ALL BITS ARE READ/WRITE.

D. DATA BUFFER REGISTER (177166) BIT DESIGNATIONS (READ ONLY)

1. NON-PACK MODE (CD11/CD20)

BIT	CORRESPONDING CARD IMAGE
11	ZONE 12
10	ZONE 11
9-0	ZONES 0-9, RESPECTIVELY
15	ALWAYS SET (CD20 ONLY)
14	READ CHECK (CD20 ONLY)
13	PICK CHECK (CD20 ONLY)
12	STACK CHECK (CD20 ONLY)

2. PACK MODE (CD11/CD20)

BITS 7 THRU 3 ARE ENCODED AS FOLLOWS:

BIT	CORRESPONDING CARD IMAGE
7	ZONE 12
6	ZONE 11
5	ZONE 0
4	ZONE 9
3	ZONE 8

BITS 2 THRU 0 REPRESENT AN OCTAL CODE ENCODED AS FOLLOWS:

BIT 02	BIT 01	BIT 00	CARD ZONE
0	0	0	ZONES 1-7
0	0	1	ZONE 1
0	1	0	ZONE 2
0	1	1	ZONE 3
1	0	0	ZONE 4
1	0	1	ZONE 5
1	1	0	ZONE 6
1	1	1	ZONE 7

BITS 8 THRU 15 ARE UNUSED.

6.0 SUB-TEST SUMMARIES
-----6.1 INSTRUCTION TESTS

INITIALIZATION OF ALL REGISTERS
 READ/WRITE OF STATUS REGISTER
 READ/WRITE OF COLUMN COUNT REGISTER
 READ/WRITE OF BUS ADDRESS REGISTER
 CONTROLLER READY TO CLEAR BIT00 OF CDS (177160)
 'HOPPER EMPTY' (BIT02 OF CDS) TO BE CLEAR AFTER CARD READ
 INTERRUPT FROM CONTROLLER READY
 NO INTERRUPT WITH CPU AT LEVEL 7
 INTERRUPTS ON LEVELS 7 THRU 1
 NO INTERRUPT WITH INTERRUPT ENABLE SET ONLY
 SIMULTANEOUS INTERRUPTS AT MORE THAN 1 LEVEL
 NON-EXISTANT MEMORY DETECTION
 BYTE LOADING OF COLUMN COUNT REGISTER
 BYTE LOADING OF BUS ADDRESS REGISTER
 DATIP LOADING OF COLUMN COUNT REGISTER
 DATIP LOADING OF BUS ADDRESS REGISTER
 WORD COUNT OVERFLOW TO 2ND CARD
 NON-PACK MODE TRANSFER TO 000 ADDRESS

6.2 DATA RELIABILITY TEST

TESTING IS DONE ON BOTH ALPHANUMERIC AND BINARY (PACKED AND UNPACKED) DATA UTILIZING 80. CARD DECKS.

- A. ALPHANUMERIC
 REFERENCE THE ALPHANUMERIC TABLE IN THE PROGRAM LISTING (BEGINNING AT THE LABEL 'ALPCD:') FOR THE IMAGE CODES PUNCHED IN THE 80. COLUMNS OF THE 1ST CARD. EACH SUCCESSIVE CARD IN THE DECK USES THE SAME SEQUENCE OF CODES ROTATED ONE COLUMN TO THE LEFT. THE PACKED FORM OF THE IMAGE CODES FOLLOWS THE 'ALPCD:' TABLE.
- B. BINARY
 REFERENCE THE BINARY TABLE IN THE PROGRAM LISTING (BEGINNING AT THE LABEL 'BINCD:') FOR THE BINARY CODES PUNCHED IN THE 80. COLUMNS OF THE 1ST CARD. KEEP IN MIND THE ZONE ENCODING DISCUSSED IN SECTION 5, PARAGRAPH D). EACH SUCCESSIVE CARD IN THE DECK USES THE SAME SEQUENCE OF CODES ROTATED ONE COLUMN TO THE LEFT. THE PACKED FORM OF THE BINARY CODES FOLLOWS THE 'BINCD:' TABLE.

6.3 ERROR FUNCTION TESTS FOR M1000/M200/M1200

DATA LATE
 ERROR AND OFF-LINE BITS
 INTERRUPT ON OFF TO ON-LINE TRANSITION
 INPUT HOPPER EMPTY
 OUTPUT STACKER FULL
 PICK CHECK ERROR
 STACK CHECK ERROR
 END OF FILE AND HOPPER CHECK (M1200 ONLY)
 READ CHECK ERROR
 READING A SINGLE DATA PATTERN

6.4

F02

CHECKING OF CARDS WHICH HAVE ALL COLUMNS IDENTICALLY PUNCHED IS DONE, THUS ALLOWING SPECIFIC TYPES OF DATA FAILURES TO BE MORE EASILY STUDIED. THE PATTERN INPUT FROM THE USER IS STORED AND THEN EACH COLUMN OF EACH CARD IS COMPARED AGAINST IT. IF A DISCREPANCY OCCURS, THE ERROR IS PRINTED OUT, ALONG WITH THE TOTAL NO. OF CARDS READ AS WELL AS THE TOTAL NO. OF DATA ERRORS DISCOVERED UP TO THAT POINT. (NOTE: ALL PRINTOUTS ARE IN OCTAL). WHEN THE INPUT HOPPER BECOMES EMPTY, THE TERMINAL WILL ECHO A 'BELL'. THE PROGRAM WILL THEN WAIT FOR MORE CARDS TO BE LOADED AND THE CARD READER TO BE PUT BACK ON-LINE (I.E. PRESSING 'RESET' BUTTON). (REFERENCE SECTION 2.1, PARAGRAPH E.)

6.5 LOOPING ON A SELECTED TEST

THIS ALLOWS A SINGLE SUB-TEST TO BE RUN CONTINUOUSLY BY SELECTING THE TEST (INSTRUCTION PORTION ONLY) AND LOADING THE ADDRESS OF THE SCOPE INSTRUCTION AT THE BEGINNING OF THE TEST. (REFERENCE SECTION 2.1, PARAGRAPH D.)

7.0 FLOW CHARTS

8.0 PROGRAM LISTING

CD11/CD20 CARD READER DIAGNOSTIC

DEC FLO VER 00.1! 19-FEB-76 14:44 PAGE A

FLOW CHART

CD11/CD20 CARD READER DIAGNOSTIC

F02

COPYRIGHT 1976
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

TABLE OF CONTENTS

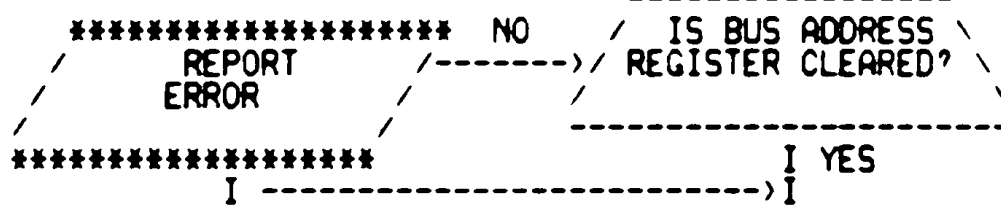
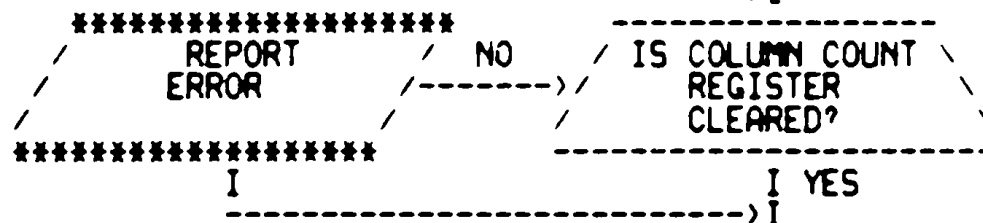
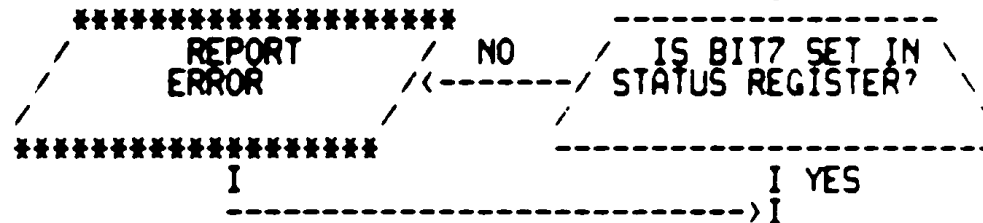
PAGE 02	TEST FOR INIT. OF ALL REGISTERS
PAGE 03	TEST READ/WRITE STATUS REGISTER
PAGE 04	TEST READ/WRITE OF COLUMN COUNT REGISTER
PAGE 05	TEST READ/WRITE OF BUS ADDRESS REGISTER
PAGE 06	TEST CONTROLLER READY TO CLEAR BIT0
PAGE 07	TEST BIT 2 TO BE CLEAR AFTER CARD READ
PAGE 08	TEST INTERRUPT FROM CONTROLLER READY
PAGE 09	TEST NO INTERRUPT ON CONTROLLER READY & CPU AT LEVEL 7
PAGE 10	TESTS FOR INTERRUPTS
PAGE 11	TEST FOR INTERRUPTS (CONT'D)
PAGE 12	SIMILTANEOUS INTERRUPTS AT MORE THAN ONE LEVEL
PAGE 13	NON-EXISTANT MEMORY DETECTION
PAGE 14	BYTE & DATIP LOAD OF COLUMN COUNT REGISTER
PAGE 15	WORD COUNT OVERFLOW TO 2ND CARD
PAGE 16	BUS ADDRESS ODD & TRANSFER IN NON-PACK MODE
PAGE 17	DATA RELIABILITY TESTING
PAGE 24	ERROR FUNCTION TESTING OF MODEL M1200
PAGE 40	PROGRAM TO LOOP ON SINGLE DATA PATTERN
PAGE 44	PROGRAM TO LOOP ON TEST
PAGE 45	INITIALIZATION ROUTINES

```
*****  
*BEGIN * START ADDRESS = 200  
*****  
I  
*****  
* VECTOR SETUPS AND *  
* HARDWARE/SOFTWARE *  
* "SWR" DETERMINATION *  
*****  
I  
I  
*****  
* INITIALIZE *  
* POINTERS AND *  
* FLAGS *  
*****  
I  
I  
*****  
* TYPE MAINDEC *  
* TITLE AND *  
* REV. LEVEL *  
*****  
I  
I  
*****  
*TST1(02) *  
*****
```

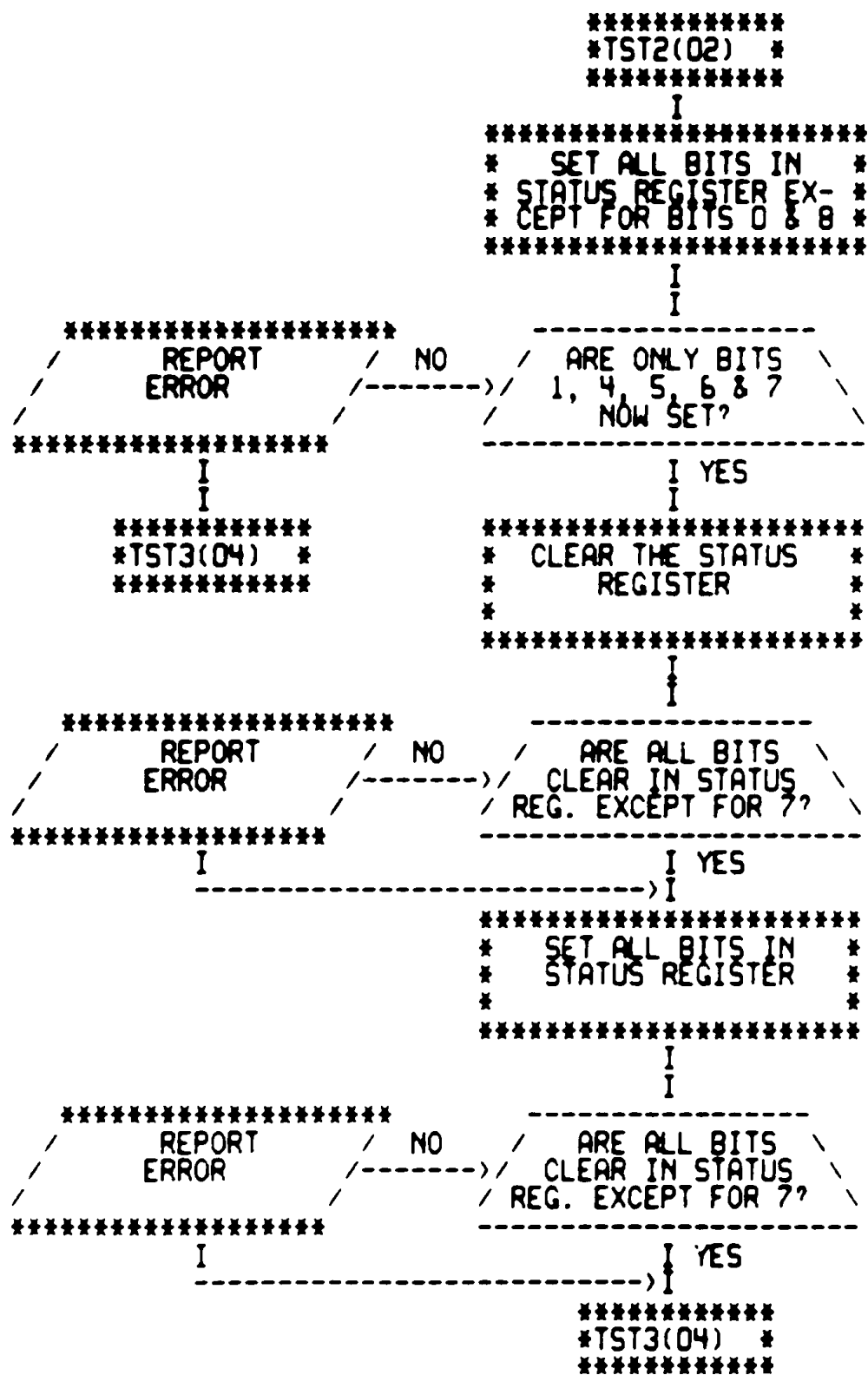
*TST1(01) *

** CKOFFL(45)*--> ** CKOFFL **

* SEND OUT AN *
* INITIALIZATION *
* PULSE (RESET) *



*TST2(03) *



*TST3(03) *

I

* LOAD ALL BITS OF *
* THE COLUMN COUNT *
* REGISTER *

I

I

/ REPORT / NO / IS THE CONTENTS /
ERROR /-----> / OF THE COLUMN /
***** / COUNT REG.=177777? /

I

I YES

/ REPORT / NO / IS BIT 7 SET IN /
ERROR /-----> / THE STATUS /
***** / REGISTER /

I

I YES

* ISSUE A POWER CLEAR *
* VIA THE STATUS *
* REGISTER *

I

I

/ REPORT / NO / IS THE CONTENTS /
ERROR /-----> / OF THE COLUMN /
***** / COUNT REG. NOW 0? /

I

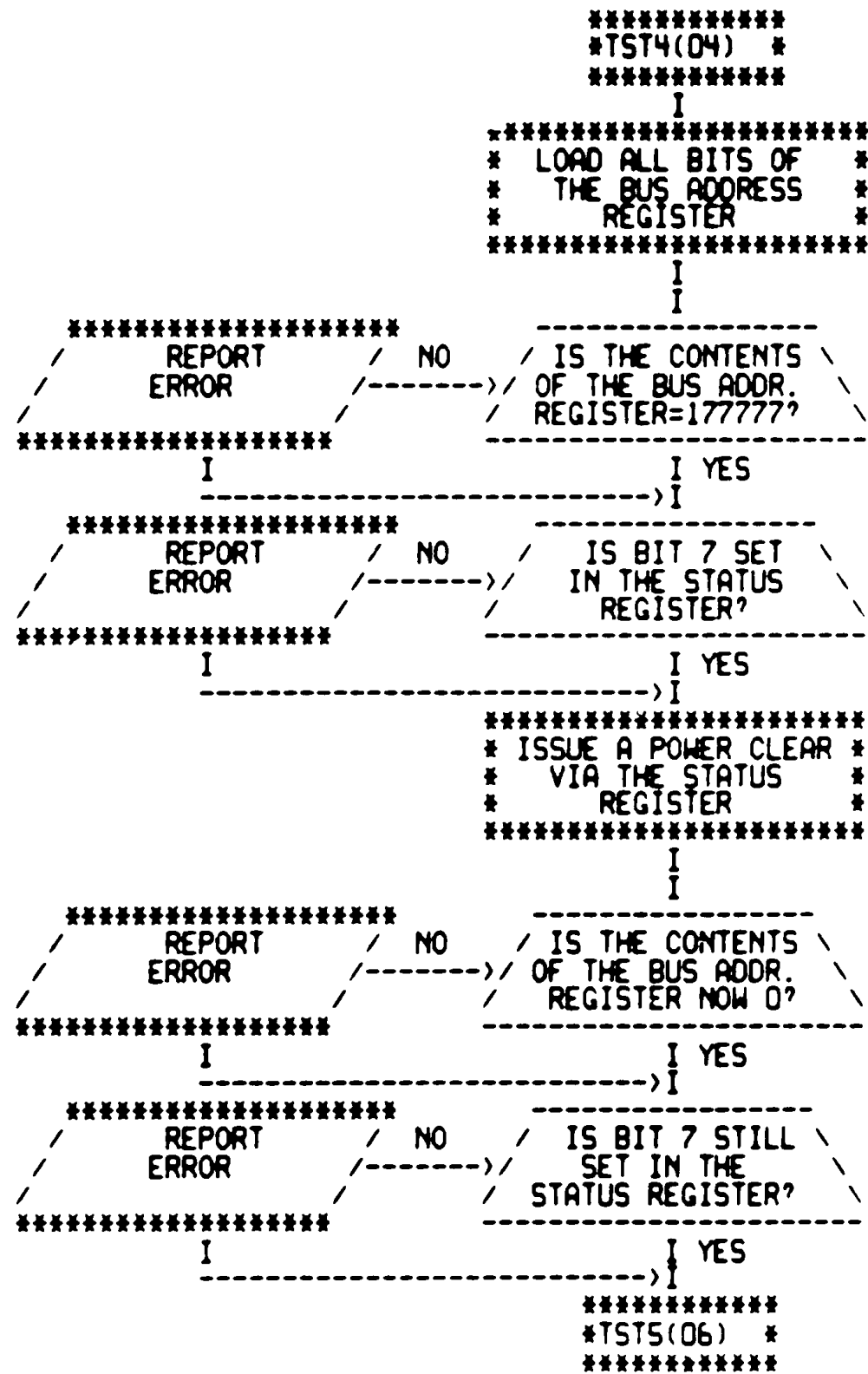
I YES

/ REPORT / NO / IS BIT 7 STILL /
ERROR /-----> / SET IN THE STATUS /
***** / REGISTER? /

I

I YES

*TST4(05) *




```
*****
*TST6A(07) *
*****
      I COUNT
*****
*INITIALIZE A COUNTER *
* TO WAIT FOR CON- *
* TROLLER READY *
*****
      I
*****
* STORE CURRENT PROC- *
* ESSOR STATUS & CLEAR *
* THE TRACE BIT (BIT4) *
*****
----->I
I / IS CONTROLLER \ YES *****
I / READY? \ ----->*TST6B(07) *
I \
I
I I NO *****
I *DECREMENT THE COUNTER*
I * USED TO WAIT FOR *
I * CONTROLLER READY *
I *****
I
I / HAS THE COUNTER \
I NO / OVERFLOWED TO 0? \
I \
I
I YES *****
I / REPORT *
I / ERROR *
I \
I *****
I
I *****
I *TST7(08) *
I *****
```

```
*****
*TST6(06) *
*****
      I
*****
**          **          *****
**   CKOFFL   **----->*CKOFFL(45)*
**          **          *****
*****
      I
*****
* CLEAR THE STATUS *
* REGISTER *
*****
      I
*****
* SET UP COLUMN COUNT *
* REGISTER TO READ *
* 20 COLUMNS *
*****
      I
*****
* SET UP MEMORY ADDR. *
* FOR COLUMN CONTENTS *
* TRANSFER *
*****
      I
*****
* START READING *
* A CARD *
*****
      I
*****
I / IS CONTROLLER \ YES *****
I / BUSY? \ ----->REPORT *
I \          \ ERROR *
I
I NO *****
I /
I /
I \
I *****
*****
*TST6A(07) *
*****
```

```
*****
*RESTORE PROCESSOR *
* STATUS TO WHAT IT *
* WAS BEFORE CARD READ *
*****
      I
-----
/ IS CONTROLLER \ NO *****
/ READY? \ ----->REPORT *
\          \ ERROR *
\
I YES *****
I /
I /
I \
I *****
I / IS ERROR BIT \ YES *****
I / (BIT15) SET IN \ ----->REPORT *
I / STATUS REGISTER? \ ERROR *
I \
I NO *****
I /
I /
I \
I *****
*****
*TST7(08) *
*****
```



```

*****
*TINT7(08) *
*****
I
-----
/ WAS CONTROLLER \ NO \
/  READY?       \---> \ REPORT
/                   \ /  ERROR
/                   \ /
-----
I YES
I <-----
*****
* RESET STACK *
* POINTER FROM *
* INTERRUPT *
*****
I
-----
/ DID ANY KIND \ YES \
/ OF ERROR SHOW \---> \ REPORT
/ UP?           \ /  ERROR
/                   \ /
-----
I NO
I <-----
*****
* DISABLE *
* INTERRUPTS *
*****
I
-----
I <-----
*****
* RESET TRAPCATCHER *
* VECTOR LOCATIONS 230 *
* AND 232 *
*****
I
*****
*TST10(09) *
*****

```

```

*****
*TST7(07) *
*****
I
*****
**          **
**   INIT   **-----> *INIT(45) *
**          **
*****
I
*****
* SET RETURN POINT *
* AND PS FOR WHEN AN *
* INTERRUPT OCCURS *
*****
I
*****
* SET CPU TO *
* PRIORITY *
* LEVEL 0 *
*****
I
*****
* SET COLUMN COUNT TO *
* 31(10) AND BUS ADDR. *
* TO "BUFBEQ" *
*****
I
*****
* SET INTERRUPT *
* ENABLE AND *
* READ *
*****
I
*****
* WAIT FOR *
* CONTROLLER *
* READY *
*****
I
-----
/ DID AN INTER- \ NO \
/  RUPT OCCUR? \---> \
/                   \ /
-----
I YES
*****
*TINT7(08) *
*****

```

```

*****
*INTN(08) *
*****
I
*****
*GIVE CONTROL BACK TO *
* CPU AND DISABLE *
* INTERRUPTS *
*****
I
-----
/ REPORT
/ ERROR
/
-----
*****
*CONT7 *
*****

```

```

*****
*TINT10(09)*
*****
I
*****
REPORT
ERROR
*****
I
*****
* RESET STACK *
* POINTER FROM *
* INTERRUPT *
*****
I<-----*T10GO *
I
*****
* DISABLE *
* INTERRUPTS *
*****
I
*****
* RESET TRAPCATCHER *
* VECTOR LOCATIONS 230 *
* AND 232 *
*****
I
*****
*TST11(10)*
*****

```

```

*****
*TST10(08)*
*****
I
*****
**          **          *****
**      INIT          **---->*INIT(45)*
**          **          *****
*****
I
*****
* SET RETURN POINT *
* AND PS FOR WHEN AN *
* INTERRUPT OCCURS *
*****
I
*****
* SET CPU TO *
* PRIORITY LEVEL *
*      7      *
*****
I
*****
* SET COLUMN COUNT TO *
* 6(10) AND BUS ADDR. *
* TO "BUFBE" *
*****
I
*****
* SET INTERRUPT *
* ENABLE AND *
* READ *
*****
I
*****
* WAIT FOR *
* CONTROLLER *
* READY *
*****
I
-----
/ DID AN \ NO          *****
/ INTERRUPT \----->*T10GO *
/ OCCUR ? \          *****
-----
I YES
*****
*TINT10(09)*
*****

```

```

*****
*TINT11(10)*
*****
I
-----
/ IS CONTROLLER \ NO
/ READY SET? \
-----
I YES
I<-----
*****
* DISABLE *
* INTERRUPTS *
*
*****
I
*****
* RESET TRAPCATCHER *
* VECTORS 230 & 232 *
*
*****
I
*****
* RESET STACK *
* POINTER FROM *
* INTERRUPT *
*****
I
-----
/ HAVE WE INTER- \ YES
/ RUPTED BEFORE AT \
/ ANOTHER LEVEL? \
-----
I NO
*****
* SET PREVIOUS INTER- *
* RUPT FLAG (INTFLG) & *
* STORE CURRENT LEVEL *
*****
I
*****
* TYPE MSG. INDICATING *
* WHAT THE INTERRUPT *
* LEVEL WAS *
*****
I
*****
* GO TO NEXT *
* TEST *
*
*****

```

```

*****
*TST11(09)*
*****
I
*****
**
** INIT **
**
*****
I
*****
* SET RETURN POINT *
* AND PS FOR WHEN AN *
* INTERRUPT OCCURS *
*****
I
*****
* SET CPU TO *
* PRIORITY *
* LEVEL 6 *
*****
I
*****
* SET COLUMN COUNT TO *
* 80(10) AND BUS ADDR. *
* TO "BUFBEQ" *
*****
I
*****
* SET INTERRUPT *
* ENABLE AND *
* READ *
*****
I
*****
* WAIT FOR *
* CONTROLLER *
* READY *
*****
I
-----
/ DID AN INTER- \ YES
/ RUPT OCCUR? \
-----
I NO
-----
/ IS CONTROLLER \ NO
/ READY SET? \
-----
I YES
*****
*PR70K(10)*
*****

```

```

*****
*PR70K(10)*
*****
I
*****
**
** GIVE FULL **
** CONTROL BACK **
** TO CPU **
*****
I
*****
* DISABLE *
* INTERRUPTS *
*
*****
I
*****
* RESET TRAPCATCHER *
* VECTOR LOCATIONS 230 *
* AND 232 *
*****
I
-----
/ HAVE WE INTER-? \ NO
/ RUPTED BEFORE AT \ \-> * GO ON TO NEXT *
/ ANOTHER LEVEL? \ * TEST *
-----
I YES
-----
/ WAS THE OTHER \ YES * REPORT INTER- *
/ LEVEL HIGHER? \ \-> * RUPT OCCURED *
\ \ * AT HIGHER LVL. *
-----
I NO
I<-----
*****
* GO ON TO NEXT *
* TEST *
*
*****

```

NOTE: TST12, AND TST13 ARE STRUCTURED SIMILARY AND WILL TEST CPU LEVELS 5 AND 4, RESPECTIVELY


```

-----*
TEST 14 THRU 17 ARE STRUCTURED SIMILARLY

TO THAT SHOWN FOR TEST 11 *****
                        *TST11(10) *
                        *****

TESTS 14 THRU 17 COVER CPU LEVELS 3,2,1 AND 0;
THE ONLY DIFFERENCE BEING THAT WE ARE NOW
LOOKING FOR AN INTERRUPT TO OCCUR SINCE THE CPU
LEVELS ARE BELOW THE DEVICE LEVEL OF 4.
-----*

```

```

*****
*TST20A(11)*
*****
I
-----*
/ DID AN INTERRUPT \ YES * REPORT ERROR *
/ FINALLY OCCUR? \ ---) * NO INTERRUPT *
\                   \ * SHOULD OCCUR *
-----*
I NO
*****
* GIVE CONTROL *
* BACK TO CPU *
*
*****
I<-----I
*****
* DISABLE *
* INTERRUPTS *
*
*****
I
*****
* RESET TRAPCATCHER *
* VECTOR LOCATIONS *
* 230 & 232 *
*****
I
*****
*TST21(12) *
*****

```

```

*****
*TST20 *
*****
I
-----*
**
** INIT **-----)*INIT(45) *
**
**
*****
I
*****
*SET RETURN POINT AND *
* PS FOR WHEN AN *
* INTERRUPT OCCURS *
*****
I
*****
* SET CPU TO *
* PRIORITY LEVEL *
* 0 *
*****
I
*****
* SET COLUMN COUNT TO *
* 1(10) AND BUS ADDR. *
* TO "BUFBEQ" *
*****
I
*****
* ENABLE *
* INTERRUPTS *
*
*****
I
*****
* WAIT AWHILE TO *
* SEE IF AN INTERRUPT *-----)*TST20A(11)*
* OCCURS *
*****

```



```

*****
*TINT22(13)*
*****
I
*****
* RESTORE TRAPCATCHER *
* VECTOR LOCATIONS 230 *
*      & 232          *
*****
I
-----
/ IS CONTROLLER \ NO
/   READY?      \
-----
I YES
I<-----
/ IS ERROR BIT 5 \ NO
/   SET?         \
-----
I YES
I<-----
/ IS NXM BIT 09 \ NO
/   SET?        \
-----
I YES
I<-----
/ IS EXTENDED \ NO
/ MEMORY BIT 17 \
/   SET?      \
-----
I YES
I<-----
/ IS EXTENDED \ NO
/ MEMORY BIT 16 \
/   SET?      \
-----
I YES
I<-----
*****
*T22A(13) *
*****

```

```

*****
*TST22(12) *
*****
I
*****
**
**      INIT      **
**
*****
** SET RETURN POINT AND **
** PS FOR WHEN AN      **
** INTERRUPT OCCURS    **
*****
I
*****
* SET CPU TO *
* PRIORITY LEVEL *
*      0      *
*****
I
*****
* SET COLUMN *
* COUNT TO READ *
* 5(10) COLUMNS *
*****
I
*****
* SET BUS ADDRESS TO *
* NON-EXISTANT MEMORY *
* I.E. LOC. 160000 *
*****
I
*****
* SET INTERRUPT *
* ENABLE, READ & *
* EXT. MEM. BITS *
*****
I
-----
/ DID AN \ NO
/ INTERRUPT OCCUR? \
-----
I YES
*****
* RESTORE STACK *
* POINTER FROM *
* INTERRUPT *
*****
I
*****
*TINT22(13)*
*****

```

```

*****
*T22A(13) *
*****
I
-----
/ ANY OTHER ERROR \ YES * REPORT *
/ BITS SET ? E.G. \ -> * ERROR *
/ BITS 2,10,1.,ETC. \ *
-----
I NO
I<-----
/ DOES BUS ADDR. \ NO * REPORT *
/ REGISTER CONTENTS \ -> * ERROR *
/ =160002 ? \ *
-----
I YES
I<-----
/ DOES COLUMN CNT \ NO * REPORT *
/ REGISTER SHOW 4 \ -> * ERROR *
/ COLUMNS LEFT ? \ *
-----
I YES
I<-----
*****
*TST23(14) *
*****

```

H03

```

*****
*TST23(13) *
*****
      I
*****
* INITIALIZE *
* COLUMN COUNT *
* REGISTER TO 0 *
*****
      I
*****
* LOAD LOWER BYTE OF *
* COLUMN COUNT REG. *
* WITH THE VALUE 252 *
*****
      I
-----
/ DID UPPER BYTE \NO * REPORT *
/ GET LOADED WITH \-->* ERROR *
/ THE VALUE 252 ALSO? \ *
-----
      I YES
      I<-----I
*****
*TST24(14) *
*****

*****
*TST24(14) *
*****
      I
*****
* INITIALIZE *
* COLUMN COUNT *
* REGISTER TO 0 *
*****
      I
*****
* LOAD HIGH BYTE OF *
* COLUMN COUNT REG. *
* WITH THE VALUE 252 *
*****
      I
-----
/ DID LOWER BYTE \ * REPORT *
/ GET LOADED WITH \-->* ERROR *
/ THE VALUE 252 ALSO? \ *
-----
      I<-----I
*****
*TST25(14) *
*****

*****
*TST25(14) *
*****
      I
*****
* INITIALIZE *
* COLUMN COUNT *
* REGISTER TO 0 *
*****
      I
*****
* LOAD COLUMN COUNT *
* REGISTER WITH 10000 *
* AND NEGATE IT *
*****
      I
-----
/ DID CONTENTS OF \YES * REPORT *
/ COLUMN COUNT REG. \-->* ERROR *
/ CHANGE ? \ *
-----
      I NO
      I<-----I
*****
*TST26 *
*****

```

```

*-----*
NOTE: TESTS 26, 27 AND 30 ARE CARBON COPIES OF TESTS 23,
      24, AND 25, RESPECTIVELY. ONLY DIFFERENCE BEING
      THAT TESTS 23 - 25 OPERATE ON COLUMN COUNT REGISTER,
      AND TESTS 26 - 30 OPERATE ON BUS ADDRESS REGISTER.
*-----*

```



```
*****
*TINT31(15)*
*****
I
-----
/ IS CONTROLLER \ NO
/  READY SET ?  \ ---->
\-----/
I YES
I <-----
*****
*  DISABLE  *
*  FURTHER  *
*  INTERRUPTS *
*****
I
*****
* RESTORE TRAPCATCHER *
* VECTOR LOCATIONS 230 *
*      AND 232      *
*****
I
*****
*  RESET STACK  *
*    FROM      *
*  INTERRUPT    *
*****
I
-----
/  HERE 2 \ NO
/  CARDS READ? \ ---->
\-----/
I
I <-----
*****
*TST32(16) *
*****

*****
*TST31 *
*****
I
*****
*  INIT  *
*****
I
*****
* SET UP RETURN *
* ADDR. AND PS *
* FOR INTERRUPT *
*****
I
*****
* SET PROCESSOR *
*   TO LEVEL   *
*     0        *
*****
I
*****
* SET COLUMN COUNT TO *
* 81(10) I.E.-WRAP *
* AROUND TO 2ND CARD *
*****
I
*****
* SET A START ADDRESS *
* FOR DUMP OF CARD *
*   CONTENTS          *
*****
I
*****
* SET INTERRUPT *
* ENABLE AND *
*   READ      *
*****
I
-----
/ DID WE GET AN \ YES
/  INTERRUPT?  \ ---->
\-----/
I NO
*****
*TINT31(15)*
*****

*****
*INIT(45) *
*****

*****
*T23G0 *
*****
I
*****
* GIVE CONTROL *
*   BACK TO   *
*   CPU      *
*****
I
*****
*  DISABLE  *
*  INTERRUPTS *
*****
I
*****
* RESTORE TRAPCATCHER *
* VECTOR LOCATIONS 230 *
*      &232          *
*****
I
*****
/  REPORT \
/  ERROR  \
\-----/
I
*****
*TST32(16) *
*****

*****
*T31G0 *
*****
```

```

*****
*ENOCK(16) *
*****
I
-----
/ IS SNK07 > SET? \ YES -----> *DATST(17) *
\-----
I
*****
* DING *
* A *
* LING? *
*****
I
*****
*RESTRT *
*****

```

```

*****
*TST32(15) *
*****
I
*****
**          ** -----> *INIT(45) *
**      INIT          **
**          **
*****
I
*****
* SET UP COLUMN COUNT *
* TO READ 1(10) COLUMN *
*          *
*****
I
*****
* SET BUFFER ADDRESS, *
* FOR COLUMN DUMP, TO *
* ODD ADDRESS *
*****
I
*****
* SET RETURN PC & PS *
* FOR ILLEGAL INSTR. *
* TRAP (LOC. 10) *
*****
I
*****
* READ A *
* CARD *
*          *
*****
I
-----
/ DID WE GET AN \ NO \-----> REPORT
/ ILLEGAL \-----> ERROR
/ INSTRUCTION TRAP? \-----
-----
I
*****
* RESET STACK *
* FROM ILLEGAL *
* INSTR. TRAP *
*****
I
-----
*****
* RESTORE TRAP- *
* CATCHER FOR *
* LOCS. 10 & 12 *
*****

```

```

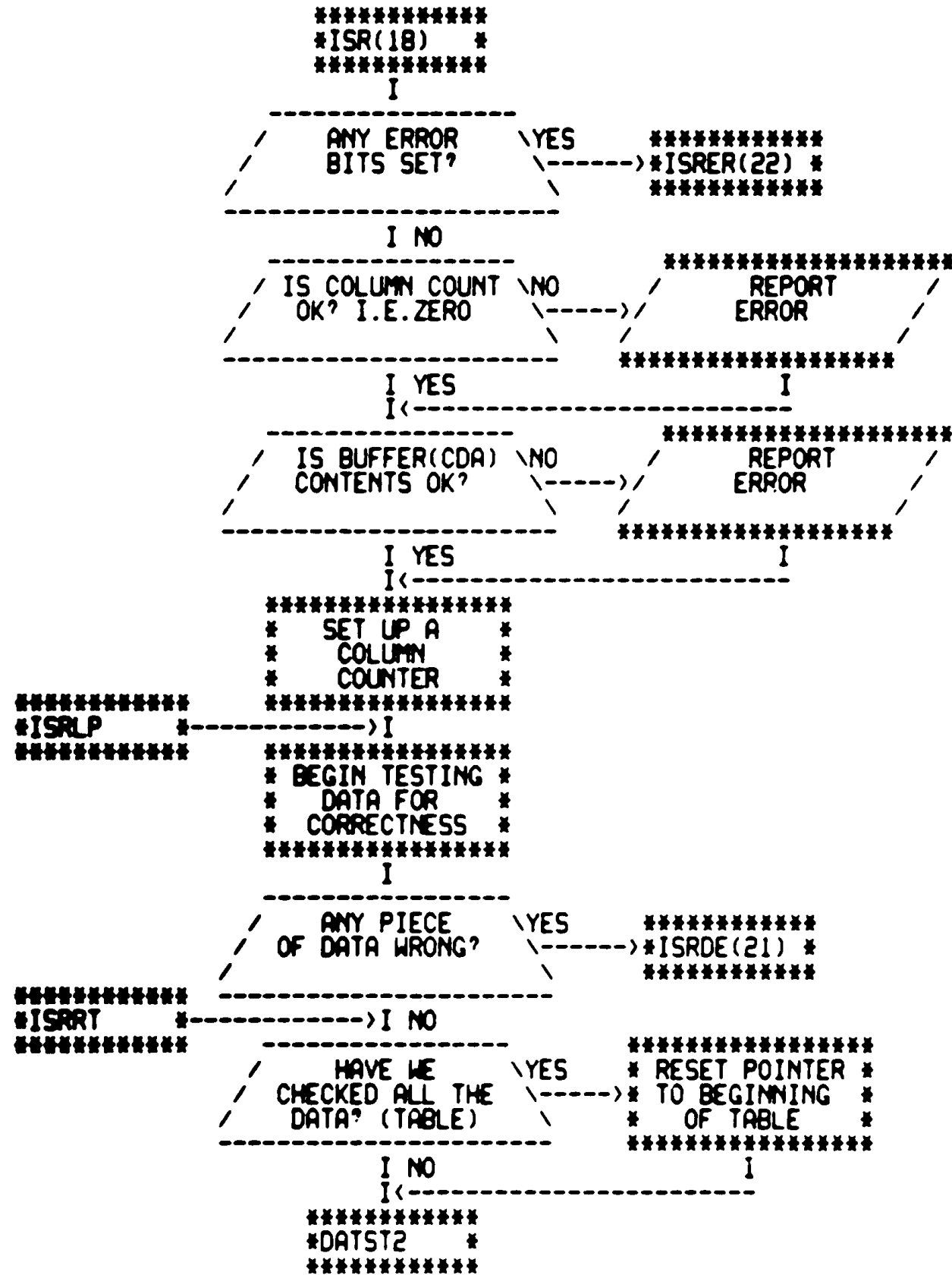
*****
*ENOCK(16) *
*****

```

K03

```
*****
#DATST(16) * UNPACKED MODE EXAMPLE
*****
I
/*****
TYPE LEAD-IN
POSITIONAL
MESSAGES
*****/
I
*****
* INITIALIZE CARD *
* COUNT AND COLUMN *
* COUNT TO ZERO *
*****
I
-----
/ ARE WE TESTING \ YES
A \-----> * LOAD BINARY DATA *
/ BINARY DECK ? \ * TABLE POINTERS *
-----
I NO
*****
* LOAD ALPHANUMERIC *
* DATA TABLE *
* POINTERS *
*****
I<-----
*****
** INIT **-----> *INIT(45) *
*****
I
*****
* SET UP RETURN ADDR. *
* FOR INTERRUPT *
* SERVICING *
*****
I
*****
* SET CARD *
* SIZE AND WORK- *
* ING OFFSET *
*****
I
*****
#DATST1(18)*
*****
```

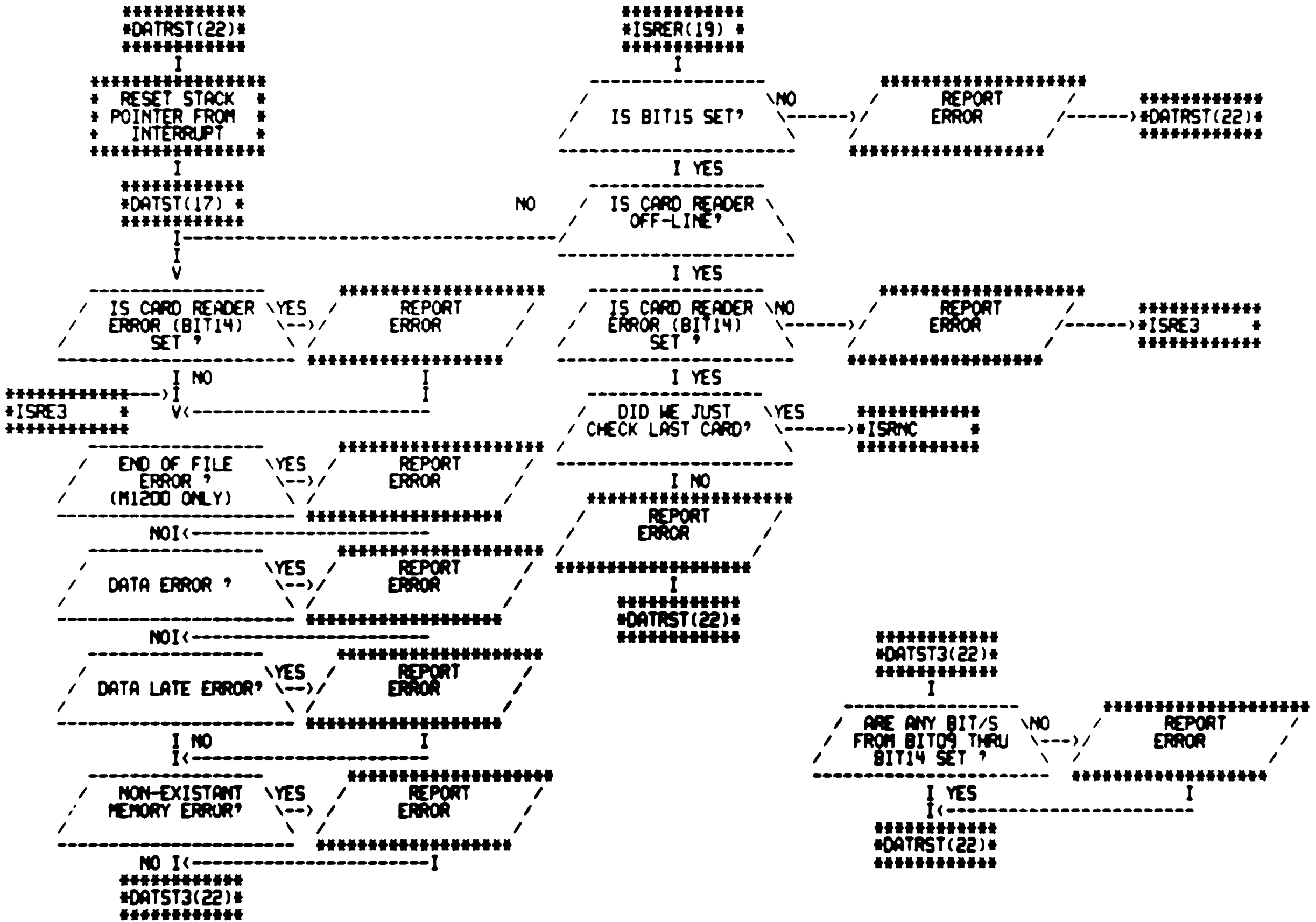
```
*****
#DATST1(17)*
*****
I
*****
#SET COL. COUNT = 80. #
# SET GPR = RD FOR #
# ADDR. SELECTION #
*****
I
*****
# ENABLE #
# INTERRUPTS #
*****
I
-----
/ DO WE WANT PACK \ \NO
/ MODE? \
-----
I YES
*****
# SET UP FOR # NO / WAS IMAGE MODE \
# ACCEPTANCE OF * <----- / SELECTED ALSO? \
# PACK MODE *
*****
I
-----
> I <-----
I
*****
# READ #
# A #
# CARD #
*****
#BKGND * <----- > I <-----
*****
/ HAVE WE \ \NO
/ FINISHED READING \
/ A CARD YET ? \
-----
I YES
-----
/ IS CONTROLLER \ \NO / REPORT
/ READY ? \ / ERROR
-----
I YES I
I <----- I
-----
*****NO / HAD WE SELECTED \ YES *****
#ISR(19) * <----- / DATA PACK MODE? \ -----> *PSR *
*****
```

```
*****  
*SRETRN(20)*  
*****  
I  
*****  
* CALCULATE NEW SIZE *  
* OF COLUMNS TO BE *  
* READ *  
*****  
I  
*****  
* RESET CARD READER *  
* BUFFERS AS PER *  
* PRESENT POSITION *  
*****  
I  
*****  
* READ NEXT *  
* CARD *  
*****  
I  
*****  
*BKGND *  
*****
```

```
*****  
*DATST2 *  
*****  
-----  
/ HAVE WE REACHED \YES  
/ THE END OF THE ----->  
/ MEMORY BUFFER ? \  
-----  
I NO  
*****  
* UPDATE *  
* COLUMN *  
* COUNT *  
*****  
I  
*****  
* UPDATE TABLE *  
* OFFSET FOR *  
* CARD #1 IN DECK *  
*****  
I  
-----  
/ HAVE WE LOOKED \NO  
/ AT LAST COLUMN OF ----->  
/ DECK ? \  
-----  
I YES  
*****  
* STEP UP TO *  
* NEXT *  
* CARD *  
*****  
I  
*****  
* UPDATE TABLE *  
* POINTER FOR *  
* NEXT CARD *  
*****  
I  
*****  
*ISRLP *  
*****
```

```
*****  
* STEP UP TO *  
* NEXT CARD *  
*****  
I  
-----  
/ IMAGE MODE \NO * SET UP FOR *  
/ SELECTED ? \-----> * PACKING MODE *  
-----  
I YES I  
I<-----  
*****  
*SRETRN(20)*  
*****  
*****  
*ISRLP *  
*****
```

DATA RELIABILITY TESTING FOR PACKED MODE WILL

START AT *****
 *DATST(17) * & BRANCH OFF TO*PSR *

WHERE DATA WILL BE HANDLED AS OUTLINED IN

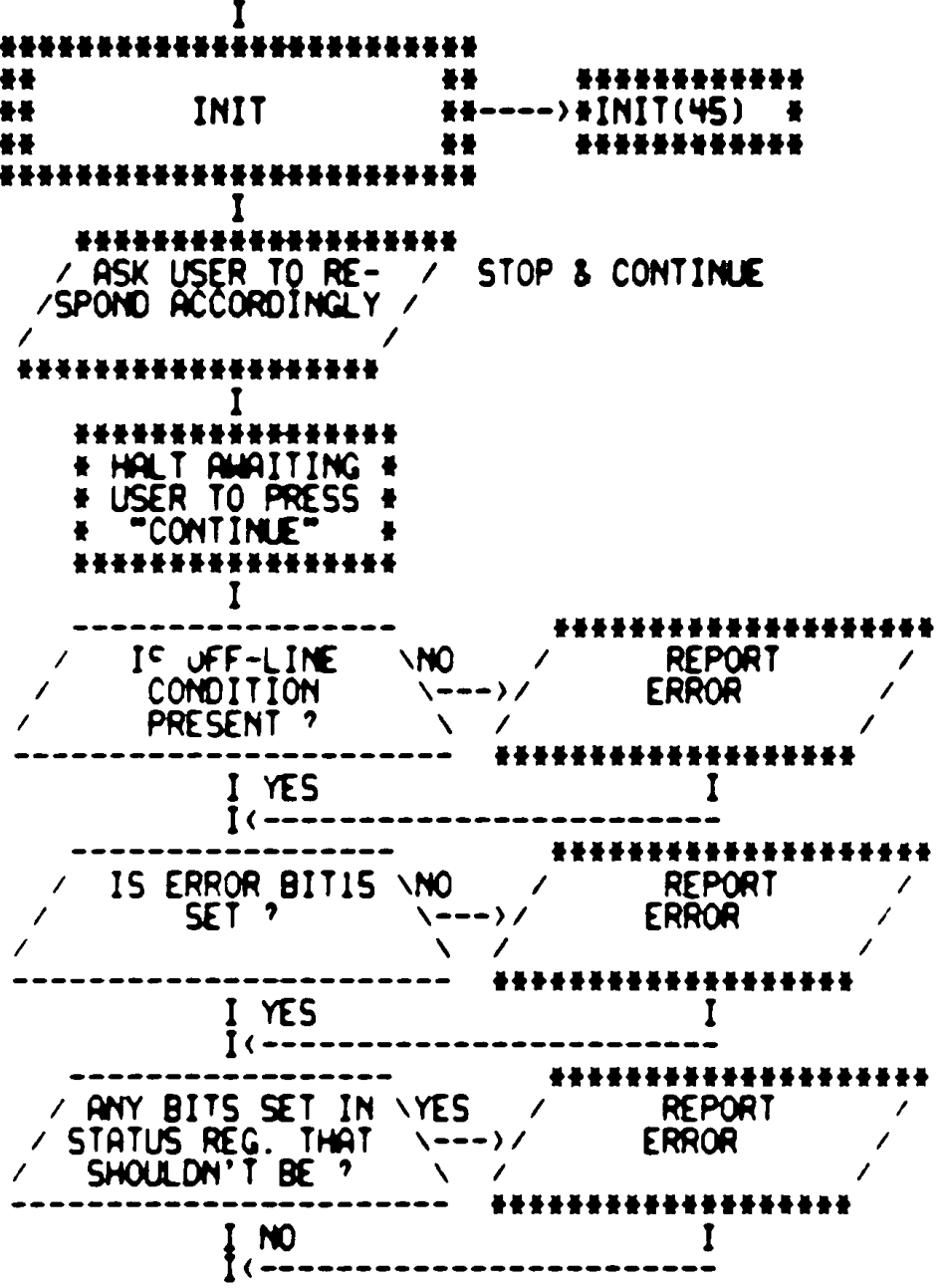
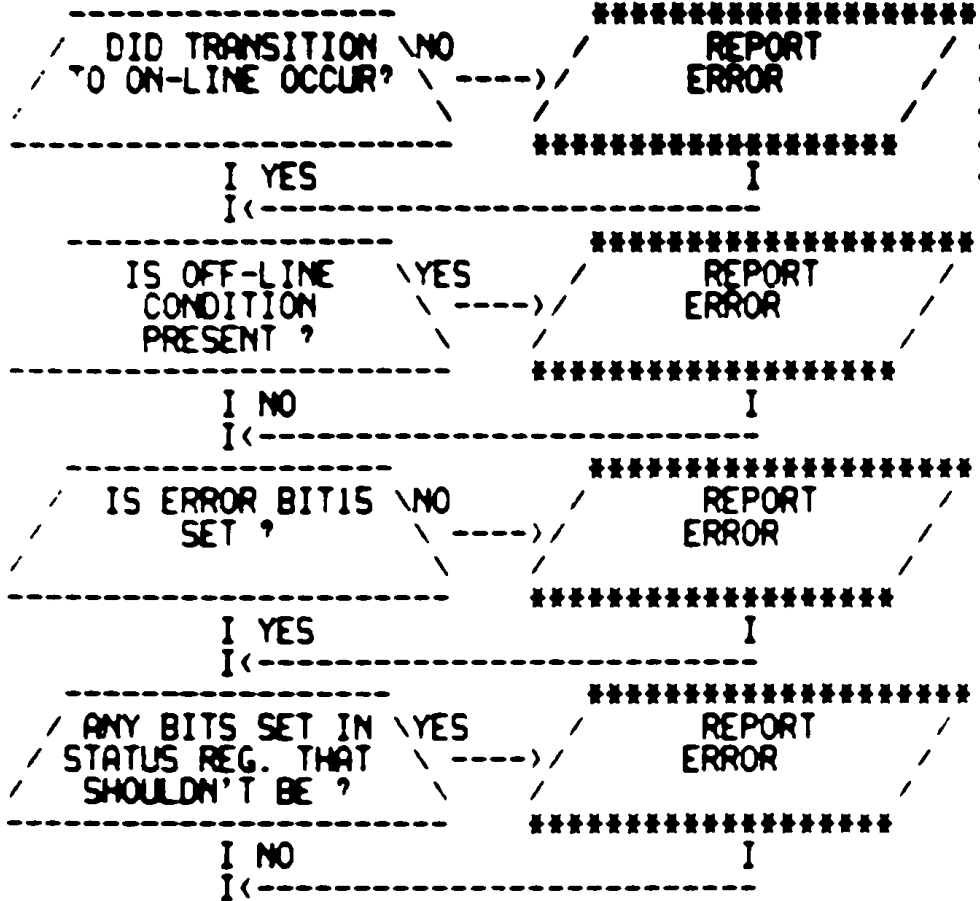
SECTION STARTING AT *****
 *ISR(19) *WITH ONLY ONE

EXCEPTION: DATA IS HANDLED USING BYTE CONSTRUCTION

F04

```
*****
*TST34A *
*****
```

```
*****
*TST34(24) * ERROR + OFF-LINE
*****
```



```

*****
RESET & CONTINUE / ASK USER TO RE- / * HALT AWAITING *          *****
                  \ SPOND ACCORDINGLY \ * USER TO PRESS * ----> *TST34A *
                  / "CONTINUE" / * "CONTINUE" *          *****
*****

```

```
*****
*TST35(26) *
*****
```



```
*****  
*TST35C(27)*  
*****  
I  
*****  
* RESET STACK *  
* POINTER FROM *  
* INTERRUPT *  
*****  
I  
-----  
/ DID TRANSITION \ NO  
TO ON-LINE OCCUR? \-----> /  
-----  
I YES  
I<-----  
-----  
/ IS OFF-LINE \ YES  
CONDITION STILL \-----> /  
PRESENT  
-----  
I NO  
I<-----  
-----  
/ IS ERROR BIT15 \ NO  
SET ? \-----> /  
-----  
I YES  
I<-----  
-----  
/ ARE ANY BITS SET \ YES  
THAT SHOULDN'T \-----> /  
BE  
-----  
I NO  
I<-----  
-----  
*****  
*TST36(28) *  
*****
```

```
*****  
*TST35B(26)*  
*****  
I  
-----  
/ OFF-LINE \ NO  
CONDITION \-----> /  
PRESENT ?  
-----  
I YES  
I<-----  
-----  
/ IS ERROR BIT15 \ NO  
SET ? \-----> /  
-----  
I YES  
I<-----  
-----  
/ ARE ANY BITS SET \ YES  
THAT SHOULDN'T \-----> /  
BE ?  
-----  
I NO  
I<-----  
-----  
*****  
* SET RETURN PC & PS *  
* FOR WHEN AN INTER- *  
* RUPT OCCURS *  
*****  
I  
*****  
* SET CPU TO *  
* PRIORITY LEVEL *  
* 0 *  
*****  
I  
*****  
/ ASK USER TO RE- / PRESS RESET  
/ SPOND ACCORDINGLY /  
-----  
I  
-----  
/ DID AN INTER- \ NO  
RUPT OCCUR ? \-----> /  
-----  
*****  
* HANG AWAITING *  
* INTERRUPT *  
* *  
*****
```

```
***** YES  
*TST35C(27)*  
*****
```



```
*****  
*TST36A(28)*  
*****  
I  
*****  
* SET RETURN PC & PS *  
*FOR WHEN AN INTERRUPT*  
* OCCURS *  
*****  
I  
*****  
* SET CPU TO *  
*PRIORITY LEVEL *  
* 0 & SET "IE" *  
*****  
I  
*****  
/ ASK USER TO / RESTORE CARDS  
RESPOND ACCORDING- / & RESET  
LY /  
*****  
I  
-----  
/ DID AN INTERRUPT \ NO  
OCCUR ? /-----> * HANG AWAITING *  
* INTERRUPT *  
* *  
-----  
I YES  
*****  
* RESET STACK *  
* POINTER FROM *  
* INTERRUPT *  
*****  
I  
*****  
* SET RETURN PC & PS *  
*FOR WHEN AN INTERRUPT *  
* OCCURS *  
*****  
I  
*****  
* SET CPU LEVEL *  
* TO 0 & INIT. *  
*COL.CNT & ADDR.*  
*****  
I  
-----  
/ DID AN INTERRUPT \ NO  
OCCUR ON ATTEMPT /-----> * HANG AWAITING *  
TO READ A CARD ? / * INTERRUPT *  
* *  
-----  
I YES  
*****  
*TST36B(28)*  
*****
```

```
*****  
*TST36(27) * INPUT HOPPER EMPTY  
*****  
I  
*****  
** INIT ** *INIT(45) *  
** ** * *  
*****  
I  
*****  
/ ASK USER TO RE- REMOVE CARDS  
SPOND ACCORDINGLY / & CONTINUE  
*****  
I  
*****  
* HALT AWAITING *  
* USER TO PRESS *  
* CONTINUE *  
*****  
I  
-----  
/ IS OFF-LINE \ NO  
CONDITION PRESENT? /-----> * REPORT *  
* ERROR *  
* *  
-----  
I YES  
I<-----  
-----  
/ IS ERROR BIT15 \ NO  
SET ? /-----> * REPORT *  
* ERROR *  
* *  
-----  
I YES  
I<-----  
-----  
/ IS CARD READER \ NO  
ERROR BIT14 SET? /-----> * REPORT *  
* ERROR *  
* *  
-----  
I YES  
I<-----  
-----  
/ ANY BITS SET \ YES  
THAT SHOULDN'T /-----> * REPORT *  
BE ? / * ERROR *  
* *  
-----  
I NO  
I<-----  
*****  
*TST36A(28)*  
*****
```

```
*****  
*TST36B(28)*  
*****  
I  
*****  
* RESET STACK *  
* POINTER FROM *  
* INTERRUPT *  
*****  
I  
-----  
/ IS CARD READER \ NO * REPORT  
STATUS = 300 ? / * ERROR *  
* *  
-----  
I YES  
I<-----  
*****  
*TST37(29) *  
*****
```

J04

```
*****
*TST37A(29)*
*****
I
*****
* SET RETURN PC & PS *
* FOR AN INTERRUPT *
* OCCURRENCE *
*****
I
*****
*SET CPU LEVEL= *
*0 AND SET "IE" *
* BIT *
*****
I
/ ASK USER TO RE-
/SPOND ACCORDINGLY / RESET
*****
I
-----
/ DID INTERRUPT \NO * HANG AWAITING *
/ OCCUR ? \-> * INTERRUPT *
-----
I YES
*****
* RESET STACK *
* POINTER FROM *
* INTERRUPT *
*****
I
*****
* SET RETURN PC & PS *
* FOR AN INTERRUPT *
* OCCURRENCE *
*****
I
*****
* SET CPU LEVEL TO 0 *
* AND INIT. COL. COUNT *
* & BUS ADDR. REGS. *
*****
I
-----
/ DID INTERRUPT \NO * HANG AWAITING *
/ OCCUR ON READING \-> * INTERRUPT *
/ A CARD ? \ *
-----
I YES
*****
*TST37B(29)*
*****
```

```
*****
* HANG AWAITING *
* INTERRUPT *
*****
I YES
*****
* HANG AWAITING *
* INTERRUPT *
*****
I YES
*****
* HANG AWAITING *
* INTERRUPT *
*****
```

```
*****
*TST37(28) *
*****
I
*****
** INIT **
**----->*INIT(45) *
**
*****
I
*****
/ ASK USER TO RE-
/SPOND ACCORDINGLY /
*****
I
*****
* HALT AWAITING *
* USER TO PRESS *
* CONTINUE *
*****
I
-----
/ IS OFF-LINE \NO
/ CONDITION \-----> / REPORT
/ PRESENT ? \ ERROR
-----
I YES
I<-----
-----
/ IS ERROR BIT15 \NO
/ SET ? \-----> / REPORT
/ ERROR
-----
I YES
I<-----
-----
/ IS CARD READER \NO
/ ERROR BIT14 SET? \-----> / REPORT
/ ERROR
-----
I YES
I<-----
-----
/ ARE ANY EXTRA \YES
/ BITS SET THAT \-----> / REPORT
/ SHOULDN'T BE ? \ ERROR
-----
I NO
I<-----
*****
*TST37A(29)*
*****
```

OUTPUT STACKER FULL

PRESSURE ARM
DOWN & CONTINUE

```
*****
*TST37B(29)*
*****
I
*****
* RESET STACK *
* POINTER FROM *
* INTERRUPT *
*****
I
-----
/ IS CONTENTS OF \NO * REPORT
/ STATUS REGISTER \ * ERROR
/ = 300 ? \ *
-----
YESI<-----I
*****
*TST40(30) *
*****
```

```
*****
/ IS OFF-LINE \NO
/ CONDITION \-----> / REPORT
/ PRESENT ? \ ERROR
-----
I YES
I<-----
-----
/ IS ERROR BIT15 \NO
/ SET ? \-----> / REPORT
/ ERROR
-----
I YES
I<-----
-----
/ IS CARD READER \NO
/ ERROR BIT14 SET? \-----> / REPORT
/ ERROR
-----
I YES
I<-----
-----
/ ARE ANY EXTRA \YES
/ BITS SET THAT \-----> / REPORT
/ SHOULDN'T BE ? \ ERROR
-----
I NO
I<-----
```

KOH

```

*****
I
/ IS OFF-LINE \ \NO
/ CONDITION PRESENT? \-----> / REPORT
/ \ \-----> / ERROR
/ \ \-----> /
/ \ \-----> /
I YES I
I<-----> I
/ IS ERROR BIT15 \ \NO
/ SET? \-----> / REPORT
/ \ \-----> / ERROR
/ \ \-----> /
I YES I
I<-----> I
/ IS CARD READER \ \NO
/ ERROR BIT14 SET? \-----> / REPORT
/ \ \-----> / ERROR
/ \ \-----> /
I YES I
I<-----> I
/ HAS ECO #14 \ \NO
/ BEEN INSTALLED ON \ \-----> / REPORT
/ THIS CARD READER? \ \-----> / ERROR
/ \ \-----> /
I YES I
I<-----> I
/ HAS A PICK CHECK \ \NO
/ ERROR BEEN \ \-----> / REPORT
/ DETECTED? \ \-----> / ERROR
/ \ \-----> /
I YES ***** I
I<-----TST40B-----> I
I ***** I
I
/ ARE ANY BITS SET \ YES
/ IN STATUS REGISTER \ \-----> / REPORT
/ THAT SHOULDN'T BE? \ \-----> / ERROR
/ \ \-----> /
I NO I
I<-----> I
*****
*TST40C(31)*
*****

```

```

***** KOH *****
I
*****
** INIT **
**-----> *INIT(45) *
**
*****
I
*****
/ ASK USER TO RE- / HOLD CAP UNDER SWITCH
/ SPOND ACCORDINGLY / DOWN & CONTINUE
/ \ \-----> /
/ \ \-----> /
I
*****
* HALT AWAITING *
* USER TO PRESS *
* CONTINUE *
*****
I
/ IS OFF-LINE \ \NO
/ CONDITION NOW \ \-----> / REPORT
/ PRESENT? \ \-----> / ERROR
/ \ \-----> /
I YES I
I<-----> I
/ DID WE GET A \ \NO
/ TRANSITION TO \ \-----> * WAIT FOR IT *
/ ON-LINE? \ \-----> *
/ \ \-----> *
I YES I
I<-----> I
/ IS CONTENTS OF \ \NO
/ STATUS REGISTER = \ \-----> / REPORT
/ 140210? \ \-----> / ERROR
/ \ \-----> /
I YES I
I<-----> I
*****
* SET COLUMN COUNT *
* BUS ADDR. AND READ *
* A CARD *
*****
I
*****
*TST40A(30)*
*****

```

```

*****
*TST400(31)*
*****
I
-----
/ DID AN INTERRUPT \ NO
/ OCCUR ? \ ----> * HANG AWAITING *
                * INTERRUPT *
                *
                *****
I YES
*****
* RESET STACK *
* POINTER FROM *
* INTERRUPT *
*****
I
-----
/ IS CONTENTS OF \ NO
/ STATUS REGISTER = \ ---> / REPORT
300 ? \ / ERROR
                /
                /
I YES
I <----- I
*****
*TST41(32) *
*****

```

```

*****
*TST40C(30)*
*****
I
*****
*SET RETURN TO PC & PS*
* FOR WHEN AN INTER- *
* RUPT OCCURS *
*****
I
*****
* SET CPU TO LEVEL 0 *
* AND SET "IE" BIT *
*
*
*****
I
*****
/ ASK USER TO RE- / RESTORE CARDS
/ SPOND ACCORDINGLY / AND RESET
*****
I
-----
/ DID INTERRUPT \ NO
/ OCCUR ? \ ----> * HANG AWAITING *
                * INTERRUPT *
                *
                *****
I
*****
* RESET STACK *
* POINTER FROM *
* INTERRUPT *
*****
I
*****
* SET RETURN PC & PS *
* FOR WHEN AN INTERRUPT *
* OCCURS *
*****
I
*****
* SET CPU LEVEL TO 0 *
* AND INIT. COL. COUNT *
* & BUS ADDR. REGS. *
*****
I
*****
* READ A *
* CARD *
*
*
*****
I
*****
*TST400(31)*
*****

```

M04

TST41A(32)

TST41(31)

