

CD11/CD20

CARD READER DIAGNOSTIC
MD-11-DZCDB-B

EP-DZCDB-B-DL-B

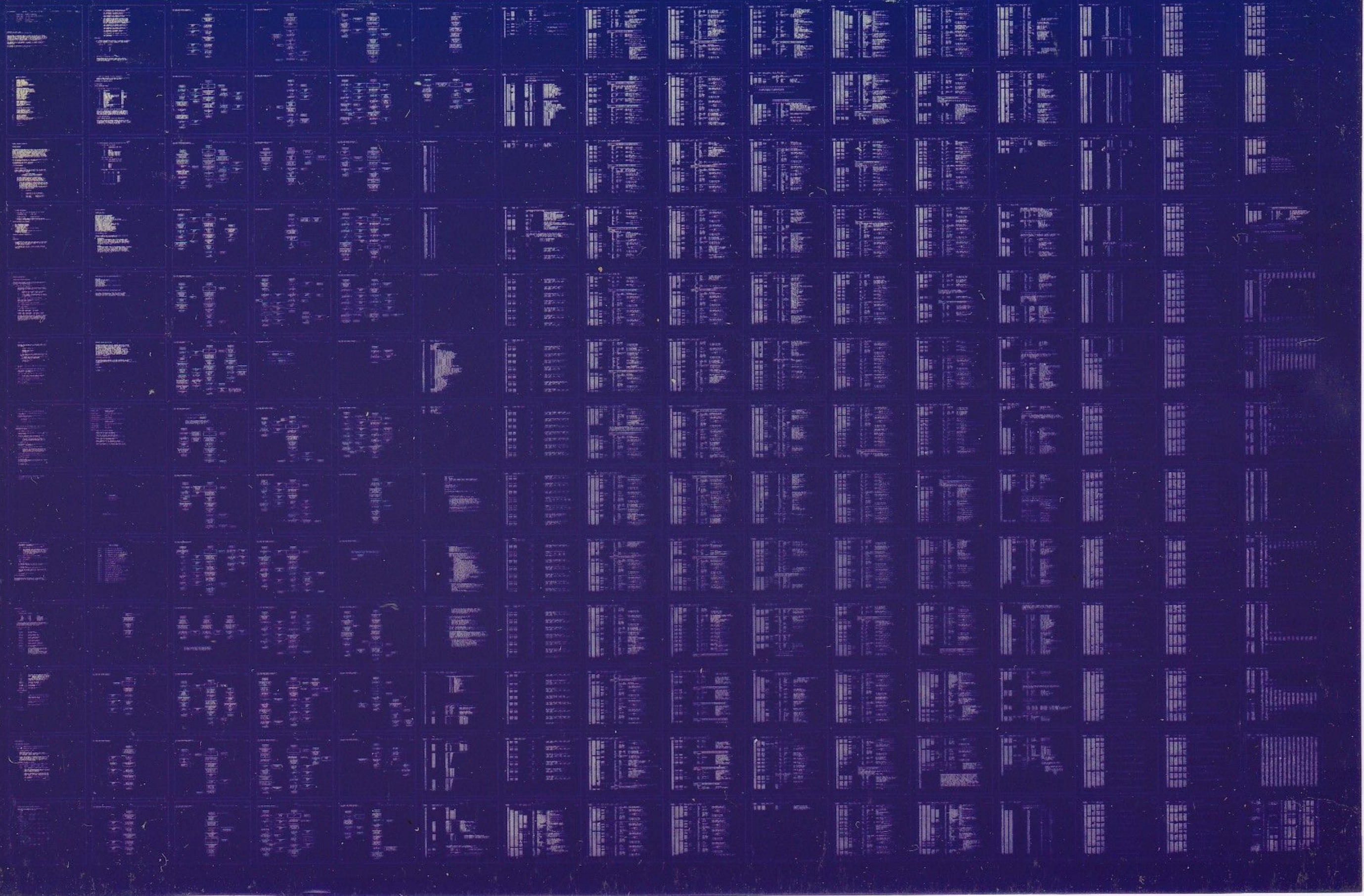
OCT 1977

COPYRIGHT © 76-1977

digital

FICHE 1 OF 2

MADE IN USA



CD11/CD20

CARD READER DIAGNOSTIC
MD-11-DZCDB-B

EP-DZCDB-B-DL-B

OCT 1977

COPYRIGHT © 76-1977

digital

FICHE 2 OF 2

MADE IN USA

This microfiche strip contains 16 frames of data. The frames are arranged vertically and contain various patterns of dots and lines, likely representing binary data or diagnostic test results. The data is too small to be legible in this image.

B01

EOF1DX000180411

00010000

770920 IDENTIFICATION P10 411

IZHOR1DZCDBBSEQ

00010000

770920
SEQ 0001

PRODUCT CODE: MAINDEC-11-DZCDB-B-D
PRODUCT NAME: CD11/CD20 CARD READER DIAGNOSTIC
DATE CREATED: JULY 1977
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: BRUCE BURGESS / ED RYAN

COPYRIGHT (C) 1976, 1977
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 M120J/RS1200 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/RS1200 |
| 6.4 | READING A SINGLE DATA PATTERN |
| 6.5 | LOOPING ON A SELECTED TEST |
| 7.0 | HISTORY |
| 8.0 | FLOW CHARTS |
| 9.0 | 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 DOCUMENTATION M1000, M200, M1200 OR AN RS1200 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 DOCUMENTATION MODEL CARD READER (MODELS M200, M1000, M1200, RS1200).

***** 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:

ALPHANUMERIC DECK

BINARY DECK

MAINDEC-89-D1B1-C

MAINDEC-89-D1B2-C

IN ADDITION TO SPARE CARDS FOR ERROR FUNCTION TESTING,
A SPECIAL MIS-REGISTERED CARD IS ALSO REQUIRED (FOR RS1200 ONLY)..

1.3 RELATED DOCUMENTS AND STANDARDS

-
- A. CD11 ENGINEERING DRAWINGS
 - B. PDP11 PERIPHERALS HANDBOOK
 - C. PDP11 PROCESSOR HANDBOOK
 - D. MAINDEC-11-DZQAC-B1
SYSMAC.SML DIAGNOSTIC UTILITIES PACKAGE
 - E. MAINDEC-11-DZQXA
'XXDP' USER'S GUIDE
 - F. DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS
PROGRAMMING PRACTICES
DOC. NO. 175-003-009-00

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.

2.1 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 TTY 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.B

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.B

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

MAINDEC-11-DZCDB REV.B

ETC.

C. ERROR FUNCTIONS FOR CARD READER MODEL M1200 OR RS1200

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 ERROR
FUNCTION TESTS"
3. ANSWER THE QUESTION "M1200 OR RS1200" BY TYPING
A Y FOR THE RS1200 OR AN N FOR THE M1200.

NOTE: SINCE ONLY THE RS1200 HAS THE CAPABILITY
TO DETECT A MIS-REGISTERED CARD, ONLY THIS
MODEL SHOULD ENTER THE MISTERED CARD TEST.

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

I01

(SW<11> MUST NOT BE SET), THEN PRESS 'CONTINUE'
*** HOWEVER *** (SW<14> MUST BE SET... MUST BE SET ...)

SEQ 0008

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 SWS<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 OR RS1200 |

2.4 EXECUTION TIMES

(TO BE DETERMINED)

ERROR INFORMATION

SEQ 0012

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.

- WHERE: (PC) REPRESENTS THE PROGRAM COUNTER LOCATION IN THE PROGRAM WHERE THE ERROR OCCURRED.
- (SP) REPRESENTS THE CURRENT POSITION OF THE STACK POINTER (GENERAL PURPOSE REGISTER 6)
- (CDS) REPRESENTS THE CURRENT CONTENTS OF THE CARD READER CONTROL STATUS REGISTER (177160)
- (CDO) REPRESENTS THE CURRENT CONTENTS OF THE DATA BUFFER. INFORMATION IS ONLY VALID DURING DATA TRANSFERS EXCEPT ON THE CD20 CARD READER WHICH USES THE DATA BUFFER REGISTER (177166) AS A 2ND STATUS REGISTER DURING NON-DATA TRANSFER PERIODS. (REFERENCE SECTION 1.2, PARAGRAPH A.)
- (CDC) REPRESENTS THE CURRENT CONTENTS OF THE CARD READER COLUMN COUNT REGISTER (177162)
- (CDA) REPRESENTS THE CURRENT CONTENTS OF THE CARD READER BUS ADDRESS REGISTER (177164)
- (PS) REPRESENTS THE CURRENT CONTENTS OF THE PROCESSOR STATUS REGISTER (177776)
- 'WAS' REPRESENTS THE INCORRECT VALUE FOUND
- 'SHB' REPRESENTS THE CORRECT VALUE THAT SHOULD HAVE BEEN FOUND

3.2 ERROR HALTS

- A. ALL UNUSED LOCATIONS FROM LOCATION 4 TO LOCATION 776 CONTAINS A +2 HALT SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS. LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS. I.E.,

| <u>LOCATION</u> | <u>CONTENTS</u> |
|-----------------|-----------------|
| 0 | HALT |
| 4 | 6 |
| 6 | HALT |
| 10 | 12 |
| 12 | HALT |
| | : |
| | : |
| | ETC. |

- B. WHEN SW<15>=1 INDICATING TO HALT ON ERROR
- C. DURING ERROR FUNCTION TESTING OF READ CHECK, WHEN AN INTERRUPT DOES NOT OCCUR AFTER CARDS HAVE BEEN RESTORED TO THE INPUT HOPPER AND THE 'RESET' BUTTON ON THE CARD READER HAS BEEN PRESSED.

- 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/RS1200 ONLY) | READ |
| 14 | CARD READER ERROR | READ |
| | (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 ODD 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.

ERROR FUNCTION TESTS FOR M1000/M200/M1200/RS1200

SEQ 0018

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/RS1200 ONLY)
READ CHECK ERROR

INSTRUCTIONS FOR MAKING A MIS-REGISTERED CARD

CUT A SMALL RECTANGULAR HOLE ABOUT THE SIZE OF A NORMAL
PUNCHED HOLE IN A BLANK CARD. MAKE SURE THE LEADING
EDGE OF THE HOLE FALLS IN A POSITION WHICH IS NORMALLY
CONSIDERED TO BE OUT OF REGISTRATION. (BETWEEN MARKED COLUMNS)

4 READING A SINGLE DATA PATTERN

SEQ 0019

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 HISTORY
-----8.0 FLOW CHARTS
-----9.0 PROGRAM LISTING

THIS IS A HISTORY OF THE DEVELOPMENT OF MAINDEC-11-DZCDB

PRODUCT CODE: MAINDEC-11-DZCDB-A
 VERSION 000.001

PRODUCT NAME: CD11/CD20 CARD READER DIAGNOSTIC

ORIGINAL RELEASE: MARCH 21 1976

ORIGINAL AUTHOR: BRUCE BURGESS

PRODUCT CODE: MAINDEC-11-DZCDB-B
 VERSION 000.002

PRODUCT NAME: CD11/CD20 CARD READER DIAGNOSTIC

DATE RELEASED: MAY 1977

UPDATE AUTHOR: GREG GLEZMAN

UPDATES:

1. ADDED TEST FOR MONITOR PRESENCE.
2. ADDED A HALT TO ALLOW THE OPERATOR TO SET THE SR
 SWITCHES IN THE CASE OF NO MONITOR.
3. FIXED THE (LOW BYTE) LOAD TESTS FOR THE
 CDC AND CDA REGISTERS SO THEY FUNCTION WITHOUT ERROR.
4. ADDED A TSET FOR THE RS1200 CARD READER
 TO TEST FOR A MIS-REGISTERED CARD.
5. ADDED A QUESTION FOR THE OPERATOR TO INFORM
 THE PROGRAM IF IT IS TESTING AN RS1200 CARD READER.
6. CHANGE THE MESSAGE TO THE OPERATOR PERTAINING TO THE
 STACK CHECK ERROR TEST. THE CHANGE MAKES THE PROCEDURE

FLOW CHART

CD11/CD20 CARD READER DIAGNOSTIC

COPYRIGHT 1977
DIGITAL EQUIPMENT CORPORATION

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-EXISTIA T MEMORY DETECTION |
| PAGE 14 | BYTE & DAT* 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 |


```
*****  
*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) *
```

CD11/CD20 CARD READER DIAGNOSTIC
TEST FOR INIT. OF ALL REGISTERS

*TST1(01) *

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

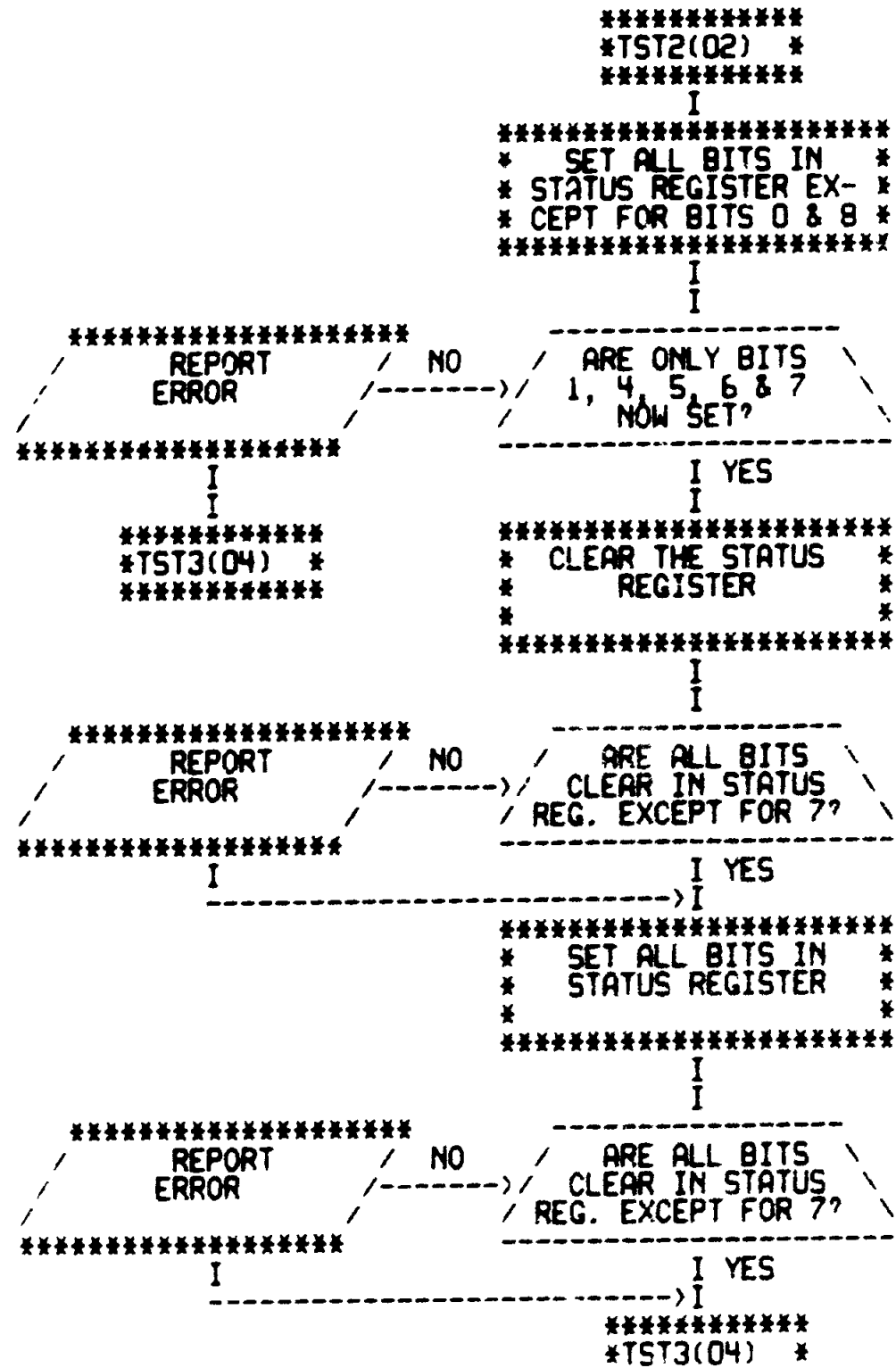
* SEND OUT AN *
* INITIALIZATION *
* PULSE (RESET) *

REPORT ERROR NO IS BIT7 SET IN
STATUS REGISTER?

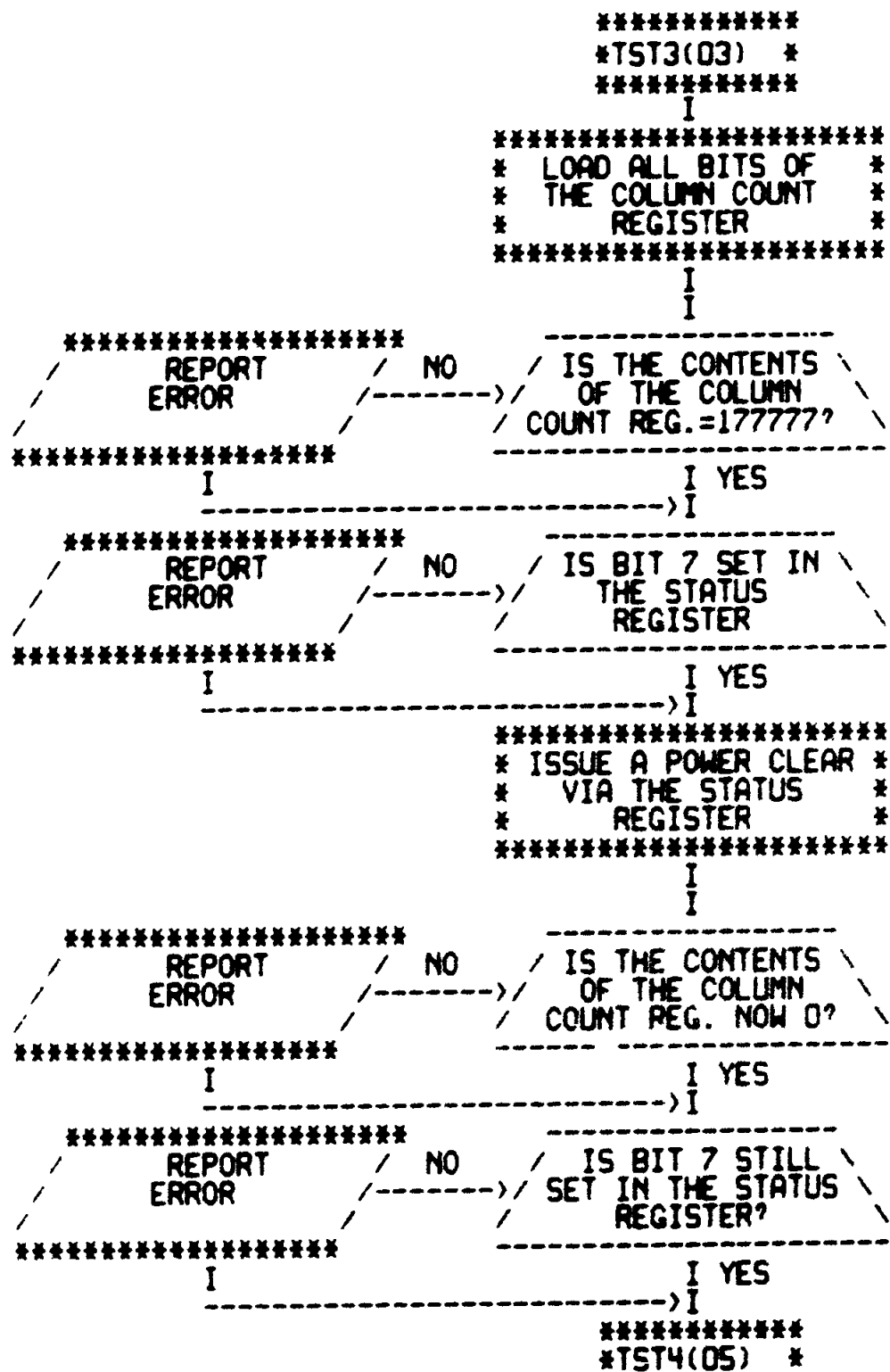
REPORT ERROR NO IS COLUMN COUNT
REGISTER
CLEARED?

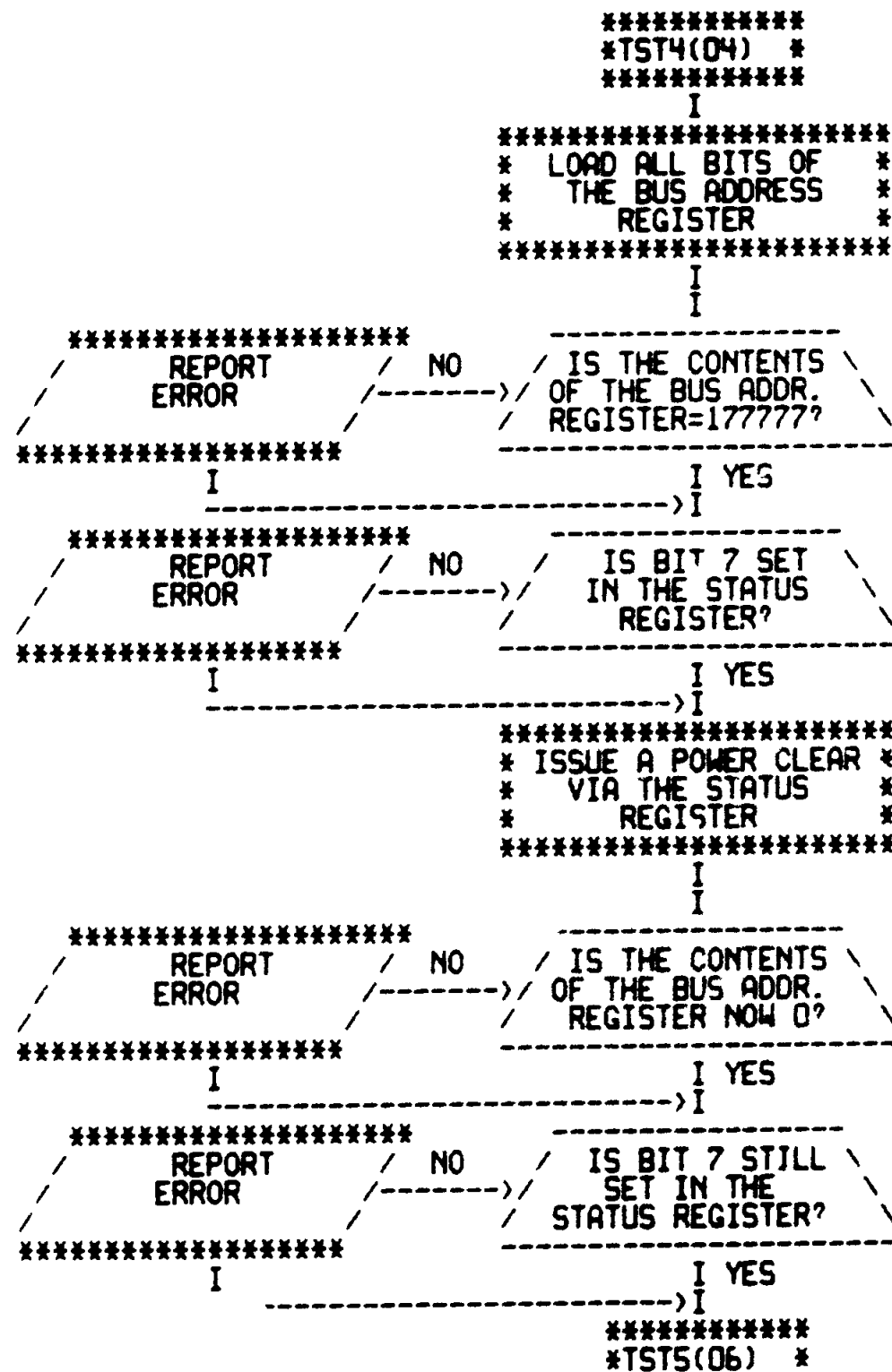
REPORT ERROR NO IS BUS ADDRESS
REGISTER CLEARED?