STACK CHECK

/ IS OFF-LINE \ NO
/ CONDITION PRESENT \-----> / REPORT
? / ERROR /

** INIT **
**-----> *INIT(45) *
**-----

I YES
I<-----

/ IS ERROR BIT15 \ NO
/ SET ? \-----> / REPORT
? / ERROR /

/ ASK USER TO RES- / BLOCK PHOTOCELL
/ POND ACCORDINGLY / AND RESET

I YES
I<-----

/ IS CARD READER \ NO
/ ERROR BIT14 SET ? \-----> / REPORT
? / ERROR /

/ HAS OFF-LINE \ NO
/ OCCURRED YET? \-----> * AWAIT *
? / * OFF-LINE *
* CONDITION *

I YES
I<-----

/ HAS ECO #14 BEEN \ NO
/ INSTALLED ON THIS \-----> *TST41B *
/ CARD READER ? \-----> *
? / *

I YES
/ HAS TRANSITION \ NO
/ TO ON-LINE OCCUR- / * AWAIT *
/ ED YET ? \-----> * TRANSITION TO *
? / * ON-LINE *

I YES
I<-----

/ DID WE DETECT A \ NO
/ STACK CHECK ERROR \-----> / REPORT
? / ERROR /

/ IS CONTENTS OF \ NO
/ STATUS REGISTER \-----> / REPORT
= 100210 ? \-----> / ERROR /
? /

I YES
I<-----

/ ARE ANY BITS SET \ YES
/ IN STATUS REGISTER \-----> / REPORT
/ THAT SHOULDN'T BE? \-----> / ERROR /
? /

I YES
I<-----

* SET UP COL. COUNT & *
* BUS ADDR. REG. THEN *
* READ A CARD *

I YES
I<-----

/ ARE ANY BITS SET \ YES
/ IN STATUS REGISTER \-----> / REPORT
/ THAT SHOULDN'T BE? \-----> / ERROR /
? /

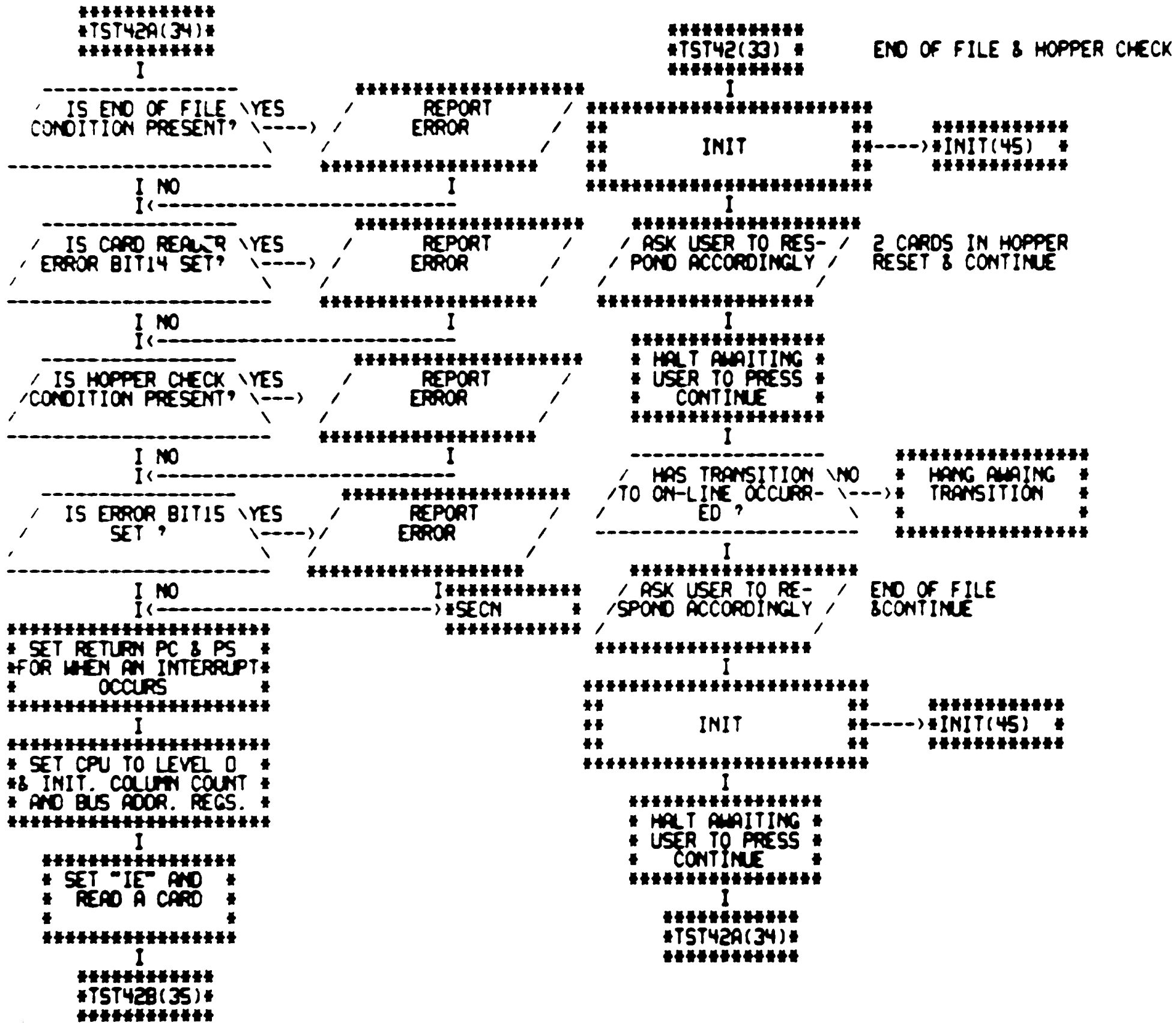
I
/ HAS CONTROLLER \ NO
/ READY SET YET ? \-----> * WAIT *
? / * FOR *
* IT *

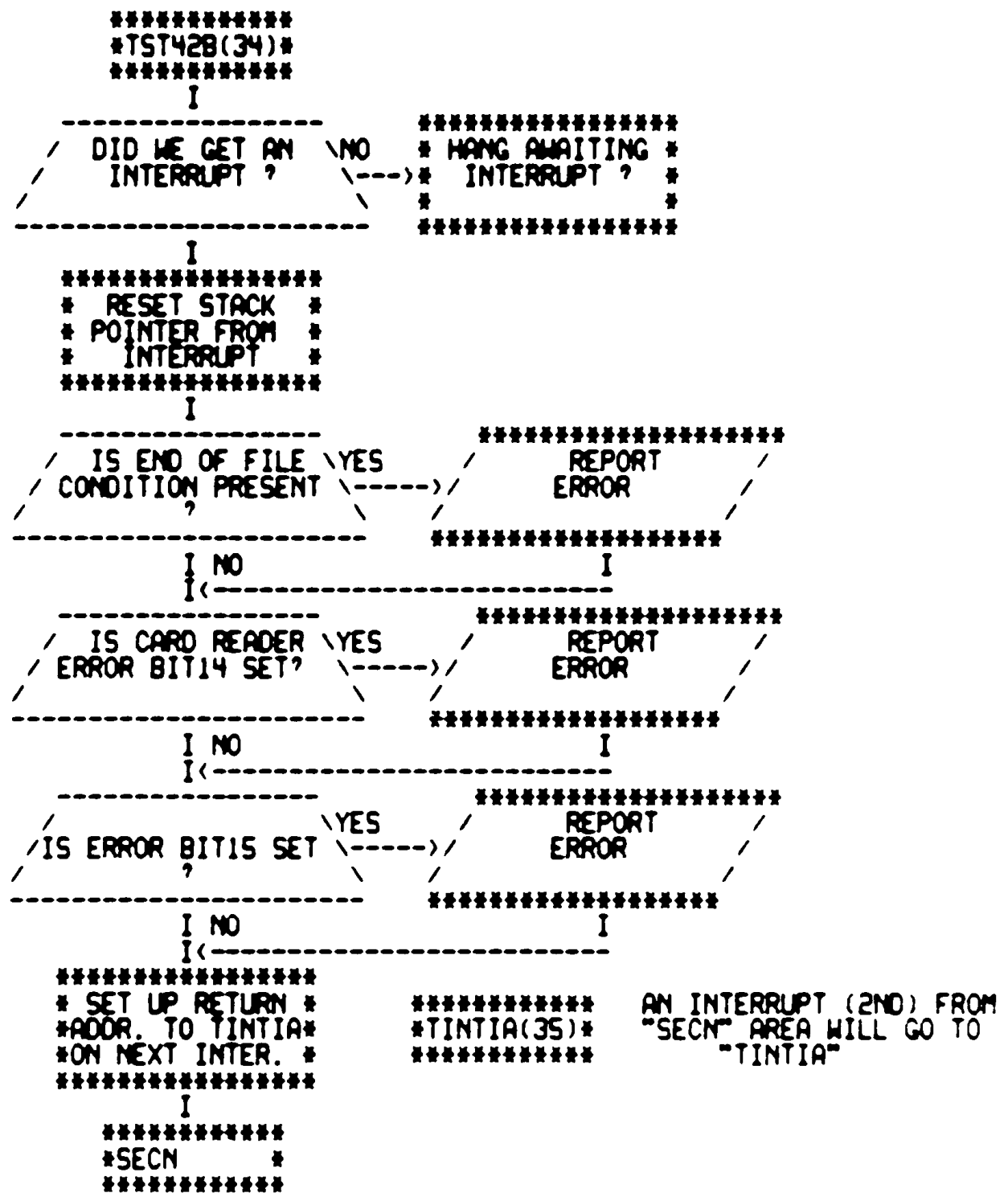
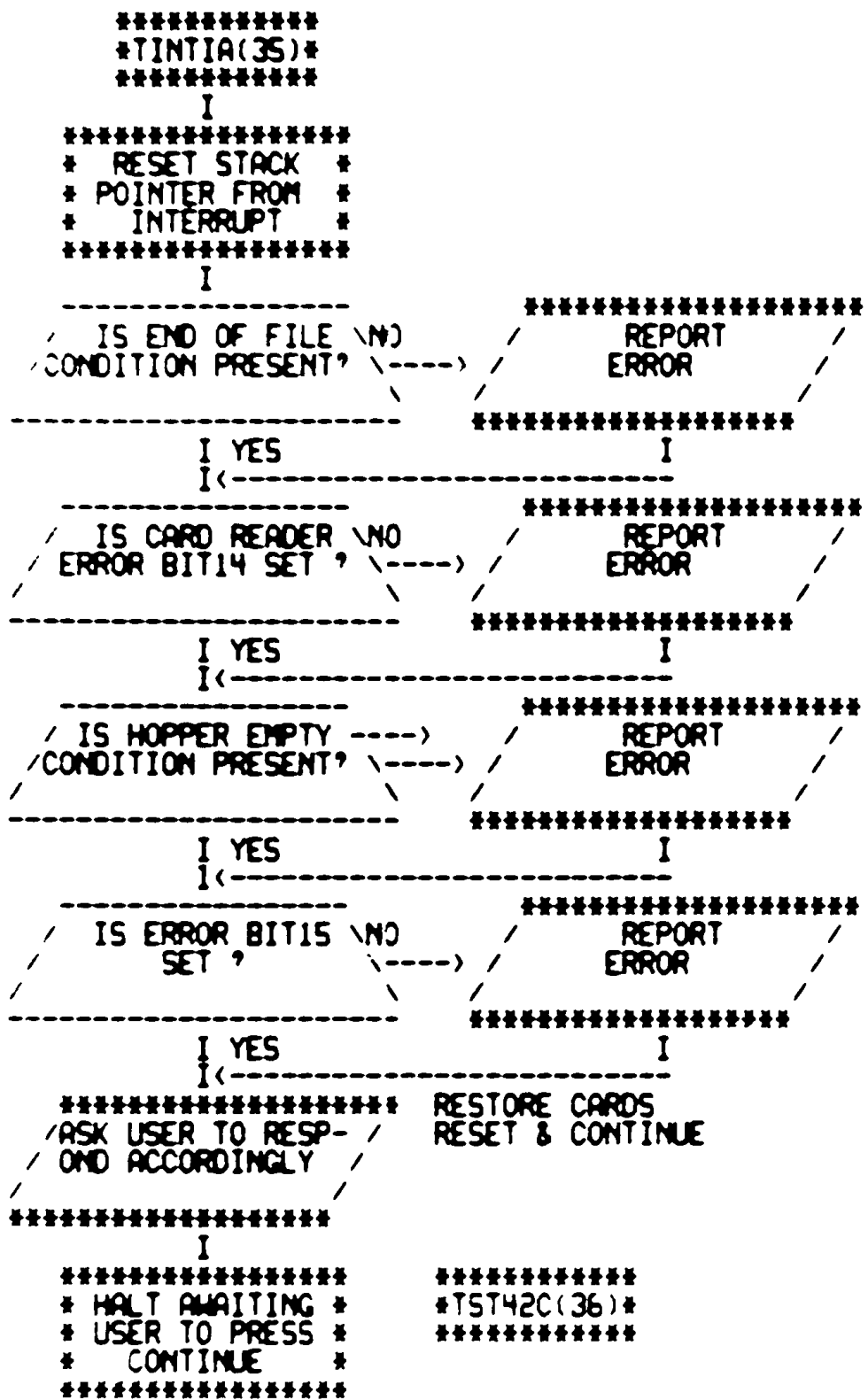
TST41C(33)

TST41A(32)

```
*****  
*TST41D(33)*  
*****  
I  
*****  
* READ A *  
* CARD *  
* *  
*****  
I  
-----  
/ DID AN INTERRUPT OCCUR? \ NO * HANG AWAITING *  
-----> INTERRUPT *  
-----  
I YES  
*****  
* RESET STACK *  
* POINTER FROM *  
* INTERRUPT *  
*****  
I  
-----  
/ IS CONTENTS OF STATUS REGISTER = 300 ? \ NO * HANG AWAITING *  
-----> REPORT ERROR *  
-----  
I YES  
I <-----  
*****  
*TST42(34) *  
*****
```

```
*****  
*TST41C(32)*  
*****  
I  
*****  
* SET RETURN PC & PS *  
* FOR WHEN AN INTERRUPT *  
* OCCURS *  
*****  
I  
*****  
* SET CPU TO LEVEL 0 & *  
* SET "IE" BIT *  
* *  
*****  
I  
*****  
/ ASK USER TO RE- REMOVE JAMMED CARD  
/ SPOND ACCORDINGLY / AND RESET  
*****  
I  
-----  
/ DID WE GET AN INTERRUPT ? \ NO * HANG AWAITING *  
-----> INTERRUPT *  
-----  
I YES  
*****  
* RESET STACK *  
* POINTER FROM *  
* INTERRUPT *  
*****  
I  
*****  
* SET RETURN PC & PS *  
* FOR WHEN AN INTERRUPT *  
* OCCURS *  
*****  
I  
*****  
* SET CPU TO LEVEL 0 & *  
* INIT. COLUMN COUNT *  
* & BUS ADDR. REGS. *  
*****  
I  
*****  
*TST41D(33)*  
*****
```





TST42C(35)

I

```
-----
/ HAS TRANSITION \ NO      * HANG AWAITING *
/ TO ON-LINE OCCUR- \-----> * TRANSITION *
/ RED , \
-----
I YES
-----
/ IS END OF FILE \ YES / REPORT
/ CONDITION PRESENT \-----> ERROR
/ , \
-----
I<-----
I
```

TST43(37)

TST43C(37)

I

/ ARE ANY EXTRA \ YES / *****
/ BITS SET IN "CDS" \----> / REPORT
/ THAT SHOULDN'T BE? \ / ERROR

I NO

I<

* SET UP RETURN PC & *
* PS FOR WHEN AN INTER-*
* RUPT OCCURS *

I

*SET CPU TO LEVEL 0 & *
* ENABLE INTERRUPTS *
* *

I

/ ASK USER TO RESP- / RESTORE CARDS
/ OND ACCORDINGLY / & RESET

I

/ HAS THE INTER- \ NO / *****
/ RUPT OCCURRED? \----> / WAIT FOR
/ / IT /

I YES

* HALT AWAITING *
* USER TO PRESS *
* CONTINUE *

I

* RESET STACK *
* POINTER FROM *
* INTERRUPT *

I

*ER12CD *

```
*-----*  
ERROR FUNCTION TESTING OF CARD READER MODELS M1000 OR  
M200 IS IDENTICAL TO THAT OF AN M1200 AS OUTLINED  
  
*****  
STARTING AT*ER1200(24)* WITH ONLY ONE EXCEPTION:  
*****  
  
*****  
*TST42(34) * IS NOT EXECUTED (START ADDRESS = 210)  
*****  
*-----*
```

```

*****
*SAME1(40) *
*****
I
-----
/ WAS IMAGE MODE \ YES
/ BEEN SELECTED \
ALSO ? \-----
-----
I NO
*****
*SET PACKING MODE IN- *
* DICATOR IN STATUS *
* REGISTER *
*****
I
I<-----*CKREAD *
*****
* INITIALIZE *
* COLUMN COUNT *
* TO ZERO *
*****
I
*****
* SET UP BUFFER ADDR. *
* TO "BUFBEQ" & COLUMN *
* COUNT TO 120(8) COLS.*
*****
I
*****
* START READING *
* A CARD *
*****
I
*****
* UPDATE THE *
* CARD COUNT *
*****
I
-----
/ IS CONTROLLER \ NO
/ READY \----->* HANG AWAITING *
* CONTROLLER *
* READY *
*****
I YES
*****
*SAME2(41) *
*****

```

```

*****
*CKSAME *
*****
I
*****
/ ASK USER TO LOAD /
/ PATTERN INTO SWR /
<11:0> /
*****
I
*****
* HALT AWAITING *
* USER TO PRESS *
* "CONTINUE" *
*****
I
*****
*STORE PATTERN SELECT-*
*ED AND MASK OFF BITS *
* 12 THRU 15 *
*****
I
*****
/ ASK USER TO SET /
/ DESIRED SWITCH /
/ REGISTER OPTIONS /
*****
I
*****
* HALT AWAITING *
* USER TO PRESS *
* CONTINUE *
*****
I
*****
*CKSTRT *----->I
*****
**
** INIT **----->*INIT(45) *
**
*****
I
*****
* CLEAR CARD COUNT *
* (TOTCRD) & ERROR *
* COUNT (TOTERR) *
*****
I<-----*CKLOOP *
*****
/ HAS PACK MODE \ NO
/ BEEN SELECTED \----->*CKREAD *
*****
I YES
*****
*SAME1(40) *
*****

```

START ADDRESS = 240

I05

*SAME2(40) *

I

DO WE HAVE AN ERROR CONDITION? \YES *****
-----> *CKERR(42) *

I NO

DOES DATA MATCH? \NO

*CKLOP3 *

I YES

PACKED MODE ? \NO *****
-----> *CKFAIL(43) *

I YES

* UPDATE THE *
* COLUMN *
* *

IMAGE MODE
PATH

CKLOP2(41)

* GO BACK TO * NO
* LOOK AT *
* NEXT COLUMN *

HAVE WE LOOKED AT ALL 80(10) COLUMNS ? \NO
-----> *CKFAIL(43) *

I YES

*CKLOOP *

DOES DATA MATCH? \NO *****
-----> *CKFAIL(43) *

I YES

* UPDATE *
* COLUMN *
* COUNT *

HAVE WE LOOKED AT ALL 80(10) COLUMNS ? \NO *****
-----> * GO BACK TO *
* LOOK AT NEXT *
* COLUMN *

I YES

*CKLOOP *

CKLOP2(41)

*CKERR(41) *

*****I05

I

IS OFF-LINE CON- NO
DITION PRESENT?

I YES

* RING *
* A *
* DING *

I

ARE WE GETTING
A TRANSITION TO
ON-LINE ?

I YES

*CKSTRT *

* HANG AWAITING * NO
* TRANSITION *
*

WAS THERE A
DATA ERROR ?

YES

I NO

REAL LIVE
DEADLY ERROR !

I

*CKLOOP *

YES

WERE WE
ATTEMPTING PACK
MODE ?

I NO

NO

WERE WE ATTEMPT-
ING IMAGE MODE ?

I YES

*CKLOP3 *

```

*****
*FAIL1(43) *
*****
I
-----
/ MALT ON ERROR ? \ YES *
\ -> *          HALT *
\ *
-----
I NO
*****
* UPDATE COLUMN *
* COUNT FOR NEXT *
* COLUMN LOOKUP *
*****
I
-----
/ HAVE WE LOOKED \ YES *****
AT ALL 80(10) \ -> *CKLOOP *
COLS. ? \ *
-----
I NO
-----
/ ARE WE DOING \ NO *****
PACKED MODE ? \ -> *CKLOP2(41)*
-----
I YES
*****
*CKLOP3 *
*****

```

```

*****
*CKFAIL(41)*
*****
I
*****
*UPDATE TOTAL *
*OF ERRORS FOUND*
* (TOTERR) *
*****
I
-----
/ INHIBIT ERROR \ YES *****
PRINTOUT ? \ -> *FAIL1(43) *
\ *
-----
I NO
*****
TYPE COLUMN
HEADINGS
*****
I
*****
*STEP UP COLUMN *
* COUNT FOR *
* ERROR REPORT *
*****
I
*****
TYPE COLUMN IN
WHICH ERROR WAS
DETECTED
*****
I
*****
* DROP COLUMN *
* COUNT BACK TO *
* ORIGINAL VALUE *
*****
I
-----
/ ARE WE DOING \ YES *****
PACKED MODE ? \ -> * TYPE INCORRECT BYTE *
\ * VALUE, CARD NO. & *
\ * TOTAL NO. OF ERRORS *
-----
I NO
*****
* TYPE INCORRECT WORD *
* VALUE, CARD NO. & *
* TOTAL NO. OF ERRORS *
*****
I<-----
*****
*FAIL1(43) *
*****

```


LOS

*TESTX * START ADDRESS = 220

I

/ ASK USER TO LOAD /
/ "SCOPE" ADDR. OF /
/ DESIRED TEST /

I

* HALT AWAITING *
* USER TO PRESS *
* CONTINUE *

I

* STORE ADDRESS & *
* CHANGE IT TO ADDR. OF *
* 1ST INSR. AFTER SCOPE *

I

/ ASK USER TO SET /
/ DESIRED SWITCH /
/ REGISTER OPTIONS /

I

* HANG AWAITING *
* USER TO PRESS *
* CONTINUE *

I

* STORE ADDR. LOADED BY *
* USER IN "SLPADR" TO *
* BE PICKED UP BY SCOPE *

I

* JUMP TO TEST *
* SELECTED ! *

CKOFFL(02)

I

/ IS OFF-LINE CON- \ NO
/ DITION PRESENT ? \ ---->

I YES

/ TELL USER HE IS \
/ OFF-LINE ! PRESS \
/ RESET & CONTINUE \

I

* HANG AWAITING *
* USER TO PRESS *
* CONTINUE *

I

CKOFFL(45)

*INIT(08) *

I

*RETURN TO WHERE ROU- * **
* TIME "CKOFFL" WAS * **
* LAST CALLED * **

CKOFFL

CKOFFL(45)

I

/ IS CONTROLLER \ NO
/ READY ? \ ---->

I YES

* ISSUE A *
* POWER CLEAR *
* *

I

/ IS STATUS REGIS- \ NO
/ TER NOW ALL \ ---->
/ CLEARED OUT ? \

I YES

*RETURN TO WHERE ROU- *
* TIME "INIT" WAS LAST *
* CALLED *

* HANG AWAITING *
* CONTROLLER *
* READY *

REPORT
ERROR

2011/CD20 CARD READER DIAGNOSTIC
FLOW CHART CROSS REFERENCE LIST

TST11	09	10	11	
TST2	02	03		
TST20	11			
TST20A	11	11		
TST21	11	12		
TST22	12	13		
TST23	13	14		
TST24	14	14		
TST25	14	14		
TST26	14			
TST3	03	03	04	
TST31	15	15	16	
TST32	15	15	16	
TST33	K	K	24	
TST33A	K	K		
TST34	K	K		
TST34A	K	K		
TST35	K	K	26	
TST35A	K	K		
TST35B	K	K		
TST35C	K	K	28	
TST36	L	L		
TST36A	R	R		
TST36B	R	R		
TST37	R	R		
TST37A	R	R		
TST37B	R	R		
TST4	5	5		
TST40	9	9		
TST40A	9	9		
TST40B	9	9		
TST40C	9	9		
TST40D	31	31		
TST41	31	32		
TST41A	32	32		
TST41B	32	32		
TST41C	32	32		
TST41D	32	32		
TST42	32	34	39	
TST42A	34	34		
TST42B	34	35		
TST42C	35	36		
TST43	36	37		
TST43A	37	37		
TST43B	37	37		
TST43C	37	38		
TST5	05	06		
TST5A	06	06		
TST5B	06	06		
TST6	06	06	06	07
TST6A	07	07		
TST6B	07			

TST7 07
 07 07 08

42	OPERATOR PROCEDURES
137	DEFINITIONS AND VECTOR ASSIGNMENTS
169	BASIC DEFINITIONS
284	TRAP CATCHER
294	STARTING ADDRESS(ES)
299	ACT11 HOOKS
332	COMMON TAGS
412	ERROR POINTER TABLE
865	LOGIC FUNCTION TESTS
935	T1 TEST FOR INIT. OF ALL REGISTERS
974	T2 TEST READ/WRITE OF STATUS REGISTER
1000	T3 TEST READ/WRITE OF COLUMN COUNT REGISTER
1024	T4 TEST READ/WRITE OF BUS ADDRESS REGISTER
1048	T5 TEST CONTROLLER READY TO CLEAR BIT0
1099	T6 TEST BIT2 TO BE CLEAR AFTER CARD READ
1149	T7 TEST INTERRUPT FROM CONTROLLER READY
1220	T10 TEST NO INTERRUPT ON CONTROLLER READY & CPU AT LEVEL 7
1270	T11 TEST FOR AN INTERRUPT ON LEVEL 7
1369	T12 TEST FOR AN INTERRUPT ON LEVEL 6
1468	T13 TEST FOR AN INTERRUPT ON LEVEL 5
1567	T14 TEST FOR AN INTERRUPT ON LEVEL 4
1661	T15 TEST FOR AN INTERRUPT ON LEVEL 3
1760	T16 TEST FOR AN INTERRUPT ON LEVEL 2
1859	T17 TEST FOR AN INTERRUPT ON LEVEL 1
1955	T20 TEST NO INTERRUPT WITH IE SET & REST CLEARED
2010	T21 SIMULTANEOUS INTERRUPTS AT MORE THAN 1 LEVEL
2080	T22 NON-EXISTANT MEMORY DETECTION
2169	T23 EXECUTE DATI,DATOB(LOW BYTE) LOAD ON COLUMN COUNT
2187	T24 EXECUTE DATI,DATOB(HIGH BYTE) LOAD ON COLUMN COUNT
2205	T25 EXECUTE DATI,DATIP ON COLUMN COUNT REGISTER
2223	T26 EXECUTE DATI,DATOB(LOW BYTE) LOAD ON BUS ADDRESS
2241	T27 EXECUTE DATI,DATOB(HIGH BYTE) LOAD ON BUS ADDRESS
2259	T30 EXECUTE DATI,DATIP ON BUS ADDRESS REGISTER
2277	T31 WORD COUNT OVERFLOW TO 2ND CARD
2375	T32 BUS ADDRESS ODD & TRANSFER IN NON-PACK MODE
2420	DATA RELIABILITY TEST
2878	END OF PASS ROUTINE
2996	ERROR +1 FUNCTION TESTS
3089	T33 TEST DATA LATE
3125	T34 TEST ERROR AND OFF LINE BITS
3364	T36 TEST INPUT HOPPER EMPTY
3492	T37 TEST OUTPUT STACKER FULL
3619	T40 TEST PICK CHECK ERROR
3775	T41 TEST STACK CHECK ERROR
3943	T42 TEST 'END OF FILE' AND HOPPER CHECK
4084	T43 TEST READ CHECK ERROR
4185	LOOP ON TEST ROUTINE
4287	IDENTICALLY PUNCHED CARDS TEST
4481	COMMON ROUTINES
4509	ERROR MESSAGE TIMEOUT ROUTINE
4558	ERROR HANDLER ROUTINE
4603	SCOPE HANDLER ROUTINE
4652	TYPE ROUTINE
4726	BINARY TO OCTAL (ASCII) AND TYPE

MAINDEC - 11 - DZCDB-A MACY11 27(663) 19-FEB-76 14:38
DZCDBA.P11 TABLE OF CONTENTS

SEQ 0069

4805	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
4874	TRAP DECODER
4890	TRAP TABLE
4906	POWER DOWN AND UP ROUTINES
4956	DATA TABLES

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

```
.NLIST MC,MD,CND
.LIST ME
.ENABL ABS,AMA
.MCALL .HEADER,STARS,.SCMTAG,.SCATCH,.SERROR,.SERRTYP,SETPRI
.MCALL .EQUAT,.SETUP,.$TYPOCT,$TYPE,$TRAP,$POWER,GETPRI
.MCALL NEWTST,.$SCOPE,.$SEOP,.$ACT11,.$STYPDEC,COMMENT,ENDCOMMENT
```

176000

SSWR=176000

```
.TITLE MAINDEC - 11 - DZCDB-A
;#COPYRIGHT (C) 1976
;#DIGITAL EQUIPMENT CORP.
;#MAYNARD, MASS. 01754
;*
;#PROGRAM BY B. BURGESS
;*
;#THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;#PACKAGE (MAINDEC-11-DZQAC-B1),AUG 29,1975.
;*
$TN=1
```

000001

```
;COPYRIGHT (C) 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 AVAILASLE 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.
```

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

.SBTTL OPERATOR PROCEDURES

: STARTING ADDRESSES ARE:

200=INSTRUCTION AND DATA TEST FOR THE CD11
 210=ERROR FUNCTION TEST OF CD11 (M-1000/M-200)
 220=SINGLE TEST LOOP
 240=READ SINGLE DATA PATTERN TEST
 250=ERROR FUNCTION TEST FOR CD11 (M-1200)

: SWITCH REGISTER SETTINGS FOR THE INSTRUCTION AND DATA TEST ARE:

SW02=1 RUN IN DATA IMAGE MODE ONLY
 SW03=1 RUN IN DATA PACKING MODE ONLY (IGNORED IF SW02=1)
 SW04=1 FOR THE BINARY TEST DECK
 SW05=1 TO HALT AT THE END OF A STANDARD 80 CARD TEST DECK. (HITTING CONTINUE WILL START TESTING OF THE NEXT DECK IN ACCORDANCE WITH CURRENT SWR SETTINGS).
 =0 TO CONTINUE FROM ONE DECK TO THE NEXT. AFTER THE LAST DECK IN THE HOPPER IS RUN, THE PROGRAM WAITS FOR THE CARD READER TO COME BACK ON-LINE AND RUNS THRU A SERIES OF CHECKS OF OFF-LINE AND COMING ON-LINE OPERATIONS OF THE READER. WHEN THE READER IS BACK ON-LINE AND THE CHECKS ARE COMPLETE, THE DATA TEST IS RESUMED.
 SW06=1 TO RUN THE COMBINED INSTRUCTION AND DATA TEST WHEN CONTINUING FROM ONE DECK TO THE NEXT
 =0 TO RUN ONLY THE DATA TEST ON EVERY DECK AFTER THE FIRST
 SW07=1 TO RUN ONLY THE INSTRUCTION TEST CONTINUALLY. SETTING SW06 AND SW07 AT THE END OF A DECK WILL CAUSE THE INSTRUCTION TEST TO BE RUN CONTINUOUSLY FROM THEN ON (NOTE THAT IF SW7 IS SET, THE PROGRAM MAY HANG WHEN THE CARD READER RUNS OUT OF CARDS)
 SW11=1 TO INHIBIT SUBPROGRAM ITERATION (NOTE THAT IF PROGRAM FLOW IS ALLOWED TO ENTER THE DATA SUBTEST WHEN SW11 IS SET, DATA ERRORS WILL OCCUR SINCE THE CARD COUNT WILL BE INCORRECT.)
 SW12=1 TO INHIBIT TRACE TRAPPING
 SW13=1 TO INHIBIT PRINTOUT
 SW14=1 FOR SCOPE LOOP & LOOP ON TEST
 SW15=1 TO HALT ON ERROR

: OPERATING PROCEDURE FOR THE INSTRUCTION AND DATA TEST:

1. LOAD TEST DECK IN CARD READER AND PRESS "START" ON THE CARD READER. IF THE DECK BEING USED IS NOT A STANDARD TEST DECK, ONLY THE INSTRUCTION PORTION OF THE TEST CAN BE RUN. (SW7 MUST BE SET TO ONE TO INDICATE THIS).
2. LOAD SA 200, THEN SET THE SWITCH REGISTER SWITCHES TO THE DESIRED COMBINATION
3. PRESS "START" ON THE CONSOLE
4. NOTE THAT RUNNING THE COMPLETE INSTRUCTION TEST REQUIRES THAT THE INPUT HOPPER MUST RUN OUT OF CARDS

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

AT THE END OF A TEST DECK AT LEAST ONCE. WHEN THIS OCCURS, THE PROCESSOR SHOULD CONTINUE TO RUN. LOADING A DECK INTO THE INPUT HOPPER AND PRESSING "RESET" ON THE CARD READER SHOULD CAUSE THE BELL TO RING AND THE CARD READER TO RESUME READING CARDS. IF THIS DOES NOT OCCUR, IT IS A FAULT AND SHOULD BE FIXED.

; SPECIAL SWITCH REGISTER SETTINGS FOR THE ERROR FUNCTION TEST:
SW14=1 TO LOOP THRU THE CURRENT SUBTEST
SW15=1 TO HALT ON ERROR

; OPERATING PROCEDURE FOR THE ERROR FUNCTION TEST:
1. LOAD A FEW SPARE CARDS INTO THE INPUT HOPPER.
2. PRESS "RESET" ON THE CARD READER.
3. LOAD THE SA, THEN SET THE DESIRED SWITCH OPTIONS.
4. PRESS "START" ON THE CONSOLE.
5. FOLLOW THE INSTRUCTIONS AS THEY ARE PRINTED OUT.

; SINGLE TEST LOOP (SA 220) HALTS TWICE!
1ST HALT - LOAD STARTING ADDRESS OF DESIRED TEST (TEST1 TO TEST 22)
2ND HALT - SET SWR OPTIONS (BIT 11 MUST = 0)
THIS TEST USES TRACE TRAPPING WHERE APPLICABLE IF SW12 IS NOT SET

; DESCRIPTION OF SINGLE DATA PATTERN TEST
THIS TEST IS DESIGNED TO AID IN THE LOCATION OF DIFFICULT DATA ERROR PROBLEMS AND PERHAPS HELP IN SOME CARD READER ADJUSTMENTS. IT CONTINUOUSLY READS CARDS WHICH HAVE ALL COLUMNS PUNCHED OR MARKED IDENTICALLY, CHECKING THE DATA AGAINST A PATTERN SET UP ON THE SWITCHES INITIALLY. ANY ERRORS ARE PRINTED OUT, ALONG WITH A COUNT OF THE TOTAL NUMBER OF CARDS READ AND THE TOTAL NUMBER OF DATA ERRORS WHICH HAVE OCCURRED SINCE THE TEST WAS STARTED.

; OPERATING PROCEDURE FOR SINGLE DATA PATTERN TEST:
1. LOAD TEST DECK OF IDENTICAL CARDS IN THE INPUT HOPPER, AND PUT THE CARD READER ON-LINE (I.E. - PRESS "RESET" ON CARD READER).
2. LOAD SA 240, THEN PRESS "START" ON THE CONSOLE.
3. AT THE INITIAL HALT SET THE CORRECT CARD-IMAGE DATA PATTERN IN SW11-SW00, THEN PRESS CONTINUE.
4. WHEN THE READER RUNS OUT OF CARDS IT WILL RING THE BELL. RELOADING THE DECK AND PRESSING "RESET" ON THE CARD READER WILL CONTINUE THE TEST.

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

```

*****
:STATUS AND CONTROL REGISTER (COST) BIT DESIGNATIONS
:   BIT 0   READ
:   BIT 1   DATA PACKING
:   BIT 2   HOPPER EMPTY
:   BIT 3   READER TRANSITION TO ON LINE
:   BIT 4   ADDRESS BIT 16
:   BIT 5   ADDRESS BIT 17
:   BIT 6   INTERRUPT ENABLE
:   BIT 7   CONTROLLER READY
:   BIT 8   POWER CLEAR
:   BIT 9   NON-EXISTENT MEMORY
:   BIT 10  DATA LATE
:   BIT 11  DATA ERROR
:   BIT 12  OFF-LINE
:   BIT 13  END OF FILE (M1200 ONLY)
:   BIT 14  CARD READER ERROR
:   BIT 15  ERROR
*****

```

.SBTTL BASIC DEFINITIONS

```

: *INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100 STACK= 1100
: .EQUIV EMT,ERROR ;: BASIC DEFINITION OF ERROR CALL
: .EQUIV IOT,SCOPE ;: BASIC DEFINITION OF SCOPE CALL
177776 PS= 177776 ;: PROCESSOR STATUS WORD
: .EQUIV PS,PSW
177774 STKLMT= 177774 ;: STACK LIMIT REGISTER
177772 PIRO= 177772 ;: PROGRAM INTERRUPT REQUEST REGISTER
177570 DSWR= 177570 ;: HARDWARE SWITCH REGISTER
177570 ODISP= 177570 ;: HARDWARE DISPLAY REGISTER

```

```

: *GENERAL PURPOSE REGISTER DEFINITIONS
183 000000 R0= %0 ;: GENERAL REGISTER
184 000001 R1= %1 ;: GENERAL REGISTER
185 000002 R2= %2 ;: GENERAL REGISTER
186 000003 R3= %3 ;: GENERAL REGISTER
187 000004 R4= %4 ;: GENERAL REGISTER
188 000005 R5= %5 ;: GENERAL REGISTER
189 000006 R6= %6 ;: GENERAL REGISTER
190 000007 R7= %7 ;: GENERAL REGISTER
191 .EQUIV R6,SP ;: STACK POINTER

```

```

192      .EQUIV R7,PC          ;;PROGRAM COUNTER
193
194      ;*PRIORITY LEVEL DEFINITIONS
195      000000      PR0=      0          ;;PRIORITY LEVEL 0
196      000040      PR1=     40          ;;PRIORITY LEVEL 1
197      000100      PR2=    100          ;;PRIORITY LEVEL 2
198      000140      PR3=    140          ;;PRIORITY LEVEL 3
199      000200      PR4=    200          ;;PRIORITY LEVEL 4
200      000240      PR5=    240          ;;PRIORITY LEVEL 5
201      000300      PR6=    300          ;;PRIORITY LEVEL 6
202      000340      PR7=    340          ;;PRIORITY LEVEL 7
203
204      ;*"SWITCH REGISTER" SWITCH DEFINITIONS
205      100000      SW15=   100000
206      040000      SW14=    40000
207      020000      SW13=    20000
208      010000      SW12=    10000
209      004000      SW11=    4000
210      002000      SW10=    2000
211      001000      SW09=    1000
212      000400      SW08=    400
213      000200      SW07=    200
214      000100      SW06=    100
215      000040      SW05=    40
216      000020      SW04=    20
217      000010      SW03=    10
218      000004      SW02=    4
219      000002      SW01=    2
220      000001      SW00=    1
221      .EQUIV SW09,SW9
222      .EQUIV SW08,SW8
223      .EQUIV SW07,SW7
224      .EQUIV SW06,SW6
225      .EQUIV SW05,SW5
226      .EQUIV SW04,SW4
227      .EQUIV SW03,SW3
228      .EQUIV SW02,SW2
229      .EQUIV SW01,SW1
230      .EQUIV SW00,SW0
231
232      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
233      100000      BIT15=  100000
234      040000      BIT14=   40000
235      020000      BIT13=   20000
236      010000      BIT12=   10000
237      004000      BIT11=   4000
238      002000      BIT10=   2000
239      001000      BIT09=   1000
240      000400      BIT08=   400
241      000200      BIT07=   200
242      000100      BIT06=   100
243      000040      BIT05=   40
244      000020      BIT04=   20
245      000010      BIT03=   10

```

```

246      000004      BIT02= 4
247      000002      BIT01= 2
248      000001      BIT00= 1
249      .EQUIV      BIT09, BIT9
250      .EQUIV      BIT08, BIT8
251      .EQUIV      BIT07, BIT7
252      .EQUIV      BIT06, BIT6
253      .EQUIV      BIT05, BIT5
254      .EQUIV      BIT04, BIT4
255      .EQUIV      BIT03, BIT3
256      .EQUIV      BIT02, BIT2
257      .EQUIV      BIT01, BIT1
258      .EQUIV      BIT00, BIT0
259
260      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
261      000004      ERRVEC= 4      ;; TIME OUT AND OTHER ERRORS
262      000010      RESVEC= 10     ;; RESERVED AND ILLEGAL INSTRUCTIONS
263      000014      TBITVEC= 14    ;; "T" BIT
264      000014      TRIVEC= 14     ;; TRACE TRAP
265      000014      BPTVEC= 14     ;; BREAKPOINT TRAP (BPT)
266      000020      IOTVEC= 20     ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
267      000024      PWRVEC= 24     ;; POWER FAIL
268      000030      EMTVEC= 30     ;; EMULATOR TRAP (EMT) **ERROR**
269      000034      TRAPVEC= 34    ;; "TRAP" TRAP
270      000060      TKVEC= 60      ;; TTY KEYBOARD VECTOR
271      000064      TPVEC= 64      ;; TTY PRINTER VECTOR
272      000240      PIRQVEC= 240   ;; PROGRAM INTERRUPT REQUEST VECTOR
273
274      ;SPECIAL EQUATES
275      000000      DUMMY= 0
276      000002      ADINT= %2      ;; CONTAINS ADDRESS OF INTERRUPT VECTOR
277      000003      CDS= %3       ;; CONTAINS ADDRESS OF CARD READER STATUS REGISTER
278      000004      CDC= %4       ;; CONTAINS ADDRESS OF CARD READER COLUMN COUNT
279      000005      CDA= %5       ;; CONTAINS ADDRESS OF CARD READER BUS ADDRESS REG.
280      000005      TTY= %5
281      000360      TRACE= 360
282
283      .SBTTL TRAP CATCHER
284
285      .=0
286      000000
287      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
288      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
289      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
290      .=174
291      000174      000000      DISPREG: .WORD 0      ;; SOFTWARE DISPLAY REGISTER
292      000176      000000      SWREG: .WORD 0        ;; SOFTWARE SWITCH REGISTER
293
294      .SBTTL STARTING ADDRESS(ES)
295      000200      000137      002170      JMP @#BEGIN      ;; JUMP TO STARTING ADDRESS OF PROGRAM
296
297      ;*****
298
299      .SBTTL ACT11 HOOKS

```



```

300          ;HOOKS REQUIRED BY ACT11
301          $SVPC=.          ;SAVE PC
302          .=46
303 000046   $ENDAD          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
304          .=52
305 000052   .WORD 20000     ;;2)SET LOC.52 TO 20000
306          .=$SVPC        ;; RESTORE PC
307
308          .=14
309 000014   .WORD TRTRAP
310 000016   .WORD 340
311
312          ;ADDITIONAL STARTING ADDRESSES
313
314          .=210
315 000210   JMP ERCD11      ;FOR CD11 (M1000/M200) ERROR FUNCTION TESTS
316          .=220
317 000220   JMP TESTX       ;FOR LOOP WHICH WILL CONTINUTALLY RUN
318          .=240           ;ANY SINGLE SUBTEST
319          .=240
320          JMP CKSAME       ;TO READ A SINGLE DATA PATTERN
321          .=250           ;CONTINUOUSLY
322 000240   JMP ER1200      ;FOR CD11 (M1200) ERROR FUNCTION TEST
323
324
325 000250   .=250
326          JMP ER1200
327
328
329
  
```

```

330 ;*****
331
332 .SBTTL COMMON TAGS
333
334 ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
335 ;*USED IN THE PROGRAM.
336
337 001100 . =1100
338 001100 $CMTAG: .WORD 0 ; START OF COMMON TAGS
339 001100 $PASS: .WORD 0 ; CONTAINS PASS COUNT
340 001102 000 $STNM: .BYTE 0 ; CONTAINS THE TEST NUMBER
341 001103 000 $ERFLG: .BYTE 0 ; CONTAINS ERROR FLAG
342 001104 000000 $ICNT: .WORD 0 ; CONTAINS SUBTEST ITERATION COUNT
343 001106 000000 $LPADR: .WORD 0 ; CONTAINS SCOPE LOOP ADDRESS
344 001110 000000 $LPERR: .WORD 0 ; CONTAINS SCOPE RETURN FOR ERRORS
345 001112 000000 $ERTTL: .WORD 0 ; CONTAINS TOTAL ERRORS DETECTED
346 001114 000 $ITEMB: .BYTE 0 ; CONTAINS ITEM CONTROL BYTE
347 001115 001 $ERMAX: .BYTE 1 ; CONTAINS MAX. ERRORS PER TEST
348 001116 000000 $ERRPC: .WORD 0 ; CONTAINS PC OF LAST ERROR INSTRUCTION
349 001120 000000 $GDADR: .WORD 0 ; CONTAINS ADDRESS OF 'GOOD' DATA
350 001122 000000 $BDADR: .WORD 0 ; CONTAINS ADDRESS OF 'BAD' DATA
351 001124 000000 $GDDAT: .WORD 0 ; CONTAINS 'GOOD' DATA
352 001126 000000 $BDDAT: .WORD 0 ; CONTAINS 'BAD' DATA
353 001130 000000 .WORD 0 ; RESERVED--NOT TO BE USED
354 001132 000000 .WORD 0
355 001134 000000 .WORD 0
356 001136 177570 $SWR: .WORD DSWR ; ADDRESS OF SWITCH REGISTER
357 001140 177570 $DISPLAY: .WORD DDISP ; ADDRESS OF DISPLAY REGISTER
358 001142 177560 $TKS: 177560 ; TTY KBD STATUS
359 001144 177562 $TKB: 177562 ; TTY KBD BUFFER
360 001146 177564 $TPS: 177564 ; TTY PRINTER STATUS REG. ADDRESS
361 001150 177566 $TPB: 177566 ; TTY PRINTER BUFFER REG. ADDRESS
362 001152 000 $NULL: .BYTE 0 ; CONTAINS NULL CHARACTER FOR FILLS
363 001153 002 $FILLS: .BYTE 2 ; CONTAINS # OF FILLER CHARACTERS REQUIRED
364 001154 012 $FILLC: .BYTE 12 ; INSERT FILL CHARS. AFTER A "LINE FEED"
365 001155 000 $TPFLG: .BYTE 0 ; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
366 001156 000000 $REGAD: .WORD 0 ; CONTAINS THE ADDRESS FROM
367 ; WHICH ($REGO) WAS OBTAINED
368 001160 000000 $REG0: .WORD 0 ; CONTAINS (($REGAD)+0)
369 001162 000000 $REG1: .WORD 0 ; CONTAINS (($REGAD)+2)
370 001164 000000 $REG2: .WORD 0 ; CONTAINS (($REGAD)+4)
371 001166 000000 $REG3: .WORD 0 ; CONTAINS (($REGAD)+6)
372 001170 000000 $REG4: .WORD 0 ; CONTAINS (($REGAD)+10)
373 001172 000000 $REG5: .WORD 0 ; CONTAINS (($REGAD)+12)
374 001174 000000 $REG6: .WORD 0 ; CONTAINS (($REGAD)+14)
375 001176 000000 $REG7: .WORD 0 ; CONTAINS (($REGAD)+16)
376 001200 000000 $TMP0: .WORD 0 ; USER DEFINED
377 001202 000000 $TMP1: .WORD 0 ; USER DEFINED
378 001204 000000 $TMP2: .WORD 0 ; USER DEFINED
379 001206 000000 $TMP3: .WORD 0 ; USER DEFINED
380 001210 000000 $TMP4: .WORD 0 ; USER DEFINED
381 001212 000000 $TMP5: .WORD 0 ; USER DEFINED
382 001214 000000 $TMP6: .WORD 0 ; USER DEFINED
383 001216 000000 $TMP7: .WORD 0 ; USER DEFINED

```

384 001220 000000
385 001222 177607 000377
386 001226 077
387 001227 015
388 001230 000012

\$TIMES: 0
\$BELL: .ASCIZ <207><377><377>
\$QUES: .ASCII /?/
\$CRLF: .ASCII <15>
\$LF: .ASCIZ <12>

::MAX. NUMBER OF ITERATIONS
::CODE FOR BELL
::QUESTION MARK
::CARRIAGE RETURN
::LINE FEED

```

389 ;*****
390 ;LOAD POINTERS AND GENERAL STORAGE
391 001232 177160 CDST: 177160 ;ADDRESS OF CARD READER STATUS REGISTER #1
392 ;THIS REGISTER'S INFORMATION IS VALID
393 ;DURING DATA TRANSFERS
394 001234 177162 COCC: 177162 ;ADDRESS OF CARD READER COLUMN COUNT
395 001236 177164 CDBA: 177164 ;ADDRESS OF CARD READER BUS ADDRESS
396 001240 177166 CDOB: 177166 ;ADDRESS OF CARD READER STATUS REGISTER #2
397 ;THIS REGISTER'S INFORMATION IS VALID
398 ;DURING NON-DATA TRANSFER PERIODS
399 001242 000002 TRTRAP: RTI ;RETURN FROM TRACE LOOP
400 001244 000230 INTVC: 230 ;ADDRESS OF CARD READER INTERRUPT VECTOR
401 001246 000232 ;
402 001250 000000 COUNT: 0 ;USED FOR TIMING, ETC.
403 001252 000000 INTFLG: 0 ;CONTAINS LEVEL THAT INTERRUPT IS FOUND AT
404 001254 000000 TRFLG: 0 ;TOGGLED TO SWITCH BETWEEN TRACE TRAPPING AND NORMAL FLO
405 001256 000000 PROC: 0 ;STORES PROCESSOR STATUS WHEN TRACE TRAP MUST BE CLEARED
406 ;IN A SUBTEST
407 001260 000000 ERFLG: 0 ;SET TO ZERO TO OUTPUT DATA ERROR HEADING
408 001262 000000 CKRF: 0 ;FLAG FOR CHECKERBOARD DECK
409 001264 000000 COUNTG: 0 ;USED AS COUNTER IN TESTG
410 001266 000000 CD1000: 0 ;M-1200 OR M-1000/M-200 CARD READER DETECTOR
411
412 .SBTTL ERROR POINTER TABLE
413
414 ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
415 ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
416 ;*LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
417 ;*NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
418 ;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
419
420 ;* EM ;:POINTS TO THE ERROR MESSAGE
421 ;* DH ;:POINTS TO THE DATA HEADER
422 ;* DT ;:POINTS TO THE DATA
423 ;* DF ;:POINTS TO THE DATA FORMAT
424
425
426 001270 SERRTB:
427 ;ITEM 1
428
429 001270 034616 EM1 ;STATUS REGISTER (CDS) BIT07 NOT SET BY
430 ;INITIALIZATION PULSE
431 001272 042600 DH1 ;(PC) (SP) (CDS) (CDD) (PS)
432 001274 043562 DT1 ;SERRPC, SREG6, STMP0, STMP1, SREG7
433 001276 000000 0 ;OCTAL VALUES
434
435 ;ITEM 2
436
437 001300 034713 EM2 ;COLUMN COUNT REGISTER (CDC) NOT CLEARED BY
438 ;INITIALIZATION PULSE
439 001302 042646 DH2 ;(PC) (SP) (CDS) (CDD) (CDC) (PS)
440 001304 043576 DT2 ;SERRPC, SREG6, STMP0, STMP1, STMP2, SREG7
441 001306 000000 0 ;OCTAL VALUES

```

443					
444					
445					
446	001310	035014	EM3		: BUS ADDRESS REGISTER (CDA) NOT CLEARED BY
447					: INITIALIZATION PULSE
448	001312	042724	DH3		: (PC) (SP) (CDS) (CDD) (CDA) (PS)
449	001314	043614	DT3		: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP3, \$REG7
450	001316	000000	0		: OCTAL VALUES
451					
452					
453					
454					
455	001320	035114	EM4		: STATUS REGISTER CONTENTS INCORRECT
456	001322	042600	DH1		: (PC) (SP) (CDS) (CDD) (PS)
457	001324	043562	DT1		: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$REG7
458	001326	000000	0		: OCTAL VALUES
459					
460					
461	001330	035157	EM5		: COLUMN COUNT REGISTER (CDC) NOT ABLE TO
462					: BE LOADED WITH ALL ONE'S
463	001332	042646	DH2		: (PC) (SP) (CDS) (CDD) (CDC) (PS)
464	001334	043576	DT2		: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$REG7
465	001336	000000	0		: OCTAL VALUES
466					
467					
468					
469	001340	035260	EM6		: COLUMN COUNT REGISTER (CDC) NOT CLEARED
470					: BY POWER CLEAR
471	001342	042646	DH2		: (PC) (SP) (CDS) (CDD) (CDC) (PS)
472	001344	043576	DT2		: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$REG7
473	001346	000000	0		: OCTAL VALUES
474					
475					
476					
477	001350	035350	EM7		: BUS ADDRESS REGISTER (CDA) NOT ABLE TO BE
478					: LOADED WITH ALL ONE'S
479	001352	042724	DH3		: (PC) (SP) (CDS) (CDD) (CDA) (PS)
480	001354	043614	DT3		: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP3, \$REG7
481	001356	000000	0		: OCTAL VALUES
482					
483					
484					
485	001360	035450	EM10		: BUS ADDRESS REGISTER (CDA) NOT CLEARED
486					: BY POWER CLEAR
487	001362	042724	DH3		: (PC) (SP) (CDS) (CDD) (CDA) (PS)
488	001364	043614	DT3		: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP3, \$REG7
489	001366	000000	0		: OCTAL VALUES
490					
491					
492					
493	001370	035537	EM11		: CONTROLLER READY DIDN'T CLEAR ON
494					: READING A CARD
495	001372	042600	DH1		: (PC) (SP) (CDS) (CDD) (PS)
496	001374	043562	DT1		: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$REG7

497	001376	000000	0	:OCTAL VALUES
498				
499				
500				:ITEM 12
501	001400	035621	EM12	:CONTROLLER READY DIDN'T CLEAR BIT00
502				:OF STATUS REGISTER (CDS)
503	001402	042600	DH1	: (PC) (SP) (CDS) (CDD) (PS)
504	001404	043562	DT1	:SERRPC, SREG6, STMP0, STMP1, SREG7
505	001406	000000	0	:OCTAL VALUES
506				
507				:ITEM 13
508				
509	001410	035720	EM13	:CONTROLLER DIDN'T SET WITHIN ONE SECOND
510	001412	042600	DH1	: (PC) (SP) (CDS) (CDD) (PS)
511	001414	043562	DT1	:SERRPC, SREG6, STMP0, STMP1, SREG7
512	001416	000000	0	:OCTAL VALUES
513				
514				:ITEM 14
515				
516	001420	035767	EM14	:ERROR (BIT15) SET IN STATUS REGISTER (CDS)
517	001422	042600	DH1	: (PC) (SP) (CDS) (CDD) (PS)
518	001424	043562	DT1	:SERRPC, SREG6, STMP0, STMP1, SREG7
519	001426	000000	0	:OCTAL VALUES
520				
521				:ITEM 15
522				
523	001430	036042	EM15	:BIT/S SET IN STATUS REGISTER (CDS) THAT
524				:SHOULDN'T BE
525	001432	042600	DH1	: (PC) (SP) (CDS) (CDD) (PS)
526	001434	043562	DT1	:SERRPC, SREG6, STMP0, STMP1, SREG7
527	001436	000000	0	:OCTAL VALUES
528				
529				:ITEM 16
530				
531	001440	036137	EM16	: 'BUSY' SET IN STATUS REGISTER (CDS)
532				:SHOULDN'T BE
533	001442	042600	DH1	: (PC) (SP) (CDS) (CDD) (PS)
534	001444	043562	DT1	:SERRPC, SREG6, STMP0, STMP1, SREG7
535	001446	000000	0	:OCTAL VALUES
536				
537				:ITEM 17
538				
539	001450	036223	EM17	:RESTORING CPU STATUS AFTER READING A CARD
540				:CLEARED CONTROLLER READY IN STATUS REGISTER
541	001452	042600	DH1	: (PC) (SP) (CDS) (CDD) (PS)
542	001454	043562	DT1	:SERRPC, SREG6, STMP0, STMP1, SREG7
543	001456	000000	0	:OCTAL VALUES
544				
545				:ITEM 20
546				
547	001460	036352	EM20	:NO INTERRUPT OCCURRED
548	001462	043002	DH4	: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
549	001464	043632	DT4	:SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
550	001466	000000	0	:OCTAL VALUES

```

551
552
553 ;ITEM 21
554 001470 036400 EM21 ;AN INTERRUPT OCCURRED - SHOULDN'T HAVE
555 001472 043002 DM4 ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
556 001474 043632 DT4 ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
557 001476 000000 0 ;OCTAL VALUES
558
559 ;ITEM 22
560
561 001500 036450 EM22 ; INTERRUPT ALREADY OCCURRED AT A HIGHER LEVEL
562 001502 043002 DM4 ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
563 001504 043632 DT4 ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
564 001506 000000 0 ;OCTAL VALUES
565
566 ;ITEM 23
567
568 001510 036525 EM23 ; CONTROLLER READY NOT SET
569 001512 043002 DM4 ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
570 001514 043632 DT4 ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
571 001516 000000 0 ;OCTAL VALUES
572
573 ;ITEM 24
574
575 001520 036556 EM24 ; INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
576 001522 043002 DM4 ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
577 001524 043632 DT4 ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
578 001526 000000 0 ;OCTAL VALUES
579
580 ;ITEM 25
581
582 001530 036632 EM25 ; AN INTERRUPT OCCURRED AT TWO DIFFERENT LEVELS
583 001532 043002 DM4 ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
584 001534 043632 DT4 ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
585 001536 000000 0 ;OCTAL VALUES
586
587 ;ITEM 26
588
589 001540 036710 EM26 ; ERROR (BIT15) NOT SET IN STATUS REGISTER
590 ; (CDS) - SHOULD BE
591 001542 043002 DM4 ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
592 001544 043632 DT4 ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
593 001546 000000 0 ;OCTAL VALUES
594
595 ;ITEM 27
596
597 001550 037005 EM27 ; NON-EXISTANT MEMORY (BIT09) NOT SET IN
598 ; STATUS REGISTER (CDS) - SHOULD BE
599 001552 043002 DM4 ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
600 001554 043632 DT4 ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
601 001556 000000 0 ;OCTAL VALUES
602
603 ;ITEM 30
604

```


605	001560	037122	EM30	: EXTENDED MEMORY (BIT05) NOT SET IN STATUS REGISTER (CDS)
606				: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
607	001562	043002	DH4	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
608	001564	043632	DT4	: OCTAL VALUES
609	001566	000000	0	
610				
611				: ITEM 31
612				
613	001570	037214	EM31	: EXTENDED MEMORY (BIT04) NOT SET IN STATUS REGISTER (CDS)
614				: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
615	001572	043002	DH4	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
616	001574	043632	DT4	: OCTAL VALUES
617	001576	000000	0	
618				
619				: ITEM 32
620				
621	001600	036042	EM15	
622	001602	043002	DH4	: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
623	001604	043632	DT4	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
624	001606	000000	0	: OCTAL VALUES
625				
626				: ITEM 33
627				
628	001610	037306	EM32	: CONTENTS OF BUS ADDRESS REGISTER (CDA)
629				: INCORRECT
630	001612	043070	DH5	: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (CDA) (PS)
631				: WAS SHB
632	001614	043652	DT5	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, STMP4, SREG7
633	001616	000000	0	: OCTAL VALUES
634				
635				: ITEM 34
636				
637	001620	037370	EM33	: CONTENTS OF COLUMN COUNT REGISTER (CDC)
638				: INCORRECT
639	001622	043254	DH6	: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (CDC) (PS)
640				: WAS SHB
641	001624	043652	DT5	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, STMP4, SREG7
642	001626	000000	0	: OCTAL VALUES
643				
644				: ITEM 35
645				
646	001630	037453	EM34	: READER OFF-LINE BUT CARD READER ERROR
647				: (BIT14) NOT SET IN STATUS REGISTER (CDS)
648	001632	042600	DH1	: (PC) (SP) (CDS) (CDD) (PS)
649	001634	043562	DT1	: SERRPC, SREG6, STMP0, STMP1, SREG7
650	001636	000000	0	: OCTAL VALUES
651				
652				: ITEM 36
653				
654	001640	037573	EM35	: NO TRANSITION TO ON-LINE (BIT03) OCCURRED
655				: IN STATUS REGISTER (CDS)
656	001642	042600	DH1	: (PC) (SP) (CDS) (CDD) (PS)
657	001644	043562	DT1	: SERRPC, SREG6, STMP0, STMP1, SREG7
658	001646	000000	0	: OCTAL VALUES

659									
660									
661									
662	001650	042153	EM63						
663	001652	043002	DH4						
664	001654	043632	DT4						
665	001656	000000	0						
666									
667									
668									
669	001660	037677	EM36						
670									
671	001662	043440	DH7						
672									
673	001664	043674	DT6						
674	001666	000000	0						
675									
676									
677									
678	001670	037756	EM37						
679	001672	043002	DH4						
680	001674	043632	DT4						
681	001676	000000	0						
682									
683									
684									
685	001700	040025	EM40						
686									
687	001702	043002	DH4						
688	001704	043632	DT4						
689	001706	000000	0						
690									
691									
692									
693	001710	040110	EM41						
694									
695	001712	043002	DH4						
696	001714	043632	DT4						
697	001716	000000	0						
698									
699									
700									
701	001720	040172	EM42						
702									
703	001722	043002	DH4						
704	001724	043632	DT4						
705	001726	000000	0						
706									
707									
708									
709	001730	040267	EM43						
710									
711	001732	043002	DH4						
712	001734	043632	DT4						

;ITEM 37

;DISASTEROUS ERROR BUT NO ERROR BITS SET
;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
;OCTAL VALUES

;ITEM 40

;CONTENTS OF STATUS REGISTER #2 (CDD)
;INCORRECT
;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, SREG6, STMP0, STMP1, STMP4, SREG7
;OCTAL VALUES

;ITEM 41

;NO INTERRUPT WITH PROCESSOR AT LEVEL 0
;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
;OCTAL VALUES

;ITEM 42

;DATA LATE (BIT10) NOT SET IN STATUS
;REGISTER (CDS)
;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
;OCTAL VALUES

;ITEM 43

;OFF-LINE (BIT12) NOT SET IN STATUS
;REGISTER (CDS)
;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
;OCTAL VALUES

;ITEM 44

;OFF-LINE (BIT12) SET IN STATUS REGISTER
;(CDS) - SHOULDN'T BE
;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
;OCTAL VALUES

;ITEM 45

;PICK CHECK (BIT13) NOT SET IN STATUS
;REGISTER #2 (CDD)
;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7

713	001736	000000	0	;OCTAL VALUES
714				
715				;ITEM 46
716				
717	001740	040356	EM44	;STACK CHECK (BIT12) NOT SET IN STATUS
718				;REGISTER #2 (CDD)
719	001742	043002	DH4	; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
720	001744	043632	DT4	;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
721	001746	000000	0	;OCTAL VALUES
722				
723				;ITEM 47
724				
725	001750	040446	EM45	;END OF FILE (BIT13) SET IN STATUS
726				;REGISTER (CDS) - SHOULDN'T BE
727	001752	043002	DH4	; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
728	001754	043632	DT4	;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
729	001756	000000	0	;OCTAL VALUES
730				
731				;ITEM 50
732				
733	001760	040554	EM46	;READ CHECK (BIT14) SET IN STATUS
734				;REGISTER (CDS) - SHOULDN'T BE
735	001762	043002	DH4	; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
736	001764	043632	DT4	;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
737	001766	000000	0	;OCTAL VALUES
738				
739				;ITEM 51
740				
741	001770	040661	EM47	;HOPPER CHECK (BIT02) SET IN STATUS
742				;REGISTER (CDS) - SHOULDN'T BE
743	001772	043002	DH4	; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
744	001774	043632	DT4	;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
745	001776	000000	0	;OCTAL VALUES
746				
747				;ITEM 52
748				
749	002000	040770	EM50	;END OF FILE (BIT13) OF STATUS REGISTER
750				; (CDS) NOT SET - SHOULD BE
751	002002	043002	DH4	; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
752	002004	043632	DT4	;SERRPC, STMP0, STMP1, STMP2, STMP3, SREG7
753	002006	000000	0	;OCTAL VALUES
754				
755				;ITEM 53
756				
757	002010	041076	EM51	;READ CHECK (BIT14) OF STATUS REGISTER
758				; (CDS) NOT SET - SHOULD BE
759	002012	043002	DH4	; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
760	002014	043632	DT4	;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
761	002016	000000	0	;OCTAL VALUES
762				
763				;ITEM 54
764				
765	002020	041203	EM52	;HOPPER CHECK (BIT12) OF STATUS REGISTER
766				;CDS) NOT SET - SHOULD BE

767	002022	043002	DH4	: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
768	002024	043632	DT4	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
769	002026	000000	0	: OCTAL VALUES
770				
771				: ITEM 55
772				
773	002030	041312	EMS3	: END OF FILE (BIT13) OF STATUS REGISTER
774				: (CDS) DIDN'T CLEAR WITH TRANSITION
775				: TO ON-LINE
776	002032	043002	DH4	: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
777	002034	043632	DT4	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
778	002036	000000	0	: OCTAL VALUES
779				
780				: ITEM 56
781				
782	002040	041441	EMS4	: READ CHECK (BIT14) NOT SET IN STATUS
783				: REGISTER #2 (CDD)
784	002042	043002	DH4	: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
785	002044	043632	DT4	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
786	002046	000000	0	: OCTAL VALUES
787				
788				: ITEM 57
789				
790	002050	041530	EMS5	: CARD READER ERROR BUT NO BOTH CARD
791	002052	043002	DH4	: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
792	002054	043632	DT4	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
793	002056	000000	0	: OCTAL VALUES
794				
795				: ITEM 60
796				
797	002060	041574	EMS6	: CARD READER ERROR BUT READER NOT OFF-LINE
798	002062	043002	DH4	: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
799	002064	043632	DT4	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
800	002066	000000	0	: OCTAL VALUES
801				
802				: ITEM 61
803				
804	002070	041637	EMS7	: END OF FILE ERROR (BIT13) OF STATUS
805				: REGISTER (CDS)
806	002072	043002	DH4	: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
807	002074	043632	DT4	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
808	002076	000000	0	: OCTAL VALUES
809				
810				: ITEM 62
811				
812	002100	041722	EM60	: DATA ERROR (BIT11) OF STATUS REGISTER (CDS)
813	002102	043002	DH4	: (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
814	002104	043632	DT4	: SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7
815	002106	000000	0	: OCTAL VALUES
816				
817				: ITEM 63
818				
819	002110	041776	EM61	: DATA LATE ERROR (BIT10) OF STATUS
820				: REGISTER (CDS)

821	002112	043002	DH4	;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
822	002114	043632	DT4	;\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$TMP2,\$TMP3,\$REG7
823	002116	000000	0	;\$CTAL VALUES
824				
825				;
826				;ITEM 64
827	002120	042057	EM62	;
828				NON-EXISTANT MEMORY ERROR (BIT09) OF
829	002122	043002	DH4	STATUS REGISTER (CDS)
830	002124	043632	DT4	;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
831	002126	000000	0	;\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$TMP2,\$TMP3,\$REG7
832				;\$CTAL VALUES
833				;
834				;ITEM 65
835	002130	042223	EM64	;
836				DATA PACKING (BIT01) OF STATUS REGISTER
837	002132	043002	DH4	(CDS) NOT SET - IT SHOULD BE
838	002134	043632	DT4	;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
839	002136	000000	0	;\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$TMP2,\$TMP3,\$REG7
840				;\$CTAL VALUES
841				;
842				;ITEM 66
843	002140	042331	EM65	;
844				SHOULD BE ADDRESSING BINARY DECK -
845	002142	043002	DH4	PROGRAM DOESN'T AGREE
846	002144	043632	DT4	;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
847	002146	000000	0	;\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$TMP2,\$TMP3,\$REG7
848				;\$CTAL VALUES
849				;
850				;ITEM 67
851	002150	042422	EM66	;
852				CONTENTS OF STATUS REGISTER (CDS)
853	002152	042600	DH1	INCORRECT - SHOULD BE ZERO
854	002154	043562	DT1	;(PC) (SP) (CDS) (CDD) (PS)
855	002156	000000	0	;\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$REG7
856				;\$CTAL VALUES
857				;
858				;ITEM 70
859	002160	042521	EM67	;
860				ODD BUS ADDRESS CAUSED A TRAP IN
861	002162	043002	DH4	NON-PACK MODE
862	002164	043632	DT4	;(PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
863	002166	000000	0	;\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$TMP2,\$TMP3,\$REG7
864				;\$CTAL VALUES

865											
866											
867											
868	002170					BEGIN:					
869	002170	012706	001100			MOV	#SCMTAG,R6	;; FIRST LOCATION TO BE CLEARED			
870	002174	005026				CLR	(R6)+	;; CLEAR MEMORY LOCATION			
871	002176	022706	001126			CMP	#SDDOAT,R6	;; DONE?			
872	002202	001374				BNE	.-6	;; LOOP BACK IF NO			
873	002204	012706	001100			MOV	#STACK,SP	;; SETUP THE STACK POINTER			
874	002210	012737	026024	000020		MOV	#SSCOPE,2#IOTVEC	;; IOT VECTOR FOR SCOPE ROUTINE			
875	002216	012737	000340	000022		MOV	#340,2#IOTVEC+2	;; LEVEL 7			
876	002224	012737	025650	000030		MOV	#ERROR,2#EMTVEC	;; EMT VECTOR FOR ERROR ROUTINE			
877	002232	012737	000340	000032		MOV	#340,2#EMTVEC+2	;; LEVEL 7			
878	002240	012737	027072	000034		MOV	#STRAP,2#TRAPVEC	;; TRAP VECTOR FOR TRAP CALLS			
879	002246	012737	000340	000036		MOV	#340,2#TRAPVEC+2	;; LEVEL 7			
880	002254	012737	027126	000024		MOV	#SPWRON,2#PWRVEC	;; POWER FAILURE VECTOR			
881	002262	012737	000340	000026		MOV	#340,2#PWRVEC+2	;; LEVEL 7			
882	002270	013737	014740	014732		MOV	#ENDCT,SEOPCT	;; SETUP END-OF-PROGRAM COUNTER			
883	002276	005037	001220			CLR	#TIMES	;; INITIALIZE NUMBER OF ITERATIONS			
884	002302	012737	015104	000014		MOV	#SRTN,2#TBITVEC	;; SET "T" BIT VECTOR TO SRTN			
885	002310	012737	000340	000016		MOV	#340,2#TBITVEC+2	;; LEVEL 7			
886	002316	012737	000002	015104		MOV	#RTI,SRTN	;; SET SRTN TO A RTI			
887	002324	012737	002352	000010		MOV	#655,2#RESVEC	;; TRY TO DO A RTT			
888	002332	005046				CLR	-(SP)	;; DUMMY PS			
889	002334	012746	002342			MOV	#645,-(SP)	;; AND PC			
890	002340	000006				RTT		;; TRY THE RTT			
891	002342	012737	000006	015104	645:	MOV	#RTT,SRTN	;; RTT IS LEGAL--SET SRTN TO A RTT			
892	002350	000402				BR	665				
893	002352	062706	000010		655:	ADD	#10,SP	;; RTT ILLEGAL--CLEAN OFF THE STACK			
894	002356	012737	000012	000010	665:	MOV	#RESVEC+2,2#RESVEC	;; RESTORE TRAP CATCHER			
895	002364	005037	015112			CLR	#TBIT	;; CLEAR "T" BIT SWITCH			
896	002370	012737	002370	001106		MOV	#,SLPADR	;; INITIALIZE THE LOOP ADDRESS FOR SCOPE			
897	002376	013746	000004			MOV	2#4,-(SP)	;; SAVE ERROR VECTOR			
898	002402	013746	000006			MOV	2#6,-(SP)				
899	002406	012737	002422	000004		MOV	#67\$,4	;; SET UP TIME OUT VECTOR			
900	002414	005777	176516			TST	2#SWR	;; TRY TO REFERENCE HARDWARE SWR			
901	002420	000407				BR	685	;; BRANCH IF NO TIMEOUT TRAP OCCURS			
902	002422	012737	000176	001136	675:	MOV	#SWREG,SWR	;; POINT TO SOFTWARE SWR			
903	002430	012737	000174	001140		MOV	#DISPREG,DISPLAY	;; POINT TO SOFTWARE DISPLAY REG			
904	002436	022626				CMP	(SP)+,(SP)+	;; RESTORE STACK			
905	002440	012637	000006		685:	MOV	(SP)+,2#6	;; RESTORE ERROR VECTOR			
906	002444	012637	000004			MOV	(SP)+,2#4				
907	002450	012737	002170	027274		MOV	#BEGIN,RETURN	;; SAVE RETURN POINT FOR THIS SECTION FOR POWER FAILURE RETURN			
908											
909	002456	004737	043712			JSR	PC, SETUP	;; INITIALIZE POINTERS AND FLAGS			
910	002462	104400	030254			TYPE,	#TIMES	;; TYPE MAINDEC TITLE & REV. LEVEL			
911	002466	104400	030314			TYPE,	#STADDR	;; TYPE STARTING ADDRESSES MESSAGE			
912	002472	104400	030754			TYPE,	#MLOGIC	;; INDICATE "ENTERING LOGIC TESTS"			
913	002476	104400	015131			TYPE,	#ENULL	;; TIME TO FINISH ABOVE MESSAGE			
914	002502	000432				BR		;; GO TO INSTRUCTION TESTS			
915	002504	005737	001254		RESTR:	TST	#TRFLG	;; CHECK FOR TRACE TRAPPING			
916	002510	001012				BNE		;; IF SET, TRACE TRAP			
917	002512				NOTRP:						
918	002512	013746	000340			MOV	#PR7,-(SP)	;; PUT NEW PS ON STACK			

```

919 002516 012746 002524      MOV      #64$, -(SP)      ;; PUT NEW PC ON STACK
920 002522 000002      RTI                    ;; POP NEW PC AND PS
921 002524      64$:
922 002524 104400 030754      TYPE,    MLOGIC          ; INDICATE "ENTERING LOGIC TESTS"
923 002530 104400 015131      TYPE,    $ENULL         ; TIME TO FINISH ABOVE MESSAGE
924 002534 000415      BR      TST1            ; GO TO INSTRUCTION TESTS
925 002536 104400 030754      TRAPX:  TYPE,    MLOGIC          ; INDICATE "ENTERING LOGIC TESTS"
926 002542 104400 015131      TYPE,    $ENULL         ; TIME TO FINISH ABOVE MESSAGE
927 002546 032777 010000 176362  BIT      #10000, @SWR    ; CHECK SW12
928 002554 001356      BNE     NOTRP           ; BRANCH IF SET TO CLEAR TRACE BIT
929 002556 013746 000360      MOV     TRACE, -(SP)    ; PUT NEW PS ON STACK
930 002562 012746 002570      MOV     #64$, -(SP)    ; PUT NEW PC ON STACK
931 002566 000002      RTI                    ; POP NEW PC AND PS
932 002570      64$:
933
934
935 ;*****
936 ;*TEST 1 TEST FOR INIT. OF ALL REGISTERS
937 ;*****
938 002570 000004      †TST1: SCOPE
939 002572 004737 025470      JSR     PC, CKOFFL     ; CHECK FOR OFF-LINE SET
940 002576 000005      RESET          ; SEND OUT INIT
941 002600 022713 000200      CMP     #200, @CDS    ; CHECK FOR STATUS REGISTER BIT 7 SET
942 002604 001401      BEQ     1$          ; BRANCH IF OK
943 002606 104001      ERROR +1          ; STATUS REGISTER NOT CORRECTLY INITIALIZED
944
945 002610 005714      1$: TST     @CDC      ; CHECK FOR COLUMN COUNT CLEARED
946 002612 001401      BEQ     2$          ; BR IF OK
947 002614 104002      ERROR +2          ; COLUMN COUNT NOT CLEARED BY INIT
948
949 002616 005715      2$: TST     @CDA      ; CHECK FOR BUS ADDRESS CLEARED
950 002620 001401      BEQ     3$          ; BR IF OK
951 002622 104003      ERROR +3          ; BUS ADDRESS NOT CLEARED BY INIT
952
953 002624 005777 176410      3$: TST     @CDOB     ; TEST BIT15 OF STATUS REGISTER #2
954 002630 100011      BPL     4$          ; BRANCH IF NOT SET INDICATING
955 002632 022777 107777 176400      CMP     #107777, @CDOB ; OLD CD11 CONTROLLER
956 002640 001415      BEQ     5$          ; IS CONTENTS OF STATUS REGISTER #2
957 002642 012737 107777 001210      MOV     #107777, $TMP4 ; CORRECT FOR NO ERRORS?
958 002650 104040      ERROR +40         ; BRANCH IF OK!
959 002652 000410      BR      5$          ; CORRECT CONTENTS OF 'CDD' FOR
960 002654 022777 007777 176356 4$: CMP     #007777, @CDOB ; ERROR REPORT
961 002662 001404      BEQ     5$          ; CONTENTS OF 'CDOB' STATUS REGISTER
962 002664 012737 007777 001210      MOV     #007777, $TMP4 ; #2 NOT = 107777
963 002672 104040      ERROR +40         ; GO TO NEXT TEST
964 002674      5$:
965 002676 001404      BEQ     5$          ; WE HAVE AN OLD CD11 CONTROLLER!
966 002678 012737 007777 001210      MOV     #007777, $TMP4 ; IS CONTENTS OF STATUS REGISTER #2
967 002680 104040      ERROR +40         ; CORRECT FOR NO ERRORS?
968 002682 001404      BEQ     5$          ; BRANCH IF OK!
969 002684 012737 007777 001210      MOV     #007777, $TMP4 ; CORRECT CONTENTS OF 'CDD' FOR
970 002686 104040      ERROR +40         ; ERROR REPORT
971 002688 001404      BEQ     5$          ; CONTENTS OF 'CDOB' STATUS REGISTER
972 002690 001404      BEQ     5$          ; #2 NOT = 007777

```


M07

MAINDEC - 11 - DZCDB-A
DZCDBA.P11 T2

MACY11 27(663) 19-FEB-76 14:38 PAGE 21
TEST READ/WRITE OF STATUS REGISTER

SEQ 0090

```

973 ;*TEST 2 TEST READ/WRITE OF STATUS REGISTER
974 ;*****
975 002674 000004 †TST2: SCOPE
976 ; ONLY BITS 1,4,5, AND 6 OF THE STATUS REGISTER SHOULD BE
977 ; ABLE TO BE SET TO ONE AND READ BACK AS ONE
978 002676 052713 177376 BIS #177376, @CDS ; SET ALL BITS BUT 0 AND 8
979 002702 022713 000362 CMP #362, @CDS ; ONLY BITS 1,4,5,6, AND 7 SHOULD BE SET
980 002706 001402 BEQ 1$ ; BRANCH IF OK
981 002710 104004 ERROR +4 ; STATUS REGISTER DIDN'T CONTAIN 362
982 002712 000413 BR TST3 ; BRANCH AFTER FAILURE
983
984 ; CLEARING STATUS REGISTER SHOULD CLEAR BITS 1,4,5, AND 6
985 002714 005013 1$: CLR @CDS ; CLEAR BITS 1,4,5, AND 6
986 002716 022713 000200 CMP #200, @CDS ; CHECK FOR ALL BITS CLEAR BUT 7
987 002722 001401 BEQ 2$ ; BRANCH IF OK
988 002724 104004 ERROR +4 ; STATUS REGISTER DIDN'T CONTAIN 200
989
990 ; SETTING ALL BITS SHOULD DO A POWER CLEAR
991 002726 012713 177777 2$: MOV #177777, @CDS ; SET ALL BITS OF THE STATUS REGISTER
992 002732 022713 000200 CMP #200, @CDS ; CHECK FOR ALL BITS CLEAR BUT 7
993 002736 001401 BEQ 3$ ; BRANCH IF OK
994 002740 104004 ERROR +4 ; STATUS REGISTER DIDN'T CONTAIN 200
995
996 002742 3$:
997 ;*****
998 ;*TEST 3 TEST READ/WRITE OF COLUMN COUNT REGISTER
999 ;*****
1000 002742 000004 †TST3: SCOPE
1001 002744 012714 177777 MOV #177777, @CDC ; LOAD ALL BITS
1002 002750 022714 177777 CMP #177777, @CDC ; TEST TO SEE IF IT CAN BE READ
1003 002754 001401 BEQ 1$ ; BRANCH IF OK
1004 002756 104005 ERROR +5 ; CDCC FAILED TO READ/WRITE
1005
1006 002760 022713 000200 1$: CMP #200, @CDS ; CHECK STATUS REG
1007 002764 001401 BEQ 2$ ; BRANCH IF OK
1008 002766 104004 ERROR +4 ; STATUS REG CHANGED
1009
1010 002770 052713 000400 2$: BIS #400, @CDS ; DO A POWER CLEAR
1011 002774 005714 TST @CDC ; CHECK FOR COLUMN COUNT CLEARED
1012 002776 001401 BEQ 3$ ; BRANCH IF OK
1013 003000 104006 ERROR +6 ; COLUMN COUNT NOT CLEARED BY POWER CLEAR
1014
1015 003002 022713 000200 3$: CMP #200, @CDS ; CHECK STATUS REG
1016 003006 001401 BEQ 4$ ; BRANCH IF OK
1017 003010 104004 ERROR +4 ; STATUS REG CHANGED
1018
1019 003012 4$:
1020 ;*****
1021 ;*TEST 4 TEST READ/WRITE OF BUS ADDRESS REGISTER
1022 ;*****
1023 003012 000004 †TST4: SCOPE
1024 003014 012715 177777 MOV #177777, @CDA ; LOAD ALL BITS
1025 003020 022715 177777 CMP #177777, @CDA ; TEST TO SEE IF IT CAN BE READ
1026 003024 001401 BEQ 1$ ; BRANCH IF OK

```

```

1027 003026 104007          ERROR +7          ;CDBA FAILED TO READ/WRITE
1028
1029 003030 022713 000200 15:  CMP      #200,  @CDS  ;CHECK STATUS REG
1030 003034 001401          BEQ      25          ;BRANCH IF OK
1031 003036 104004          ERROR +4          ;STATUS REG CHANGED
1032
1033 003040 052713 000400 25:  BIS      #400,  @CDS  ;DO A POWER CLEAR
1034 003044 005715          TST      @CDA          ;CHECK FOR BUS ADDRESS CLEARED
1035 003046 001401          BEQ      35          ;BRANCH IF OK
1036 003050 104010          ERROR +10         ;BUS ADDRESS NOT CLEARED BY POWER CLEAR
1037
1038 003052 022713 000200 35:  CMP      #200,  @CDS  ;CHECK STATUS REG
1039 003056 001401          BEQ      45          ;BRANCH IF OK
1040 003060 104004          ERROR +4          ;STATUS REG CHANGED
1041
1042 003062          45:
1043          ;*****
1044          ;*TEST 5      TEST CONTROLLER READY TO CLEAR BIT0
1045          ;*****
1046 003062 000004          †STS:  SCOPE
1047          ;BIT 0 SHOULD ALWAYS READ AS BEING EQUAL TO ZERO
1048 003064 004737 025470          JSR      PC,  CKOFFL ;CHECK FOR OFF-LINE SET
1049 003070 012714 177777          MOV      #-1,  @CDC  ;SET UP COLUMN COUNT TO READ 1 COLUMN
1050 003074 012715 043766          MOV      #BUFBEQ,@CDA ;SET UP BUS ADDRESS
1051          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,PROC"
1052 003100 005046          CLR      -(SP)      ;;
1053 003102 013746 000034          MOV      34, -(SP)  ;;SAVE CURRENT TRAP VECTOR
1054 003106 012737 003116 000034          MOV      #64$,34   ;;SETUP NEW TRAP VECTOT
1055 003114 104400          TRAP          ;;PUSH OLD PSW AN PCON STACK
1056 003116 016666 000002 000006 64$: MOV      2(SP),6(SP) ;;
1057 003124 012716 003132          MOV      #65$, (SP) ;;
1058 003130 000002          RTI          ;;RESTORE PSW
1059 003132 012637 000034 65$: MOV      (SP)+,34   ;;RESTORE OLD TRAP VECTOR
1060 003136 012637 001256          MOV      (SP)+,PROC
1061 003142 013746 000000          MOV      PROC, -(SP) ;;PUT NEW PS ON STACK
1062 003146 012746 003154          MOV      #66$, -(SP) ;;PUT NEW PC ON STACK
1063 003152 000002          RTI          ;;POP NEW PC AND PS
1064 003154          66$:
1065 003154 005037 001250          CLR      COUNT      ;INITIALIZE COUNTER
1066 003160 005213          INC      @CDS        ;START READING A CARD
1067 003162 105713          TSTB     @CDS        ;CHECK FOR CONTROLLER READY CLEARED
1068 003164 100001          BPL      LOOPS      ;BRANCH IF OK
1069 003166 104011          ERROR +11         ;CONTROLLER READY DIDN'T CLEAR
1070
1071 003170 032713 000001  LOOPS: BIT      #1,  @CDS  ;CHECK BIT 0
1072 003174 001402          BEQ      15          ;BRANCH IF NOT SET
1073 003176 104012          ERROR +12         ;BIT 0 READ AS A ONE
1074 003200 000423          BR      TST6        ;BRANCH AFTER FAILURE
1075 003202 005237 001250 15:  INC      COUNT      ;WAIT ABOUT
1076 003206 001370          BNE      LOOPS
1077 003210 013746 001256          MOV      PROC, -(SP) ;;PUT NEW PS ON STACK
1078 003214 012746 003222          MOV      #64$, -(SP) ;;PUT NEW PC ON STACK
1079 003220 000002          RTI          ;;POP NEW PC AND PS
1080 003222          64$:

```

```

1081 003222 105713          TSTB  2CDS          ;CHECK CONTROLLER READY
1082 003224 100401          BMI   2S           ;CONTINUE IF SET
1083 003226 104013          ERROR +13         ;CONTROLLER READY DIDN'T SET WITHIN 1 SEC
1084 003230 005713          2S:  TST  2CDS
1085 003232 100002          BPL   3S           ;
1086 003234 104014          ERROR +14         ;ERROR BIT SET
1087 003236 000404          BR    TST6
1088 003240 032713 177577 3S:  BIT  177577,2CDS ;CHECK FOR ANY OTHER BITS
1089 003244 001401          BEQ   4S           ;BRANCH IF OK
1090 003246 104015          ERROR +15         ;EXTRA BIT(S) SET
1091
1092 003250
1093
1094
1095
1096 003250 000004          4S:
1097
1098
1099 003252 004737 025470          ;*****
1100 003256 005013          ;*TEST 6          TEST BIT2 TO BE CLEAR AFTER CARD READ
1101 003260 012714 177754          ;*****
1102 003264 012715 043766          ;TST6:  SCOPE
1103 003270 005213          ;IT SHOULD REMAIN NOT SET
1104 003272 032713 063004          ;THIS SHOULD HAPPEN WITHIN ABOUT 1 SECOND
1105 003276 001401          T3R   PC,CKOFFL   ;CHECK FOR OFF-LINE SET
1106 003300 104016          CLR   2CDS        ;INITIALIZE STATUS REGISTER
1107 003302 005037 001250          1S:  MOV  8-20, 2CDC ;SET UP COLUMN COUNT TO READ 20 COLUMNS
1108
1109 003306 005046          MOV  8BUFBEQ,2CDA ;SET UP BUS ADDRESS
1110 003310 013746 000034          INC  2CDS         ;READ A CARD
1111 003314 012737 003324 000034          BIT  84,2CDS     ;CHECK HOPPER EMPTY
1112 003322 104400          BEQ   1S         ;HOPPER EMPTY SET
1113 003324 016666 000002 000006 64S: CLR  COUNT      ;SET UP WAIT COUNTER
1114 003332 012716 003340          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,PROC"
1115 003336 000002          CLR  -(SP)
1116 003340 012637 000034 65S: MOV  34,-(SF)   ;SAVE CURRENT TRAP VECTOR
1117 003344 012637 001256          MOV  864S,34     ;SETUP NEW TRAP VECTOT
1118 003350 013746 000000          TRAP ;PUSH OLD PSW AN PCON STACK
1119 003354 012746 003362          MOV  2(SP),6(SP) ;
1120 003360 000002          MOV  865S,(SP)   ;REPLACE OLD PC WITH NEW
1121 003362          RTI              ;RESTORE PSW
1122 003364 105713          65S: MOV  (SP)+,34 ;RESTORE OLD TRAP VECTOR
1123 003366 100405          MOV  (SP)+,PROC  ;
1124 003366 005337 001250          MOV  PRO,-(SP)   ;PUT NEW PS ON STACK
1125 003372 001373          MOV  866S,-(SP) ;PUT NEW PC ON STACK
1126 003374 104013          RTI              ;POP NEW PC AND PS
1127 003376 000413          66S:
1128 003400          LOOP6A: TSTB  2CDS ;CHECK READY
1129 003400 013746 001256          BMI  LOOP6B      ;BRANCH IF READY
1130 003404 012746 003412          DEC  COUNT       ;WAIT ABOUT 1 SEC.
1131 003410 000002          BNE  LOOP6A
1132 003412          ERROR +13         ;READING A CARD DIDN'T SET READY
1133 003412 105713          BR    TST7
1134 003414 100401          LOOP6B: MOV  PROC,-(SP) ;PUT NEW PS ON STACK
1134 003414 100401          MOV  864S,-(SP) ;PUT NEW PC ON STACK
1134 003414 100401          RTI              ;POP NEW PC AND PS
1134 003414 105713          64S:
1134 003414 100401          LOOP6: TSTB  2CDS ;CHECK CONTROLLER READY
1134 003414 100401          BMI  DONE6       ;BRANCH IF SET

```

```

1135 003416 104017          ERROR +17          ;RESTORING STATUS RESET READY
1136
1137 003420 005713      DONE6:  TST      2CDS          ;CHECK ERROR BIT 15
1138 003422 100001      BPL      1S          ;BRANCH IF OK
1139 003424 104014          ERROR +14          ;ERROR BIT 15 WAS SET
1140
1141 003426
1142
1143
1144
1145 003426 000004      1S:
1146 003430 004737 025442      ;*****
1147 003434 012712 003710      ;*TEST 7      TEST INTERRUPT FROM CONTROLLER READY
1148
1149
1149 003440 005046      ;*****
1150 003442 013746 000034      ;*ST7:  SCOPE
1151 003446 012737 003456 000034      JSR      PC      INIT      ;INITIALIZE
1152 003454 104400      MOV      2(ADINT),ADINT    ;LOAD RETURN POINTER
1153 003456 016666 000002 000006 64S:  CLR      -(SP)          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1154 003464 012716 003472      MOV      34, -(SP)        ;SAVE CURRENT TRAP VECTOR
1155 003470 000002      MOV      #64S, 34        ;SETUP NEW TRAP VECTOT
1156 003472 012637 000034 64S:  TRAP          ;PUSH OLD PSH AN PCON STACK
1157 003476 012637 001214      MOV      2(SP), 6(SP)    ;
1158 003502 052737 000340 001214      MOV      #65S, (SP)      ;REPLACE OLD PC WITH NEW
1159 003510 013746 001214      RTI          ;RESTORE PSH
1160 003514 012746 003522      MOV      (SP)+, 34        ;RESTORE OLD TRAP VECTOR
1161 003520 000002      MOV      (SP)+, $TMP6
1162 003522      BIS      #340, $TMP6
1163
1164 003522 005046      ;PUT NEW PS ON STACK
1165 003524 013746 000034      MOV      $TMP6, -(SP)    ;PUT NEW PC ON STACK
1166 003530 012737 003540 000034      MOV      #66S, -(SP)    ;POP NEW PC AND PS
1167 003536 104400      RTI          ;
1168 003540 016666 000002 000006 66S:  ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1169 003546 012716 003554      CLR      -(SP)          ;
1170 003552 000002      MOV      34, -(SP)      ;SAVE CURRENT TRAP VECTOR
1171 003554 012637 000034 67S:  MOV      #67S, 34        ;SETUP NEW TRAP VECTOT
1172 003560 012662 000002      TRAP          ;PUSH OLD PSH AN PCON STACK
1173
1174 003564 005046      MOV      2(SP), 6(SP)    ;
1175 003566 013746 000034      MOV      #68S, (SP)     ;REPLACE OLD PC WITH NEW
1176 003572 012737 003602 000034      RTI          ;RESTORE PSH
1177 003600 104400      MOV      (SP)+, 34        ;RESTORE OLD TRAP VECTOR
1178 003602 016666 000002 000006 68S:  MOV      (SP)+, 2(ADINT) ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1179 003610 012716 003616      CLR      -(SP)          ;
1180 003614 000002      MOV      34, -(SP)      ;SAVE CURRENT TRAP VECTOR
1181 003616 012637 000034 70S:  MOV      #69S, 34        ;SETUP NEW TRAP VECTOT
1182 003622 012637 001214      TRAP          ;PUSH OLD PSH AN PCON STACK
1183 003626 042737 000340 001214      MOV      2(SP), 6(SP)    ;
1184 003634 013746 001214      MOV      #70S, (SP)     ;REPLACE OLD PC WITH NEW
1185 003640 012746 003646      RTI          ;RESTORE PSH
1186 003644 000002      MOV      (SP)+, 34        ;RESTORE OLD TRAP VECTOR
1187 003646      MOV      (SP)+, $TMP6
1188 003646 012714 177741      BIC      #340, $TMP6
1188 003646      MOV      $TMP6, -(SP)    ;PUT NEW PS ON STACK
1188 003646      MOV      #71S, -(SP)    ;PUT NEW PC ON STACK
1188 003646      RTI          ;POP NEW PC AND PS
1188 003646      MOV      #-31., 2CDC    ;SET UP COLUMN COUNT TO READ 31 COLUMNS

```

```

1189 003652 012715 043766      MOV      #BUFBEQ, @COA      ;SET UP BUS ADDRESS
1190 003656 012713 000101      MOV      #101, @CDS        ;SET INTERRUPT ENABLE AND READ
1191 003662 105713          1S:    TSTB     @CDS              ;WAIT FOR CONTROLLER READY
1192 003664 100376          BPL      1S
1193          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
1194 003666 016246 000002      MOV      2(ADINT), -(SP)   ;;PUT NEW PS ON STACK
1195 003672 012746 003700      MOV      #72$, -(SP)      ;;PUT NEW PC ON STACK
1196 003676 000002          RTI                          ;; POP NEW PC AND PS
1197 003700          72$:
1198 003700 042713 000100      BIC      #100, @CDS        ;CLEAR INTERRUPT ENABLE
1199 003704 104020          ERROR +20                    ;NO INTERRUPT OCCURRED
1200 003706 000410          BR       CONT7
1201 003710 105713          TINT7: TSTB     @CDS          ;CHECK CONTROLLER READY
1202 003712 100401          BMI     1S                  ;BRANCH IF SET
1203 003714 104023          ERROR +23                    ;CONTROLLER READY NOT SET
1204 003716 022626          1S:    CMP      (SP)+, (SP)+ ;RESTORE STACK POINTER
1205 003720 005713          TST     @CDS                ;MAKE SURE NO ERROR OCCURRED
1206 003722 100001          BPL     2S
1207 003724 104014          ERROR +14                    ;BIT 15 WAS SET
1208 003726 005013          2S:    CLR     @CDS            ;DISABLE INTERRUPTS
1209 003730 012712 000232          CONT7: MOV     #232, @ADINT   ;CHANGE INTERRUPT RETURN ADDRESS
1210 003734 005037 000232          CLR     @#232                ;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1211
1212          ;*****
1213          ;*TEST 10      TEST NO INTERRUPT ON CONTROLLER READY & CPU AT LEVEL 7
1214          ;*****
1215 003740 000004          TST10: SCOPE
1216 003742 004737 025442          JSR     PC, INIT            ;INITIALIZE
1217 003746 012712 004120          MOV     #TINT10, @ADINT    ;SETUP RETURN
1218          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1219 003752 005046          CLR     -(SP)              ;;
1220 003754 013746 000034          MOV     34, -(SP)          ;;SAVE CURRENT TRAP VECTOR
1221 003760 012737 003770 000034          MOV     #64$, 34          ;;SETUP NEW TRAP VECTOT
1222 003766 104400          TRAP                          ;;PUSH OLD PSW AN PC ON STACK
1223 003770 016666 000002 000006 64$: MOV     2(SP), 6(SP)        ;;
1224 003776 012716 004004          MOV     #65$, (SP)        ;;
1225 004002 000002          RTI                          ;;RESTORE PSW
1226 004004 012637 000034 65$: MOV     (SP)+, 34          ;;RESTORE OLD TRAP VECTOR
1227 004010 012637 001214          MOV     (SP)+, $TMP6
1228 004014 052737 000340 001214          BIS     #340, $TMP6
1229 004022 013746 001214          MOV     $TMP6, -(SP)      ;;PUT NEW PS ON STACK
1230 004026 012746 004034          MOV     #66$, -(SP)      ;;PUT NEW PC ON STACK
1231 004032 000002          RTI                          ;; POP NEW PC AND PS
1232 004034          66$:
1233          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1234 004034 005046          CLR     -(SP)              ;;
1235 004036 013746 000034          MOV     34, -(SP)          ;;SAVE CURRENT TRAP VECTOR
1236 004042 012737 004052 000034          MOV     #67$, 34          ;;SETUP NEW TRAP VECTOT
1237 004050 104400          TRAP                          ;;PUSH OLD PSW AN PC ON STACK
1238 004052 016666 000002 000006 67$: MOV     2(SP), 6(SP)        ;;
1239 004060 012716 004066          MOV     #68$, (SP)        ;;
1240 004064 000002          RTI                          ;;RESTORE PSW
1241 004066 012637 000034 68$: MOV     (SP)+, 34          ;;RESTORE OLD TRAP VECTOR
1242 004072 012662 000002          MOV     (SP)+, 2(ADINT)

```

E08

MAINDEC - 11 - DZCDB-A
DZCDBA.P11 T10

MACY11 27(663) 19-FEB-76 14:38 PAGE 26
TEST NO INTERRUPT ON CONTROLLER READY & CPU AT LEVEL 7

SEQ 0095

1243	004076	012714	177703	MOV	#-61, 2CDC	;SET UP COLUMN COUNT TO READ 61 COLUMNS
1244	004102	012715	043766	MOV	#BUFBEQ, 2CDA	;SET UP BUS ADDRESS
1245	004106	012713	000101	MOV	#101, 2CDS	;SET INTERRUPT ENABLE AND READ
1246	004112	105713		1\$: TSTB	2CDS	;WAIT FOR CONTROLLER READY
1247	004114	100376		BPL	1\$	
1248	004116	000402		BR	T10GO	;CONTINUE IF NO INTERRUPT OCCURRED
1249	004120	104021		TINT10: ERROR +21		;AN INTERRUPT OCCURRED
1250	004122	022626		CMP	(SP)+, (SP)+	;RESTORE STACK POINTER
1251	004124	005013		T10GO: CLR	2CDS	;CLEAR INTERRUPT ENABLE
1252	004126	012712	000232	MOV	#232, 2ADINT	;CHANGE INTERRUPT RETURN ADDRESS
1253	004132	005037	000232	CLR	#232	;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1254						
1255						;FIND THE LEVEL AT WHICH AN INTERRUPT OCCURS
1256						;PRINT OUT A MESSAGE STATING THIS LEVEL IF IT IS OTHER THAN THE STANDARD
1257						; (LEVEL 6) MAKE CERTAIN THAT IT ALWAYS OCCURS AT THIS LEVEL
1258						; THE MESSAGE STATING THE LEVEL IS PRINTED ONLY ONCE, AND THE PROGRAM MUST
1259						; BE STARTED OVER AT LOCATION 200 FOR IT TO BE PRINTED AGAIN
1260						
1261						*****
1262						*TEST 11 TEST FOR AN INTERRUPT ON LEVEL 7
1263						*****
1264	004136	000004		TST11: SCOPE		
1265	004140	004737	025442	JSR	PC, INIT	;INITIALIZE
1266	004144	012712	004526	MOV	#TINT11, 2ADINT	;SETUP RETURN ADDRESS
1267						;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1268	004150	005046		CLR	-(SP)	
1269	004152	013746	000034	MOV	34, -(SP)	;SAVE CURRENT TRAP VECTOR
1270	004156	012737	004166	MOV	#64\$, 34	;SETUP NEW TRAP VECTOR
1271	004164	104400		TRAP		;PUSH OLD PSW AN PCON STACK
1272	004166	016666	000002	64\$: MOV	2(SP), 6(SP)	
1273	004174	012716	004202	MOV	#65\$, (SP)	
1274	004200	000002		RTI		;RESTORE PSW
1275	004202	012637	000034	65\$: MOV	(SP)+, 34	;RESTORE OLD TRAP VECTOR
1276	004206	012637	001214	MOV	(SP)+, \$TMP6	
1277	004212	052737	000340	BIS	#340, \$TMP6	
1278	004220	013746	001214	MOV	\$TMP6, -(SP)	;PUT NEW PS ON STACK
1279	004224	012746	004232	MOV	#66\$, -(SP)	;PUT NEW PC ON STACK
1280	004230	000002		RTI		;POP NEW PC AND PS
1281	004232			66\$:		
1282						;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)."
1283	004232	005046		CLR	-(SP)	
1284	004234	013746	000034	MOV	34, -(SP)	;SAVE CURRENT TRAP VECTOR
1285	004240	012737	004250	MOV	#67\$, 34	;SETUP NEW TRAP VECTOR
1286	004246	104400		TRAP		;PUSH OLD PSW AN PCON STACK
1287	004250	016666	000002	67\$: MOV	2(SP), 6(SP)	
1288	004256	012716	004264	MOV	#68\$, (SP)	
1289	004262	000002		RTI		;RESTORE PSW
1290	004264	012637	000034	68\$: MOV	(SP)+, 34	;RESTORE OLD TRAP VECTOR
1291	004270	012662	000002	MOV	(SP)+, 2(ADINT)	
1292						;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1293	004274	005046		CLR	-(SP)	
1294	004276	013746	000034	MOV	34, -(SP)	;SAVE CURRENT TRAP VECTOR
1295	004302	012737	004312	MOV	#69\$, 34	;SETUP NEW TRAP VECTOR
1296	004310	104400		TRAP		;PUSH OLD PSW AN PCON STACK

1297	004312	016666	000002	000006	69S:	MOV	2(SP), 6(SP)	::	
1298	004320	012716	004326			MOV	#70S, (SP)	::	REPLACE OLD PC WITH NEW
1299	004324	000002				RTI		::	RESTORE PSW
1300	004326	012637	000034		70S:	MOV	(SP)+, 34	::	RESTORE OLD TRAP VECTOR
1301	004332	012637	001214			MOV	(SP)+, \$TMP6		
1302	004336	042737	000340	001214		BIC	#340, \$TMP6		
1303	004344	013746	001214			MOV	\$TMP6, -(SP)	::	PUT NEW PS ON STACK
1304	004350	012746	004356			MOV	#71S, -(SP)	::	PUT NEW PC ON STACK
1305	004354	000002				RTI		::	POP NEW PC AND PS
1306	004356				71S:				
1307	004356	005046				CLR	-(SP)	::	
1308	004360	013746	000034			MOV	34, -(SP)	::	SAVE CURRENT TRAP VECTOR
1309	004364	012737	004374	000034		MOV	#72S, 34	::	SETUP NEW TRAP VECTOR
1310	004372	104400				TRAP		::	PUSH OLD PSW AN PC ON STACK
1311	004374	016666	000002	000006	72S:	MOV	2(SP), 6(SP)	::	
1312	004402	012716	004410			MOV	#73S, (SP)	::	REPLACE OLD PC WITH NEW
1313	004406	000002				RTI		::	RESTORE PSW
1314	004410	012637	000034		73S:	MOV	(SP)+, 34	::	RESTORE OLD TRAP VECTOR
1315	004414	014637	001214			MOV	-(SP), \$TMP6		
1316	004420	052737	000300	001214		BIS	#300, \$TMP6		
1317	004426	013746	001214			MOV	\$TMP6, -(SP)	::	PUT NEW PS ON STACK
1318	004432	012746	004440			MOV	#74S, -(SP)	::	PUT NEW PC ON STACK
1319	004436	000002				RTI		::	POP NEW PC AND PS
1320	004440				74S:				
1321	004440	012714	177660			MOV	#-80, @CDC	::	SET UP COLUMN COUNT TO READ 80 COLUMNS
1322	004444	012715	043766			MOV	#BUFBEG, @CDA	::	SET UP BUS ADDRESS
1323	004450	012713	000101			MOV	#101, @CDS	::	SET INTERRUPT ENABLE AND READ
1324	004454	105713			1S:	TSTB	@CDS	::	WAIT FOR CONTROLLER READY
1325	004456	100376				BPL	1S		
1326									THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(@DINT), PS"
1327	004460	016246	000002			MOV	2(@DINT), -(SP)	::	PUT NEW PS ON STACK
1328	004464	012746	004472			MOV	#75S, -(SP)	::	PUT NEW PC ON STACK
1329	004470	000002				RTI		::	POP NEW PC AND PS
1330	004472				75S:				
1331	004472	005013				CLR	@CDS	::	DISABLE INTERRUPTS
1332	004474	012712	000232			MOV	#232, @ADINT	::	CHANGE INTERRUPT RETURN ADDRESS
1333	004500	005037	000232			CLR	@#232	::	TO CAUSE A HALT IF AN INTERRUPT OCCURS
1334	004504	005737	001252			TST	INTFLG	::	TEST FOR A PREVIOUS INTERRUPT
1335	004510	001441				BEQ	TST12	::	BRANCH IF NONE
1336	004512	023727	001252	100007		CMP	INTFLG, #100007	::	CHECK PREVIOUS LEVEL
1337	004520	100435				BMI	TST12	::	BRANCH IF LOWER
1338	004522	104022				ERROR	+22	::	INTERRUPT ALREADY OCCURRED AT LVL 7 OR HIGHER
1339	004524	000433				BR	TST12		
1340	004526	105713			TINT11:	TSTB	@CDS	::	MAKE SURE CONTROLLER READY IS SET
1341	004530	100401				BMI	1S	::	BRANCH IF SET
1342	004532	104023				ERROR	+23	::	CONTROLLER READY WASN'T SET
1343	004534	005013			1S:	CLR	@CDS	::	DISABLE FURTHER INTERRUPTS
1344	004536	012712	000232			MOV	#232, @ADINT	::	CHANGE INTERRUPT RETURN ADDRESS
1345	004542	005037	000232			CLR	@#232	::	TO CAUSE A HALT IF AN INTERRUPT OCCURS
1346	004546	022626				CMP	(SP)+, (SP)+	::	RESTORE STACK POINTER
1347	004550	005737	001252			TST	INTFLG	::	CHECK FOR PREVIOUS FLAG
1348	004554	100412				BMI	SET7	::	BRANCH IF FLAG SET
1349	004556	012737	100007	001252		MOV	#100007, INTFLG	::	SET FLAG AND LEVEL
1350	004564	104400	031430			TYPE,	MSG4	::	PRINT MESSAGE "THE INTERRUPT LEVEL WAS"


```

1351 004570 012746 000007      MOV      #7,-(SP)
1352 004574 104402      TYPOS
1353 004576      001      .BYTE 1
1354 004577      000      .BYTE 0
1355 004600 000405      BR      TST12
1356 004602 023727 001252 100007 SET7: CMP      INTFLG, #100007 ;CHECK PREVIOUS LEVEL
1357 004610 100001      BPL     TST12
1358 004612 104024      ERROR +24 ;INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1359
1360 ;*****
1361 ;*TEST 12 TEST FOR AN INTERRUPT ON LEVEL 6
1362 ;*****
1362 004614 000004      TST12: SCOPE
1363 004616 004737 025442      JSR     PC, INIT ;INITIALIZE
1364 004622 012712 005204      MOV     #TINT12,ADINT ;SETUP RETURN ADDRESS
1365 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1366 004626 005046      CLR     -(SP) ;
1367 004630 013746 000034      MOV     34,-(SP) ;SAVE CURRENT TRAP VECTOR
1368 004634 012737 004644 000034      MOV     #64$,34 ;SETUP NEW TRAP VECTOT
1369 004642 104400      TRAP ;PUSH OLD PSW AN PCOM STACK
1370 004644 016666 000002 000006 64$: MOV     2(SP),6(SP) ;
1371 004652 012716 004660      MOV     #65$,1(SP) ;
1372 004656 000002 ;REPLACE OLD PC WITH NEW
1373 004660 012637 000034 65$: RTI ;RESTORE PSW
1374 004664 012637 001214      MOV     (SP)+,34 ;RESTORE OLD TRAP VECTOR
1375 004670 052737 000340 001214      BIS     #340,$TMP6
1376 004676 013746 001214      MOV     $TMP6,-(SP) ;PUT NEW PS ON STACK
1377 004702 012746 004710      MOV     #66$,-(SP) ;PUT NEW PC ON STACK
1378 004706 000002 ;POP NEW PC AND PS
1379 004710 66$: RTI
1380 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1381 004710 005046      CLR     -(SP) ;
1382 004712 013746 000034      MOV     34,-(SP) ;SAVE CURRENT TRAP VECTOR
1383 004716 012737 004726 000034      MOV     #67$,34 ;SETUP NEW TRAP VECTOT
1384 004724 104400      TRAP ;PUSH OLD PSW AN PCOM STACK
1385 004726 016666 000002 000006 67$: MOV     2(SP),6(SP) ;
1386 004734 012716 004742      MOV     #68$,1(SP) ;
1387 004740 000002 ;REPLACE OLD PC WITH NEW
1388 004742 012637 000034 68$: RTI ;RESTORE PSW
1389 004746 012662 000002      MOV     (SP)+,34 ;RESTORE OLD TRAP VECTOR
1390 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1391 004752 005046      CLR     -(SP) ;
1392 004754 013746 000034      MOV     34,-(SP) ;SAVE CURRENT TRAP VECTOR
1393 004760 012737 004770 000034      MOV     #69$,34 ;SETUP NEW TRAP VECTOT
1394 004766 104400      TRAP ;PUSH OLD PSW AN PCOM STACK
1395 004770 016666 000002 000006 69$: MOV     2(SP),6(SP) ;
1396 004776 012716 005004      MOV     #70$,1(SP) ;
1397 005002 000002 ;REPLACE OLD PC WITH NEW
1398 005004 012637 000034 70$: RTI ;RESTORE PSW
1399 005010 012637 001214      MOV     (SP)+,34 ;RESTORE OLD TRAP VECTOR
1400 005014 042737 000340 001214      BIC     #340,$TMP6
1401 005022 013746 001214      MOV     $TMP6,-(SP) ;PUT NEW PS ON STACK
1402 005026 012746 005034      MOV     #71$,-(SP) ;PUT NEW PC ON STACK
1403 005032 000002 ;POP NEW PC AND PS
1404 005034 71$: RTI

```

```

1405 005034 005046 CLR -(SP)
1406 005036 013746 000034 MOV 34, -(SP) ;SAVE CURRENT TRAP VECTOR
1407 005042 012737 005052 000034 MOV #72$, 34 ;SETUP NEW TRAP VECTOT
1408 005050 104400 TRAP ;PUSH OLD PSW AN PC ON STACK
1409 005052 016666 000002 000006 72$: MOV 2(SP), 6(SP)
1410 005060 012716 005066 MOV #73$, (SP) ;REPLACE OLD PC WITH NEW
1411 005064 000002 RTI ;RESTORE PSW
1412 005066 012637 000034 73$: MOV (SP)+, 34 ;RESTORE OLD TRAP VECTOR
1413 005072 014637 001214 MOV -(SP), $TMP6
1414 005076 052737 000240 001214 BIS #240, $TMP6
1415 005104 013746 001214 MOV $TMP6, -(SP) ;PUT NEW PS ON STACK
1416 005110 012746 005116 MOV #74$, -(SP) ;PUT NEW PC ON STACK
1417 005114 000002 RTI ;POP NEW PC AND PS
1418 005116 74$:
1419 005116 012714 177660 MOV #-80, @CDC ;SET UP COLUMN COUNT TO READ 80 COLUMNS
1420 005122 012715 043766 MOV #BUFBE$, @CDA ;SET UP BUS ADDRESS
1421 005126 012713 000101 MOV #101, @CDS ;SET INTERRUPT ENABLE AND READ
1422 005132 105713 1$: TSTB @CDS ;WAIT FOR CONTROLLER READY
1423 005134 100376 BPL 1$
1424 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT), PS"
1425 005136 016246 000002 MOV 2(ADINT), -(SP) ;PUT NEW PS ON STACK
1426 005142 012746 005150 MOV #75$, -(SP) ;PUT NEW PC ON STACK
1427 005146 000002 RTI ;POP NEW PC AND PS
1428 005150 75$:
1429 005150 005013 CLR @CDS ;DISABLE INTERRUPTS
1430 005152 012712 000232 MOV #232, @ADINT ;CHANGE INTERRUPT RETURN ADDRESS
1431 005156 005037 000232 CLR @#232 ;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1432 005162 005737 001252 TST INTFLG ;TEST FOR A PREVIOUS INTERRUPT
1433 005166 001441 BEQ TST13 ;BRANCH IF NONE
1434 005170 023727 001252 100006 CMP INTFLG, #100006 ;CHECK PREVIOUS LEVEL
1435 005176 100435 BMI TST13 ;BRANCH IF LOWER
1436 005200 104022 ERROR +22 ;INTERRUPT ALREADY OCCURRED AT LVL 6 OR HIGHER
1437 005202 000433 BR TST13
1438 005204 105713 TINT12: TSTB @CDS ;MAKE SURE CONTROLLER READY IS SET
1439 005206 100401 BMI 1$ ;BRANCH IF SET
1440 005210 104023 ERROR +23 ;CONTROLLER READY WASN'T SET
1441 005212 005013 1$: CLR @CDS ;DISABLE FURTHER INTERRUPTS
1442 005214 012712 000232 MOV #232, @ADINT ;CHANGE INTERRUPT RETURN ADDRESS
1443 005220 005037 000232 CLR @#232 ;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1444 005224 022626 CMP (SP)+, (SP)+ ;RESTORE STACK POINTER
1445 005226 005737 001252 TST INTFLG ;CHECK FOR PREVIOUS FLAG
1446 005232 100412 BMI SET6 ;BRANCH IF FLAG SET
1447 005234 012737 100006 001252 MOV #100006, INTFLG ;SET FLAG AND LEVEL
1448 005242 104400 331430 TYPE, MSG4 ;PRINT MESSAGE "THE INTERRUPT LEVEL WAS"
1449 005246 012746 000006 MOV #6, -(SP)
1450 005252 104402 TYPOS
1451 005254 001 .BYTE 1
1452 005255 000 .BYTE 0
1453 005256 000405 BR TST13
1454 005260 023727 001252 100006 SET6: CMP INTFLG, #100006 ;CHECK PREVIOUS LEVEL
1455 005266 100001 BPL TST13
1456 005270 104024 ERROR +24 ;INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1457 ;*****
1458 ;*TEST 13 TEST FOR AN INTERRUPT ON LEVEL 5

```

```

1459          :*****
1460 005272 000004 †ST13: SCOPE
1461 005274 004737 025442 JSR PC INIT : INITIALIZE
1462 005300 012712 005662 MOV #INT13,ADINT : SETUP RETURN ADDRESS
1463          : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1464 005304 005046 CLR -(SP)
1465 005306 013746 000034 MOV 34, -(SP) : SAVE CURRENT TRAP VECTOR
1466 005312 012737 005322 000034 MOV #64$,34 : SETUP NEW TRAP VECTOT
1467 005320 104400 TRAP : PUSH OLD PSW AN PCOM STACK
1468 005322 016666 000002 000006 64$: MOV 2(SP), 6(SP)
1469 005330 012716 005336 MOV #65$, (SP) : REPLACE OLD PC WITH NEW
1470 005334 000002 RTI : RESTORE PSW
1471 005336 012637 000034 65$: MOV (SP)+, 34 : RESTORE OLD TRAP VECTOR
1472 005342 012637 001214 MOV (SP)+, $TMP6
1473 005346 052737 000340 001214 BIS #340, $TMP6
1474 005354 013746 001214 MOV $TMP6, -(SP) : PUT NEW PS ON STACK
1475 005360 012746 005366 MOV #66$, -(SP) : PUT NEW PC ON STACK
1476 005364 000002 RTI : POP NEW PC AND PS
1477 005366 66$:
1478          : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1479 005366 005046 CLR -(SP)
1480 005370 013746 000034 MOV 34, -(SP) : SAVE CURRENT TRAP VECTOR
1481 005374 012737 005404 000034 MOV #67$,34 : SETUP NEW TRAP VECTOT
1482 005402 104400 TRAP : PUSH OLD PSW AN PCOM STACK
1483 005404 016666 000002 000006 67$: MOV 2(SP), 6(SP)
1484 005412 012716 005420 MOV #68$, (SP) : REPLACE OLD PC WITH NEW
1485 005416 000002 RTI : RESTORE PSW
1486 005420 012637 000034 68$: MOV (SP)+, 34 : RESTORE OLD TRAP VECTOR
1487 005424 012662 000002 MOV (SP)+, 2(ADINT)
1488          : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1489 005430 005046 CLR -(SP)
1490 005432 013746 000034 MOV 34, -(SP) : SAVE CURRENT TRAP VECTOR
1491 005436 012737 005446 000034 MOV #69$,34 : SETUP NEW TRAP VECTOT
1492 005444 104400 TRAP : PUSH OLD PSW AN PCOM STACK
1493 005446 016666 000002 000006 69$: MOV 2(SP), 6(SP)
1494 005454 012716 005462 MOV #70$, (SP) : REPLACE OLD PC WITH NEW
1495 005460 000002 RTI : RESTORE PSW
1496 005462 012637 000034 70$: MOV (SP)+, 34 : RESTORE OLD TRAP VECTOR
1497 005466 012637 001214 MOV (SP)+, $TMP6
1498 005472 042737 000340 001214 BIC #340, $TMP6
1499 005500 013746 001214 MOV $TMP6, -(SP) : PUT NEW PS ON STACK
1500 005504 012746 005512 MOV #71$, -(SP) : PUT NEW PC ON STACK
1501 005510 000002 RTI : POP NEW PC AND PS
1502 005512 71$:
1503 005512 005046 CLR -(SP)
1504 005514 013746 000034 MOV 34, -(SP) : SAVE CURRENT TRAP VECTOR
1505 005520 012737 005530 000034 MOV #72$,34 : SETUP NEW TRAP VECTOT
1506 005526 104400 TRAP : PUSH OLD PSW AN PCOM STACK
1507 005530 016666 000002 000006 72$: MOV 2(SP), 6(SP)
1508 005536 012716 005544 MOV #73$, (SP) : REPLACE OLD PC WITH NEW
1509 005542 000002 RTI : RESTORE PSW
1510 005544 012637 000034 73$: MOV (SP)+, 34 : RESTORE OLD TRAP VECTOR
1511 005550 014637 001214 MOV -(SP), $TMP6
1512 005554 052737 000200 001214 BIS #200, $TMP6
    
```

```

1513 005562 013746 001214      MOV      STMP6,-(SP)      ;;PUT NEW PS ON STACK
1514 005566 012746 005574      MOV      #74$,-(SP)     ;;PUT NEW PC ON STACK
1515 005572 000002      RTI                          ;;POP NEW PC AND PS
1516 005574      74$:
1517 005574 012714 177660      MOV      #-80, @CDC      ;;SET UP COLUMN COUNT TO READ 80 COLUMNS
1518 005600 012715 043766      MOV      #BUFBEQ, @CDA   ;;SET UP BUS ADDRESS
1519 005604 012713 000101      MOV      #101, @CDS     ;;SET INTERRUPT ENABLE AND READ
1520 005610 105713      1$:      TSTB    @CDS            ;;WAIT FOR CONTROLLER READY
1521 005612 100376      BPL     1$
1522      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
1523 005614 016246 000002      MOV      2(ADINT),-(SP)  ;;PUT NEW PS ON STACK
1524 005620 012746 005626      MOV      #75$,-(SP)     ;;PUT NEW PC ON STACK
1525 005624 000002      RTI                          ;;POP NEW PC AND PS
1526 005626      75$:
1527 005626 005013      CLR     @CDS              ;;DISABLE INTERRUPTS
1528 005630 012712 000232      MOV      #232, @ADINT    ;;CHANGE INTERRUPT RETURN ADDRESS
1529 005634 005037 000232      CLR     @#232            ;;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1530 005640 005737 001252      TST     INTFLG           ;;TEST FOR A PREVIOUS INTERRUPT
1531 005644 001441      BEQ     TST14            ;;BRANCH IF NONE
1532 005646 023727 001252 100005      CMP     INTFLG, #100005  ;;CHECK PREVIOUS LEVEL
1533 005654 100435      BMT     TST14            ;;BRANCH IF LOWER
1534 005656 104022      ERROR  +22              ;;INTERRUPT ALREADY OCCURRED AT LVL 5 OR HIGHER
1535 005660 000433      BR     TST14
1536 005662 105713      TINT13: TSTB    @CDS        ;;MAKE SURE CONTROLLER READY IS SET
1537 005664 100401      BMT     1$              ;;BRANCH IF SET
1538 005666 104023      ERROR  +23              ;;CONTROLLER READY WASN'T SET
1539 005670 005013      1$:      CLR     @CDS              ;;DISABLE FURTHER INTERRUPTS
1540 005672 012712 000232      MOV      #232, @ADINT    ;;CHANGE INTERRUPT RETURN ADDRESS
1541 005676 005037 000232      CLR     @#232            ;;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1542 005702 022626      CMP     (SP)+, (SP)+     ;;RESTORE STACK POINTER
1543 005704 005737 001252      TST     INTFLG           ;;CHECK FOR PREVIOUS FLAG
1544 005710 100412      BMT     SET5             ;;BRANCH IF FLAG SET
1545 005712 012737 100005 001252      MOV      #100005, INTFLG ;;SET FLAG AND LEVEL
1546 005720 104400 031430      TYPE,   MSG4             ;;PRINT MESSAGE "THE INTERRUPT LEVEL WAS"
1547 005724 012746 000005      MOV      #5,-(SP)
1548 005730 104402      TYPOS
1549 005732      .BYTE  1
1550 005733      .BYTE  0
1551 005734 000405      BR     TST14
1552 005736 023727 001252 100005      SET5:  CMP     INTFLG, #100005 ;;CHECK PREVIOUS LEVEL
1553 005744 100001      BPL     TST14
1554 005746 104024      ERROR  +24              ;;INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1555      ;*****
1556      ;*TEST 14 TEST FOR AN INTERRUPT ON LEVEL 4
1557      ;*****
1558 005750 000004      TST14: SCOPE
1559 005752 004737 025442      JSR     PC, INIT        ;;INITIALIZE
1560 005756 012712 006340      MOV      #TINT14, @ADINT ;;SETUP RETURN ADDRESS
1561      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #34,PS"
1562 005762 005046      CLR     -(SP)           ;;
1563 005764 013746 000034      MOV      34,-(SP)       ;;SAVE CURRENT TRAP VECTOR
1564 005770 012737 006000 000034      MOV      #64$,34        ;;SETUP NEW TRAP VECTOT
1565 005776 104400      TRAP
1566 006000 016666 000002 000006 64$:  MOV      2(SP),6(SP)    ;;PUSH OLD PSW AN PC ON STACK

```

```

1567 006006 012716 006014      MOV      #65$, (SP)          ;; REPLACE OLD PC WITH NEW
1568 006012 000002      RTI                          ;; RESTORE PSW
1569 006014 012637 000034      65$: MOV      (SP)+, 34      ;; RESTORE OLD TRAP VECTOR
1570 006020 012637 001214      MOV      (SP)+, $TMP6
1571 006024 052737 000340 001214      BIS      #340, $TMP6
1572 006032 013746 001214      MOV      $TMP6, -(SP)      ;; PUT NEW PS ON STACK
1573 006036 012746 006044      MOV      #66$, -(SP)      ;; PUT NEW PC ON STACK
1574 006042 000002      RTI                          ;; POP NEW PC AND PS
1575 006044      66$:
1576      ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1577 006044 005046      CLR      -(SP)
1578 006046 013746 000034      MOV      34, -(SP)      ;; SAVE CURRENT TRAP VECTOR
1579 006052 012737 006062 000034      MOV      #67$, 34      ;; SETUP NEW TRAP VECTOR
1580 006060 104400      TRAP
1581 006062 016666 000002 000006      67$: MOV      2(SP), 6(SP)      ;;
1582 006070 012716 006076      MOV      #68$, (SP)      ;; REPLACE OLD PC WITH NEW
1583 006074 000002      RTI                          ;; RESTORE PSW
1584 006076 012637 000034      68$: MOV      (SP)+, 34      ;; RESTORE OLD TRAP VECTOR
1585 006102 012662 000002      MOV      (SP)+, 2(ADINT)
1586      ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1587 006106 005046      CLR      -(SP)
1588 006110 013746 000034      MOV      34, -(SP)      ;; SAVE CURRENT TRAP VECTOR
1589 006114 012737 006124 000034      MOV      #69$, 34      ;; SETUP NEW TRAP VECTOR
1590 006122 104400      TRAP
1591 006124 016666 000002 000006      69$: MOV      2(SP), 6(SP)      ;;
1592 006132 012716 006140      MOV      #70$, (SP)      ;; REPLACE OLD PC WITH NEW
1593 006136 000002      RTI                          ;; RESTORE PSW
1594 006140 012637 000034      70$: MOV      (SP)+, 34      ;; RESTORE OLD TRAP VECTOR
1595 006144 012637 001214      MOV      (SP)+, $TMP6
1596 006150 042737 000340 001214      BIC      #340, $TMP6
1597 006156 013746 001214      MOV      $TMP6, -(SP)      ;; PUT NEW PS ON STACK
1598 006162 012746 006170      MOV      #71$, -(SP)      ;; PUT NEW PC ON STACK
1599 006166 000002      RTI                          ;; POP NEW PC AND PS
1600 006170      71$:
1601 006170 005046      CLR      -(SP)
1602 006172 013746 000034      MOV      34, -(SP)      ;; SAVE CURRENT TRAP VECTOR
1603 006176 012737 006206 000034      MOV      #72$, 34      ;; SETUP NEW TRAP VECTOR
1604 006204 104400      TRAP
1605 006206 016666 000002 000006      72$: MOV      2(SP), 6(SP)      ;;
1606 006214 012716 006222      MOV      #73$, (SP)      ;; REPLACE OLD PC WITH NEW
1607 006220 000002      RTI                          ;; RESTORE PSW
1608 006222 012637 000034      73$: MOV      (SP)+, 34      ;; RESTORE OLD TRAP VECTOR
1609 006226 014637 001214      MOV      -(SP), $TMP6
1610 006232 052737 000140 001214      BIS      #140, $TMP6
1611 006240 013746 001214      MOV      $TMP6, -(SP)      ;; PUT NEW PS ON STACK
1612 006244 012746 006252      MOV      #74$, -(SP)      ;; PUT NEW PC ON STACK
1613 006250 000002      RTI                          ;; POP NEW PC AND PS
1614 006252      74$:
1615 006252 012714 177660      MOV      #-80, @CDC      ;; SET UP COLUMN COUNT TO READ 80 COLUMNS
1616 006256 012715 043766      MOV      #BUFBEQ, @CDA   ;; SET UP BUS ADDRESS
1617 006262 012713 000101      MOV      #101, @CDS     ;; SET INTERRUPT ENABLE AND READ
1618 006266 105713      1$: TSTB    @CDS          ;; WAIT FOR CONTROLLER READY
1619 006270 100376      BPL     1$
1620      ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"

```

```

1621 006272 016246 000002      MOV      2(ADINT),-(SP)  ;; PUT NEW PS ON STACK
1622 006276 012746 006304      MOV      #75$,-(SP)    ;; PUT NEW PC ON STACK
1623 006302 000002      RTI                          ;; POP NEW PC AND PS
1624 006304      75$:
1625 006304 005013      CLR      @CDS             ; DISABLE INTERRUPTS
1626 006306 012712 000232      MOV      #232, @ADINT   ; CHANGE INTERRUPT RETURN ADDRESS
1627 006312 005037 000232      CLR      @#232          ; TO CAUSE A HALT IF AN INTERRUPT OCCURS
1628 006316 005737 001252      TST     INTFLG          ; TEST FOR A PREVIOUS INTERRUPT
1629 006322 001433      BEQ     TST15           ; BRANCH IF NONE
1630 006324 023727 001252 100004      CMP     INTFLG, #100004 ; CHECK PREVIOUS LEVEL
1631 006332 100427      BMI     TST15           ; BRANCH IF LOWER
1632 006334 104022      ERROR +22              ; INTERRUPT ALREADY OCCURRED AT LVL 4 OR HIGHER
1633 006336 000425      BR      TST15
1634 006340 105713      TINT14: TSTB @CDS        ; MAKE SURE CONTROLLER READY IS SET
1635 006342 100401      BMI     1$             ; BRANCH IF SET
1636 006344 104023      ERROR +23              ; CONTROLLER READY WASN'T SET
1637 006346 005013      1$: CLR @CDS           ; DISABLE FURTHER INTERRUPTS
1638 006350 012712 000232      MOV      #232, @ADINT   ; CHANGE INTERRUPT RETURN ADDRESS
1639 006354 005037 000232      CLR      @#232          ; TO CAUSE A HALT IF AN INTERRUPT OCCURS
1640 006360 022626      CMP     (SP)+, (SP)+    ; RESTORE STACK POINTER
1641 006362 005737 001252      TST     INTFLG          ; CHECK FOR PREVIOUS FLAG
1642 006366 100404      BMI     SET4            ; BRANCH IF FLAG SET
1643 006370 012737 100004 001252      MOV      #100004, INTFLG ; SET FLAG AND LEVEL
1644 006376 000405      BR      TST15
1645 006400 023727 001252 100004 SET4: CMP INTFLG, #100004 ; CHECK PREVIOUS LEVEL
1646 006406 100001      BPL     TST15
1647 006410 104024      ERROR +24              ; INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1648 ;*****
1649 ;*TEST IS TEST FOR AN INTERRUPT ON LEVEL 3
1650 ;*****
1651 006412 000004      TST15: SCOPE
1652 006414 001737 025442      JSR     PC, INIT        ; INITIALIZE
1653 006420 012712 007002      MOV     #TINT15, @ADINT ; SETUP RETURN ADDRESS
1654 ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1655 006424 005046      CLR     -(SP)
1656 006426 013746 000034      MOV     34, -(SP)      ; SAVE CURRENT TRAP VECTOR
1657 006432 012737 006442 000034      MOV     #64$, 34      ; SETUP NEW TRAP VECTOR
1658 006440 104400      TRAP
1659 006442 016666 000002 000006 64$: MOV 2(SP), 6(SP)
1660 006450 012716 006456      MOV     #65$, (SP)    ;
1661 006454 000002      RTI                          ; RESTORE PSW
1662 006456 012637 000034      65$: MOV (SP)+, 34
1663 006462 012637 001214      MOV     (SP)+, $TMP6   ; RESTORE OLD TRAP VECTOR
1664 006466 052737 000340 001214      BIS     #340, $TMP6
1665 006474 013746 001214      MOV     $TMP6, -(SP)   ; PUT NEW PS ON STACK
1666 006500 012746 006506      MOV     #66$, -(SP)   ; PUT NEW PC ON STACK
1667 006504 000002      RTI                          ; POP NEW PC AND PS
1668 006506      66$:
1669 ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1670 006506 005046      CLR     -(SP)
1671 006510 013746 000034      MOV     34, -(SP)     ; SAVE CURRENT TRAP VECTOR
1672 006514 012737 006524 000034      MOV     #67$, 34     ; SETUP NEW TRAP VECTOR
1673 006522 104400      TRAP
1674 006524 016666 000002 000006 67$: MOV 2(SP), 6(SP)