*TST2(03) *



CD11/CD20 CARD READER DIAGNOSTIC
TEST READ/WRITE OF COLUMN COUNT REGISTER





CD11/CD20 CARD READER DIAGNOSTIC
TEST INTERRUPT FROM CONTROLLER READY

```

*****
*TINT7(08) *
*****
I
-----
/ WAS CONTROLLER READY? \ NO \
\-----> REPORT ERROR /
-----
I YES
I <-----
*****
* RESET STACK *
* POINTER FROM *
* INTERRUPT *
*****
I
-----
/ DID ANY KIND OF ERROR SHOW UP? \ YES \
\-----> REPORT ERROR /
-----
I NO
I <-----
*****
* DISABLE *
* INTERRUPTS *
*****
I
-----
I <----- *CONT7 *
*****
*****
* 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 "BUFBE" *
*****
I
*****
* SET INTERRUPT *
* ENABLE AND *
* READ *
*****
I
*****
* WAIT FOR *
* CONTROLLER *
* READY *
*****
I
-----
/ DID AN INTERRUPT OCCUR? \ NO \
\-----> *INTN(08) *
-----
I YES
*****
*TINT7(08) *

```

```

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

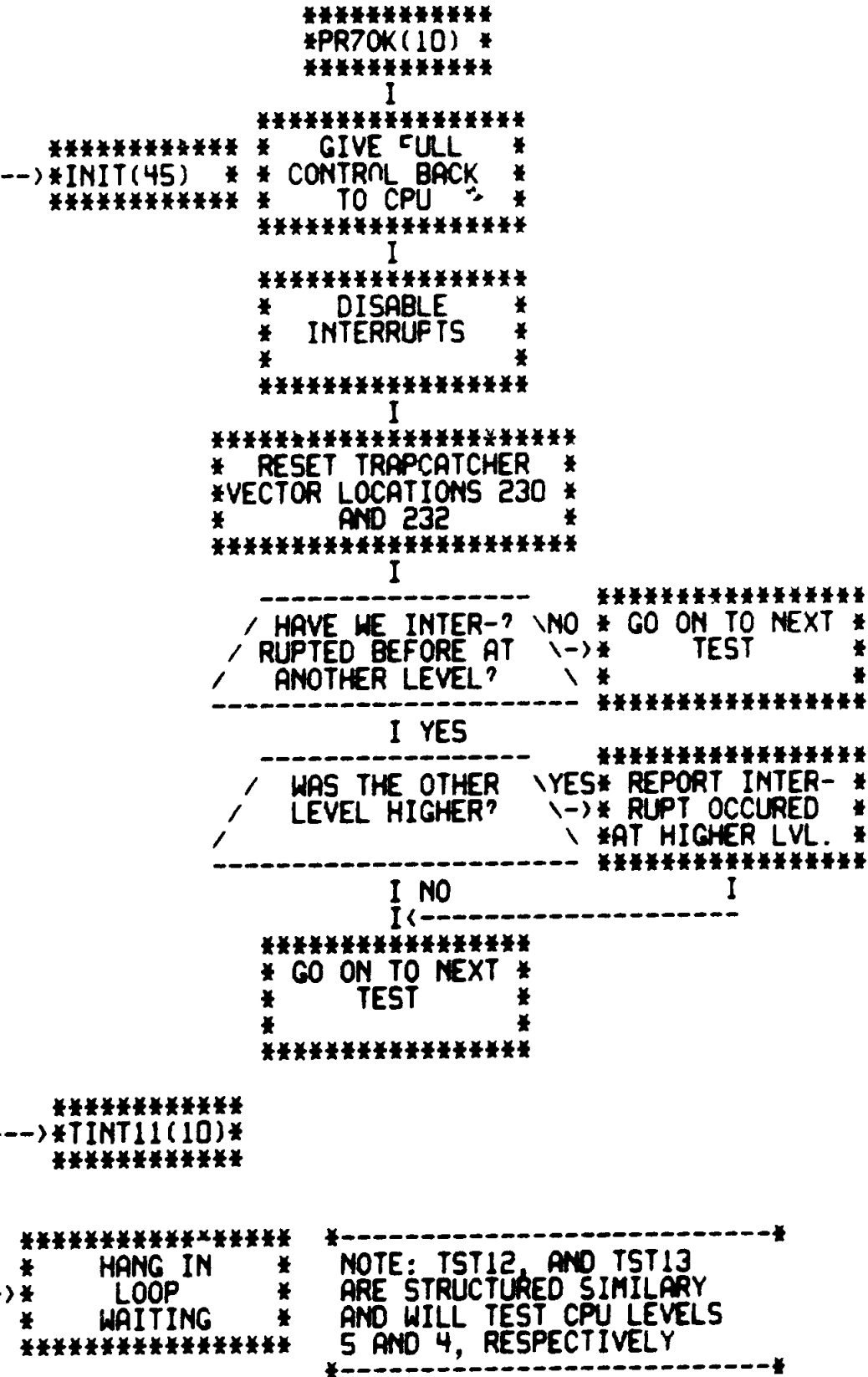
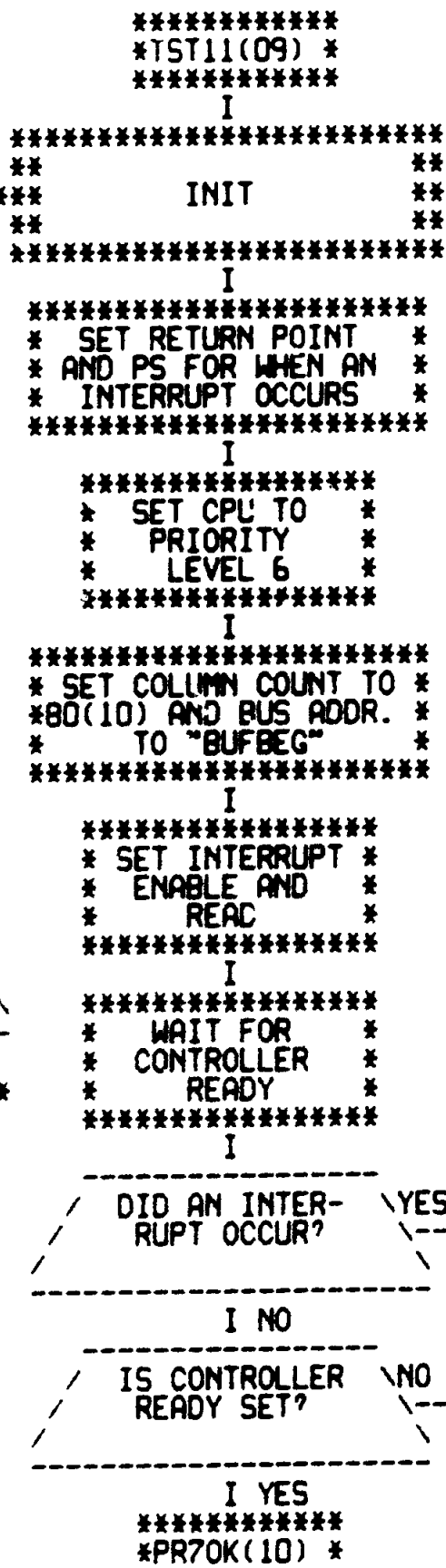
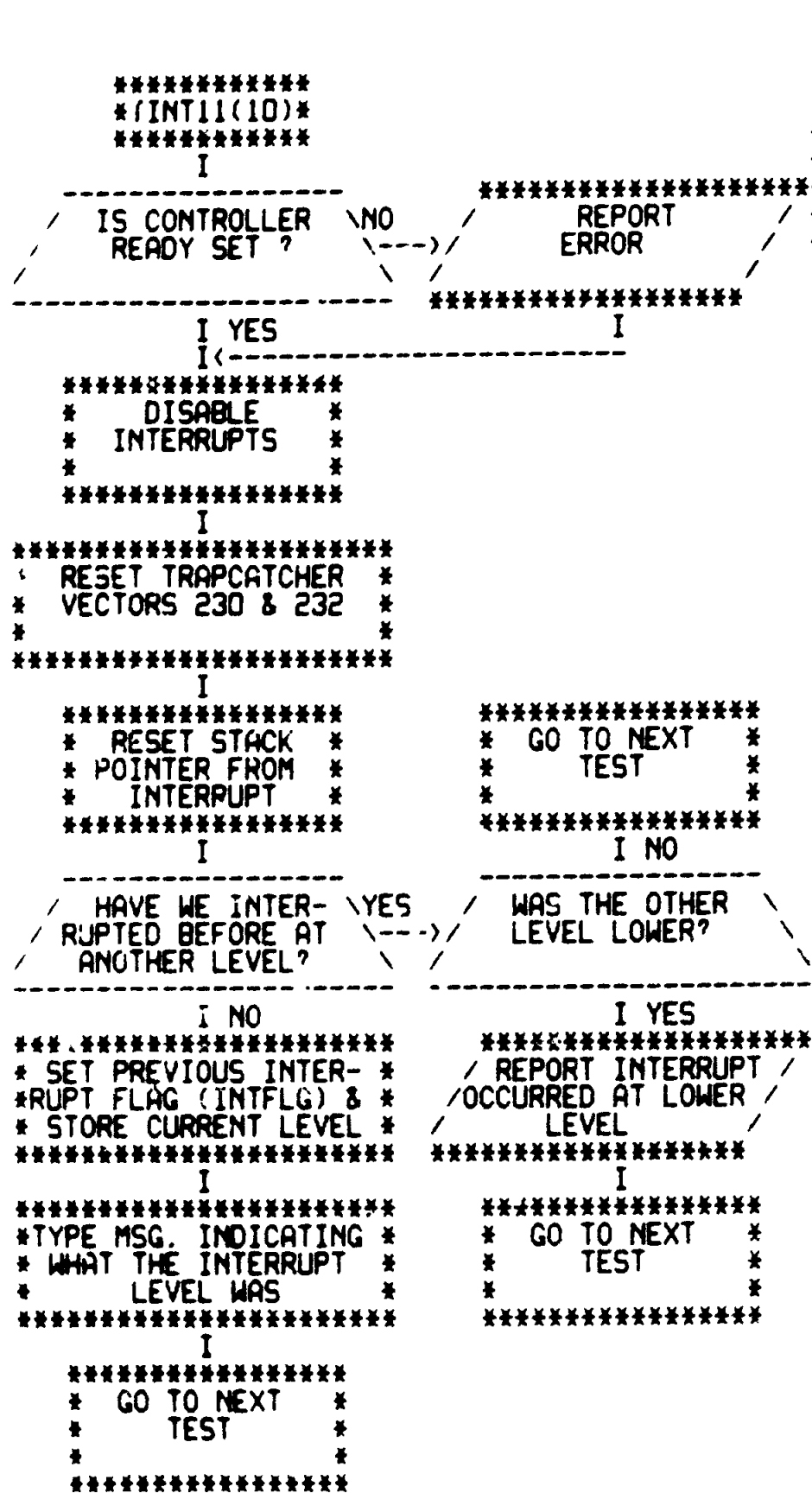
```

```

*****
*TST10(08)*
*****
I
*****
**          **
**   INIT   **
**          **
*****
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 "BUFBEG" *
*****
I
*****
* SET INTERRUPT *
* ENABLE AND *
* READ *
*****
I
*****
* WAIT FOR *
* CONTROLLER *
* READY *
*****
I
/ DID AN \ NO
  INTERRUPT \----->*T10GO *
  OCCUR ? \
*****
I YES
*****
*TINT10(09)*

```

CD1:/CD20 CARD READER DIAGNOSTIC TESTS FOR INTERRUPTS



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
FINALLY OCCUR? \--->

* REPORT ERROR *
* NO INTERRUPT *
* SHOULD OCCUR *

I NO

* GIVE CONTROL *
* BACK TO CPU *
* *

I

* RESET STACK *
* POINTER FROM *
* INTERRUPT *

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 *
* OCCURS *

TST20A(11)

CD11/CD20 CARD READER DIAGNOSTIC
SIMULTANEOUS INTERRUPTS AT MORE THAN ONE LEVEL

TINT21(12)

I

TST21(11)

I

| | | | |
|-----------------------------------------------|------------------------------------------------------------------------|----------------------------|--------------------------------|
| DID WE RECEIVE A 2ND INTERRUPT ? YES NO | ***** * RESET STACK * * POINTER FROM * * INTERRUPT * ***** | ***** * INIT * ***** | ***** * INIT(45) * ***** |
|-----------------------------------------------|------------------------------------------------------------------------|----------------------------|--------------------------------|

I NO

* GIVE CONTROL *
* BACK TO CPU *

I

* REPORT *
* ERROR *

I

* SET RETURN POINT *
* AND PS FOR WHEN AN *
* INTERRUPT OCCURS *

I

* DISABLE *
* INTERRUPTS *

I

* SET CPU TO *
* PRIORITY *
* LEVEL 0 *

I

* RESTORE TRAPCATCHER *
* VECTOR LOCATIONS 230 *
* AND 232 *

I

* SET COLUMN COUNT TO *
* 1(10) AND BUS ADDR. *
* TO "BUFBEQ" *

I

TST22(13)

I

* SET INTERRUPT *
* ENABLE AND *
* READ *

| | |
|---------------------------------------|---------------------------------------------------|
| DID AN INTERRUPT OCCUR ? YES NO | ***** * HANG IN LOOP * * WAITING * ***** |
|---------------------------------------|---------------------------------------------------|

I YES

* RESET STACK *
* POINTER FROM *
* INTERRUPT *

| | | |
|---------------------------------------------------------------------|-------------------------------------------------------------|--------------------------------|
| ***** * SET RETURN * * ADDR. FOR NEXT * * POSSIBLE INTER.* | ***** * SET CPU BACK * * TO PRIORITY * * LEVEL 0 * | ***** *TINT21(12)* ***** |
|---------------------------------------------------------------------|-------------------------------------------------------------|--------------------------------|

CD11/CD20 CARD READER DIAGNOSTIC
NON-EXISTANT MEMORY DETECTION

```

*****
*TINT22(13)*
*****
I
*****
* RESTORE TRAPCATCHER *
* VECTOR LOCATIONS 230 *
*      & 232          *
*****
I
-----
/ IS CONTROLLER \ NO \
 \ READY?       /
-----
I YES
I <-----
/ IS ERROR BIT15 \ NO \
 \ SET?          /
-----
I YES
I <-----
/ IS NXM BIT09  \ NO \
 \ SET?         /
-----
I YES
I <-----
/ IS EXTENDED  \ NO \
 \ MEMORY BIT 17 SET?
-----
I YES
I <-----
/ IS EXTENDED  \ NO \
 \ MEMORY BIT16 SET?
-----
I YES
I <-----
*****
*T22A(13) *
*****

```

```

*****
*TST22(12) *
*****
I
*****
**
**      INIT      **
**
*****
I
*****
*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. 16000 *
*****
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 *
 \ BITS SET ? E.G. \ -> *
 \ BITS 2,10,11,ETC. \ *
-----
I NO
I <-----
/ DOES BUS ADDR. \ NO *
 \ REGISTER CONTENTS \ -> *
 \ =160002 ?      \ *
-----
I YES
I <-----
/ DOES COLUMN CNT \ NO *
 \ REGISTER SHOW 4 \ -> *
 \ COLUMNS LEFT ? \ *
-----
I YES
I <-----
*****
*TST23(14) *
*****
*****
* HANG IN LOOP *
*      WAITING *
*****

```

CO11/CO20 CARD READER DIAGNOSTIC
BYTE & DATIP LOAD OF COLUMN COUNT REGISTER

```

*****
*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<-----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<-----I
*****
*TST26 *
*****

```

```

*****
*TST26 *
*****

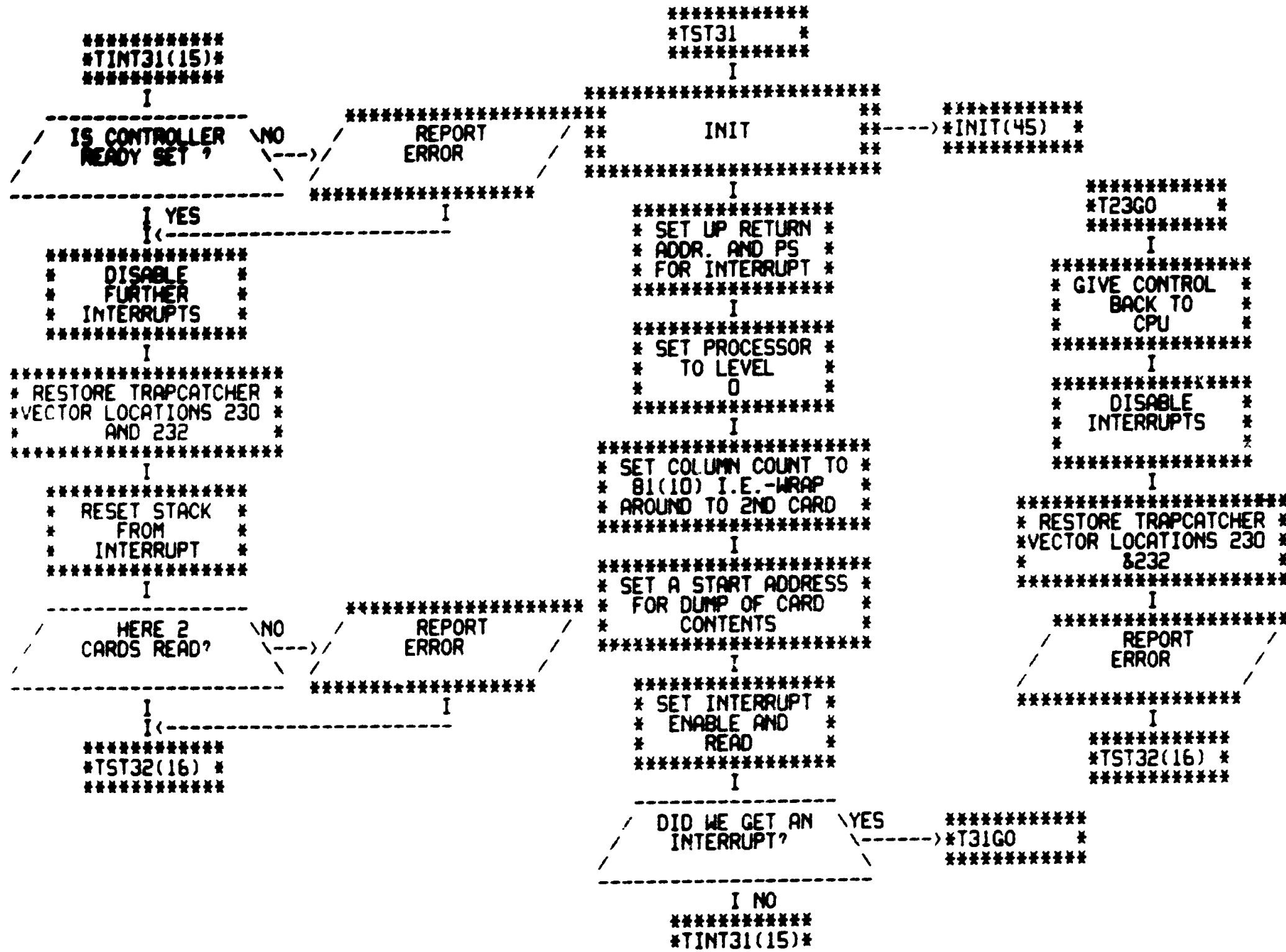
```

```

*****
*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.



CD11/CD20 CARD READER DIAGNOSTIC
BUS ADDRESS ODD & TRANSFER IN NON-PACK MODE

```

*****
*ENDCK(16) *
*****
I
-----
/ IS SNK07 > SET? \ YES
-----> *DATST(17) *
*****

I
*****
* DING *
* A *
* LING? *
*****
I
*****
*RESTR *
*****

```

```

*****
*TST32(15) *
*****
I
*****
**          **
**   INIT   **-----> *INIT(45) *
**          **
*****

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
  ILLEGAL      \---->
INSTRUCTION TRAP? \
-----
*****
I
*****
* RESET STACK *
* FROM ILLEGAL *
* INSTR. TRAP *
*****
I
-----
I<-----
*****
* RESTORE TRAP- *
* CATCHER FOR  *
* LOCS. 10 & 12 *

```

*ENDCK(16) *

```
*****  
*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 I  
***** I  
* LOAD ALPHANUMERIC * I  
* DATA TABLE * I  
* POINTERS * I  
***** I  
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

I YES

* SET UP FOR * NO
* ACCEPTANCE OF * <----- / WAS IMAGE MODE
* PACK MODE * / SELECTED ALSO? \-----I

I YES

-----> I <-----

* READ *
* A *
* CARD *

*BKGND * <-----> I <-----

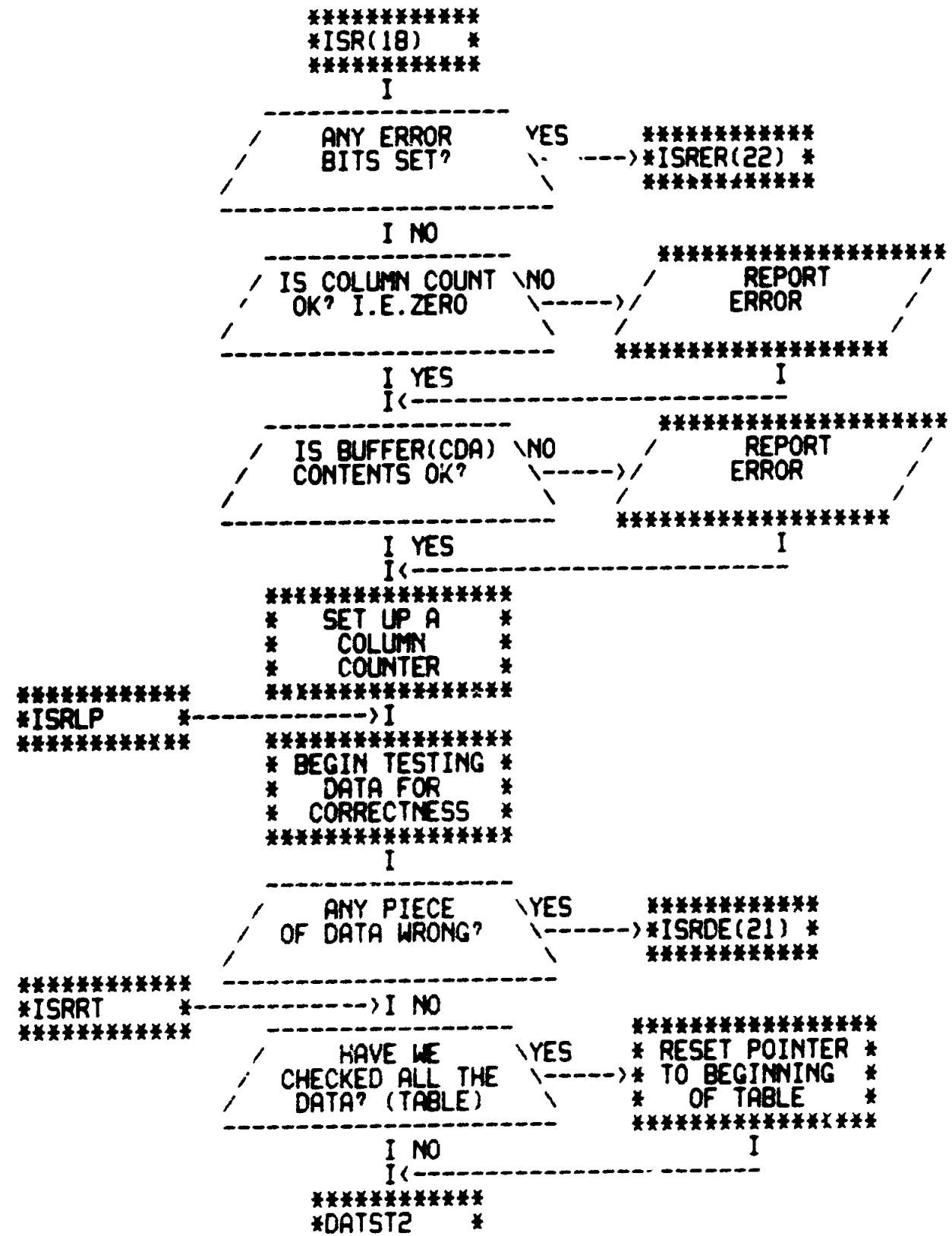
/ HAVE WE \ \NO
FINISHED READING /
A CARD YET ? /-----I

I YES

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

I YES

/ 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 ? \ -----> * STEP UP TO *
* NEXT CARD *
*****
I NO
*****
* UPDATE *
* COLUMN *
* COUNT *
*****
I
*****
* UPDATE TABLE *
* OFFSET FOR *
* CARD #1 IN DECK *
*****
I
-----
/ HAVE WE LOOKED \ NO
/ AT LAST COLUMN OF \
/ DECK ? \ -----> *ISRLP *
*****
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
/ SELECTED ? \ -----> * SET UP FOR *
* PACKING MODE *
*****

```

```

I YES
I <-----
*****
#SRETRN(20)*
*****

```

```

*****
*ISRLP *
*****

```

*ISRDE(19) *

I

```

-----
/ IS THIS THE \ YES
/ FIRST CARD ? \-----> * CALCULATE * * RESET CARD *
* PRESENT POSIT- *--> * COUNTER *
* ION OF IMPORT. * * (CDCNT) *
-----
I NO
I<-----

```

```

-----
/ INHIBIT ERROR \ YES
/ PR'NTOUT ? \-----> I
-----
I NO

```

```

*****
* TYPE OUT HEADING, *
* DECK, CARD COUNT, & *
* COLUMN NUMBER *
*****
I
I
*****
* TYPE OUT THE "WAS" *
* AND "SHOULD BE" *
* DATA *
*****
I<-----

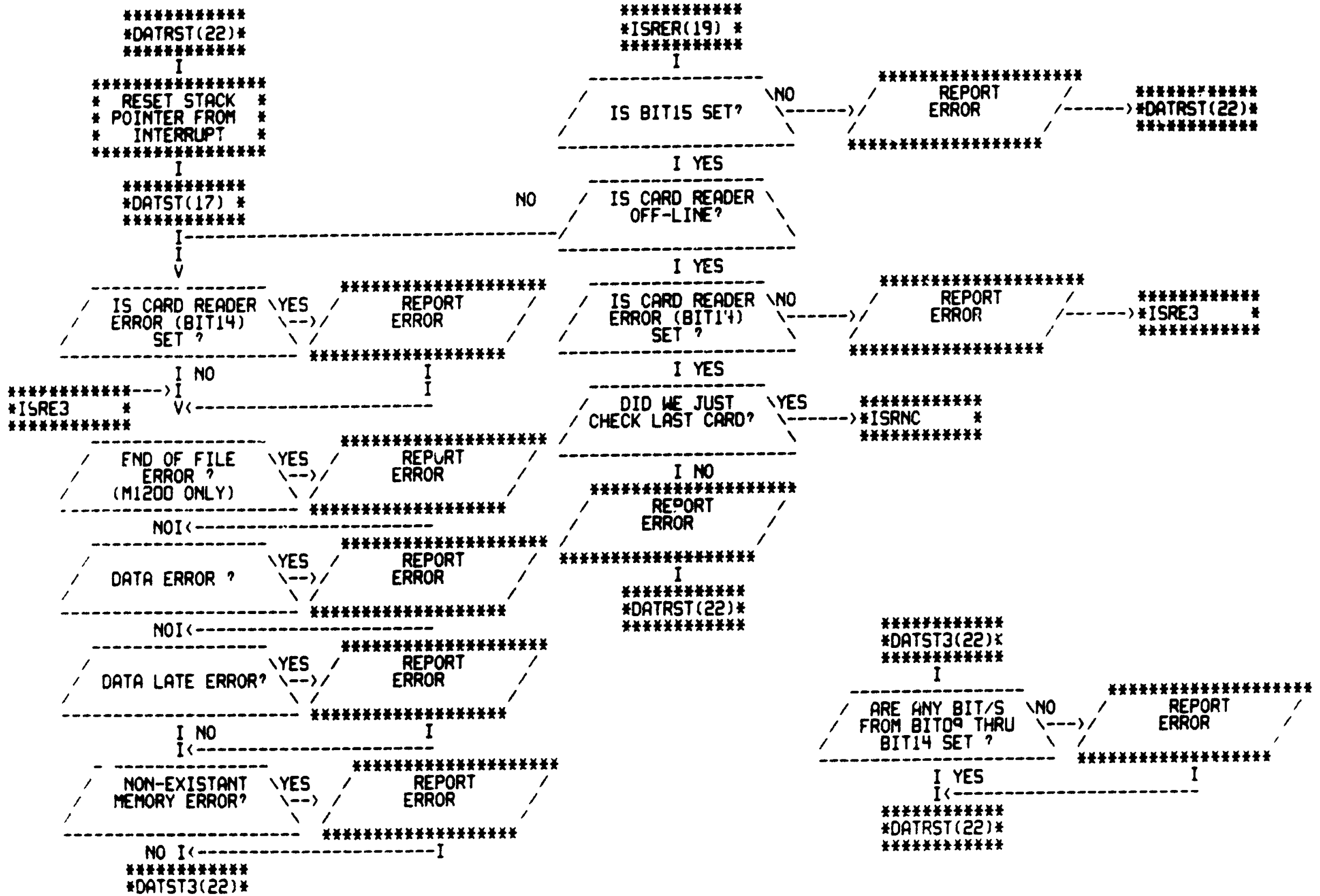
```

```

-----
/ HALT \ YES
/ ON \-----> * *
/ ERROR ? * HALT *
-----
I NO

```

*ISRRT *



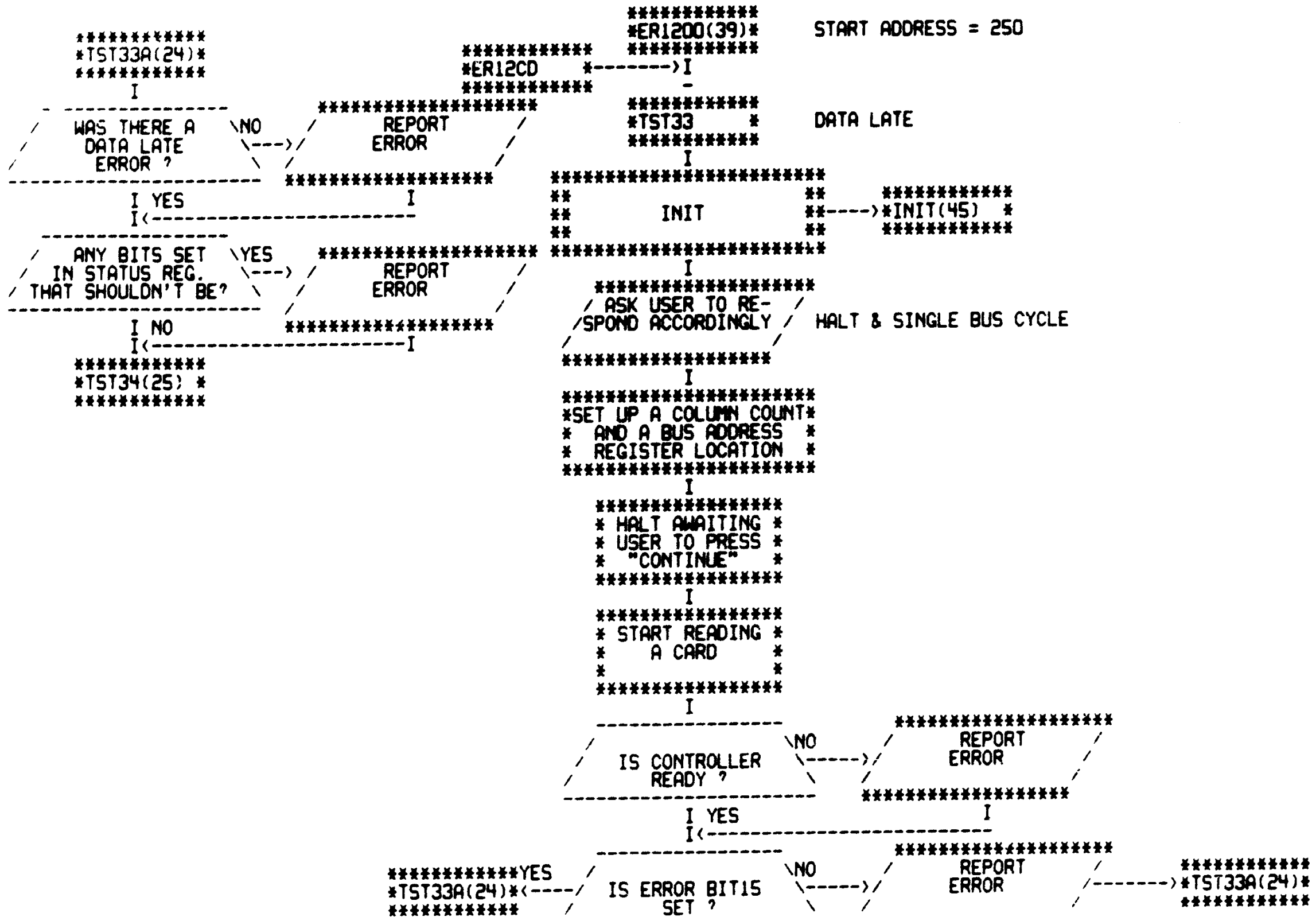
DATA RELIABILITY TESTING FOR PACKED MODE WILL

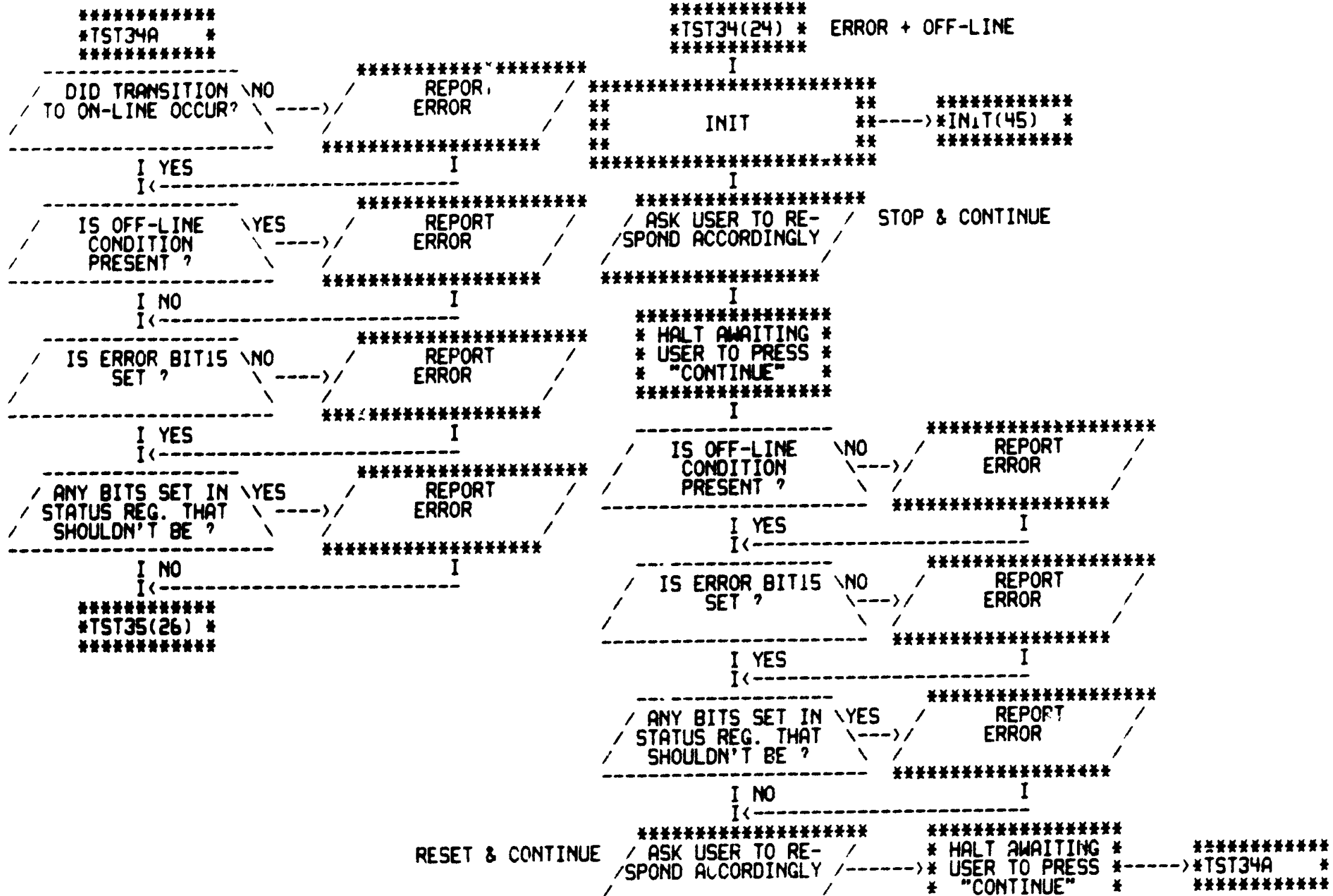
START AT *DATST(17) * & BRANCH OFF TO*PSR *

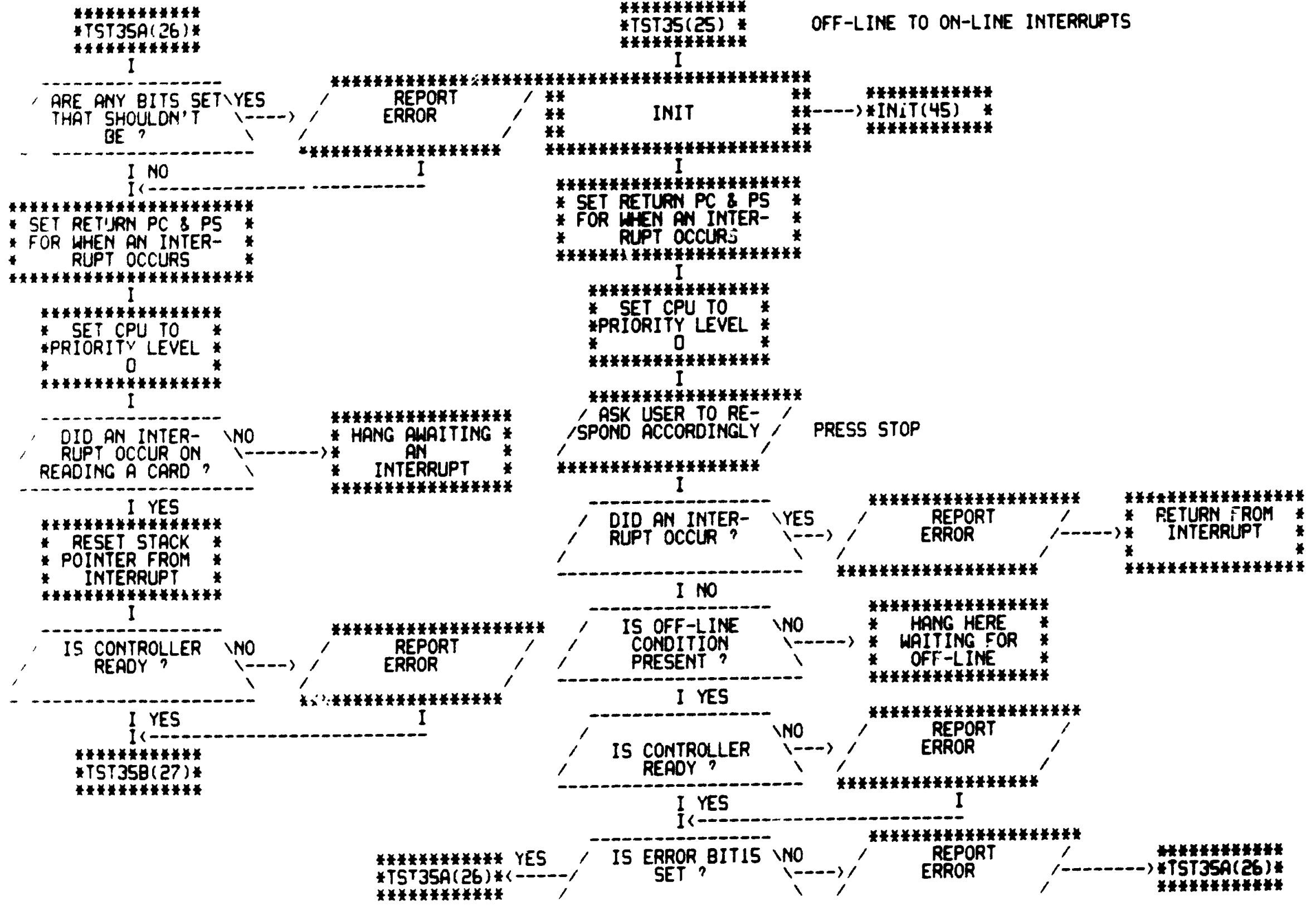
WHERE DATA WILL BE HANDLED AS OUTLINED IN

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

EXCEPTION: DATA IS HANDLED USING BYTE CONSTRUCTION







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

```
*****  
*TST35B(26)*  
*****  
I  
-----  
/ OFF-LINE \ NO  
CONDITION \-----> / REPORT  
PRESENT ? ERROR  
-----  
I YES  
I<-----  
-----  
/ IS ERROR BIT15 \ NO  
SET ? \-----> / REPORT  
----- ERROR  
I YES  
I<-----  
-----  
/ ARE ANY BITS SET \ YES  
THAT SHOULDN'T \-----> / REPORT  
BE ? ERROR  
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 ? \-----> /  
-----
```

```
***** YES / DID AN INTER- \ NO * HANG AWAITING *  
*TST35C(27)*<-----> / RUPT OCCUR ? \-----> * INTERRUPT *  
*****
```

```
*****  
*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  
STATUS = 300 ? \ * REPORT  
* ERROR *  
*****  
I YES  
I<-----  
*****  
*TST37(29)*  
*****
```

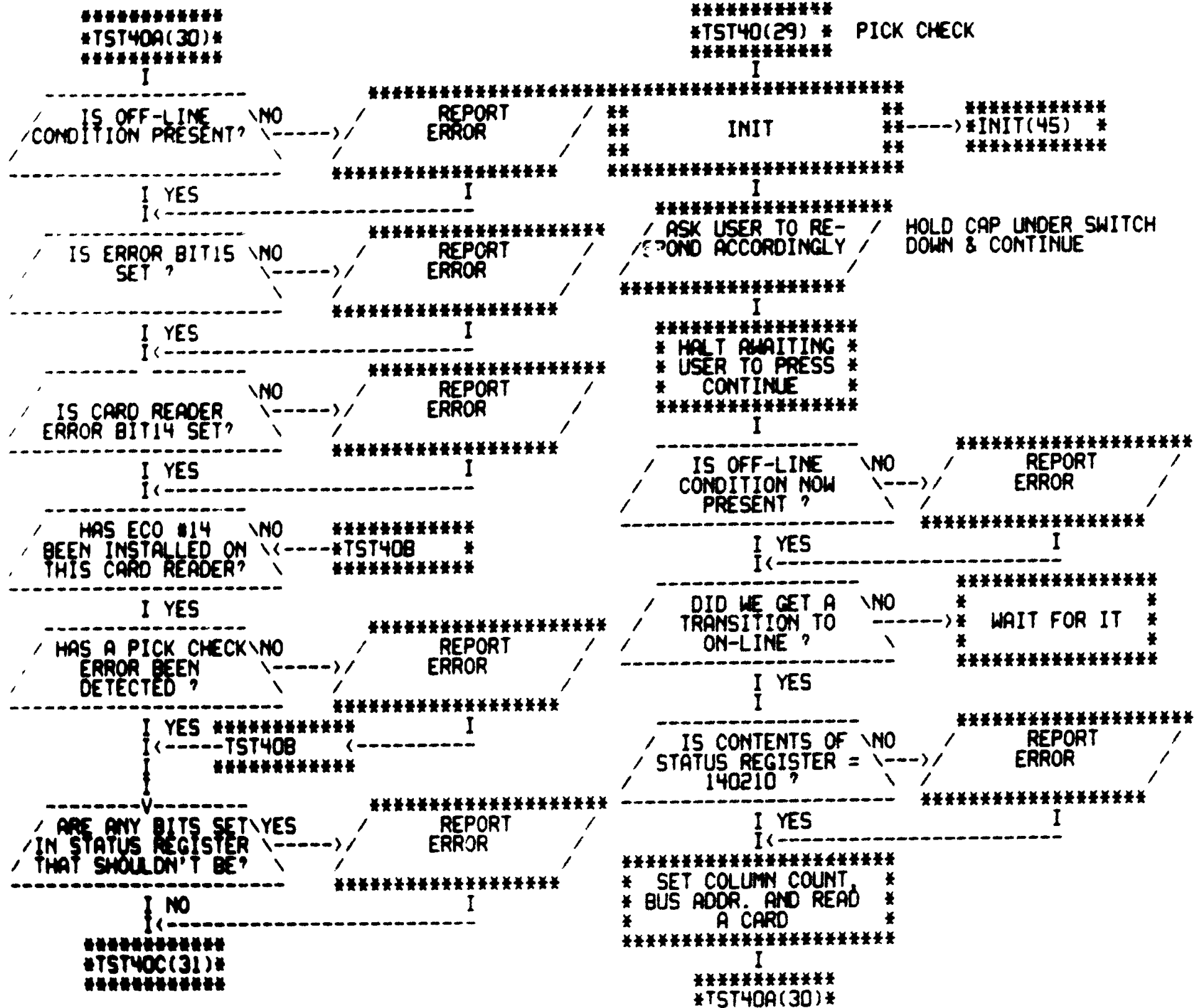
```
*****  
*TST37A(29)*  
*****  
I  
*****  
* SET RETURN PC & PS *  
* FOR AN INTERRUPT *  
* OCCURRENCE *  
*****  
I  
*****  
*SET CPU LEVEL= *  
*0 AND SET "IE" *  
* BIT *  
*****  
I  
*****  
/ ASK USER TO RE- /  
/ SPOND ACCORDINGLY /  
*****  
I  
*****  
/ DID INTERRUPT \ NO * HANG AWAITING *  
/ 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)*  
*****
```

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

OUTPUT STACKER FULL

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

CD11/CD20 CARD READER DIAGNOSTIC
ERROR FUNCTION TESTING OF MODEL M1200



CD11/CD20 CARD READER DIAGNOSTIC
ERROR FUNCTION TESTING OF MODEL M1200

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 = \--->
300 ? \----->

* REPORT *
* ERROR *

I YES

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- /
SPOND ACCORDINGLY /

RESTORE CARDS
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)

0011/0120 CARD READER DIAGNOSTIC
ERROR FUNCTION TESTING OF MODEL M1200

TST41A(32)

I

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

REPORT
ERROR

I YES
I<-----

I

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

REPORT
ERROR

I YES
I<-----

I

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

REPORT
ERROR

I YES
I<-----

I

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

*TST41B *

I YES
I<-----

I

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

REPORT
ERROR

I YES *****
I<-----*TST41B *<-----

I

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

REPORT
ERROR

I NO
I<-----

TST41C(33)

*TST41(31) *

STACK CHECK

I

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

I

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

I

/ HAS OFF-LINE \ NO
/ OCCURRED YET? \----->
?

* AWAIT *
* OFF-LINE *
* CONDITION *

I YES

/ HAS TRANSITION \ NO
/ TO ON-LINE OCCURR- /
/ ED YET ? \----->
?

* AWAIT *
* TRANSITION TO *
* ON-LINE *

I YES

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

REPORT
ERROR

I YES
I<-----

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

I

/ HAS CONTROLLER \ NO
/ READY SET YET ? \----->
?

* WAIT *
* FOR *
* IT *

I YES

TST41A(32)

TST41D(33)

I

* READ A *
* CARD *
* *

I

/ DID AN \ NO
/ INTERRUPT OCCUR? \----->

* HANG AWAITING *
* INTERRUPT *
* *

I YES

* RESET STACK *
* POINTER FROM *
* INTERRUPT *

I

/ IS CONTENTS OF \ NO
/ STATUS REGISTER = \----->
300 ?

* 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 \ NO
/ INTERRUPT ? \----->

* HANG AWAITING *
* INTERRUPT *
* *

I YES

* RESET STACK *
* POINTER FROM *
* INTERRUPT *

I

* SET RETURN PC & PS *
* FOR WHEN AN INTERRUPT *
* OCCURS *

I

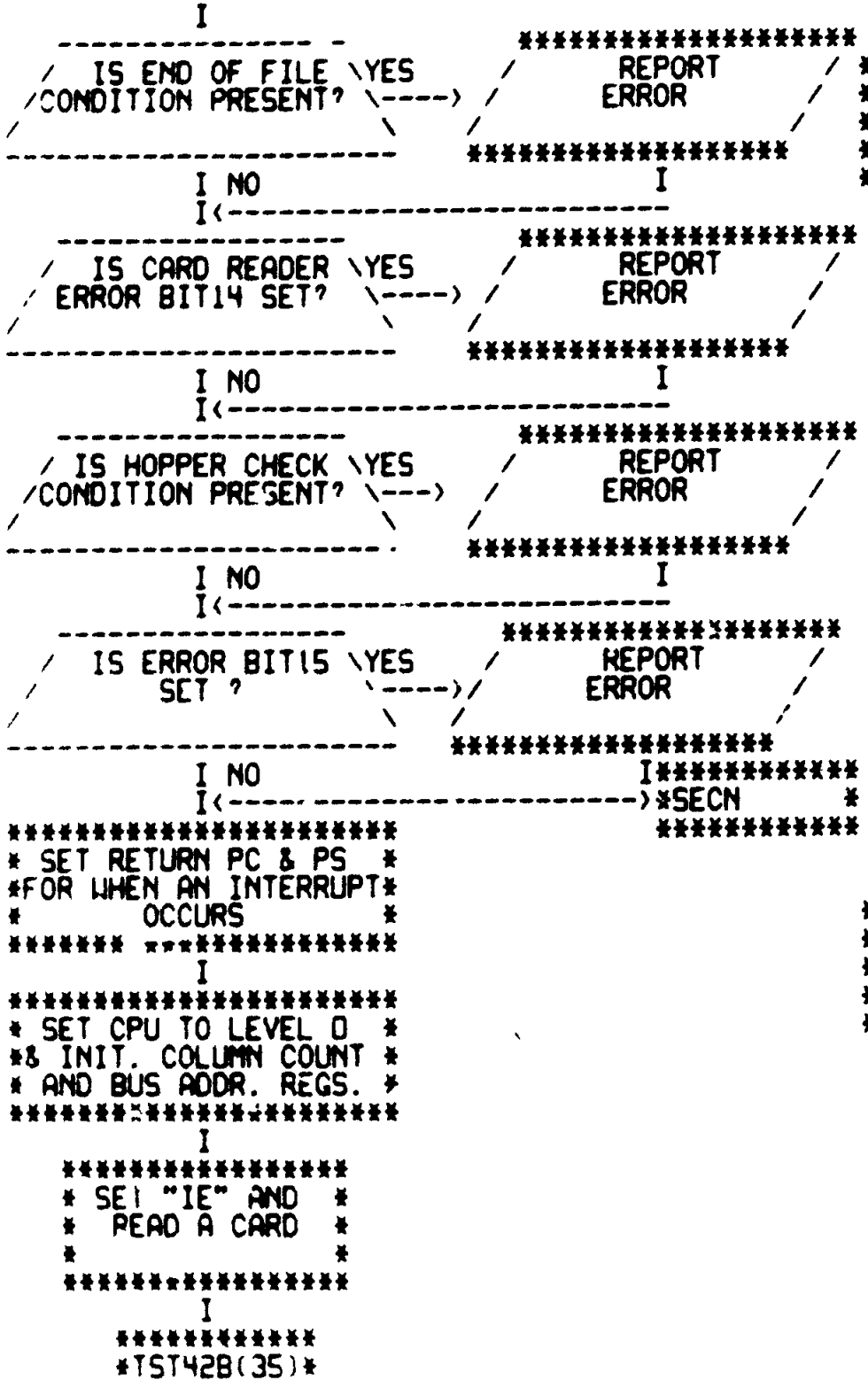
* SET CPU TO LEVEL 0 & *
* INIT. COLUMN COUNT *
* & BUS ADDR. REGS. *

I

TST41D(33)

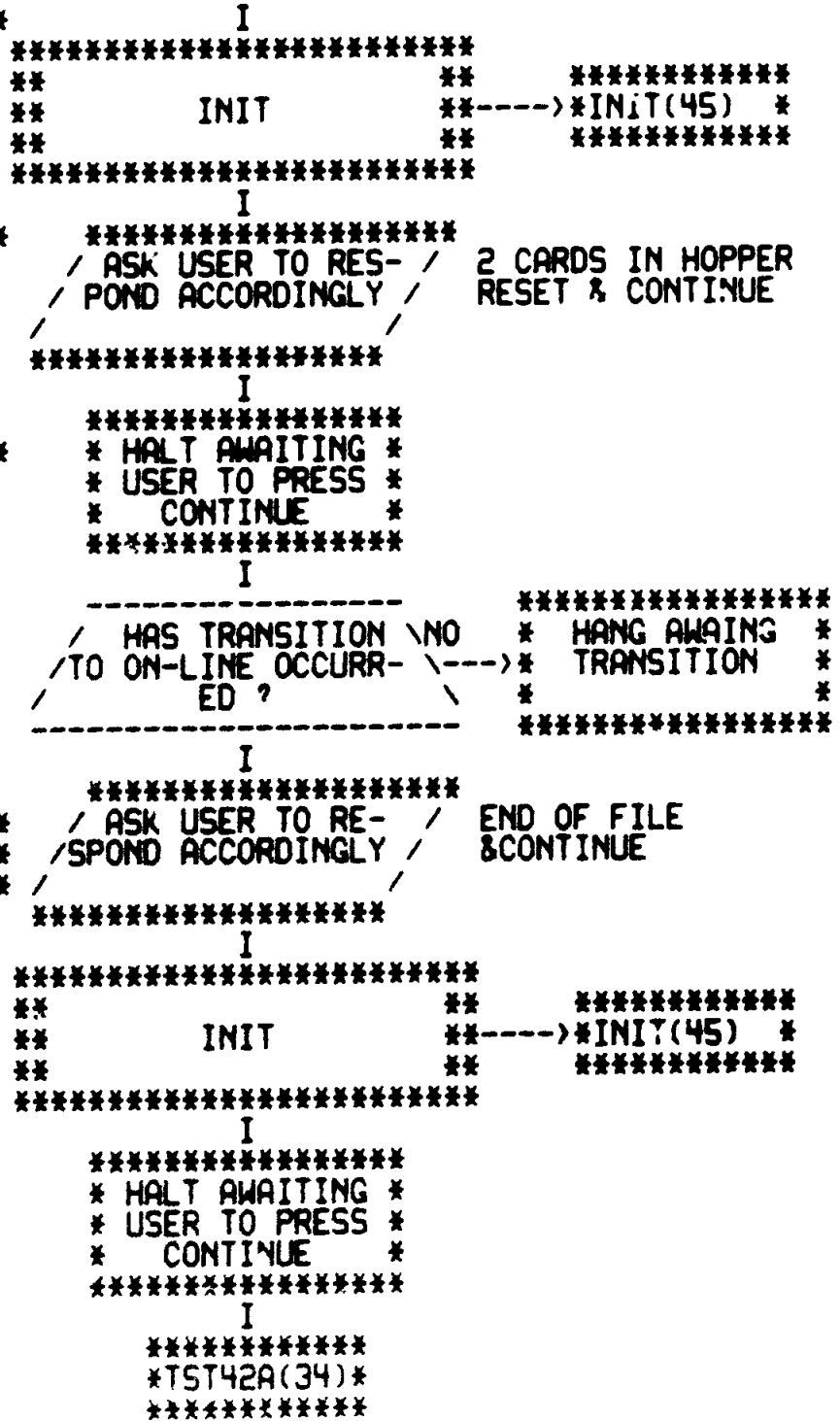
CD11/CD20 CARD READER DIAGNOSTIC
ERROR FUNCTION TESTING OF MODEL M1200

TST42A(34)



TST42(33)

END OF FILE & HOPPER CHECK



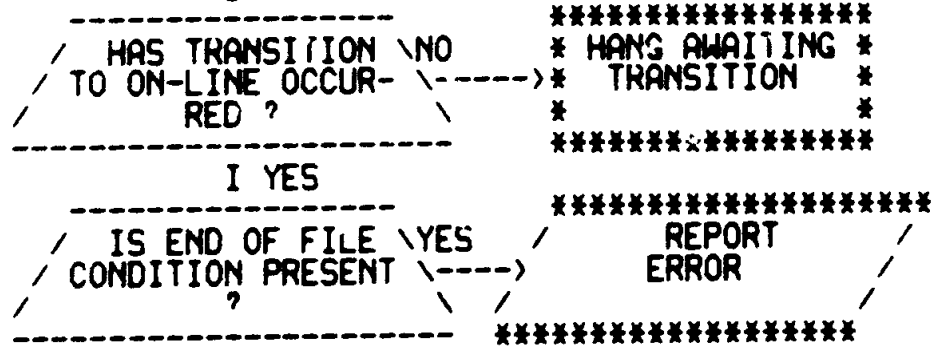
```
*****  
*TINTIA(35)*  
*****  
I  
*****  
* RESET STACK *  
* POINTER FROM *  
* INTERRUPT *  
*****  
I  
-----  
/ IS END OF FILE \ NO  
/ CONDITION PRESENT? \-----> / REPORT  
----- ERROR  
I YES I  
-----  
/ IS CARD READER \ NO  
/ ERROR BIT14 SET ? \-----> / REPORT  
----- ERROR  
I YES I  
-----  
/ IS HOPPER EMPTY \-----> / REPORT  
/ CONDITION PRESENT? \-----> ERROR  
I YES I  
-----  
/ IS ERROR BIT15 \ NO  
/ SET ? \-----> / REPORT  
----- ERROR  
I YES I  
-----  
/ ASK USER TO RESP- / RESTORE CARDS  
/ OND ACCORDINGLY / RESET & CONTINUE  
-----  
I  
*****  
* HALT AWAITING *  
* USER TO PRESS *  
* CONTINUE *  
*****
```