```

MO8

1675	006532	012716	006540			MOV	#68\$, (SP)		;; REPLACE OLD PC WITH NEW
1676	006536	000002				RTI			;; RESTORE PSW
1677	006540	012637	000034		68\$:	MOV	(SP)+, 34		;; RESTORE OLD TRAP VECTOR
1678	006544	012662	000002			MOV	(SP)+, 2(ADINT)		
1679						; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340, PS"			
1680	006550	005046				CLR	-(SP)		
1681	006552	013746	000034			MOV	34, -(SP)		;; SAVE CURRENT TRAP VECTOR
1682	006556	012737	006566	000034		MOV	#69\$, 34		;; SETUP NEW TRAP VECTOR
1683	006564	104400				TRAP			;; PUSH OLD PSW AN PCON STACK
1684	006566	016666	000002	000006	69\$:	MOV	2(SP), 6(SP)		
1685	006574	012716	006602			MOV	#70\$, (SP)		;; REPLACE OLD PC WITH NEW
1686	006600	000002				RTI			;; RESTORE PSW
1687	006602	012637	000034		70\$:	MOV	(SP)+, 34		;; RESTORE OLD TRAP VECTOR
1688	006606	012637	001214			MOV	(SP)+, \$TMP6		
1689	006612	042737	000340	001214		BIC	#340, \$TMP6		
1690	006620	013746	001214			MOV	\$TMP6, -(SP)		;; PUT NEW PS ON STACK
1691	006624	012746	006632			MOV	#71\$, -(SP)		;; PUT NEW PC ON STACK
1692	006630	000002				RTI			;; POP NEW PC AND PS
1693	006632				71\$:				
1694	006632	005046				CLR	-(SP)		
1695	006634	013746	000034			MOV	34, -(SP)		;; SAVE CURRENT TRAP VECTOR
1696	006640	012737	006650	000034		MOV	#72\$, 34		;; SETUP NEW TRAP VECTOR
1697	006646	104400				TRAP			;; PUSH OLD PSW AN PCON STACK
1698	006650	016666	000002	000006	72\$:	MOV	2(SP), 6(SP)		
1699	006656	012716	006664			MOV	#73\$, (SP)		;; REPLACE OLD PC WITH NEW
1700	006662	000002				RTI			;; RESTORE PSW
1701	006664	012637	000034		73\$:	MOV	(SP)+, 34		;; RESTORE OLD TRAP VECTOR
1702	006670	014637	001214			MOV	-(SP), \$TMP6		
1703	006674	052737	000100	001214		BIS	#100, \$TMP6		
1704	006702	013746	001214			MOV	\$TMP6, -(SP)		;; PUT NEW PS ON STACK
1705	006706	012746	006714			MOV	#74\$, -(SP)		;; PUT NEW PC ON STACK
1706	006712	000002				RTI			;; POP NEW PC AND PS
1707	006714				74\$:				
1708	006714	012714	177660			MOV	#-80, \$ACDC		;; SET UP COLUMN COUNT TO READ 80 COLUMNS
1709	006720	012715	043766			MOV	#BUF8EG, \$ACDA		;; SET UP BUS ADDRESS
1710	006724	012713	000101			MOV	#101, \$ACDS		;; SET INTERRUPT ENABLE AND READ
1711	006730	105713			1\$:	TSTB	\$ACDS		;; WAIT FOR CONTROLLER READY
1712	006732	100376				BPL	1\$		
1713						; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT), PS"			
1714	006734	016246	000002			MOV	2(ADINT), -(SP)		;; PUT NEW PS ON STACK
1715	006740	012746	006746			MOV	#75\$, -(SP)		;; PUT NEW PC ON STACK
1716	006744	000002				RTI			;; POP NEW PC AND PS
1717	006746				75\$:				
1718	006746	005013				CLR	\$ACDS		;; DISABLE INTERRUPTS
1719	006750	012712	000232			MOV	#232, \$ADINT		;; CHANGE INTERRUPT RETURN ADDRESS
1720	006754	005037	000232			CLR	\$#232		;; TO CAUSE A HALT IF AN INTERRUPT OCCURS
1721	006760	005737	001252			TST	INTFLG		;; TEST FOR A PREVIOUS INTERRUPT
1722	006764	001441				BEQ	TST16		;; BRANCH IF NONE
1723	006766	023727	001252	100003		CMP	INTFLG, #100003		;; CHECK PREVIOUS LEVEL
1724	006774	100435				BMI	TST16		;; BRANCH IF LOWER
1725	006776	104022				ERROR	+22		;; INTERRUPT ALREADY OCCURRED AT LVL 3 OR HIGHER
1726	007000	000433				BR	TST16		
1727	007002	105713			TINT15:	TSTB	\$ACDS		;; MAKE SURE CONTROLLER READY IS SET
1728	007004	100401				BMI	1\$;; BRANCH IF SET


```

1729 007006 104023          ERROR +23          ;CONTROLLER READY WASN'T SET
1730 007010 005013          1$: CLR          2CD5          ;DISABLE FURTHER INTERRUPTS
1731 007012 012712 000232  MOV          #232, 2ADINT ;CHANGE INTERRUPT RETURN ADDRESS
1732 007016 005037 000232  CLR          2#232          ;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1733 007022 022626          CMP          (SP)+, (SP)+ ;RESTORE STACK POINTER
1734 007024 005737 001252  TST          INTFLG        ;CHECK FOR PREVIOUS FLAG
1735 007030 100412          BMI          SET3          ;BRANCH IF FLAG SET
1736 007032 012737 100003 001252  MOV          #100003, INTFLG ;SET FLAG AND LEVEL
1737 007040 104400 031430  TYPE,        MSG4          ;PRINT MESSAGE "THE INTERRUPT LEVEL WAS"
1738 007044 012746 000003  MOV          #3, -(SP)
1739 007050 104402          TYPOS
1740 007052          001          .BYTE        1
1741 007053          000          .BYTE        0
1742 007054 000405          BR          TST16
1743 007056 023727 001252 100003 SET3: CMP          INTFLG, #100003 ;CHECK PREVIOUS LEVEL
1744 007064 100001          BPL          TST16
1745 007066 104024          ERROR +24          ;INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1746          ;*****
1747          ;*TEST 16 TEST FOR AN INTERRUPT ON LEVEL 2
1748          ;*****
1749 007070 000004          TST16: SCOPE
1750 007072 004737 025442  JSR          PC, INIT      ;INITIALIZE
1751 007076 012712 007460  MOV          #TINT16, 2ADINT ;SETUP RETURN ADDRESS
1752          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1753 007102 005046          CLR          -(SP)
1754 007104 013746 000034  MOV          34, -(SP)      ;SAVE CURRENT TRAP VECTOR
1755 007110 012737 007120 000034  MOV          #64$, 34      ;SETUP NEW TRAP VECTOT
1756 007116 104400          TRAP
1757 007120 016666 000002 000006 64$: MOV          2(SP), 6(SP) ;PUSH OLD PSW AN PCON STACK
1758 007126 012716 007134  MOV          #65$, (SP)    ;REPLACE OLD PC WITH NEW
1759 007132 000002          RTI          ;RESTORE PSW
1760 007134 012637 000034 65$: MOV          (SP)+, 34   ;RESTORE OLD TRAP VECTOR
1761 007140 012637 001214  MOV          (SP)+, $TMP6
1762 007144 052737 000340 001214  BIS          #340, $TMP6
1763 007152 013746 001214  MOV          $TMP6, -(SP) ;PUT NEW PS ON STACK
1764 007156 012746 007164  MOV          #66$, -(SP) ;PUT NEW PC ON STACK
1765 007162 000002          RTI          ;POP NEW PC AND PS
1766 007164          66$:
1767          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1768 007164 005046          CLR          -(SP)
1769 007166 013746 000034  MOV          34, -(SP)    ;SAVE CURRENT TRAP VECTOR
1770 007172 012737 007202 000034  MOV          #67$, 34     ;SETUP NEW TRAP VECTOT
1771 007200 104400          TRAP
1772 007202 016666 000002 000006 67$: MOV          2(SP), 6(SP) ;PUSH OLD PSW AN PCON STACK
1773 007210 012716 007216  MOV          #68$, (SP)  ;REPLACE OLD PC WITH NEW
1774 007214 000002          RTI          ;RESTORE PSW
1775 007216 012637 000034 68$: MOV          (SP)+, 34   ;RESTORE OLD TRAP VECTOR
1776 007222 012662 000002  MOV          (SP)+, 2(ADINT)
1777          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1778 007226 005046          CLR          -(SP)
1779 007230 013746 000034  MOV          34, -(SP)    ;SAVE CURRENT TRAP VECTOR
1780 007234 012737 007244 000034  MOV          #69$, 34     ;SETUP NEW TRAP VECTOT
1781 007242 104400          TRAP
1782 007244 016666 000002 000006 69$: MOV          2(SP), 6(SP) ;PUSH OLD PSW AN PCON STACK

```

```

1783 007252 012716 007260      MOV      #70$, (SP)          ;; REPLACE OLD PC WITH NEW
1784 007256 000002      RTI                          ;; RESTORE PSW
1785 007260 012637 000034      70$:   MOV      (SP)+, 34          ;; RESTORE OLD TRAP VECTOR
1786 007264 012637 001214      MOV      (SP)+, $TMP6
1787 007270 042737 000340 001214      BIC      #340, $TMP6
1788 007276 013746 001214      MOV      $TMP6, -(SP)      ;; PUT NEW PS ON STACK
1789 007302 012746 007310      MOV      #71$, -(SP)      ;; PUT NEW PC ON STACK
1790 007306 000002      RTI                          ;; POP NEW PC AND PS
1791 007310      71$:
1792 007310 005046      CLR      -(SP)              ;;
1793 007312 013746 000034      MOV      34, -(SP)          ;; SAVE CURRENT TRAP VECTOR
1794 007316 012737 007326 000034      MOV      #72$, 34          ;; SETUP NEW TRAP VECTOR
1795 007324 104400      TRAP
1796 007326 016666 000002 000006 72$:   MOV      2(SP), 6(SP)      ;; PUSH OLD PSW AN PC ON STACK
1797 007334 012716 007342      MOV      #73$, (SP)
1798 007340 000002      RTI                          ;; REPLACE OLD PC WITH NEW
1799 007342 012637 000034      73$:   MOV      (SP)+, 34          ;; RESTORE PSW
1800 007346 014637 001214      MOV      -(SP), $TMP6      ;; RESTORE OLD TRAP VECTOR
1801 007352 052737 000040 001214      BIS      #040, $TMP6
1802 007360 013746 001214      MOV      $TMP6, -(SP)      ;; PUT NEW PS ON STACK
1803 007364 012746 007372      MOV      #74$, -(SP)      ;; PUT NEW PC ON STACK
1804 007370 000002      RTI                          ;; POP NEW PC AND PS
1805 007372      74$:
1806 007372 012714 177660      MOV      #-80, @CDB        ;; SET UP COLUMN COUNT TO READ 80 COLUMNS
1807 007376 012715 043766      MOV      @BFBEG, @CDBA     ;; SET UP BUS ADDRESS
1808 007402 012713 000101      MOV      #101, @CDBS      ;; SET INTERRUPT ENABLE AND READ
1809 007406 105713      1$:   TSTB     @CDBS            ;; WAIT FOR CONTROLLER READY
1810 007410 100376      BPL     1$
1811      ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT), PS"
1812 007412 016246 000002      MOV      2(ADINT), -(SP)   ;; PUT NEW PS ON STACK
1813 007416 012746 007424      MOV      #75$, -(SP)      ;; PUT NEW PC ON STACK
1814 007422 000002      RTI                          ;; POP NEW PC AND PS
1815 007424      75$:
1816 007424 005013      CLR      @CDBS            ;; DISABLE INTERRUPTS
1817 007426 012712 000232      MOV      #232, @ADINT     ;; CHANGE INTERRUPT RETURN ADDRESS
1818 007432 005037 000232      CLR      @232            ;; TO CAUSE A HALT IF AN INTERRUPT OCCURS
1819 007436 005737 001252      TST     INTFLG          ;; TEST FOR A PREVIOUS INTERRUPT
1820 007442 001441      BEQ     TST17           ;; BRANCH IF NONE
1821 007444 023727 001252 100002      CMP     INTFLG, #100002   ;; CHECK PREVIOUS LEVEL
1822 007452 100435      BMI     TST17           ;; BRANCH IF LOWER
1823 007454 104022      ERROR +22              ;; INTERRUPT ALREADY OCCURRED AT LVL 2 OR HIGHER
1824 007456 000433      BR     TST17
1825 007460 105713      TINT16: TSTB @CDBS        ;; MAKE SURE CONTROLLER READY IS SET
1826 007462 100401      BMI     1$              ;; BRANCH IF SET
1827 007464 104023      ERROR +23              ;; CONTROLLER READY WASN'T SET
1828 007466 005013      1$:   CLR      @CDBS            ;; DISABLE FURTHER INTERRUPTS
1829 007470 012712 000232      MOV      #232, @ADINT     ;; CHANGE INTERRUPT RETURN ADDRESS
1830 007474 005037 000232      CLR      @232            ;; TO CAUSE A HALT IF AN INTERRUPT OCCURS
1831 007500 022626      CMP     (SP)+, (SP)+     ;; RESTORE STACK POINTER
1832 007502 005737 001252      TST     INTFLG          ;; CHECK FOR PREVIOUS FLAG
1833 007506 100412      BMI     SET2            ;; BRANCH IF FLAG SET
1834 007510 012737 100002 001252      MOV      #100002, INTFLG  ;; SET FLAG AND LEVEL
1835 007516 104400 031430      TYPE,   MSG4            ;; PRINT MESSAGE "THE INTERRUPT LEVEL WAS"
1836 007522 012746 000002      MOV      #2, -(SP)
    
```

```

1837 007526 104402          TYPOS
1838 007530          001      .BYTE      1
1839 007531          000      .BYTE      0
1840 007532 000405          BR          TST17
1841 007534 023727 001252 100002 SET2: CMP      INTFLG, #100002 ;CHECK PREVIOUS LEVEL
1842 007542 100001          BPL          TST17
1843 007544 104024          ERROR +24          ;INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1844
1845          ;*****
1846          ;*TEST 17      TEST FOR AN INTERRUPT ON LEVEL 1
1847          ;*****
1847 007546 000004          TST17: SCOPE
1848 007550 004737 025442      JSR      PC      INIT      ;INITIALIZE
1849 007554 012712 010120      MOV      #TINT17,ADINT ;SETUP RETURN ADDRESS
1850          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1851 007560 005046          CLR      -(SP)
1852 007562 013746 000034      MOV      34, -(SP)      ;SAVE CURRENT TRAP VECTOR
1853 007566 012737 007576 000034      MOV      #64$,34      ;SETUP NEW TRAP VECTOR
1854 007574 104400          TRAP
1855 007576 016666 000002 000006 64$: MOV      2(SP),6(SP)      ;PUSH OLD PSH AN PCON STACK
1856 007604 012716 007612      MOV      #65$, (SP)
1857 007610 000002          RTI
1858 007612 012637 000034 65$: MOV      (SP)+,34      ;REPLACE OLD PC WITH NEW
1859 007616 012637 001214          MOV      (SP)+,$TMP6      ;RESTORE PSH
1860 007622 052737 000340 001214      BIS      #340,$TMP6      ;RESTORE OLD TRAP VECTOR
1861 007630 013746 001214      MOV      $TMP6, -(SP)
1862 007634 012746 007642      MOV      #66$, -(SP)
1863 007640 000002          RTI
1864 007642          66$:
1865          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1866 007642 005046          CLR      -(SP)
1867 007644 013746 000034      MOV      34, -(SP)
1868 007650 012737 007660 000034      MOV      #67$,34
1869 007656 104400          TRAP
1870 007660 016666 000002 000006 67$: MOV      2(SP),6(SP)
1871 007666 012716 007674      MOV      #68$, (SP)
1872 007672 000002          RTI
1873 007674 012637 000034 68$: MOV      (SP)+,34
1874 007700 012662 000002      MOV      (SP)+,2(ADINT)
1875          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1876 007704 005046          CLR      -(SP)
1877 007706 013746 000034      MOV      34, -(SP)
1878 007712 012737 007722 000034      MOV      #69$,34
1879 007720 104400          TRAP
1880 007722 016666 000002 000006 69$: MOV      2(SP),6(SP)
1881 007730 012716 007736      MOV      #70$, (SP)
1882 007734 000002          RTI
1883 007736 012637 000034 70$: MOV      (SP)+,34
1884 007742 012637 001214          MOV      (SP)+,$TMP6
1885 007746 042737 000340 001214      BIC      #340,$TMP6
1886 007754 013746 001214      MOV      $TMP6, -(SP)
1887 007760 012746 007766      MOV      #71$, -(SP)
1888 007764 000002          RTI
1889 007766          71$:
1890 007766 005046          CLR      -(SP)

```

```

1891 007770 013746 000034      MOV      34, -(SP)      ;; SAVE CURRENT TRAP VECTOR
1892 007774 012737 010004 000034      MOV      #72$, 34      ;; SETUP NEW TRAP VECTOR
1893 010002 104400      TRAP                      ;; PUSH OLD PSW AN PC ON STACK
1894 010004 016666 000002 000006 72$:      MOV      2(SP), 6(SP)  ;;
1895 010012 012716 010020      MOV      #73$, (SP)    ;;
1896 010016 000002      RTI                      ;; RESTORE PSW
1897 010020 012637 000034 73$:      MOV      (SP)+, 34      ;; RESTORE OLD TRAP VECTOR
1898 010024 014637 001214      MOV      -(SP), $TMP6
1899 010030 052737 000000 001214      BIS      #000, $TMP6
1900 010036 013746 001214      MOV      $TMP6, -(SP)  ;; PUT NEW PS ON STACK
1901 010042 012746 010050      MOV      #74$, -(SP)  ;; PUT NEW PC ON STACK
1902 010046 000002      RTI                      ;; POP NEW PC AND PS
1903 010050 74$:
1904 010050 012714 177660      MOV      #80, #CDB      ;; SET UP COLUMN COUNT TO READ 80 COLUMNS
1905 010054 012715 043766      MOV      #BUF BEG, #CDBA ;; SET UP BUS ADDRESS
1906 010060 012713 000101      MOV      #101, #CDBS   ;; SET INTERRUPT ENABLE AND READ
1907 010064 105713 1$:      TSTB    #CDBS          ;; WAIT FOR CONTROLLER READY
1908 010066 100376      BPL     1$
1909      ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT), PS"
1910 010070 016246 000002      MOV      2(ADINT), -(SP) ;; PUT NEW PS ON STACK
1911 010074 012746 010102      MOV      #75$, -(SP)    ;; PUT NEW PC ON STACK
1912 010100 000002      RTI                      ;; POP NEW PC AND PS
1913 010102 75$:
1914 010102 005013      CLR     #CDBS          ;; DISABLE INTERRUPTS
1915 010104 012712 000232      MOV      #232, #ADINT  ;; CHANGE INTERRUPT RETURN ADDRESS
1916 010110 005037 000232      CLR     #232          ;; TO CAUSE A HALT IF AN INTERRUPT OCCURS
1917 010114 104041      ERROR  +41           ;; NO INTERRUPT WITH PROCESSOR AT LEVEL C
1918 010116 000433      BR     #TST20
1919 010120 105713 75$:      TSTB    #CDBS          ;; MAKE SURE CONTROLLER READY IS SET
1920 010122 100401      BMI     1$           ;; BRANCH IF SET
1921 010124 104023      ERROR  +23           ;; CONTROLLER READY WASN'T SET
1922 010126 005013 1$:      CLR     #CDBS          ;; DISABLE FURTHER INTERRUPTS
1923 010130 012712 000232      MOV      #232, #ADINT  ;; CHANGE INTERRUPT RETURN ADDRESS
1924 010134 005037 000232      CLR     #232          ;; TO CAUSE A HALT IF AN INTERRUPT OCCURS
1925 010140 022626      CMP     (SP)+, (SP)+  ;; RESTORE STACK POINTER
1926 010142 005737 001252      TST     INTFLG        ;; CHECK FOR PREVIOUS FLAG
1927 010146 100412      BMI     SET1          ;; BRANCH IF FLAG SET
1928 010150 012737 100001 001252      MOV      #100001, INTFLG ;; SET FLAG AND LEVEL
1929 010156 104400 031430      TYPE,   MSG4          ;; PRINT MESSAGE "THE INTERRUPT LEVEL WAS"
1930 010162 012746 000001      MOV      #1, -(SP)
1931 010166 104402      TYPOS
1932 010170      .BYTE  1
1933 010171      .BYTE  0
1934 010172 000405      BR     #TST20
1935 010174 023727 001252 100001 SET1: CMP    INTFLG, #100001 ;; CHECK PREVIOUS LEVEL
1936 010202 100001      BPL     #TST20
1937 010204 104024      ERROR  +24           ;; INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1938
1939      ;*****
1940      ;*TEST 20      TEST NO INTERRUPT WITH IE SET & REST CLEARED
1941      ;*****
1942 010206 000004      TST20: SCOPE
1943 010210 004737 025442      JSR     PC, INIT      ;; INITIALIZE CSR TO ZERO
1944 010214 012712 010422      MOV     #TINT20, #ADINT ;; SETUP RETURN ADDRESS

```

E09

MAINDEC - 11 - DZCDB-A
DZCDBA.P11 T20

MACY11 27(663) 19-FEB-76 14:38 PAGE 39
TEST NO INTERRUPT WITH IE SET & REST CLEARED

SEQ 0108

```

1945                                     ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1946 010220 005046                       CLR      -(SP)
1947 010222 013746 000034                 MOV      34, -(SP)
1948 010226 012737 010236 000034         MOV      #64$, 34
1949 010234 104400                       TRAP
1950 010236 016666 000002 000006 64$:   MOV      2(SP), 6(SP)
1951 010244 012716 010252                 MOV      #65$, (SP)
1952 010250 000002                       RTI
1953 010252 012637 000034 65$:           MOV      (SP)+, 34
1954 010256 012637 001214                 MOV      (SP)+, $TMP6
1955 010262 052737 000340 001214         BIS      #340, $TMP6
1956 010270 013746 001214                 MOV      $TMP6, -(SP)
1957 010274 012746 010302                 MOV      #66$, -(SP)
1958 010300 000002                       RTI
1959 010302                               66$:
1960                                     ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1961 010302 005046                       CLR      -(SP)
1962 010304 013746 000034                 MOV      34, -(SP)
1963 010310 012737 010320 000034         MOV      #67$, 34
1964 010316 104400                       TRAP
1965 010320 016666 000002 000006 67$:   MOV      2(SP), 6(SP)
1966 010326 012716 010334                 MOV      #68$, (SP)
1967 010332 000002                       RTI
1968 010334 012637 000034 68$:           MOV      (SP)+, 34
1969 010340 012662 000002                 MOV      (SP)+, 2(ADINT)
1970 010344 013746 000000                 MOV      PRO, -(SP)
1971 010350 012746 010356                 MOV      #69$, -(SP)
1972 010354 000002                       RTI
1973 010356                               69$:
1974 010356 012714 177777                 MOV      #-1, @CDC
1975 010362 012715 043766                 MOV      #BUFBEQ, @CDA
1976 010366 012713 000100                 MOV      #100, @CDS
1977 010372 005037 001250                 CLR      COUNT
1978 010376 005237 001250 1$:           INC      COUNT
1979 010402 001375                         BNE     1$
1980                                     ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
1981 010404 016246 000002                 MOV      2(ADINT), -(SP)
1982 010410 012746 010416                 MOV      #70$, -(SP)
1983 010414 000002                       RTI
1984 010416                               70$:
1985 010416 005013                         CLR      @CDS
1986 010420 000403                         BR      CONT20
1987 010422 104021                         TINT20: ERROR +21
1988 010424 022626                         CMP      (SP)+, (SP)+
1989 010426 005013                         CLR      @CDS
1990 010430 005037 000232                 CONT20: CLR      @#232
1991 010434 012712 000232                 MOV      #232, @ADINT
1992
1993 ;*****
1994 ;*TEST 21 SIMULTANEOUS INTERRUPTS AT MORE THAN 1 LEVEL
1995 ;*****
1996 010440 000004                         TST21: SCOPE
1997 010442 004737 025442                 JSR     PC, INIT
1998 010446 012712 010676                 MOV     #TINT21, @ADINT

```

```

; SAVE CURRENT TRAP VECTOR
; SETUP NEW TRAP VECTOR
; PUSH OLD PSW AN PC ON STACK
;
; REPLACE OLD PC WITH NEW
; RESTORE PSW
; RESTORE OLD TRAP VECTOR
;
; PUT NEW PS ON STACK
; PUT NEW PC ON STACK
; POP NEW PC AND PS

```

```

; SAVE CURRENT TRAP VECTOR
; SETUP NEW TRAP VECTOR
; PUSH OLD PSW AN PC ON STACK
;
; REPLACE OLD PC WITH NEW
; RESTORE PSW
; RESTORE OLD TRAP VECTOR
;
; PUT NEW PS ON STACK
; PUT NEW PC ON STACK
; POP NEW PC AND PS

```

```

; SET UP COLUMN COUNT TO READ 1 COLUMN
; SET UP BUS ADDRESS
; ENABLE INTERRUPTS
; INITIALIZE COUNTER
; WAIT AWHILE

```

```

; PUT NEW PS ON STACK
; PUT NEW PC ON STACK
; POP NEW PC AND PS
; DISABLE FURTHER INTERRUPTS
; AN INTERRUPT OCCURRED
; RESTORE STACK
; DISABLE FURTHER INTERRUPTS
; CHANGE INTERRUPT RETURN ADDRESS TO
; CAUSE A HALT IF AN INTERRUPT OCCURS

```

```

;*****
;*TEST 21 SIMULTANEOUS INTERRUPTS AT MORE THAN 1 LEVEL
;*****
TST21: SCOPE
      JSR     PC, INIT
      MOV     #TINT21, @ADINT

```

```

1999                                     :THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
2000 010452 005046 CLR -(SP)
2001 010454 013746 000034 MOV 34, -(SP) ;;SAVE CURRENT TRAP VECTOR
2002 010460 012737 010470 000034 MOV #64$, 34 ;;SETUP NEW TRAP VECTOR
2003 010466 104400 TRAP ;;PUSH OLD PSW AN PCOM STACK
2004 010470 016666 000002 000006 64$: MOV 2(SP), 6(SP)
2005 010476 012716 010504 MOV #65$, (SP) ;;REPLACE OLD PC WITH NEW
2006 010502 000002 RTI ;;RESTORE PSW
2007 010504 012637 000034 65$: MOV (SP)+, 34 ;;RESTORE OLD TRAP VECTOR
2008 010510 012637 001214 MOV (SP)+, $TMP6
2009 010514 052737 000340 001214 BIS #340, $TMP6
2010 010522 013746 001214 MOV $TMP6, -(SP) ;;PUT NEW PS ON STACK
2011 010526 012746 010534 MOV #66$, -(SP) ;;PUT NEW PC ON STACK
2012 010532 000002 RTI ;;POP NEW PC AND PS
2013 010534 66$:
2014                                     :THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
2015 010534 005046 CLR -(SP)
2016 010536 013746 000034 MOV 34, -(SP) ;;SAVE CURRENT TRAP VECTOR
2017 010542 012737 010552 000034 MOV #67$, 34 ;;SETUP NEW TRAP VECTOR
2018 010550 104400 TRAP ;;PUSH OLD PSW AN PCOM STACK
2019 010552 016666 000002 000006 67$: MOV 2(SP), 6(SP)
2020 010560 012716 010566 MOV #68$, (SP) ;;REPLACE OLD PC WITH NEW
2021 010564 000002 RTI ;;RESTORE PSW
2022 010566 012637 000034 68$: MOV (SP)+, 34 ;;RESTORE OLD TRAP VECTOR
2023 010572 012662 000002 MOV (SP)+, 2(ADINT)
2024                                     :THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
2025 010576 005046 CLR -(SP)
2026 010600 013746 000034 MOV 34, -(SP) ;;SAVE CURRENT TRAP VECTOR
2027 010604 012737 010614 000034 MOV #69$, 34 ;;SETUP NEW TRAP VECTOR
2028 010612 104400 TRAP ;;PUSH OLD PSW AN PCOM STACK
2029 010614 016666 000002 000006 69$: MOV 2(SP), 6(SP)
2030 010622 012716 010630 MOV #70$, (SP) ;;REPLACE OLD PC WITH NEW
2031 010626 000002 RTI ;;RESTORE PSW
2032 010630 012637 000034 70$: MOV (SP)+, 34 ;;RESTORE OLD TRAP VECTOR
2033 010634 012637 001214 MOV (SP)+, $TMP6
2034 010640 042737 000340 001214 BIC #340, $TMP6
2035 010646 013746 001214 MOV $TMP6, -(SP) ;;PUT NEW PS ON STACK
2036 010652 012746 010660 MOV #71$, -(SP) ;;PUT NEW PC ON STACK
2037 010656 000002 RTI ;;POP NEW PC AND PS
2038 010660 71$:
2039 010660 012714 177777 MOV #-1, $CDC ;;SET UP COLUMN COUNT TO READ 1 COLUMN
2040 010664 012715 043766 MOV #BUFBEQ, $CDA ;;SET UP BUS ADDRESS
2041 010670 012713 000101 MOV #101, $CDS ;;SET INTERRUPT ENABLE AND READ
2042 010674 000777 BR ;;WAIT FOR INTERRUPT
2043 010676 022626 TINT21: CMP (SP)+, (SP)+ ;;RESTORE STACK POINTER
2044 010700 012712 010734 MOV #TINA21, $ADINT ;;CHANGE RETRUN ADDRESS
2045 010704 013746 000000 MOV PRO, -(SP) ;;PUT NEW PS ON STACK
2046 010710 012746 010716 MOV #64$, -(SP) ;;PUT NEW PC ON STACK
2047 010714 000002 RTI ;;POP NEW PC AND PS
2048 010716 64$:
2049 010716 000240 NOP ;;WAIT
2050                                     :THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
2051 010720 016246 000002 MOV 2(ADINT), -(SP) ;;PUT NEW PS ON STACK
2052 010724 012746 010732 MOV #65$, -(SP) ;;PUT NEW PC ON STACK

```

```

2053 010730 000002          RTI          ;; POP NEW PC AND PS
2054 010732
2055 010732 000402          65$: BR      CONT21
2056 010734 022626          TINA21: CMP   (SP)+, (SP)+ ;RESTORE STACK
2057 010736 104025          ERROR +25 ;THE INTERRUPT OCCURRED AT 2 LEVELS
2058 010740 005013          CONT21: CLR   2CDS          ;DISABLE INTERRUPTS
2059 010742 005037 000232   CLR   2#232          ;CHANGE INTERRUPT RETURN ADDRESS TO
2060 010746 012712 000232   MOV   #232, 2ADINT ;CAUSE A HALT IF AN INTERRUPT OCCURS
2061
2062
2063 ;*****
2064 ;*TEST 22      NON-EXISTANT MEMORY DETECTION
2065 ;*****
2065 010752 000004          †ST22: SCOPE
2066 010754 004737 025442   JSR   PC, INIT      ;INITIALIZE CSR TO ZERO
2067 010760 012712 011210   MOV   #TINT22, 2ADINT ;SETUP RETURN ADDRESS
2068 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
2069 010764 005046          CLR   -(SP)          ;;
2070 010766 013746 000034   MOV   34, -(SP)      ;SAVE CURRENT TRAP VECTOR
2071 010772 012737 011002 000034   MOV   #64$, 34       ;SETUP NEW TRAP VECTOT
2072 011000 104400          TRAP          ;; PUSH OLD PSW AN PCOM STACK
2073 011002 016666 000002 000006 64$: MOV   2(SP), 6(SP)    ;;
2074 011010 012716 011016   MOV   #65$, (SP)     ;REPLACE OLD PC WITH NEW
2075 011014 000002          RTI          ;; RESTORE PSW
2076 011016 012637 000034 65$: MOV   (SP)+, 34      ;RESTORE OLD TRAP VECTOR
2077 011022 012637 001214   MOV   (SP)+, $TMP6
2078 011026 052737 000340 001214   BIS   #340, $TMP6
2079 011034 013746 001214   MOV   $TMP6, -(SP)   ;; PUT NEW PS ON STACK
2080 011040 012746 011046   MOV   #66$, -(SP)   ;; PUT NEW PC ON STACK
2081 011044 000002          RTI          ;; POP NEW PC AND PS
2082 011046
2083 66$:
2084 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
2084 011046 005046          CLR   -(SP)          ;;
2085 011050 013746 000034   MOV   34, -(SP)      ;SAVE CURRENT TRAP VECTOR
2086 011054 012737 011064 000034   MOV   #67$, 34       ;SETUP NEW TRAP VECTOT
2087 011062 104400          TRAP          ;; PUSH OLD PSW AN PCOM STACK
2088 011064 016666 000002 000006 67$: MOV   2(SP), 6(SP)    ;;
2089 011072 012716 011100   MOV   #68$, (SP)     ;REPLACE OLD PC WITH NEW
2090 011076 000002          RTI          ;; RESTORE PSW
2091 011100 012637 000034 68$: MOV   (SP)+, 34      ;RESTORE OLD TRAP VECTOR
2092 011104 012662 000002   MOV   (SP)+, 2(ADINT)
2093 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
2094 011110 005046          CLR   -(SP)          ;;
2095 011112 013746 000034   MOV   34, -(SP)      ;SAVE CURRENT TRAP VECTOR
2096 011116 012737 011126 000034   MOV   #69$, 34       ;SETUP NEW TRAP VECTOT
2097 011124 104400          TRAP          ;; PUSH OLD PSW AN PCOM STACK
2098 011126 016666 000002 000006 69$: MOV   2(SP), 6(SP)    ;;
2099 011134 012716 011142   MOV   #70$, (SP)     ;REPLACE OLD PC WITH NEW
2100 011140 000002          RTI          ;; RESTORE PSW
2101 011142 012637 000034 70$: MOV   (SP)+, 34      ;RESTORE OLD TRAP VECTOR
2102 011146 012637 001214   MOV   (SP)+, $TMP6
2103 011152 042737 000340 001214   BIC   #340, $TMP6
2104 011160 013746 001214   MOV   $TMP6, -(SP)   ;; PUT NEW PS ON STACK
2105 011164 012746 011172   MOV   #71$, -(SP)   ;; PUT NEW PC ON STACK
2106 011170 000002          RTI          ;; POP NEW PC AND PS

```



```

2107 011172          71$:
2108 011172 012714 177773      MOV    #-5,  @CDC      ;SET UP COLUMN COUNT TO READ 1 COLUMN
2109 011176 012715 160000      MOV    #160000,@CDA    ;SET UP BUS ADDRESS TO NON-EXISTANT MEMORY
2110 011202 012713 000161      MOV    #161,  @CDS    ;SET INTERRUPT ENABLE AND READ, X MEM BITS SET
2111 011206 000777          BR      .              ;WAIT FOR INTERRUPT
2112 011210 022626      TINT22: CMP    (SP)+, (SP)+ ;RESTORE STACK
2113 011212 005037 000232      CLR    @232,      ;CHANGE INTERRUPT RETURN ADDRESS TO
2114 011216 012712 000232      MOV    #232,  @ADINT  ;CAUSE A HALT IF AN INTERRUPT OCCURS
2115 011222 105713      TSTB   @CDS      ;CHECK FOR CONTROLLER READY
2116 011224 100401      BMI    15        ;BRANCH IF SET OK
2117 011226 104023      ERROR +23        ;CONTROLLER READY DIDN'T SET
2118
2119 011230 005713      1$:   TST    @CDS      ;CHECK FOR ERROR (BIT 15)
2120 011232 100401      BMI    25        ;BRANCH IF SET OK
2121 011234 104026      ERROR +26        ;ERROR BIT 15 NOT SET
2122
2123 011236 032713 001000      2$:   BIT    #1000, @CDS  ;CHECK FOR NON-EXISTANT MEMORY (BIT 9)
2124 011242 001001      BNE    35        ;BRANCH IF SET OK
2125 011244 104027      ERROR +27        ;BIT 9 NOT SET
2126
2127 011246 032713 000040      3$:   BIT    #40,   @CDS  ;CHECK FOR EXTENDED MEMORY BIT 17 SET
2128 011252 001001      BNE    45        ;BRANCH IF SET OK
2129 011254 104030      ERROR +30        ;EX-MEM BIT 17 GOT CLEARED
2130
2131 011256 032713 000020      4$:   BIT    #20,   @CDS  ;CHECK FOR EX-MEM (BIT 4)
2132 011262 001001      BNE    55        ;BRANCH IF SET OK
2133 011264 104031      ERROR +31        ;EX-MEM (BIT 4) GOT CLEARED
2134
2135 011266 032713 076417      5$:   BIT    #076417,@CDS ;CHECK FOR ANY OTHER BITS
2136 011272 001401      BEQ    65        ;BRANCH IF OK
2137 011274 104032      ERROR +32        ;EXTRA ERROR BITS SET
2138
2139 011276 022715 160002      6$:   CMP    #160002,@CDA  ;CHECK ADDRESS BUFFER
2140 011302 001404      BEQ    75        ;BRANCH IF OK
2141 011304 012737 160002 001210  MOV    #160002,$TMP4 ;CORRECT 'CDA' CONTENTS FOR ERROR REPORT
2142 011312 104033      ERROR +33        ;BUS ADDRESS REG CHANGED
2143
2144 011314 022714 177774      7$:   CMP    #-4,   @CDC    ;CHECK COLUMN COUNT REG
2145 011320 001404      BEQ    10$,      ;BRANCH IF OK
2146 011322 012737 177774 001210  MOV    #-4,$TMP4  ;CORRECT 'CDC' CONTENTS FOR ERROR REPORT
2147 011330 104034      ERROR +34        ;COLUMN COUNT REG CHANGED
2148 011332
2149
2150
2151
2152
2153 011332 000004      ;*****
2154 011334 005014      ;*TEST 23 EXECUTE DATI,DATOB(LOW BYTE) LOAD ON COLUMN COUNT
2155
2156 011336 012700 125252      ;*****
2157 011342 110014      †ST23: SCOPE
2158
2159 011344 020014      CLR    @CDC      ;CLEAR COLUMN COUNT REGISTER
2160
                ;PRIOR TO BYTE LOADING
                MOV    #125252,RO ;SET VALUE FOR BYTE LOAD FROM RO
                MOVB   RO,@CDC    ;ATTEMPT TO LOAD LOWER BYTE
                ;OF COLUMN COUNT REGISTER
                CMP    RO,@CDC    ;DID HIGH BYTE GET LOADED AS
                ;WELL ON A LOW BYTE LOADING?

```

```

2161 011346 001403          BEQ      1$          ; BRANCH IF YES
2162 011350 010037 001210  MOV     RO,$TMP4    ; STORE GOOD DATA - SHOULD BE
2163 011354 104034          ERROR +34         ; HIGH BYTE NOT LOADED ON A LOW
2164                                     ; BYTE LOAD OF COLUMN COUNT REGISTER
2165 011356          IS:
2166
2167 ;*****
2168 ;*TEST 24 EXECUTE DATI,DATOB(HIGH BYTE) LOAD ON COLUMN COUNT
2169 ;*****
2170 011356 000004 †ST24: SCOPE
2171 011360 005014          CLR     @CDC        ; CLEAR COLUMN COUNT REGISTER
2172                                     ; PRIOR TO BYTE LOADING
2173 011362 012700 125252  MOV     #125252,RO  ; SET VALUE FOR BYTE LOAD FROM RO
2174 011366 110064 000001  MOVB   RO,+1(R4)   ; ATTEMPT TO LOAD HIGH BYTE
2175                                     ; OF COLUMN COUNT REGISTER
2176 011372 005714          TST     @CDC        ; DID LOW BYTE GET LOADED AS
2177                                     ; WELL ON HIGH BYTE LOADING?
2178 011374 001403          BEQ     1$          ; BRANCH IF NO
2179 011376 005037 001210  CLR     $TMP4      ; STORE GOOD DATA - SHOULD BE
2180 011402 104034          ERROR +34         ; LOW BYTE LOADED ON A HIGH
2181                                     ; BYTE LOAD OF COLUMN COUNT REGISTER
2182 011404          IS:
2183
2184 ;*****
2185 ;*TEST 25 EXECUTE DATI,DATIP ON COLUMN COUNT REGISTER
2186 ;*****
2187 011404 000004 †ST25: SCOPE
2188 011406 005014          CLR     @CDC        ; CLEAR COLUMN COUNT REGISTER
2189                                     ; PRIOR TO 'DATIP' PROCESS
2190 011410 012714 100000  MOV     #100000,@CDC ; SET A KNOWN VALUE INTO COLUMN
2191                                     ; COUNT REGISTER
2192 011414 005414          NEG     @CDC        ; PERFORM DATIP, ON COLUMN COUNT
2193                                     ; REGISTER
2194 011416 022714 100000  CMP     #100000,@CDC ; IS CONTENTS = 100000?
2195 011422 001404          BEQ     1$          ; BRANCH IF YES
2196 011424 012737 100000 001210 MOV     #100000,$TMP4 ; STORE GOOD DATA - SHOULD BE
2197 011432 104034          ERROR +34         ; CONTENTS OF COLUMN COUNT REGISTER
2198                                     ; INCORRECT
2199 011434          IS:
2200
2201 ;*****
2202 ;*TEST 26 EXECUTE DATI,DATOB(LOW BYTE) LOAD ON BUS ADDRESS
2203 ;*****
2204 011434 000004 †ST26: SCOPE
2205 011436 005015          CLR     @CDA        ; CLEAR BUS ADDRESS REGISTER
2206                                     ; PRIOR TO BYTE LOADING
2207 011440 012700 125252  MOV     #125252,RO  ; SET VALUE FOR BYTE LOAD FROM RO
2208 011444 110015          MOVB   RO,@CDA     ; ATTEMPT TO LOAD LOWER BYTE
2209                                     ; OF BUS ADDRESS REGISTER
2210 011446 020015          CMP     RO,@CDA    ; DID HIGH BYTE GET LOADED AS
2211                                     ; WELL ON A LOW BYTE LOADING?
2212 011450 001403          BEQ     1$          ; BRANCH IF YES
2213 011452 010037 001210  MOV     RO,$TMP4    ; STORE GOOD DATA - SHOULD BE
2214 011456 104033          ERROR +33         ; HIGH BYTE NOT LOADED ON A LOW

```

```

2215                                     ;BYTE LOAD OF BUS ADDRESS REGISTER
2216 011460                               1$:
2217
2218 ;*****
2219 ;*TEST 27 EXECUTE DATI,DATOB(HIGH BYTE) LOAD ON BUS ADDRESS
2220 ;*****
2221 011460 000004 1ST27: SCOPE
2222 011462 005015 CLR QCD A ;CLEAR BUS ADDRESS REGISTER
2223 ;PRIOR TO BYTE LOADING
2224 011464 012700 125252 MOV #125252,R0 ;SET VALUE FOR BYTE LOAD FROM R0
2225 011470 110065 000001 MOVB R0,+1(R5) ;ATTEMPT TO LOAD HIGH BYTE
2226 ;OF BUS ADDRESS REGISTER
2227 011474 005715 TST QCD A ;DID LOW BYTE GET LOADED AS
2228 ;WELL ON HIGH BYTE LOADING?
2229 011476 001403 BEQ 1$ ;BRANCH IF NO
2230 011500 005037 001210 CLR $TMP4 ;STORE GOOD DATA - SHOULD BE
2231 011504 104033 ERROR +33 ;LOW BYTE LOADED ON A HIGH
2232 ;BYTE LOAD OF BUS ADDRESS REGISTER
2233 011506                               1$:
2234
2235 ;*****
2236 ;*TEST 30 EXECUTE DATI,DATIP ON BUS ADDRESS REGISTER
2237 ;*****
2238 011506 000004 1ST30: SCOPE
2239 011510 005015 CLR QCD A ;CLEAR BUS ADDRESS REGISTER
2240 ;PRIOR TO 'DATIP' PROCESS
2241 011512 012715 100000 MOV #100000,QCD A ;SET A KNOWN VALUE INTO BUS
2242 ;ADDRESS REGISTER
2243 011516 005415 NEG QCD A ;PERFORM DATIP, ON BUS ADDRESS
2244 ;REGISTER
2245 011520 022715 100000 CMP #100000,QCD A ;IS CONTENTS = 100000?
2246 011524 001404 BEQ 1$ ;BRANCH IF YES
2247 011526 012737 100000 001210 MOV #100000,$TMP4 ;STORE GOOD DATA - SHOULD BE
2248 011534 104033 ERROR +33 ;CONTENTS OF BUS ADDRESS REGISTER
2249 ;INCORRECT
2250 011536                               1$:
2251
2252 ;*****
2253 ;*TEST 31 WORD COUNT OVERFLOW TO 2ND CARD
2254 ;*****
2255 011536 000004 1ST31: SCOPE
2256 011540 004737 025442 JSR PC INIT ;INITIALIZE
2257 011544 012712 012112 MOV #TINT31,QADINT ;SET UP A RETURN ADDRESS
2258 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
2259 011550 005046 CLR -(SP) ;
2260 011552 013746 000034 MOV 34,-(SP) ;SAVE CURRENT TRAP VECTOR
2261 011556 012737 011566 000034 MOV #64$,$34 ;SETUP NEW TRAP VECTOT
2262 011564 104400 TRAP ;PUSH OLD PSW AN PCON STACK
2263 011566 016666 000002 000006 64$: MOV 2(SP),6(SP) ;
2264 011574 012716 011602 MOV #65$,(SP) ;REPLACE OLD PC WITH NEW
2265 011600 000002 RTI ;RESTORE PSW
2266 011602 012637 000034 65$: MOV (SP)+,$34 ;RESTORE OLD TRAP VECTOR
2267 011606 012637 001214 MOV (SP)+,$TMP6
2268 011612 052737 000340 001214 BIS #340,$TMP6

```

```

2269 011620 013746 001214      MOV      $TMP6,-(SP)      ;; PUT NEW PS ON STACK
2270 011624 012746 011632      MOV      #66$,-(SP)      ;; PUT NEW PC ON STACK
2271 011630 000002                      RTI                      ;; POP NEW PC AND PS
2272 011632                      66$:
2273                      ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
2274 011632 005046      CLR      -(SP)          ;;
2275 011634 013746 000034      MOV      34,-(SP)      ;; SAVE CURRENT TRAP VECTOR
2276 011640 012737 011650 000034      MOV      #67$,34      ;; SETUP NEW TRAP VECTOR
2277 011646 104400      TRAP                      ;; PUSH OLD PSW AN PC ON STACK
2278 011650 016666 000002 000006 67$:      MOV      2(SP),6(SP)      ;;
2279 011656 012716 011664      MOV      #68$, (SP)      ;; REPLACE OLD PC WITH NEW
2280 011662 000002                      RTI                      ;; RESTORE PSW
2281 011664 012637 000034 68$:      MOV      (SP)+,34      ;; RESTORE OLD TRAP VECTOR
2282 011670 012662 000002      MOV      (SP)+,2(ADINT)
2283                      ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
2284 011674 005046      CLR      -(SP)          ;;
2285 011676 013746 000034      MOV      34,-(SP)      ;; SAVE CURRENT TRAP VECTOR
2286 011702 012737 011712 000034      MOV      #69$,34      ;; SETUP NEW TRAP VECTOR
2287 011710 104400      TRAP                      ;; PUSH OLD PSW AN PC ON STACK
2288 011712 016666 000002 000006 69$:      MOV      2(SP),6(SP)      ;;
2289 011720 012716 011726      MOV      #70$, (SP)      ;; REPLACE OLD PC WITH NEW
2290 011724 000002                      RTI                      ;; RESTORE PSW
2291 011726 012637 000034 70$:      MOV      (SP)+,34      ;; RESTORE OLD TRAP VECTOR
2292 011732 012637 001214      MOV      (SP)+,$TMP6
2293 011736 042737 000340 001214      BIC      #340,$TMP6
2294 011744 013746 001214      MOV      $TMP6,-(SP)      ;; PUT NEW PS ON STACK
2295 011750 012746 011756      MOV      #71$,-(SP)      ;; PUT NEW PC ON STACK
2296 011754 000002                      RTI                      ;; POP NEW PC AND PS
2297 011756                      71$:
2298 011756 005046      CLR      -(SP)          ;;
2299 011760 013746 000034      MOV      34,-(SP)      ;; SAVE CURRENT TRAP VECTOR
2300 011764 012737 011774 000034      MOV      #72$,34      ;; SETUP NEW TRAP VECTOR
2301 011772 104400      TRAP                      ;; PUSH OLD PSW AN PC ON STACK
2302 011774 016666 000002 000006 72$:      MOV      2(SP),6(SP)      ;;
2303 012002 012716 012010      MOV      #73$, (SP)      ;; REPLACE OLD PC WITH NEW
2304 012006 000002                      RTI                      ;; RESTORE PSW
2305 012010 012637 000034 73$:      MOV      (SP)+,34      ;; RESTORE OLD TRAP VECTOR
2306 012014 014637 001214      MOV      -(SP),$TMP6
2307 012020 052737 000000 001214      BIC      #000,$TMP6
2308 012026 013746 001214      MOV      $TMP6,-(SP)      ;; PUT NEW PS ON STACK
2309 012032 012746 012040      MOV      #74$,-(SP)      ;; PUT NEW PC ON STACK
2310 012036 000002                      RTI                      ;; POP NEW PC AND PS
2311 012040                      74$:
2312 012040 012714 177657      MOV      #-81.,@CDC      ;; SET COLUMN COUNT TO READ 81. COLUMNS
2313                      ;; I.E. - WRAP AROUND INTO 2ND CARD
2314                      ;; AUTOMATICALLY
2315 012044 012715 043766      MOV      #BUFBEQ,@CDA      ;; SET UP STARTING BUFFER ADDRESS
2316                      ;; FOR DUMP OF CARD/S CONTENTS
2317 012050 012713 000101      MOV      #101,@CDS      ;; SET INTERRUPT ENABLE & READ
2318 012054 105713 1$:      TSTB    @CDS      ;; WE COME HERE AWAITING THE INTERRUPT
2319 012056 100376      BPL     1$      ;; AWAIT INTERRUPT
2320                      ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
2321 012060 016246 000002      MOV      2(ADINT),-(SP)      ;; PUT NEW PS ON STACK
2322 012064 012746 012072      MOV      #75$,-(SP)      ;; PUT NEW PC ON STACK

```

```

2323 012070 000002          RTI          ;; POP NEW PC AND PS
2324 012072          75$:
2325
2326
2327 012072 005013          CLR      @CDS          ; AND GIVE CONTROL BACK TO THE
2328 012074 012712 000232  MOV      @232,@ADINT   ; PROCESSOR
2329 012100 005037 000232  CLR      @#232        ; DISABLE INTERRUPTS
2330
2331 012104 104041          ERROR +41 ; RESTORE INTERRUPT RETURN ADDRESS
2332
2333 012106 000137 012152  JMP      T31END        ; TO 'HALT' IF AN UNEXPECTED INTERRUPT
2334 012112 105713          TINT31: TSTB     @CDS        ; OCCURS
2335 012114 100401          BMI      1$           ; NO INTERRUPT WITH PROCESSOR AT
2336 012116 104023          ERROR +23            ; LEVEL 0
2337 012120 005013          1$: CLR      @CDS        ; GO TO NEXT TEST
2338 012122 012712 000232  MOV      @232,@ADINT   ; IS CONTROLLER READY SET?
2339 012126 005037 000232  CLR      @#232        ; BRANCH IF YES
2340
2341 012132 022626          CMP      (SP)+,(SP)+  ; INDICATE CONTROLLER READY NOT SET
2342 012134 022715 044230  CMP      @#BUFBEG+242,@CDA ; DISABLE FURTHER INTERRUPTS
2343 012140 001404          BEQ      T31END        ; RESTORE INTERRUPT RETURN ADDRESS
2344 012142 012737 044230 001210 MOV      @#BUFBEG+242,$TMP4 ; TO 'HALT' IF AN UNEXPECTED INTERRUPT
2345 012150 104033          ERROR +33            ; OCCURS
2346
2347 012152          T31END:                ; RESET STACK FROM THE INTERRUPT
2348
2349
2350
2351
2352 012152 000004          ;*****
2353 012154 004737 025442  ;*TEST 32      BUS ADDRESS ODD & TRANSFER IN NON-PACK MODE
2354 012160 012714 177777  ;*****
2355 012164 012715 043767  ;*****
2356
2357 012170 012737 012214 000010  ;*****
2358 012176 012737 000340 000012  ;*****
2359 012204 005213          ;*****
2360 012206 105713          ;*****
2361 012210 100376          ;*****
2362 012212 000402          ;*****
2363 012214 104070          ;*****
2364
2365 012216 022626          ;*****
2366
2367 012220 012737 000012 000010  ;*****
2368 012226 005037 000012          ;*****
2369
2370
2371 012232 000004          ;*****
2372 012234 032777 000200 166674  ;*****
2373 012242 001406          ;*****
2374 012244 104400 001222          ;*****
2375 012250 005137 001254          ;*****
2376 012254 000137 002504          ;*****

          JSR      PC INIT          ; INITIALIZE
          MOV      #-1,@CDC          ; SET COLUMN COUNT TO READ 1 COLUMN
          MOV      @#BUFBEG+1,@CDA   ; SET BUFFER ADDRESS, FOR COLUMN DUMP,
          ; TO HIGH BYTE OF WORD
          MOV      @25,RESVEC        ; SET RETURN FOR ILLEGAL INSTRUCTION
          MOV      @340,RESVEC+2     ; SET PS FOR ILLEGAL INSTRUCTION
          INC      @CDS              ; READ A CARD
          1$: TSTB     @CDS          ; WAIT FOR CONTROLLER READY
          BPL      1$                ; WAIT TILL DONE!
          BR      3$                ; GO TO NEXT TEST
          2$: ERROR +70            ; ODD BUS ADDRESS CAUSED A TRAP IN
          ; NON-PACK MODE
          CMP      (SP)+,(SP)+        ; RESET STACK FROM ILLEGAL INSTRUCTION
          ; TRAP
          3$: MOV      @12,10        ; RESTORE TRAPCATCHER AREA FOR
          CLR      @#12              ; UNEXPECTED TRAPS TO LOCATION 10
          ; CHECK SW7 AND RETURN TO TEST1 IF SET, AFTER RINGING BELL
          ; OTHERWISE GO INTO THE DATA TEST
          ENDCK: SCOPE
          BIT      @200,@SWR          ; IS SW<07> SET?
          BEQ      DAT$              ; BRANCH IF NOT TO DATA TESTING
          TYPE,   $BELL              ; RING-A-DING
          COM     TRFLG              ; TOGGLE TRACE FLAG
          JMP     RESTR              ; GO BACK TO INSTRUCTION TESTS

```



```

2431                                     : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
2432 012436 005046 CLR -(SP) ;;
2433 012440 013746 000034 MOV 34, -(SP) ;; SAVE CURRENT TRAP VECTOR
2434 012444 012737 012454 000034 MOV #64$, 34 ;; SETUP NEW TRAP VECTOR
2435 012452 104400 TRAP ;; PUSH OLD PSW AN PCON STACK
2436 012454 016666 000002 000006 64$: MOV 2(SP), 6(SP) ;;
2437 012462 012716 012470 MOV #65$, (SP) ;;
2438 012466 000002 ;; REPLACE OLD PC WITH NEW
2439 012470 012637 000034 65$: MOV (SP)+, 34 ;; RESTORE PSW
2440 012474 012637 001214 MOV (SP)+, $TMP6 ;; RESTORE OLD TRAP VECTOR
2441 012500 042737 000340 001214 BIC #340, $TMP6
2442 012506 013746 001214 MOV $TMP6, -(SP) ;; PUT NEW PS ON STACK
2443 012512 012746 012520 MOV #66$, -(SP) ;; PUT NEW PC ON STACK
2444 012516 000002 RTI ;; POP NEW PC AND PS
2445 012520 66$:
2446                                     : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
2447 012520 005046 CLR -(SP) ;;
2448 012522 013746 000034 MOV 34, -(SP) ;; SAVE CURRENT TRAP VECTOR
2449 012526 012737 012536 000034 MOV #67$, 34 ;; SETUP NEW TRAP VECTOR
2450 012534 104400 TRAP ;; PUSH OLD PSW AN PCON STACK
2451 012536 016666 000002 000006 67$: MOV 2(SP), 6(SP) ;;
2452 012544 012716 012552 MOV #68$, (SP) ;;
2453 012550 000002 ;; REPLACE OLD PC WITH NEW
2454 012552 012637 000034 68$: MOV (SP)+, 34 ;; RESTORE PSW
2455 012556 012662 000002 MOV (SP)+, 2(ADINT) ;; RESTORE OLD TRAP VECTOR
2456 012562 013701 015272 MOV TSTART, R1 ;; SET UP TABLE POINTER
2457 012566 012700 043766 MOV #BUFBEQ, R0 ;; SET UP BUFFER POINTER
2458 012572 012737 177660 015260 MOV #-120, SIZE ;; SET UP "SIZE"
2459 012600 012737 177660 015262 MOV #-120, OFFSET
2460 012606 013714 015260 MOV SIZE, @CDC ;; SET UP COLUMN COUNT
2461 012612 010015 MOV R0, @CDA ;; SET UP ADDRESS REG
2462 012614 012713 000100 MOV #100, @CDS ;; ENABLE INTERRUPTS
2463 012620 032777 000010 166310 BIT #10, @SWR ;; CHECK FOR PACK MODE ONLY
2464 012626 001406 BEQ CDREAD ;; BRANCH IF NOT SET
2465 012630 032777 000004 166300 BIT #4, @SWR ;; CHECK FOR IMAGE MODE ONLY
2466 012636 001002 BNE CDREAD ;; BRANCH IF SET
2467 012640 004737 014466 JSR PC, PAKSET ;; SET UP FOR PACKING MODE
2468 012644 005213 CDREAD: INC @CDS ;; READ
2469 012646 032713 004000 BKGND: BIT #4000, @CDS ;; CHECK FOR DATA ERROR
2470 012652 001775 BEQ BKGND
2471 012654 011437 015304 MOV @CDC, DERCNT ;; SAVE THE COLUMN COUNT
2472 012660 032713 004000 BKGND1: BIT #4000, @CDS ;; CHECK FOR DATA ERROR
2473 012664 001375 BNE BKGND1 ;; BRANCH IF SET
2474 012666 005037 015304 CLR DERCNT ;; CLR COLUMN COUNT SAVER
2475 012672 000765 BR BKGND
2476
2477                                     ; IN INTERRUPT SERVICE ROUTINE WHICH RUNS DATA RELIABILITY TEST
2478 012674 105713 SRVC: TSTB @CDS ;; CHECK CONTROLLER READY
2479 012676 100401 BMI 1$ ;; BRANCH IF SET
2480 012700 104023 ERROR +23 ;; CONTROLLER READY NOT SET
2481 012702 032713 000002 1$: BIT #2, @CDS ;; CHECK FOR DATA PACK MODE
2482 012706 001402 BEQ ISR ;; BRANCH IF IMAGE MODE
2483 012710 000137 013604 JMP PSR ;; JUMP TO PACK MODE ROUTINE
2484

```



```

2603 013430 014046      MOV      -(RO),-(SP)      ;PUSH WAS DATA
2604 013432 104402      TYPOS                      ;ONTO STACK
2605 013434      006      .BYTE      6              ;FOR
2606 013435      001      .BYTE      1              ;ERROR TYPEOUT
2607 013436 022021      CMP      (RO)+, (R1)+    ;RESET POINTERS
2608 013440 005777 165472  ISRDE4: TST      @SWR      ;CHECK FOR HALT ON ERROR
2609 013444 100001      BPL      1$              ;BRANCH IF HALT ON ERROR NOT SET
2610 013446 000000      HALT                      ;HALT ON ERROR SET
2611 013450 000137 013042  1$:      JMP      ISRRT
2612
2613 ;INTERRUPT DUE TO SOME KIND OF ERROR
2614 ;THESE ERRORS ARE DISASTEROUS, THEREFORE THE DATA TEST IS RESTARTED
2615 ISRER:  BMI      ISRE1      ;BRANCH ON ERROR BIT 15
2616          ERROR +26        ;ERROR BIT 15 NOT SET
2617          BR      ISRST
2618
2619 ISRER1: BIT      #10000, @CDS ;CHECK FOR OFF-LINE
2620          BEQ      ISRE2
2621          BIT      #40000, @CDS ;CHECK FOR CARD READER ERROR
2622          BNE      1$        ;BRANCH IF SET
2623          ERROR +35        ;OFF-LINE BUT NOT CARD READER ERROR
2624          BR      ISRE3
2625
2626 1$:      JSR      PC,      LASTCD ;CHECK FOR LAST CARD
2627          BGE      2$        ;BRANCH IF BOTH CARD
2628          ERROR +57        ;CARD READER ERROR BUT NOT BOTH CARD
2629          BR      ISRST
2630 2$:      JMP      ISRNC      ;IF BOTH CARD - GO HERE!
2631
2632 ISRER2: BIT      #40000, @CDS ;CHECK FOR CARD READER ERROR
2633          BEQ      ISRE3
2634          ERROR +60        ;CARD READER ERROR BUT NOT OFF LINE
2635
2636 ISRER3: BIT      #20000, @CDS
2637          BEQ      1$
2638          ERROR +61        ;END OF FILE ERROR (M1200 ONLY)
2639
2640 1$:      BIT      #4000, @CDS
2641          BEQ      2$
2642          ERROR +62        ;DATA ERROR
2643
2644 2$:      BIT      #2000, @CDS
2645          BEQ      3$
2646          ERROR +63        ;DATA LATE ERROR
2647
2648 3$:      BIT      #1000, @CDS
2649          BEQ      4$
2650          ERROR +64
2651
2652 4$:      BIT      #077000, @CDS ;NON-EXISTANT MEMORY ERROR
2653          BNE      ISRST    ;CHECK ALL ERROR BITS
2654          ERROR +37        ;BRANCH IF AT LEAST ONE
2655          ;NONE OF THE ERROR BITS SET
2656          JMP      DATRST   ;RESTART THE ENTIRE DATA TEST
2657
2658 PSR:    BIT      #177475, @CDS ;CHECK ALL BITS EXCEPT 1,6 AND 7

```



```

2755 014306 004737 014614 1$: JSR PC, LASTCD ;CHECK FOR LAST CARD
2756 014312 002402 BLT 2$ ;BRANCH IF NOT
2757 014314 000137 C13660 JMP PSRNC ;BRANCH IF BOTH CARD
2758 014320 104057 2$: ERROR +57 ;CARD READER ERROR BUT NOT BOTH CARD
2759 014322 000137 015252 JMP DATRST ;RESTART THE ENTIRE TEST
2760
2761 014326 032713 040000 PSRE3: BIT #40000, QCD5 ;CHECK FOR CARD READER ERROR
2762 014332 001401 BEQ PSRE4 ;BRANCH IF NOT
2763 014334 104060 ERROR +60 ;CARD READER ERROR BUT NOT OFF LINE
2764
2765 014336 032713 020000 PSRE4: BIT #20000, QCD5
2766 014342 001401 BEQ 1$
2767 014344 104061 ERROR +61 ;END OF FILE ERROR (M1200 ONLY)
2768
2769 014346 032713 002000 1$: BIT #2000, QCD5
2770 014352 001401 BEQ 2$
2771 014354 104063 ERROR +63 ;DATA LATE ERROR
2772
2773 014356 032713 001000 2$: BIT #1000, QCD5
2774 014362 001401 BEQ 3$
2775 014364 104064 ERROR +64 ;NON-EXISTANT MEMORY ERROR
2776 014366 032713 077000 3$: BIT #077000, QCD5 ;CHECK ALL ERROR BITS
2777 014372 001001 BNE PSRST ;BRANCH IF AT LEAST ONE
2778 014374 104037 ERROR +37 ;NONE OF THE ERROR BITS SET
2779 014376 000137 013666 PSRST: JMP PSRLP ;GO CHECK THE DATA
2780
2781 ;RETURN PORTION OF INTERRUPT SERVICE ROUTINE
2782 ;CALCULATES A NEW "SIZE" (NUMBER OF COLUMNS TO BE READ)
2783 ;SETS UP THE CARD READER BUFFERS, AND ISSUES THE READ COMMAND
2784 ;THEN DOES AN RTI TO THE BACKGROUND ROUTINE
2785 014402 063737 015262 015260 SRETRN: ADD OFFSET, SIZE
2786 014410 100404 BMI SRETR1
2787 014412 012737 177660 015262 MOV #120, OFFSET
2788 014420 000770 BR SRETRN
2789 014422 032737 001000 015260 SRETR1: BIT #001000, SIZE
2790 014430 001004 BNE SRETR4
2791 014432 012737 000120 015262 SRETR3: MOV #120, OFFSET
2792 014440 000760 BR SRETRN
2793 014442 004737 014614 SRETR4: JSR PC, LASTCD ;CHECK FOR MORE THAN 80 CARDS
2794 014446 003371 BGT SRETR3 ;BRANCH IF GREATER
2795 014450 013714 015260 MOV SIZE, QCD5 ;SET UP COLUMN COUNT
2796 014454 012700 043766 MOV #BUFBEQ, RO ;RESET TABLE POINTER
2797 014460 010015 MOV RO, QCD4 ;SET UP ADDRESS REG
2798 014462 005213 INC QCD5 ;READ
2799 014464 000002 RTI
2800
2801
2802 ;SUBROUTINE TO SET PACKING MODE AND MOVE THE POINTERS FOR THE DATA.
2803 014466 062737 000240 015272 PAKSET: ADD #160., TSTART ;MOVE TABLE POINTER TO PACKED TABLE
2804 014474 062737 000120 015274 ADD #80., TEND
2805 014502 062701 000240 ADD #160., R1 ;UPDATE TABLE POINTER
2806 014506 163701 015276 SUB CDCNT, R1 ;COMPENSATE FOR BYTES
2807 014512 052713 000002 BIS #2, QCD5 ;SET PACKING MODE BIT
2808 014516 000207 RTS PC

```

```

2809
2810      ;SUBROUTINE TO TYPE HEADING, TYPE OF DECK, CARD COUNT, AND COLUMN COUNT
2811 014520 005737 001260 TYHEAD: TST ERFLG ;CHECK FOR FIRST ERROR
2812 014524 001004 BNE NOHEAD ;BRANCH IF NOT
2813 014526 005237 001260 INC ERFLG ;SET FLAG
2814 014532 104400 033003 TYPE, MSG13 ;TYPE HEADING FOR DATA ERRORS
2815 014536 104400 NOHEAD: TYPE ;OUTPUT TYPE OF DECK
2816 014540 000000 DECK: DUMMY ;POINTER TO DECK TITLE
2817 014542 104400 030243 TYPE, SPACE
2818 014546 005237 015276 INC CDCNT ;ADJUST CADR COUNT
2819 014552 013746 015276 MOV CDCNT,-(SP)
2820 014556 104404 TYPDS
2821 014560 005337 015276 DEC CDCNT ;READJUST CADR COUNT
2822 014564 104400 030236 TYPE, SPACE-5
2823 014570 005237 015300 INC CLCNT ;ADJUST COLUMN COUNT
2824 014574 013746 015300 MOV CLCNT,-(SP)
2825 014600 104404 TYPDS
2826 014602 005337 015300 DEC CLCNT ;READJUST COLUMN COUNT
2827 014606 104400 030241 TYPE, SPACE-2
2828 014612 000207 RTS PC
2829
2830      ;SUBROUTINE TO CHECK FOR LAST CARD
2831 014614 013737 015260 015266 LASTCD: MOV SIZE, TEMP1
2832 014622 013737 015276 015270 MOV CDCNT, TEMP2
2833 014630 005237 015270 LSTCD1: INC TEMP2
2834 014634 062737 000120 015266 ADD #120, TEMP1
2835 014642 100772 BMI LSTCD1
2836 014644 023727 015270 000120 CMP TEMP2, #80.
2837 014652 000207 RTS PC
2838
2839      ;SUBROUTINE TO KEEP TRACK OF CARDS
2840 014654 005037 015300 NXCRD: CLR CLCNT ;RESET COLUMN COUNT
2841 014660 005037 001212 CLR STMP5
2842 014664 005237 015276 INC CDCNT ;KEEP TRACK OF CARDS
2843 014670 023727 015276 000120 CMP CDCNT, #120 ;CHECK FOR BOTH CARD
2844 014676 002001 BGE SEOP
2845 014700 000207 RTS PC ;RETURN
2846
2847
2848      ;*****
2849
2850      .SBTTL END OF PASS ROUTINE
2851
2852      ;*INCREMENT THE PASS NUMBER ($PASS)
2853      ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
2854      ;*IF SW12=1 INHIBIT TRACE TRAP
2855      ;*IF THERES A MONITOR GO TO IT
2856      ;*IF THERE ISN'T JUMP TO HOOK1
2857
2858 SEOP:
2859 014702 005726 TST (SP)+ ;CORRECT STACK POINTER TO REPLACE 'RTS'
2860 014704 022626 CMP (SP)+,(SP)+ ;CORRECT STACK POINTER TO REPLACE 'RTI'
2861 014706 005037 001102 CLR $STNM ;;ZERO THE TEST NUMBER
2862 014712 005037 001220 CLR $TIMES ;;ZERO THE NUMBER OF ITERATIONS

```



```

2863 014716 005237 001100      INC      $PASS      ;; INCREMENT THE PASS NUMBER
2864 014722 042737 100000 001100      BIC      #100000,$PASS ;; DON'T ALLOW A NEG. NUMBER
2865 014730 005327          DEC      (PC)+      ;; LOOP?
2866 014732 000001          SEOPCT: .WORD      1
2867 014734 003032          BGT      $DOAGN     ;; YES
2868 014736 012737          MOV      (PC)+,2(PC)+ ;; RESTORE COUNTER
2869 014740 000001          SENDCT: .WORD      1
2870 014742 014732          SEOPCT
2871 014744 104400 015114      TYPE     $ENDMG     ;; TYPE "END PASS #"
2872 014750 013746 001100      MOV      $PASS,-(SP) ;; SAVE $PASS FOR TYPEOUT
2873 014754 104404          TYPDS   ;; GO TYPE--DECIMAL ASCII WITH SIGN
2874 014756 104400 015131      TYPE     , $ENULL   ;; TYPE A NULL CHARACTER
2875 014762          $GET42:
2876 014762 013700 000042      MOV      @#42,R0    ;; GET MONITOR ADDRESS
2877 014766 001415          BEQ     $DOAGN     ;; BRANCH IF NO MONITOR
2878 014770 000005          RESET  ;; MONITOR CLEAR WORLD
2879 014772 005046          CLR     -(SP)     ;; INSURE THE "T" BIT IS CLEAR
2880 014774 012746 015012      MOV      #SENDAD,-(SP) ;; SETUP FOR AN RTI OR RTT
2881 015000 000441          BR      $RTRN     ;; GO DO AN RTI OR RTT TO LOAD THE PSW
2882
2883
2884 015002 013700 000042      MOV      @#42,R0    ;; GET MONITOR ADDRESS
2885 015006 001405          BEQ     $DOAGN     ;; BRANCH IF NO MONITOR
2886 015010 000005          RESET  ;; CLEAR THE WORLD
2887 015012 004710          SENDAD: JSR     PC,(R0) ;; GO TO MONITOR
2888 015014 000240          NOP
2889 015016 000240          NOP
2890 015020 000240          NOP
2891 015022          $DOAGN:
2892 015022 005046          CLR     -(SP)     ;; RESERVE A STACK LOC. FOR THE PS
2893 015024 013746 000034      MOV      @#34,-(SP) ;; SETUP THE TRAP VECTOR
2894 015030 012737 015040 000034      MOV      #15,@#34  ;; TO GET THE PS
2895 015036 104400          TRAP
2896 015040 005726          15:   TST     (SP)+      ;; CLEAN OFF THE USED PC
2897 015042 012666 000002      MOV      (SP)+,2(SP) ;; SAVE OFF THE PS
2898 015046 012637 000034      MOV      (SP)+,@#34  ;; RESTORE TRAP VECTOR
2899 015052 042716 000020      BIC     #20,(SP)   ;; CLEAR THE "T" BIT
2900 015056 032777 010000 164052      BIT     #BIT12,$SWR ;; RUN WITH TRACE TRAP?
2901 015064 001005          BNE     2$        ;; BR IF NO
2902 015066 005137 015112      COM     $TBIT     ;; IS IT TIME FOR TRACE TRAP
2903 015072 100402          BMI     2$        ;; BR IF NO
2904 015074 052716 000020      BIS     #20,(SP)   ;; SET TRACE TRAP
2905 015100 012746 015106      2$:   MOV      #SLOOP,-(SP) ;; JUMP TO START OF TEST
2906 015104 000002          $RTRN: RTI      ;; RETURN--THIS IS CHANGED TO
2907
2908
2909 015106          $LOOP:
2910 015106 000137 015134      JMP     @#HOOK1    ;; RETURN
2911 015112 000000          $TBIT: 0
2912 015114 005015 047105 020104      $ENDMG: .ASCIZ <15><12>/END PASS #/
2913 015122 040520 051523 021440
2914 015130 000
2915 015131 377 377 000 $ENULL: .BYTE -1,-1,0 ;; NULL CHARACTER STRING
2916

```

```

2917 015134 032777 000040 163774 HOOK1: BIT      #40, @SWR      ;CHECK @SWR FOR HALT AT END OF DECK
2918 015142 001402          BEQ      ONLINE      ;CONTINUE IF NOT SET
2919 015144 000000          HALT                    ;END OF DECK, SWS SET
2920 015146 000427          BR       DECKCK
2921
2922 015150 032713 010000  ONLINE: BIT      #10000, @CDS ;CHECK FOR OFF-LINE
2923 015154 001424          BEQ      DECKCK      ;BRANCH IF NOT
2924 015156 005713          TST      @CDS        ;CHECK FOR ERROR (BIT 15)
2925 015160 100401          BMI     15          ;BRANCH IF SET OK
2926 015162 104026          ERROR +26          ;ERROR BIT 15 NOT SET
2927
2928 015164 032713 040000  15:   BIT      #40000, @CDS ;CHECK FOR CARD READER ERROR
2929 015170 001001          BNE     25          ;BRANCH IF SET OK
2930 015172 104035          ERROR +35          ;OFF-LINE NOT DUE TO CARD READER ERROR
2931
2932 015174 032713 023471  25:   BIT      #023471, @CDS ;CHECK FOR EXTRA BITS SET
2933 015200 001401          BEQ      35          ;BRANCH IF OK
2934 015202 104015          ERROR +15          ;EXTRA ERROR BITS SET
2935
2936 015204 012712 015214  35:   MOV      #ONINT, @ADINT ;SET UP INTERRUPT VECTOR
2937 015210 000001          45:   WAIT                    ;WAIT FOR AN INTERRUPT
2938 015212 000776          BR       45          ;WAIT ON TRACE TRAPS
2939
2940 015214 032713 000010  ONINT: BIT      #10,   @CDS ;CHECK FOR TRANSITION TO ON LINE
2941 015220 001001          BNE     15          ;BRANCH IF SET OK
2942 015222 104036          ERROR +36          ;INTERRUPT BY OTHER THAN BIT 3 SETTING
2943
2944 015224 022626          15:   CMP      (SP)+, (SP)+ ;RESTORE THE STACK
2945          ;WHEN CONTINUING FROM ONE DECK TO ANOTHER, CHECK SWS FOR TYPE
2946          ;OF TESTING TO BE PERFORMED
2947 015226 005137 001254  DECKCK: COM     TRFLG      ;TOGGLE TRACE FLAG
2948 015232 032777 000100 163676 BIT      #100, @SWR      ;CHECK SWS
2949 015240 001402          BEQ      15          ;BRANCH IF NOT SET
2950 015242 000137 002504          JMP      RESTRT      ;RERUN COMBINED INSTRUCTION AND DATA TEST
2951 015246 000137 012260          15:   JMP      DATST
2952
2953 015252 022626          DATRST: CMP     (SP)+, (SP)+ ;RESTORE THE STACK
2954 015254 000137 012260          JMP      DATST      ;RESTART DATA TEST
2955
2956 015260 177660          SIZE:  -120
2957 015262 177660          OFFSET: -120
2958 015264 000000          BUFEND: 0
2959 015266 000000          TEMP1: 0
2960 015270 000000          TEMP2: 0
2961 015272 000000          TSTART: 0
2962 015274 000000          TEND: 0
2963 015276 000000          CDCNT: 0
2964 015300 000000          CLCNT: 0
2965 015302 000000          PTOFF: 0
2966 015304 000000          DERCNT: 0
;STARTING ADDRESS OF DATA TABLE
;END ADDRESS OF DATA TABLE
;NUMBER OF CARD BEING READ
;NUMBER OF COLUMN BEING CHECKED
;OFFSET TO POINTER FOR DATA PRINTOUT
;DATA ERROR COLUMN COUNT
    
```

```

2967
2968
2969 015306
2970 015306 012706 001100
2971 015312 005026
2972 015314 022706 001126
2973 015320 001374
2974 015322 012706 001100
2975 015326 012737 026024 000020
2976 015334 012737 000340 000022
2977 015342 012737 025650 000030
2978 015350 012737 000340 000032
2979 015356 012737 027072 000034
2980 015364 012737 000340 000036
2981 015372 012737 027126 000024
2982 015400 012737 000340 000026
2983 015406 013737 014740 014732
2984 015414 005037 001220
2985 015420 012737 015104 000014
2986 015426 012737 000340 000016
2987 015434 012737 000002 015104
2988 015442 012737 015470 000010
2989 015450 005046
2990 015452 012746 015460
2991 015456 000006
2992 015460 012737 000006 015104 64$:
2993 015466 000402
2994 015470 062706 000010 65$:
2995 015474 012737 000012 000010 66$:
2996 015502 005037 015112
2997 015506 012737 015506 001106
2998 015514 013746 000004
2999 015520 013746 000006
3000 015524 012737 015540 000004
3001 015532 005777 163400
3002 015536 000407
3003 015540 012737 000176 001136 67$:
3004 015546 012737 000174 001140
3005 015554 022626
3006 015556 012637 000006 68$:
3007 015562 012637 000004
3008 015566 104400 031031
3009
3010 015572 005037 001266
3011 015576 000137 016074
3012 015602
3013 015602 012706 001100
3014 015606 005026
3015 015610 022706 001126
3016 015614 001374
3017 015616 012706 001100
3018 015622 012737 026024 000020
3019 015630 012737 000340 000022
3020 015636 012737 025650 000030

```

: SETUP FOR ERROR FUNCTION TEST
ER1200:

```

MOV #SCMTAG,R6 ;: FIRST LOCATION TO BE CLEARED
CLR (R6)+ ;: CLEAR MEMORY LOCATION
CMP #SBDDAT,R6 ;: DONE?
BNE -6 ;: LOOP BACK IF NO
MOV #STACK,SP ;: SETUP THE STACK POINTER
MOV #SCOPE, @IOTVEC ;: IOT VECTOR FOR SCOPE ROUTINE
MOV #340, @IOTVEC+2 ;: LEVEL 7
MOV #ERROR, @EMTVEC ;: EMT VECTOR FOR ERROR ROUTINE
MOV #340, @EMTVEC+2 ;: LEVEL 7
MOV #STRAP, @TRAPVEC ;: TRAP VECTOR FOR TRAP CALLS
MOV #340, @TRAPVEC+2 ;: LEVEL 7
MOV #SPWRON, @PWAVEC ;: POWER FAILURE VECTOR
MOV #340, @PWAVEC+2 ;: LEVEL 7
SENDCT, SEOPCT ;: SETUP END-OF-PROGRAM COUNTER
CLR $TIMES ;: INITIALIZE NUMBER OF ITERATIONS
MOV #SRTM, @TBITVEC ;: SET "T" BIT VECTOR TO SRTM
MOV #340, @TBITVEC+2 ;: LEVEL 7
MOV #RTI, SRTM ;: SET SRTM TO A RTI
MOV #65$, @RESVEC ;: TRY TO DO A RTT
CLR -(SP) ;: DUMMY PS
MOV #64$, -(SP) ;: AND PC
RTT ;: TRY THE RTT
MOV #RTT, SRTM ;: RTT IS LEGAL--SET SRTM TO A RTT
BR 66$
ADD #10, SP ;: RTT ILLEGAL--CLEAN OFF THE STACK
MOV #RESVEC+2, @RESVEC ;: RESTORE TRAP CATCHER
CLR $TBIT ;: CLEAR "T" BIT SWITCH
MOV #. $LPADR ;: INITIALIZE THE LOOP ADDRESS FOR SCOPE
MOV @#4, -(SP) ;: SAVE ERROR VECTOR
MOV @#6, -(SP)
MOV #67$, 4 ;: SET UP TIME OUT VECTOR
TST @SWR ;: TRY TO REFERENCE HARDWARE SWR
BR 68$ ;: BRANCH IF NO TIMEOUT TRAP OCCURS
MOV #SWREG, SWR ;: POINT TO SOFTWARE SWR
MOV #DISPREG, DISPLAY ;: POINT TO SOFTWARE DISPLAY REG
CMP (SP)+, (SP)+ ;: RESTORE STACK
MOV (SP)+, @#6 ;: RESTORE ERROR VECTOR
MOV (SP)+, @#4
TYPE, M1200E ;: INDICATE "ENTERING ERROR FUNCTION
;: TESTING OF AN M-1200"
CLR CD1000 ;: CARD READER IS M-1200
JMP ER12CD

```

ERCD11:

```

MOV #SCMTAG,R6 ;: FIRST LOCATION TO BE CLEARED
CLR (R6)+ ;: CLEAR MEMORY LOCATION
CMP #SBDDAT,R6 ;: DONE?
BNE -6 ;: LOOP BACK IF NO
MOV #STACK,SP ;: SETUP THE STACK POINTER
MOV #SCOPE, @IOTVEC ;: IOT VECTOR FOR SCOPE ROUTINE
MOV #340, @IOTVEC+2 ;: LEVEL 7
MOV #ERROR, @EMTVEC ;: EMT VECTOR FOR ERROR ROUTINE

```

```

3021 015644 012737 000340 000032      MOV      #340,2#EMTVEC+2 ;:LEVEL 7
3022 015652 012737 027072 000034      MOV      #STRAP,2#TRAPVEC ;:TRAP VECTOR FOR TRAP CALLS
3023 015660 012737 000340 000036      MOV      #340,2#TRAPVEC+2 ;:LEVEL 7
3024 015666 012737 027126 000024      MOV      #SPWRON,2#PWVEC ;:POWER FAILURE VECTOR
3025 015674 012737 000340 000026      MOV      #340,2#PWVEC+2 ;:LEVEL 7
3026 015702 013737 014740 014732      MOV      SENDCT,SEOPCT ;:SETUP END-OF-PROGRAM COUNTER
3027 015710 005037 001220 ;:INITIALIZE NUMBER OF ITERATIONS
3028 015714 012737 015104 000014      MOV      #SRTM,2#TBITVEC ;:SET "T" BIT VECTOR TO SRTM
3029 015722 012737 000340 000016      MOV      #340,2#TBITVEC+2 ;:LEVEL 7
3030 015730 012737 000002 015104      MOV      #RTI,SRTM ;:SET SRTM TO A RTI
3031 015736 012737 015764 000010      MOV      #655,2#RESVEC ;:TRY TO DO A RTT
3032 015744 005046 ;:DUMMY PS
3033 015746 012746 015754      MOV      #645,-(SP) ;:AND PC
3034 015752 000006 ;:TRY THE RTT
3035 015754 012737 000006 015104 645:  MOV      #RTT,SRTM ;:RTT IS LEGAL--SET SRTM TO A RTT
3036 015762 000402 ;:
3037 015764 062706 000010 655:  ADD      #10,SP ;:RTT ILLEGAL--CLEAN OFF THE STACK
3038 015770 012737 000012 000010 665:  MOV      #RESVEC+2,2#RESVEC ;:RESTORE TRAP CATCHER
3039 015776 005037 015112 ;:CLEAR "T" BIT SWITCH
3040 016002 012737 016002 001106      MOV      #. ,SLPADR ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
3041 016010 013746 000004 ;:SAVE ERROR VECTOR
3042 016014 013746 000006 ;:
3043 016020 012737 016034 000004      MOV      #67$,4 ;:SET UP TIME OUT VECTOR
3044 016026 005777 163104      TST     2#SWR ;:TRY TO REFERENCE HARDWARE SWR
3045 016032 000407 ;:BRANCH IF NO TIMEOUT TRAP OCCURS
3046 016034 012737 000176 001136 675:  MOV      #SWREG,SWR ;:POINT TO SOFTWARE SWR
3047 016042 012737 000174 001140      MOV      #DISPREG,DISPLAY ;:POINT TO SOFTWARE DISPLAY REG
3048 016050 022626 ;:RESTORE STACK
3049 016052 012637 000006 685:  MOV      (SP)+,2#6 ;:RESTORE ERROR VECTOR
3050 016056 012637 000004 ;:
3051 016062 104400 031077      TYPE,   M1000E ;:INDICATE "ENTERING ERROR FUNCTION
3052 ;:TESTING OF AN M1000/M200"
3053 016066 012737 177777 001266      MOV      #177777,CD1000 ;:CARD READER IS M1000/M200
3054 016074 012737 016106 027274  ER12CD:  MOV      #ERCD12,RETURN ;:SAVE RETURN POINT FOR THIS
3055 ;:SECTION FOR POWER FAILURE RETURN
3056 016102 004737 043712      JSR     PC,SETUP ;:INITIALIZE REGISTERS
3057 016106 104400 030254  ERCD12:  TYPE,   STMES ;:TYPE MAINDEC TITLE & REV. LEVEL
3058
3059 ;:*****
3060 ;:*TEST 33 TEST DATA LATE
3061 ;:*****
3062 016112 000004      †ST33:  SCOPE
3063 ;:HALT SHOULD CAUSE DATA LATE ERROR (BIT 10)
3064 ;:SHOULD SET ERROR (BIT 15)
3065 016114 004737 025442      JSR     PC,INIT ;:INITIALIZE STATUS REGISTER
3066 016120 104400 030251      TYPE,   CRLF
3067 016124 104400 033535      TYPE,   MSG22 ;:"WHEN PRINTING STOPS PUT HALT AND
3068 ;:SINGLE BUS CYCLE DOWN, AND HIT 'CONTINUE' ON THE
3069 ;:CONSOLE UNTIL ONE CARD IS READ
3070 ;:THEN PUT UP THE TWO SWITCHES AND HIT
3071 ;:'CONTINUE' ON THE CONSOLE
3072 016130 104400 030246      TYPE,   CRLF-3 ;:MOVE MESSAGE UP ON TTY
3073 016134 012714 177701      MOV      #-77,2CDC ;:SET UP COLUMN COUNT
3074 016140 012715 043766      MOV      #BUFBEQ,2CDA ;:SET UP BUS ADDRESS

```

```

3075 016144 000000          HALT
3076 016146 005213          INC          @CDS          ; START READING
3077 016150 105713          TSTB         @CDS          ; CHECK FOR CONTROLLER READY
3078 016152 001001          BNE          1$           ; BRANCH IF SET OK
3079 016154 104023          ERROR +23              ; CONTROLLER READY FAILED TO SET
3080
3081 016156 005713          1$: TST          @CDS          ; CHECK FOR ERROR ( BIT 15)
3082 016160 001001          BNE          2$           ; BRANCH IF SET OK
3083 016162 104026          ERROR +26              ; ERROR BIT 15 NOT SET
3084
3085 016164 032713 002000          2$: BIT          #2000, @CDS ; CHECK FOR DATA LATE ERROR (BIT 10)
3086 016170 001001          BNE          3$           ; BRANCH IF SET OK
3087 016172 104042          ERROR +42              ; DATA LATE BIT 10 NOT SET
3088
3089 016174 032713 075577          3$: BIT          #075577,@CDS ; CHECK FOR ANY OTHER BITS
3090 016200 001401          BEQ          4$           ; BRANCH IF OK
3091 016202 104004          ERROR +4               ; EXTRA BITS SET IN STATUS WORD
3092
3093 016204          4$:
3094          ;*****
3095          ;*TEST 34      TEST ERROR AND OFF LINE BITS
3096          ;*****
3097 016204 000004          †ST34: SCOPE
3098          ; THE CARD READER GOING OFF-LINE SHOULD SET ERROR (BIT 15)
3099          ; AND OFF-LINE (BIT 12)
3100          ; GOING BACK ON LINE SHOULD SET "TRANSITION TO ON-LINE" (BIT 3)
3101 016206 004737 025442          JSR          PC_INIT      ; INITIALIZE STATUS REGISTER
3102 016212 104400 031375          TYPE,      MSG3          ; "PRESS CARD READER 'STOP'"
3103 016216 104400 031330          TYPE,      MSG2          ; "THEN HIT 'CONTINUE' ON THE CONSOLE"
3104 016222 104400 030246          TYPE,      CRLF-3        ; MOVE MESSAGE UP ON TTY
3105 016226 000000          HALT
3106 016230 032713 010000          BIT          #10000,@CDS ; CHECK BIT 12
3107 016234 001001          BNE          1$           ; BRANCH IF SET
3108 016236 104043          ERROR +43              ; OFF-LINE (BIT 12) WASN'T SET
3109
3110 016240 005713          1$: TST          @CDS          ; CHECK BIT 15
3111 016242 100401          BMI          2$           ; BRANCH IF SET
3112 016244 104026          ERROR +26              ; ERROR (BIT 15) WASN'T SET
3113
3114 016246 031327 067577          2$: BIT          @CDS, #067577 ; CHECK FOR EXTRA BITS
3115 016252 001401          BEQ          3$           ; BRANCH IF OK
3116 016254 104015          ERROR +15              ; STATUS WORD ERROR
3117
3118 016256 104400 031274          3$: TYPE,      MSG1          ; "PRESS CARD READER 'RESET'";
3119 016262 104400 031330          TYPE,      MSG2          ; "THEN HIT 'CONTINUE' ON THE CONSOLE"
3120 016266 104400 030246          TYPE,      CRLF-3        ; MOVE MESSAGE UP ON TTY
3121 016272 000000          HALT
3122
3123 016274 032713 000010          BIT          #10, @CDS    ; CHECK FOR TRANSITION TO ON-LINE(BIT 3)
3124 016300 001001          BNE          4$           ; BRANCH IF SET OK
3125 016302 104036          ERROR +36              ; TRANSITION TO ON-LINE FAILED TO SET
3126
3127 016304 032713 010000          4$: BIT          #10000, @CDS ; CHECK FOR OFF-LINE
3128 016310 001401          BEQ          5$           ; BRANCH IF OK
    
```

```

3129 016312 104044          ERROR +44          ;OFF-LINE STILL SET
3130
3131 016314 005713          55:   TST      ACDS          ;CHECK ERROR (BIT 15)
3132 016316 100401          BMI      65          ;BRANCH IF STILL SET
3133 016320 104026          ERROR +26          ;ERROR BIT 15 CLEARED
3134
3135 016322 032713 077567    65:   BIT      #077567,ACDS ;CHECK FOR EXTRA BITS
3136 016326 001401          BEQ      75          ;BRANCH IF OK
3137 016330 104015          ERROR +15          ;EXTRA STATUS BITS SET
3138
3139 016332          75:
3140          ;*****
3141 016332 000004          †ST35: SCOPE
3142          ;TRYING TO READ WHEN CARD READER IS OFF-LINE SHOULD CAUSE AN INTERRUPT
3143          ;CHECK THAT AN INTERRUPT OCCURS WHEN THE CARD READER COMES ON LINE
3144 016334 004737 025442      JSR      PC,INIT      ;INITIALIZE STATUS REGISTER
3145 016340 012712 016576      MOV      #TINTC,ADINT ;LOAD RETURN POINTER
3146          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3147 016344 005046          CLR      -(SP)        ;
3148 016346 013746 000034      MOV      34, -(SP)    ;SAVE CURRENT TRAP VECTOR
3149 016352 012737 016362 000034  MOV      #64$,34     ;SETUP NEW TRAP VECTOT
3150 016360 104400          TRAP                    ;PUSH OLD PSW AN PCON STACK
3151 016362 016666 000002 000006 645:  MOV      2(SP),6(SP)  ;
3152 016370 012716 016376          MOV      #65$, (SP)  ;REPLACE OLD PC WITH NEW
3153 016374 000002          RTI                    ;RESTORE PSW
3154 016376 012637 000034 655:  MOV      (SP)+,34     ;RESTORE OLD TRAP VECTOR
3155 016402 012637 001214      MOV      (SP)+,$TMP6
3156 016406 052737 000340 001214  BIS      #340,$TMP6
3157 016414 013746 001214      MOV      $TMP6,-(SP)  ;PUT NEW PS ON STACK
3158 016420 012746 016426      MOV      #66$,-(SP)  ;PUT NEW PC ON STACK
3159 016424 000002          RTI                    ;POP NEW PC AND PS
3160 016426          665:
3161          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
3162 016426 005046          CLR      -(SP)        ;
3163 016430 013746 000034      MOV      34, -(SP)    ;SAVE CURRENT TRAP VECTOR
3164 016434 012737 016444 000034  MOV      #67$,34     ;SETUP NEW TRAP VECTOT
3165 016442 104400          TRAP                    ;PUSH OLD PSW AN PCON STACK
3166 016444 016666 000002 000006 675:  MOV      2(SP),6(SP)  ;
3167 016452 012716 016460          MOV      #68$, (SP)  ;REPLACE OLD PC WITH NEW
3168 016456 000002          RTI                    ;RESTORE PSW
3169 016460 012637 000034 685:  MOV      (SP)+,34     ;RESTORE OLD TRAP VECTOR
3170 016464 012662 000002      MOV      (SP)+,2(ADINT)
3171          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3172 016470 005046          CLR      -(SP)        ;
3173 016472 013746 000034      MOV      34, -(SP)    ;SAVE CURRENT TRAP VECTOR
3174 016476 012737 016506 000034  MOV      #69$,34     ;SETUP NEW TRAP VECTOT
3175 016504 104400          TRAP                    ;PUSH OLD PSW AN PCON STACK
3176 016506 016666 000002 000006 695:  MOV      2(SP),6(SP)  ;
3177 016514 012716 016522          MOV      #70$, (SP)  ;REPLACE OLD PC WITH NEW
3178 016520 000002          RTI                    ;RESTORE PSW
3179 016522 012637 000034 705:  MOV      (SP)+,34     ;RESTORE OLD TRAP VECTOR
3180 016526 012637 001214      MOV      (SP)+,$TMP6
3181 016532 042737 000340 001214  BIC      #340,$TMP6
3182 016540 013746 001214      MOV      $TMP6,-(SP) ;PUT NEW PS ON STACK
    
```

```

3183 016544 012746 016552      MOV      #715, -(SP)      ;;PUT NEW PC ON STACK
3184 016550 000002      RTI                      ;; POP NEW PC AND PS
3185 016552      715:
3186 016552 012713 000100      MOV      #100, 2(CDS)    ;SET INTERRUPT ENABLE
3187 016556 104400 031375      TYPE,   MSG3            ;"PRESS CARD READER 'STOP'"
3188 016562 104400 030246      TYPE,   CRLF-3         ;MOVE MESSAGE UP ON TTY
3189 016566 032713 010000      TLOPC: BIT      #10000, 2(CDS) ;WAIT FOR OFF-LINE TO SET
3190 016572 001775      BEQ     TLOPC
3191 016574 000402      BR      CONTC          ;SKIP INTERRUPT HANDLER
3192
3193 016576 104021      TINTC: ERROR +21      ;'STOP' SHOULDN'T CAUSE AN INTERRUPT
3194 016600 000002      RTI                      ;RETURN FROM THE INTERRUPT
3195
3196 016602 105713      CONTC: TSTB     2(CDS)    ;CHECK CONTROLLER READY BIT 7
3197 016604 100401      BMI     15             ;BRANCH IF OK
3198 016606 104023      ERROR +23            ;CU READY DIDN'T SET YET
3199
3200 016610 005713      15:   TST      2(CDS)    ;CHECK ERROR BIT
3201 016612 100401      BMI     25             ;BRANCH IF SET
3202 016614 104026      ERROR +26            ;ERROR (BIT 15) NOT SET
3203
3204 016616 032713 067477      25:   BIT      #067477, 2(CDS) ;CHECK FOR EXTRA BITS
3205 016622 001401      BEQ     35             ;BRANCH IF OK
3206 016624 104004      ERROR +4             ;STATUS WORD ERROR
3207
3208 016626 012712 017044      35:   MOV      #TINTCA, 2(ADINT) ;LOAD RETURN POINTER
3209      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340, PS"
3210 016632 005046      CLR     -(SP)          ;;
3211 016634 013746 000034      MOV     34, -(SP)      ;;SAVE CURRENT TRAP VECTOR
3212 016640 012737 016650 000034      MOV     #645, 34      ;;SETUP NEW TRAP VECTOT
3213 016646 104400      TRAP                    ;;PUSH OLD PSH AN PCON STACK
3214 016650 016666 000002 000006 645:  MOV     2(SP), 6(SP)   ;;
3215 016656 012716 016664      MOV     #655, (SP)    ;;REPLACE OLD PC WITH NEW
3216 016662 000002      RTI                      ;;RESTORE PSH
3217 016664 012637 000034      655:  MOV     (SP)+, 34      ;;RESTORE OLD TRAP VECTOR
3218 016670 012637 001214      MOV     (SP)+, $TMP6
3219 016674 052737 000340 001214      BIS     #340, $TMP6
3220 016702 013746 001214      MOV     $TMP6, -(SP)
3221 016706 012746 016714      MOV     #665, -(SP)   ;;PUT NEW PS ON STACK
3222 016712 000002      RTI                      ;;PUT NEW PC ON STACK
3223 016714      ;; POP NEW PC AND PS
3224
3225 016714 005046      665:  ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, 2(ADINT)"
3226 016716 013746 000034      CLR     -(SP)          ;;
3227 016722 012737 016732 000034      MOV     34, -(SP)      ;;SAVE CURRENT TRAP VECTOR
3228 016730 104400      MOV     #675, 34      ;;SETUP NEW TRAP VECTOT
3229 016732 016666 000002 000006 675:  TRAP                    ;;PUSH OLD PSH AN PCON STACK
3230 016740 012716 016746      MOV     2(SP), 6(SP)   ;;
3231 016744 000002      MOV     #685, (SP)    ;;REPLACE OLD PC WITH NEW
3232 016746 012637 000034      RTI                      ;;RESTORE PSH
3233 016752 012662 000002      685:  MOV     (SP)+, 34      ;;RESTORE OLD TRAP VECTOR
3234      MOV     (SP)+, 2(ADINT)
3235 016756 005046      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340, PS"
3236 016760 013746 000034      CLR     -(SP)          ;;
MOV     34, -(SP)      ;;SAVE CURRENT TRAP VECTOR

```

3237	016764	012737	016774	000034		MOV	#69\$,34	::	SETUP NEW TRAP VECTOT
3238	016772	104400				TRAP		::	PUSH OLD PSW AN PC ON STACK
3239	016774	016666	000002	J 0:06	69\$:	MOV	2(SP),6(SP)	::	
3240	017002	012716	017010			MOV	#70\$, (SP)	::	REPLACE OLD PC WITH NEW
3241	017006	000002				RTI		::	RESTORE PSW
3242	017010	012637	000034		70\$:	MOV	(SP)+,34	::	RESTORE OLD TRAP VECTOR
3243	017014	012637	001214			MOV	(SP)+,\$TMP6	::	
3244	017020	042737	000340	001214		BIC	#340,\$TMP6	::	
3245	017026	013746	001214			MOV	\$TMP6,-(SP)	::	PUT NEW PS ON STACK
3246	017032	012746	017040			MOV	#71\$,-(SP)	::	PUT NEW PC ON STACK
3247	017036	000002				RTI		::	POP NEW PC AND PS
3248	017040				71\$:			::	
3249	017040	005213				INC	ACDS	::	TRY TO READ A CARD
3250	017042	000777				BR	.	::	WAIT FOR THE INTERRUPT
3251								::	
3252	017044	022626			TINTCA:	CMP	(SP)+, (SP)+	::	RESTORE THE STACK
3253	017046	105713				TSTB	ACDS	::	CHECK CONTROLLER READY BIT 7
3254	017050	100401				BMI	1\$::	BRANCH IF OK
3255	017052	104023				ERROR	+23	::	CU READY DIDN'T SET YET
3256								::	
3257	017054	032713	010000		1\$:	BIT	#10000, ACDS	::	CHECK FOR OFF-LINE BIT 12
3258	017060	001001				BNE	2\$::	BRANCH IF OK
3259	017062	104043				ERROR	+43	::	OFF-LINE BIT 12 NOT SET
3260								::	
3261	017064	005713			2\$:	TST	ACDS	::	CHECK ERROR BIT
3262	017066	100401				BMI	3\$::	BRANCH IF SET
3263	017070	104026				ERROR	+26	::	ERROR (BIT 15) NOT SET
3264								::	
3265	017072	032713	067477		3\$:	BIT	#067477, ACDS	::	CHECK FOR EXTRA BITS
3266	017076	001401				BEQ	4\$::	BRANCH IF OK
3267	017100	104004				ERROR	+4	::	STATUS WORD ERROR
3268								::	
3269	017102	012712	017326		4\$:	MOV	#TINTCB,ADINT	::	LOAD RETURN POINTER
3270								::	THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3271	017106	005046				CLR	-(SP)	::	
3272	017110	013746	000034			MOV	34,-(SP)	::	SAVE CURRENT TRAP VECTOR
3273	017114	012737	017124	000034		MOV	#64\$,34	::	SETUP NEW TRAP VECTOT
3274	017122	104400				TRAP		::	PUSH OLD PSW AN PC ON STACK
3275	017124	016666	000002	000006	64\$:	MOV	2(SP),6(SP)	::	
3276	017132	012716	017140			MOV	#65\$, (SP)	::	REPLACE OLD PC WITH NEW
3277	017136	000002				RTI		::	RESTORE PSW
3278	017140	012637	000034		65\$:	MOV	(SP)+,34	::	RESTORE OLD TRAP VECTOR
3279	017144	012637	001214			MOV	(SP)+,\$TMP6	::	
3280	017150	052737	000340	001214		BIS	#340,\$TMP6	::	
3281	017156	013746	001214			MOV	\$TMP6,-(SP)	::	PUT NEW PS ON STACK
3282	017162	012746	017170			MOV	#66\$,-(SP)	::	PUT NEW PC ON STACK
3283	017166	000002				RTI		::	POP NEW PC AND PS
3284	017170				66\$:			::	
3285								::	THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
3286	017170	005046				CLR	-(SP)	::	
3287	017172	013746	000034			MOV	34,-(SP)	::	SAVE CURRENT TRAP VECTOR
3288	017176	012737	017206	000034		MOV	#67\$,34	::	SETUP NEW TRAP VECTOT
3289	017204	104400				TRAP		::	PUSH OLD PSW AN PC ON STACK
3290	017206	016666	000002	000006	67\$:	MOV	2(SP),6(SP)	::	


```

3291 017214 012716 017222      MOV      #68$, (SP)          ;; REPLACE OLD PC WITH NEW
3292 017220 000002                RTI                          ;; RESTORE PSH
3293 017222 012637 000034      68$:  MOV      (SP)+, 34      ;; RESTORE OLD TRAP VECTOR
3294 017226 012662 000002      MOV      (SP)+, 2(ADINT)
3295                ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3296 017232 005046                CLR      -(SP)              ;
3297 017234 013746 000034      MOV      34, -(SP)          ; SAVE CURRENT TRAP VECTOR
3298 017240 012737 017250 000034  MOV      #69$, 34          ; SETUP NEW TRAP VECTOR
3299 017246 104400                TRAP
3300 017250 016666 000002 000006 69$:  MOV      2(SP), 6(SP)      ;
3301 017256 012716 017264      MOV      #70$, (SP)        ; REPLACE OLD PC WITH NEW
3302 017262 000002                RTI                          ;; RESTORE PSH
3303 017264 012637 000034      70$:  MOV      (SP)+, 34      ;; RESTORE OLD TRAP VECTOR
3304 017270 012637 001214      MOV      (SP)+, $TMP6
3305 017274 042737 000340 001214  BIC      #340, $TMP6
3306 017302 013746 001214      MOV      $TMP6, -(SP)      ; PUT NEW PS ON STACK
3307 017306 012746 017314      MOV      #71$, -(SP)      ; PUT NEW PC ON STACK
3308 017312 000002                RTI                          ; POP NEW PC AND PS
3309 017314
3310 017314 104400 031274      71$:  TYPE,   MSG1          ; "PRESS CARD READER 'RESET'"
3311 017320 104400 030246      TYPE,   CRLF-3          ; MOVE MESSAGE UP ON TTY
3312 017324 000777                BR      .                ; WAIT FOR THE INTERRUPT
3313
3314 017326 022626                TINTCB: CMP      (SP)+, (SP)+ ; RESTORE THE STACK
3315 017330 032713 000010      BIT      #10,  %CDS      ; CHECK FOR TRANSITION TO ON-LINE (BIT 3)
3316 017334 001001                BNE      1$              ; BRANCH IF SET OK
3317 017336 104036                ERROR +36              ; TRANSITION TO ON-LINE FAILED TO SET
3318
3319 017340 032713 010000      1$:  BIT      #10000, %CDS  ; CHECK FOR OFF-LINE
3320 017344 001401                BEQ      2$              ; BRANCH IF OK
3321 017346 104044                ERROR +44              ; OFF-LINE STILL SET
3322
3323 017350 005713                2$:  TST      %CDS          ; CHECK ERROR (BIT 15)
3324 017352 100401                BMI      3$              ; BRANCH IF STILL SET
3325 017354 104026                ERROR +26              ; ERROR BIT 15 CLEARED
3326
3327 017356 032713 077467      3$:  BIT      #077467, %CDS ; CHECK FOR EXTRA BITS
3328 017362 001401                BEQ      4$              ; BRANCH IF OK
3329 017364 104015                ERROR +15              ; EXTRA STATUS BITS SET
3330
3331 017366
3332
3333                ; *****
3334                ; *TEST 36 TEST INPUT HOPPER EMPTY
3335                ; *****
3336                ; *ST36: SCOPE
3337                ; INPUT HOPPER EMPTY SHOULD SET SPECIAL CONDITION
3338                ; CHECK THAT INTERRUPTS OCCUR WHEN THE CARD READER COMES ON LINE
3339 017370 004737 025442      JSR      PC, INIT        ; INITIALIZE STATUS REGISTER
3340 017374 104400 031463      TYPE,   MSG5            ; "REMOVE ALL CARDS FROM THE INPUT HOPPER"
3341 017400 104400 031330      TYPE,   MSG2            ; "THEN HIT 'CONTINUE' ON THE CCNSOLE"
3342 017404 104400 030246      TYPE,   CRLF-3          ; MOVE MESSAGE UP ON TTY
3343 017410 000000                HALT
3344 017412 032713 010000      BIT      #10000, %CDS    ; CHECK BIT 12
3345 017416 001001                BNE      1$              ; BRANCH IF SET

```

```

3345 017420 104043          ERROR +43          ;OFF-LINE (BIT 12) WASN'T SET
3346
3347 017422 005713          15:  TST      2CDS          ;CHECK ERROR BIT
3348 017424 100401          BMI      25          ;BRANCH IF SET
3349 017426 104026          ERROR +26          ;ERROR (BIT 15) NOT SET
3350
3351 017430 032713 040000          25:  BIT      #40000, 2CDS ;CHECK FOR CARD READER ERROR
3352 017434 001001          BNE      35          ;BRANCH IF SET
3353 017436 104035          ERROR +35          ;CARD READER ERROR BIT 14 NOT SET
3354
3355 017440 032713 027573          35:  BIT      #027573, 2CDS ;CHECK FOR EXTRA BITS
3356 017444 001401          BEQ      45          ;BRANCH IF OK
3357 017446 104015          ERROR +15          ;STATUS WORD ERROR
3358
3359 017450 012712 017704          45:  MOV      #TINTD, 2ADINT ;LOAD RETURN POINTER
3360          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3361 017454 005046          CLR      -(SP)          ;
3362 017456 013746 000034          MOV      34, -(SP)      ;SAVE CURRENT TRAP VECTOR
3363 017462 012737 017472 000034          MOV      #64$, 34      ;SETUP NEW TRAP VECTOR
3364 017470 104400          TRAP          ;PUSH OLD PSW AN PCC. STACK
3365 017472 016666 000002 000006 64$: MOV      2(SP), 6(SP)  ;
3366 017500 012716 017506          MOV      #65$, (SP)    ;REPLACE OLD PC WITH NEW
3367 017504 000002          RTI          ;RESTORE PSW
3368 017506 012637 000034          55:  MOV      (SP)+, 34    ;RESTORE OLD TRAP VECTOR
3369 017512 012637 001214          MOV      (SP)+, $TMP6  ;
3370 017516 052737 000340 001214          BIS      #340, $TMP6  ;
3371 017524 013746 001214          MOV      $TMP6, -(SP)  ;PUT NEW PS ON STACK
3372 017530 012746 017536          MOV      #66$, -(SP)  ;PUT NEW PC ON STACK
3373 017534 000002          RTI          ;POP NEW PC AND PS
3374 017536
3375          66$:  ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, 2(ADINT)"
3376 017536 005046          CLR      -(SP)          ;
3377 017540 013746 000034          MOV      34, -(SP)      ;SAVE CURRENT TRAP VECTOR
3378 017544 012737 017554 000034          MOV      #67$, 34      ;SETUP NEW TRAP VECTOR
3379 017552 104400          TRAP          ;PUSH OLD PSW AN PCON STACK
3380 017554 016666 000002 000006 67$: MOV      2(SP), 6(SP)  ;
3381 017562 012716 017570          MOV      #68$, (SP)    ;REPLACE OLD PC WITH NEW
3382 017566 000002          RTI          ;RESTORE PSW
3383 017570 012637 000034          68$: MOV      (SP)+, 34    ;RESTORE OLD TRAP VECTOR
3384 017574 012662 000002          MOV      (SP)+, 2(ADINT) ;
3385          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3386 017600 005046          CLR      -(SP)          ;
3387 017602 013746 000034          MOV      34, -(SP)      ;SAVE CURRENT TRAP VECTOR
3388 017606 012737 017616 000034          MOV      #69$, 34      ;SETUP NEW TRAP VECTOR
3389 017614 104400          TRAP          ;PUSH OLD PSW AN PCON STACK
3390 017616 016666 000002 000006 69$: MOV      2(SP), 6(SP)  ;
3391 017624 012716 017632          MOV      #70$, (SP)    ;REPLACE OLD PC WITH NEW
3392 017630 000002          RTI          ;RESTORE PSW
3393 017632 012637 000034          70$: MOV      (SP)+, 34    ;RESTORE OLD TRAP VECTOR
3394 017636 012637 001214          MOV      (SP)+, $TMP6  ;
3395 017642 042737 000340 001214          BIC      #340, $TMP6  ;
3396 017650 013746 001214          MOV      $TMP6, -(SP)  ;PUT NEW PS ON STACK
3397 017654 012746 017662          MOV      #71$, -(SP)  ;PUT NEW PC ON STACK
3398 017660 000002          RTI          ;POP NEW PC AND PS

```

```

3399 017662
3400 017662 012713 000100
3401 017666 104400 031534
3402 017672 104400 031274
3403 017676 104400 030246
3404 017702 000777
3405
3406 017704 022626
3407 017706 012712 020134
3408
3409 017712 005046
3410 017714 013746 000034
3411 017720 012737 017730 000034
3412 017726 104400
3413 017730 016666 000002 000006
3414 017736 012716 017744
3415 017742 000002
3416 017744 012637 000034
3417 017750 012637 001214
3418 017754 052737 000340 001214
3419 017762 013746 001214
3420 017766 012746 017774
3421 017772 000002
3422 017774
3423
3424 017774 005046
3425 017776 013746 000034
3426 020002 012737 020012 000034
3427 020010 104400
3428 020012 016666 000002 000006
3429 020020 012716 020026
3430 020024 000002
3431 020026 012637 000034
3432 020032 012662 000002
3433
3434 020036 005046
3435 020040 013746 000034
3436 020044 012737 020054 000034
3437 020052 104400
3438 020054 016666 000002 000006
3439 020062 012716 020070
3440 020066 000002
3441 020070 012637 000034
3442 020074 012637 001214
3443 020100 042737 000340 001214
3444 020106 013746 001214
3445 020112 012746 020120
3446 020116 000002
3447 020120
3448 020120 012714 177701
3449 020124 012715 043766
3450 020130 005213
3451 020132 000777
3452

715:
MOV #100, @CDS ;SET INTERRUPT ENABLE
TYPE, MSG6 ;"RESTORE CARDS TO THE INPUT HOPPER"
TYPE, MSG1 ;"PRESS CARD READER 'RESET'"
TYPE, CRLF-3 ;MOVE MESSAGE UP ON TTY
BR . ;WAIT FOR THE INTERRUPT

TIMED:
CMP (SP)+, (SP)+ ;RESTORE THE STACK
MOV #TINTDA, @ADINT ;LOAD RETURN POINTER
;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
CLR -(SP)
MOV 34, -(SP) ;SAVE CURRENT TRAP VECTOR
MOV #64$, 34 ;SETUP NEW TRAP VECTOR
TRAP ;PUSH OLD PSW AN PCON STACK
MOV 2(SP), 6(SP)
MOV #65$, (SP) ;REPLACE OLD PC WITH NEW
RTI ;RESTORE PSW
MOV (SP)+, 34 ;RESTORE OLD TRAP VECTOR
MOV (SP)+, $TMP6
BIS #340, $TMP6
MOV $TMP6, -(SP) ;PUT NEW PS ON STACK
MOV #66$, -(SP) ;PUT NEW PC ON STACK
RTI ;POP NEW PC AND PS

64$:
;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(@ADINT)"
CLR -(SP)
MOV 34, -(SP) ;SAVE CURRENT TRAP VECTOR
MOV #67$, 34 ;SETUP NEW TRAP VECTOR
TRAP ;PUSH OLD PSW AN PCON STACK
MOV 2(SP), 6(SP)
MOV #68$, (SP) ;REPLACE OLD PC WITH NEW
RTI ;RESTORE PSW
MOV (SP)+, 34 ;RESTORE OLD TRAP VECTOR
MOV (SP)+, 2(@ADINT)
;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
CLR -(SP)
MOV 34, -(SP) ;SAVE CURRENT TRAP VECTOR
MOV #69$, 34 ;SETUP NEW TRAP VECTOR
TRAP ;PUSH OLD PSW AN PCON STACK
MOV 2(SP), 6(SP)
MOV #70$, (SP) ;REPLACE OLD PC WITH NEW
RTI ;RESTORE PSW
MOV (SP)+, 34 ;RESTORE OLD TRAP VECTOR
MOV (SP)+, $TMP6
BIC #340, $TMP6
MOV $TMP6, -(SP) ;PUT NEW PS ON STACK
MOV #71$, -(SP) ;PUT NEW PC ON STACK
RTI ;POP NEW PC AND PS

65$:
705:
MOV #-77, @CDC ;SET UP COLUMN COUNT
MOV #BUFBEG, @CDA ;SET UP BUS ADDRESS
INC @CDS ;START READING
BR . ;WAIT FOR AN INTERRUPT
    
```

```

3453 020134 022626          TINTDA: CMP      (SP)+ (SP)+ ;RESTORE THE STACK
3454 020136 022713 000300  CMP      #000300,ACDS ;CHECK THE CARD READER STATUS
3455 020142 001401          BEQ      15 ;BRANCH IF OK
3456 020144 104004          ERROR +4 ;CARD READER STATUS ERROR
3457
3458 020146          15:
3459 ;*****
3460 ;*TEST 37 TEST OUTPUT STACKER FULL
3461 ;*****
3462 020146 000004          †ST37: SCOPE
3463 ;OUTPUT STACKER FULL SHOULD SET BITS 15, 14, 12, 7
3464 020150 004737 025442      JSR      PC,INIT ;INITIALIZE STATUS REGISTER
3465 020154 104400 031600      TYPE,   MSG7 ;"PULL OUTPUT STACKER PRESSURE ARM
3466 ;ALL THE WAY DOWN"
3467 020160 104400 031330      TYPE,   MSG2 ;"THEN HIT 'CONTINUE' ON THE CONSOLE"
3468 020164 104400 030246      TYPE,   CRLF-3 ;MOVE MESSAGE UP ON TTY
3469 020170 000000          HALT
3470 020172 032713 010000      BIT      #10000,ACDS ;CHECK OFF-LINE BIT12
3471 020176 001001          BNE     15 ;BRANCH IF SET
3472 020200 104043          ERROR +43 ;OFF-LINE (BIT 12) WASN'T SET
3473
3474 020202 005713          15.  TST      ACDS ;CHECK ERROR BIT 15
3475 020204 100401          BMI     25 ;BRANCH IF SET
3476 020206 104026          ERROR +26 ;ERROR BIT 15 NOT SET
3477
3478 020210 032713 040000      25:  BIT      #40000, ACDS ;CHECK FOR CARD READER ERROR
3479 020214 001001          BNE     35 ;BRANCH IF SET
3480 020216 104035          ERROR +35 ;CARD READER ERROR BIT 14 NOT SET
3481
3482 020220 032713 027577      35:  BIT      #027577,ACDS ;CHECK FOR EXTRA BITS
3483 020224 001401          BEQ     45 ;BRANCH IF OK
3484 020226 104015          ERROR +15 ;STATUS WORD ERROR
3485
3486 020230 012712 020460      45:  MOV      #TINTE,ADINT ;LOAD RETURN POINTER
3487 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3488 CLR      -(SP) ;
3489 MOV      34, -(SP) ;SAVE CURRENT TRAP VECTOR
3490 MOV      #64$,34 ;SETUP NEW TRAP VECTOT
3491 TRAP ;PUSH OLD PSW AN PCON STACK
3492 020252 016666 000002 000006 64$: MOV      2(SP),6(SP) ;
3493 020260 012716 020266      MOV      #65$, (SP) ;
3494 020264 000002          RTI ;RESTORE PSW
3495 020266 012637 000034      65$: MOV      (SP)+,34 ;RESTORE OLD TRAP VECTOR
3496 020272 012637 001214      MOV      (SP)+,$TMP6 ;
3497 020276 052737 000340 001214      BIS      #340,$TMP6 ;
3498 020304 013746 001214      MOV      $TMP6, -(SP) ;PUT NEW PS ON STACK
3499 020310 012746 020316      MOV      #66$, -(SP) ;PUT NEW PC ON STACK
3500 020314 000002          RTI ;POP NEW PC AND PS
3501 020316          66$:
3502 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
3503 CLR      -(SP) ;
3504 MOV      34, -(SP) ;SAVE CURRENT TRAP VECTOR
3505 MOV      #67$,34 ;SETUP NEW TRAP VECTOT
3506 TRAP ;PUSH OLD PSW AN PCON STACK

```

H11

MAINDEC - 11 - DZCOB-A MACY11 27(663) 19-FEB-76 14:38 PAGE 68
 DZCOBA.P11 T37 TEST OUTPUT STACKER FULL

SEQ 0137

```

3507 020334 016666 000002 000006 67$: MOV 2(SP),6(SP) ;;
3508 020342 012716 020350 MOV #68$, (SP) ;; REPLACE OLD PC WITH NEW
3509 020346 000002 RTI ;; RESTORE PSW
3510 020350 012637 000034 68$: MOV (SP)+,34 ;; RESTORE OLD TRAP VECTOR
3511 020354 012662 000002 MOV (SP)+,2(ADINT)
3512 : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3513 020360 005046 CLR -(SP) ;;
3514 020362 013746 000034 MOV 34,-(SP) ;; SAVE CURRENT TRAP VECTOR
3515 020366 012737 020376 000034 MOV #69$,34 ;; SETUP NEW TRAP VECTOR
3516 020374 104400 TRAP ;; PUSH OLD PSW AN PCON STACK
3517 020376 016666 000002 000006 69$: MOV 2(SP),6(SP) ;;
3518 020404 012716 020412 MOV #70$, (SP) ;; REPLACE OLD PC WITH NEW
3519 020410 000002 RTI ;; RESTORE PSW
3520 020412 012637 000034 70$: MOV (SP)+,34 ;; RESTORE OLD TRAP VECTOR
3521 020416 012637 001214 MOV (SP)+,$TMP6
3522 020422 042737 000340 001214 BIC #340,$TMP6
3523 020430 013746 001214 MOV $TMP6,-(SP) ;; PUT NEW PS ON STACK
3524 020434 012746 020442 MOV #71$,-(SP) ;; PUT NEW PC ON STACK
3525 020440 000002 RTI ;; POP NEW PC AND PS
3526 020442 71$:
3527 020442 012713 000100 MOV #100, ACDS ;; SET INTERRUPT ENABLE
3528 020446 104400 031274 TYPE, MSG1 ;; "PRESS CARD READER 'RESET'"
3529 020452 104400 030246 TYPE, CRLF-3 ;; MOVE MESSAGE UP ON TTY
3530 020456 000777 BR . ;; WAIT FOR THE INTERRUPT
3531
3532 020460 022626 T INTE: CMP (SP)+, (SP)+ ;; RESTORE THE STACK
3533 020462 012712 020710 MOV #TINTEA,ADINT ;; LOAD RETURN POINTER
3534 : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3535 020466 005046 CLR -(SP) ;;
3536 020470 013746 000034 MOV 34,-(SP) ;; SAVE CURRENT TRAP VECTOR
3537 020474 012737 020504 000034 MOV #64$,34 ;; SETUP NEW TRAP VECTOR
3538 020502 104400 TRAP ;; PUSH OLD PSW AN PCON STACK
3539 020504 016666 000002 000006 64$: MOV 2(SP),6(SP) ;;
3540 020512 012716 020520 MOV #65$, (SP) ;; REPLACE OLD PC WITH NEW
3541 020516 000002 RTI ;; RESTORE PSW
3542 020520 012637 000034 65$: MOV (SP)+,34 ;; RESTORE OLD TRAP VECTOR
3543 020524 012637 001214 MOV (SP)+,$TMP6
3544 020530 052737 000340 001214 BIS #340,$TMP6
3545 020536 013746 001214 MOV $TMP6,-(SP) ;; PUT NEW PS ON STACK
3546 020542 012746 020550 MOV #66$,-(SP) ;; PUT NEW PC ON STACK
3547 020546 000002 RTI ;; POP NEW PC AND PS
3548 020550 66$:
3549 : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
3550 020550 005046 CLR -(SP) ;;
3551 020552 013746 000034 MOV 34,-(SP) ;; SAVE CURRENT TRAP VECTOR
3552 020556 012737 020566 000034 MOV #67$,34 ;; SETUP NEW TRAP VECTOR
3553 020564 104400 TRAP ;; PUSH OLD PSW AN PCON STACK
3554 020566 016666 000002 000006 67$: MOV 2(SP),6(SP) ;;
3555 020574 012716 020602 MOV #68$, (SP) ;; REPLACE OLD PC WITH NEW
3556 020600 000002 RTI ;; RESTORE PSW
3557 020602 012637 000034 68$: MOV (SP)+,34 ;; RESTORE OLD TRAP VECTOR
3558 020606 012662 000002 MOV (SP)+,2(ADINT)
3559 : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3560 020612 005046 CLR -(SP) ;;
  
```

```

3561 020614 013746 000034      MOV      34, -(SP)      ;; SAVE CURRENT TRAP VECTOR
3562 020620 012737 020630 000034      MOV      #695, 34      ;; SETUP NEW TRAP VECTOR
3563 020626 104400      TRAP                      ;; PUSH OLD PSW AN PCOM STACK
3564 020630 016666 000002 000006 695:      MOV      2(SP), 6(SP)  ;;
3565 020636 012716 020644      MOV      #705, (SP)    ;; REPLACE OLD PC WITH NEW
3566 020642 000002      RTI                      ;; RESTORE PSW
3567 020644 012637 000034 705:      MOV      (SP)+, 34     ;; RESTORE OLD TRAP VECTOR
3568 020650 012637 001214      MOV      (SP)+, $TMP6
3569 020654 042737 000340 001214      BIC      #340, $TMP6
3570 020662 013746 001214      MOV      $TMP6, -(SP)  ;; PUT NEW PS ON STACK
3571 020666 012746 020674      MOV      #715, -(SP)  ;; PUT NEW PC ON STACK
3572 020672 000002      RTI                      ;; POP NEW PC AND PS
3573 020674 715:
3574 020674 012714 177701      MOV      #-77, @CDC    ;; SET UP COLUMN COUNT
3575 020700 012715 043766      MOV      #BUFBEG, @CDA ;; SET UP BUS ADDRESS
3576 020704 005213      INC      @CDS          ;; START READING
3577 020706 000777      BR      .              ;; WAIT FOR AN INTERRUPT
3578
3579 020710 022626      TINTEA: CMP      (SP)+, (SP)+ ;; RESTORE THE STACK
3580 020712 022713 000300      CMP      #000300, @CDS ;; CHECK THE CARD READER STATUS
3581 020716 001401      BEQ      15            ;; BRANCH IF OK
3582 020720 104004      ERROR +4              ;; CARD READER STATUS ERROR
3583
3584 020722 15:
3585 *****
3586 ;*TEST 40      TEST PICK CHECK ERROR
3587 *****
3588 020722 000004      †ST40: SCOPE
3589 ;A PICK CHECK ERROR SHOULD SET BIT 15, BIT 14, AND BIT 12
3590 ;THIS ERROR OCCURS WHEN THE FEED MECHANISM FAILS TO DELIVER A CARD TO
3591 ;THE READ STATION WITHIN 400 MS.
3592 ;CAN ALSO BE FORCED BY FOLDING A CARD IN HALF
3593 ;AND PLACING IT INTO CARD READER 'INPUT HOPPER'
3594 020724 004737 025442      JSR      PC INIT
3595 020730 104400 031463      TYPE,   MSG5          ;"REMOVE ALL CARDS FROM THE INPUT HOPPER"
3596 020734 104400 031330      TYPE,   MSG2          ;"THEN HIT 'CONTINUE' ON THE CONSOLE"
3597 020740 104400 031702      TYPE,   MSG8          ;"HOLD DOWN THE SWITCH UNDER THE CAP
3598 ;OF THE INPUT HOPPER"
3599 020744 104400 031274      TYPE,   MSG1          ;"PRESS CARD READER 'RESET'"
3600 020750 104400 030246      TYPE,   CRLF-3        ;MOVE MESSAGE UP ON TTY
3601 020754 000000      HALT
3602 020756 032713 010000      BIT      #10000, @CDS ;CHECK FOR OFF-LINE
3603 020762 001001      BNE     15            ;BRANCH IF SET
3604 020764 104043      ERROR +43           ;OFF LINE NOT SET AFTER "CONTINUE"
3605
3606 020766 032713 000010 15:      BIT      #10, @CDS    ;CHECK FOR "TRANSITION TO ON LINE"
3607 020772 001775      BEQ     15            ;WAIT FOR IT
3608 020774 022713 140210      CMP     #140210, @CDS ;CHECK FOR CORRECT STATUS BITS
3609 021000 001401      BEQ     25            ;BRANCH IF OK
3610 021002 104004      ERROR +4             ;STATUS NOT EQUAL TO 140210
3611
3612 021004 012714 177701 25:      MOV     #-77, @CDC    ;SET UP COLUMN COUNT
3613 021010 012715 043766      MOV     #BUFBEG, @CDA ;SET UP BUS ADDRESS
3614 021014 005213      INC     @CDS          ;READ

```

```

3615 021016 105713          3S:  TSTB   ACDS      ;CHECK CONTROLLER READY
3616 021020 100376          BPL     3S        ;WAIT FOR CONTROLLER READY
3617 021022 032713 010000  EIT     #10000,ACDS ;CHECK BIT12
3618 021026 001001          BNE     4S        ;BRANCH IF SET
3619 021030 104043          ERROR +43       ;OFF-LINE (BIT 12) WASN'T SET
3620
3621 021032 005713          4S:  TST     ACDS      ;CHECK SPECIAL CONDITION BIT
3622 021034 100401          BMI     5S        ;BRANCH IF SET
3623 021036 104026          ERROR +26       ;SPECIAL CONDITION NOT SET
3624
3625 021040 032713 040000  5S:  BIT     #40000, ACDS ;CHECK FOR CARD READER ERROR
3626 021044 001001          BNE     6S        ;BRANCH IF SET
3627 021046 104035          ERROR +35       ;CARD READER ERROR BIT 14 NOT SET
3628
3629 021050 005777 160164  6S:  TST     ACDOB      ;TEST BIT15 OF STATUS REGISTER #2
3630 021054 100005          BPL     7S        ;BRANCH IF NOT SET INDICATING
3631
3632 021056 032777 020000 160154  BIT     #20000,ACDOB ;OLD CD11 CONTROLLER
3633 021064 001001          BNE     7S        ;IS PICK CHECK INDICATOR SET?
3634 021066 104045          ERROR +45       ;BRANCH IF SET
3635
3636 021070 031327 027577  7S:  BIT     ACDS, #027577 ;CHECK FOR EXTRA BITS
3637 021074 001401          BEQ     10S       ;BRANCH IF OK
3638 021076 104015          ERROR +15       ;STATUS WORD ERROR
3639
3640 021100 012712 021334  10S: MOV     #TINTF, ADINT ;LOAD RETURN POINTER
3641
3642 021104 005046          CLR     -(SP)     ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3643 021106 013746 000034  MOV     34, -(SP) ;;;SAVE CURRENT TRAP VECTOR
3644 021112 012737 021122 000034  MOV     #64S, 34  ;;;SETUP NEW TRAP VECTOT
3645 021120 104400          TRAP                    ;;;PUSH OLD PSW AN PCON STACK
3646 021122 016666 000002 000006  64S: MOV     2(SP), 6(SP) ;;;
3647 021130 012716 021136          MOV     #65S, (SP) ;;;REPLACE OLD PC WITH NEW
3648 021134 000002          RTI                    ;;;RESTORE PSW
3649 021136 012637 000034  65S: MOV     (SP)+, 34    ;;;RESTORE OLD TRAP VECTOR
3650 021142 012637 001214          MOV     (SP)+, $TMP6
3651 021146 052737 000340 001214  BIS     #340, $TMP6
3652 021154 013746 001214          MOV     $TMP6, -(SP) ;;;PUT NEW PS ON STACK
3653 021160 012746 021166          MOV     #66S, -(SP) ;;;PUT NEW PC ON STACK
3654 021164 000002          RTI                    ;;;POP NEW PC AND PS
3655 021166
3656
3657 021166 005046          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
3658 021170 013746 000034  CLR     -(SP)     ;;;
3659 021174 012737 021204 000034  MOV     34, -(SP) ;;;SAVE CURRENT TRAP VECTOR
3660 021202 104400          MOV     #67S, 34  ;;;SETUP NEW TRAP VECTOT
3661 021204 016666 000002 000006  67S: MOV     2(SP), 6(SP) ;;;PUSH OLD PSW AN PCON STACK
3662 021212 012716 021220          MOV     #68S, (SP) ;;;
3663 021216 000002          RTI                    ;;;REPLACE OLD PC WITH NEW
3664 021220 012637 000034  68S: MOV     (SP)+, 34    ;;;RESTORE PSW
3665 021224 012662 000002          MOV     (SP)+, 2(ADINT) ;;;RESTORE OLD TRAP VECTOR
3666
3667 021230 005046          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3668 021232 013746 000034  CLR     -(SP)     ;;;
    MOV     34, -(SP) ;;;SAVE CURRENT TRAP VECTOR
    
```

```

3669 021236 012737 021246 000034      MOV      #69$,34      ;;SETUP NEW TRAP VECTOT
3670 021244 104400      TRAP                      ;;PUSH OLD PSW AN PCOM STACK
3671 021246 016666 000002 000006 69$:  MOV      2(SP),6(SP)  ;;
3672 021254 012716 021262      MOV      #70$, (SP)  ;;
3673 021260 000002      RTI                      ;;RESTORE PSW
3674 021262 012637 000034      MOV      (SP)+,34      ;;RESTORE OLD TRAP VECTOR
3675 021266 012637 001214      MOV      (SP)+,$TMP6
3676 021272 042737 000340 001214  BIC      #340,$TMP6
3677 021300 013746 001214      MOV      $TMP6,-(SP)  ;;PUT NEW PS ON STACK
3678 021304 012746 021312      MOV      #71$,-(SP)  ;;PUT NEW PC ON STACK
3679 021310 000002      RTI                      ;;POP NEW PC AND PS
3680 021312
3681 021312 012713 000100      MOV      #100, 2CDS  ;;SET INTERRUPT ENABLE
3682 021316 104400 031534      TYPE,   MSG6          ;;RESTORE CARDS TO THE INPUT HOPPER"
3683 021322 104400 031274      TYPE,   MSG1          ;;PRESS CARD READER 'RESET'"
3684 021326 104400 030246      TYPE,   CRLF-3       ;;MOVE MESSAGE UP ON TTY
3685 021332 000777      BR      .              ;;WAIT FOR THE INTERRUPT
3686
3687 021334 022626      TINTF:  CMP      (SP)+,(SP)+  ;;RESTORE THE STACK
3688 021336 012712 021564      MOV      #TINTFA,ADINT  ;;LOAD RETURN POINTER
3689      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3690 021342 005046      CLR      -(SP)
3691 021344 013746 000034      MOV      34,-(SP)      ;;SAVE CURRENT TRAP VECTOR
3692 021350 012737 021360 000034      MOV      #64$,34      ;;SETUP NEW TRAP VECTOT
3693 021356 104400      TRAP                      ;;PUSH OLD PSW AN PCOM STACK
3694 021360 016666 000002 000006 64$:  MOV      2(SP),6(SP)  ;;
3695 021366 012716 021374      MOV      #65$, (SP)  ;;
3696 021372 000002      RTI                      ;;RESTORE PSW
3697 021374 012637 000034      MOV      (SP)+,34      ;;RESTORE OLD TRAP VECTOR
3698 021400 012637 001214      MOV      (SP)+,$TMP6
3699 021404 052737 000340 001214  BIS      #340,$TMP6
3700 021412 013746 001214      MOV      $TMP6,-(SP)  ;;PUT NEW PS ON STACK
3701 021416 012746 021424      MOV      #66$,-(SP)  ;;PUT NEW PC ON STACK
3702 021422 000002      RTI                      ;;POP NEW PC AND PS
3703 021424
3704      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
3705 021424 005046      CLR      -(SP)
3706 021426 013746 000034      MOV      34,-(SP)      ;;SAVE CURRENT TRAP VECTOR
3707 021432 012737 021442 000034      MOV      #67$,34      ;;SETUP NEW TRAP VECTOT
3708 021440 104400      TRAP                      ;;PUSH OLD PSW AN PCOM STACK
3709 021442 016666 000002 000006 67$:  MOV      2(SP),6(SP)  ;;
3710 021450 012716 021456      MOV      #68$, (SP)  ;;
3711 021454 000002      RTI                      ;;RESTORE PSW
3712 021456 012637 000034      MOV      (SP)+,34      ;;RESTORE OLD TRAP VECTOR
3713 021462 012662 000002      MOV      (SP)+,2(ADINT)
3714      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3715 021466 005046      CLR      -(SP)
3716 021470 013746 000034      MOV      34,-(SP)      ;;SAVE CURRENT TRAP VECTOR
3717 021474 012737 021504 000034      MOV      #69$,34      ;;SETUP NEW TRAP VECTOT
3718 021502 104400      TRAP                      ;;PUSH OLD PSW AN PCOM STACK
3719 021504 016666 000002 000006 69$:  MOV      2(SP),6(SP)  ;;
3720 021512 012716 021520      MOV      #70$, (SP)  ;;
3721 021516 000002      RTI                      ;;RESTORE PSW
3722 021520 012637 000034      MOV      (SP)+,34      ;;RESTORE OLD TRAP VECTOR
    
```


3723 021524 012637 001214
 3724 021530 042737 000340 001214
 3725 021536 013746 001214
 3726 021542 012746 021550
 3727 021546 000002
 3728 021550
 3729 021550 012714 177701
 3730 021554 012715 043766
 3731 021560 005213
 3732 021562 000777
 3733
 3734 021564 022626
 3735 021566 022713 000300
 3736 021572 001401
 3737 021574 104004
 3738
 3739 021576
 3740
 3741
 3742
 3743 021576 000004
 3744
 3745
 3746
 3747 021600 004737 025442
 3748 021604 104400 031375
 3749 021610 104400 031773
 3750
 3751
 3752
 3753
 3754
 3755
 3756
 3757 021614 104400 031274
 3758 021620 104400 030246
 3759 021624 032713 010000
 3760 021630 001775
 3761 021632 032713 000010
 3762 021636 001775
 3763 021640 022713 100210
 3764 021644 001401
 3765 021646 104004
 3766
 3767 021650 012714 177701
 3768 021654 012715 043766
 3769 021660 005213
 3770 021662 105713
 3771 021664 100376
 3772 021666 032713 010000
 3773 021672 001001
 3774 021674 104043
 3775
 3776 021676 005713

```

MOV (SP)+,$TMP6
BIC #340,$TMP6
MOV $TMP6,-(SP)
MOV #715,-(SP)
RTI
;PUT NEW PS ON STACK
;PUT NEW PC ON STACK
;POP NEW PC AND PS

715:
MOV #-77,@CDC
MOV #BUFBEQ,@CDA
INC @CDS
BR .
;SET UP COLUMN COUNT
;SET UP BUS ADDRESS
;START READING
;WAIT FOR AN INTERRUPT

TINTFA: CMP (SP)+,(SP)+
CMP #000300,@CDS
BEQ 15
ERROR +4
;RESTORE THE STACK
;CHECK THE CARD READER STATUS
;BRANCH IF OK
;CARD READER STATUS ERROR

15:
;*****
;*TEST 41 TEST STACK CHECK ERROR
;*****
TST41: SCOPE
;A STACK CHECK ERROR SHOULD SET BIT 15, BIT 14, AND BIT 12
;THIS ERROR OCCURS WHEN THE FEED MECHANISM FAILS TO DELIVER A CARD TO
;THE READ STATION

JSR PC,INIT
TYPE,MSG3
TYPE,MSG9
;"PRESS CARD READER 'STOP'"
;"SLIDE A CARD FROM THE OUTPUT HOPPER ABOUT
;HALF AN INCH BACK INTO THE READ HEAD
;BLOCKING THE PHOTO CELL
;NOTE: SOME CARD READER MODELS MAY HAVE
;A LARGE ROLLER BLOCKING ACCESS TO THE PHOTOCCELL
;IN WHICH CASE, THE PHOTOCCELL CAN BE
;BLOCKED BY TEARING OFF A PIECE OF CARD
;AND SLIPPING IT IN FRONT OF THE PHOTOCCELL
;"PRESS CARD READER 'RESET'"
MOVE MESSAGE UP ON TTY

TLOPG: BIT #10000,@CDS
BEQ TLOPG
;CHECK FOR OFF-LINE
;WAIT FOR OFF-LINE

TLOPGA: BIT #10,@CDS
BEQ TLOPGA
;CHECK FOR "TRANSITION TO ON LINE"
;WAIT FOR IT

CMP #100210,@CDS
BEQ 15
ERROR +4
;CHECK FOR CORRECT STATUS BITS
;BRANCH IF OK
;STATUS NOT EQUAL TO 100210

15:
MOV #-77,@CDC
MOV #BUFBEQ,@CDA
INC @CDS
TLOPGB: TSTB @CDS
BPL TLOPGB
BIT #10000,@CDS
BNE 15
ERROR +43
;SET UP COLUMN COUNT
;SET UP BUS ADDRESS
;READ
;CHECK CONTROLLER READY
;WAIT FOR CONTROLLER READY
;CHECK BIT12
;BRANCH IF SET
;OFF-LINE (BIT 12) WASN'T SET

15:
TST @CDS
;CHECK SPECIAL CONDITION BIT