```
*****  
*TST42B(34)*  
*****  
I  
-----  
/ DID WE GET AN \ NO  
/ INTEPRUPT ? \-----> / HANG AWAITING *  
----- * INTERRUPT ? *  
----- *  
----- *****  
I  
*****  
* RESET STACK *  
* POINTER FROM *  
* INTERRUPT *  
*****  
I  
-----  
/ IS END OF FILE \ YES  
/ CONDITION PRESENT ? \-----> / REPORT  
----- ERROR  
I NO I  
-----  
/ IS CARD READER \ YES  
/ ERROR BIT14 SET? \-----> / REPORT  
----- ERROR  
I NO I  
-----  
/ IS ERROR BIT15 \ YES  
/ SET ? \-----> / REPORT  
----- ERROR  
I NO I  
-----  
/ SET UP RETURN *  
/ ADDR. TO TINTIA*  
/ ON NEXT INTER. *  
-----  
I  
*****  
*SECN *  
*****  
***** AN INTERRUPT (2ND) FROM  
*TINTIA(35)* "SECN" AREA WILL GO TO  
***** "TINTIA"
```

```
*****  
*TST42C(36)*  
*****
```

TST42C(35)

I

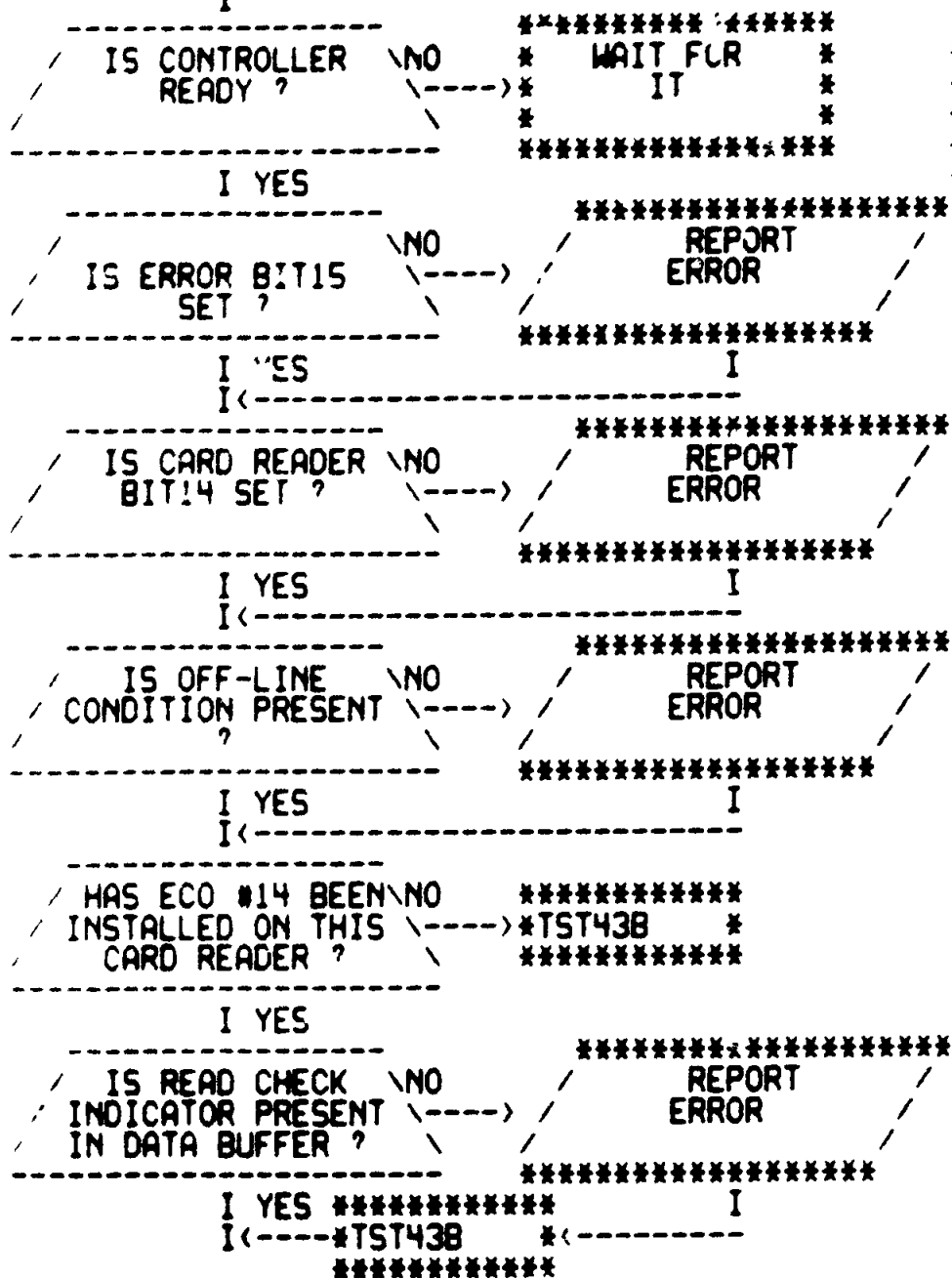


*TST43(37) *

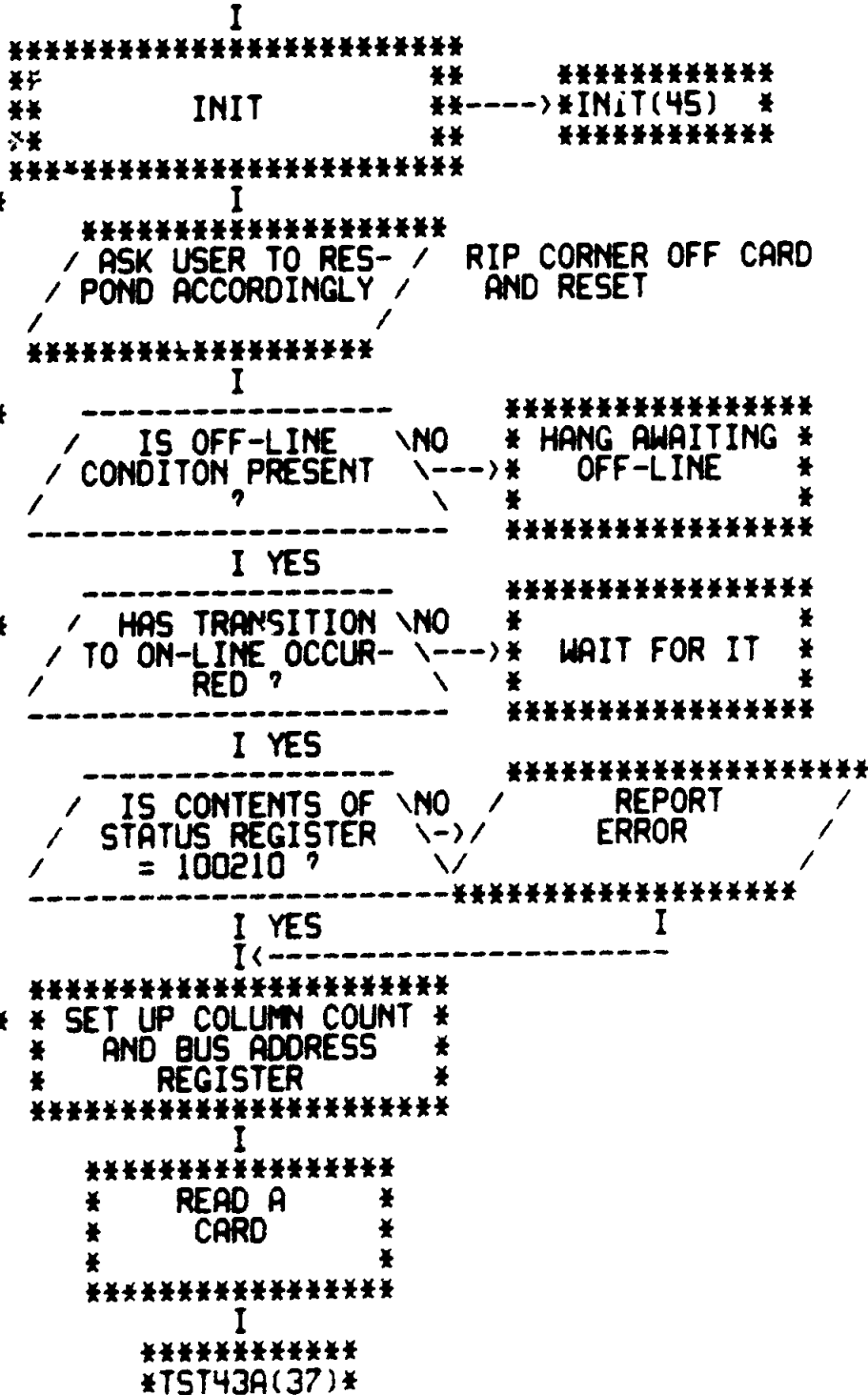
TST43A(37)

TST43(36)

READ CHECK



TST43C(38)



```
*****
*TST43C(37)*
*****
I
-----
/ ARE ANY EXTRA \ YES / REPORT
/ BITS SET IN "CDS" \-----> ERROR
/ THAT SHOULDN'T BE? \
-----
I NO I
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 * WAIT FOR *
/ RUPT OCCURRED? \-----> * IT *
/ * *
/ * *
/ * *
-----
I YES
*****
* HALT AWAITING *
* USER TO PRESS *
* CONTINUE *
*****
I
*****
* RESET STACK *
* POINTER FROM *
* INTERRUPT *
*****
I
*****
*ER12CD *
```

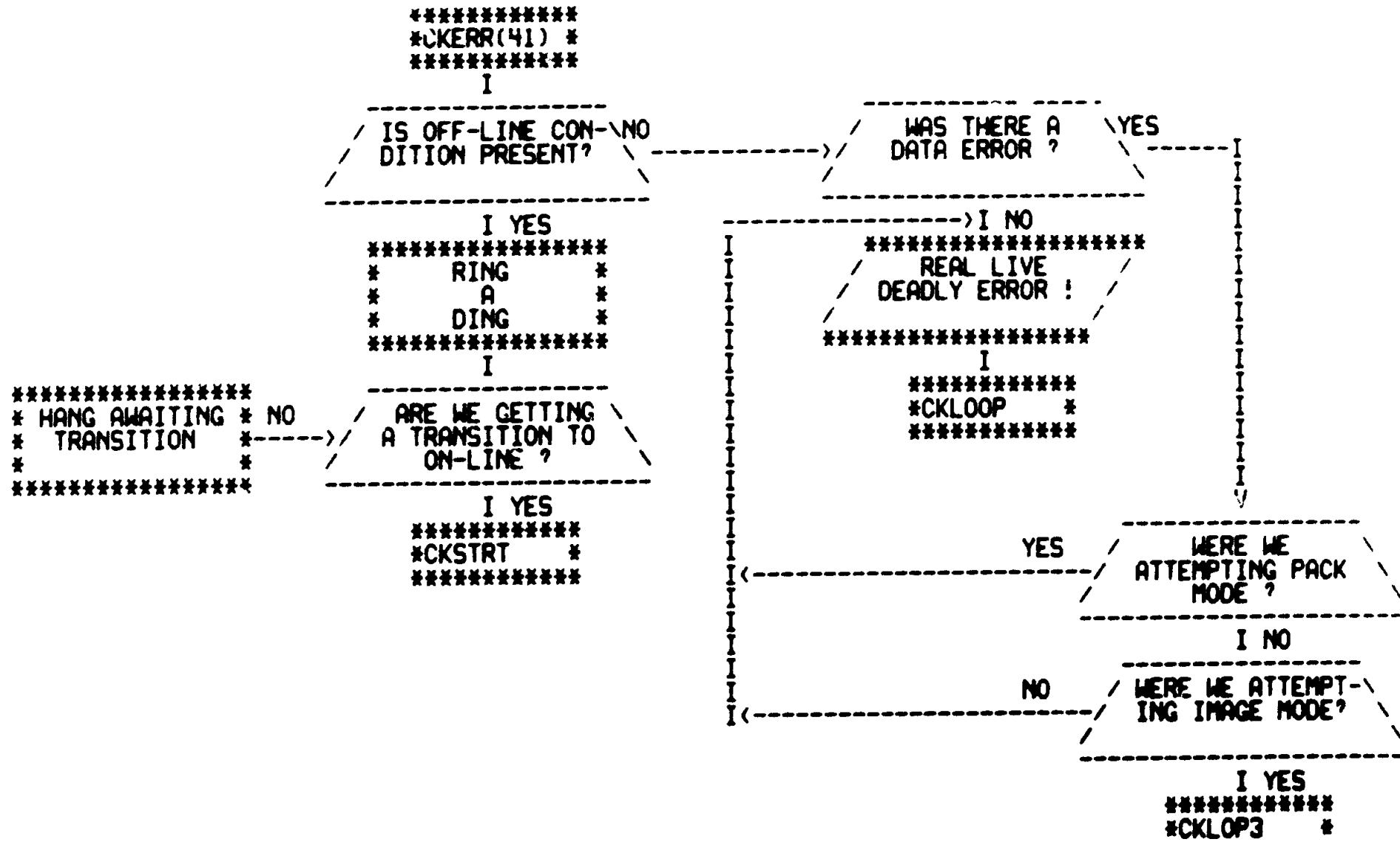
ERROR FUNCTION TESTING OF CARD READER MODELS M1000 OR
M200 IS IDENTICAL TO THAT OF AN M1200 AS OUTLINED

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 \----->  
-----  
I YES  
*****  
*SAME2(41) *  
*****
```

```
*****  
*CKSAME * START ADDRESS = 240  
*****  
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) *
```

CD11/CD20 CARD READER DIAGNOSTIC
PROGRAM TO LOOP ON SINGLE DATA PATTERN

```

*****
*FAIL1(43) *
*****
I
-----
/ HALT 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) *

```

CD11/CD20 CARD READER DIAGNOSTIC
PROGRAM TO LOOP ON TEST

```

*****
*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 COM- \ 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)

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

* HANG AWAITING *
* CONTROLLER *
* READY *

I YES

* ISSUE A *
* POWER CLEAR *
* *

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

REPORT
ERROR

I YES
I<-----

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

CD11/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 | | | |
| TST32 | 15 | 15 | 16 | |
| TST33 | 24 | | | |
| TST33A | 24 | 24 | 24 | |
| TST34 | 24 | 25 | | |
| TST34A | 25 | 25 | | |
| TST35 | 25 | 26 | | |
| TST35A | 26 | 26 | 26 | |
| TST35B | 26 | 27 | | |
| TST35C | 27 | 27 | | |
| TST36 | 27 | 28 | 28 | |
| TST36A | 28 | 28 | | |
| TST36B | 28 | 28 | | |
| TST37 | 28 | 29 | | |
| TST37A | 29 | 29 | | |
| TST37B | 29 | 29 | | |
| TST4 | 04 | 05 | | |
| TST40 | 29 | 30 | | |
| TST40A | 30 | 30 | | |
| TST40B | 30 | 30 | | |
| TST40C | 30 | 31 | | |
| TST40D | 31 | 31 | | |
| TST41 | 31 | 32 | | |
| TST41A | 32 | 32 | | |
| TST41B | 32 | 32 | | |
| TST41C | 32 | 33 | | |
| TST41D | 33 | 33 | | |
| TST42 | 33 | 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 | |
| TST6A | 07 | 07 | | 07 |

CD11/CD20 CARD READER DIAGNOSTIC
FLOW CHART CROSS REFERENCE LIST

F06

DECFLO VER 00.12 01-JUL-77 08:45 PAGE 48

SEQ 0070

07

| | |
|------|------------------------------------------------------------|
| 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 |
| 862 | LOGIC FUNCTION TESTS |
| 942 | T1 TEST FOR INIT. OF ALL REGISTERS |
| 981 | T2 TEST READ/WRITE OF STATUS REGISTER |
| 1007 | T3 TEST READ/WRITE OF COLUMN COUNT REGISTER |
| 1031 | T4 TEST READ/WRITE OF BUS ADDRESS REGISTER |
| 1055 | T5 TEST CONTROLLER READY TO CLEAR BIT0 |
| 1106 | T6 TEST BIT2 TO BE CLEAR AFTER CARD READ |
| 1156 | T7 TEST INTERRUPT FROM CONTROLLER READY |
| 1227 | T10 TEST NO INTERRUPT ON CONTROLLER READY & CPU AT LEVEL 7 |
| 1277 | T11 TEST FOR AN INTERRUPT ON LEVEL 7 |
| 1376 | T12 TEST FOR AN INTERRUPT ON LEVEL 6 |
| 1475 | T13 TEST FOR AN INTERRUPT ON LEVEL 5 |
| 1574 | T14 TEST FOR AN INTERRUPT ON LEVEL 4 |
| 1668 | T15 TEST FOR AN INTERRUPT ON LEVEL 3 |
| 1767 | T16 TEST FOR AN INTERRUPT ON LEVEL 2 |
| 1866 | T17 TEST FOR AN INTERRUPT ON LEVEL 1 |
| 1962 | T20 TEST NO INTERRUPT WITH IE SET & REST CLEARED |
| 2017 | T21 SIMULTANEOUS INTERRUPTS AT MORE THAN 1 LEVEL |
| 2087 | T22 NON-EXISTANT MEMORY DETECTION |
| 2176 | T23 EXECUTE DATI,DATOB(LOW BYTE) LOAD ON COLUMN COUNT |
| 2195 | T24 EXECUTE DATI,DATOB(HIGH BYTE) LOAD ON COLUMN COUNT |
| 2213 | T25 EXECUTE DATI,DATIP ON COLUMN COUNT REGISTER |
| 2231 | T26 EXECUTE DATI,DATOB(LOW BYTE) LOAD ON BUS ADDRESS |
| 2250 | T27 EXECUTE DATI,DATOB(HIGH BYTE) LOAD ON BUS ADDRESS |
| 2268 | T30 EXECUTE DATI,DATIP ON BUS ADDRESS REGISTER |
| 2286 | T31 WORD COUNT OVERFLOW TO 2ND CARD |
| 2384 | T32 BUS ADDRESS ODD & TRANSFER IN NON-PACK MODE |
| 2429 | DATA RELIABILITY TEST |
| 2887 | END OF PASS ROUTINE |
| 3005 | ERROR +1 FUNCTION TESTS |
| 3099 | T33 TEST DATA LATE |
| 3136 | T34 TEST ERROR AND OFF LINE BITS |
| 3377 | T36 TEST INPUT HOPPER EMPTY |
| 3506 | T37 TEST OUTPUT STACKER FULL |
| 3634 | T40 TEST PICK CHECK ERROR |
| 3791 | T41 TEST STACK CHECK ERROR |
| 3959 | T42 TEST 'END OF FILE' AND HOPPER CHECK |
| 4101 | T43 TEST READ CHECK ERROR |
| 4246 | LOOP ON TEST ROUTINE |
| 4348 | IDENTICALLY PUNCHED CARDS TEST |
| 4542 | COMMON ROUTINES |
| 4570 | ERROR MESSAGE TIMEOUT ROUTINE |
| 4619 | ERROR HANDLER ROUTINE |
| 4664 | SCOPE HANDLER ROUTINE |
| 4713 | TYPE ROUTINE |
| 4787 | BINARY TO OCTAL (ASCII) AND TYPE |

H06

MAINDEC - 11 - 0ZCDB-B MACY11 27(654) i-JUL-77 08:39
DZCDB.P.1 TABLE OF CONTENTS

SEQ 0072

| | |
|------|--------------------------------------------|
| 4866 | CONVERT BINARY TO DECIMAL AND TYPE ROUTINE |
| 4935 | TRAP DECODER |
| 4951 | TRAP TABLE |
| 4967 | POWER DOWN AND UP ROUTINES |
| 5017 | DATA TABLES |

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

.NLIST MC,MD,CND
.LIST ME
.ENABL ABS,AMA
.MCALL .HEADER,STARS,.SCMTAG,.SCATCH,.SERROR,.SERRTYP,SETPRI
.MCALL .EQUAT,.SETUP,.STYPOCT,.STYPE,.STRAP,.SPOWER,GETPRI
.MCALL NEWTST,.SSCOPE,.SEOP,.SACT11,.STYPDEC,COMMENT,ENDCOMMENT

176000

\$\$WR=176000

.TITLE MAINDEC - 11 - DZCDB-B
;*COPYRIGHT (C) 1977
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
*
;*PROGRAM BY E. RYAN
*
*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
*PACKAGE (MAINDEC-11-DZQAC-B1), AUG 29, 1975.
*
*
\$TN=1

000001

\$TN=1

;COPYRIGHT (C) 1976, 1977
;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 AVAILABLSLE 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

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.

96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137

138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191

```

*****
:STATUS AND CONTROL REGISTER (CDST) 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 PIRQ= 177772 ;: PROGRAM INTERRUPT REQUEST REGISTER
177570 DSWR= 177570 ;: HARDWARE SWITCH REGISTER
177570 DDISP= 177570 ;: HARDWARE DISPLAY REGISTER

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

```

192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245

000000
000040
000100
000140
000200
000240
000300
000340

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

.EQUIV R7,PC ;;PROGRAM COUNTER

.*PRIORITY LEVEL DEFINITIONS

PR0= 0 ;;PRIORITY LEVEL 0
PR1= 40 ;;PRIORITY LEVEL 1
PR2= 100 ;;PRIORITY LEVEL 2
PR3= 140 ;;PRIORITY LEVEL 3
PR4= 200 ;;PRIORITY LEVEL 4
PR5= 240 ;;PRIORITY LEVEL 5
PR6= 300 ;;PRIORITY LEVEL 6
PR7= 340 ;;PRIORITY LEVEL 7

.*"SWITCH REGISTER" SWITCH DEFINITIONS

SW15= 100000
SW14= 40000
SW13= 20000
SW12= 10000
SW11= 4000
SW10= 2000
SW09= 1000
SW08= 400
SW07= 200
SW06= 100
SW05= 40
SW04= 20
SW03= 10
SW02= 4
SW01= 2
SW00= 1
.EQUIV SW09,SW9
.EQUIV SW08,SW8
.EQUIV SW07,SW7
.EQUIV SW06,SW6
.EQUIV SW05,SW5
.EQUIV SW04,SW4
.EQUIV SW03,SW3
.EQUIV SW02,SW2
.EQUIV SW01,SW1
.EQUIV SW00,SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010

246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299

000004
000002
000001

BIT02= 4
BIT01= 2
BIT00= 1
.EQUIV BIT09,BIT9
.EQUIV BIT08,BIT8
.EQUIV BIT07,BIT7
.EQUIV BIT06,BIT6
.EQUIV BIT05,BIT5
.EQUIV BIT04,BIT4
.EQUIV BIT03,BIT3
.EQUIV BIT02,BIT2
.EQUIV BIT01,BIT1
.EQUIV BIT00,BIT0

;*BASIC "CPU" TRAP VECTOR ADDRESSES
ERRVEC= 4 ; TIME OUT AND OTHER ERRORS
RESVEC= 10 ; RESERVED AND ILLEGAL INSTRUCTIONS
TBITVEC=14 ; "T" BIT
TRTVEC= 14 ; TRACE TRAP
BPTVEC= 14 ; BREAKPOINT TRAP (BPT)
IOTVEC= 20 ; INPUT/OUTPUT TRAP (IOT) **SCOPE**
PWRVEC= 24 ; POWER FAIL
EMTVEC= 30 ; EMULATOR TRAP (EMT) **ERROR**
TRAPVEC=34 ; "TRAP" TRAP
TKVEC= 60 ; TTY KEYBOARD VECTOR
TPVEC= 64 ; TTY PRINTER VECTOR
PIRGVEC=240 ; PROGRAM INTERRUPT REQUEST VECTOR

;SPECIAL EQUATES

DUMMY= 0
ADINT= %2 ; CONTAINS ADDRESS OF INTERRUPT VECTOR
CDS= %3 ; CONTAINS ADDRESS OF CARD READER STATUS REGISTER
CDC= %4 ; CONTAINS ADDRESS OF CARD READER COLUMN COUNT
CDA= %5 ; CONTAINS ADDRESS OF CARD READER BUS ADDRESS REG.
TTY= %5
TRACE= 360

.SBTTL TRAP CATCHER

000000

.=0
;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

000174 000000
000176 000000

.=174
DISPREG: .WORD 0 ; SOFTWARE DISPLAY REGISTER
SWREG: .WORD 0 ; SOFTWARE SWITCH REGISTER

000200 000137 002170

.SBTTL STARTING ADDRESS(ES)
JMP @BEGIN ; JUMP TO STARTING ADDRESS OF PROGRAM

.SBTTL ACT11 HOOKS

300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329

000204
000046
015046
000052
020000
000204

000014
001242
000340

000210 000137 015642

000220 000137 024100

000240 000137 024624

000250 000137 015342

;HOOKS REQUIRED BY ACT11

SSVPC=
.=46
SENDAD
.=52
.WORD 20000
.=SSVPC

.=14
.WORD TRTRAP
.WORD 340

;SAVE PC
;;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
;;2)SET LOC.52 TO 20000
;; RESTORE PC

;ADDITIONAL STARTING ADDRESSES

.=210
JMP ERCD11

.=220
JMP TESTX

.=240
JMP CKSAME

.=250
JMP ER1200

;FOR CD11 (M1000/M200) ERROR FUNCTION TESTS

;FOR LOOP WHICH WILL CONTINUTALLY RUN
;ANY SINGLE SUBTEST

;TO READ A SINGLE DATA PATTERN
;CONTINUOUSLY

;FOR CD11 (M1200) ERROR FUNCTION TEST

007

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

\$TIMES: 0 ; ; MAX. NUMBER OF ITERATIONS
\$BELL: .ASCIZ <207><377><377> ; ; CODE FOR BEL!
\$QUES: .ASCII /?/ ; ; QUESTION MARK
\$CRLF: .ASCII <15> ; ; CARRIAGE RETURN
\$LF: .ASCIZ <12> ; ; 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 CDCC: 177162 ;ADDRESS OF CARD READER COLUMN COUNT
395 001236 177164 CDCA: 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 FLOW
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 $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
417 ;*NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
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 $ERRTB:
427
428 ;ITEM 1
429
430 001270 036272 EM1 ;STATUS REGISTER (CDS) BIT07 NOT SET BY
431 ;INITIALIZATION PULSE
432 001272 044254 DH1 ;(PC) (SP) (CDS) (CDD) (PS)
433 001274 045236 DT1 ;$ERRPC, $REG6, $TMP0, $TMP1, $REG7
434 001276 000000 0 ;OCTAL VALUES
435
436 ;ITEM 2
437
438 001300 036367 EM2 ;COLUMN COUNT REGISTER (CDC) NOT CLEARED BY
439 ;INITIALIZATION PULSE
440 001302 044322 DH2 ;(PC) (SP) (CDS) (CDD) (CDC) (PS)
441 001304 045252 DT2 ;$ERRPC, $REG6, $TMP0, $TMP1, $TMP2, $REG7
442 001306 000000 0 ;OCTAL VALUES