```

```

3777 021700 100401          BMI      2$          ;BRANCH IF SET
3778 021702 104026          ERROR +26          ;SPECIAL CONDITION NOT SET
3779
3780 021704 032713 040000  2$:  BIT      #40000, ACDS ;CHECK FOR CARD READER ERROR
3781 021710 001001          BNE      3$          ;BRANCH IF SET
3782 021712 104035          ERROR +35          ;CARD READER ERROR BIT 14 NOT SET
3783
3784 021714 005777 157320  3$:  TST      ACDD8      ;TEST BIT15 OF STATUS REGISTER #2
3785 021720 100005          BPL      4$          ;BRANCH IF NOT SET INDICATING
3786
3787 021722 032777 010000 157310  BIT      #10000, ACDD8 ;OLD CD11 CONTROLLER
3788 021730 001001          BNE      4$          ;IS STACK CHECK INDICATOR SET?
3789 021732 104046          ERROR +46          ;BRANCH IF SET
3790
3791 021734 032713 027577  4$:  BIT      #027577, ACDS ;CHECK FOR EXTRA BITS
3792 021740 001401          BEQ      5$          ;BRANCH IF OK
3793 021742 104015          ERROR +15          ;STATUS WORD ERROR
3794
3795 021744 012712 022200  5$:  MOV      #TINTG, ADINT ;LOAD RETURN POINTER
3796
3797 021750 005046          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340, PS"
3798 021752 013746 000034  CLR      -(SP)      ;
3799 021756 012737 021766 000034  MOV      34, -(SP)  ;SAVE CURRENT TRAP VECTOR
3800 021764 104400          MOV      #64$, 34   ;SETUP NEW TRAP VECTOT
3801 021766 016666 000002 000006 64$:  TRAP          ;PUSH OLD PSW AN PCON STACK
3802 021774 012716 022002          MOV      2(SP), 6(SP) ;
3803 022000 000002          MOV      #65$, (SP) ;REPLACE OLD PC WITH NEW
3804 022002 012637 000034  65$:  RTI          ;RESTORE PSW
3805 022006 012637 001214          ;RESTORE OLD TRAP VECTOR
3806 022012 052737 000340 001214  MOV      (SP)+, 34
3807 022020 013746 001214          MOV      (SP)+, $TMP6
3808 022024 012746 022032          BIS      #340, $TMP6 ;PUT NEW PS ON STACK
3809 022030 000002          MOV      $TMP6, -(SP) ;PUT NEW PC ON STACK
3810 022032          MOV      #66$, -(SP) ;POP NEW PC AND PS
3811
3812 022032 005046          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, 2(ADINT)"
3813 022034 013746 000034  CLR      -(SP)      ;
3814 022040 012737 022050 000034  MOV      34, -(SP)  ;SAVE CURRENT TRAP VECTOR
3815 022046 104400          MOV      #67$, 34   ;SETUP NEW TRAP VECTOT
3816 022050 016666 000002 000006 67$:  TRAP          ;PUSH OLD PSW AN PCON STACK
3817 022056 012716 022064          MOV      2(SP), 6(SP) ;
3818 022062 000002          MOV      #68$, (SP) ;REPLACE OLD PC WITH NEW
3819 022064 012637 000034  68$:  RTI          ;RESTORE PSW
3820 022070 012662 000002          ;RESTORE OLD TRAP VECTOR
3821
3822 022074 005046          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340, PS"
3823 022076 013746 000034  CLR      -(SP)      ;
3824 022102 012737 022112 000034  MOV      34, -(SP)  ;SAVE CURRENT TRAP VECTOR
3825 022110 104400          MOV      #69$, 34   ;SETUP NEW TRAP VECTOT
3826 022112 016666 000002 000006 69$:  TRAP          ;PUSH OLD PSW AN PCON STACK
3827 022120 012716 022126          MOV      2(SP), 6(SP) ;
3828 022124 000002          MOV      #70$, (SP) ;REPLACE OLD PC WITH NEW
3829 022126 012637 000034  70$:  RTI          ;RESTORE PSW
3830 022132 012637 001214          ;RESTORE OLD TRAP VECTOR
3831

```

3831	022136	042737	000340	001214	BIC	#340,\$TMP6	
3832	022144	013746	001214		MOV	\$TMP6,-(SP)	:: PUT NEW PS ON STACK
3833	022150	012746	022156		MOV	#71\$,-(SP)	:: PUT NEW PC ON STACK
3834	022154	000002			RTI		:: POP NEW PC AND PS
3835	022156			71\$:			
3836	022156	012713	000100		MOV	#100,ACDCS	:: SET INTERRUPT ENABLE
3837	022162	104400	032475		TYPE,	MSG10	:: "REMOVE JAMMED CARDS"
3838	022166	104400	031274		TYPE,	MSG1	:: "PRESS CARD READER 'RESET'"
3839	022172	104400	030246		TYPE,	CRLF-3	:: MOVE MESSAGE UP ON TTY
3840	022176	000777			BR	.	:: WAIT FOR THE INTERRUPT
3841							
3842	022200	022626		TINTG:	CMP	(SP)+,(SP)+	:: RESTORE THE STACK
3843	022202	012712	022430		MOV	#TINTGA,ADINT	:: LOAD RETURN POINTER
3844							:: THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3845	022206	005046			CLR	-(SP)	
3846	022210	013746	000034		MOV	34,-(SP)	:: SAVE CURRENT TRAP VECTOR
3847	022214	012737	022224	000034	MOV	#64\$,34	:: SETUP NEW TRAP VECTOT
3848	022222	104400			TRAP		:: PUSH OLD PSW AN PCON STACK
3849	022224	016666	000002	000006	64\$:	MOV	2(SP),6(SP)
3850	022232	012716	02240		MOV	#65\$, (SP)	:: REPLACE OLD PC WITH NEW
3851	022236	000002			RTI		:: RESTORE PSW
3852	022240	012637	000034	65\$:	MOV	(SP)+,34	:: RESTORE OLD TRAP VECTOR
3853	022244	012637	001214		MOV	(SP)+,\$TMP6	
3854	022250	052737	000340	001214	BIS	#340,\$TMP6	
3855	022256	013746	001214		MOV	\$TMP6,-(SP)	:: PUT NEW PS ON STACK
3856	022262	012746	022270		MOV	#66\$,-(SP)	:: PUT NEW PC ON STACK
3857	022266	000002			RTI		:: POP NEW PC AND PS
3858	022270			66\$:			
3859							:: THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
3860	022270	005046			CLR	-(SP)	
3861	022272	013746	000034		MOV	34,-(SP)	:: SAVE CURRENT TRAP VECTOR
3862	022276	012737	022306	000034	MOV	#67\$,34	:: SETUP NEW TRAP VECTOT
3863	022304	104400			TRAP		:: PUSH OLD PSW AN PCON STACK
3864	022306	016666	000002	000006	67\$:	MOV	2(SP),6(SP)
3865	022314	012716	022322		MOV	#68\$, (SP)	:: REPLACE OLD PC WITH NEW
3866	022320	000002			RTI		:: RESTORE PSW
3867	022322	012637	000034	68\$:	MOV	(SP)+,34	:: RESTORE OLD TRAP VECTOR
3868	022326	012662	000002		MOV	(SP)+,2(ADINT)	
3869							:: THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3870	022332	005046			CLR	-(SP)	
3871	022334	013746	000034		MOV	34,-(SP)	:: SAVE CURRENT TRAP VECTOR
3872	022340	012737	022350	000034	MOV	#69\$,34	:: SETUP NEW TRAP VECTOT
3873	022346	104400			TRAP		:: PUSH OLD PSW AN PCON STACK
3874	022350	016666	000002	000006	69\$:	MOV	2(SP),6(SP)
3875	022356	012716	022364		MOV	#70\$, (SP)	:: REPLACE OLD PC WITH NEW
3876	022362	000002			RTI		:: RESTORE PSW
3877	022364	012637	000034	70\$:	MOV	(SP)+,34	:: RESTORE OLD TRAP VECTOR
3878	022370	012637	001214		MOV	(SP)+,\$TMP6	
3879	022374	042737	000340	001214	BIC	#340,\$TMP6	
3880	022402	013746	001214		MOV	\$TMP6,-(SP)	:: PUT NEW PS ON STACK
3881	022406	012746	022414		MOV	#71\$,-(SP)	:: PUT NEW PC ON STACK
3882	022412	000002			RTI		:: POP NEW PC AND PS
3883	022414			71\$:			
3884	022414	012714	177701		MOV	#-77, ACDC	:: SET UP COLUMN COUNT

```

3885 022420 012715 043766      MOV      #BUFBEQ, @CDA      ;SET UP BUS ADDRESS
3886 022424 005213              INC      @CDS              ;START READING
3887 022426 000777              BR          .              ;WAIT FOR AN INTERRUPT
3888
3889 022430 022626      TINTGA: CMP      (SP)+, (SP)+ ;RESTORE THE STACK
3890 022432 022713 000300      CMP      #000300, @CDS    ;CHECK THE CARD READER STATUS
3891 022436 001401              BEQ      1$              ;BRANCH IF OK
3892 022440 104004              ERROR +4                ;CARD READER STATUS ERROR
3893
3894 022442              1$:
3895              ;ON M-1000/M-200 BIT 13 IS ALWAYS CLEARED
3896              ;ON M-1200 IF END OF FILE BUTTON IS PRESSED WITH INPUT
3897              ;HOPPER LOADED THEN WHEN INPUT HOPPER BECOMES EMPTY
3898              ;HOPPER CHECK INDICATOR LIGHT COMES ON AND BITS
3899              ;13 14 AND 15 ARE SET
3900
3901
3902 022442 005737 001266      TST      C01000          ;IS READER M1000/M200?
3903 022446 001402              BEQ      TSTM12          ;BRANCH IF READER IS M-1200
3904 022450 000137 023162      JMP      TSTM10          ;OUT OF THIS TEST IF M1000/M200
3905
3906 022454              TSTM12:
3907              ;*****
3908              ;*TEST 42      TEST 'END OF FILE' AND HOPPER CHECK
3909              ;*****
3910 022454 000004      TST42: SCOPE
3911 022456 004737 025442      JSR      PC,      INIT
3912 022462 104400 033436      TYPE,   MSG20      ;"PUT ANY TWO CARDS IN INPUT HOPPER"
3913 022466 104400 031274      TYPE,   MSG1       ;"PRESS CARD READER 'RESET'"
3914 022472 104400 031330      TYPE,   MSG2       ;"THEN HIT 'CONTINUE' ON THE CONSOLE"
3915 022476 104400 030246      TYPE,   CRLF-3     ;MOVE MESSAGE UP ON TTY
3916 022502 000000      HALT
3917
3918 022504 032713 000010      1$:  BIT      #10,    @CDS    ;CHECK FOR TRANSITION TO ON LINE
3919 022510 001775              BEQ      1$          ;WAIT FOR IT
3920 022512 104400 033502      TYPE,   MSG21      ;"PRESS END OF FILE BUTTON"
3921 022516 104400 031330      TYPE,   MSG2       ;"THEN HIT 'CONTINUE' ON THE CONSOLE"
3922 022522 104400 030246      TYPE,   CRLF-3     ;MOVE MESSAGE UP ON TTY
3923 022526 004737 025442      JSR      PC,      INIT
3924 022532 000000      HALT
3925
3926
3927 022534 032713 020000      BIT      #20000, @CDS    ;CHECK BIT 13
3928 022540 001401              BEQ      2$          ;BRANCH IF NOT SET
3929 022542 104047              ERROR +47          ;EOF SET FROM BEGINNING
3930
3931
3932 022544 032713 040000      2$:  BIT      #40000, @CDS ;CHECK BIT 14
3933 022550 001401              BEQ      3$          ;BRANCH IF NOT SET
3934 022552 104050              ERROR +50          ;READER CHECK ERROR SET FROM BEGINNING
3935
3936
3937 022554 032713 000004      3$:  BIT      #4,     @CDS  ;CHECK BIT 2
3938 022560 001401              BEQ      4$          ;BRANCH IF NOT SET

```

```

3939 022562 104051          ERROR +51          ;HOPPER CHECK SET FROM BEGINNING
3940
3941 022564 005713          4$:  TST      ACDS          ;CHECK ERROR BIT
3942 022566 100001          BPL      5$              ;BRANCH IF NOT SET
3943 022570 104014          ERROR +14          ;ERROR SET FROM BEGINNING
3944
3945
3946
3947
3948 022572 012712 023024          5$:  MOV      #TINTI, ADINT ;LOAD RETURN POINTER
3949 022576          SECN:
3950          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3951 022576 005046          CLR      -(SP)
3952 022600 013746 000034          MOV      34, -(SP)      ;SAVE CURRENT TRAP VECTOR
3953 022604 012737 022614 000034          MOV      #64$, 34      ;SETUP NEW TRAP VECTOR
3954 022612 104400          TRAP
3955 022614 016666 000002 000006 64$:  MOV      2(SP), 6(SP)    ;PUSH OLD PSW AN PCOM STACK
3956 022622 012716 022630          MOV      #65$, (SP)
3957 022626 000002          RTI
3958 022630 012637 000034          65$:  MOV      (SP)+, 34      ;REPLACE OLD PC WITH NEW
3959 022634 012637 001214          MOV      (SP)+, $TMP6   ;RESTORE PSW
3960 022640 052737 000340 001214          BIS      #340, $TMP6   ;RESTORE OLD TRAP VECTOR
3961 022646 013746 001214          MOV      $TMP6, -(SP)
3962 022652 012746 022660          MOV      #66$, -(SP)
3963 022656 000002          RTI
3964 022660          66$:
3965          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
3966 022660 005046          CLR      -(SP)
3967 022662 013746 000034          MOV      34, -(SP)
3968 022666 012737 022676 000034          MOV      #67$, 34      ;SAVE CURRENT TRAP VECTOR
3969 022674 104400          TRAP
3970 022676 016666 000002 000006 67$:  MOV      2(SP), 6(SP)    ;SETUP NEW TRAP VECTOR
3971 022704 012716 022712          MOV      #68$, (SP)    ;PUSH OLD PSW AN PCOM STACK
3972 022710 000002          RTI
3973 022712 012637 000034          68$:  MOV      (SP)+, 34      ;REPLACE OLD PC WITH NEW
3974 022716 012662 000002          MOV      (SP)+, 2(ADINT) ;RESTORE PSW
3975          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3976 022722 005046          CLR      -(SP)
3977 022724 013746 000034          MOV      34, -(SP)
3978 022730 012737 022740 000034          MOV      #69$, 34      ;SAVE CURRENT TRAP VECTOR
3979 022736 104400          TRAP
3980 022740 016666 000002 000006 69$:  MOV      2(SP), 6(SP)    ;SETUP NEW TRAP VECTOR
3981 022746 012716 022754          MOV      #70$, (SP)    ;PUSH OLD PSW AN PCOM STACK
3982 022752 000002          RTI
3983 022754 012637 000034          70$:  MOV      (SP)+, 34      ;REPLACE OLD PC WITH NEW
3984 022760 012637 001214          MOV      (SP)+, $TMP6   ;RESTORE PSW
3985 022764 042737 000340 001214          BIC      #340, $TMP6   ;RESTORE OLD TRAP VECTOR
3986 022772 013746 001214          MOV      $TMP6, -(SP)
3987 022776 012746 023004          MOV      #71$, -(SP)
3988 023002 000002          RTI
3989 023004          71$:
3990 023004 012713 000100          MOV      #100, ACDS     ;SET INTERRUPT ENABLE
3991 023010 012714 177701          MOV      #-77, ACDC    ;SET UP COLUMN COUNT
3992 023014 012715 043766          MOV      #BUFBEQ, ACDA ;SET UP BUS ADDRESS
    
```

```

3993 023020 005213      INC      @CDS      ;START READER
3994 023022 000777      BR          ;WAIT FOR AN INTERRUPT
3995
3996
3997 023024 022626      TINTI:  CMP      (SP)+, (SP)+ ;RESTORE THE STACK
3998
3999 023026 032713 020000      BIT      @20000, @CDS ;CHECK BIT 13
4000 023032 001401      BEQ      1$ ;BRANCH IF NOT SET
4001 023034 104047      ERROR +47 ;EOF SET AT END OF ONE CARD
4002
4003 023036 032713 040000      1$:      BIT      @40000, @CDS ;CHECK BIT 14
4004 023042 001401      BEQ      2$ ;BRANCH IF NOT SET
4005 023044 104050      ERROR +50 ;READER CHECK ERROR SET AT END OF ONE CARD
4006
4007 023046 005713      2$:      TST      @CDS ;CHECK ERROR BIT
4008 023050 100001      BPL      3$ ;BRANCH IF NOT SET
4009 023052 104014      ERROR +14 ;ERROR SET AT END OF ONE CARD
4010
4011 023054 012712 023062      3$:      MOV      @TINTIA, @ADINT ;LOAD RETURN POINTER
4012 023060 000646      BR          ;READ SECOND CARD
4013 023062 022626      TINTIA:  CMP      (SP)+, (SP)+ ;RESTORE THE STACK
4014
4015 023064 032713 020000      BIT      @20000, @CDS ;CHECK BIT 13
4016 023070 001001      BNE      1$ ;BRANCH IF SET
4017 023072 104052      ERROR +52 ;EOF NOT SET AT END OF FILE
4018
4019 023074 032713 040000      1$:      BIT      @40000, @CDS ;CHECK BIT 14
4020 023100 001001      BNE      2$ ;BRANCH IF SET
4021 023102 104053      ERROR +53 ;READER CHECK NOT SET AT END OF FILE
4022
4023 023104 032713 000004      2$:      BIT      @4, @CDS ;CHECK BIT 2
4024 023110 001001      BNE      3$ ;BRANCH IF SET
4025 023112 104054      ERROR +54 ;HOPPER CHECK NOT SET WHEN HOPPER EMPTY
4026
4027
4028 023114 005713      3$:      TST      @CDS ;CHECK ERROR BIT
4029 023116 100401      BMI      4$ ;BRANCH IF SET
4030 023120 104026      ERROR +26 ;ERROR BIT NOT SET AT END OF FILE
4031
4032 023122 104400 031534      4$:      TYPE,   MSG6 ;"RESTORE CARDS TO THE INPUT HOPPER"
4033 023126 104400 031274      TYPE,   MSG1 ;"PRESS CARD READER RESET"
4034 023132 104400 031330      TYPE,   MSG2 ;"THEN HIT CONTINUE ON THE CONSOLE"
4035 023136 104400 030246      TYPE,   CRLF-3 ;MOVE MESSAGE UP ON TTY
4036 023142 000000      HALT
4037
4038 023144 032713 000010      5$:      BIT      @10, @CDS ;CHECK TRANSITION TO ON LINE
4039 023150 001775      BEQ      5$ ;WAIT FOR IT
4040
4041
4042 023152 032713 020000      BIT      @20000, @CDS ;CHECK BIT 13
4043 023156 001401      BEQ      TSTM10 ;BRANCH IF NOT SET
4044 023160 104055      ERROR +55 ;EOF DIDN'T CLEAR BY TRANSITION TO ON LINE
4045
4046 023162      TSTM10:

```

```

4047
4048
4049
4050 023162 000004
4051
4052
4053
4054 023164 004737 025442
4055 023170 104400 032574
4056
4057 023174 104400 031274
4058 023200 104400 030246
4059 023204 032713 010000
4060 023210 001775
4061 023212 032713 000010
4062 023216 001775
4063 023220 022713 140210
4064 023224 001401
4065 023226 104004
4066
4067 023230 012714 177701
4068 023234 012715 043766
4069 023240 005213
4070 023242 105713
4071 023244 100376
4072 023246 032713 010000
4073 023252 001001
4074 023254 104043
4075
4076 023256 005713
4077 023260 100401
4078 023262 104026
4079
4080 023264 032713 040000
4081 023270 001001
4082 023272 104053
4083
4084 023274 005777 155740
4085 023300 100005
4086
4087 023302 032777 040000 155730
4088 023310 001001
4089 023312 104056
4090
4091 023314 032713 027577
4092 023320 001401
4093 023322 104015
4094
4095 023324 012712 023562
4096
4097 023330 005046
4098 023332 013746 000034
4099 023336 012737 023346 000034
4100 023344 104400

```

```

:*****
:*TEST 43 TEST READ CHECK ERROR
:*****
TST43: SCOPE
:A READ CHECK ERROR SHOULD SET BIT 15, BIT 14, AND BIT 12
:THIS ERROR OCCURS WHEN THE READ ELECTRONICS IN THE CARD
:READER DISAGREES WITH THE NORMAL UNPUNCHED AREA OF THE CARD
JSR PC,INIT
TYPE, MSG12 ;"PLACE SPECIAL DARK LIGHT CHECK CARD ONLY
;AT THE FRONT OF THE INPUT STACK"
TYPE, MSG1 ;"PRESS CARD READER 'RESET'"
TYPE, CRLF-3 ;MOVE MESSAGE UP ON TTY
TLOPH: BIT #10000, @CDS ;CHECK FOR OF LINE
BEQ TLOPH ;WAIT FOR OFF-LINE
TLOPHA: BIT #10, @CDS ;CHECK FOR "TRANSITION TO ON LINE"
BEQ TLOPHA ;WAIT FOR IT
CMP #140210, @CDS ;CHECK FOR CORRECT STATUS BITS
BEQ 15 ;BRANCH IF OK
ERROR +4 ;STATUS NOT EQUAL TO 140210

15: MOV #-77, @CDS ;SET UP COLUMN COUNT
MOV #BUFBEQ, @CDA ;SET UP BUS ADDRESS
INC @CDS ;READ
TLOPHB: TSTB @CDS ;CHECK CONTROLLER READY
BPL TLOPHB ;WAIT FOR CONTROLLER READY
BIT #10000, @CDS ;CHECK BIT12
BNE 15 ;BRANCH IF SET
ERROR +43 ;OFF-LINE (BIT 12) WASN'T SET

15: TST @CDS ;CHECK SPECIAL CONDITION BIT
BMI 25 ;BRANCH IF SET
ERROR +26 ;SPECIAL CONDITION NOT SET

25: BIT #40000, @CDS ;CHECK FOR CARD READER ERROR
BNE 35 ;BRANCH IF SET
ERROR +53 ;CARD READER ERROR BIT 14 NOT SET

35: TST @CDD8 ;TEST BIT15 OF STATUS REGISTER #2
BPL 45 ;BRANCH IF NOT SET INDICATING
;OLD CD11 CONTROLLER
BIT #40000, @CDD8 ;IS READ CHECK INDICATOR SET?
BNE 45 ;BRANCH IF SET
ERROR +56 ;READ CHECK BIT14 NOT SET

45: BIT #027577, @CDS ;CHECK FOR EXTRA BITS
BEQ 55 ;BRANCH IF OK
ERROR +15 ;STATUS WORD ERROR

55: MOV #TINTH, @ADINT ;LOAD RETURN POINTER
;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
CLR -(SP)
MOV 34, -(SP) ;SAVE CURRENT TRAP VECTOR
MOV #64$, 34 ;SETUP NEW TRAP VECTOR
TRAP ;PUSH OLD PSW AN PCON STACK

```

```

4101 023346 016666 000002 000006 64S:  MOV      2(SP), 6(SP)      ;;
4102 023354 012716 023362          MOV      #65S, (SP)      ;;
4103 023360 000002          RTI              ;;RESTORE PSW
4104 023362 012637 000034          MOV      (SP)+, 34      ;;RESTORE OLD TRAP VECTOR
4105 023366 012637 001214          MOV      (SP)+, $TMP6
4106 023372 052737 000340 001214          BIS      #340, $TMP6
4107 023400 013746 001214          MOV      $TMP6, -(SP)   ;;PUT NEW PS ON STACK
4108 023404 012746 023412          MOV      #66S, -(SP)   ;;PUT NEW PC ON STACK
4109 023410 000002          RTI              ;;POP NEW PC AND PS
4110 023412          66S:
4111          :THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, 2(ADINT),"
4112 023412 005046          CLR      -(SP)
4113 023414 013746 000034          MOV      34, -(SP)     ;;SAVE CURRENT TRAP VECTOR
4114 023420 012737 023430 000034          MOV      #67S, 34      ;;SETUP NEW TRAP VECTOR
4115 023426 104400          TRAP
4116 023430 016666 000002 000006 67S:  MOV      2(SP), 6(SP)   ;;
4117 023436 012716 023444          MOV      #68S, (SP)   ;;
4118 023442 000002          RTI              ;;RESTORE PSW
4119 023444 012637 000034          MOV      (SP)+, 34     ;;RESTORE OLD TRAP VECTOR
4120 023450 012662 000002          MOV      (SP)+, 2(ADINT)
4121          :THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340, PS"
4122 023454 005046          CLR      -(SP)
4123 023456 013746 000034          MOV      34, -(SP)     ;;SAVE CURRENT TRAP VECTOR
4124 023462 012737 023472 000034          MOV      #69S, 34      ;;SETUP NEW TRAP VECTOR
4125 023470 104400          TRAP
4126 023472 016666 000002 000006 69S:  MOV      2(SP), 6(SP)   ;;
4127 023500 012716 023506          MOV      #70S, (SP)   ;;
4128 023504 000002          RTI              ;;RESTORE PSW
4129 023506 012637 000034          MOV      (SP)+, 34     ;;RESTORE OLD TRAP VECTOR
4130 023512 012637 001214          MOV      (SP)+, $TMP6
4131 023516 042737 000340 001214          BIC      #340, $TMP6
4132 023524 013746 001214          MOV      $TMP6, -(SP)  ;;PUT NEW PS ON STACK
4133 023530 012746 023536          MOV      #71S, -(SP)  ;;PUT NEW PC ON STACK
4134 023534 000002          RTI              ;;POP NEW PC AND PS
4135 023536          71S:
4136 023536 012713 000100          MOV      #100, 2CDS    ;;SET INTERRUPT ENABLE
4137 023542 104400 031534          TYPE,   MSG6          ;;RESTORE CARDS TO THE INPUT HOPPER"
4138 023546 104400 031274          TYPE,   MSG1          ;;PRESS CARD READER 'RESET'"
4139 023552 104400 030246          TYPE,   CRLF-3        ;;MOVE MESSAGE UP ON TTY
4140 023556 000777          BR
4141 023560 000000          HALT
4142 023562 022626          TINTH:  CMP      (SP)+, (SP)+  ;;RESTORE THE STACK
4143 023564 000004          SCOPE
4144 023566 104400 001222          TYPE,   $BELL         ;;RING-A-DING
4145 023572 000137 016074          JMP      ER12CD       ;;LOOP BACK TO THE BEGINNING
4146
4147
4148
4149
4150          ;*****
4151          ;ROUTINE TO LOOP THRU A SINGLE INSTRUCTION TEST OR ERROR FUNCTION TEST
4152          ;NOTE THAT SW11 MUST BE DOWN AFTER 2ND HALT
4153          ;NOTE THAT SW<14> MUST BE SET.....MUST BE SET.....MUST BE SET.....
4154          ;*****
023576 TESTX:

```


4155	023576	012706	001100			MOV	#SCMTAG,R6	:: FIRST LOCATION TO BE CLEARED
4156	023602	005026				CLR	(R6)+	:: CLEAR MEMORY LOCATION
4157	023604	022706	001126			CMP	#SBOOAT,R6	:: DONE?
4158	023610	001374				BNE	.-6	:: LOOP BACK IF NO
4159	023612	012706	001100			MOV	#STACK,SP	:: SETUP THE STACK POINTER
4160	023616	012737	026024	000020		MOV	#SCOPE,@IOTVEC	:: IOT VECTOR FOR SCOPE ROUTINE
4161	023624	012737	000340	000022		MOV	#340,@IOTVEC+2	:: LEVEL 7
4162	023632	012737	025650	000030		MOV	#ERROR,@EMTVEC	:: EMT VECTOR FOR ERROR ROUTINE
4163	023640	012737	000340	000032		MOV	#340,@EMTVEC+2	:: LEVEL 7
4164	023646	012737	027072	000034		MOV	#TRAP,@TRAPVEC	:: TRAP VECTOR FOR TRAP CALLS
4165	023654	012737	000340	000036		MOV	#340,@TRAPVEC+2	:: LEVEL 7
4166	023662	012737	027126	000024		MOV	#SPWRON,@PWAVEC	:: POWER FAILURE VECTOR
4167	023670	012737	000340	000026		MOV	#340,@PWAVEC+2	:: LEVEL 7
4168	023676	013737	014740	014732		MOV	#ENDCT,#EOPCT	:: SETUP END-OF-PROGRAM COUNTER
4169	023704	005037	001220			CLR	#TIMES	:: INITIALIZE NUMBER OF ITERATIONS
4170	023710	012737	015104	000014		MOV	#SRTN,@TBITVEC	:: SET "T" BIT VECTOR TO SRTN
4171	023716	012737	000340	000016		MOV	#340,@TBITVEC+2	:: LEVEL 7
4172	023724	012737	000002	015104		MOV	#RTI,SRTN	:: SET SRTN TO A RTI
4173	023732	012737	023760	000010		MOV	#65S,@RESVEC	:: TRY TO DO A RTT
4174	023740	005046				CLR	-(SP)	:: DUMMY PS
4175	023742	012746	023750			MOV	#64S,-(SP)	:: AND PC
4176	023746	000006				RTT		:: TRY THE RTT
4177	023750	012737	000006	015104	64S:	MOV	#RTT,SRTN	:: RTT IS LEGAL--SET SRTN TO A RTT
4178	023756	000402				BR	66S	
4179	023760	062706	000010		65S:	ADD	#10,SP	:: RTT ILLEGAL--CLEAN OFF THE STACK
4180	023764	012737	000012	000010	66S:	MOV	#RESVEC+2,@RESVEC	:: RESTORE TRAP CATCHER
4181	023772	005037	015112			CLR	#TBIT	:: CLEAR "T" BIT SWITCH
4182	023776	012737	023776	001106		MOV	#SLPADR	:: INITIALIZE THE LOOP ADDRESS FOR SCOPE
4183	024004	013746	000004			MOV	@#4,-(SP)	:: SAVE ERROR VECTOR
4184	024010	013746	000006			MOV	@#6,-(SP)	
4185	024014	012737	024030	000004		MOV	#67S,4	:: SET UP TIME OUT VECTOR
4186	024022	005777	155110			TST	@SWR	:: TRY TO REFERENCE HARDWARE SWR
4187	024026	000407				BR	68S	:: BRANCH IF NO TIMEOUT TRAP OCCURS
4188	024030	012737	000176	001136	67S:	MOV	#SWREG,SWR	:: POINT TO SOFTWARE SWR
4189	024036	012737	000174	001140		MOV	#DISPREG,DISPLAY	:: POINT TO SOFTWARE DISPLAY REG
4190	024044	022626				CMP	(SP)+,(SP)+	:: RESTORE STACK
4191	024046	012637	000006		68S:	MOV	(SP)+,@#6	:: RESTORE ERROR VECTOR
4192	024052	012637	000004			MOV	(SP)+,@#4	
4193	024056	004737	043712			JSR	PC,SETUP	:: SETUP POINTERS AND FLAGS
4194	024062	104400	034073			TYPE,	MSG23	:: ASK USER TO LOAD ADDRESS OF DESIRED
4195								:: TEST INTO SWITCH REGISTER, THEN
4196								:: PRESS CONTINUE
4197	024066	000000				HALT		:: WAIT FOR STARTING ADDRESS
4198	024070	017737	155042	024320		MOV	@SWR,RETRNX	:: STORE STARTING ADDRESS
4199	024076	062737	000002	024320		ADD	#2,RETRNX	:: CHANGE TO FIRST ADDRESS AFTER SCOPE INSTRUCTION
4200	024104	104400	034342			TYPE,	MSG24	:: ASK USER TO SET SWITCH REGISTER
4201								:: OPTIONS, THEN PRESS CONTINUE
4202	024110	000000				HALT		:: SET SWR OPTIONS (BIT 11 MUST = 0)
4203	024112	032777	010000	155016		BIT	#10000,@SWR	:: CHECK SW12
4204	024120	001432				BEQ	1S	:: BRANCH IF NOT SET
4205								:: THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #20,PS"
4206	024122	005046				CLR	-(SP)	::
4207	024124	013746	000034			MOV	34,-(SP)	:: SAVE CURRENT TRAP VECTOR
4208	024130	012737	024140	000034		MOV	#69S,34	:: SETUP NEW TRAP VECTOR

```

4209 024136 104400 TRAP ;; PUSH OLD PSW AN PCON STACK
4210 024140 016666 000002 000006 69S: MOV 2(SP),6(SP) ;;
4211 024146 012716 024154 MOV #70S,(SP) ;; REPLACE OLD PC WITH NEW
4212 024152 000002 RTI ;; RESTORE PSW
4213 024154 012637 000034 70S: MOV (SP)+,34 ;; RESTORE OLD TRAP VECTOR
4214 024160 012637 001214 MOV (SP)+,$TMP6
4215 024164 042737 000020 001214 BIC #20,$TMP6
4216 024172 013746 001214 MOV $TMP6,-(SP) ;; PUT NEW PS ON STACK
4217 024176 012746 024204 MOV #71S,-(SP) ;; PUT NEW PC ON STACK
4218 024202 000002 RTI ;; POP NEW PC AND PS
4219 024204 71S:
4220 024204 000431 BR 2S ;SKIP NEXT INSTRUCTION
4221 024206 1S:
4222 : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "SIS #20,PS"
4223 024206 005046 CLR -(SP) ;;
4224 024210 013746 000034 MOV 34,-(SP) ;; SAVE CURRENT TRAP VECTOR
4225 024214 012737 024224 000034 MOV #72S,34 ;; SETUP NEW TRAP VECTOT
4226 024222 104400 TRAP ;; PUSH OLD PSW AN PCON STACK
4227 024224 016666 000002 000006 72S: MOV 2(SP),6(SP) ;;
4228 024232 012716 024240 MOV #73S,(SP) ;; REPLACE OLD PC WITH NEW
4229 024236 000002 RTI ;; RESTORE PSW
4230 024240 012637 000034 73S: MOV (SP)+,34 ;; RESTORE OLD TRAP VECTOR
4231 024244 012637 001214 MOV (SP)+,$TMP6
4232 024250 052737 000020 001214 BIS #20,$TMP6
4233 024256 013746 001214 MOV $TMP6,-(SP) ;; PUT NEW PS ON STACK
4234 024262 012746 024270 MOV #74S,-(SP) ;; PUT NEW PC ON STACK
4235 024266 000002 RTI ;; POP NEW PC AND PS
4236 024270 74S:
4237 024270 012737 024276 027274 2S: MOV #XLOOP,RETURN ;SAVE RETURN POINT FOR THIS SECTION
4238 ;FOR POWER FAILURE RETURN
4239 024276 104400 030254 XLOOP: TYPE, STMES ;TYPE MAINDEC TITLE & REV. LEVEL
4240 024302 104400 031152 TYPE, MLOOP ;INDICATE "ENTERING A LOOP ON TEST
4241 ;SELECTED BY THE USER"
4242 024306 013737 024320 001106 MOV RETRNX,$LPADR ;STORE 1ST ADDRESS OF TEST FOR LOOPING
4243 ;$LPADR WILL BE PICKED UP BY 'SCOPE'
4244 024314 000177 000000 JMP @RETRNX ;JUMP TO TEST SELECTED
4245 024320 000000 RETRNX: 0

```

4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262

```

*****
;ROUTINE TO CHECK CARDS WHICH HAVE ALL COLUMNS IDENTICALLY PUNCHED.
;THIS ROUTINE ALLOWS SPECIFIC TYPES OF DATA FAILURES TO BE STUDIED
;EASILY. THE ROUTINE HALTS ONCE AT THE START. SET THE CORRECT CARD
;IMAGE PATTERN IN SW11-SW00, THEN HIT CONTINUE (AFTER THE DECK IS
;LOADED AND CARD READER IS ON-LINE). THE PATTERN IS STORED, AND THEN
;EACH COLUMN OF EACH CARD IS READ TWICE AND COMPARED WITH IT. IF A
;DISCREPANCY OCCURS, THE ERROR IS PRINTED OUT ALONG WITH THE TOTAL
;NUMBER OF CARDS READ AND THE TOTAL NUMBER OF DATA ERRORS DISCOVERED
;UP TO THAT POINT (ALL PRINTOUTS ARE IN OCTAL). WHEN THE INPUT HOPPER
;IS EMPTY, THE ROUTINE RINGS THE BELL AND WAITS FOR MORE CARDS TO BE
;LOADED AND THE CARD READER TO BE PUT BACK ON-LINE.
;SW15=1 CAUSES A HALT AFTER AN ERROR, AND SW13=1 INHIBITS ERROR PRINTOUTS.
*****

```


4317	024660	006037	025436	ROR	CDPK0			
4318	024664	106137	025437	ROLB	CDPK1			
4319	024670	106037	025436	RORB	CDPK0			
4320	024674	106137	025437	ROLB	CDPK1			
4321	024700	106137	025437	ROLB	CDPK1			
4322	024704	106137	025437	ROLB	CDPK1			
4323	024710	106137	025437	ROLB	CDPK1			
4324	024714	012701	000007	MOV	#7	R1		
4325	024720	006037	025436	CKLOP1: ROR	CDPK0			
4326	024724	103004		BCC	CKOVR			
4327	024726	005237	025440	INC	DERFLG			
4328	024732	150137	025437	BISB	R1,CDPK1			
4329	024736	005301		CKOVR: DEC	R1			
4330	024740	001367		BNE	CKLOP1			
4331	024742	104400	034342	TYPE,	MSG24			;ASK USER TO SET SWITCH REGISTER ;OPTIONS, THEN PRESS CONTINUE
4332								;WAIT FOR SWITCH SETTINGS
4333	024746	000000		CKSTRT: HALT				
4334	024750	004737	025442	JSR	PC,INIT			
4335	024754	005037	025432	CLR	TOTCRD			;INITIALIZE CARD COUNT
4336	024760	005037	025430	CLR	TOTERR			;INITIALIZE ERROR COUNT
4337	024764	005037	001260	CLR	ERFLG			;CLEAR FLAG FOR PRINTING ERROR HEADING
4338	024770	105037	025441	CKLOOP: CLR	DERFLG+1			
4339	024774	032777	000010	BIT	#10	JSWR		;CHECK FOR PACK MODE ONLY
4340	025002	001410		BEG	CKREAD			;BRANCH IF NOT SET
4341	025004	032777	000004	BIT	#4	JSWR		;CHECK FOR IMAGE MODE ONLY
4342	025012	001004		BNE	CKREAD			;BRANCH IF SET
4343	025014	052713	000002	BIS	#2	ACDS		;SET PACKING MODE
4344	025020	105137	025441	CKREAD: COMB	DERFLG+1			
4345	025024	005037	015300	CLR	CLCNT			;INITIALIZE COLUMN COUNT
4346	025030	012700	043766	MOV	#BUFBEQ,RO			;SET UP BUFFER POINTER
4347	025034	012714	177660	MOV	#-120,	ACDC		;SET UP COLUMN COUNTER
4348	025040	010015		MOV	RO,	ACDA		;SET UP BUS ADDRESS
4349	025042	005213		INC	ACDS			;START READING CARD
4350	025044	005237	025432	CKLP1: INC	TOTCRD			;INCREMENT CARD COUNT
4351	025050	105713		TSTB	ACDS			;CHECK CONTROLLER READY
4352	025052	100376		BPL	CKLP1			;LOOP IF NOT SET
4353	025054	005713		TST	ACDS			;CHECK FOR ERROR
4354	025056	100435		BMI	CKERR			;BRANCH IF ERROR SET
4355	025060	005737	025440	TST	DERFLG			
4356	025064	100012		BPL	CKLOP2			
4357	025066	122037	025437	CKLOP3: CMPB	(RO)+,CDPK1			;CHECK DATA (PACKED MODE)
4358	025072	001054		BNE	CKFAIL			
4359	025074	005237	015300	INC	CLCNT			
4360	025100	023727	015300	CMP	CLCNT,#120			
4361	025106	001367		BNE	CKLOP3			
4362	025110	000727		BR	CKLOOP			
4363	025112	012037	001200	CKLOP2: MOV	(RO)+,\$TMP0			;GET VALUE READ FROM CARD
4364	025116	042737	170000	BIC	#170000,\$TMP0			;STRIP OFF BITS 15 THRU 12
4365								;IN THE EVENT WE ARE TESTING A CD20
4366	025124	023737	001200	CMP	\$TMP0,CARDIM			;CHECK THE DATA (IMAGE MODE)
4367	025132	001034	025434	BNE	CKFAIL			;BRANCH IF DATA ERROR
4368	025134	005237	015300	INC	CLCNT			;COUNT THE COLUMNS
4369	025140	023727	015300	CMP	CLCNT,#120			;CHECK FOR LAST COLUMN
4370	025146	001361		BNE	CKLOP2			

4371	025150	000707		BR	CKLOOP		
4372							
4373	025152	032713	010000	CKERR:	BIT #10000, ACDS		;CHECK FOR OFFLINE
4374	025156	001406			BEQ CKERR1		;BRANCH IF NOT
4375	025160	104400	001222		TYPE, SBELL		;RING-A-DING
4376	025164	032713	000010	CKERR3:	BIT #10, ACDS		;CHECK TRANSITION TO ON-LINE
4377	025170	001775			BEQ CKERR3		;BRANCH IF OFF-LINE
4378	025172	000666			BR CKSTRT		;START OVER
4379							
4380	025174	032713	004000	CKERR1:	BIT #4000, ACDS		;CHECK FOR DATA ERROR
4381	025200	001407			BEQ CKERR2		
4382	025202	005737	025440		TST DERFLG		
4383	025206	100004			BPL CKERR2		
4384	025210	122737	000001 025440		CMPB #1, DERFLG		
4385	025216	003323			BGT CKLOP3		;BRANCH IF LEGIT
4386	025220	104062		CKERR2:	ERROR +62		;REAL, LIVE ERROR.
4387	025222	000662			BR CKLOOP		
4388							
4389	025224	005237	025430	CKFAIL:	INC TOTERR		;COUNT ERRORS
4390	025230	032777	020000 153700		BIT #20000, ASWR		;CHECK FOR INHIBITING PRINTOUT
4391	025236	001052			BNE CKERROR		;BRANCH AROUND PRINTOUT IF SET
4392	025240	005737	001260		TST ERFLG		;TEST FLAG TO PRINT HEADING
4393	025244	001004			BNE CKNOHD		;BRANCH IF ALREADY DONE
4394	025246	005237	001260		INC ERFLG		;PRINT HEADING ONCE ONLY
4395	025252	104400	033372		TYPE, MSG19		;OUTPUT HEADING
4396	025256	104400	030251	CKNOHD:	TYPE, CRLF		;OUTPUT CARRIAGE RETURN, LINEFEED
4397	025262	005237	015300		INC CLCNT		;STEP UP COLUMN COUNT FOR ERROR REPORT
4398	025266	013746	015300		MOV CLCNT, -(SP)		;PRINT THE COLUMN OF THE CARD
4399	025272	104404			TYPDS		;ON WHICH AN ERROR WAS DETECTED
4400	025274	104400	030243		TYPE, SPACE		
4401	025300	005337	015300		DEC CLCNT		;DROP COLUMN COUNT BACK AFTER REPORTING
4402	025304	005737	025440		TST DERFLG		
4403	025310	100007			BPL CKNOPK		
4404	025312	104400	030243		TYPE, SPACE		
4405	025316	114046			MOVB -(RO), -(SP)		;PRINT THE INCORRECT VALUE
4406	025320	104402			TYPDS		;THAT WAS READ FROM THE
4407	025322	003			.BYTE 3		;COLUMN INDICATED ABOVE
4408	025323	000			.BYTE 0		; (PACKED MODE)
4409	025324	105720			TSTB (RO)+		;CORRECT MEMORY LOCATION AFTER PRINTOUT
4410	025326	000404			BR CKOVR1		
4411	025330	014046		CKNOPK:	MOV -(RO), -(SP)		;PRINT THE INCORRECT VALUE THAT
4412	025332	104402			TYPDS		;WAS READ FROM THE COLUMN
4413	025334	006			.BYTE 6		;INDICATED ABOVE
4414	025335	001			.BYTE 1		; (IMAGE MODE)
4415	025336	005720			TST (RO)+		;CORRECT THE MEMORY LOCATION AFTER PRINTOUT
4416	025340	104400	030243	CKOVR1:	TYPE, SPACE		
4417	025344	013746	025432		MOV TOTCRD, -(SP)		;PRINT THE CARD NUMBER ON WHICH
4418	025350	104404			TYPDS		;THE ERROR WAS FOUND
4419	025352	104400	030243		TYPE, SPACE		
4420	025356	013746	025430		MOV TOTERR, -(SP)		;PRINT THE TOTAL NUMBER OF ERRORS
4421	025362	104404			TYPDS		;ACCUMULATED TO NOW
4422	025364	005777	153546	CKERROR:	TST ASWR		;CHECK SWIS TO HALT ON ERROR
4423	025370	100001			BPL IS		;BRANCH IF NOT SET
4424	025372	000000			HALT		;HALT ON ERROR

4425	025374	005237	015300		1\$:	INC	CLCNT	
4426								
4427	025400	023727	015300	000120		CMP	CLCNT, #120	
4428	025406	001002				BNE	2\$	
4429	025410	000137	024770			JMP	CKLOOP	
4430	025414	032777	000004	153514	2\$:	BIT	#4, JSWR	
4431	025422	001233				BNE	CKLOP2	
4432	025424	000137	025066			JMP	CKLOP3	
4433								
4434								
4435	025430	000000				TOTERR:	0	
4436	025432	000000				TOTCRD:	0	
4437	025434	000000				CARDIM:	0	
4438	025436	000				CDPK0:	.BYTE	0
4439	025437	000				CDPK1:	.BYTE	0
4440	025440	000000				DERFLG:	0	

```

:STEP UP COLUMN COUNT TO BE CORRECT FOR
:NEXT POSSIBLE COLUMN TO BE CHECKED
:HAVE WE FINISHED A CARD?
:BRANCH IF NO
:OTHERWISE, GO TO SET UP FOR NEXT CARD
:ARE WE DOING PACKED MODE?
:BRANCH IF NOT
:OTHERWISE, CONTINUE CHECKING COLUMNS
:IN PACKED MODE

```

```

4441
4442 ;ISSUE MESSAGE IF CARD READER IS OFF-LINE
4443 ;WAIT FOR BUSY TO CLEAR IN CASE CARD READER IS STILL READING A CARD
4444 ;INITIALIZE STATUS REGISTER AND USE ERROR HALT IF IT DOESN'T CLEAR PROPERLY
4445 ;NOTE THAT PROGRAM WILL HANG HERE IF BUSY REMAINS SET
4446 025442 004737 025470 INIT: JSR PC CKOFFL ;SEE IF OFF-LINE BIT IS SET
4447 025446 105713 1S: TSTB 2CDS ;WAIT FOR CONTROLLER READY IN CASE
4448 025450 100376 BPL 1S ;A CARD IS STILL BEING READ
4449 025452 012713 000400 MOV #400, 2CDS ;INITIALIZE THE CARD READER
4450 025456 022713 000200 CMP #200, 2CDS ;MAKE SURC INITIALIZATION OK
4451 025462 001401 BEQ 2S ;BRANCH IF ALL BITS ZERO
4452 025464 104067 ERROR +67 ;NOT ALL BITS OF STATUS REGISTER ARE ZERO
4453 025466 030207 2S: RTS PC ;RETURN
4454
4455 ;SUBROUTINE TO CHECK FOR BIT 12 (OFF-LINE) BEING SET IN CARD
4456 ;READER CSR, AND PRINT OUT A MESSAGE IF IT IS
4457 025470 032713 010000 CKOFFL: BIT #10000, 2CDS ;CHECK BIT 12
4458 025474 001001 BNE 1S ;BRANCH IF SET
4459 025476 000207 RTS PC ;RETURN IF NOT SET
4460 025500 104400 033340 1S: TYPE, MSG18 ;"CARD READER OFF-LINE"
4461 025504 104400 033125 TYPE, MSG17 ;"REMEDY THE CONDITION ... ETC."
4462 025510 000000 HALT ;WAIT FOR CONTINUE
4463 025512 000766 BR CKOFFL ;CHECK AGAIN
4464
4465
4466 ;*****
4467
4468 .SBTTL ERROR MESSAGE TYPEOUT ROUTINE
4469
4470 ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
4471 ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
4472 ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
4473
4474 $ERRTYF:
4475 025514 104400 001227 TYPE $SCRLF ;;"CARRIAGE RETURN" & "LINE FEED"
4476 025520 010046 MOV RO, -(SP) ;SAVE RO
4477 025522 005000 CLR RO ;;PICKUP THE ITEM INDEX
4478 025524 153700 001114 BISB 2*$ITEMB, RO
4479 025530 001004 BNE 1S ;; IF ITEM NUMBER IS ZERO, JUST
4480 ;; TYPE THE PC OF THE ERROR
4481 025532 013746 001116 MOV $ERRPC, -(SP) ;; SAVE $ERRPC FOR TYPEOUT
4482 ;; ERROR ADDRESS
4483 025536 104401 TYP0C ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
4484 025540 000426 BR 6S ;; GET OUT
4485 025542 005300 1S: DEC RO ;; ADJUST THE INDEX SO THAT IT WILL
4486 025544 006300 ASL RO ;; WORK FOR THE ERROR TABLE
4487 025546 006300 ASL RO
4488 025550 006300 ASL RO
4489 025552 062700 001270 ADD #$ERRTB, RO ;; FORM TABLE POINTER
4490 025556 012037 025566 MOV (RO)+, 2S ;; PICKUP "ERROR MESSAGE" POINTER
4491 025562 001404 BEQ 3S ;; SKIP TYPEOUT IF NO POINTER
4492 025564 104400 TYPE ;; TYPE THE "ERROR MESSAGE"
4493 025566 000000 2S: .WORD 0 ;;"ERROR MESSAGE" POINTER GOES HERE
4494 025570 104400 001227 TYPE $SCRLF ;;"CARRIAGE RETURN" & "LINE FEED"

```

```

4495 025574 012037 025604 35: MOV (R0)+,45 ;: PICKUP "DATA HEADER" POINTER
4496 025600 001404 BEQ 55 ;: SKIP TYPEOUT IF 0
4497 025602 104400 TYPE ;: TYPE THE "DATA HEADER"
4498 025604 000000 45: .WORD 0 ;: "DATA HEADER" POINTER GOES HERE
4499 025606 104400 001227 TYPE ,SCLF ;: "CARRIAGE RETURN" & "LINE FEED"
4500 025612 011000 55: MOV (R0),R0 ;: PICKUP "DATA TABLE" POINTER
4501 025614 001004 BNE 75 ;: GO TYPE THE DATA
4502 025616 012600 65: MOV (SP)+,R0 ;: RESTORE R0
4503 025620 104400 001227 TYPE ,SCLF ;: "CARRIAGE RETURN" & "LINE FEED"
4504 025624 000207 RTS PC ;: RETURN
4505 025626 75:
4506 025626 013046 MOV 2(R0)+,-(SP) ;: SAVE 2(R0)+ FOR TYPEOUT
4507 025630 104401 TYPOC ;: GO TYPE--OCTAL ASCII(ALL DIGITS)
4508 025632 005710 TST (R0) ;: IS THERE ANOTHER NUMBER?
4509 025634 001770 BEQ 65 ;: BR IF NO
4510 025636 104400 025644 TYPE ,85 ;: TYPE TWO(2) SPACES
4511 025642 000771 BR 75 ;: LOOP
4512 025644 020040 000 85: .ASCIZ / / ;: TWO(2) SPACES
4513 025650 .EVEN
4514
4515 ;:*****
4516
4517 .SBTTL ERROR HANDLER ROUTINE
4518
4519 ;: *THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
4520 ;: *SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
4521 ;: *AND GO TO $ERRTYP ON ERROR
4522 ;: *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4523 ;: *SW15=1 HALT ON ERROR
4524 ;: *SW13=1 INHIBIT ERROR TYPEOUTS
4525 ;: *SW10=1 BELL ON ERROR
4526 ;: *CALL
4527 ;: * ERROR N ;: ;ERROR=EMT AND N=ERROR ITEM NUMBER
4528
4529 $ERROR:
4530 025650 010637 001174 MOV SP,$REG6 ;: STACK POINTER POSITION
4531 025654 016637 000002 001176 MOV 2(SP),$REG7 ;: CONTENTS OF 'PSW'
4532 025662 011337 001200 MOV 2CDS,$TMP0 ;: CONTENTS OF DEVICE 'CDS'
4533 025666 017737 153346 001202 MOV 2CDD8,$TMP1 ;: CONTENTS OF DEVICE 'CDD'
4534 025674 011437 001204 MOV 2CDC,$TMP2 ;: CONTENTS OF DEVICE 'CDC'
4535 025700 011537 001206 MOV 2CDA,$TMP3 ;: CONTENTS OF DEVICE 'CDA'
4536 025704 105237 001103 75: INCB $ERFLG ;: SET THE ERROR FLAG
4537 025710 001775 BEQ 75 ;: DON'T LET THE FLAG GO TO ZERO
4538 025712 013777 001102 153220 MOV $STNM,2DISPLAY ;: DISPLAY TEST NUMBER AND ERROR FLAG
4539 025720 032777 002000 153210 BIT #BIT10,2SWR ;: BELL ON ERROR?
4540 025726 001402 BEQ 15 ;: NO - SKIP
4541 025730 104400 001222 TYPE ,SBELL ;: RING BELL
4542 025734 005237 001112 15: INC $ERTTL ;: COUNT THE NUMBER OF ERRORS
4543 025740 011637 001116 MOV (SP),$ERRPC ;: GET ADDRESS OF ERROR INSTRUCTION
4544 025744 162737 000002 001116 SUB #2,$ERRPC
4545 025752 117737 153140 001114 MOVB 2$ERRPC,$ITEMB ;: STRIP AND SAVE THE ERROR ITEM CODE
4546 025760 032777 020000 153150 BIT #BIT13,2SWR ;: SKIP TYPEOUT IF SET
4547 025766 001004 BNE 205 ;: SKIP TYPEOUTS
4548 025770 004737 025514 JSR PC,$ERRTYP ;: GO TO USER ERROR ROUTINE
    
```



```

4549 025774 104400 001227          TYPE      .SCLF
4550 026000
4551 026000 005777 153132          20$:      TST      2SWR      ;; HALT ON ERROR
4552 026004 100006          2$:      BPL      3$          ;; SKIP IF CONTINUE
4553 026006 000000          HALT          ;; HALT ON ERROR!
4554 026010 022737 015012 000042          CMP      2SENDAD,2042  ;; ACT-11 AUTO-ACCEPT?
4555 026016 001001          BNE      3$          ;; BRANCH IF NO
4556 026020 000000          HALT          ;; YES
4557 026022
4558 026022 000002          3$:      RTI          ;; RETURN
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573 026024
4574 026024 032777 040000 153104          $SCOPE:
4575 026032 001055          1$:      BIT      2BIT14,2SWR  ;; LOOP ON PRESENT TEST?
4576
4577 026034 000416          BNE      $OVER          ;; YES IF SW14=1
4578
4579 026036 013746 000004          : 222222START OF CODE FOR THE XOR TESTER22222
4580 026042 012737 026062 000004          $XTSTR: BR      6$          ;; IF RUNNING ON THE "XOR" TESTER CHANGE
4581 026050 005737 177060          MOV      2ERRVEC, -(SP)  ;; THIS INSTRUCTION TO A "NOP" (NOP=240)
4582 026054 012637 000004          MOV      258,2ERRVEC  ;; SAVE THE CONTENTS OF THE ERROR VECTOR
4583 026060 000436          TST      2177060  ;; SET FOR TIMEOUT
4584 026062 022626          MOV      (SP)+,2ERRVEC  ;; TIME OUT ON XOR?
4585 026064 012637 000004          BR      25VLAD  ;; RESTORE THE ERROR VECTOR
4586 026070 000436          BR      25VLAD  ;; GO TO THE NEXT TEST
4587 026072
4588 026072 105737 001103          5$:      CMP      (SP)+,(SP)+  ;; CLEAR THE STACK AFTER A TIME OUT
4589 026076 001404          MOV      (SP)+,2ERRVEC  ;; RESTORE THE ERROR VECTOR
4590 026100 105037 001103          BR      $OVER  ;; LOOP ON THE PRESENT TEST
4591 026104 005037 001220          6$: ; 222222END OF CODE FOR THE XOR TESTER22222
4592 026110 032777 004000 153020          2$:      TSTB     SERFLG  ;; WAS AN ERROR OCCURRED?
4593 026116 001011          BEQ     3$          ;; BR IF NO
4594 026120 005737 001100          4$:      CLRB     SERFLG  ;; ZERO THE ERROR FLAG
4595 026124 001406          CLR     $TIMES  ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
4596 026126 005237 001104          3$:      BIT      2BIT11,2SWR  ;; INHIBIT ITERATIONS?
4597 026132 023737 001220 001104          BNE     1$          ;; BR IF YES
4598 026140 002012          TST     $PASS  ;; IF FIRST PASS OF PROGRAM
4599 026142 012737 000001 001104          BEQ     1$          ;; INHIBIT ITERATIONS
4600 026150 013737 026202 001220          INC     $ICNT  ;; INCREMENT ITERATION COUNT
4601 026156 105237 001102          CMP     $TIMES,$ICNT  ;; CHECK THE NUMBER OF ITERATIONS MADE
4602 026162 011637 001106          BGE     $OVER  ;; BR IF MORE ITERATION REQUIRED
          1$:      MOV     21,$ICNT  ;; REINITIALIZE THE ITERATION COUNTER
          2$:      MOV     2MXCNT,$TIMES  ;; SET NUMBER OF ITERATIONS TO DO
          3$:      INCB     $STNM  ;; COUNT TEST NUMBERS
          4$:      MOV     (SP),2LPAOR  ;; SAVE SCOPE LOOP ADDRESS
    
```

```

4603 026166 013777 001102 152744 $OVER: MOV $STNM, $DISPLAY ;; DISPLAY TEST NUMBER
4604 026174 013716 001106          MOV $LPRDR, (SP) ;; FUDGE RETURN ADDRESS
4605 026200 000002          RTI ;; FIXES PS
4606 026202 000001          $MXCNT: 1 ;; MAX. NUMBER OF ITERATIONS
4607
4608 ;*****
4609
4610 .SBTTL TYPE ROUTINE
4611
4612 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
4613 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
4614 ;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
4615 ;*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
4616 ;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
4617 ;*
4618 ;*CALL:
4619 ;*1) USING A TRAP INSTRUCTION
4620 ;* TYPE ,MESADR ;; MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
4621 ;*OR
4622 ;* TYPE
4623 ;* MESADR
4624 ;*
4625
4626 026204 105737 001155 $TYPE: TSTB $STPFLG ;; IS THERE A TERMINAL?
4627 026210 100002          BPL 1$ ;; BR IF YES
4628 026212 000000          HALT ;; HALT HERE IF NO TERMINAL
4629 026214 000407          BR 3$ ;; LEAVE
4630 026216 010046          1$: MOV RO, -(SP) ;; SAVE RO
4631 026220 017600 000002          MOV $2(SP), RO ;; GET ADDRESS OF ASCIZ STRING
4632 026224 112046          2$: MOVB (RO)+, -(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
4633 026226 001005          BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
4634 026230 005726          TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
4635 026232 012600          60$: MOV (SP)+, RO ;; RESTORE RO
4636 026234 062716 000002          3$: ADD $2, (SP) ;; ADJUST RETURN PC
4637 026240 000002          RTI ;; RETURN
4638 026242 122716 000011          4$: CMPB $HT, (SP) ;; BRANCH IF <HT>
4639 026246 001426          BEQ 8$
4640 026250 122716 000200          CMPB $TCRLF, (SP) ;; BRANCH IF NOT <CRLF>
4641 026254 001004          BNE 5$
4642 026256 005726          TST (SP)+ ;; POP <CR><LF> EQUIV
4643 026260 104400          TYPE ;; TYPE A CR AND LF
4644 026262 001227          $CRLF
4645 026264 000757          BR 2$ ;; GET NEXT CHARACTER
4646 026266 004737 026350          5$: JSR PC, $TYPEC ;; GO TYPE THIS CHARACTER
4647 026272 123726 001154          6$: CMPB $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
4648 026276 001352          BNE 2$ ;; IF NO GO GET NEXT CHAR.
4649 026300 013746 001152          MOV $NULL, -(SP) ;; GET # OF FILLER CHARS. NEEDED
4650 ;; AND THE NULL CHAR.
4651 026304 105366 000001          7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
4652 026310 002770          BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
4653 026312 004737 026350          JSR PC, $TYPEC ;; GO TYPE A NULL
4654 026316 105337 026414          DECB $CHARCNT ;; DO NOT COUNT AS A COUNT
4655 026322 000770          BR 7$ ;; LOOP
4656

```

```

4657 ;HORIZONTAL TAB PROCESSOR
4658
4659 026324 112716 000040 85: MOVB #40,(SP) ;:REPLACE TAB WITH SPACE
4660 026330 004737 026350 95: TSR PC,$TYPEC ;:TYPE A SPACE
4661 026334 132737 000007 026414 BITB #7,$CHARCNT ;:BRANCH IF NOT AT
4662 026342 001372 BNE 95 ;:TAB STOP
4663 026344 005726 TST (SP)+ ;:POP SPACE OFF STACK
4664 026346 000726 BR 25 ;:GET NEXT CHARACTER
4665 026350 105777 152572 $TYPEC: TSTB $STPB ;:WAIT UNTIL PRINTER IS READY
4666 026354 100375 BPL $TYPEC
4667 026356 116677 000002 152564 MOVB 2(SP),$STPB ;:LOAD CHAR TO BE TYPED INTO DATA REG.
4668 026364 122766 000015 000002 CMPB #15,2(SP) ;:BRANCH IF
4669 026372 001003 BNE 15 ;:NOT (CR)
4670 026374 105037 026414 CLRB $CHARCNT ;:
4671 026400 000406 BR $TYPEX ;:EXIT
4672 026402 122766 000012 000002 15: CMPB #12,2(SP) ;:BRANCH IF
4673 026410 002002 BGE $TYPEX ;:<LF>
4674 026412 105227 INCB (PC)+ ;:INC SPACE
4675 026414 000000 $CHARCNT: WORD 0 ;:COUNT
4676 026416 000207 $TYPEX: RTS PC
4677 ;: EQUATES
4678 000011 †HT=11
4679 000200 TCRLF=200
4680
4681
4682 ;*****
4683
4684 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
4685
4686 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
4687 ;*OCTAL (ASCII) NUMBER AND TYPE IT.
4688 ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
4689 ;*CALL:
4690 ;* MOV NUM,-(SP) ;:NUMBER TO BE TYPED
4691 ;* TYPOS ;:CALL FOR TYPEOUT
4692 ;* .BYTE N ;:N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
4693 ;* .BYTE M ;:M=1 OR 0
4694 ;* ;:1=TYPE LEADING ZEROS
4695 ;* ;:0=SUPPRESS LEADING ZEROS
4696 ;*
4697 ;*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
4698 ;*$TYPOS OR $TYPOC
4699 ;*CALL:
4700 ;* MOV NUM,-(SP) ;:NUMBER TO BE TYPED
4701 ;* TYPON ;:CALL FOR TYPEOUT
4702 ;*
4703 ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
4704 ;*CALL:
4705 ;* MOV NUM,-(SP) ;:NUMBER TO BE TYPED
4706 ;* TYPOC ;:CALL FOR TYPEOUT
4707 ;*
4708 026420 017646 000000 $TYPOS: MOV 2(SP),-(SP) ;:PICKUP THE MODE
4709 026424 116637 000001 026643 MOVB 1(SP),$OFILL ;:LOAD ZERO FILL SWITCH
4710 026432 112637 026645 MOVB (SP)+,$SOMODE+1 ;:NUMBER OF DIGITS TO TYPE
    
```

```

4711 026436 062716 000002      ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
4712 026442 000406      BR       $TYPON
4713 026444 112737 000001 026643 $TYPON: MOV    #1,$OFILL      ;;SET THE ZERO FILL SWITCH
4714 026452 112737 000006 026645      MOV    #6,$SOMODE+1  ;;SET FOR SIX(6) DIGITS
4715 026460 112737 000005 026642 $TYPON: MOV    #5,$SOCNT  ;;SET THE ITERATION COUNT
4716 026466 010346      MOV    R3,-(SP)     ;;SAVE R3
4717 026470 010446      MOV    R4,-(SP)     ;;SAVE R4
4718 026472 010546      MOV    R5,-(SP)     ;;SAVE R5
4719 026474 113704 026645      MOV    $SOMODE+1,R4  ;;GET THE NUMBER OF DIGITS TO TYPE
4720 026500 005404      NEG    R4
4721 026502 062704 000006      ADD    #6,R4        ;;SUBTRACT IT FROM MAX. ALLOWED
4722 026506 110437 026644      MOV    R4,$SOMODE   ;;SAVE IT FOR USE
4723 026512 113704 026643      MOV    $OFILL,R4    ;;GET THE ZERO FILL SWITCH
4724 026516 016605 000012      MOV    12(SP),R5    ;;PICKUP THE INPUT NUMBER
4725 026522 005003      CLR    R3           ;;CLEAR THE OUTPUT WORD
4726 026524 006105      1$:    ROL    R5     ;;ROTATE MSB INTO "C"
4727 026526 000404      BR     3$          ;;GO DO MSB
4728 026530 006105      2$:    ROL    R5     ;;FORM THIS DIGIT
4729 026532 006105      ROL    R5
4730 026534 006105      ROL    R5
4731 026536 010503      MOV    R5,R3
4732 026540 006103      3$:    ROL    R3     ;;GET LSB OF THIS DIGIT
4733 026542 105337 026644      DECB  $SOMODE       ;;TYPE THIS DIGIT?
4734 026546 100016      BPL   7$           ;;BR IF NO
4735 026550 042703 177770      BIC   #177770,R3   ;;GET RID OF JUNK
4736 026554 001002      BNE   4$           ;;TEST FOR 0
4737 026556 005704      TST  R4           ;;SUPPRESS THIS 0?
4738 026560 001403      BEQ  5$           ;;BR IF YES
4739 026562 005204      4$:    INC   R4     ;;DON'T SUPPRESS ANYMORE 0'S
4740 026564 052703 000060      BIS   #'0,R3      ;;MAKE THIS DIGIT ASCII
4741 026570 052703 000040      5$:    BIS   #' ,R3  ;;MAKE ASCII IF NOT ALREADY
4742 026574 110337 026640      MOV    R3,R5       ;;SAVE FOR TYPING
4743 026600 104400 026640      TYPE  #8$         ;;GO TYPE THIS DIGIT
4744 026604 105337 026642      7$:    DECB  $SOCNT  ;;COUNT BY 1
4745 026610 003347      BGT  2$           ;;BR IF MORE TO DO
4746 026612 002402      BLT  6$           ;;BR IF DONE
4747 026614 005204      INC  R4           ;;INSURE LAST DIGIT ISN'T A BLANK
4748 026616 000744      BR   2$          ;;GO DO THE LAST DIGIT
4749 026620 012605      6$:    MOV   (SP)+,R5  ;;RESTORE R5
4750 026622 012604      MOV   (SP)+,R4     ;;RESTORE R4
4751 026624 012603      MOV   (SP)+,R3     ;;RESTORE R3
4752 026626 016666 000002 000004      MOV   2(SP),4(SP)  ;;SET THE STACK FOR RETURNING
4753 026634 012616      MOV   (SP)+,(SP)
4754 026636 000002      RTI
4755 026640 000      8$:    .BYTE 0        ;;RETURN
4756 026641 000      .BYTE 0        ;;STORAGE FOR ASCII DIGIT
4757 026642 000      $SOCNT: .BYTE 0  ;;TERMINATOR FOR TYPE ROUTINE
4758 026643 000      $OFILL: .BYTE 0  ;;OCTAL DIGIT COUNTER
4759 026644 000000      $SOMODE: .WORD 0  ;;ZERO FILL SWITCH
4760
4761
4762
4763
4764