```

| | | | | | |
|-----|--------|--------|------|--|--|
| 443 | | | | | |
| 444 | | | | | |
| 445 | | | | | |
| 446 | 001310 | 036470 | EM3 | | |
| 447 | | | | | |
| 448 | 001312 | 044400 | DH3 | | |
| 449 | 001314 | 045270 | DT3 | | |
| 450 | 001316 | 000000 | 0 | | |
| 451 | | | | | |
| 452 | | | | | |
| 453 | | | | | |
| 454 | 001320 | 036570 | EM4 | | |
| 455 | 001322 | 044254 | DH1 | | |
| 456 | 001324 | 045236 | DT1 | | |
| 457 | 001326 | 000000 | 0 | | |
| 458 | | | | | |
| 459 | | | | | |
| 460 | | | | | |
| 461 | 001330 | 036633 | EM5 | | |
| 462 | | | | | |
| 463 | 001332 | 044322 | DH2 | | |
| 464 | 001334 | 045252 | DT2 | | |
| 465 | 001336 | 000000 | 0 | | |
| 466 | | | | | |
| 467 | | | | | |
| 468 | | | | | |
| 469 | 001340 | 036734 | EM6 | | |
| 470 | | | | | |
| 471 | 001342 | 044322 | DH2 | | |
| 472 | 001344 | 045252 | DT2 | | |
| 473 | 001346 | 000000 | 0 | | |
| 474 | | | | | |
| 475 | | | | | |
| 476 | | | | | |
| 477 | 001350 | 037024 | EM7 | | |
| 478 | | | | | |
| 479 | 001352 | 044400 | DH3 | | |
| 480 | 001354 | 045270 | DT3 | | |
| 481 | 001356 | 000000 | 0 | | |
| 482 | | | | | |
| 483 | | | | | |
| 484 | | | | | |
| 485 | 001360 | 037124 | EM10 | | |
| 486 | | | | | |
| 487 | 001362 | 044400 | DH3 | | |
| 488 | 001364 | 045270 | DT3 | | |
| 489 | 001366 | 000000 | 0 | | |
| 490 | | | | | |
| 491 | | | | | |
| 492 | | | | | |
| 493 | 001370 | 037213 | EM11 | | |
| 494 | | | | | |
| 495 | 001372 | 044254 | DH1 | | |
| 496 | 001374 | 045236 | DT1 | | |

; ITEM 3

; BUS ADDRESS REGISTER (CDA) NOT CLEARED BY
 ; INITIALIZATION PULSE
 ; (PC) (SP) (CDS) (CDD) (CDA) (PS)
 ; SERRPC, SREG6, STMP0, STMP1, STMP3, SREG7
 ; OCTAL VALUES

; ITEM 4

; STATUS REGISTER CONTENTS INCORRECT
 ; (PC) (SP) (CDS) (CDD) (PS)
 ; SERRPC, SREG6, STMP0, STMP1, SREG7
 ; OCTAL VALUES

; ITEM 5

; COLUMN COUNT REGISTER (CDC) NOT ABLE TO
 ; BE LOADED WITH ALL ONE'S
 ; (PC) (SP) (CDS) (CDD) (CDC) (PS)
 ; SERRPC, SREG6, STMP0, STMP1, STMP2, SREG7
 ; OCTAL VALUES

; ITEM 6

; COLUMN COUNT REGISTER (CDC) NOT CLEARED
 ; BY POWER CLEAR
 ; (PC) (SP) (CDS) (CDD) (CDC) (PS)
 ; SERRPC, SREG6, STMP0, STMP1, STMP2, SREG7
 ; OCTAL VALUES

; ITEM 7

; BUS ADDRESS REGISTER (CDA) NOT ABLE TO BE
 ; LOADED WITH ALL ONE'S
 ; (PC) (SP) (CDS) (CDD) (CDA) (PS)
 ; SERRPC, SREG6, STMP0, STMP1, STMP3, SREG7
 ; OCTAL VALUES

; ITEM 10

; BUS ADDRESS REGISTER (CDA) NOT CLEARED
 ; BY POWER CLEAR
 ; (PC) (SP) (CDS) (CDD) (CDA) (PS)
 ; SERRPC, SREG6, STMP0, STMP1, STMP3, SREG7
 ; OCTAL VALUES

; ITEM 11

; CONTROLLER READY DIDN'T CLEAR ON
 ; READING A CARD
 ; (PC) (SP) (CDS) (CDD) (PS)
 ; SERRPC, SREG6, STMP0, STMP1, SREG7

| | | | | |
|-----|--------|--------|------|---------------------------------------------------|
| 497 | 001376 | 000000 | 0 | ;OCTAL VALUES |
| 498 | | | | |
| 499 | | | | |
| 500 | | | | |
| 501 | 001400 | 037275 | EM12 | ;CONTROLLER READY DIDN'T CLEAR BIT00 |
| 502 | | | | ;OF STATUS REGISTER (CDS) |
| 503 | 001402 | 044254 | DH1 | ; (PC) (SP) (CDS) (CDD) (PS) |
| 504 | 001404 | 045236 | DT1 | ;SERRPC,\$REG6,\$TMPO,\$TMP1,\$REG7 |
| 505 | 001406 | 000000 | 0 | ;OCTAL VALUES |
| 506 | | | | |
| 507 | | | | |
| 508 | | | | |
| 509 | 001410 | 037374 | EM13 | ;CONTROLLER DIDN'T SET WITHIN ONE SECOND |
| 510 | 001412 | 044254 | DH1 | ; (PC) (SP) (CDS) (CDD) (PS) |
| 511 | 001414 | 045236 | DT1 | ;SERRPC,\$REG6,\$TMPO,\$TMP1,\$REG7 |
| 512 | 001416 | 000000 | 0 | ;OCTAL VALUES |
| 513 | | | | |
| 514 | | | | |
| 515 | | | | |
| 516 | 001420 | 037443 | EM14 | ;ERROR (BIT15) SET IN STATUS REGISTER (CDS) |
| 517 | 001422 | 044254 | DH1 | ; (PC) (SP) (CDS) (CDD) (PS) |
| 518 | 001424 | 045236 | DT1 | ;SERRPC,\$REG6,\$TMPO,\$TMP1,\$REG7 |
| 519 | 001426 | 000000 | 0 | ;OCTAL VALUES |
| 520 | | | | |
| 521 | | | | |
| 522 | | | | |
| 523 | 001430 | 037516 | EM15 | ;BIT/S SET IN STATUS REGISTER (CDS) THAT |
| 524 | | | | ;SHOULDN'T BE |
| 525 | 001432 | 044254 | DH1 | ; (PC) (SP) (CDS) (CDD) (PS) |
| 526 | 001434 | 045236 | DT1 | ;SERRPC,\$REG6,\$TMPO,\$TMP1,\$REG7 |
| 527 | 001436 | 000000 | 0 | ;OCTAL VALUES |
| 528 | | | | |
| 529 | | | | |
| 530 | | | | |
| 531 | 001440 | 037613 | EM16 | ; 'BUSY' SET IN STATUS REGISTER (CDS) |
| 532 | | | | ;SHOULDN'T BE |
| 533 | 001442 | 044254 | DH1 | ; (PC) (SP) (CDS) (CDD) (PS) |
| 534 | 001444 | 045236 | DT1 | ;SERRPC,\$REG6,\$TMPO,\$TMP1,\$REG7 |
| 535 | 001446 | 000000 | 0 | ;OCTAL VALUES |
| 536 | | | | |
| 537 | | | | |
| 538 | | | | |
| 539 | 001450 | 037677 | EM17 | ;RESTORING CPU STATUS AFTER READING A CARD |
| 540 | | | | ;CLEARED CONTROLLER READY IN STATUS REGISTER |
| 541 | 001452 | 044254 | DH1 | ; (PC) (SP) (CDS) (CDD) (PS) |
| 542 | 001454 | 045236 | DT1 | ;SERRPC,\$REG6,\$TMPO,\$TMP1,\$REG7 |
| 543 | 001456 | 000000 | 0 | ;OCTAL VALUES |
| 544 | | | | |
| 545 | | | | |
| 546 | | | | |
| 547 | 001460 | 040026 | EM20 | ;NO INTERRUPT OCCURRED |
| 548 | 001462 | 044456 | DH4 | ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS) |
| 549 | 001464 | 045306 | DT4 | ;SERRPC,\$REG6,\$TMPO,\$TMP1,\$TMP2,\$TMP3,\$REG7 |
| 550 | 001466 | 000000 | 0 | ;OCTAL VALUES |

| | | | | | | | | | |
|-----|--------|--------|--|------|--|--|--|--|--|
| 551 | | | | | | | | | |
| 552 | | | | | | | | | |
| 553 | | | | | | | | | |
| 554 | 001470 | 040054 | | EM21 | | | | | |
| 555 | 001472 | 044456 | | DH4 | | | | | |
| 556 | 001474 | 045306 | | DT4 | | | | | |
| 557 | 001476 | 000000 | | 0 | | | | | |
| 558 | | | | | | | | | |
| 559 | | | | | | | | | |
| 560 | | | | | | | | | |
| 561 | 001500 | 040124 | | EM22 | | | | | |
| 562 | 001502 | 044456 | | DH4 | | | | | |
| 563 | 001504 | 045306 | | DT4 | | | | | |
| 564 | 001506 | 000000 | | 0 | | | | | |
| 565 | | | | | | | | | |
| 566 | | | | | | | | | |
| 567 | | | | | | | | | |
| 568 | 001510 | 040201 | | EM23 | | | | | |
| 569 | 001512 | 044456 | | DH4 | | | | | |
| 570 | 001514 | 045306 | | DT4 | | | | | |
| 571 | 001516 | 000000 | | 0 | | | | | |
| 572 | | | | | | | | | |
| 573 | | | | | | | | | |
| 574 | | | | | | | | | |
| 575 | 001520 | 040232 | | EM24 | | | | | |
| 576 | 001522 | 044456 | | DH4 | | | | | |
| 577 | 001524 | 045306 | | DT4 | | | | | |
| 578 | 001526 | 000000 | | 0 | | | | | |
| 579 | | | | | | | | | |
| 580 | | | | | | | | | |
| 581 | | | | | | | | | |
| 582 | 001530 | 040306 | | EM25 | | | | | |
| 583 | 001532 | 044456 | | DH4 | | | | | |
| 584 | 001534 | 045306 | | DT4 | | | | | |
| 585 | 001536 | 000000 | | 0 | | | | | |
| 586 | | | | | | | | | |
| 587 | | | | | | | | | |
| 588 | | | | | | | | | |
| 589 | 001540 | 040364 | | EM26 | | | | | |
| 590 | | | | | | | | | |
| 591 | 001542 | 044456 | | DH4 | | | | | |
| 592 | 001544 | 045306 | | DT4 | | | | | |
| 593 | 001546 | 000000 | | 0 | | | | | |
| 594 | | | | | | | | | |
| 595 | | | | | | | | | |
| 596 | | | | | | | | | |
| 597 | 001550 | 040461 | | EM27 | | | | | |
| 598 | | | | | | | | | |
| 599 | 001552 | 044456 | | DH4 | | | | | |
| 600 | 001554 | 045306 | | DT4 | | | | | |
| 601 | 001556 | 000000 | | 0 | | | | | |
| 602 | | | | | | | | | |
| 603 | | | | | | | | | |
| 604 | | | | | | | | | |

;ITEM 21

;AN INTERRUPT OCCURRED - SHOULDN'T HAVE
; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$TMP3, \$REG7
;OCTAL VALUES

;ITEM 22

;INTERRUPT ALREADY OCCURRED AT A HIGHER LEVEL
; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$TMP3, \$REG7
;OCTAL VALUES

;ITEM 23

;CONTROLLER READY NOT SET
; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$TMP3, \$REG7
;OCTAL VALUES

;ITEM 24

;INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$TMP3, \$REG7
;OCTAL VALUES

;ITEM 25

;AN INTERRUPT OCCURRED AT TWO DIFFERENT LEVELS
; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$TMP3, \$REG7
;OCTAL VALUES

;ITEM 26

;ERROR (BIT15) NOT SET IN STATUS REGISTER
; (CDS) - SHOULD BE
; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$TMP3, \$REG7
;OCTAL VALUES

;ITEM 27

;NON-EXISTANT MEMORY (BIT09) NOT SET IN
;STATUS REGISTER (CDS) - SHOULD BE
; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS)
;SERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$TMP3, \$REG7
;OCTAL VALUES

;ITEM 30

| | | | | |
|-----|--------|--------|------|-----------------------------------------------------------|
| 605 | 001560 | 040576 | EM30 | : EXTENDED MEMORY (BIT05) NOT SET IN STATUS |
| 606 | | | | : REGISTER(CDS) |
| 607 | 001562 | 044456 | DH4 | : (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS) |
| 608 | 001564 | 045306 | DT4 | : SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 |
| 609 | 001566 | 000000 | 0 | : OCTAL VALUES |
| 610 | | | | |
| 611 | | | | : ; ITEM 31 |
| 612 | | | | |
| 613 | 001570 | 040670 | EM31 | : EXTENDED MEMORY (BIT04) NOT SET IN STATUS |
| 614 | | | | : REGISTER (CDS) |
| 615 | 001572 | 044456 | DH4 | : (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS) |
| 616 | 001574 | 045306 | DT4 | : SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 |
| 617 | 001576 | 000000 | 0 | : OCTAL VALUES |
| 618 | | | | |
| 619 | | | | : ; ITEM 32 |
| 620 | | | | |
| 621 | 001600 | 037516 | EM15 | : |
| 622 | 001602 | 044456 | DH4 | : (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS) |
| 623 | 001604 | 045306 | DT4 | : SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 |
| 624 | 001606 | 000000 | 0 | : OCTAL VALUES |
| 625 | | | | |
| 626 | | | | : ; ITEM 33 |
| 627 | | | | |
| 628 | 001610 | 040762 | EM32 | : CONTENTS OF BUS ADDRESS REGISTER (CDA) |
| 629 | | | | : INCORRECT |
| 630 | 001612 | 044544 | DH5 | : (PC) (SP) (CDS) (CDD) (CDC) (CDA) (CDA) (PS) |
| 631 | | | | : WAS SHB |
| 632 | 001614 | 045326 | DT5 | : SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, STMP4, SREG7 |
| 633 | 001616 | 000000 | 0 | : OCTAL VALUES |
| 634 | | | | |
| 635 | | | | : ; ITEM 34 |
| 636 | | | | |
| 637 | 001620 | 041044 | EM33 | : CONTENTS OF COLUMN COUNT REGISTER (CDC) |
| 638 | | | | : INCORRECT |
| 639 | 001622 | 044730 | DH6 | : (PC) (SP) (CDS) (CDD) (CDC) (CDA) (CDC) (PS) |
| 640 | | | | : WAS SHB |
| 641 | 001624 | 045326 | DT5 | : SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, STMP4, SREG7 |
| 642 | 001626 | 000000 | 0 | : OCTAL VALUES |
| 643 | | | | |
| 644 | | | | : ; ITEM 35 |
| 645 | | | | |
| 646 | 001630 | 041127 | EM34 | : READER OFF-LINE BUT CARD READER ERROR |
| 647 | | | | : (BIT14) NOT SET IN STATUS REGISTER (CDS) |
| 648 | 001632 | 044254 | DH1 | : (PC) (SP) (CDS) (CDD) (PS) |
| 649 | 001634 | 045236 | DT1 | : SERRPC, SREG6, STMP0, STMP1, SREG7 |
| 650 | 001636 | 000000 | 0 | : OCTAL VALUES |
| 651 | | | | |
| 652 | | | | : ; ITEM 36 |
| 653 | | | | |
| 654 | 001640 | 041247 | EM35 | : NO TRANSITION TO ON-LINE (BIT03) OCCURRED |
| 655 | | | | : IN STATUS REGISTER (CDS) |
| 656 | 001642 | 044254 | DH1 | : (PC) (SP) (CDS) (CDD) (PS) |
| 657 | 001644 | 045236 | DT1 | : SERRPC, SREG6, STMP0, STMP1, SREG7 |
| 658 | 001646 | 000000 | 0 | : OCTAL VALUES |

| | | | | | | | | | |
|-----|--------|--------|------|--|--|--|--|--|--|
| 659 | | | | | | | | | |
| 660 | | | | | | | | | |
| 661 | | | | | | | | | |
| 662 | 001650 | 043627 | EM63 | | | | | | |
| 663 | 001652 | 044456 | DH4 | | | | | | |
| 664 | 001654 | 045306 | DT4 | | | | | | |
| 665 | 001656 | 000000 | 0 | | | | | | |
| 666 | | | | | | | | | |
| 667 | | | | | | | | | |
| 668 | | | | | | | | | |
| 669 | 001660 | 041353 | EM36 | | | | | | |
| 670 | | | | | | | | | |
| 671 | 001662 | 045114 | DH7 | | | | | | |
| 672 | | | | | | | | | |
| 673 | 001664 | 045350 | DT6 | | | | | | |
| 674 | 001666 | 000000 | 0 | | | | | | |
| 675 | | | | | | | | | |
| 676 | | | | | | | | | |
| 677 | | | | | | | | | |
| 678 | 001670 | 041432 | EM37 | | | | | | |
| 679 | 001672 | 044456 | DH4 | | | | | | |
| 680 | 001674 | 045306 | DT4 | | | | | | |
| 681 | 001676 | 000000 | 0 | | | | | | |
| 682 | | | | | | | | | |
| 683 | | | | | | | | | |
| 684 | | | | | | | | | |
| 685 | 001700 | 041301 | EM40 | | | | | | |
| 686 | | | | | | | | | |
| 687 | 001702 | 044456 | DH4 | | | | | | |
| 688 | 001704 | 045306 | DT4 | | | | | | |
| 689 | 001706 | 000000 | 0 | | | | | | |
| 690 | | | | | | | | | |
| 691 | | | | | | | | | |
| 692 | | | | | | | | | |
| 693 | 001710 | 041561 | EM41 | | | | | | |
| 694 | | | | | | | | | |
| 695 | 001712 | 044456 | DH4 | | | | | | |
| 696 | 001714 | 045306 | DT4 | | | | | | |
| 697 | 001716 | 000000 | 0 | | | | | | |
| 698 | | | | | | | | | |
| 699 | | | | | | | | | |
| 700 | | | | | | | | | |
| 701 | 001720 | 041646 | EM42 | | | | | | |
| 702 | | | | | | | | | |
| 703 | 001722 | 044456 | DH4 | | | | | | |
| 704 | 001724 | 045306 | DT4 | | | | | | |
| 705 | 001726 | 000000 | 0 | | | | | | |
| 706 | | | | | | | | | |
| 707 | | | | | | | | | |
| 708 | | | | | | | | | |
| 709 | 001730 | 041743 | EM43 | | | | | | |
| 710 | | | | | | | | | |
| 711 | 001732 | 044456 | DH4 | | | | | | |
| 712 | 001734 | 045306 | 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)
 ; WAS SHB
 ; 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 | 042032 | EM44 | ;STACK CHECK (BIT12) NOT SET IN STATUS | |
| 718 | | | | ;REGISTER #2 (CDD) | |
| 719 | 001742 | 044456 | DH4 | (PC) (SP) (CDS) (CDD) (CDC) (CDA) | (PS) |
| 720 | 001744 | 045306 | DT4 | ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 | |
| 721 | 001746 | 000000 | 0 | ;OCTAL VALUES | |
| 722 | | | | | |
| 723 | | | | ;ITEM 47 | |
| 724 | | | | | |
| 725 | 001750 | 042122 | EM45 | ;END OF FILE (BIT13) SET IN STATUS | |
| 726 | | | | ;REGISTER (CDS) - SHOULDN'T BE | |
| 727 | 001752 | 044456 | DH4 | (PC) (SP) (CDS) (CDD) (CDC) (CDA) | (PS) |
| 728 | 001754 | 045306 | DT4 | ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 | |
| 729 | 001756 | 000000 | 0 | ;OCTAL VALUES | |
| 730 | | | | | |
| 731 | | | | ;ITEM 50 | |
| 732 | | | | | |
| 733 | 001760 | 042230 | EM46 | ;READ CHECK (BIT14) SET IN STATUS | |
| 734 | | | | ;REGISTER (CDS) - SHOULDN'T BE | |
| 735 | 001762 | 044456 | DH4 | (PC) (SP) (CDS) (CDD) (CDC) (CDA) | (PS) |
| 736 | 001764 | 045306 | DT4 | ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 | |
| 737 | 001766 | 000000 | 0 | ;OCTAL VALUES | |
| 738 | | | | | |
| 739 | | | | ;ITEM 51 | |
| 740 | | | | | |
| 741 | 001770 | 042335 | EM47 | ;HOPPER CHECK (BIT02) SET IN STATUS | |
| 742 | | | | ;REGISTER (CDS) - SHOULDN'T BE | |
| 743 | 001772 | 044456 | DH4 | (PC) (SP) (CDS) (CDD) (CDC) (CDA) | (PS) |
| 744 | 001774 | 045306 | DT4 | ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 | |
| 745 | 001776 | 000000 | 0 | ;OCTAL VALUES | |
| 746 | | | | | |
| 747 | | | | ;ITEM 52 | |
| 748 | | | | | |
| 749 | 002000 | 042444 | EM50 | ;END OF FILE (BIT13) OF STATUS REGISTER | |
| 750 | | | | ; (CDS) NOT SET - SHOULD BE | |
| 751 | 002002 | 044456 | DH4 | (PC) (SP) (CDS) (CDD) (CDC) (CDA) | (PS) |
| 752 | 002004 | 045306 | DT4 | ;SERRPC, STMP0, STMP1, STMP2, STMP3, SREG7 | |
| 753 | 002006 | 000000 | 0 | ;OCTAL VALUES | |
| 754 | | | | | |
| 755 | | | | ;ITEM 53 | |
| 756 | | | | | |
| 757 | 002010 | 042552 | EM51 | ;READ CHECK (BIT14) OF STATUS REGISTER | |
| 758 | | | | ; (CDS) NOT SET - SHOULD BE | |
| 759 | 002012 | 044456 | DH4 | (PC) (SP) (CDS) (CDD) (CDC) (CDA) | (PS) |
| 760 | 002014 | 045306 | DT4 | ;SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 | |
| 761 | 002016 | 000000 | 0 | ;OCTAL VALUES | |
| 762 | | | | | |
| 763 | | | | ;ITEM 54 | |
| 764 | | | | | |
| 765 | 002020 | 042657 | EM52 | ;HOPPER CHECK (BIT12) OF STATUS REGISTER | |
| 766 | | | | ; (CDS) NOT SET - SHOULD BE | |

| Line | Code | Address | Value | Description | Flags |
|------|--------|---------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 821 | 002112 | 044456 | DH4 | ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS) ; SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 ; OCTAL VALUES | |
| 822 | 002114 | 045306 | DT4 | | |
| 823 | 002116 | 000000 | 0 | | |
| 824 | | | | ; ITEM 64 | |
| 827 | 002120 | 043533 | EM62 | ; NON-EXISTANT MEMORY ERROR (BIT09) OF ; STATUS REGISTER (CDS) ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS) ; SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 ; OCTAL VALUES | |
| 829 | 002122 | 044456 | DH4 | | |
| 830 | 002124 | 045306 | DT4 | | |
| 831 | 002126 | 000000 | 0 | | |
| 832 | | | | ; ITEM 65 | |
| 834 | 002130 | 043677 | EM64 | ; DATA PACKING (BIT01) OF STATUS REGISTER ; (CDS) NOT SET - IT SHOULD BE ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS) ; SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 ; OCTAL VALUES | |
| 837 | 002132 | 044456 | DH4 | | |
| 838 | 002134 | 045306 | DT4 | | |
| 839 | 002136 | 000000 | 0 | | |
| 840 | | | | ; ITEM 66 | |
| 843 | 002140 | 044005 | EM65 | ; SHOULD BE ADDRESSING BINARY DECK - ; PROGRAM DOESN'T AGREE ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS) ; SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 ; OCTAL VALUES | |
| 845 | 002142 | 044456 | DH4 | | |
| 846 | 002144 | 045306 | DT4 | | |
| 847 | 002146 | 000000 | 0 | | |
| 848 | | | | ; ITEM 67 | |
| 851 | 002150 | 044076 | EM66 | ; CONTENTS OF STATUS REGISTER (CDS) ; INCORRECT - SHOULD BE ZERO ; (PC) (SP) (CDS) (CDD) (PS) ; SERRPC, SREG6, STMP0, STMP1, SREG7 ; OCTAL VALUES | |
| 853 | 002152 | 044254 | DH1 | | |
| 854 | 002154 | 045236 | DT1 | | |
| 855 | 002156 | 000000 | 0 | | |
| 856 | | | | ; ITEM 70 | |
| 858 | 002160 | 044175 | EM67 | ; ODD BUS ADDRESS CAUSED A TRAP IN ; NON-PACK MODE ; (PC) (SP) (CDS) (CDD) (CDC) (CDA) (PS) ; SERRPC, SREG6, STMP0, STMP1, STMP2, STMP3, SREG7 ; OCTAL VALUES | |
| 861 | 002162 | 044156 | DH4 | | |
| 862 | 002164 | 045306 | DT4 | | |
| 863 | 002166 | 000000 | 0 | | |
| 864 | | | | | |

```

865
866
867
868 002170
869 002170 012706 001100 BEGIN: MOV #SCMTAG,R6 ; FIRST LOCATION TO BE CLEARED
870 002174 005036 CLR (R6)+ ; CLEAR MEMORY LOCATION
871 002176 022706 001126 CMP #BODAT,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 026326 000020 MOV #SCOPE,#IOTVEC ; IOT VECTOR FOR SCOPE ROUTINE
875 002216 012737 000340 000022 MOV #340,#IOTVEC+2 ; LEVEL 7
876 002224 012737 026152 000030 MOV #ERROR,#EMTVEC ; EMT VECTOR FOR ERROR ROUTINE
877 002232 012737 000340 000032 MOV #340,#EMTVEC+2 ; LEVEL 7
878 002240 012737 027374 000034 MOV #TRAP,#TRAPVEC ; TRAP VECTOR FOR TRAP CALLS
879 002246 012737 000340 000036 MOV #340,#TRAPVEC+2 ; LEVEL 7
880 002254 012737 027430 000024 MOV #SPWRN,#PWAVEC ; POWER FAILURE VECTOR
881 002262 012737 000340 000026 MOV #340,#PWAVEC+2 ; LEVEL 7
882 002270 013737 014774 014766 MOV SENDCT,SEOPCT ; SETUP END-OF-PROGRAM COUNTER
883 002276 005037 001220 CLR $TIMES ; INITIALIZE NUMBER OF ITERATIONS
884 002302 012737 015140 000014 MOV $SRTN,#TBITVEC ; SET "T" BIT VECTOR TO $SRTN
885 002310 012737 000340 000016 MOV #340,#TBITVEC+2 ; LEVEL 7
886 002316 012737 000002 015140 MOV $RTI,$SRTN ; SET $SRTN TO A RTI
887 002324 012737 002352 000010 MOV #65,$RESVEC ; TRY TO DO A RTT
888 002332 005046 CLR -(SP) ; DUMMY PS
889 002334 012746 002342 MOV #64$,-(SP) ; AND PC
890 002340 000006 RTT ; TRY THE RTT
891 002342 000006 015140 64$: MOV #RTT,$SRTN ; RTT IS LEGAL--SET $SRTN TO A RTT
892 002350 000402 BR 66$ ;
893 002352 062706 000010 65$: ADD #10,SP ; RTT ILLEGAL--CLEAN OFF THE STACK
894 002356 012737 000012 000010 66$: MOV #RESVEC+2,#RESVEC ; RESTORE TRAP CATCHER
895 002364 005037 015146 CLR $TBIT ; CLEAR "T" BIT SWITCH
896 002370 012737 002370 001106 MOV #,$SLPADR ; INITIALIZE THE LOOP ADDRESS FOR SCOPE
897 002376 013746 000004 MOV #4,-(SP) ; SAVE ERROR VECTOR
898 002402 013746 000006 MOV #6,-(SP)
899 002406 012737 002422 000004 MOV #67$,4 ; SET UP TIME OUT VECTOR
900 002414 005777 176516 TST $SWR ; TRY TO REFERENCE HARDWARE SWR
901 002420 000407 BR 68$ ; BRANCH IF NO TIMEOUT TRAP OCCURS
902 002422 012737 000176 001136 67$: MOV #SWREG,SWR ; POINT TO SOFTWARE SWR
903 002430 012737 000174 001140 MOV #DISPREG,DISPLAY ; POINT TO SOFTWARE DISPLAY REG
904 002436 022626 CMP (SP)+,(SP)+ ; RESTORE STACK
905 002440 012637 000006 68$: MOV (SP)+,#6 ; RESTORE ERROR VECTOR
906 002444 012637 000004 MOV (SP)+,#4
907 002450 012737 002170 027576 MOV #BEGIN,RETURN ; SAVE RETURN POINT FOR THIS
908 ; SECTION FOR POWER FAILURE RETURN
909 002456 004737 045366 JSR PC, SETUP ; INITIALIZE POINTERS AND FLAGS
910 002462 005737 000042 TST #42 ; TEST FOR MONITOR
911 002466 001015 BNE HEADIN ; BRANCH IF MONITOR IS IN CONTROL
912 002470 104400 030556 TYPE, $TIMES ; TYPE MAINDEC TITLE & REV. LEVEL
913 002474 104400 030616 TYPE, $STADDR ; TYPE STARTING ADDRESSES MESSAGE
914 002500 104400 030550 TYPE, CRLF-3
915 002504 104400 035463 TYPE, MSG31 ; USER IS TO SET SR SWITCHES
916 002510 104400 031632 TYPE, MSG2 ; HIT CONTINUE
917 002514 000000 HALT
918 002516 104400 030553 TYPE, CRLF

```

```

919 002522 104400 031256 HEADIN: TYPE, MLOGIC ;INDICATE "ENTERING LOGIC TESTS"
920 002526 104400 015165 TYPE, SNULL ;TIME TO FINISH ABOVE MESSAGE.
921 002532 000432 BR TST1 ;GO TO INSTRUCTION TESTS
922 002534 005737 001254 RESTRY: TST TRFLG ;CHECK FOR TRACE TRAPPING
923 002540 001012 BNE TRAPX ;IF SET, TRACE TRAP
924 002542
925 002542 013746 000340 NOTRP: MOV PR7, -(SP) ;: PUT NEW PS ON STACK
926 002546 012746 002554 MOV #64$, -(SP) ;: PUT NEW PC ON STACK
927 002552 000002 RTI ;: POP NEW PC AND PS
928 002554
929 002554 104400 031256 64$: TYPE, MLOGIC ;INDICATE "ENTERING LOGIC TESTS"
930 002560 104400 015165 TYPE, SNULL ;TIME TO FINISH ABOVE MESSAGE
931 002564 000415 BR TST1 ;GO TO INSTRUCTION TESTS
932 002566 104400 031256 TRAPX: TYPE, MLOGIC ;INDICATE "ENTERING LOGIC TESTS"
933 002572 104400 015165 TYPE, SNULL ;TIME TO FINISH ABOVE MESSAGE
934 002576 032777 010000 176332 BIT #10000, %SWR ;CHECK SW12
935 002604 001356 BNE NOTRP ;BRANCH IF SET TO CLEAR TRACE BIT
936 002606 013746 000360 MOV TRACE, -(SP) ;: PUT NEW PS ON STACK
937 002612 012746 002620 MOV #64$, -(SP) ;: PUT NEW PC ON STACK
938 002616 000002 RTI ;: POP NEW PC AND PS
939 002620
940
941
942
943
944 002620 000004 *TEST 1 TEST FOR INIT. OF ALL REGISTERS
945 002622 004737 025772 TST1: SCOPE
946 002626 000005 JSR PC, CKOFFL ;CHECK FOR OFF-LINE SET
947 002630 022713 000200 RESET ;SEND OUT INIT
948 002634 001401 CMP #200, %CDS ;CHECK FOR STATUS REGISTER BIT 7 SET
949 002636 104001 BEQ 1$ ;BRANCH IF OK
950 ;STATUS REGISTER NOT CORRECTLY INITIALIZED
951 002640 005714 1$: TST %CDC ;CHECK FOR COLUMN COUNT CLEARED
952 002642 001401 BEQ 2$ ;BR IF OK
953 002644 104302 ERROR +2 ;COLUMN COUNT NOT CLEARED BY INIT
954
955 002646 005715 2$: TST %CDA ;CHECK FOR BUS ADDRESS CLEARED
956 002650 001401 BEQ 3$ ;BR IF OK
957 002652 104003 ERROR +3 ;BUS ADDRESS NOT CLEARED BY INIT
958
959 002654 005777 176360 3$: TST %CDD8 ;TEST BIT15 OF STATUS REGISTER #2
960 002660 100011 BPL 4$ ;BRANCH IF NOT SET INDICATING
961 ;OLD CD11 CONTROLLER
962 002662 022777 107777 176350 CMP #107777, %CDD8 ;IS CONTENTS OF STATUS REGISTER #2
963 ;CORRECT FOR NO ERRORS?
964 002670 001415 BEQ 5$ ;BRANCH IF OK!
965 002672 012737 107777 001210 MOV #107777, %STMP4 ;CORRECT CONTENTS OF 'CDD' FOR
966 ;ERROR REPORT
967 002700 104040 ERROR +40 ;CONTENTS OF 'CDD8' STATUS REGISTER
968 ;#2 NOT = 107777
969 002702 000410 BR 5$ ;GO TO NEXT TEST
970 002704 022777 007777 176326 4$: CMP #007777, %CDD8 ;WE HAVE AN OLD CD11 CONTROLLER!
971 ;IS CONTENTS OF STATUS REGISTER #2
972 ;CORRECT FOR NO ERRORS?

```

```

973 002712 001404          BEQ      5$          ;BRANCH IF OK!
974 002714 012737 007777 001210  MOV      #007777,STMP4 ;CORRECT CONTENTS OF 'CDD' FOR
975                                ;ERROR REPORT
976 002722 104040          ERROR +40 ;CONTENTS OF 'CDOB' STATUS REGISTER
977                                ;#2 NOT = 007777
978 002724
979
980
981
982 002724 000004          5$:
983                                ;*****
984                                ;*TEST 2      TEST READ/WRITE OF STATUS REGISTER
985                                ;*****
986                                †TST2:  SCOPE
987                                ;ONLY BITS 1,4,5, AND 6 OF THE STATUS REGISTER SHOULD BE
988                                ;ABLE TO BE SET TO ONE AND READ BACK AS ONE
989 002726 052713 177376    BIS      #177376,ACDS ;SET ALL BITS BUT 0 AND 8
990 002732 022713 000362    CMP      #362, ACDS  ;ONLY BITS 1,4,5,6, AND 7 SHOULD BE SET
991 002736 001402          BEQ      1$          ;BRANCH IF OK
992 002740 104004          ERROR +4 ;STATUS REGISTER DIDN'T CONTAIN 362
993 002742 000413          BR       TST3      ;BRANCH AFTER FAILURE
994
995                                1$:
996                                ;CLEARING STATUS REGISTER SHOULD CLEAR BITS 1,4,5, AND 6
997 002744 005013          CLR      ACDS      ;CLEAR BITS 1,4,5, AND 6
998 002746 022713 000200    CMP      #200, ACDS ;CHECK FOR ALL BITS CLEAR BUT 7
999 002752 001401          BEQ      2$          ;BRANCH IF OK
1000 002754 104004          ERROR +4 ;STATUS REGISTER DIDN'T CONTAIN 200
1001
1002                                2$:
1003                                ;SETTING ALL BITS SHOULD DO A POWER CLEAR
1004 002756 012713 177777    MOV      #177777,ACDS ;SET ALL BITS OF THE STATUS REGISTER
1005 002762 022713 000200    CMP      #200, ACDS ;CHECK FOR ALL BITS CLEAR BUT 7
1006 002766 001401          BEQ      3$          ;BRANCH IF OK
1007 002770 104004          ERROR +4 ;STATUS REGISTER DIDN'T CONTAIN 200
1008
1009                                3$:
1010                                ;*****
1011                                ;*TEST 3      TEST READ/WRITE OF COLUMN COUNT REGISTER
1012                                ;*****
1013                                †TST3:  SCOPE
1014 002772 000004          MOV      #177777,ACDC ;LOAD ALL BITS
1015 002774 012714 177777    CMP      #177777,ACDC ;TEST TO SEE IF IT CAN BE READ
1016 003000 022714 177777    BEQ      1$          ;BRANCH IF OK
1017 003004 001401          ERROR +5 ;CDCC FAILED TO READ/WRITE
1018 003006 104005
1019                                1$:
1020 003010 022713 000200    CMP      #200, ACDS ;CHECK STATUS REG
1021 003014 001401          BEQ      2$          ;BRANCH IF OK
1022 003016 104004          ERROR +4 ;STATUS REG CHANGED
1023
1024                                2$:
1025 003020 052713 000400    BIS      #400, ACDS ;DO A POWER CLEAR
1026 003024 005714          TST      ACDC      ;CHECK FOR COLUMN COUNT CLEARED
1027 003026 001401          BEQ      3$          ;BRANCH IF OK
1028 003030 104006          ERROR +6 ;COLUMN COUNT NOT CLEARED BY POWER CLEAR
1029
1030                                3$:
1031 003032 022713 000200    CMP      #200, ACDS ;CHECK STATUS REG
1032 003036 001401          BEQ      4$          ;BRANCH IF OK
1033 003040 104004          ERROR +4 ;STATUS REG CHANGED
1034
1035                                4$:

```

```

1027 ;*****
1028 ;*TEST 4 TEST READ/WRITE OF BUS ADDRESS REGISTER
1029 ;*****
1030 003042 000004          1ST4: SCOPE
1031 003044 012715 177777  MOV #177777, @CDA ;LOAD ALL BITS
1032 003050 022715 177777  CMP #177777, @CDA ;TEST TO SEE IF IT CAN BE READ
1033 003054 001401        BEQ 1$ ;BRANCH IF OK
1034 003056 104007        ERROR +7 ;CDBA FAILED TO READ/WRITE
1035
1036 003060 022713 000200  1$:  CMP #200, @CDS ;CHECK STATUS REG
1037 003064 001401        BEQ 2$ ;BRANCH IF OK
1038 003066 104004        ERROR +4 ;STATUS REG CHANGED
1039
1040 003070 052713 000400  2$:  BIS #400, @CDS ;DO A POWER CLEAR
1041 003074 005715        TST @CDA ;CHECK FOR BUS ADDRESS CLEARED
1042 003076 001401        BEQ 3$ ;BRANCH IF OK
1043 003100 104010        ERROR +10 ;BUS ADDRESS NOT CLEARED BY POWER CLEAR
1044
1045 003102 022713 000200  3$:  CMP #200, @CDS ;CHECK STATUS REG
1046 003106 001401        BEQ 4$ ;BRANCH IF OK
1047 003110 104004        ERROR +4 ;STATUS REG CHANGED
1048
1049 003112          4$:
1050 ;*****
1051 ;*TEST 5 TEST CONTROLLER READY TO CLEAR BIT 0
1052 ;*****
1053 003112 000004          1ST5: SCOPE
1054 ;BIT 0 SHOULD ALWAYS READ AS BEING EQUAL TO ZERO
1055 003114 004737 025772  JSR PC, @CKOFFL ;CHECK FOR OFF-LINE SET
1056 003120 012714 177777  MOV #-1, @CDC ;SET UP COLUMN COUNT TO READ 1 COLUMN
1057 003124 012715 045442  MOV #BUFBEQ, @CDA ;SET UP BUS ADDRESS
1058 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, PROC"
1059 003130 005046        CLR -(SP) ;
1060 003132 013746 000034  MOV 34, -(SP) ;SAVE CURRENT TRAP VECTOR
1061 003136 012737 003146 000034  MOV #64$, 34 ;SETUP NEW TRAP VECTOT
1062 003144 104400        TRAP ;PUSH OLD PSW AN PCON STACK
1063 003146 016666 000002 000006 64$: MOV 2(SP), 6(SP) ;
1064 003154 012716 003162        MOV #65$, (SP) ;
1065 003160 000002        RTI ;RESTORE PSW
1066 003162 012637 000034 65$: MOV (SP)+, 34 ;RESTORE OLD TRAP VECTOR
1067 003166 012637 001256        MOV (SP)+, PROC ;
1068 003172 013746 000000        MOV PROC, -(SP) ;PUT NEW PS ON STACK
1069 003176 012746 003204        MOV #66$, -(SP) ;PUT NEW PC ON STACK
1070 003202 000002        RTI ;POP NEW PC AND PS
1071
1072 003204 005037 001250 66$: CLR COUNT ;INITIALIZE COUNTER
1073 003210 005213        INC @CDS ;START READING A CARD
1074 003212 105713        TSTB @CDS ;CHECK FOR CONTROLLER READY CLEARED
1075 003214 100001        BPL LOOP5 ;BRANCH IF OK
1076 003216 104011        ERROR +11 ;CONTROLLER READY DIDN'T CLEAR
1077
1078 003220 032713 000001  LOOP5: BIT #1, @CDS ;CHECK BIT 0
1079 003224 001402        BEQ 1$ ;BRANCH IF NOT SET
1080 003226 104012        ERROR +12 ;BIT 0 READ AS A ONE

```



```

1081 003230 000423          BR      TST6          ;BRANCH AFTER FAILURE
1082 003232 005237 001250 1$:  INC      COUNT        ;WAIT ABOUT
1083 003236 001370          BNE     LOOP5
1084 003240 013746 001256  MOV     PROC,-(SP)    ;;PUT NEW PS ON STACK
1085 003244 012746 003252  MOV     #64$,-(SP)   ;;PUT NEW PC ON STACK
1086 003250 000002          RTI                    ;; POP NEW PC AND PS
1087 003252
1088 003252 105713          TSTB   @CDS          ;CHECK CONTROLLER READY
1089 003254 100401          BMI   2$            ;CONTINUE IF SET
1090 003256 104013          ERROR +13          ;CONTROLLER READY DIDN'T SET WITHIN 1 SEC
1091 003260 005713          2$:  TST   @CDS
1092 003262 100002          BPL   3$
1093 003264 104014          ERROR +14          ;ERROR BIT SET
1094 003266 000404          BR      TST6
1095 003270 03271? 177577 3$:  BIT   #177577,@CDS ;CHECK FOR ANY OTHER BITS
1096 003274 001401          BEQ   4$            ;BRANCH IF OK
1097 003276 104015          ERROR +15          ;EXTRA BIT(S) SET
1098
1099 003300          4$:
1100 ;*****
1101 ;*TEST 6 TEST BIT2 TO BE CLEAR AFTER CARD READ
1102 ;*****
1103 003300 000004          TST6:  SCOPE
1104 ;IT SHOULD REMAIN NOT SET
1105 ;THIS SHOULD HAPPEN WITHIN ABOUT 1 SECOND
1106 003302 004737 025772          JSR   PC_CKOFF1    ;CHECK FOR OFF-LINE SET
1107 003306 005013          CLR   @CDS        ;INITIALIZE STATUS REGISTER
1108 003310 012714 177754          MOV   #-20,@CDC   ;SET UP COLUMN COUNT TO READ 20 COLUMNS
1109 003314 012715 045442          MOV   #BUFBEG,@CDA ;SET UP BUS ADDRESS
1110 003320 005213          INC   @CDS        ;READ A CARD
1111 003322 032713 000004          BIT   #4,@CDS     ;CHECK HOPPER EMPT'
1112 003326 001401          BEQ   1$
1113 003330 104016          ERROR +16          ;HOPPER EMPTY SET
1114 003332 005037 001250 1$:  CLR   COUNT        ;SET UP WAIT COUNTER
1115 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,PROC"
1116 003336 005046          CLR   -(SP)
1117 003340 013746 000034          MOV   34,-(SP)    ;;SAVE CURRENT TRAP VECTOR
1118 003344 012737 003354 000034          MOV   #64$,34    ;;SETUP NEW TRAP VECTOT
1119 003352 104400          TRAP
1120 003354 016666 000002 000006 64$:  MOV   2(SP),6(SP) ;;
1121 003362 012716 003370          MOV   #65$,1(SP) ;;REPLACE OLD PC WITH NEW
1122 003366 000002          RTI                    ;;RESTORE PSW
1123 003370 012637 000034          65$:  MOV   (SP)+,34    ;;RESTORE OLD TRAF VECTOR
1124 003374 012637 001256          MOV   (SP)+,PROC
1125 003400 013746 000000          MOV   PROC,-(SP)  ;;PUT NEW PS ON STACK
1126 003404 012746 003412          MOV   #66$,-(SP) ;;PUT NEW PC ON STACK
1127 003410 000002          RTI                    ;; POP NEW PC AND PS
1128 003412
1129 003412 105713          LOOP6A: TSTB @CDS      ;CHECK READY
1130 003414 100405          BMI   LOOP6B        ;BRANCH IF READY
1131 003416 005337 001250          DEC   COUNT        ;WAIT ABOUT 1 SEC.
1132 003422 001?73
1133 003424 104013          BNE   LOOP6A
1134 003426 000413          ERROR +13          ;READING A CARD DIDN'T SET READY
BR      TST7

```

```

1135 003430          LOOP6B:
1136 003430 013746 001256      MOV     PROC,-(SP)      ;; PUT NEW PS ON STACK
1137 003434 012746 003442      MOV     #64$,-(SP)    ;; PUT NEW PC ON STACK
1138 003440 000002              RTI                    ;; POP NEW PC AND PS
1139 003442          64$:
1140 003442 105713      LOOP6:  TSTB     QCD5      ;CHECK CONTROLLER READY
1141 003444 100401      BMI     DONE6        ;BRANCH IF SET
1142 003446 104017      ERROR +17          ;RESTORING STATUS RESET READY
1143
1144 003450 005713      DONE6:  TST     QCD5      ;CHECK ERROR BIT 15
1145 003452 100001      BPL     1$           ;BRANCH IF OK
1146 003454 104014      ERROR +14          ;ERROR BIT 15 WAS SET
1147
1148 003456          1$:
1149
1150          ;*****
1151          ;*TEST 7 TEST INTERRUPT FROM CONTROLLER READY
1152          ;*****
1152 003456 000004      TST7:  SCOPE
1153 003460 004737 025744      JSR     PC,INIT      ;INITIALIZE
1154 003464 012712 003740      MOV     #TINT7,ADINT ;LOAD RETURN POINTER
1155          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1156 003470 005046      CLR     -(SP)        ;;
1157 003472 013746 000034      MOV     34,-(SP)     ;; SAVE CURRENT TRAP VECTOR
1158 003476 012737 003506 000034      MOV     #64$,34     ;; SETUP NEW TRAP VECTOT
1159 003504 104400      TRAP                    ;; PUSH OLD PSW AN PCON STACK
1160 003506 016666 000002 000006 64$:  MOV     2(SP),6(SP)  ;;
1161 003514 012716 003522      MOV     #65$,1$      ;; REPLACE OLD PC WITH NEW
1162 003520 000002              RTI                    ;; RESTORE PSW
1163 003522 012637 000034 65$:  MOV     (SP)+,34     ;; RESTORE OLD TRAP VECTOR
1164 003526 012637 001214      MOV     (SP)+,$TMP6
1165 003532 052737 000340 001214      BIS     #340,$TMP6
1166 003540 013746 001214      MOV     $TMP6,-(SP)  ;; PUT NEW PS ON STACK
1167 003544 012746 003552      MOV     #66$,-(SP)  ;; PUT NEW PC ON STACK
1168 003550 000002              RTI                    ;; POP NEW PC AND PS
1169 003552          66$:
1170          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1171 003552 005046      CLR     -(SP)        ;;
1172 003554 013746 000034      MOV     34,-(SP)     ;; SAVE CURRENT TRAP VECTOR
1173 003560 012737 003576 000034      MOV     #67$,34     ;; SETUP NEW TRAP VECTOT
1174 003566 104400      TRAP                    ;; PUSH OLD PSW AN PCON STACK
1175 003570 016666 000002 000006 67$:  MOV     2(SP),6(SP)  ;;
1176 003576 012716 003604      MOV     #68$,1$      ;; REPLACE OLD PC WITH NEW
1177 003602 000002              RTI                    ;; RESTORE PSW
1178 003604 012637 000034 68$:  MOV     (SP)+,34     ;; RESTORE OLD TRAP VECTOR
1179 003610 012662 000002      MOV     (SP)+,2(ADINT)
1180          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1181 003614 005046      CLR     -(SP)        ;;
1182 003616 013746 000034      MOV     34,-(SP)     ;; SAVE CURRENT TRAP VECTOR
1183 003622 012737 003632 000034      MOV     #69$,34     ;; SETUP NEW TRAP VECTOT
1184 003630 104400      TRAP                    ;; PUSH OLD PSW AN PCON STACK
1185 003632 016666 000002 000006 69$:  MOV     2(SP),6(SP)  ;;
1186 003640 012716 003646      MOV     #70$,1$      ;; REPLACE OLD PC WITH NEW
1187 003644 000002              RTI                    ;; RESTORE PSW
1188 003646 012637 000034 70$:  MOV     (SP)+,34     ;; RESTORE OLD TRAP VECTOR

```

```

1189 003652 012637 001214      MOV      (SP)+,$TMP6
1190 003656 042737 000340 001214      BIC      #340,$TMP6
1191 003664 013746 001214      MOV      $TMP6,-(SP)      ;; PUT NEW PS ON STACK
1192 003670 012746 003676      MOV      #71$,-(SP)      ;; PUT NEW PC ON STACK
1193 003674 000002      RTI      ;; POP NEW PC AND PS
1194 003676      71$:
1195 003676 012714 177741      MOV      #-31,@CDC      ;; SET UP COLUMN COUNT TO READ 31 COLUMNS
1196 003702 012715 045442      MOV      #BUFBE@CDA      ;; SET UP BUS ADDRESS
1197 003706 012713 000101      MOV      #101,@CDS      ;; SET INTERRUPT ENABLE AND READ
1198 003712 105713      1$:      TSTB     @CDS      ;; WAIT FOR CONTROLLER READY
1199 003714 100376      BPL      1$
1200      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
1201 003716 016246 000002      MOV      2(ADINT),-(SP)  ;; PUT NEW PS ON STACK
1202 003722 012746 003730      MOV      #72$,-(SP)      ;; PUT NEW PC ON STACK
1203 003726 000002      RTI      ;; POP NEW PC AND PS
1204 003730      72$:
1205 003730 042713 000100      BIC      #100,@CDS      ;; CLEAR INTERRUPT ENABLE
1206 003734 104020      ERROR +20      ;; NO INTERRUPT OCCURRED
1207 003736 070410      BR      CONT7
1208 003740 105713      TINT7:  TSTB     @CDS      ;; CHECK CONTROLLER READY
1209 003742 100401      BMI      1$      ;; BRANCH IF SET
1210 003744 104023      ERROR +23      ;; CONTROLLER READY NOT SET
1211 003746 022626      1$:      CMP      (SP)+,(SP)+    ;; RESTORE STACK POINTER
1212 003750 005713      TST      @CDS      ;; MAKE SURE NO ERROR OCCURRED
1213 003752 100001      BPL      2$
1214 003754 104014      ERROR +14      ;; BIT 15 WAS SET
1215 003756 005013      2$:      CLR      @CDS      ;; DISABLE INTERRUPTS
1216 003760 012712 000232      CONT7:  MOV      #232,@ADINT    ;; CHANGE INTERRUPT RETURN ADDRESS
1217 003764 005037 000232      CLR      @#232      ;; TO CAUSE A HALT IF AN INTERRUPT OCCURS
1218
1219      ;*****
1220      ;*TEST 10 TEST NO INTERRUPT ON CONTROLLER READY & CPU AT LEVEL 7
1221      ;*****
1222 003770 000004      TST10:  SCOPE
1223 003772 004737 025744      JSR      PC,INIT      ;; INITIALIZE
1224 003776 012712 004150      MOV      #TINT10,@ADINT ;; SETUP RETURN
1225      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1226 004002 005046      CLR      -(SP)      ;;
1227 004004 013746 000034      MOV      34,-(SP)    ;; SAVE CURRENT TRAP VECTOR
1228 004010 012737 004020 000034      MOV      #64$,34    ;; SETUP NEW TRAP VECTOR
1229 004016 104400      TRAP      ;; PUSH OLD PSW AN PCON STACK
1230 004020 016666 000002 000006 64$:      MOV      2(SP),6(SP)  ;;
1231 004026 012716 004034      MOV      #65$,SP    ;; REPLACE OLD PC WITH NEW
1232 004032 000002      RTI      ;; RESTORE PSW
1233 004034 012637 000034      65$:      MOV      (SP)+,34    ;; RESTORE OLD TRAP VECTOR
1234 004040 012637 001214      MOV      (SP)+,$TMP6
1235 004044 052737 000340 001214      BIS      #340,$TMP6
1236 004052 013746 001214      MOV      $TMP6,-(SP)  ;; PUT NEW PS ON STACK
1237 004056 012746 004064      MOV      #66$,-(SP)  ;; PUT NEW PC ON STACK
1238 004062 000002      RTI      ;; POP NEW PC AND PS
1239 004064      66$:
1240      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1241 004064 005046      CLR      -(SP)      ;;
1242 004066 013746 000034      MOV      34,-(SP)    ;; SAVE CURRENT TRAP VECTOR