```

```

*****
.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

F13

```

4765                                     ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
4766                                     ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
4767                                     ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
4768                                     ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
4769                                     ;*REPLACED WITH SPACES.
4770                                     ;*CALL:
4771                                     ;*   MOV     NUM,-(SP)           ;;PUT THE BINARY NUMBER ON THE STACK
4772                                     ;*   TYPDS           ;;GO TO THE ROUTINE
4773
4774 026646                                $TYPDS:
4775 026646 010046                        MOV     R0,-(SP)           ;;PUSH R0 ON STACK
4776 026650 010146                        MOV     R1,-(SP)           ;;PUSH R1 ON STACK
4777 026652 010246                        MOV     R2,-(SP)           ;;PUSH R2 ON STACK
4778 026654 010346                        MOV     R3,-(SP)           ;;PUSH R3 ON STACK
4779 026656 010546                        MOV     R5,-(SP)           ;;PUSH R5 ON STACK
4780 026660 012746 020200                MOV     #20200,-(SP)      ;;SET BLANK SWITCH AND SIGN
4781 026664 016605 000020                MOV     20(SP),R5         ;;GET THE INPUT NUMBER
4782 026670 100004                        BPL     1$                ;;BR IF INPUT IS POS.
4783 026672 005405                        NEG     R5                ;;MAKE THE BINARY NUMBER POS.
4784 026674 112766 000055 000001        MOVB   #'-,1(SP)         ;;MAKE THE ASCII NUMBER NEG.
4785 026702 005000                        1$:   CLR     R0           ;;ZERO THE CONSTANTS INDEX
4786 026704 012703 027062                MOV     #SDBLK,R3        ;;SETUP THE OUTPUT POINTER
4787 026710 112723 000040                MOVB   #'',(R3)+         ;;SET THE FIRST CHARACTER TO A BLANK
4788 026714 005002                        2$:   CLR     R2           ;;CLEAR THE BCD NUMBER
4789 026716 016001 027052                MOV     $DTBL(R0),R1     ;;GET THE CONSTANT
4790 026722 160105                        3$:   SUB     R1,R5         ;;FORM THIS BCD DIGIT
4791 026724 002402                        BLT     4$                ;;BR IF DONE
4792 026726 005202                        INC     R2                ;;INCREASE THE BCD DIGIT BY 1
4793 026730 000774                        BR      3$
4794 026732 060105                        4$:   ADD     R1,R5         ;;ADD BACK THE CONSTANT
4795 026734 005702                        TST     R2                ;;CHECK IF BCD DIGIT=0
4796 026736 001002                        BNE     5$                ;;FALL THROUGH IF 0
4797 026740 105716                        TSTB   (SP)              ;;STILL DOING LEADING 0'S?
4798 026742 100407                        BMI     7$                ;;BR IF YES
4799 026744 106316                        5$:   ASLB   (SP)         ;;MSD?
4800 026746 103003                        BCC     6$                ;;BR IF NO
4801 026750 116663 000001 177777        MOVB   1(SP),-1(R3)      ;;YES--SET THE SIGN
4802 026756 052702 000060                BIS     #'0,R2           ;;MAKE THE BCD DIGIT ASCII
4803 026762 052702 000040                6$:   BIS     #' ,R2       ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
4804 026766 110223                        MOVB   R2,(R3)+         ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
4805 026770 005720                        TST     (R0)+           ;;JUST INCREMENTING
4806 026772 020027 000010                CMP     R0,#10          ;;CHECK THE TABLE INDEX
4807 026776 002746                        BLT     2$                ;;GO DO THE NEXT DIGIT
4808 027000 003002                        BGT     8$                ;;GO TO EXIT
4809 027002 010502                        MOV     R5,R2           ;;GET THE LSD
4810 027004 000764                        BR      6$                ;;GO CHANGE TO ASCII
4811 027006 105726                        8$:   TSTB   (SP)+       ;;WAS THE LSD THE FIRST NON-ZERO?
4812 027010 100003                        BPL     9$                ;;BR IF NO
4813 027012 116663 177777 177776        MOVB   -1(SP),-2(R3)     ;;YES--SET THE SIGN FOR TYPING
4814 027020 105013                        9$:   CLRB   (R3)         ;;SET THE TERMINATOR
4815 027022 012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
4816 027024 012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
4817 027026 012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
4818 027030 012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1

```

```

4819 027032 012600          MOV      (SP)+,RO      ;; POP STACK INTO RO
4820 027034 104400 027062  TYPE      $OBLK      ;; NOW TYPE THE NUMBER
4821 027040 016666 000002 000004  MOV      2(SP),4(SP)  ;; ADJUST THE STACK
4822 027046 012616          MOV      (SP)+,(SP)
4823 027050 000002          RTI                          ;; RETURN TO USER
4824 027052 023420          SOTBL: 10000.
4825 027054 001750          1000.
4826 027056 000144          100.
4827 027060 000012          10.
4828 027062 000004          $OBLK: .BLKW 4
4829
4830 ;*****
4831
4832 .SBTTL TRAP DECODER
4833
4834 ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
4835 ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
4836 ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
4837 ;*GO TO THAT ROUTINE.
4838
4839 027072 010046          $TRAP: MOV      RO,-(SP)      ;; SAVE RO
4840 027074 016600 000002  MOV      2(SP),RO      ;; GET TRAP ADDRESS
4841 027100 005740          TST      -(RO)        ;; BACKUP BY 2
4842 027102 111000          MOVB    (RO),RO      ;; GET RIGHT BYTE OF TRAP
4843 027104 006300          ASL     RO           ;; POSITION FOR INDEXING
4844 027106 016000 027114  MOV      $TRPAD(RO),RO ;; INDEX TO TABLE
4845 027112 000200          RTS      RO          ;; GO TO ROUTINE
4846
4847
4848 .SBTTL TRAP TABLE
4849
4850 ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
4851 ;*BY THE "TRAP" INSTRUCTION.
4852
4853 :          ROUTINE
4854 :          -----
4855 $TRPAD:
4856 027114 026204          $TYPE   ;; CALL=TYPE      TRAP+0(104400) TTY TYPEOUT ROUTINE
4857 027116 026444          $TYPOC  ;; CALL=TYPOC    TRAP+1(104401) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
4858 027120 026420          $TYPOS  ;; CALL=TYPOS    TRAP+2(104402) TYPE OCTAL NUMBER (NO LEADING ZEROS)
4859 027122 026460          $TYPON  ;; CALL=TYPON    TRAP+3(104403) TYPE OCTAL NUMBER (AS PER LAST CALL)
4860 027124 026646          $TYPDS  ;; CALL=TYPDS    TRAP+4(104404) TYPE DECIMAL NUMBER (WITH SIGN)
4861
4862 ;*****
4863
4864 .SBTTL POWER DOWN AND UP ROUTINES
4865
4866 :POWER DOWN ROUTINE
4867 027126 012737 027262 000024 $PWRDN: MOV      #SILLUP,2#PWRVEC ;; SET FOR FAST UP
4868 027134 012737 000340 000026  MOV      #340,2#PWRVEC+2 ;; PRIO:7
4869 027142 010046          MOV      RO,-(SP)    ;; PUSH RO ON STACK
4870 027144 010146          MOV      R1,-(SP)   ;; PUSH R1 ON STACK
4871 027146 010246          MOV      R2,-(SP)   ;; PUSH R2 ON STACK
4872 027150 010346          MOV      R3,-(SP)   ;; PUSH R3 ON STACK

```

```

4873 027152 010446          MOV      R4,-(SP)          ;; PUSH R4 ON STACK
4874 027154 010546          MOV      R5,-(SP)          ;; PUSH R5 ON STACK
4875 027156 010637 027266    MOV      SP,$SAVR6         ;; SAVE SP
4876 027162 012737 027174 000024  MOV      $SPWRUP,$PWRVEC   ;; SET UP VECTOR
4877 027170 000000          HALT
4878 027172 000776          BR       .-2              ;; HANG UP
4879
4880
4881 027174 013706 027266          :POWER UP ROUTINE
4882 027200 005037 027266    $PWRUP: MOV      $SAVR6,SP   ;; GET SP
4883 027204 005237 027266    CLR      $SAVR6           ;; WAIT LOOP FOR THE TTY
4884 027210 001375          IS:   INC      $SAVR6     ;; WAIT FOR THE INC
4885 027212 012605          BNE     IS               ;; OF WORD
4886 027214 012604          MOV     (SP)+,R5         ;; POP STACK INTO R5
4887 027216 012603          MOV     (SP)+,R4         ;; POP STACK INTO R4
4888 027220 012602          MOV     (SP)+,R3         ;; POP STACK INTO R3
4889 027222 012601          MOV     (SP)+,R2         ;; POP STACK INTO R2
4890 027224 012600          MOV     (SP)+,R1         ;; POP STACK INTO R1
4891 027226 012737 027126 000024  MOV     $SPWRDN,$PWRVEC   ;; SET UP THE POWER DOWN VECTOR
4892 027234 012737 000340 000026  MOV     #340,$PWRVEC+2   ;; PRIO:7
4893 027242 104400          TYPE
4894 027244 030622          $PWRMG: .WORD  POWMES    ;; REPORT THE POWER FAILURE
4895 027246 012716          MOV     (PC)+,(SP)      ;; POWER FAIL MESSAGE POINTER
4896 027250 027270          $PWRAD: .WORD  DISPATCH  ;; RESTART AT DISPATCH
4897 027252 042766 000020 000002  BIC     #20,2(SP)       ;; RESTART ADDRESS
4898 027260 000002          RTI
4899 027262 000000          $ILLUP: HALT
4900 027264 000776          BR       .-2              ;; CLEAR "I" BIT
4901 027266 000000          $SAVR6: 0
4902
4903 027270 000177 000000          DISPATCH: JMP      @RETURN   ;; THE POWER UP SEQUENCE WAS STARTED
4904
4905
4906 027274 000000          RETURN:  .WORD  0        ;; BEFORE THE POWER DOWN WAS COMPLETE
4907
4908
4909
4910
4911
4912
4913

```

```

;; ATTEMPT TO RESTART LAST SECTION
;; BEING RUN BEFORE POWER FAILURE
;; OCCURRED
;; THIS LOCATION HOLDS THE STARTING
;; ADDRESS OF THE SECTION BEING RUN
;; BEFORE THE POWER FAILURE OCCURRED
;; EITHER:
;; BEGIN:
;; ERCD12:
;; XLOOP:
;; CKSAME:

```


4968 027436 000000
 4969 027440 010400
 4970 027442 020200
 4971 027444 030100
 4972 027446 040040
 4973 027450 050020
 4974 027452 060010
 4975 027454 070004
 4976 027456 000002
 4977 027460 000001
 4978 027462 020202
 4979 027464 030102
 4980 027466 040042
 4981 027470 050022
 4982 027472 060012
 4983 027474 070006
 4984 027476 004000
 4985 027500 014400
 4986 027502 024200
 4987 027504 034100
 4988 027506 044040
 4989 027510 054020
 4990 027512 064010
 4991 027514 074004
 4992 027516 004002
 4993 027520 004001
 4994 027522 024202
 4995 027524 034102
 4996 027526 044042
 4997 027530 054022
 4998 027532 064012
 4999 027534 074006

0000
 10400
 20200
 30100
 40040
 50020
 60010
 70004
 0002
 0001
 20202
 30102
 40042
 50022
 60012
 70006
 4000
 14400
 24200
 34100
 44040
 54020
 64010
 74004
 4002
 4001
 24202
 34102
 44042
 54022
 64012
 74006
 ALPEND: 74006

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

SPACE
 BLANK

:ALPHANUMERIC DECK DATA TABLE
 :THE VALUE IS THE ENCODED FORM OF THE DATA

5000
 5001
 5002
 5003
 5004 027536 200
 5005 027537 201
 5006 027540 202
 5007 027541 203
 5008 027542 204
 5009 027543 205
 5010 027544 206
 5011 027545 207
 5012 027546 210
 5013 027547 220
 5014 027550 212
 5015 027551 213
 5016 027552 214
 5017 027553 215
 5018 027554 216
 5019 027555 217
 5020 027556 100
 5021 027557 101

ALPCDP: .BYTE 200
 .BYTE 201
 .BYTE 202
 .BYTE 203
 .BYTE 204
 .BYTE 205
 .BYTE 206
 .BYTE 207
 .BYTE 210
 .BYTE 220
 .BYTE 212
 .BYTE 213
 .BYTE 214
 .BYTE 215
 .BYTE 216
 .BYTE 217
 .BYTE 100
 .BYTE 101

COLUMN ASCII PUNCH
 1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18

5076	027646	210
5077	027647	220
5078	027650	212
5079	027651	213
5080	027652	214
5081	027653	215
5082	027654	216
5083	027655	217

.BYTE	210
.BYTE	220
.BYTE	212
.BYTE	213
.BYTE	214
.BYTE	215
.BYTE	216
.BYTE	217

:73	H	12
:74	I	12
:75	(12
:76	:	12
:77	(12
:78	(12
:79	+	12
:80	:	12

ALPENP: .BYTE 217
:BINARY DECK DATA TABLE

5084		
5085		
5086		
5087	027 56	000000
5088	027660	000001
5089	027662	000002
5090	027664	070004
5091	027666	060010
5092	027670	050020
5093	027672	040040
5094	027674	030100
5095	027676	020200
5096	027700	010400
5097	027702	001000
5098	027704	002000
5099	027706	004000
5100	027710	171111
5101	027712	172222
5102	027714	173333
5103	027716	174444
5104	027720	175555
5105	027722	176666
5106	027724	177777
5107	027726	061010
5108	027730	161212
5109	027732	171313
5110	027734	171414
5111	027736	171515
5112	027740	171616
5113	027742	171717
5114	027744	052020
5115	027746	172121
5116	027750	172323
5117	027752	172424
5118	027754	172525
5119	027756	172626
5120	027760	172727
5121	027762	173030
5122	027764	173131
5123	027766	173232
5124	027770	173434
5125	027772	173535
5126	027774	173636
5127	027776	173737
5128	030000	044040
5129	030002	174141

BINCD: 0
1
2
70004
60010
50020
40040
30100
20200
10400
1000
2000
4000
171111
172222
173333
174444
175555
176666
177777
61010
161212
171313
171414
171515
171616
171717
52020
172121
172323
172424
172525
172626
172727
173030
173131
173232
173434
173535
173636
173737
44040
174141

COLUMN	PUNCH
:1	BLANK
: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	

5130	030004	164242	164242	44
5131	030006	174343	174343	45
5132	030010	174545	174545	46
5133	030012	174646	174646	47
5134	030014	174747	174747	48
5135	030016	165050	165050	49
5136	030020	175151	175151	50
5137	030022	165252	165252	51
5138	030024	175353	175353	52
5139	030026	175454	175454	53
5140	030030	175656	175656	54
5141	030032	175757	175757	55
5142	030034	156060	156060	56
5143	030036	176161	176161	57
5144	030040	176262	176262	58
5145	030042	176363	176363	59
5146	030044	176464	176464	60
5147	030046	176565	176565	61
5148	030050	176767	176767	62
5149	030052	177070	177070	63
5150	030054	177171	177171	64
5151	030056	177272	177272	65
5152	030060	177373	177373	66
5153	030062	177474	177474	67
5154	030064	177575	177575	68
5155	030066	177676	177676	69
5156	030070	030101	30101	70
5157	030072	020202	20202	71
5158	030074	130303	130303	72
5159	030076	170404	170404	73
5160	030100	170505	170505	74
5161	030102	170606	170606	75
5162	030104	170707	170707	76
5163	030106	163210	163210	77
5164	030110	170123	170123	78
5165	030112	177654	177654	79
5166	030114	174567	174567	80

BINEND:

; BINARY DECK DATA TABLE
; THE VALUE IS THE ENCODED VALUE, WHICH ORS THE OCTAL REPRESENTATION OF
; ROWS ONE THRU SEVEN

			COLUMN	ASCII	PUNCH
5172	030116	000	1	SPACE	BLANK
5173	030117	020	2	9	9
5174	030120	010	3	8	8
5175	030121	007	4	7	7
5176	030122	006	5	6	6
5177	030123	005	6	5	5
5178	030124	004	7	4	4
5179	030125	003	8	3	3
5180	030126	002	9	2	2
5181	030127	001	10	1	1
5182	030130	040	11	0	0
5183	030131	100	12	-	11

BINCDP:

.BYTE	0
.BYTE	20
.BYTE	10
.BYTE	7
.BYTE	6
.BYTE	5
.BYTE	4
.BYTE	3
.BYTE	2
.BYTE	1
.BYTE	40
.BYTE	100

5184	030132	200	.BYTE	200	13		
5185	030133	067	.BYTE	67	14		
5186	030134	117	.BYTE	117	15		
5187	030135	177	.BYTE	177	16		
5188	030136	207	.BYTE	207	17		
5189	030137	267	.BYTE	267	18		
5190	030140	317	.BYTE	317	19		
5191	030141	377	.BYTE	377	20		
5192	030142	046	.BYTE	46	21		
5193	030143	056	.BYTE	56	22		
5194	030144	077	.BYTE	77	23		
5195	030145	047	.BYTE	47	24		
5196	030146	067	.BYTE	67	25		
5197	030147	057	.BYTE	57	26		
5198	030150	077	.BYTE	77	27		
5199	030151	105	.BYTE	105	28		
5200	030152	127	.BYTE	127	29		
5201	030153	137	.BYTE	137	30		
5202	030154	107	.BYTE	107	31		
5203	030155	127	.BYTE	127	32		
5204	030156	117	.BYTE	117	33		
5205	030157	137	.BYTE	137	34		
5206	030160	147	.BYTE	147	35		
5207	030161	167	.BYTE	167	36		
5208	030162	157	.BYTE	157	37		
5209	030163	147	.BYTE	147	38		
5210	030164	167	.BYTE	167	39		
5211	030165	157	.BYTE	157	40		
5212	030166	177	.BYTE	177	41		
5213	030167	204	.BYTE	204	42		
5214	030170	227	.BYTE	227	43		
5215	030171	216	.BYTE	216	44		
5216	030172	237	.BYTE	237	45		
5217	030173	227	.BYTE	227	46		
5218	030174	217	.BYTE	217	47		
5219	030175	237	.BYTE	237	48		
5220	030176	246	.BYTE	246	49		
5221	030177	267	.BYTE	267	50		
5222	030200	256	.BYTE	256	51		
5223	030201	277	.BYTE	277	52		
5224	030202	247	.BYTE	247	53		
5225	030203	257	.BYTE	257	54		
5226	030204	277	.BYTE	277	55		
5227	030205	305	.BYTE	305	56		
5228	030206	327	.BYTE	327	57		
5229	030207	317	.BYTE	317	58		
5230	030210	337	.BYTE	337	59		
5231	030211	307	.BYTE	307	60		
5232	030212	327	.BYTE	327	61		
5233	030213	337	.BYTE	337	62		
5234	030214	347	.BYTE	347	63		
5235	030215	367	.BYTE	367	64		
5236	030216	357	.BYTE	357	65		
5237	030217	377	.BYTE	377	66		

8

12

5238	030220	347				.BYTE	347	:67
5239	030221	367				.BYTE	367	:68
5240	030222	357				.BYTE	357	:69
5241	030223	023				.BYTE	23	:70
5242	030224	012				.BYTE	12	:71
5243	030225	033				.BYTE	33	:72
5244	030226	007				.BYTE	7	:73
5245	030227	027				.BYTE	27	:74
5246	030230	017				.BYTE	17	:75
5247	030231	037				.BYTE	37	:76
5248	030232	146				.BYTE	146	:77
5249	030233	037				.BYTE	37	:78
5250	030234	347				.BYTE	347	:79
5251	030235	237				.BYTE	237	:80
5252								
5253	030236	020040	020040	040		.ASCII	/ / /	
5254	030243	040	000040			SPACE:	.ASCIZ / / /	
5255	030246	005012	012				.ASCII <12><12><12>	
5256	030251	015	000012			CRLF:	.ASCIZ <15><12>	
5257								
5258	030254	005015	046412	044501		STMS:	.ASCIZ <15><12><12>"MAINDEC-11-DZCDB REV. A"<15><12>	
5259	030262	042116	041505	030455				
5260	030270	026461	055104	042103				
5261	030276	020102	020040	051040				
5262	030304	053105	020056	006501				
5263	030312	000012						
5264								
5265	030314	005015	052123	051101		STADDR:	.ASCII <15><12>"STARTING ADDRESSES ARE:"	
5266	030322	044524	043516	040440				
5267	030330	042104	042522	051523				
5268	030336	051505	040440	042522				
5269	030344	072						
5270	030345	015	031012	030060			.ASCII <15><12>"200 = INSTRUCTION AND DATA TEST"	
5271	030352	036440	044440	051516				
5272	030360	051124	041523	044524				
5273	030366	047117	040440	042116				
5274	030374	042040	052101	020101				
5275	030402	042524	052123					
5276	030406	005015	030462	020060			.ASCII <15><12>"210 = ERROR FUNCTION TEST (M-1000/M-200)"	
5277	030414	020075	051105	047522				
5278	030422	020122	052506	041516				
5279	030430	044524	047117	052040				
5280	030436	051505	020124	046450				
5281	030444	030455	030060	027460				
5282	030452	026515	030062	024460				
5283	030460	005015	031062	020060			.ASCII <15><12>"220 = SINGLE TEST LOOP"	
5284	030466	020075	044523	043516				
5285	030474	042514	052040	051505				
5286	030502	020124	047514	050117				
5287	030510	005015	032062	020060			.ASCII <15><12>"240 = READ SINGLE DATA PATTERN TEST"	
5288	030516	020075	042522	042101				
5289	030524	051440	047111	046107				
5290	030532	020105	040504	040524				
5291	030540	050040	052101	042524				

5292	030546	047122	052040	051505
5293	030554	124		
5294	030555	015	031012	030065
5295	030563	036440	042440	051122
5296	030570	051117	043040	047125
5297	030576	052103	047511	020116
5298	030604	042524	052123	024040
5299	030612	026515	031061	030060
5300	030620	000051		
5301				
5302	030622	005015	050012	053517
5303	030630	051105	043040	044501
5304	030636	052514	042522	047440
5305	030644	041503	051125	042522
5306	030652	020104	020055	044527
5307	030660	046114	040440	052124
5308	030666	046505	052120	052040
5309	030674	117		
5310	030675	015	051012	051505
5311	030702	040524	052122	051440
5312	030710	041505	044524	047117
5313	030716	047440	020106	051120
5314	030724	043517	040522	020115
5315	030732	047506	046522	051105
5316	030740	054514	044440	020116
5317	030746	051525	006505	000012
5318				
5319	030754	005015	047105	042524
5320	030762	044522	043516	046040
5321	030770	043517	041511	052040
5322	030776	051505	051524	000
5323				
5324	031003	015	042412	052116
5325	031010	051105	047111	020107
5326	031016	040504	040524	052040
5327	031024	051505	051524	000
5328				
5329	031031	015	042412	052116
5330	031036	051105	047111	020107
5331	031044	030515	030062	020060
5332	031052	051105	047522	020122
5333	031060	052506	041516	044524
5334	031066	047117	052040	051505
5335	031074	051524	000	
5336				
5337	031077	015	042412	052116
5338	031104	051105	047111	020107
5339	031112	030515	030060	027460
5340	031120	031115	030060	042440
5341	031126	051122	051117	043040
5342	031134	047125	052103	047511
5343	031142	020116	042524	052123
5344	031150	000123		
5345				

.ASCIZ (15)(12)"250 = ERROR FUNCTION TEST (M-1200)"

POWRES: .ASCII (15)(12)(12)"POWER FAILURE OCCURRED - WILL ATTEMPT TO"

.ASCIZ (15)(12)"RESTART SECTION OF PROGRAM FORMERLY IN USE"(15)(12)

MLOGIC: .ASCIZ (15)(12)"ENTERING LOGIC TESTS"

MDATA: .ASCIZ (15)(12)"ENTERING DATA TESTS"

M1200E: .ASCIZ (15)(12)"ENTERING M1200 ERROR FUNCTION TESTS"

M1000E: .ASCIZ (15)(12)"ENTERING M1000/M200 ERROR FUNCTION TESTS"

5346	031152	005015	047105	042524	ML00P: .ASCIZ <15><12>"ENTERING A LOOP ON TEST SELECTED BY USER"
5347	031160	044522	043516	040440	
5348	031166	046040	047517	020120	
5349	031174	047117	052040	051505	
5350	031202	020124	042523	042514	
5351	031210	052103	042105	041040	
5352	031216	020131	051525	051105	
5353	031224	000			
5354					
5355	031225	015	042412	052116	MPATS: .ASCIZ <15><12>"ENTERING SINGLE DATA PATTERN TESTING"
5356	031232	051105	047111	020107	
5357	031240	044523	043516	042514	
5358	031246	042040	052101	020101	
5359	031254	040520	052124	051105	
5360	031262	020116	042524	052123	
5361	031270	047111	000107		
5362					
5363	031274	005015	051120	051505	MSG1: .ASCIZ <15><12>/PRESS CARD READER 'RESET'/'
5364	031302	020123	040503	042122	
5365	031310	051040	040505	042504	
5366	031316	020122	051047	051505	
5367	031324	052105	000047		
5368	031330	005015	044124	047105	MSG2: .ASCIZ <15><12>/THEN HIT 'CONTINUE' ON THE CONSOLE/'
5369	031336	044040	052111	023440	
5370	031344	047503	052116	047111	
5371	031352	042525	020047	047117	
5372	031360	052040	042510	041440	
5373	031366	047117	047523	042514	
5374	031374	000			
5375	031375	015	050012	042522	MSG3: .ASCIZ <15><12>/PRESS CARD READER 'STOP'/'
5376	031402	051523	041440	051101	
5377	031410	020104	042522	042101	
5378	031416	051105	023440	052123	
5379	031424	050117	000047		
5380	031430	005015	044124	020105	MSG4: .ASCIZ <15><12>/THE INTERRUPT LEVEL WAS /
5381	031436	047111	042524	051122	
5382	031444	050125	020124	042514	
5383	031452	042526	020114	040527	
5384	031460	020123	000		
5385	031463	015	051012	046505	MSG5: .ASCIZ <15><12>/REMOVE ALL CARDS FROM THE INPUT HOPPER.
5386	031470	053117	020105	046101	
5387	031476	020114	040503	042122	
5388	031504	020123	051106	046517	
5389	031512	052040	042510	044440	
5390	031520	050116	052125	044040	
5391	031526	050117	042520	000122	
5392	031534	005015	042522	052123	MSG6: .ASCIZ <15><12>/RESTORE CARDS TO THE INPUT HOPPER
5393	031542	051117	020105	040503	
5394	031550	042122	020123	047524	
5395	031556	052040	042510	044440	
5396	031564	050116	052125	044040	
5397	031572	050117	042520	000122	
5398	031600	005015	052520	046114	MSG7: .ASCII <15><12>/PULL OUTPUT STACKER PRESSURE ARM DOWN
5399	031606	047440	052125	052520	

5400	031614	020124	052123	041501
5401	031622	042513	020122	051120
5402	031630	051505	052523	042522
5403	031636	040440	046522	042040
5404	031644	053517	020116	
5405	031650	047125	044524	020114
5406	031656	047510	050120	051105
5407	031664	041440	042510	045503
5408	031672	046040	043511	052110
5409	031700	000123		
5410	031702	005015	047510	042114
5411	031710	042040	053517	020116
5412	031716	044124	020105	053523
5413	031724	052111	044103	052440
5414	031732	042116	051105	052040
5415	031740	042510	041440	050101
5416	031746	047440	020106	044124
5417	031754	020105	047111	052520
5418	031762	020124		
5419	031764	047510	050120	051105
5420	031772	000		
5421	031773	015	051412	044514
5422	032000	042504	040440	041440
5423	032006	051101	020104	051106
5424	032014	046517	052040	042510
5425	032022	047440	052125	052520
5426	032030	020124	047510	050120
5427	032036	051105	040440	047502
5428	032044	052125	044040	046101
5429	032052	020106	047101	044440
5430	032060	041516	020110	
5431	032064	005015	020040	040502
5432	032072	045503	044440	052116
5433	032100	020117	044124	020105
5434	032106	042522	042101	044040
5435	032114	040505	026104	041040
5436	032122	047514	045503	047111
5437	032130	020107	044124	020105
5438	032136	044120	052117	020117
5439	032144	042503	046114	
5440	032150	005015	047516	042524
5441	032156	020072	047523	042515
5442	032164	041440	051101	020104
5443	032172	042522	042101	051105
5444	032200	046440	042117	046105
5445	032206	020123	040515	020131
5446	032214	040510	042526	040440
5447	032222	046040	051101	042507
5448	032230	005015	047522	046114
5449	032236	051105	041040	047514
5450	032244	045503	047111	020107
5451	032252	041501	042503	051523
5452	032260	052040	020117	044124
5453	032266	020105	044120	052117

.ASCIZ /UNTIL HOPPER CHECK LIGHTS/

MSG8: .ASCII <15><12>/HOLD DOWN THE SWITCH UNDER THE CAP OF THE INPUT /

.ASCIZ /HOPPER/

MSG9: .ASCII <15><12>'SLIDE A CARD FROM THE OUTPUT HOPPER ABOUT HALF AN INCH /

.ASCII <15><12>/ BACK INTO THE READ HEAD, BLOCKING THE PHOTO CELL/

.ASCII <15><12>'NOTE: SOME CARD READER MODELS MAY HAVE A LARGE'

.ASCII <15><12>'ROLLER BLOCKING ACCESS TO THE PHOTOCELL. IN'

5454	032274	041517	046105	026114	
5455	032302	044440	116		
5456	032305	015	053412	044510	.ASCII <15><12>"WHICH CASE THE PHOTOCCELL CAN BE BLOCKED"
5457	032312	044103	041440	051501	
5458	032320	020105	044124	020105	
5459	032326	044120	052117	041517	
5460	032334	046105	020114	040503	
5461	032342	020116	042502	041040	
5462	032350	047514	045503	042105	
5463	032356	005015	054502	052040	.ASCII <15><12>"BY TEARING OFF A PIECE OF CARD AND SLIPPING"
5464	032364	040505	044522	043516	
5465	032372	047440	043106	040440	
5466	032400	050040	042511	042503	
5467	032406	047440	020106	040503	
5468	032414	042122	040440	042116	
5469	032422	051440	044514	050120	
5470	032430	047111	107		
5471	032433	015	044412	020124	.ASCIZ <15><12>"IT IN IN FRONT OF THE PHOTOCCELL"
5472	032440	047111	044440	020116	
5473	032446	051106	047117	020124	
5474	032454	043117	052040	042510	
5475	032462	050040	047510	047524	
5476	032470	042503	046114	000	
5477	032475	015	051012	046505	MSG10: .ASCIZ <15><12>/REMOVE JAMMED CARD/
5478	032502	053117	020105	040512	
5479	032510	046515	042105	041440	
5480	032516	051101	000104		
5481	032522	005015	047510	042114	MSG11: .ASCIZ <15><12>/HOLD THE OUTPUT STACKER GATE OPEN. THEN/
5482	032530	052040	042510	047440	
5483	032536	052125	052520	020124	
5484	032544	052123	041501	042513	
5485	032552	020122	040507	042524	
5486	032560	047440	042520	027116	
5487	032566	052040	042510	000116	
5488	032574	005015	046120	041501	MSG12: .ASCII <15><12>/PLACE SPECIAL DARK-LIGHT CHECK CARD ONLY INTO HOPPER.
5489	032602	020105	050123	041505	
5490	032610	040511	020114	040504	
5491	032616	045522	046055	043511	
5492	032624	052110	041440	042510	
5493	032632	045503	041440	051101	
5494	032640	020104	047117	054514	
5495	032646	044440	052116	020117	
5496	032654	047510	050120	051105	
5497	032662	005015	027111	027105	.ASCII <15><12>/I.E. - TEAR A CORNER OFF A CARD AND PLACE IT/
5498	032670	026440	052040	040505	
5499	032676	020122	020101	047503	
5500	032704	047122	051105	047440	
5501	032712	043106	040440	041440	
5502	032720	051101	020104	047101	
5503	032726	020104	046120	041501	
5504	032734	020105	052111		
5505	032740	005015	052101	052040	.ASCIZ <15><12> /AT THE BOTTOM OF THE INPUT STACK.
5506	032746	042510	041040	052117	
5507	032754	047524	020115	043117	

5508	032762	052040	042510	044440					
5509	032770	050116	052125	051440					
5510	032776	040524	045503	000					
5511	033003	015	042012	041505	MSG13:	.ASCIZ	<15><12>/DECK	CARD NUM	CARD COL SHB WAS/
5512	033010	020113	020040	020040					
5513	033016	040503	042122	047040					
5514	033024	046525	020040	020040					
5515	033032	041440	051101	020104					
5516	033040	047503	020114	020040					
5517	033046	044123	020102	020040					
5518	033054	020040	040527	000123					
5519	033062	005015	046101	044120	MSG14:	.ASCIZ	<15><12>/ALPHA /		
5520	033070	020101	000						
5521	033073	015	041012	047111	MSG15:	.ASCIZ	<15><12>/BINARY/		
5522	033100	051101	000131						
5523	033104	005015	044502	020124	MSG16:	.ASCIZ	<15><12>/BIT 15 WAS SET/		
5524	033112	032461	053440	051501					
5525	033120	051440	052105	000					
5526	033125	015	051012	046505	MSG17:	.ASCII	<15><12>/REMEDY THE CONDITION BY RELOADING INPUT HOPPER/		
5527	033132	042105	020131	044124					
5528	033140	020105	047503	042116					
5529	033146	052111	047511	020116					
5530	033154	054502	051040	046105					
5531	033162	040517	044504	043516					
5532	033170	044440	050116	052125					
5533	033176	044040	050117	042520					
5534	033204	122							
5535	033205	015	053412	052111		.ASCII	<15><12>/WITH CARD DECK - PRESS 'RESET' BUTTON/		
5536	033212	020110	040503	042122					
5537	033220	042040	041505	020113					
5538	033226	020055	051120	051505					
5539	033234	020123	051047	051505					
5540	033242	052105	020047	052502					
5541	033250	052124	047117						
5542	033254	005015	047117	041440		.ASCIZ	<15><12>/ON CARD READER AND 'CONTINUE' SWITCH ON CPU PANEL/		
5543	033262	051101	020104	042522					
5544	033270	042101	051105	040440					
5545	033276	042116	023440	047503					
5546	033304	052116	047111	042525					
5547	033312	020047	053523	052111					
5548	033320	044103	047440	020116					
5549	033326	050103	020125	040520					
5550	033334	042516	000114						
5551	033340	005015	040503	042122	MSG18:	.ASCIZ	<15><12>/CARD READER IS OFF-LINE/		
5552	033346	051040	040505	042504					
5553	033354	020122	051511	047440					
5554	033362	043106	046055	047111					
5555	033370	000105							
5556	033372	005015	020040	041440	MSG19:	.ASCIZ	<15><12>/	COL. WAS CARD NUM. ERRORS/	
5557	033400	046117	020056	020040					
5558	033406	040527	020123	020040					
5559	033414	040503	042122	047040					
5560	033422	046525	020056	042440					
5561	033430	051122	051117	000123					

5562	033436	005015	052520	020124
5563	033444	047101	020131	053524
5564	033452	020117	040503	042122
5565	033460	020123	047111	044440
5566	033466	050116	052125	044040
5567	033474	050117	042520	000122
5568	033502	005015	051120	051505
5569	033510	020123	047105	020104
5570	033516	043117	043040	046111
5571	033524	020105	052502	052124
5572	033532	047117	000	
5573	033535	015	053412	042510
5574	033542	020116	051120	047111
5575	033550	044524	043516	051440
5576	033556	047524	051520	050040
5577	033564	052125	023440	040510
5578	033572	052114	020047	047101
5579	033600	020104	051447	047111
5580	033606	046107	020105	052502
5581	033614	020123	054503	046103
5582	033622	023505		
5583	033624	005015	044450	020106
5584	033632	051120	051505	047105
5585	033640	024524	041440	047117
5586	033646	047523	042514	051440
5587	033654	044527	041524	042510
5588	033662	020123	047504	047127
5589	033670	040440	042116	044040
5590	033676	052111		
5591	033700	005015	041447	047117
5592	033706	044524	052516	023505
5593	033714	047440	020116	044124
5594	033722	020105	047503	051516
5595	033730	046117	020105	047125
5596	033736	044524	020114	047117
5597	033744	020105	040503	042122
5598	033752	044440	020123	042522
5599	033760	042101		
5600	033762	005015	044124	047105
5601	033770	050040	052125	052440
5602	033776	020120	044124	020105
5603	034004	047117	020105	051117
5604	034012	052040	047527	051440
5605	034020	044527	041524	042510
5606	034026	020123	047101	020104
5607	034034	044510	124	
5608	034037	015	023412	047503
5609	034044	052116	047111	042525
5610	034052	020047	047117	052040
5611	034060	042510	041440	047117
5612	034066	047523	042514	000
5613				
5614	034073	015	046012	040517
5615	034100	020104	044124	020105

MSG20: .ASCIZ <15><12>/PUT ANY TWO CARDS IN INPUT HOPPER/

MSG21: .ASCIZ <15><12>/PRESS END OF FILE BUTTON/

MSG22: .ASCII <15><12>/WHEN PRINTING STOPS PUT 'HALT' AND 'SINGLE BUS CYCLE'/

.ASCII <15><12>'(IF PRESENT) CONSOLE SWITCHES DOWN AND HIT/

.ASCII <15><12>'CONTINUE' ON THE CONSOLE UNTIL ONE CARD IS READ/

.ASCII <15><12>/THEN PUT UP THE ONE OR TWO SWITCHES AND HIT/

.ASCIZ <15><12>'CONTINUE' ON THE CONSOLE

MSG23: .ASCII <15><12>'LOAD THE ADDRESS OF THE DESIRED TEST INTO THE'

5616	034106	042101	051104	051505
5617	034114	020123	043117	052040
5618	034122	042510	042040	051505
5619	034130	051111	042105	052040
5620	034136	051505	020124	047111
5621	034144	047524	052040	042510
5622	034152	005015	053523	052111
5623	034160	044103	051040	043505
5624	034166	051511	042524	020122
5625	034174	020055	027111	027105
5626	034202	052040	042510	040440
5627	034210	042104	042522	051523
5628	034216	047440	020106	044124
5629	034224	020105	041523	050117
5630	034232	105		
5631	034233	015	044412	051516
5632	034240	051124	041525	044524
5633	034246	047117	040440	020124
5634	034254	044124	020105	042502
5635	034262	044507	047116	047111
5636	034270	020107	043117	052040
5637	034276	042510	052040	051505
5638	034304	124		
5639	034305	015	052012	042510
5640	034312	020116	051120	051505
5641	034320	020123	047503	052116
5642	034326	047111	042525	051440
5643	034334	044527	041524	000110
5644				
5645	034342	005015	042523	020124
5646	034350	042504	044523	042522
5647	034356	020104	053523	052111
5648	034364	044103	051040	043505
5649	034372	051511	042524	020122
5650	034400	050117	044524	047117
5651	034406	123		
5652	034407	015	052012	042510
5653	034414	020116	051120	051505
5654	034422	020123	047503	052116
5655	034430	047111	042525	051440
5656	034436	044527	041524	000110
5657				
5658	034444	005015	047514	042101
5659	034452	042040	051505	051111
5660	034460	042105	050040	052101
5661	034466	042524	047122	053040
5662	034474	046101	042525	044440
5663	034502	052116	020117	044124
5664	034510	020105	053523	052111
5665	034516	044103		
5666	034520	005015	042522	044507
5667	034526	052123	051105	026440
5668	034534	044440	042456	020056
5669	034542	047111	047524	051440

.ASCII <15><12>"SWITCH REGISTER - I.E. THE ADDRESS OF THE SCOPE"

.ASCII <15><12>"INSTRUCTION AT THE BEGINNING OF THE TEST"

.ASCIZ <15><12>"THEN PRESS CONTINUE SWITCH"

MSG24: .ASCII <15><12>"SET DESIRED SWITCH REGISTER OPTIONS"

.ASCIZ <15><12>"THEN PRESS CONTINUE SWITCH"

MSG25: .ASCII <15><12>"LOAD DESIRED PATTERN VALUE INTO THE SWITCH"

.ASCII <15><12>"REGISTER - I.E. INTO SWS<11:0>"

5670	034550	051527	030474	035061
5671	034556	037060		
5672	034560	005015	044124	047105
5673	034566	050040	042522	051523
5674	034574	041440	047117	044524
5675	034602	052516	020105	053523
5676	034610	052111	044103	000
5677				
5678		034616		
5679				
5680				
5681				
5682				
5683	034616	052123	052101	051525
5684	034624	051040	043505	051511
5685	034632	042524	020122	041450
5686	034640	051504	020051	044502
5687	034646	030124	020067	047516
5688	034654	020124	042523	020124
5689	034662	054502		
5690	034664	005015	047111	052111
5691	034672	040511	044514	040532
5692	034700	044524	047117	050040
5693	034706	046125	042523	000
5694				
5695	034713	103	046117	046525
5696	034720	020116	047503	047125
5697	034726	020124	042522	044507
5698	034734	052123	051105	024040
5699	034742	042103	024503	047040
5700	034750	052117	041440	042514
5701	034756	051101	042105	041040
5702	034764	131		
5703	034765	015	044412	044516
5704	034772	044524	046101	055111
5705	035000	052101	047511	020116
5706	035006	052520	051514	000105
5707				
5708	035014	052502	020123	042101
5709	035022	051104	051505	020123
5710	035030	042522	044507	052123
5711	035036	051105	024040	042103
5712	035044	024501	047040	052117
5713	035052	041440	042514	051101
5714	035060	042105	041040	131
5715	035065	015	044412	044516
5716	035072	044524	046101	055111
5717	035100	052101	047511	020116
5718	035106	052520	051514	000105
5719				
5720	035114	052123	052101	051525
5721	035122	051040	043505	051511
5722	035130	042524	020122	047503
5723	035136	052116	047105	051524

.ASCIZ <15><12>"THEN PRESS CONTINUE SWITCH"

.EVEN

;ERROR ITEMS MESSAGE TABLE

EM1: .ASCII "STATUS REGISTER (CDS) BIT07 NOT SET BY"

.ASCIZ <15><12>"INITIALIZATION PULSE"

EM2: .ASCII "COLUMN COUNT REGISTER (CDC) NOT CLEARED BY"

.ASCIZ <15><12>"INITIALIZATION PULSE"

EM3: .ASCII "BUS ADDRESS REGISTER (CDA) NOT CLEARED BY"

.ASCIZ <15><12>"INITIALIZATION PULSE"

EM4: .ASCIZ "STATUS REGISTER CONTENTS INCORRECT"

5724	035144	044440	041516	051117		
5725	035152	042522	052103	000		
5726						
5727	035157	103	046117	046525	EMS:	.ASCII "COLUMN COUNT REGISTER (CDC) NOT ABLE TO"
5728	035164	020116	047503	047125		
5729	035172	020124	042522	044507		
5730	035200	052123	051105	024040		
5731	035206	042103	024503	047040		
5732	035214	052117	040440	046102		
5733	035222	020105	047524			
5734	035226	005015	042502	046040		.ASCIZ <15><12>"BE LOADED WITH ALL ONES"
5735	035234	040517	042504	020104		
5736	035242	044527	044124	040440		
5737	035250	046114	047440	042516		
5738	035256	000123				
5739						
5740	035260	047503	052514	047115	EM6:	.ASCII "COLUMN COUNT REGISTER (CDC) NOT CLEARED"
5741	035266	041440	052517	052116		
5742	035274	051040	043505	051511		
5743	035302	042524	020122	041450		
5744	035310	041504	020051	047516		
5745	035316	020124	046103	040505		
5746	035324	042522	104			
5747	035327	015	041012	020131		.ASCIZ <15><12>"BY POWER CLEAR"
5748	035334	047520	042527	020122		
5749	035342	046103	040505	000122		
5750						
5751	035350	052502	020123	042101	EM7:	.ASCII "BUS ADDRESS REGISTER (CDA) NOT ABLE TO BE"
5752	035356	051104	051505	020123		
5753	035364	042522	044507	052123		
5754	035372	051105	024040	042103		
5755	035400	024501	047040	052117		
5756	035406	040440	046102	020105		
5757	035414	047524	041040	105		
5758	035421	015	046012	040517		.ASCIZ <15><12>"LOADED WITH ALL ONES"
5759	035426	042504	020104	044527		
5760	035434	044124	040440	046114		
5761	035442	047440	042516	000123		
5762						
5763	035450	052502	020123	042101	EM10:	.ASCII "BUS ADDRESS REGISTER (CDA) NOT CLEARED"
5764	035456	051104	051505	020123		
5765	035464	042522	044507	052123		
5766	035472	051105	024040	042103		
5767	035500	024501	047040	052117		
5768	035506	041440	042514	051101		
5769	035514	042105				
5770	035516	005015	054502	050040		.ASCIZ <15><12>"BY POWER CLEAR"
5771	035524	053517	051105	041440		
5772	035532	042514	051101	000		
5773						
5774	035537	103	047117	051124	EM11:	.ASCII "CONTROLLER READY DID NOT CLEAR ON"
5775	035544	046117	042514	020122		
5776	035552	042522	042101	020131		
5777	035560	044504	020104	047516		

5778	035566	020124	046103	040505	
5779	035574	020122	047117		
5780	035600	005015	042522	042101	.ASCIZ <15><12>"READING A CARD"
5781	035606	047111	020107	020101	
5782	035614	040503	042122	000	
5783					
5784	035621	103	047117	051124	EM12: .ASCII "CONTROLLER READY DID NOT CLEAR BIT00"
5785	035626	046117	042514	020122	
5786	035634	042522	042101	020131	
5787	035642	044504	020104	047516	
5788	035650	020124	046103	040505	
5789	035656	020122	044502	030124	
5790	035664	060			
5791	035665	015	047412	020106	.ASCIZ <15><12>"OF STATUS REGISTER (CDS)"
5792	035672	052123	052101	051525	
5793	035700	051040	043505	051511	
5794	035706	042524	020122	041450	
5795	035714	051504	000051		
5796					
5797	035720	047503	052116	047522	EM13: .ASCIZ "CONTROLLER DID NOT SET WITHIN 1 SECOND"
5798	035726	046114	051105	042040	
5799	035734	042111	047040	052117	
5800	035742	051440	052105	053440	
5801	035750	052111	044510	020116	
5802	035756	020061	042523	047503	
5803	035764	042116	000		
5804					
5805	035767	105	051122	051117	EM14: .ASCIZ "ERROR (BIT15) SET IN STATUS REGISTER (CDS)"
5806	035774	024040	044502	030524	
5807	036002	024465	051440	052105	
5808	036010	044440	020116	052123	
5809	036016	052101	051525	051040	
5810	036024	043505	051511	042524	
5811	036032	020122	041450	051504	
5812	036040	000051			
5813					
5814	036042	044502	020124	051117	EM15: .ASCII "BIT OR BITS SET IN STATUS REGISTER (CDS) THAT"
5815	036050	041040	052111	020123	
5816	036056	042523	020124	047111	
5817	036064	051440	040524	052524	
5818	036072	020123	042522	044507	
5819	036100	052123	051105	024040	
5820	036106	042103	024523	052040	
5821	036114	040510	124		
5822	036117	015	051412	047510	.ASCIZ <15><12>"SHOULD NOT BE"
5823	036124	046125	020104	047516	
5824	036132	020124	042502	000	
5825					
5826	036137	102	051525	020131	EM16: .ASCII "BUSY SET IN STATUS REGISTER (CDS)"
5827	036144	042523	020124	047111	
5828	036152	051440	040524	052524	
5829	036160	020123	042522	044507	
5830	036166	052123	051105	024040	
5831	036174	042103	024523		

M14

MAINDEC - 11 - DZCDB-A
DZCDBA.P11MACY11 27(663)
DATA TABLES

19-FEB-76 14:38 PAGE 112

SEQ 0181

5832	036200	005015	052111	051440		.ASCIZ <15><12>"IT SHOULD NOT BE"
5833	036206	047510	046125	020104		
5834	036214	047516	020124	042502		
5835	036222	000				
5836						
5837	036223	122	051505	047524	EM17:	.ASCII "RESTORING CPU STATUS AFTER READING A CARD"
5838	036230	044522	043516	041440		
5839	036236	052520	051440	040524		
5840	036244	052524	020123	043101		
5841	036252	042524	020122	042522		
5842	036260	042101	047111	020107		
5843	036266	020101	040503	042122		
5844	036274	005015	046103	040505		.ASCIZ <15><12>"CLEARED CONTROLLER READY IN STATUS REGISTER"
5845	036302	042522	020104	047503		
5846	036310	052116	047522	046114		
5847	036316	051105	051040	040505		
5848	036324	054504	044440	020116		
5849	036332	052123	052101	051525		
5850	036340	051040	043505	051511		
5851	036346	042524	000122			
5852						
5853	036352	047516	044440	052116	EM20:	.ASCIZ "NO INTERRUPT OCCURRED"
5854	036360	051105	052522	052120		
5855	036366	047440	041503	051125		
5856	036374	042522	000104			
5857						
5858	036400	047101	044440	052116	EM21:	.ASCIZ "AN INTERRUPT OCCURRED - SHOULD NOT HAVE"
5859	036406	051105	052522	052120		
5860	036414	047440	041503	051125		
5861	036422	042522	020104	020055		
5862	036430	044123	052517	042114		
5863	036436	047040	052117	044040		
5864	036444	053101	000105			
5865						
5866	036450	047111	042524	051122	EM22:	.ASCIZ "INTERRUPT ALREADY OCCURRED AT A HIGHER LEVEL"
5867	036456	050125	020124	046101		
5868	036464	042522	042101	020131		
5869	036472	041517	052503	051122		
5870	036500	042105	040440	020124		
5871	036506	020101	044510	044107		
5872	036514	051105	046040	053105		
5873	036522	046105	000			
5874						
5875	036525	103	047117	051124	EM23:	.ASCIZ "CONTROLLER READY NOT SET"
5876	036532	046117	042514	020122		
5877	036540	042522	042101	020131		
5878	036546	047516	020124	042523		
5879	036554	000124				
5880						
5881	036556	047111	042524	051122	EM24:	.ASCIZ "INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL"
5882	036564	050125	020124	046101		
5883	036572	042522	042101	020131		
5884	036600	041517	052503	051122		
5885	036606	042105	040440	020124		

5886	036614	020101	047514	042527	
5887	036622	020122	042514	042526	
5888	036630	000114			
5889					
5890	036632	047101	044440	052116	EM25: .ASCIZ "AN INTERRUPT OCCURRED AT TWO DIFFERENT LEVELS"
5891	036640	051105	052522	052120	
5892	036646	047440	041503	051125	
5893	036654	042522	020104	052101	
5894	036662	052040	047527	042040	
5895	036670	043111	042506	042522	
5896	036676	052116	046040	053105	
5897	036704	046105	000123		
5898					
5899	036710	051105	047522	020122	EM26: .ASCII "ERROR (BIT15) NOT SET IN STATUS REGISTER (CDS)"
5900	036716	041050	052111	032461	
5901	036724	020051	047516	020124	
5902	036732	042523	020124	047111	
5903	036740	051440	040524	052524	
5904	036746	020123	042522	044507	
5905	036754	052123	051105	024040	
5906	036762	042103	024523		
5907	036766	005015	052111	051440	.ASCIZ <15><12>"IT SHOULD BE"
5908	036774	047510	046125	020104	
5909	037002	042502	000		
5910					
5911	037005	116	047117	042455	EM27: .ASCII "NON-EXISTANT MEMORY (BIT09) NOT SET IN"
5912	037012	044530	052123	047101	
5913	037020	020124	042515	047515	
5914	037026	054522	024040	044502	
5915	037034	030124	024471	047040	
5916	037042	052117	051440	052105	
5917	037050	044440	116		
5918	037053	015	051412	040524	.ASCIZ <15><12>"STATUS REGISTER (CDS) - IT SHOULD BE"
5919	037060	052524	020123	042522	
5920	037066	044507	052123	051105	
5921	037074	024040	042103	024523	
5922	037102	026440	044440	020124	
5923	037110	044123	052517	042114	
5924	037116	041040	000105		
5925					
5926	037122	054105	042524	042116	EM30: .ASCII "EXTENDED MEMORY (BIT05) NOT SET IS STATUS"
5927	037130	042105	046440	046505	
5928	037136	051117	020131	041050	
5929	037144	052111	032460	020051	
5930	037152	047516	020124	042523	
5931	037160	020124	051511	051440	
5932	037166	040524	052524	123	
5933	037173	015	051012	043505	.ASCIZ <15><12>"REGISTER (CDS)"
5934	037200	051511	042524	020122	
5935	037206	041450	051504	000051	
5936					
5937	037214	054105	042524	042116	EM31: .ASCII "EXTENDED MEMORY (BIT04) NOT SET IN STATUS"
5938	037222	042105	046440	046505	
5939	037230	051117	020131	041050	

5940	037236	052111	032060	020051	
5941	037244	047516	020124	042523	
5942	037252	020124	047111	051440	
5943	037260	040524	052524	123	
5944	037265	015	051012	043505	.ASCIZ <15><12>"REGISTER (CDS)"
5945	037272	051511	042524	020122	
5946	037300	041450	051504	000051	
5947					
5948	037306	047503	052116	047105	EM32: .ASCII "CONTENTS OF BUS ADDRESS REGISTER (CDA)"
5949	037314	051524	047440	020106	
5950	037322	052502	020123	042101	
5951	037330	051104	051505	020123	
5952	037336	042522	044507	052123	
5953	037344	051105	024040	042103	
5954	037352	024501			
5955	037354	005015	047111	047503	.ASCIZ <15><12>"INCORRECT"
5956	037362	051122	041505	000124	
5957					
5958	037370	047503	052116	047105	EM33: .ASCII "CONTENTS OF COLUMN COUNT REGISTER (CDC)"
5959	037376	051524	047440	020106	
5960	037404	047503	052514	047115	
5961	037412	041440	052517	052116	
5962	037420	051040	043505	051511	
5963	037426	042524	020122	041450	
5964	037434	041504	051		
5965	037437	015	044412	041516	.ASCIZ <15><12>"INCORRECT"
5966	037444	051117	042522	052103	
5967	037452	000			
5968					
5969	037453	122	040505	042504	EM34: .ASCII "READER OFF-LINE BUT CARD READER ERROR"
5970	037460	020122	043117	026506	
5971	037466	044514	042516	041040	
5972	037474	052125	041440	051101	
5973	037502	020104	042522	042101	
5974	037510	051105	042440	051122	
5975	037516	051117			
5976	037520	005015	041050	052111	.ASCIZ <15><12>"(BIT14) NOT SET IN STATUS REGISTER (CDS)"
5977	037526	032061	020051	047516	
5978	037534	020124	042523	020124	
5979	037542	047111	051440	040524	
5980	037550	052524	020123	042522	
5981	037556	044507	052123	051105	
5982	037564	024040	042103	024523	
5983	037572	000			
5984					
5985	037573	116	020117	051124	EM35: .ASCII "NO TRANSITION TO ON-LINE (BIT03) OCCURRED"
5986	037600	047101	044523	044524	
5987	037606	047117	052040	020117	
5988	037614	047117	046055	047111	
5989	037622	020105	041050	052111	
5990	037630	031460	020051	041517	
5991	037636	052503	051122	042105	
5992	037644	005015	047111	051440	.ASCIZ <15><12>"IN STATUS REGISTER (CDS)"
5993	037652	040524	052524	020123	

5994	037660	042522	044507	052123	
5995	037666	051105	024040	042103	
5996	037674	024523	000		
5997					
5998	037677	103	047117	042524	EM36: .ASCIZ "CONTENTS OF STATUS REGISTER #2 (CDD) INCORRECT"
5999	037704	052116	020123	043117	
6000	037712	051440	040524	052524	
6001	037720	020123	042522	044507	
6002	037726	052123	051105	021440	
6003	037734	020062	041450	042104	
6004	037742	020051	047111	047503	
6005	037750	051122	041505	000124	
6006					
6007	037756	047516	044440	052116	EM37: .ASCIZ "NO INTERRUPT WITH PROCESSOR AT LEVEL 0"
6008	037764	051105	052522	052120	
6009	037772	053440	052111	020110	
6010	040000	051120	041517	051505	
6011	040006	047523	020122	052101	
6012	040014	046040	053105	046105	
6013	040022	030040	000		
6014					
6015	040025	104	052101	020101	EM40: .ASCIZ "DATA LATE (BIT10) NOT SET IN STATUS REGISTER (CDS)"
6016	040032	040514	042524	024040	
6017	040040	044502	030524	024460	
6018	040046	047040	052117	051440	
6019	040054	052105	044440	020116	
6020	040062	052123	052101	051525	
6021	040070	051040	043505	051511	
6022	040076	042524	020122	041450	
6023	040104	051504	000051		
6024					
6025	040110	043117	026506	044514	EM41: .ASCIZ "OFF-LINE (BIT12) NOT SET IN STATUS REGISTER (CDS)"
6026	040116	042516	024040	044502	
6027	040124	030524	024462	047040	
6028	040132	052117	051440	052105	
6029	040140	044440	020116	052123	
6030	040146	052101	051525	051040	
6031	040154	043505	051511	042524	
6032	040162	020122	041450	051504	
6033	040170	000051			
6034					
6035	040172	043117	026506	044514	EM42: .ASCII "OFF-LINE (BIT12) SET IN STATUS REGISTER"
6036	040200	042516	024040	044502	
6037	040206	030524	024462	051440	
6038	040214	052105	044440	020116	
6039	040222	052123	052101	051525	
6040	040230	051040	043505	051511	
6041	040236	042524	122		
6042	040241	015	020012	020055	.ASCIZ "<15><12>" - IT SHOULD NOT BE"
6043	040246	052111	051440	047510	
6044	040254	046125	020104	047516	
6045	040262	020124	042502	000	
6046					
6047	040267	120	041511	020113	EM43: .ASCIZ "PICK CHECK (BIT13) NOT SET IN STATUS REGISTER #2 (CDD)"

6048	040274	044103	041505	020113
6049	040302	041050	052111	031461
6050	040310	020051	047516	020124
6051	040316	042523	020124	047111
6052	040324	051440	040524	052524
6053	040332	020123	042522	044507
6054	040340	052123	051105	021440
6055	040346	020062	041450	042104
6056	040354	000051		
6057				
6058	040356	052123	041501	020113
6059	040364	044103	041505	020113
6060	040372	041050	052111	031061
6061	040400	020051	047516	020124
6062	040406	042523	020124	047111
6063	040414	051440	040524	052524
6064	040422	020123	042522	044507
6065	040430	052123	051105	021440
6066	040436	020062	041450	042104
6067	040444	000051		
6068				
6069	040446	047105	020104	043117
6070	040454	043040	046111	020105
6071	040462	041050	052111	031461
6072	040470	020051	042523	020124
6073	040476	047111	051440	040524
6074	040504	052524	020123	042522
6075	040512	044507	052123	051105
6076	040520	024040	042103	024523
6077	040526	005015	026440	044440
6078	040534	020124	044123	052517
6079	040542	042114	047040	052117
6080	040550	041040	000105	
6081				
6082	040554	042522	042101	041440
6083	040562	042510	045503	024040
6084	040570	044502	030524	024464
6085	040576	051440	052105	044440
6086	040604	020116	052123	052101
6087	040612	051525	051040	043505
6088	040620	051511	042524	020122
6089	040626	041450	051504	051
6090	040633	015	020012	020055
6091	040640	052111	051440	047510
6092	040646	046125	020104	047516
6093	040654	020124	042502	000
6094				
6095	040661	110	050117	042520
6096	040666	020122	044103	041505
6097	040674	020113	041050	052111
6098	040702	031060	020051	042523
6099	040710	020124	047111	051440
6100	040716	040524	052524	020123
6101	040724	042522	044507	052123

EM44: .ASCIZ "STACK CHECK (BIT12) NOT SET IN STATUS REGISTER #2 (CDS)"

EM45: .ASCII "END OF FILE (BIT13) SET IN STATUS REGISTER (CDS)"

.ASCIZ <15><12>" - IT SHOULD NOT BE"

EM46: .ASCII "READ CHECK (BIT14) SET IN STATUS REGISTER (CDS)"

.ASCIZ <15><12>" - IT SHOULD NOT BE"

EM47: .ASCII "HOPPER CHECK (BIT02) SET IN STATUS REGISTER (CDS)"

6102	040732	051105	024040	042103
6103	040740	024523		
6104	040742	005015	026440	044440
6105	040750	020124	044123	052517
6106	040756	042114	047040	052117
6107	040764	041040	000105	
6108				
6109	040770	047105	020104	043117
6110	040776	043040	046111	020105
6111	041004	041050	052111	031461
6112	041012	020051	043117	051440
6113	041020	040524	052524	020123
6114	041026	042522	044507	052123
6115	041034	051105	024040	042103
6116	041042	024523	047040	052117
6117	041050	051440	052105	
6118	041054	005015	026440	044440
6119	041062	020124	044123	052517
6120	041070	042114	041040	000105
6121				
6122	041076	042522	042101	041440
6123	041104	042510	045503	024040
6124	041112	044502	030524	024464
6125	041120	047440	020106	052123
6126	041126	052101	051525	051040
6127	041134	043505	051511	042524
6128	041142	020122	041450	051504
6129	041150	020051	047516	020124
6130	041156	042523	124	
6131	041161	015	020012	020055
6132	041166	052111	051440	047510
6133	041174	046125	020104	042502
6134	041202	000		
6135				
6136	041203	:10	050117	042520
6137	041210	020122	044103	041505
6138	041216	020113	041050	052111
6139	041224	031061	020051	043117
6140	041232	051440	040524	052524
6141	041240	020123	042522	044507
6142	041246	052123	051105	024040
6143	041254	042103	024523	047040
6144	041262	052117	051440	052105
6145	041270	005015	026440	044440
6146	041276	020124	044123	052517
6147	041304	042114	041040	000105
6148				
6149	041312	047105	020104	043117
6150	041320	043040	046111	020105
6151	041326	041050	052111	031461
6152	041334	020051	043117	051440
6153	041342	040524	052524	020123
6154	041350	042522	044507	052123
6155	041356	051105	024040	042103

.ASCIZ <15><12>" - IT SHOULD NOT BE"

EMS0: .ASCII "END OF FILE (BIT13) OF STATUS REGISTER (CDS) NOT SET"

.ASCIZ <15><12>" - IT SHOULD BE"

EMS1: .ASCII "READ CHECK (BIT14) OF STATUS REGISTER (CDS) NOT SET"

.ASCIZ <15><12>" - IT SHOULD BE"

EMS2: .ASCII "HOPPER CHECK (BIT12) OF STATUS REGISTER (CDS) NOT SET"

.ASCIZ <15><12>" - IT SHOULD BE"

EMS3: .ASCII "END OF FILE (BIT13) OF STATUS REGISTER (CDS) DID NOT"

6156	041364	024523	042040	042111	
6157	041372	047040	052117		
6158	041376	005015	046103	040505	.ASCIZ <15><12>"CLEAR WITH TRANSITION TO ON-LINE"
6159	041404	020122	044527	044124	
6160	041412	052040	040522	051516	
6161	041420	052111	047511	020116	
6162	041426	047524	047440	026516	
6163	041434	044514	042516	000	
6164					
6165	041441	122	040505	020104	EMS4: .ASCIZ "READ CHECK (BIT14) NOT SET IN STATUS REGISTER #2 (CDD)"
6166	041446	044103	041505	020113	
6167	041454	041050	052111	032061	
6168	041462	020051	047516	020124	
6169	041470	042523	020124	047111	
6170	041476	051440	040524	052524	
6171	041504	020123	042522	044507	
6172	041512	052123	051105	021440	
6173	041520	020062	041450	042104	
6174	041526	000051			
6175					
6176	041530	040503	042122	051040	EMS5: .ASCIZ "CARD READER ERROR BUT NOT BOTH CARD"
6177	041536	040505	042504	020122	
6178	041544	051105	047522	020122	
6179	041552	052502	020124	047516	
6180	041560	020124	030070	044124	
6181	041566	041440	051101	000104	
6182					
6183	041574	040503	042122	051040	EMS6: .ASCIZ "CARD READER ERROR BUT NOT OFF-LINE"
6184	041602	040505	042504	020122	
6185	041610	051105	047522	020122	
6186	041616	052502	020124	047516	
6187	041624	020124	043117	026506	
6188	041632	044514	042516	000	
6189					
6190	041637	105	042116	047440	EMS7: .ASCIZ "END OF FILE (BIT13) ERROR OF STATUS REGISTER (CDS)"
6191	041644	020106	044506	042514	
6192	041652	024040	044502	030524	
6193	041660	024463	042440	051122	
6194	041666	051117	047440	020106	
6195	041674	052123	052101	051525	
6196	041702	051040	043505	051511	
6197	041710	042524	020122	041450	
6198	041716	051504	000051		
6199					
6200	041722	040504	040524	042440	EM60: .ASCIZ "DATA ERROR (BIT11) OF STATUS REGISTER (CDS)"
6201	041730	051122	051117	024040	
6202	041736	044502	030524	024461	
6203	041744	047440	020106	052123	
6204	041752	052101	051525	051040	
6205	041760	043505	051511	042524	
6206	041766	020122	041450	051504	
6207	041774	000051			
6208					
6209	041776	040504	040524	046040	EM61: .ASCIZ "DATA LATE (BIT11) ERROR OF STATUS REGISTER (CDS)"

6210	042004	052101	020105	041050	
6211	042012	052111	030461	020051	
6212	042020	051105	047522	020122	
6213	042026	043117	051440	040524	
6214	042034	052524	020123	042522	
6215	042042	044507	052123	051105	
6216	042050	024040	042103	024523	
6217	042056	000			
6218					
6219	042057	116	047117	042455	EM62: .ASCII "NON-EXISTANT MEMORY (BIT09) ERROR OF STATUS"
6220	042064	044530	052123	047101	
6221	042072	020124	042515	047515	
6222	042100	054522	024040	044502	
6223	042106	030124	024471	042440	
6224	042114	051122	051117	047440	
6225	042122	020106	052123	052101	
6226	042130	051525			
6227	042132	005015	042522	044507	.ASCIZ <15><12>"REGISTER (CDS)"
6228	042140	052123	051105	024040	
6229	042146	042103	024523	000	
6230					
6231	042153	104	051511	051501	EM63: .ASCIZ "DISASTEROUS ERROR BUT NO ERROR BITS SET"
6232	042160	042524	047522	051525	
6233	042166	042440	051122	051117	
6234	042174	041040	052125	047040	
6235	042202	020117	051105	047522	
6236	042210	020122	044502	051524	
6237	042216	051440	052105	000	
6238					
6239	042223	104	052101	020101	EM64: .ASCII "DATA PACKING (BIT01) OF STATUS REGISTER (CDS)"
6240	042230	040520	045503	047111	
6241	042236	020107	041050	052111	
6242	042244	030460	020051	043117	
6243	042252	051440	040524	052524	
6244	042260	020123	042522	044507	
6245	042266	052123	051105	024040	
6246	042274	042103	024523		
6247	042300	005015	047516	020124	.ASCIZ <15><12>"NOT SET - I* SHOULD BE"
6248	042306	042523	020124	020055	
6249	042314	052111	051440	047510	
6250	042322	046125	020104	042502	
6251	042330	000			
6252					
6253	042331	123	047510	046125	EM65: .ASCII "SHOULD BE ADDRESSING BINARY DECK"
6254	042336	020104	042502	040440	
6255	042344	042104	042522	051523	
6256	042352	047111	020107	044502	
6257	042360	040516	054522	042040	
6258	042366	041505	113		
6259	042371	015	050012	047522	.ASCIZ <15><12>"PROGRAM DOES NOT AGREE"
6260	042376	051107	046501	042040	
6261	042404	042517	020123	047516	
6262	042412	020124	043501	042522	
6263	042420	000105			

6264				
6265	042422	047503	052116	047105
6266	042430	051524	047440	020106
6267	042436	052123	052101	051525
6268	042444	051040	043505	051511
6269	042452	042524	020122	041450
6270	042460	051504	020051	047111
6271	042466	047503	051122	041505
6272	042474	124		
6273	042475	015	020012	020055
6274	042502	044123	052517	042114
6275	042510	041040	020105	042532
6276	042516	047522	000	
6277				
6278	042521	117	042104	041040
6279	042526	051525	040440	042104
6280	042534	042522	051523	041440
6281	042542	052501	042523	020104
6282	042550	020101	051124	050101
6283	042556	044440	020116	047516
6284	042564	026516	040520	045503
6285	042572	046440	042117	000105
6286				
6287				
6288				
6289	042600	024040	041520	020051
6290	042606	020040	024040	050123
6291	042614	020051	020040	041450
6292	042622	051504	020051	020040
6293	042630	041450	042104	020051
6294	042636	020040	024040	051520
6295	042644	000051		
6296	042646	024040	041520	020051
6297	042654	020040	024040	050123
6298	042662	020051	020040	041450
6299	042670	051504	020051	020040
6300	042676	041450	042104	020051
6301	042704	020040	041450	041504
6302	042712	020051	020040	024040
6303	042720	051520	000051	
6304	042724	024040	041520	020051
6305	042732	020040	024040	050123
6306	042740	020051	020040	041450
6307	042746	051504	020051	020040
6308	042754	041450	042104	020051
6309	042762	020040	041450	040504
6310	042770	020051	020040	024040
6311	042776	051520	000051	
6312	043002	024040	041520	020051
6313	043010	020040	024040	050123
6314	043016	020051	020040	041450
6315	043024	051504	020051	020040
6316	043032	041450	042104	020051
6317	043040	020040	041450	041504

EM66: .ASCII "CONTENTS OF STATUS REGISTER (CDS) INCORRECT"

.ASCIZ <15><12>" - SHOULD BE ZERO"

EM67: .ASCIZ "ODD BUS ADDRESS CAUSED A TRAP IN NON-PACK MODE"

:ERROR ITEMS HEADER TABLE

DH1: .ASCIZ " (PC) (SP) (CDS) (CDD) (PS)"

DH2: .ASCIZ " (PC) (SP) (CDS) (CDD) (CDC) (PS)"

DH3: .ASCIZ " (PC) (SP) (CDS) (CDD) (CDA) (PS)"

DH4: .ASCIZ " (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)"

6318	043046	020051	020040	041450
6319	043054	040504	020051	020040
6320	043062	024040	051520	000051
6321	043070	024040	041520	020051
6322	043076	020040	024040	050123
6323	043104	020051	020040	041450
6324	043112	051504	020051	020040
6325	043120	041450	042104	020051
6326	043126	020040	041450	041504
6327	043134	020051	020040	041450
6328	043142	040504	020051	020040
6329	043150	041450	040504	020051
6330	043156	020040	024040	051520
6331	043164	051		
6332	043165	015	020012	020040
6333	043172	020040	020040	020040
6334	043200	020040	020040	020040
6335	043206	020040	020040	020040
6336	043214	020040	020040	020040
6337	043222	020040	020040	020040
6338	043230	020040	020040	020040
6339	043236	020040	040527	020123
6340	043244	020040	020040	044123
6341	043252	000102		
6342	043254	024040	041520	020051
6343	043262	020040	024040	050123
6344	043270	020051	020040	041450
6345	043276	051504	020051	020040
6346	043304	041450	042104	020051
6347	043312	020040	041450	041504
6348	043320	020051	020040	041450
6349	043326	040504	020051	020040
6350	043334	041450	041504	020051
6351	043342	020040	024040	051520
6352	043350	051		
6353	043351	015	020012	020040
6354	043356	020040	020040	020040
6355	043364	020040	020040	020040
6356	043372	020040	020040	020040
6357	043400	020040	020040	020040
6358	043406	020040	020040	020040
6359	043414	040527	020123	020040
6360	043422	020040	020040	020040
6361	043430	020040	020040	044123
6362	043436	000102		
6363	043440	024040	041520	020051
6364	043446	020040	051450	024520
6365	043454	020040	024040	042103
6366	043462	024523	020040	024040
6367	043470	042103	024504	020040
6368	043476	024040	042103	024504
6369	043504	020040	020040	050050
6370	043512	024523		
6371	043514	005015	020040	020040

DH5: .ASCII " (PC) (SP) (CDS) (CDD) (CDC) (CDA) (CDA) (PS)"

.ASCIZ <15><12>" WAS SHB"

DH6: .ASCII " (PC) (SP) (CDS) (CDD) (CDC) (CDA) (CDC) (PS)"

.ASCIZ <15><12>" WAS SHB"

DH7: .ASCII " (PC) (SP) (CDS) (CDD) (CDD) (PS)"

.ASCIZ <15><12>" WAS SHB"

6372	043522	020040	020040	020040
6373	043530	020040	020040	020040
6374	043536	020040	020040	020040
6375	043544	020040	040527	020123
6376	043552	020040	020040	044123
6377	043560	000102		
6378				
6379				
6380				
6381				
6382	043562	001116	001174	001200
6383	043570	001202	001176	000000
6384	043576	001116	001174	001200
6385	043604	001202	001204	001176
6386	043612	000000		
6387	043614	001116	001174	001200
6388	043622	001202	001206	001176
6389	043630	000000		
6390	043632	001116	001174	001200
6391	043640	001202	001204	001206
6392	043646	001176	000000	
6393	043652	001116	001174	001200
6394	043660	001202	001204	001206
6395	043666	001210	001176	000000
6396	043674	001116	001174	001200
6397	043702	001202	001210	001176
6398	043710	000000		
6399				

.EVEN

;ERROR ITEMS DATA TABLE

DT1:	\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$REG7,0
DT2:	\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$TMP2,\$REG7,0
DT3:	\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$TMP3,\$REG7,0
DT4:	\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$TMP2,\$TMP3,\$REG7,0
DT5:	\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$TMP2,\$TMP3,\$TMP4,\$REG7,0
DT6:	\$ERRPC,\$REG6,\$TMP0,\$TMP1,\$TMP4,\$REG7,0