```

H08

MAINDEC - 11 - DZCDB-B
DZCDB.P11 T10

MACY11 27(654) 1-JUL-77 08:39 PAGE 26
TEST NO INTERRUPT ON CONTROLLER READY & CPU AT LEVEL 7

SEQ 0098

```

1243 004072 012737 004102 000034      MOV      #67$,34      ;;SETUP NEW TRAP VECTOT
1244 004100 104400      TRAP                      ;;PUSH OLD PSW AN PCON STACK
1245 004102 016666 000002 000006 67$:      MOV      2(SP),6(SP)  ;;
1246 004110 012716 004116      MOV      #68$, (SP)  ;;
1247 004114 000002      RTI                      ;;RESTORE PSW
1248 004116 012637 000034 68$:      MOV      (SP)+,34      ;;RESTORE OLD TRAP VECTOR
1249 004122 012662 000002      MOV      (SP)+,2(ADINT)
1250 004126 012714 177703      MOV      #-61, @CDC    ;SET UP COLUMN COUNT TO READ 61 COLUMNS
1251 004132 012715 045442      MOV      #BUFBEG, @CDA  ;SET UP BUS ADDRESS
1252 004136 012713 000101      MOV      #101, @CDS    ;SET INTERRUPT ENABLE AND READ
1253 004142 105713 1$:      TSTB    @CDS          ;WAIT FOR CONTROLLER READY
1254 004144 100376      BPL     1$
1255 004146 000402      BR      T10GO
1256 004150 104021  TINT10:  ERROR +21      ;CONTINUE IF NO INTERRUPT OCCURRED
1257 004152 022626      CMP     (SP)+,(SP)+    ;AN INTERRUPT OCCURRED
1258 004154 005013  T10GO:  CLR     @CDS          ;RESTORE STACK POINTER
1259 004156 012712 000232      MOV     #232, @ADINT   ;CLEAR INTERRUPT ENABLE
1260 004162 005037 000232      CLR     @#232         ;CHANGE INTERRUPT RETURN ADDRESS
1261                                     ;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1262                                     ;FIND THE LEVEL AT WHICH AN INTERRUPT OCCURS
1263                                     ;PRINT OUT A MESSAGE STATING THIS LEVEL IF IT IS OTHER THAN THE STANDARD
1264                                     ;(LEVEL 6) MAKE CERTAIN THAT IT ALWAYS OCCURS AT THIS LEVEL
1265                                     ;THE MESSAGE STATING THE LEVEL IS PRINTED ONLY ONCE, AND THE PROGRAM MUST
1266                                     ;BE STARTED OVER AT LOCATION 200 FOR IT TO BE PRINTED AGAIN
1267
1268                                     ;*****
1269                                     ;*TEST 11 TEST FOR AN INTERRUPT ON LEVEL 7
1270                                     ;*****
1271 004166 000004  TST11:  SCOPE
1272 004170 004737 025744      JSR     PC, INIT      ;INITIALIZE
1273 004174 012712 004556      MOV     #TINT11, @ADINT ;SETUP RETURN ADDRESS
1274                                     ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1275 004200 005046      CLR     -(SP)
1276 004202 013746 000034      MOV     34, -(SP)     ;;SAVE CURRENT TRAP VECTOR
1277 004206 012737 004216 000034      MOV     #64$,34      ;;SETUP NEW TRAP VECTOT
1278 004214 104400      TRAP                      ;;PUSH OLD PSW AN PCON STACK
1279 004216 016666 000002 000006 64$:      MOV     2(SP),6(SP)  ;;
1280 004224 012716 004232      MOV     #65$, (SP)  ;;
1281 004230 000002      RTI                      ;;RESTORE PSW
1282 004232 012637 000034 65$:      MOV     (SP)+,34      ;;RESTORE OLD TRAP VECTOR
1283 004236 012637 001214      MOV     (SP)+, $TMP6
1284 004242 052737 000340 001214      BIS     #340, $TMP6
1285 004250 013746 001214      MOV     $TMP6, -(SP)  ;;PUT NEW PS ON STACK
1286 004254 012746 004262      MOV     #66$, -(SP)  ;;PUT NEW PC ON STACK
1287 004260 000002      RTI                      ;;POP NEW PC AND PS
1288 66$:
1289                                     ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1290 004262 005046      CLR     -(SP)
1291 004264 013746 000034      MOV     34, -(SP)     ;;SAVE CURRENT TRAP VECTOR
1292 004270 012737 004300 000034      MOV     #67$,34      ;;SETUP NEW TRAP VECTOT
1293 004276 104400      TRAP                      ;;PUSH OLD PSW AN PCON STACK
1294 004300 016666 000002 000006 67$:      MOV     2(SP),6(SP)  ;;
1295 004306 012716 004314      MOV     #68$, (SP)  ;;
1296 004312 000002      RTI                      ;;RESTORE PSW

```

I08

```

1297 004314 012637 000034      68$: MOV (SP)+,34 ;;RESTORE OLD TRAP VECTOR
1298 004320 012662 000002      MOV (SP)+,2(ADINT)
1299 ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1300 004324 005046      CLR -(SP) ;;
1301 004326 013746 000034      MOV 34 -(SP) ;;SAVE CURRENT TRAP VECTOR
1302 004332 012737 004342 000034      MOV #69$,34 ;;SETUP NEW TRAP VECTOT
1303 004340 104400      TRAP ;;PUSH OLD PSW AN PCON STACK
1304 004342 016666 000002 000006 69$: MOV 2(SP),6(SP) ;;
1305 004350 012716 004356      MOV #70$, (SP) ;;REPLACE OLD PC WITH NEW
1306 004354 000002      RTI ;;RESTORE PSW
1307 004356 012637 000034      70$: MOV (SP)+,34 ;;RESTORE OLD TRAP VECTOR
1308 004362 012637 001214      MOV (SP)+,$TMP6
1309 004366 042737 000340 001214      BIC #340,$TMP6
1310 004374 013746 001214      MOV $TMP6,-(SP) ;;PUT NEW PS ON STACK
1311 004400 012746 004406      MOV #71$,-(SP) ;;PUT NEW PC ON STACK
1312 004404 000002      RTI ;;POP NEW PC AND PS
1313 004406      71$:
1314 004406 005046      CLR -(SP) ;;
1315 004410 013746 000034      MOV 34 -(SP) ;;SAVE CURRENT TRAP VECTOR
1316 004414 012737 004424 000034      MOV #72$,34 ;;SETUP NEW TRAP VECTOT
1317 004422 104400      TRAP ;;PUSH OLD PSW AN PCON STACK
1318 004424 016666 000002 000006 72$: MOV 2(SP),6(SP) ;;
1319 004432 012716 004440      MOV #73$, (SP) ;;REPLACE OLD PC WITH NEW
1320 004436 000002      RTI ;;RESTORE PSW
1321 004440 012637 000034      73$: MOV (SP)+,34 ;;RESTORE OLD TRAP VECTOR
1322 004444 012637 001214      MOV (SP)+,$TMP6
1323 004450 052737 000300 001214      BIS #300,$TMP6
1324 004456 013746 001214      MOV $TMP6,-(SP) ;;PUT NEW PS ON STACK
1325 004462 012746 004470      MOV #74$,-(SP) ;;PUT NEW PC ON STACK
1326 004466 000002      RTI ;;POP NEW PC AND PS
1327 004470      74$:
1328 004470 012714 177660      MOV #-80,$CDC ;;SET UP COLUMN COUNT TO READ 80 COLUMNS
1329 004474 012715 045442      MOV #80,$CDA ;;SET UP BUS ADDRESS
1330 004500 012713 000101      MOV #101,$CDS ;;SET INTERRUPT ENABLE AND READ
1331 004504 105713      1$: TSTB $CDS ;;WAIT FOR CONTROLLER READY
1332 004506 100376      BPL 1$
1333 ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
1334 004510 016246 000002      MOV 2(ADINT),-(SP) ;;PUT NEW PS ON STACK
1335 004514 012746 004522      MOV #75$,-(SP) ;;PUT NEW PC ON STACK
1336 004520 000002      RTI ;;POP NEW PC AND PS
1337 004522      75$:
1338 004522 005013      CLR $CDS ;;DISABLE INTERRUPTS
1339 004524 012712 000232      MOV #232,$ADINT ;;CHANGE INTERRUPT RETURN ADDRESS
1340 004530 005037 000232      CLR #232 ;;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1341 004534 005737 001252      TST INTFLG ;;TEST FOR A PREVIOUS INTERRUPT
1342 004540 001441      BEQ TST12 ;;BRANCH IF NONE
1343 004542 023727 001252 100007      CMP INTFLG,#100007 ;;CHECK PREVIOUS LEVEL
1344 004550 100435      BMI TST12 ;;BRANCH IF LOWER
1345 004552 104022      ERROR +22 ;;INTERRUPT ALREADY OCCURRED AT LVL 7 OR HIGHER
1346 004554 000433      BR TST12
1347 004556 105713      TINT11: TSTB $CDS ;;MAKE SURE CONTROLLER READY IS SET
1348 004560 100401      BMI 1$ ;;BRANCH IF SET
1349 004562 104023      ERROR +23 ;;CONTROLLER READY WASN'T SET
1350 004564 005013      1$: CLR $CDS ;;DISABLE FURTHER INTERRUPTS
    
```

J08

MAINDEC - 11 - DZCOB-B MACY11 27(654) 1-JUL-77 08:39 PAGE 28
 DZCOB.P11 T11 TEST FOR AN INTERRUPT ON LEVEL 7

SEQ 0100

```

1351 004566 012712 000232      MOV      #232,  ADINT  ;CHANGE INTERRUPT RETURN ADDRESS
1352 004572 005037 000232      CLR      #232      ;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1353 004576 022626      CMP      (SP)+, (SP)+ ;RESTORE STACK POINTER
1354 004600 005737 001252      TST      INTFLC    ;CHECK FOR PREVIOUS FLAG
1355 004604 100412      BMI     SET7       ;BRANCH IF FLAG SET
1356 004606 012737 100007 001252      MOV      #100007 INTFLG ;SET FLAG AND LEVEL
1357 004614 104400 031741      TYPE,   MSG4       ;PRINT MESSAGE "THE INTERRUPT LEVEL WAS"
1358 004620 012746 000007      MOV      #7, -(SP)
1359 004624 104402      TYPOS
1360 004626      .BYTE  1
1361 004627      .BYTE  0
1362 004630 000405      BR      TST12
1363 004632 023727 001252 100007 SET7: CMP      INTFLG, #100007 ;CHECK PREVIOUS LEVEL
1364 004640 100001      BPL     TST12
1365 004642 104024      ERROR +24 ;INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1366 *****
1367 ;*TEST 12 TEST FOR AN INTERRUPT ON LEVEL 6
1368 *****
1369 004644 000004      TST12: SCOPE
1370 004646 004737 025744      JSR     PC, INIT   ;INITIALIZE
1371 004652 012712 005234      MOV      #TINT12, ADINT ;SETUP RETURN ADDRESS
1372      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1373 004656 005046      CLR      -(SP)
1374 004660 013746 000034      MOV      34, -(SP) ;SAVE CURRENT TRAP VECTOR
1375 004664 012737 004674 000034      MOV      #64$, 34 ;SETUP NEW TRAP VECTOR
1376 004672 104400      TRAP
1377 004674 016666 000002 000006 64$: MOV      2(SP), 6(SP) ;PUSH OLD PSW AN PCON STACK
1378 004702 012716 004710      MOV      #65$, (SP) ;REPLACE JLD PC WITH NEW
1379 004706 000002      RTI     ;RESTORE PSW
1380 004710 012637 000034 65$: MOV      (SP)+, 34 ;RESTORE OLD TRAP VECTOR
1381 004714 012637 001214      MOV      (SP)+, $TMP6
1382 004720 052737 000340 001214      BIS      #340, $TMP6
1383 004726 013746 001214      MOV      $TMP6, -(SP) ;PUT NEW PS ON STACK
1384 004732 012746 004740      MOV      #65$, -(SP) ;PUT NEW PC ON STACK
1385 004736 000002      RTI     ; POP NEW PC AND PS
1386 004740 66$:
1387      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1388 004740 005046      CLR      -(SP)
1389 004742 013746 000034      MOV      34, -(SP) ;SAVE CURRENT TRAP VECTOR
1390 004746 012737 004756 000034      MOV      #67$, 34 ;SETUP NEW TRAP VECTOR
1391 004754 104400      TRAP ;PUSH OLD PSW AN PCON STACK
1392 004756 016666 000002 000006 67$: MOV      2(SP), 6(SP) ;REPLACE OLD PC WITH NEW
1393 004764 012716 004772      MOV      #68$, (SP) ;RESTORE PSW
1394 004770 000002      RTI     ;RESTORE OLD TRAP VECTOR
1395 004772 012637 000034 68$: MOV      (SP)+, 34 ;RESTORE OLD TRAP VECTOR
1396 004776 012662 000002      MOV      (SP)+, 2(ADINT)
1397      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1398 005002 005046      CLR      -(SP)
1399 005004 013746 000034      MOV      34, -(SP) ;SAVE CURRENT TRAP VECTOR
1400 005010 012737 005020 00JJ34      MOV      #69$, 34 ;SETUP NEW TRAP VECTOR
1401 005016 104400      TRAP ;PUSH OLD PSW AN PCON STACK
1402 005020 016666 000002 000006 69$: MOV      2(SP), 6(SP) ;REPLACE OLD PC WITH NEW
1403 005026 012716 005034      MOV      #70$, (SP) ;RESTORE PSW
1404 005032 000002      RTI

```

```

1405 005034 012637 000034      70$:  MOV      (SP)+,34          ;;RESTORE OLD TRAP VECTOR
1406 005040 012637 001214      MOV      (SP)+,$TMP6
1407 005044 04272 000340 001214  BIC      #340,$TMP6
1408 005052 013746 001214      MOV      $TMP6,-(SP)      ;;PUT NEW PS ON STACK
1409 005056 012746 005064      MOV      #71$,-(SP)      ;;PUT NEW PC ON STACK
1410 005062 000002      RTI                          ;; POP NEW PC AND PS
1411 005064      71$:  CLR      -(SP)              ;;
1412 005064 005046      MOV      34,-(SP)         ;;SAVE CURRENT TRAP VECTOR
1413 005066 013746 000034      MOV      #72$,34         ;;SETUP NEW TRAP VECTOR
1414 005072 012737 005102 000034  TRAP                          ;;PUSH OLD FSW AN PCON STACK
1415 005100 104400      72$:  MOV      2(SP),6(SP)      ;;
1416 005102 016666 000002 000006  MOV      #73$, (SP)      ;;REPLACE OLD PC WITH NEW
1417 005110 012716 005116      RTI                          ;;RESTORE PSW
1418 005114 000002      73$:  MOV      (SP)+,34          ;;RESTORE OLD TRAP VECTOR
1419 005116 012637 000034      MOV      (SP)+,$TMP6
1420 005122 012637 001214      BIC      #240,$TMP6
1421 005126 052737 000240 001214  MOV      $TMP6,-(SP)      ;;PUT NEW PS ON STACK
1422 075134 013746 001214      MOV      #74$,-(SP)      ;;PUT NEW PC ON STACK
1423 005140 012746 005146      RTI                          ;; POP NEW PC AND PS
1424 005144 000002      74$:  MOV      #-80, @CDC      ;;SET UP COLUMN COUNT TO READ 80 COLUMNS
1425 005146      MOV      #BUFBE@, @CDA    ;;SET UP BUS ADDRESS
1426 005146 012714 177660      MOV      #101, @CDS      ;;SET INTERRUPT ENABLE AND READ
1427 005152 012715 045442      TSTB    @CDS            ;;WAIT FOR CONTROLLER READY
1428 005156 012713 000101      1$:   BPL      1$              ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
1429 005162 105713      MOV      2(ADINT),-(SP)  ;;PUT NEW PS ON STACK
1430 005164 100376      MOV      #75$,-(SP)     ;;PUT NEW PC ON STACK
1431      RTI                          ;; POP NEW PC AND PS
1432 005166 016246 000002      75$:  CLR      @CDS            ;;DISABLE INTERRUPTS
1433 005172 012746 005200      MOV      #232, @ADINT    ;;CHANGE INTERRUPT RETURN ADDRESS
1434 005176 000002      CLR      @#232          ;;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1435 005200      TST     INTFLG          ;;TEST FOR A PREVIOUS INTERRUPT
1436 005200 005013      BEQ     TST13           ;;BRANCH IF NONE
1437 005202 012712 000232      CMP     INTFLG, #100006  ;;CHECK PREVIOUS LEVEL
1438 005206 005037 000232      BMI    TST13           ;;BRANCH IF LOWER
1439 005212 005737 001252      ERROR  +22             ; INTERRUPT ALREADY OCCURRED AT LVL 6 OR HIGHER
1440 005216 001441      BR     TST13
1441 005220 023727 001252 100006  TSTB    @CDS            ;;MAKE SURE CONTROLLER READY IS SET
1442 005226 100435      BMI    1$              ;;BRANCH IF SET
1443 005230 104022      ERROR  +23             ; CONTROLLER READY WASN'T SET
1444 005232 003433      1$:   CLR      @CDS            ;;DISABLE FURTHER INTERRUPTS
1445 005234 105713      MOV      #232, @ADINT    ;;CHANGE INTERRUPT RETURN ADDRESS
1446 005236 100401      CLR      @#232          ;;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1447 005240 104023      CMP     (SP)+, (SP)+    ;;RESTORE STACK POINTER
1448 005242 005013      TST     INTFLG          ;;CHECK FOR PREVIOUS FLAG
1449 005244 012712 000232      BMI    SET6            ;;BRANCH IF FLAG SET
1450 005250 005037 000232      MOV      #100006, INTFLG ;;SET FLAG AND LEVEL
1451 005254 022626      TYPE,  MSG4            ;;PRINT MESSAGE "THE INTERRUPT LEVEL WAS"
1452 005256 005737 001252      MOV      #6,-(SP)
1453 005262 100412      TYPOS
1454 005264 012737 100006 001252  .BYTE  1
1455 005272 104400 031711
1456 005276 012746 000006
1457 005302 104402
1458 005304 001

```

```

1459 005305 000 .BYTE 0
1460 005306 000405 BR TST13
1461 005310 023727 001252 100006 SET6: CMP INTFLG, #100006 ;CHECK PREVIOUS LEVEL
1462 005316 100001 BPL TST13
1463 005320 104024 ERROR +24 ; INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1464 *****
1465 ;*TEST 13 TEST FOR AN INTERRUPT ON LEVEL 5
1466 *****
1467 005322 000004 (ST13: SCOPE
1468 005324 004737 025744 JSR PC INIT ;INITIALIZE
1469 005330 012712 005712 MOV #TINT13,ADINT ;SETUP RETURN ADDRESS
1470 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1471 005334 005046 CLR -(SP) ;;
1472 005336 013746 000034 MOV 34,-(SP) ;;SAVE CURRENT TRAP VECTOR
1473 005342 012737 005352 000034 MOV #64$,34 ;;SETUP NEW TRAP VECTOT
1474 005350 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
1475 005352 016666 000002 000006 64$: MOV 2(SP),6(SP) ;;
1476 005360 012716 005366 MOV #65$, (SP) ;;REPLACE OLD PC WITH NEW
1477 005364 000002 RTI ;;RESTORE PSW
1478 005366 012637 000034 65$: MOV (SP)+,34 ;;RESTORE OLD TRAP VECTOR
1479 005372 012637 001214 MOV (SP)+,$TMP6
1480 005376 052737 000340 001214 BIS #340,$TMP6
1481 005404 013746 001214 MOV $TMP6,-(SP) ;;PUT NEW PS ON STACK
1482 005410 012746 005416 MOV #66$,-(SP) ;;PUT NEW PC ON STACK
1483 005414 000002 RTI ;;POP NEW PC AND PS
1484 66$:
1485 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1486 005416 005046 CLR -(SP) ;;
1487 005420 013746 000034 MOV 34,-(SP) ;;SAVE CURRENT TRAP VECTOR
1488 005424 012737 005434 000034 MOV #67$,34 ;;SETUP NEW TRAP VECTOT
1489 005432 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
1490 005434 016666 000002 000006 67$: MOV 2(SP),6(SP) ;;
1491 005442 012716 005450 MOV #68$, (SP) ;;REPLACE OLD PC WITH NEW
1492 005446 000002 RTI ;;RESTORE PSW
1493 005450 012637 000034 68$: MOV (SP)+,34 ;;RESTORE OLD TRAP VECTOR
1494 005454 012662 000002 MOV (SP)+,2(ADINT)
1495 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1496 005460 005046 CLR -(SP) ;;
1497 005462 013746 000034 MOV 34,-(SP) ;;SAVE CURRENT TRAP VECTOR
1498 005466 012737 005476 000034 MOV #69$,34 ;;SETUP NEW TRAP VECTOT
1499 005474 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
1500 005476 016666 000002 000006 69$: MOV 2(SP),6(SP) ;;
1501 005504 012716 005512 MOV #70$, (SP) ;;REPLACE OLD PC WITH NEW
1502 005510 000002 RTI ;;RESTORE PSW
1503 005512 012637 000034 70$: MOV (SP)+,34 ;;RESTORE OLD TRAP VECTOR
1504 005516 012637 001214 MOV (SP)+,$TMP6
1505 005522 042737 000340 001214 BIC #340,$TMP6
1506 005530 013746 001214 MOV $TMP6,-(SP) ;;PUT NEW PS ON STACK
1507 005534 012746 005542 MOV #71$,-(SP) ;;PUT NEW PC ON STACK
1508 005540 000002 RTI ;;POP NEW PC AND PS
1509 71$:
1510 005542 005046 CLR -(SP) ;;
1511 005544 013746 000034 MOV 34,-(SP) ;;SAVE CURRENT TRAP VECTOR
1512 005550 012737 005560 000034 MOV #72$,34 ;;SETUP NEW TRAP VECTOT

```


M08

M4INDEC - 11 - 0ZCDB-B
0ZCDB P11 T13

MACY11 27(654) 1-JUL-77 08:39 PAGE 31
TEST FOR AN INTERRUPT ON LEVEL 5

SEQ 0103

```

1513 005556 104400 TRAP ;: PUSH OLD PSW AN PCON STACK
1514 005560 016666 000002 000006 72$: MOV 2(SP),5(SP) ;:
1515 005566 012716 005574 MOV #73$, (SP) ;: REPLACE OLD PC WITH NEW
1516 005572 000002 RTI ;: RESTORE PSW
1517 005574 012637 000034 73$: MOV (SP)+,34 ;: RESTORE OLD TRAP VECTOR
1518 005600 012637 001214 MOV (SP)+,$TMP6
1519 005604 052737 000200 001214 BIS #200,$TMP6
1520 005612 013746 001214 MOV $TMP6,-(SP) ;: PUT NEW PS ON STACK
1521 005616 012746 005624 MOV #174$,-(SP) ;: PUT NEW PC ON STACK
1522 005622 000002 RTI ;: POP NEW PC AND PS
1523 005624 74$:
1524 005624 012714 177660 MOV #-80,$CDC ;: SET UP COLUMN COUNT TO READ 80 COLUMNS
1525 005630 012715 045442 MOV #BUF$BEG,$CDA ;: SET UP BUS ADDRESS
1526 005634 012713 000101 MOV #101,$CDS ;: SET INTERRUPT ENABLE AND READ
1527 0056 0 105713 1$: TSTB $CDS ;: WAIT FOR CONTROLLER READY
1528 005642 100376 BPL 1$
1529 ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
1530 005644 016246 000002 MOV 2(ADINT),-(SP) ;: PUT NEW PS ON STACK
1531 005650 012746 005656 MOV #75$,-(SP) ;: PUT NEW PC ON STACK
1532 005654 000002 RTI ;: POP NEW PC AND PS
1533 005656 75$:
1534 005656 005013 CLR $CDS ;: DISABLE INTERRUPTS
1535 005660 012712 000232 MOV #232,$ADINT ;: CHANGE INTERRUPT RETURN ADDRESS
1536 005664 005037 000232 CLR #232 ;: TO CAUSE A HALT IF AN INTERRUPT OCCURS
1537 005670 005737 001252 TST INTFLG ;: TEST FOR A PREVIOUS INTERRUPT
1538 005674 001441 BEQ TST14 ;: BRANCH IF NONE
1539 005676 023727 001252 100005 CMP INTFLG,#100005 ;: CHECK PREVIOUS LEVEL
1540 005704 100435 BMI TST14 ;: BRANCH IF LOWER
1541 005706 104022 ERROR +22 ; INTERRUPT ALREADY OCCURRED AT LVL 5 OR HIGHER
1542 005710 000433 BR TST14
1543 005712 105713 TINT13: TSTB $CDS ;: MAKE SURE CONTROLLER READY IS SET
1544 005714 100401 BMI 1$ ;: BRANCH IF SET
1545 005716 104023 ERROR +23 ;: CONTROLLER READY WASN'T SET
1546 005720 005013 1$: CLR $CDS ;: DISABLE FURTHER INTERRUPTS
1547 005722 012712 000232 MOV #232,$ADINT ;: CHANGE INTERRUPT RETURN ADDRESS
1548 005726 005037 000232 CLR #232 ;: TO CAUSE A HALT IF AN INTERRUPT OCCURS
1549 005732 022626 CMP (SP)+,(SP)+ ;: RESTORE STACK POINTER
1550 005734 005737 001252 TST INTFLG ;: CHECK FOR PREVIOUS FLAG
1551 005740 100412 BMI SET5 ;: BRANCH IF FLAG SET
1552 005742 012737 100005 001252 MOV #100005,INTFLG ;: SET FLAG AND LEVEL
1553 005750 104400 031741 TYPE,MSG4 ;: PRINT MESSAGE "THE INTERRUPT LEVEL WAS"
1554 005754 012746 000005 MOV #5,-(SP)
1555 005760 104402 TYPOS
1556 005762 001 .BYTE 1
1557 005763 000 .BYTE 0
1558 005764 000405 BR TST14
1559 005766 023727 001252 100005 SET5: CMP INTFLG,#100005 ;: CHECK PREVIOUS LEVEL
1560 005774 100001 BPL TST14
1561 005776 104024 ERROR +24 ;: INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1562 ;*****
1563 ;*TEST 14 TEST FOR AN INTERRUPT ON LEVEL 4
1564 ;*****
1565 006000 000004 †TST14: SCOPE
1566 006002 004737 025744 JSP PC, INIT ;: INITIALIZE