```

6400
6401 ;*****
6402 ;SUBROUTINE TO INITIALIZE CSR AND DBR POINTERS
6403 ;*****
6404
6405 043712 013703 001232 SETUP: MOV CDST,CDS ;SET UP STATUS REGISTER POINTER
6406 043716 013704 001234 MOV CDCC,CDC ;SET UP COLUMN COUNT REGISTER POINTER
6407 043722 013705 001236 MOV CDBA,CDA ;SET UP BUS ADDRESS REGISTER POINTER
6408 043726 013702 001244 MOV INTVC,ADINT ;LOAD ADDRESS OF INTERRUPT VECTOR
6409 043732 013712 001246 MOV INTVC+2,(ADINT) ;SET UP CARD READER TRAP VECTOR
6410 043736 005077 135304 CLR @INTVC+2 ;TO HALT
6411 043742 005037 001252 CLR INTFLG ;INITIALIZE INTERRUPT FLAG
6412 043746 005037 001254 CLR TRFLG ;INITIALIZE TRACE FLAG
6413 043752 013746 000340 MOV PR7,-(SP) ;;PUT NEW PS ON STACK
6414 043756 012746 043764 MOV #64$,-(SP) ;;PUT NEW PC ON STACK
6415 043762 000002 RTI ;;POP NEW PC AND PS
6416 043764
6417 043764 000207 64$: RTS PC ;RETURN TO MAINLINE CODE
6418
6419 ;*****

```

;*****

;*****

;

;THIS MARKS THE BEGINNING OF THE MEMORY BUFFER AREA WHERE THE
;CONTENTS OF THE CARD COLUMNS ARE DUMPED ON READ CYCLES

;

```

6432 ;*****
6433 ;*****
6434 ;*****
6435 ;*****
6436 ;*****
6437
6438 043766 000000 BUFBEQ: 0
6439 0C0001 .END

```

ADINT =%000002

ALPCD 027276
ALPCDP 027536
ALPEND 027534
ALPENP 027655
ALPI 012340
BEGIN 002170
BINCD 027656
BINCDP 030116
BINEND 030114
BINENP 030235
BIT0 = 000001
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BKGND 012646
BKGND1 012660
BPTVEC= 000014
BUFBEQ 043766

BUFEND 015264
CARDIM 025434
CDA =%000005

276*	1147*	1172*	1194	1209*	1217*	1242*	1252*	1266*	1291*	1327	1332*	1344*
1364*	1389*	1425	1430*	1442*	1462*	1487*	1523	1528*	1540*	1560*	1585*	1621
1626*	1638*	1653*	1678*	1714	1719*	1731*	1751*	1776*	1812	1817*	1829*	1849*
1874*	1910	1915*	1923*	1944*	1969*	1981	1991*	1998*	2023*	2044*	2051	2060*
2067*	2092*	2114*	2257*	2282*	2321	2328*	2338*	2430*	2455*	2936*	3145*	3170*
3208*	3233*	3269*	3294*	3359*	3384*	3407*	3432*	3486*	3511*	3533*	3558*	3640*
3665*	3688*	3713*	3795*	3820*	3843*	3868*	3948*	3974*	4011*	4095*	4120*	6408*
6409*												
2410	4920*											
5004*												
2411	4999*											
5083*												
2405	2410*											
295	868*	907										
2406	5087*											
5172*												
2407	5166*											
5251*												
258*												
248*	258											
247*	257											
246*	256											
245*	255											
244*	254											
243*	253											
242*	252											
241*	251											
240*	250											
239*	249											
257*												
238*	4539											
237*	4592											
236*	2900											
235*	4546											
234*	4574											
233*												
256*												
255*												
254*												
253*												
252*												
251*												
250*												
249*												
2469*	2470	2475										
2472*	2473											
265*												
1050	1102	1189	1244	1322	1420	1518	1616	1709	1807	1905	1975	2040
2315	2342	2344	2355	2457	2796	3074	3449	3575	3613	3730	3768	3885
3992	4068	4346	6438*									
2494*	2495*	2496*	2497	2499	2653*	2654*	2655	2657	2958*			
4313*	4314*	4315	4366	4437*								
279*	948	1024*	1025	1034	1050*	1102*	1189*	1244*	1322*	1420*	1518*	1616*
1709*	1807*	1905*	1975*	2040*	2109*	2139	2205*	2208*	2210	2222*	2227	2239*

EM20	036352	547	5853#					
EM21	036400	554	5858#					
EM22	036450	561	5866#					
EM23	036525	568	5875#					
EM24	036556	575	5881#					
EM25	036632	582	5890#					
EM26	036710	589	5899#					
EM27	037005	597	5911#					
EM3	035014	446	5708#					
EM30	037122	605	5926#					
EM31	037214	613	5937#					
EM32	037306	628	5948#					
EM33	037370	637	5958#					
EM34	037453	646	5969#					
EM35	037573	654	5985#					
EM36	037677	669	5998#					
EM37	037756	678	6007#					
EM4	035114	454	5720#					
EM40	040025	685	6015#					
EM41	040110	693	6025#					
EM42	040172	701	6035#					
EM43	040267	709	6047#					
EM44	040356	717	6058#					
EM45	040446	725	6069#					
EM46	040554	733	6082#					
EM47	040661	741	6095#					
EM5	035157	461	5727#					
EM50	040770	749	6109#					
EM51	041076	757	6122#					
EM52	041203	765	6136#					
EM53	041312	773	6149#					
EM54	041441	782	6165#					
EM55	041530	790	6176#					
EM56	041574	797	6183#					
EM57	041637	804	6190#					
EM6	035260	469	5740#					
EM60	041722	812	6200#					
EM61	041776	819	6209#					
EM62	042057	827	6219#					
EM63	042153	662	6231#					
EM64	042223	835	6239#					
EM65	042331	843	6253#					
EM66	042422	851	6265#					
EM67	042521	859	6278#					
EM7	035350	477	5751#					
ENOCK	012232	2371#						
ERCD11	015602	315	3012#					
ERCD12	016106	3054	3057#					
ERFLG	001260	407#	2403#	2811	2813#	4337#	4392	4394#
ERPVEC=	000004	261#	4579	4580#	4582#	4585#		
ER12CD	016074	3011	3054#	4145				
ER1200	015306	326	2969#					
GNS =	***** U	290	4856	4857	4858	4859	4860	
HOOK1	015134	2910	2917#					

INIT	025442	1146	1216	1265	1363	1461	1559	1652	1750	1848	1943	1997	2066	2256
		2353	2427	3065	3101	3144	3338	3464	3594	3747	3911	3923	4054	4334
		4446												
INTFLG	001252	403	1334	1336	1347	1349*	1356	1432	1434	1445	1447*	1454	1530	1532
		1543	1545*	1552	1628	1630	1641	1643*	1645	1721	1723	1734	1736*	1743
		1819	1821	1832	1834*	1841	1926	1928*	1935	6411*				
INTVC	001244	400	6408	6409	6410*									
IOTVEC=	000020	266	874*	875*	2975*	2976*	3018*	3019*	4160*	4161*	4270*	4271*		
ISR	012714	2482	2485											
ISRBE	013114	2517	2526											
ISROE	013142	2509	2512	2534										
ISROE4	013440	2579	2598											
ISRO1	013150	2536	2573											
ISRO2	013354	2535	2578											
ISRER	013454	2488	2605											
ISRE1	013462	2605	2609											
ISRE2	013520	2610	2622											
ISRE3	013530	2614	2623	2626										
ISRLP	013006	2503	2521	2524										
ISRNC	013000	2498	2502	2620										
ISRNX	013120	2527												
ISRNX1	013136	2529	2531											
ISRRT	013042	2510	2513	2564	2566	2601								
ISRST	013600	2607	2619	2642	2644									
LASTCD	014614	2616	2755	2793	2831									
LOOPS	003170	1068	1071	1076										
LOOP6	003412	1133												
LOOP6A	003362	1122	1125											
LOOP6B	003400	1123	1128											
LSTCD1	014630	2833	2835											
MDATA	031003	2399	5324											
MLOGIC	030754	912	922	925	5319									
MLOOP	031152	4240	5346											
MPATS	031225	4306	5355											
MSG1	031274	3118	3310	3402	3528	3599	3683	3757	3838	3913	4033	4057	4138	5363
MSG10	032475	3837	5477											
MSG11	032522	5481												
MSG12	032574	4055	5488											
MSG13	033003	2814	5511											
MSG14	033062	2412	5519											
MSG15	033073	2408	5521											
MSG16	033104	5523												
MSG17	033125	4461	5526											
MSG18	033340	4460	5551											
MSG19	033372	4395	5556											
MSG2	031330	3103	3119	3340	3467	3596	3914	3921	4034	5368				
MSG20	033436	3192	5562											
MSG21	033502	3192	5568											
MSG22	033535	3067	5573											
MSG23	034073	4194	5614											
MSG24	034342	4200	4331	5645										
MSG25	034444	4309	5658											
MSG3	031375	3102	3187	3748	5375									
MSG4	031430	1350	1448	1546	1737	1835	1929	5380						

1760	1761	1763*	1764*	1768*	1769*	1772*	1773*	1775	1776	1778*	1779*	1782*
1783*	1785	1786	1788*	1789*	1792*	1793*	1796*	1797*	1799	1800	1802*	1803*
1812*	1813*	1831	1836*	1851*	1852*	1855*	1856*	1858	1859	1861*	1862*	1866*
1867*	1870*	1871*	1873	1874	1876*	1877*	1880*	1881*	1883	1884	1886*	1887*
1890*	1891*	1894*	1895*	1897	1898	1900*	1901*	1901*	1910*	1911*	1925	1930*
1947*	1950*	1951*	1953	1954	1956*	1957*	1961*	1962*	1965*	1966*	1968	1969
1970*	1971*	1981*	1982*	1988	2000*	2001*	2004*	2005*	2007	2008	2010*	2011*
2015*	2016*	2019*	2020*	2022	2023	2025*	2026*	2029*	2030*	2032	2033	2035*
2036*	2043	2045*	2046*	2051*	2052*	2056	2069*	2070*	2073*	2074*	2076	2077
2079*	2080*	2084*	2085*	2088*	2089*	2091	2092	2094*	2095*	2098*	2099*	2101
2102	2104*	2105*	2112	2259*	2260*	2263*	2264*	2266	2267	2269*	2270*	2274*
2275*	2278*	2279*	2281	2282	2284*	2285*	2288*	2289*	2291	2292	2294*	2295*
2298*	2299*	2302*	2303*	2305	2306	2308*	2309*	2321*	2322*	2341	2365	2416*
2417*	2423*	2424*	2432*	2433*	2436*	2437*	2439	2440	2442*	2443*	2447*	2448*
2451*	2452*	2454	2455	2586*	2588*	2593*	2719*	2724*	2819*	2824*	2859	2860
2872*	2879*	2880*	2892*	2893*	2896	2897*	2898	2899*	2904*	2905*	2944	2952
2974*	2989*	2990*	2994*	2998*	2999*	3005	3006	3007	3017*	3032*	3033*	3037*
3041*	3042*	3048	3049	3050	3147*	3148*	3151*	3152*	3154	3155	3157*	3158*
3162*	3163*	3166*	3167*	3169	3170	3172*	3173*	3176*	3177*	3179	3180	3182*
3183*	3210*	3211*	3214*	3215*	3217	3218	3220*	3221*	3225*	3226*	3229*	3230*
3232	3233	3235*	3236*	3239*	3240*	3242	3243	3245*	3246*	3252	3271*	3272*
3275*	3276*	3278	3279	3281*	3282*	3286*	3287*	3290*	3291*	3293	3294	3296*
3297*	3300*	3301*	3303	3304	3306*	3307*	3314	3361*	3362*	3365*	3366*	3368
3369	3371*	3372*	3376*	3377*	3380*	3381*	3383	3384	3386*	3387*	3390*	3391*
3393	3394	3396*	3397*	3406	3409*	3410*	3413*	3414*	3416	3417	3419*	3420*
3424*	3425*	3428*	3429*	3431	3432	3434*	3435*	3438*	3439*	3441	3442	3444*
3445*	3453	3488*	3489*	3492*	3493*	3495	3496	3498*	3499*	3503*	3504*	3507*
3508*	3510	3511	3513*	3514*	3517*	3518*	3520	3521	3523*	3524*	3532	3535*
3536*	3539*	3540*	3542	3543	3545*	3546*	3550*	3551*	3554*	3555*	3557	3558
3560*	3561*	3564*	3565*	3567	3568	3570*	3571*	3579	3642*	3643*	3646*	3647*
3649	3650	3652*	3653*	3657*	3658*	3661*	3662*	3664	3665	3667*	3668*	3671*
3672*	3674	3675	3677*	3678*	3687	3690*	3691*	3694*	3695*	3697	3698	3700*
3701*	3705*	3706*	3709*	3710*	3712	3713	3715*	3716*	3719*	3720*	3722	3723
3725*	3726*	3734	3797*	3798*	3801*	3802*	3804	3805	3807*	3808*	3812*	3813*
3816*	3817*	3819	3820	3822*	3823*	3826*	3827*	3829	3830	3832*	3833*	3842
3845*	3846*	3849*	3850*	3852	3853	3855*	3856*	3860*	3861*	3864*	3865*	3867
3868	3870*	3871*	3874*	3875*	3877	3878	3880*	3881*	3889	3951*	3952*	3955*
3956*	3958	3959	3961*	3962*	3966*	3967*	3970*	3971*	3973	3974	3976*	3977*
3980*	3981*	3983	3984	3986*	3987*	3997	4013	4097*	4098*	4101*	4102*	4104
4105	4107*	4108*	4112*	4113*	4116*	4117*	4119	4120	4122*	4123*	4126*	4127*
4129	4130	4132*	4133*	4142	4159*	4174*	4175*	4179*	4183*	4184*	4190	4191
4192	4206*	4207*	4210*	4211*	4213	4214	4216*	4217*	4223*	4224*	4227*	4228*
4230	4231	4233*	4234*	4269*	4284*	4285*	4289*	4293*	4294*	4300	4301	4302
4398*	4405*	4411*	4417*	4420*	4476*	4481*	4502	4506*	4530	4531	4543	4579*
4582	4584	4585	4602	4604*	4630*	4631	4632*	4634	4635	4636*	4638	4640
4642	4647	4649*	4651*	4659*	4663	4667	4668	4672	4708*	4709	4710	4711*
4716*	4717*	4718*	4724	4749	4750	4751	4752*	4753*	4775*	4776*	4777*	4778*
4779*	4780*	4781	4784*	4797	4799*	4801	4811	4813	4815	4816	4817	4818
4819	4821*	4822*	4839*	4840	4869*	4870*	4871*	4872*	4873*	4874*	4875	4881*
4885	4886	4887	4888	4889	4890	4895*	4897*	6413*	6414*			
2592	2718	2723	2817	2822	2827	4400	4404	4416	4419	5254*		
2531	2684	2785*	2788	2792								
2786	2789*											
2791*	2794											

SPACE 030243
SRETRN 014402
SRETR1 014422
SRETR3 014432

SENDOCT	014740	882	2869#	2983	3026	4168	4278												
SENDMG	015114	2871	2912#																
SENULL	015131	913	923	926	2874	2915#													
SEOP	014702	2844	2858#																
SEOPCT	014732	882*	2866#	2870	2983*	3026*	4168*	4278*											
SERFLG	001103	341#	4536*	4559	4566	4588	4590*	4607											
SERMAX	001115	347#	4607																
SERORR	025650	876	2977	3020	4162	4272	4529#												
SERRPC	001116	348#	4481	4543*	4544*	4545	4559	6382	6384	6387	6390	6393	6396						
SERRTB	001270	426#	4489																
SERTY	025514	4474#	4548																
SERTTL	001112	345#	4542*	4559															
SFILLC	001154	364#	4647	4681															
SFILLS	001153	363#	4681																
SGDADR	001120	349#																	
SGDOAT	001124	351#																	
SGET42	014762	2875#																	
SHD =	000001	20	21																
SICNT	001104	342#	4596*	4597	4599*	4606													
SILLUP	027262	4867	4899#																
SITEMB	001114	346#	4478	4545*	4559														
SLF	001230	388#	4559	4681															
SLOOP	015106	2905	2909#																
SLPROR	001106	343#	896*	2997*	3040*	4182*	4242*	4292*	4602*	4604	4606								
SLPERR	001110	344#																	
SMAIL =	***** U	907	3008	3051	4193	4303	4551	4602	4632										
SMXCNT	026202	4600	4606#																
SNULL	001152	362#	4649	4681															
SNWTST=	000001	934#	972#	997#	1020#	1043#	1093#	1142#	1212#	1261#	1359#	1457#	1555#	1648#					
		1746#	1844#	1939#	1993#	2062#	2150#	2167#	2184#	2201#	2218#	2235#	2252#	2349#					
		3059#	3094#	3140#	3332#	3459#	3585#	3740#	3907#	4047#									
SOCNT	026642	4715*	4744*	4757#															
SOMODE	026644	4710*	4714*	4719	4722*	4733*	4759#												
SOVER	026166	4575	4586	4598	4603#														
SPASS	001100	339#	2863*	2864*	2872	2911	4594	4607											
SPWRAD	027250	4896#																	
SPWRDN	027126	880	2981	3024	4166	4276	4867#	4891											
SPWRMG	027244	4894#																	
SPWRUP	027174	4876	4881#																
SQUES	001226	386#	4559	4681															
SROCHR=	***** U	4861																	
SRODEC=	***** U	4861																	
SROLIN=	***** U	4861																	
SRODOCT=	***** U	4861																	
SREGAD	001156	366#																	
SREGO	001160	368#																	
SREG1	001162	369#																	
SREG2	001164	370#																	
SREG3	001166	371#																	
SREG4	001170	372#																	
SREG5	001172	373#																	
SREG6	001174	374#	4530*	6382	6384	6387	6390	6393	6396										
SREG7	001176	375#	4531*	6382	6384	6387	6390	6393	6396										
SPTRN	015104	884	886*	891*	2981	2906#	2985	2987*	2992*	3028	3030*	3035*	4170	4172*					

	4177*	4280	4282*	4287*										
\$R2A = ***** U	4861													
\$SAVRE = ***** U	4861													
\$SAVR6 027266	4875*	4881	4882*	4883*	4901*									
\$SCOPE 026024	874	2975	3018	4160	4270	4573*								
\$SETUP= 000037	328*	874	876	878	880	882	883	884	896	2861	2975	2977	2979	
	2981	2983	2984	2985	2997	3018	3020	3022	3024	3026	3027	3028	3040	
	4160	4162	4164	4166	4168	4169	4170	4182	4270	4272	4274	4276	4278	
	4279	4280	4292	4554										
\$STUP = 177777	328*													
\$SVLAD 026156	4583	4601*												
\$SVPC = 000204	301*	306												
\$SWR = 176000	8*	20	384	385	883	884	896	897	938	976	1001	1024	1047	
	1097	1146	1216	1265	1363	1461	1559	1652	1750	1848	1943	1997	2066	
	2154	2171	2189	2205	2222	2239	2256	2353	2854	2862	2876	2892	2911	
	2984	2985	2997	2998	3027	3028	3040	3041	3063	3098	3142	3336	3463	
	3589	3744	3911	4051	4169	4170	4182	4183	4279	4280	4292	4293	4522	
	4523	4524	4525	4526	4539	4546	4551	4557	4559	4567	4568	4569	4570	
	4574	4586	4588	4589	4590	4591	4592	4603	4606	4897				
\$SWRMK= 000000	4570													
\$TBIT 015112	895*	2902*	2911*	2996*	3039*	4181*	4291*							
\$TIMES 001220	384*	883*	2862*	2984*	3027*	4169*	4279*	4591*	4597	4600*	4606			
\$TKB 001144	359*													
\$TKS 001142	358*													
\$TMP0 001200	376*	2506*	2507*	2508	2541*	2542*	2543	2555*	2556*	2557	2584*	2585*	2586	
	4363*	4364*	4366	4532*	6382	6384	6387	6390	6393	6396				
\$TMP1 001202	377*	4533*	6382	6384	6387	6390	6393	6396						
\$TMP2 001204	378*	4534*	6384	6390	6393									
\$TMP3 001206	379*	4535*	6387	6390	6393									
\$TMP4 001210	380*	958*	967*	2141*	2146*	2162*	2179*	2196*	2213*	2230*	2247*	2344*	2491*	
	2499*	2651*	2657*	2668*	6393	6396								
\$TMP5 001212	381*	2400*	2519*	2575	2712	2841*								
\$TMP6 001214	382*	1157*	1158*	1159	1182*	1183*	1184	1227*	1228*	1229	1276*	1277*	1278	
	1301*	1302*	1303	1315*	1316*	1317	1374*	1375*	1376	1399*	1400*	1401	1413*	
	1414*	1415	1472*	1473*	1474	1497*	1498*	1499	1511*	1512*	1513	1570*	1571*	
	1572	1595*	1596*	1597	1609*	1610*	1611	1663*	1664*	1665	1688*	1689*	1690	
	1702*	1703*	1704	1761*	1762*	1763	1786*	1787*	1788	1800*	1801*	1802	1859*	
	1860*	1861	1884*	1885*	1886	1898*	1899*	1900	1954*	1955*	1956	2008*	2009*	
	2010	2033*	2034*	2035	2077*	2078*	2079	2102*	2103*	2104	2267*	2268*	2269	
	2292*	2293*	2294	2306*	2307*	2308	2440*	2441*	2442	3155*	3156*	3157	3180*	
	3181*	3182	3218*	3219*	3220	3243*	3244*	3245	3279*	3280*	3281	3304*	3305*	
	3306	3369*	3370*	3371	3394*	3395*	3396	3417*	3418*	3419	3442*	3443*	3444	
	3496*	3497*	3498	3521*	3522*	3523	3543*	3544*	3545	3568*	3569*	3570	3650*	
	3651*	3652	3675*	3676*	3677	3698*	3699*	3700	3723*	3724*	3725	3805*	3806*	
	3807	3830*	3831*	3832	3853*	3854*	3855	3878*	3879*	3880	3959*	3960*	3961	
	3984*	3985*	3986	4105*	4106*	4107	4130*	4131*	4132	4214*	4215*	4216	4231*	
	4232*	4233												
\$TMP7 001216	383*													
\$TN = 000044	20*	934	938*	972	976*	997	1001*	1020	1024*	1043	1047*	1093	1097*	
	1142	1146*	1212	1216*	1261	1265*	1359	1363*	1457	1461*	1555	1559*	1648	
	1652*	1746	1750*	1844	1848*	1939	1943*	1993	1997*	2062	2066*	2150	2154*	
	2167	2171*	2184	2188*	2201	2205*	2218	2222*	2235	2239*	2252	2256*	2349	
	2353*	3059	3063*	3094	3098*	3140	3142*	3332	3336*	3459	3463*	3585	3589*	
	3740	3744*	3907	3911*	4047	4051*								

270089.P11

CROSS REFERENCE TABLE

200	893	2519	2548	2565	2571	2575	2576	2668	2682	2693	2702	2708	2712	2713	2795
	2803	2804	2805	2834	2994	3037	4179	4199	4289	4489	4636	4711	4721	4794	
25	4186	4487	4488	4843											
25	4199	4800													
25	4326	945	949	957	966	980	987	993	1003	1007	1012	1016	1026	1030	1035
25	941	945	949	957	966	980	987	993	1003	1007	1012	1016	1026	1030	1035
25	1339	1072	1089	1105	1335	1433	1531	1629	1722	1820	2136	2140	2145	2161	2178
25	1339	1072	1089	1105	1335	1433	1531	1629	1722	1820	2136	2140	2145	2161	2178
25	195	2212	2229	2246	2343	2373	2405	2464	2470	2482	2490	2498	2517	2610	2623
25	195	2212	2229	2246	2343	2373	2405	2464	2470	2482	2490	2498	2517	2610	2623
25	267	2631	2635	2639	2647	2650	2656	2666	2740	2749	2762	2766	2770	2774	2877
25	267	2631	2635	2639	2647	2650	2656	2666	2740	2749	2762	2766	2770	2774	2877
25	2885	2918	2923	2933	2949	3090	3115	3128	3136	3190	3205	3266	3320	3328	3356
25	2885	2918	2923	2933	2949	3090	3115	3128	3136	3190	3205	3266	3320	3328	3356
25	3455	3483	3581	3607	3609	3637	3736	3760	3762	3764	3792	3891	3903	3919	3928
25	3455	3483	3581	3607	3609	3637	3736	3760	3762	3764	3792	3891	3903	3919	3928
25	3933	3938	4000	4004	4039	4043	4060	4062	4064	4092	4204	4340	4374	4377	4381
25	3933	3938	4000	4004	4039	4043	4060	4062	4064	4092	4204	4340	4374	4377	4381
25	4451	4491	4496	4509	4537	4540	4589	4595	4639	4738					
25	4451	4491	4496	4509	4537	4540	4589	4595	4639	4738					
25	2617	2844	4598	4673											
25	2617	2844	4598	4673											
25	2564	2570	2701	2707	2794	2867	4395	4745	4808						
25	2564	2570	2701	2707	2794	2867	4395	4745	4808						
25	1193	1198	1302	1400	1498	1596	1689	1787	1885	2034	2103	2293	2441	2507	2542
25	1193	1198	1302	1400	1498	1596	1689	1787	1885	2034	2103	2293	2441	2507	2542
25	2556	2585	2683	2864	2899	3181	3244	3305	3395	3443	3522	3569	3676	3724	3831
25	2556	2585	2683	2864	2899	3181	3244	3305	3395	3443	3522	3569	3676	3724	3831
25	3879	3985	4131	4215	4314	4364	4735	4897							
25	3879	3985	4131	4215	4314	4364	4735	4897							
25	978	1010	1033	1158	1228	1277	1316	1375	1414	1473	1512	1571	1610	1664	1703
25	978	1010	1033	1158	1228	1277	1316	1375	1414	1473	1512	1571	1610	1664	1703
25	1762	1801	1860	1899	1955	2009	2078	2268	2307	2807	2904	3156	3219	3280	3370
25	1762	1801	1860	1899	1955	2009	2078	2268	2307	2807	2904	3156	3219	3280	3370
25	3418	3497	3544	3651	3699	3806	3854	3960	4106	4232	4343	4740	4741	4802	4803
25	3418	3497	3544	3651	3699	3806	3854	3960	4106	4232	4343	4740	4741	4802	4803
25	4328	4478													
25	4328	4478													
25	927	1071	1088	1104	2123	2127	2131	2135	2372	2404	2421	2463	2465	2469	2472
25	927	1071	1088	1104	2123	2127	2131	2135	2372	2404	2421	2463	2465	2469	2472
25	2481	2485	2528	2578	2609	2611	2622	2626	2630	2634	2638	2641	2646	2677	2715
25	2481	2485	2528	2578	2609	2611	2622	2626	2630	2634	2638	2641	2646	2677	2715
25	2739	2741	2744	2748	2750	2761	2765	2769	2773	2776	2789	2900	2917	2922	2928
25	2739	2741	2744	2748	2750	2761	2765	2769	2773	2776	2789	2900	2917	2922	2928
25	2932	2940	2948	3085	3089	3105	3114	3123	3127	3135	3189	3204	3257	3265	3315
25	2932	2940	2948	3085	3089	3105	3114	3123	3127	3135	3189	3204	3257	3265	3315
25	3319	3327	3343	3351	3355	3470	3478	3482	3602	3606	3617	3625	3632	3636	3759
25	3319	3327	3343	3351	3355	3470	3478	3482	3602	3606	3617	3625	3632	3636	3759
25	3761	3772	3780	3787	3791	3918	3927	3932	3937	3999	4003	4015	4019	4023	4038
25	3761	3772	3780	3787	3791	3918	3927	3932	3937	3999	4003	4015	4019	4023	4038
25	4042	4059	4061	4072	4080	4087	4091	4203	4239	4341	4372	4376	4380	4390	4430
25	4042	4059	4061	4072	4080	4087	4091	4203	4239	4341	4372	4376	4380	4390	4430
25	4457	4539	4546	4574	4592										
25	4457	4539	4546	4574	4592										
25	4661														
25	4661														
25	2550	2695													
25	2550	2695													
25	2756	4652	4746	4791	4807										
25	2756	4652	4746	4791	4807										
25	1082	1123	1134	1202	1337	1341	1348	1435	1439	1446	1533	1537	1544	1631	1635
25	1082	1123	1134	1202	1337	1341	1348	1435	1439	1446	1533	1537	1544	1631	1635
25	1642	1724	1728	1735	1822	1826	1833	1920	1927	2116	2120	2325	2479	2505	2514
25	1642	1724	1728	1735	1822	1826	1833	1920	1927	2116	2120	2325	2479	2505	2514
25	2540	2554	2583	2605	2663	2735	2786	2835	2903	2925	3111	3132	3197	3201	3254
25	2540	2554	2583	2605	2663	2735	2786	2835	2903	2925	3111	3132	3197	3201	3254
25	3262	3324	3348	3475	3622	3777	4029	4077	4354	4798					
25	3262	3324	3348	3475	3622	3777	4029	4077	4354	4798					
25	872	916	928	1076	1125	1979	2124	2128	2132	2414	2422	2466	2473	2486	2509
25	872	916	928	1076	1125	1979	2124	2128	2132	2414	2422	2466	2473	2486	2509
25	2512	2521	2529	2535	2544	25-7	2558	2561	2573	2579	2612	2642	2661	2670	2678
25	2512	2521	2529	2535	2544	25-7	2558	2561	2573	2579	2612	2642	2661	2670	2678
25	2688	2692	2698	2710	2716	2742	2745	2751	2777	2790	2812	2901	2929	2941	2973
25	2688	2692	2698	2710	2716	2742	2745	2751	2777	2790	2812	2901	2929	2941	2973
25	3016	3078	3082	3086	3107	3124	3258	3316	3344	3352	3471	3479	3603	3618	3626
25	3016	3078	3082	3086	3107	3124	3258	3316	3344	3352	3471	3479	3603	3618	3626
25	3633	3773	3781	3788	4016	4020	4024	4073	4081	4088	4158	4268	4330	4342	4358
25	3633	3773	3781	3788	4016	4020	4024	4073	4081	4088	4158	4268	4330	4342	4358
25	4361	4367	4370	4391	4393	4428	4431	4458	4479	4501	4547	4555	4575	4593	4633
25	4361	4367	4370	4391	4393	4428	4431	4458	4479	4501	4547	4555	4575	4593	4633
25	4641	4648	4662	4669	4736	4796	4884								
25	4641	4648	4662	4669	4736	4796	4884								
25	953	1068	1085	1138	1206	1247									
25	953	1068	1085	1138	1206	1247									
25	1712	1744	1810	1842	1908	1936	2319	1325	1357	1423	1455	1521	1553	1619	1646
25	1712	1744	1810	1842	1908	1936	2319	1325	1357	1423	1455	1521	1553	1619	1646
25	4008	4071	4085	4352	4356	4383	4403	4423	4448	4552	4627	4666	4734	4782	4812
25	4008	4071	4085	4352	4356	4383	4403	4423	4448	4552	4627	4666	4734	4782	4812
25	892	901	914	924	962	982	1074	1087	1127	1200	1248	1339	1355	1437	1453
25	892	901	914	924	962	982	1074	1087	1127	1200	1248	1339	1355	1437	1453
25	1535	1551	1633	1644	1726	1742	1824	1840	1918	1934	1986	2042	2055	2111	2362
25	1535	155													

01A	870	883	888	895	985	1052	1065	1100	1107	1109	1149	1164	1174	1208	1210
	1219	1234	1251	1253	1268	1283	1293	1307	1331	1333	1343	1345	1366	1381	1391
	1405	1429	1431	1441	1443	1464	1479	1489	1503	1527	1529	1539	1541	1562	1577
	1587	1601	1625	1627	1637	1639	1655	1670	1680	1694	1718	1720	1730	1732	1753
	1768	1778	1792	1816	1818	1828	1830	1851	1865	1876	1890	1914	1916	1922	1924
	1946	1961	1977	1985	1989	1990	2000	2015	2025	2058	2059	2069	2084	2094	2113
	2154	2171	2179	2188	2205	2222	2230	2239	2259	2274	2284	2298	2327	2329	2337
	2339	2368	2400	2401	2402	2403	2432	2447	2474	2491	2577	2651	2714	2840	2841
	2861	2862	2879	2892	2971	2984	2989	2996	3010	3014	3027	3032	3039	3147	3162
	3172	3210	3225	3235	3271	3286	3296	3361	3376	3386	3409	3424	3434	3488	3503
	3513	3535	3550	3560	3642	3657	3667	3690	3705	3715	3797	3812	3822	3845	3860
	3870	3951	3966	3976	4097	4112	4122	4156	4169	4174	4181	4206	4223	4266	4279
	4284	4291	4316	4335	4336	4337	4345	4477	4591	4725	4785	4788	4882	5410	6411
	6412														
01B	4338	4590	4670	4814											
	871	904	940	955	963	979	986	392	1002	1006	1015	1025	1029	1038	1204
	1250	1336	1346	1356	1434	1444	1454	1532	1542	1552	1630	1640	1645	1723	1733
	2274	1821	1831	1841	1925	1935	1988	2043	2056	2112	2139	2144	2159	2194	2210
	2245	2341	2342	2365	2497	2508	2511	2513	2520	2543	2546	2549	2557	2560	2562
	2569	2572	2597	2655	2662	2669	2694	2700	2706	2709	2836	2843	2860	2944	2952
	2972	3005	3015	3048	3252	3314	3406	3453	3454	3532	3579	3580	3608	3687	3734
	3735	3763	3842	3889	3890	3997	4013	4063	4142	4157	4190	4267	4300	4360	4366
	4369	4427	4450	4554	4584	4597	4806								
01C	2660	2691	2697	2728	4357	4384	4638	4640	4647	4668	4672				
01D	2375	2902	2947												
01E	4344														
01F	1124	2821	2826	2865	4329	4401	4485								
01G	4651	4654	4733	4744											
01H	173														
01I	290	2600	2731	2919	3075	3105	3121	3342	3469	3601	3916	3924	4036	4141	4197
01J	4202	4312	4333	4424	4462	4553	4556	4628	4877	4899					
01K	1066	1075	1103	1978	2359	2468	2516	2518	2537	2665	2667	2690	2798	2813	2818
	2823	2833	2842	2863	3076	3249	3450	3576	3614	3731	3769	3896	3933	4069	4327
	4349	4350	4359	4368	4389	4394	4397	4425	4542	4596	4739	4747	4792	4583	
01L	4536	4601	4674												
01M	174														
01N	295	315	318	322	326	2333	2376	2483	2488	2531	2601	2620	2644	2648	2684
	2757	2759	2779	2910	2950	2951	2954	3011	3904	4145	4244	4429	4432	4903	
	909	938	1048	1099	1146	1216	1265	1363	1461	1559	1652	1750	1848	1943	1997
	2066	2256	2353	2427	2467	2522	2526	2530	2580	2616	2671	2675	2717	2755	2793
	2887	3056	3065	3101	3144	3338	3464	3594	3747	3911	3923	4054	4193	4308	4334
	4446	4548	4646	4653	4660										
01O	869	873	874	875	876	877	878	879	880	881	882	884	885	886	887
	889	891	894	896	897	898	899	902	903	905	906	907	919	919	929
	930	958	967	991	1001	1024	1049	1050	1053	1054	1056	1057	1059	1060	1061
	1062	1077	1078	1101	1102	1110	1111	1113	1114	1116	1117	1118	1119	1129	1130
	1147	1150	1151	1153	1154	1156	1157	1157	1160	1165	1166	1168	1169	1171	1172
	1175	1176	1178	1175	1181	1182	1184	1185	1188	1189	1190	1194	1195	1209	1217
	1220	1221	1223	1224	1226	1227	1229	1230	1235	1236	1238	1239	1241	1242	1243
	1244	1245	1252	1266	1269	1270	1272	1273	1275	1276	1278	1279	1284	1285	1287
	1288	1290	1291	1294	1295	1297	1298	1300	1301	1303	1304	1308	1309	1311	1312
	1314	1315	1317	1318	1321	1322	1323	1327	1328	1332	1344	1349	1351	1364	1367
	1368	1370	1371	1372	1374	1376	1377	1382	1383	1385	1386	1388	1389	1392	1393
	1395	1396	1398	1399	1401	1402	1406	1407	1409	1410	1412	1413	1415	1416	1419

1420	1421	1425	1426	1430	1442	1447	1449	1462	1465	1466	1468	1469	1471	1472
1474	1475	1480	1481	1483	1484	1486	1487	1490	1491	1493	1494	1496	1497	1499
1500	1504	1505	1507	1508	1510	1511	1513	1514	1517	1518	1519	1523	1524	1528
1540	1545	1547	1560	1563	1564	1566	1567	1569	1570	1572	1573	1578	1579	1581
1582	1584	1595	1588	1589	1591	1592	1594	1595	1597	1598	1602	1603	1605	1606
1608	1609	1611	1612	1615	1616	1617	1621	1622	1626	1638	1643	1653	1656	1657
1659	1660	1662	1663	1665	1666	1671	1672	1674	1675	1677	1678	1681	1682	1684
1685	1687	1688	1690	1691	1695	1696	1698	1699	1701	1702	1704	1705	1708	1709
1700	1714	1715	1719	1731	1736	1738	1751	1754	1755	1757	1758	1760	1761	1763
1764	1769	1770	1772	1773	1775	1776	1779	1780	1782	1783	1785	1786	1788	1789
1793	1794	1796	1797	1799	1900	1802	1803	1806	1807	1808	1812	1813	1817	1829
1834	1836	1849	1852	1853	1855	1856	1858	1859	1861	1862	1867	1868	1870	1871
1872	1874	1877	1878	1880	1881	1883	1884	1886	1887	1891	1892	1894	1895	1897
1898	1900	1901	1904	1905	1906	1910	1911	1915	1923	1928	1930	1944	1947	1948
1950	1951	1953	1954	1956	1957	1962	1963	1965	1966	1968	1969	1970	1971	1974
1975	1976	1981	1982	1991	1998	2001	2002	2004	2005	2007	2008	2010	2011	2016
2017	2019	2020	2022	2023	2026	2027	2029	2030	2032	2033	2035	2036	2039	2040
2041	2044	2045	2046	2051	2052	2050	2067	2070	2071	2073	2074	2076	2077	2079
2080	2085	2086	2088	2089	2091	2092	2095	2096	2098	2099	2101	2102	2104	2105
2108	2109	2110	2114	2141	2146	2151	2162	2173	2190	2196	2207	2213	2224	2241
2247	2257	2260	2261	2263	2264	2266	2267	2269	2270	2275	2276	2278	2279	2281
2282	2285	2286	2288	2289	2291	2292	2294	2295	2299	2300	2302	2303	2305	2306
2308	2309	2312	2315	2317	2321	2322	2328	2338	2344	2354	2355	2357	2358	2367
2406	2407	2408	2410	2411	2412	2416	2417	2423	2424	2430	2433	2434	2436	2437
2439	2440	2442	2443	2448	2449	2451	2452	2454	2455	2456	2457	2458	2459	2460
2461	2462	2471	2494	2499	2502	2505	2515	2541	2555	2574	2584	2586	2588	2593
2653	2657	2659	2664	2711	2747	2787	2791	2795	2796	2797	2819	2824	2831	2832
2868	2872	2876	2880	2884	2893	2894	2897	2898	2905	2936	2970	2974	2975	2976
2977	2978	2979	2980	2981	2982	2983	2985	2986	2987	2988	2990	2992	2995	2997
2998	2999	3000	3003	3004	3006	3007	3013	3017	3018	3019	3020	3021	3022	3023
3024	3025	3026	3028	3029	3030	3031	3033	3035	3038	3040	3041	3042	3043	3046
3047	3049	3050	3053	3054	3073	3074	3145	3148	3149	3151	3152	3154	3155	3157
3158	3163	3164	3166	3167	3169	3170	3173	3174	3176	3177	3179	3180	3182	3183
3186	3208	3211	3212	3214	3215	3217	3218	3220	3221	3226	3227	3229	3230	3232
3233	3236	3237	3239	3240	3242	3243	3245	3246	3269	3272	3273	3275	3276	3278
3279	3281	3282	3287	3288	3290	3291	3293	3294	3297	3298	3300	3301	3303	3304
3306	3307	3359	3362	3363	3365	3366	3368	3369	3371	3372	3377	3378	3380	3381
3383	3384	3387	3388	3390	3391	3393	3394	3396	3397	3400	3407	3410	3411	3413
3414	3416	3417	3419	3420	3425	3426	3428	3429	3431	3432	3435	3436	3438	3439
3441	3442	3444	3445	3448	3449	3486	3489	3490	3492	3493	3495	3496	3498	3499
3504	3505	3507	3508	3510	3511	3514	3515	3517	3518	3520	3521	3523	3524	3527
3533	3536	3537	3539	3540	3542	3543	3545	3546	3551	3552	3554	3555	3557	3558
3561	3562	3564	3565	3567	3568	3570	3571	3574	3575	3612	3613	3640	3643	3644
3646	3647	3649	3650	3652	3653	3658	3659	3661	3662	3664	3665	3669	3669	3671
3672	3674	3675	3677	3678	3681	3688	3691	3692	3694	3695	3697	3698	3700	3701
3706	3707	3709	3710	3712	3713	3716	3717	3719	3720	3722	3723	3725	3726	3729
3730	3767	3768	3795	3798	3799	3801	3802	3804	3805	3807	3808	3813	3814	3816
3817	3819	3820	3823	3824	3826	3827	3829	3830	3832	3833	3836	3843	3846	3847
3849	3850	3852	3853	3855	3856	3861	3862	3864	3865	3867	3868	3871	3872	3874
3875	3877	3878	3880	3881	3884	3885	3894	3952	3953	3955	3956	3959	3959	3961
3962	3967	3968	3970	3971	3973	3974	3977	3978	3980	3981	3983	3984	3986	3987
3990	3991	3992	4011	4067	4068	4095	4098	4099	4101	4102	4104	4105	4107	4108
4113	4114	4116	4117	4119	4120	4123	4124	4126	4127	4129	4130	4132	4133	4135
4155	4159	4160	4161	4162	4163	4164	4165	4166	4167	4168	4170	4171	4172	4173

	4175	4177	4180	4182	4183	4184	4185	4188	4189	4191	4192	4198	4207	4208	4210
	4211	4213	4214	4216	4217	4224	4225	4227	4228	4230	4231	4233	4234	4237	4242
	4265	4269	4270	4271	4272	4273	4274	4275	4276	4277	4278	4280	4281	4282	4283
	4285	4287	4290	4292	4293	4294	4295	4298	4299	4301	4302	4303	4313	4315	4324
	4346	4347	4348	4363	4398	4411	4417	4420	4449	4476	4481	4490	4495	4500	4502
	4506	4530	4531	4532	4533	4534	4535	4538	4543	4579	4580	4582	4585	4599	4600
	4602	4603	4604	4630	4631	4635	4649	4708	4716	4717	4718	4724	4731	4749	4750
	4751	4752	4753	4775	4776	4777	4778	4779	4780	4781	4786	4789	4809	4815	4816
	4817	4818	4819	4821	4822	4839	4840	4844	4867	4868	4869	4870	4871	4872	4873
	4874	4875	4876	4881	4885	4886	4887	4888	4889	4890	4891	4892	4895	6405	6406
	6407	6408	6409	6413	6414										
MOVE	2157	2174	2208	2225	2719	2724	4405	4545	4632	4659	4667	4709	4710	4713	4714
	4715	4719	4722	4723	4742	4784	4787	4801	4804	4813	4842				
NEG	2192	2243	4720	4783											
NOP	2049	2888	2889	2890											
RESET	939	2878	2886												
ROL	4726	4728	4729	4730	4732										
ROLB	4318	4320		4322	4323										
ROR	4317	4325	4321	4322	4323										
RORB	4319														
RORB	399	886	920	931	1058	1063	1079	1115	1120	1131	1155	1161	1170	1180	1186
	1196	1225	1231	1240	1274	1280	1289	1299	1305	1313	1319	1329	1372	1378	1387
	1397	1403	1411	1417	1427	1470	1476	1485	1495	1501	1509	1515	1525	1568	1574
	1583	1593	1599	1607	1613	1623	1661	1667	1676	1686	1692	1700	1706	1716	1759
	1765	1774	1784	1790	1798	1804	1914	1857	1863	1872	1882	1888	1896	1902	1912
	1952	1958	1967	1972	1983	2006	2012	2021	2031	2037	2047	2053	2075	2081	2090
	2100	2106	2265	2271	2280	2290	2296	2304	2310	2323	2418	2425	2438	2444	2453
	2799	2906	2987	3030	3153	3159	3168	3178	3184	3194	3216	3222	3231	3241	3247
	3277	3283	3292	3302	3306	3367	3373	3382	3392	3398	3415	3421	3430	3440	3446
	3494	3500	3509	3519	3525	3541	3547	3556	3566	3572	3648	3654	3663	3673	3679
	3696	3702	3711	3721	3727	3803	3809	3818	3828	3834	3851	3857	3866	3875	3882
	3957	3963	3972	3982	3988	4103	4109	4118	4126	4134	4172	4212	4218	4229	4235
	4282	4558	4605	4637	4754	4823	4898	6415							
	2808	2828	2837	2845	4453	4459	4504	4676	4845	6417					
	890	891	2991	2992	3034	3035	4176	4177	4286	4287					
	2495	2496	2551	2562	2568	2654	2679	2680	2681	2696	2699	2705	2806	4544	4790
	1055	1112	1152	1167	1177	1222	1237	1271	1286	1296	1310	1369	1384	1394	1428
	1467	1482	1492	1506	1565	1580	1590	1604	1658	1673	1683	1697	1756	1771	1781
	1795	1854	1869	1879	1893	1949	1964	2003	2018	2028	2072	2097	2097	2262	2277
	2287	2301	2435	2450	2895	3150	3165	3175	3213	3228	3238	3274	3299	3299	3364
	3379	3389	3412	3427	3437	3491	3506	3516	3538	3553	3563	3645	3660	3670	3693
	3708	3718	3800	3815	3825	3848	3863	3873	3954	3969	3979	4100	4115	4125	4209
	4226	4847	4857	4858	4859	4860									
	900	915	944	948	952	1011	1034	1084	1137	1205	1334	1347	1432	1445	1530
	1543	1628	1641	1721	1734	1819	1832	1926	2119	2176	2227	2413	2489	2503	2523
	2527	2534	2536	2538	2552	2581	2598	2649	2687	2729	2911	2959	2896	2924	3001
	3044	3081	3110	3131	3200	3261	3323	3347	3474	3621	3629	3776	3784	3902	3941
	4007	4028	4076	4084	4186	4296	4353	4355	4382	4392	4402	4415	4422	4508	4551
	4581	4594	4634	4642	4663	4737	4795	4805	4841						
	1067	1081	1122	1133	1191	1201	1246	1324	1340	1422	1436	1520	1536	1618	1634
	1711	1727	1809	1825	1907	1919	2115	2318	2334	2360	2478	2672	2676	2589	3077
	3196	3253	3615	3770	4070	4351	4409	4447	4588	4626	4665	4797	4811		
	2937														
	366	387	5253	5255	5265	5270	5276	5283	5287	5302	5398	5410	5421	5431	5440

	5446	5456	5463	5488	5497	5526	5535	5573	5583	5531	5600	5614	5622	5631	5645
	5658	5666	5683	5695	5708	5722	5740	5751	5763	5774	5784	5814	5826	5837	5899
	5911	5926	5937	5948	5958	5969	5985	6035	6069	6082	6095	6109	6122	6136	6149
.AS012	6219	6239	6253	6265	6321	6342	6362								
	385	388	2912	4512	5254	5256	5258	5294	5310	5319	5324	5329	5337	5346	5355
	5363	5368	5375	5380	5385	5392	5405	5419	5471	5477	5481	5505	5511	5519	5521
	5523	5542	5551	5556	5562	5568	5608	5639	5652	5672	5690	5703	5715	5720	5734
	5747	5758	5770	5780	5791	5797	5805	5822	5832	5844	5853	5858	5866	5875	5881
	5890	5907	5918	5933	5944	5955	5965	5976	5992	5998	6007	6015	6025	6042	6047
	6058	6077	6090	6104	6118	6131	6145	6158	6165	6176	6183	6190	6200	6209	6227
	6231	6247	6259	6273	6278	6289	6296	6304	6312	6332	6353	6371			
	4828														
	340	341	346	347	362	363	364	365	1353	1354	1451	1452	1549	1550	1740
	1741	1838	1839	1932	1933	2590	2591	2595	2596	2721	2722	2726	2727	2915	4407
	4408	4413	4414	4438	4439	4755	4756	4757	4758	5004	5005	5006	5007	5008	5009
	5010	5011	5012	5013	5014	5015	5016	5017	5018	5019	5020	5021	5022	5023	5024
	5025	5026	5027	5028	5029	5030	5031	5032	5033	5034	5035	5036	5037	5038	5039
	5040	5041	5042	5043	5044	5045	5046	5047	5048	5049	5050	5051	5052	5053	5054
	5055	5056	5057	5058	5059	5060	5061	5062	5063	5064	5065	5066	5067	5068	5069
	5070	5071	5072	5073	5074	5075	5076	5077	5078	5079	5080	5081	5082	5083	5172
	5173	5174	5175	5176	5177	5178	5179	5180	5181	5182	5183	5184	5185	5186	5187
	5188	5189	5190	5191	5192	5193	5194	5195	5196	5197	5198	5199	5200	5201	5202
	5203	5204	5205	5206	5207	5208	5209	5210	5211	5212	5213	5214	5215	5216	5217
	5218	5219	5220	5221	5222	5223	5224	5225	5226	5227	5228	5229	5230	5231	5232
	5233	5234	5235	5236	5237	5238	5239	5240	5241	5242	5243	5244	5245	5246	5247
	5248	5249	5250	5251											
.ENABL															
.ENAC	6439														
.ENCC	15	140	158	173	259	273	296	298	304	306	328	331	338	340	366
	375	384	385	386	390	411	873	874	876	878	880	882	883	884	896
	897	907	935	936	937	938	972	974	975	976	998	999	1000	1001	1021
	1022	1023	1024	1044	1045	1046	1047	1094	1095	1096	1097	1143	1144	1145	1146
	1213	1214	1215	1216	1262	1263	1264	1265	1339	1355	1360	1361	1362	1363	1437
	1453	1458	1459	1460	1461	1535	1551	1556	1557	1558	1559	1633	1644	1649	1650
	1651	1652	1726	1742	1747	1748	1749	1750	1824	1840	1845	1846	1847	1848	1917
	1918	1934	1940	1941	1942	1943	1994	1995	1996	1997	2063	2064	2065	2066	2151
	2152	2153	2154	2168	2169	2170	2171	2185	2186	2187	2188	2202	2203	2204	2205
	2219	2220	2221	2222	2236	2237	2238	2239	2253	2254	2255	2256	2350	2351	2352
	2353	2382	2388	2849	2852	2853	2854	2855	2857	2861	2867	2870	2871	2875	2883
	2910	2911	2912	2915	2916	2974	2975	2977	2979	2981	2983	2984	2985	2997	2998
	3008	3017	3018	3020	3022	3024	3026	3027	3028	3040	3041	3051	3060	3061	3062
	3053	3095	3096	3097	3098	3141	3142	3333	3334	3335	3336	3460	3461	3462	3463
	3586	3587	3588	3589	3741	3742	3743	3744	3748	3909	3910	3911	4048	4049	4050
	4051	4159	4160	4162	4164	4166	4168	4169	4170	4182	4183	4193	4269	4270	4272
	4274	4276	4278	4279	4290	4292	4293	4303	4467	4485	4514	4516	4522	4536	4543
	4548	4549	4550	4551	4557	4558	4559	4561	4567	4570	4574	4576	4587	4598	4590
	4592	4596	4601	4602	4603	4606	4607	4609	4632	4683	4762	4831	4840	4843	4856
	4857	4858	4859	4860	4861	4863	4875	4885	4895	4897	4898	4902	6402	6404	6420
	6421	6422	6423	6424	6425	6431	6433	6435	6437						
.ES01V	173	174	176	191	192	221	222	223	224	225	226	227	228	229	230
	249	250	251	252	253	254	255	256	257	258					
.EVEN	3513	5678	6378												
	11	139	157	171	231	259	293	297	302	304	328	330	337	339	366
	376	384	385	389	390	869	873	874	876	878	880	882	883	884	896

	907	934	936	938	972	974	976	997	999	1001	1020	1022	1024	1043	1045
	1047	1093	1095	1097	1142	1144	1146	1212	1214	1216	1261	1263	1265	1359	1361
	1363	1457	1459	1461	1555	1557	1559	1648	1650	1652	1746	1748	1750	1844	1846
	1848	1939	1941	1943	1993	1995	1997	2062	2064	2066	2150	2152	2154	2167	2169
	2171	2184	2186	2188	2201	2203	2205	2218	2220	2222	2235	2237	2239	2252	2254
	2256	2349	2351	2353	2380	2386	2848	2852	2853	2854	2856	2857	2859	2866	2869
	2871	2875	2876	2892	2910	2911	2912	2970	2974	2975	2977	2979	2981	2983	2984
	2985	2997	3008	3013	3017	3018	3020	3022	3024	3026	3027	3028	3040	3051	3059
	3061	3063	3094	3096	3098	3140	3142	3332	3334	3336	3459	3461	3463	3585	3587
	3589	3740	3742	3744	3907	3909	3911	4047	4049	4051	4155	4153	4160	4162	4164
	4166	4168	4169	4170	4182	4193	4265	4269	4270	4272	4274	4276	4278	4279	4280
	4292	4303	4466	4484	4500	4515	4521	4530	4539	4546	4548	4549	4551	4554	4557
	4558	4559	4560	4566	4570	4574	4586	4588	4589	4590	4592	4594	4602	4603	4605
	4606	4607	4608	4632	4682	4761	4830	4839	4843	4847	4857	4858	4859	4860	4861
	4862	4875	4885	4893	4895	4897	4902	6401	6403	6419	6421	6423	6425	6431	6432
	6433	6434	6435	6436											
..FJ	1339	1437	1535	1633	1726	1824	1917								
..FJ	140	158	171	298	304	331	337	339	366	390	873	935	936	937	938
	972	974	975	976	998	999	1000	1001	1021	1022	1023	1024	1044	1045	1046
	1047	1094	1095	1096	1097	1143	1144	1145	1146	1213	1214	1215	1216	1262	1263
	1264	1265	1360	1361	1362	1363	1458	1459	1460	1461	1556	1557	1558	1559	1649
	1650	1651	1652	1747	1748	1749	1750	1845	1846	1847	1848	1940	1941	1942	1943
	1994	1995	1996	1997	2063	2064	2065	2066	2151	2152	2153	2154	2168	2169	2170
	2171	2185	2186	2187	2188	2202	2203	2204	2205	2219	2220	2221	2222	2236	2237
	2238	2239	2253	2254	2255	2256	2350	2351	2352	2353	2849	2856	2859	2867	2870
	2911	2974	3017	3060	3061	3062	3063	3095	3096	3097	3098	3141	3142	3233	3234
	3335	3336	3460	3461	3462	3463	3586	3587	3588	3589	3741	3742	3743	3744	3908
	3909	3910	3911	4048	4049	4050	4051	4159	4269	4467	4485	4514	4516	4521	4539
	4557	4558	4559	4561	4586	4588	4590	4606	4609	4683	4762	4831	4840	4863	4892
	6402	6404	6420	6422	6424	6433	6435	6437							
..FNE	1334	1350	1432	1448	1530	1546	1628	1644	1721	1737	1819	1835	1917	1929	
..IFT	4549	4592													
..IFTF	4548	4590													
..IF	10	15	20	21	290	389	874	876	892	883	884	896	897	2853	3361
	2862	2873	2911	2916	2975	2977	2983	2984	2985	2997	2998	3018	3020	3026	3027
	3028	3040	3041	4160	4162	4168	4169	4170	4182	4183	4270	4272	4278	4279	4280
	4292	4293	4482	4507	4522	4523	4524	4525	4526	4554	4559	4567	4568	4569	4570
	4591	4603	4606	4607	4681	4856	4857	4858	4859	4860					
..RP	328	390	934	972	997	1020	1043	1093	1142	1212	1261	1359	1457	1555	1648
	1746	1844	1939	1993	2062	2150	2167	2184	2201	2218	2235	2252	2349	2859	3059
	3094	3140	3332	3459	3585	3740	3907	4047	4530	4775	4815	4869	4885		
..ST	2	138	159	272	290	328	366	368	369	370	371	372	373	374	375
	376	377	378	379	380	381	382	383	384	865	897	934	938	972	976
	997	1001	1020	1024	1043	1047	1093	1097	1142	1146	1212	1216	1261	1265	1359
	1363	1457	1461	1555	1559	1648	1652	1746	1750	1844	1848	1939	1943	1993	1997
	2052	2066	2150	2154	2167	2171	2184	2188	2201	2205	2218	2222	2235	2239	2252
	2256	2349	2353	2379	2389	2393	2861	2967	2998	3041	3059	3063	3094	3098	3140
	3142	3332	3336	3459	3463	3585	3589	3740	3744	3907	3911	4047	4051	4147	4183
	4248	4293	4441	4554	4570	4847	4856	4857	4858	4859	4660	4861	4914	6420	6422
	6424	6432	6434	6436											
..MACRO	160	161	162	163	164	165	166	167	329	330	884	1261	2847	2985	3028
	4170	4280	4465	4847											
..MACRO	4	5	6	272	897	2998	3041	4183	4293	4293					
..MACRO		138	159	273	290	328	366	368	369	370	371	372	373	374	375

	376	377	378	379	380	381	382	383	384	865	897	934	938	972	976
	997	1001	1020	1024	1043	1047	1093	1097	1142	1146	1212	1216	1261	1265	1359
	1363	1457	1461	1555	1559	1648	1652	1746	1750	1844	1848	1939	1943	1993	1997
	2062	2066	2150	2154	2167	2171	2184	2188	2201	2205	2218	2222	2235	2239	2252
	2256	2349	2353	2379	2389	2393	2861	2967	2998	3041	3059	3063	3094	3098	3140
	3142	3332	3336	3459	3463	3585	3589	3740	3744	3907	3911	4047	4051	4147	4193
	4248	4293	4441	4554	4570	4847	4856	4857	4858	4859	4860	4861	4914	6420	6422
	6424	6432	6434	6436											
.P11	42	330	389												
.REF	290	368	376	2380	2386										
.SOL	43	138	169	284	294	299	332	412	865	934	972	997	1020	1043	1093
	1142	1212	1261	1359	1457	1555	1648	1746	1844	1939	1993	2062	2150	2167	2184
	2201	2218	2235	2252	2349	2393	2850	2967	3059	3094	3232	3459	3585	3740	3907
	4047	4147	4248	4441	4468	4517	4562	4610	4684	4763	4832	4848	4864	4914	
.SOL	10														
.SOL	290	291	292	305	309	310	339	342	343	344	345	348	349	350	351
	352	353	354	355	356	357	366	368	369	370	371	372	373	374	375
	376	377	378	379	380	381	382	383	2866	2869	4493	4498	4675	4759	4894
	4896	4906													

ERRORS DETECTED: 0

*DZCDBA,DZCDBA/CRF/SOL=DZCDBA
RUN-TIME: 21 19 3 SECONDS
CORE USED: 20K