```

```

1567 006006 012712 006370      MOV      #TINT14,ADINT  ;SETUP RETURN ADDRESS
1568 006006 012712 006370      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1569 006012 005046 000034      CLR      -(SP)          ;:
1570 006014 013746 000034      MOV      34,-(SP)       ;:SAVE CURRENT TRAP VECTOR
1571 006020 012737 006030 000034      MOV      #64$,34        ;:SETUP NEW TRAP VECTOT
1572 006026 104400 000002 000006 64$: TRAP      ;:PUSH OLD PSW AN PCON STACK
1573 006030 016666 000002 000006 64$: MOV      2(SP),6(SF)    ;:
1574 006036 012716 006044 000002 000006 64$: MOV      #65$, (SP)      ;:REPLACE OLD PC WITH NEW
1575 006042 000002 000002 000006 64$: RTI          ;:RESTORE PSW
1576 006044 012637 000034 000006 65$: MOV      (SP)+,34      ;:RESTORE OLD TRAP VECTOR
1577 006050 012637 001214 000006 65$: MOV      (SP)+,$TMP6    ;:
1578 006054 052737 000340 001214 000006 65$: BIS      #340,$TMP6    ;:
1579 006062 013746 001214 000006 65$: MOV      $TMP6,-(SP)   ;:PUT NEW PS ON STACK
1580 006066 012746 006074 000006 65$: MOV      #66$,-(SP)   ;:PUT NEW PC ON STACK
1581 006072 000002 000002 000006 65$: RTI          ;:POP NEW PC AND PS
1582 006074 000002 000002 000006 66$:
1583 006074 000002 000002 000006 66$: ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1584 006074 005046 000034      CLR      -(SP)          ;:
1585 006076 013746 000034      MOV      34,-(SP)       ;:SAVE CURRENT TRAP VECTOR
1586 006102 012737 006112 000034      MOV      #67$,34        ;:SETUP NEW TRAP VECTOT
1587 006110 104400 000002 000006 67$: TRAP      ;:PUSH OLD PSW AN PCON STACK
1588 006112 016666 000002 000006 67$: MOV      2(SP),6(SP)    ;:
1589 006120 012716 006126 000002 000006 67$: MOV      #68$, (SP)      ;:REPLACE OLD PC WITH NEW
1590 006124 000002 000002 000006 67$: RTI          ;:RESTORE PSW
1591 006126 012637 000034 000006 68$: MOV      (SP)+,34      ;:RESTORE OLD TRAP VECTOR
1592 006132 012662 000002 000006 68$: MOV      (SP)+,2(ADINT) ;:
1593 006132 012662 000002 000006 68$: ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1594 006136 005046 000034      CLR      -(SP)          ;:
1595 006140 013746 000034      MOV      34,-(SP)       ;:SAVE CURRENT TRAP VECTOR
1596 006144 012737 006154 000034      MOV      #69$,34        ;:SETUP NEW TRAP VECTOT
1597 006152 104400 000002 000006 69$: TRAP      ;:PUSH OLD PSW AN PCON STACK
1598 006154 016666 000002 000006 69$: MOV      2(SP),6(SP)    ;:
1599 006162 012716 006170 000002 000006 69$: MOV      #70$, (SP)      ;:REPLACE OLD PC WITH NEW
1600 006166 000002 000002 000006 69$: RTI          ;:RESTORE PSW
1601 006170 012637 000034 000006 70$: MOV      (SP)+,34      ;:RESTORE OLD TRAP VECTOR
1602 006174 012637 001214 000006 70$: MOV      (SP)+,$TMP6    ;:
1603 006200 042737 000340 001214 000006 70$: BIC      #340,$TMP6    ;:
1604 006206 013746 001214 000006 70$: MOV      $TMP6,-(SP)   ;:PUT NEW PS ON STACK
1605 006212 012746 006220 000006 70$: MOV      #71$,-(SP)   ;:PUT NEW PC ON STACK
1606 006216 000002 000002 000006 70$: RTI          ;:POP NEW PC AND PS
1607 006220 000002 000002 000006 71$:
1608 006220 005046 000034      CLR      -(SP)          ;:
1609 006222 013746 000034      MOV      34,-(SP)       ;:SAVE CURRENT TRAP VECTOR
1610 006226 012737 006236 000034      MOV      #72$,34        ;:SETUP NEW TRAP VECTOT
1611 006234 104400 000002 000006 72$: TRAP      ;:PUSH OLD PSW AN PCON STACK
1612 006236 016666 000002 000006 72$: MOV      2(SP),6(SP)    ;:
1613 006244 012716 006252 000002 000006 72$: MOV      #73$, (SP)      ;:REPLACE OLD PC WITH NEW
1614 006250 000002 000002 000006 72$: RTI          ;:RESTORE PSW
1615 006252 012637 000034 000006 73$: MOV      (SP)+,34      ;:RESTORE OLD TRAP VECTOR
1616 006256 012637 001214 000006 73$: MOV      (SP)+,$TMP6    ;:
1617 006262 052737 000140 001214 000006 73$: BIS      #140,$TMP6    ;:
1618 006270 013746 001214 000006 73$: MOV      $TMP6,-(SP)   ;:PUT NEW PS ON STACK
1619 006274 012746 006302 000006 73$: MOV      #74$,-(SP)   ;:PUT NEW PC ON STACK
1620 006300 000002 000002 000006 73$: RTI          ;:POP NEW PC AND PS

```

```

1621 006302          74S:  MOV     #-80, @CDC      ;SET UP COLUMN COUNT TO READ 80 COLUMNS
1622 006302 012714 177660  MOV     #BUFBEQ, @CDA    ;SET UP BUS ADDRESS
1623 006306 012715 045442  MOV     #101, @CDS      ;SET INTERRUPT ENABLE AND READ
1624 006312 012713 000101  TSTB   @CDS            ;WAIT FOR CONTROLLER READY
1625 006316 105713          1S:  BPL     1S              ;
1626 006320 100376          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
1627          ;
1628 006322 016246 000002  MOV     2(ADINT),-(SP)  ;;PUT NEW PS ON STACK
1629 006326 012746 006334  MOV     #75S,-(SP)     ;;PUT NEW PC ON STACK
1630 006332 000002          RTI                    ;; POP NEW PC AND PS
1631 006334          75S:  CLR     @CDS            ;DISABLE INTERRUPTS
1632 006334 005013          MOV     #232, @ADINT   ;CHANGE INTERRUPT RETURN ADDRESS
1633 006336 012712 000232  CLR     @#232          ;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1634 006342 005037 000232  TST     INTFLG         ;TEST FOR A PREVIOUS INTERRUPT
1635 006346 105737 001252  BEQ     TST15          ;BRANCH IF NONE
1636 006352 001433          CMP     INTFLG, #100004 ;CHECK PREVIOUS LEVEL
1637 006374 023727 001252 100004 BMI     TST15          ;BRANCH IF LOWER
1638 006372 100427          ERROR +22            ;INTERRUPT ALREADY OCCURRED AT LVL 4 OR HIGHER
1639 006364 104022          BR      TST15
1640 006366 000425          TINT14: TSTB   @CDS      ;MAKE SURE CONTROLLER READY IS SET
1641 006370 105713          BMI     1S            ;BRANCH IF SET
1642 006372 100401          ERROR +23            ;CONTROLLER READY WASN'T SET
1643 006374 104023          1S:  CLR     @CDS            ;DISABLE FURTHER INTERRUPTS
1644 006376 005013          MOV     #232, @ADINT  ;CHANGE INTERRUPT RETURN ADDRESS
1645 006400 012712 000232  CLR     @#232          ;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1646 006404 005037 000232  CMP     (SP)+, (SP)+  ;RESTORE STACK POINTER
1647 006410 022626          TST     INTFLG         ;CHECK FOR PREVIOUS FLAG
1648 006412 005737 001252  BMI     SET4           ;BRANCH IF FLAG SET
1649 006416 100404          MOV     #100004, INTFLG ;SET FLAG AND LEVEL
1650 006420 012737 100004 001252 BR      TST15
1651 006426 000405          SET4: CMP    INTFLG, #100004 ;CHECK PREVIOUS LEVEL
1652 006430 023727 001252 100004 BPL     TST15
1653 006436 100001          ERROR +24            ;INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1654 006440 104024          ;*****
1655          ;*TEST 15 TEST FOR AN INTERRUPT ON LEVEL 3
1656          ;*****
1657          ;
1658 006442 000004          TST15: SCOPE
1659 006444 004737 025744  JSR     PC, INIT      ;INITIALIZE
1660 006450 012712 007032  MOV     #TINT15, @ADINT ;SETUP RETURN ADDRESS
1661          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1662 006454 005046          CLR     -(SP)         ;;
1663 006456 013746 000034  MOV     34,-(SP)      ;;SAVE CURRENT TRAP VECTOR
1664 006462 012737 006472 000034 MOV     #64S, 34      ;;SETUP NEW TRAP VECTOR
1665 006470 104400          TRAP                    ;;PUSH OLD PSW AN PCON STACK
1666 006472 016666 000002 000006 64S:  MOV     2(SP), 6(SP)  ;;
1667 006500 012716 006506          MOV     #65S, (SP)   ;;REPLACE OLD PC WITH NEW
1668 006504 000002          RTI                    ;;RESTORE PSW
1669 006506 012637 000034 65S:  MOV     (SP)+, 34     ;;RESTORE OLD TRAP VECTOR
1670 006512 012637 001214          MOV     (SP)+, $TMP6
1671 006516 052737 000340 001214 BIS     #340, $TMP6
1672 006524 013746 001214          MOV     $TMP6, -(SP)  ;;PUT NEW PS ON STACK
1673 006530 012746 006536          MOV     #66S, -(SP)  ;;PUT NEW PC ON STACK
1674 006534 000002          RTI                    ;; POP NEW PC AND PS

```

```

1675 006536          66$:
1676
1677 006536 005046      CLR      -(SP)
1678 006540 013746 000034  MOV      34, -(SP)
1679 006544 012737 006554 000034  MOV      #67$, 34
1680 006552 104400      TRAP
1681 006554 016666 000002 000006 67$:  MOV      2(SP), 6(SP)
1682 006562 012716 006570      MOV      #68$, (SP)
1683 006566 000002      RTI
1684 006570 012637 000034 68$:  MOV      (SP)+, 34
1685 006574 012662 000002      MOV      (SP)+, 2(ADINT)
1686
1687 006600 005046      CLR      -(SP)
1688 006602 013746 000034  MOV      34, -(SP)
1689 006606 012737 006616 000034  MOV      #69$, 34
1690 006614 104400      TRAP
1691 006616 016666 000002 000006 69$:  MOV      2(SP), 6(SP)
1692 006624 012716 006632      MOV      #70$, (SP)
1693 006630 000002      RTI
1694 006632 012637 000034 70$:  MOV      (SP)+, 34
1695 006636 012637 001214      MOV      (SP)+, $TMP6
1696 006642 042737 000340 001214  BIC      #340, $TMP6
1697 006650 013746 001214      MOV      $TMP6, -(SP)
1698 006654 012746 006662      MOV      #71$, -(SP)
1699 006660 000002      RTI
1700 006662          71$:
1701 006662 005046      CLR      -(SP)
1702 006664 013746 000034  MOV      34, -(SP)
1703 006670 012737 006700 000034  MOV      #72$, 34
1704 006676 104400      TRAP
1705 006700 016666 000002 000006 72$:  MOV      2(SP), 6(SP)
1706 006706 012716 006714      MOV      #73$, (SP)
1707 006712 000002      RTI
1708 006714 012637 000034 73$:  MOV      (SP)+, 34
1709 006720 012637 001214      MOV      (SP)+, $TMP6
1710 006724 052737 000100 001214  BIS      #100, $TMP6
1711 006732 013746 001214      MOV      $TMP6, -(SP)
1712 006736 012746 006744      MOV      #74$, -(SP)
1713 006742 000002      RTI
1714 006744          74$:
1715 006744 012714 177660  MOV      #-80, @CDC
1716 006750 012715 045442  MOV      #BUFBEG, @CDA
1717 006754 012713 000101  MOV      #101, @CDS
1718 006760 105713 1$:    TSTB    @CDS
1719 006762 100376  BPL
1720
1721 006764 016246 000002      MOV
1722 006770 012746 006776      MOV      #75$, -(SP)
1723 006774 000002      RTI
1724 006776          75$:
1725 006776 005013  CLR      @CDS
1726 007000 012712 000232  MOV      #232, @ADINT
1727 007004 005037 000232  CLR      @#232
1728 007010 005737 001252  TST      INTFLG

```

; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, 2(ADINT)"

;; SAVE CURRENT TRAP VECTOR
;; SETUP NEW TRAP VECTOR
;; PUSH OLD PSW AN PCON STACK

;; REPLACE OLD PC WITH NEW
;; RESTORE PSW
;; RESTORE OLD TRAP VECTOR

; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340, PS"

;; SAVE CURRENT TRAP VECTOR
;; SETUP NEW TRAP VECTOR
;; PUSH OLD PSW AN PCON 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 PCON 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 80 COLUMNS
;; SET UP BUS ADDRESS
;; SET INTERRUPT ENABLE AND READ
;; WAIT FOR CONTROLLER READY

; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT), PS"

;; PUT NEW PS ON STACK
;; PUT NEW PC ON STACK
;; POP NEW PC AND PS

;; DISABLE INTERRUPTS
;; CHANGE INTERRUPT RETURN ADDRESS
;; TO CAUSE A HALT IF AN INTERRUPT OCCURS
;; TEST FOR A PREVIOUS INTERRUPT

```

1729 007014 001441 BEQ TST16 ;BRANCH IF NONE
1730 007016 023727 001252 100003 CMP INTFLG, #100003 ;CHECK PREVIOUS LEVEL
1731 007024 100435 BMI TS-16 ;BRANCH IF LOWER
1732 007026 104022 ERROR +22 ;INTERRUPT ALREADY OCCURRED AT LVL 3 OR HIGHER
1733 007030 000433 BR TST16
1734 007032 105713 TINT15: TSTB ACDS ;MAKE SURE CONTROLLER READY IS SET
1735 007034 100401 BMI 15 ;BRANCH IF SET
1736 007036 104023 ERROR +23 ;CONTROLLER READY WASN'T SET
1737 007040 005013 15: CLR ACDS ;DISABLE FURTHER INTERRUPTS
1738 007042 012712 000232 MOV #232, ADINT ;CHANGE INTERRUPT RETURN ADDRESS
1739 007046 005037 000232 CLR #232 ;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1740 007052 022626 CMP (SP)+, (SP)+ ;RESTORE STACK POINTER
1741 007054 005737 001252 TST INTFLG ;CHECK FOR PREVIOUS FLAG
1742 007060 100412 BMI SET3 ;BRANCH IF FLAG SET
1743 007062 012737 100003 001252 MOV #100003, INTFLG ;SET FLAG AND LEVEL
1744 007070 104400 031741 TYPE, MSG4 ;PRINT MESSAGE "THE INTERRUPT LEVEL WAS"
1745 007074 012746 000003 MOV #3, -(SP)
1746 007100 104402 TYPOS
1747 007102 001 .BYTE 1
1748 007103 000 .BYTE 0
1749 007104 000405 BR TST16
1750 007106 023727 001252 100003 SET3: CMP INTFLG, #100003 ;CHECK PREVIOUS LEVEL
1751 007114 100001 BPL TST16
1752 007116 104024 ERROR +24 ;INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1753 ;*****
1754 ;*TEST 16 TEST FOR AN INTERRUPT ON LEVEL 2
1755 ;*****
1756 007120 000004 TST16: SCOPE
1757 007122 004737 025744 J3R PC, INIT ;INITIALIZE
1758 007126 012712 007510 MOV #TINT16, ADINT ;SETUP RETURN ADDRESS
1759 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340, PS"
1760 007132 005046 CLR -(SP)
1761 007134 013746 000034 MOV 34, -(SP) ;SAVE CURRENT TRAP VECTOR
1762 007140 012737 007150 000034 MOV #64$, 34 ;SETUP NEW TRAP VECTOR
1763 007146 104400 TRAP ;PUSH OLD PSW AN PCON STACK
1764 007150 016666 000002 000006 64$: MOV 2(SP), 6(SP)
1765 007156 012716 007164 MOV #65$, (SP) ;REPLACE OLD PC WITH NEW
1766 007162 000002 RTI ;RESTORE PSW
1767 007164 012637 000034 65$: MOV (SP)+, 34 ;RESTORE OLD TRAP VECTOR
1768 007170 012637 001214 MOV (SP)+, $TMP6
1769 007174 052737 000340 001214 BIS #340, $TMP6
1770 007202 013746 001214 MOV $TMP6, -(SP) ;PUT NEW PS ON STACK
1771 007206 012746 007214 MOV #66$, -(SP) ;PUT NEW PC ON STACK
1772 007212 000002 RTI ;POP NEW PC AND PS
1773 007214 66$:
1774 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, 2(ADINT)"
1775 007214 005046 CLR -(SP)
1776 007216 013746 000034 MOV 34, -(SP) ;SAVE CURRENT TRAP VECTOR
1777 007222 012737 007232 000034 MOV #67$, 34 ;SETUP NEW TRAP VECTOR
1778 007230 104400 TRAP ;PUSH OLD PSW AN PCON STACK
1779 007232 016666 000002 000006 67$: MOV 2(SP), 6(SP)
1780 007240 012716 007246 MOV #68$, (SP) ;REPLACE OLD PC WITH NEW
1781 007244 000002 RTI ;RESTORE PSW
1782 007246 012637 000034 68$: MOV (SP)+, 34 ;RESTORE OLD TRAP VECTOR

```

```

1783 007252 012662 000002      MOV      (SP)+,2(ADINT)
1784                                ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1785 007256 005046      CLR      -(SP)
1786 007260 013746 000034      MOV      34,-(SP)
1787 007264 012737 007274 000034      MOV      #69$,34
1788 007272 104400      TRAP
1789 007274 016666 000002 000006 69$:      MOV      2(SP),6(SP)
1790 007302 012716 007310      MOV      #70$, (SP)
1791 007306 000002      RTI
1792 007310 012637 000034 70$:      MOV      (SP)+,34
1793 007314 012637 001214      MOV      (SP)+,$TMP6
1794 007320 042737 000340 001214      BIC      #340,$TMP6
1795 007326 013746 001214      MOV      $TMP6,-(SP)
1796 007332 012746 007340      MOV      #71$,-(SP)
1797 007336 000002      RTI
1798 007340 71$:
1799 007340 005046      CLR      -(SP)
1800 007342 013746 000034      MOV      34,-(SP)
1801 007346 012737 007356 000034      MOV      #72$,34
1802 007354 104400      TRAP
1803 007356 016666 000002 000006 72$:      MOV      2(SP),6(SP)
1804 007364 012716 007372      MOV      #73$, (SP)
1805 007370 000002      RTI
1806 007372 012637 000034 73$:      MOV      (SP)+,34
1807 007376 012637 001214      MOV      (SP)+,$TMP6
1808 007402 052737 000040 001214      BIS      #040,$TMP6
1809 007410 013746 001214      MOV      $TMP6,-(SP)
1810 007414 012746 007422      MOV      #74$,-(SP)
1811 007420 000002      RTI
1812 007422 74$:
1813 007422 012714 177660      MOV      #-80, @CDC
1814 007426 012715 045442      MOV      #BUFBEG,@CDA
1815 007432 012713 000101      MOV      #101, @CDS
1816 007436 105713 1$:      TSTB    @CDS
1817 007440 100376      BPL
1818                                ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
1819 007442 016246 000002      MOV      2(ADINT),-(SP)
1820 007446 012746 007454      MOV      #75$,-(SP)
1821 007452 000002      RTI
1822 007454 75$:
1823 007454 005013      CLR      @CDS
1824 007456 012712 000232      MOV      #232, @ADINT
1825 007462 005037 000232      CLR      @#232
1826 007466 005737 001252      TST     INTFLG
1827 007472 001441      BEQ     TST17
1828 007474 023727 001252 100002      CMP     INTFLG, #100002
1829 007502 100435      BMI    TST17
1830 007504 104022      ERROR +22
1831 007506 000433      BR     TST17
1832 007510 105713 TINT16: TSTB    @CDS
1833 007512 100401      BMI    1$
1834 007514 104023      ERROR +23
1835 007516 005013 1$:      CLR     @CDS
1836 007520 012712 000232      MOV     #232, @ADINT

```

```

; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
; SAVE CURRENT TRAP VECTOR
; SETUP NEW TRAP VECTOT
; PUSH OLD PSW AN PCUN 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 VECTOT
; PUSH OLD PSW AN PCUN 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 80 COLUMNS
; SET UP BUS ADDRESS
; SET INTERRUPT ENABLE AND READ
; WAIT FOR CONTROLLER READY
;
; PUT NEW PS ON STACK
; PUT NEW PC ON STACK
; POP NEW PC AND PS
; DISABLE INTERRUPTS
; CHANGE INTERRUPT RETURN ADDRESS
; TO CAUSE A HALT IF AN INTERRUPT OCCURS
; TEST FOR A PREVIOUS INTERRUPT
; BRANCH IF NONE
; CHECK PREVIOUS LEVEL
; BRANCH IF LOWER
; INTERRUPT ALREADY OCCURRED AT LVL 2 OR HIGHER
; MAKE SURE CONTROLLER READY IS SET
; BRANCH IF SET
; CONTROLLER READY WASN'T SET
; DISABLE FURTHER INTERRUPTS
; CHANGE INTERRUPT RETURN ADDRESS

```

```

1837 007524 005037 000232 CLR 2#232 ;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1838 007530 022626 CMP (SP)+ (SP)+ ;RESTORE STACK POINTER
1839 007532 005737 001252 TST INTFLG ;CHECK FOR PREVIOUS FLAG
1840 007536 100412 BMI SET2 ;BRANCH IF FLAG SET
1841 007540 012737 1000J2 001252 MOV #100002,INTFLG ;SET FLAG AND LEVEL
1842 007546 104400 031741 TYPE, MSG4 ;PRINT MESSAGE "THE INTERRUPT LEVEL WAS"
1843 007552 012746 000002 MOV #2,-(SP)
1844 007556 104402 TYPOS
1845 007560 001 .BYTE 1
1846 007561 000 .BYTE 0
1847 007562 000405 BR TST17
1848 007564 023727 001252 100002 SET2: CMP INTFLG, #100002 ;CHECK PREVIOUS LEVEL
1849 007572 100001 BPL TST17
1850 007574 104024 ERROR +24 ;INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL
1851 ;*****
1852 ;*TEST 17 TEST FOR AN INTERRUPT ON LEVEL 1
1853 ;*****
1854 007576 000004 †TST17: SCOPE
1855 007600 004737 025744 JSR PC INIT ;INITIALIZE
1856 007604 012712 010150 MOV #TINT17,ADINT ;SETUP RETURN ADDRESS
1857 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
1858 007610 005046 CLR -(SP)
1859 007612 013746 000034 MOV 34,-(SP) ;SAVE CURRENT TRAP VECTOR
1860 007616 012737 007626 000034 MOV #64$,34 ;SETUP NEW TRAP VECTOR
1861 007624 104400 TRAP ;PUSH OLD PSW AN PCON STACK
1862 007626 016666 000002 000006 64$: MOV 2(SP),6(SP)
1863 007634 012716 007642 MOV #65$, (SP) ;REPLACE OLD PC WITH NEW
1864 007640 000002 RTI ;RESTORE PSW
1865 007642 012637 000034 65$: MOV (SP)+,34 ;RESTORE OLD TRAP VECTOR
1866 007646 012637 001214 MOV (SP)+,$TMP6
1867 007652 052737 000340 001214 BIS #340,$TMP6
1868 007660 013746 001214 MOV $TMP6,-(SP) ;PUT NEW PS ON STACK
1869 007664 012746 007672 MOV #66$,-(SP) ;PUT NEW PC ON STACK
1870 007670 000002 RTI ;POP NEW PC AND PS
1871 007672 66$:
1872 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
1873 007672 005046 CLR -(SP)
1874 007674 013746 000034 MOV 34,-(SP) ;SAVE CURRENT TRAP VECTOR
1875 007700 012737 007710 000034 MOV #67$,34 ;SETUP NEW TRAP VECTOR
1876 007706 104400 TRAP ;PUSH OLD PSW AN PCON STACK
1877 007710 016666 000002 000006 67$: MOV 2(SP),6(SP)
1878 007716 012716 007724 MOV #68$, (SP) ;REPLACE OLD PC WITH NEW
1879 007722 000002 RTI ;RESTORE PSW
1880 007724 012637 000034 68$: MOV (SP)+,34 ;RESTORE OLD TRAP VECTOR
1881 007730 012662 000002 MOV (SP)+,2(ADINT)
1882 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
1883 007734 005046 CLR -(SP)
1884 007736 013746 000034 MOV 34,-(SP) ;SAVE CURRENT TRAP VECTOR
1885 007742 012737 007752 000034 MOV #69$,34 ;SETUP NEW TRAP VECTOR
1886 007750 104400 TRAP ;PUSH OLD PSW AN PCON STACK
1887 007752 016666 0J0002 000006 69$: MOV 2(SP),6(SP)
1888 007760 012716 007766 MOV #70$, (SP) ;REPLACE OLD PC WITH NEW
1889 007764 000002 RTI ;RESTORE PSW
1890 007766 012637 000034 70$: MOV (SP)+,34 ;RESTORE OLD TRAP VECTOR

```

```

1891 007772 012637 001214      MOV      (SP)+,$TMP6
1892 007776 042737 000340 001214      BIC      #340,$TMP6
1893 010004 013746 001214      MOV      $TMP6,-(SP)      ;;PUT NEW PS ON STACK
1894 010010 012746 010016      MOV      #715,-(SP)      ;;PUT NEW PC ON STACK
1895 010014 000002      RTI      ;;POP NEW PC AND PS
1896 010016      71$:
1897 010016 005046      CLR      -(SP)      ;;
1898 010020 013746 000034      MOV      34,-(SP)      ;;SAVE CURRENT TRAP VECTOR
1899 010024 012737 010034 000034      MOV      #725,34      ;;SETUP NEW TRAP VECTOT
1900 010032 104400      TRAP     ;;PUSH OLD PSW AN PCON STACK
1901 010034 016666 000002 000006 72$:      MOV      2(SP),6(SP)      ;;
1902 010042 012716 010050      MOV      #735,(SP)      ;;REPLACE OLD PC WITH NEW
1903 010046 000002      RTI      ;;RESTORE PSW
1904 010050 012637 000034 73$:      MOV      (SP)+,34      ;;RESTORE OLD TRAP VECTOR
1905 010054 012637 001214      MOV      (SP)+,$TMP6
1906 010060 052737 000000 001214      BIS      #000,$TMP6
1907 010066 013746 001214      MOV      $TMP6,-(SP)      ;;PUT NEW PS ON STACK
1908 010072 012746 010100      MOV      #745,-(SP)      ;;PUT NEW PC ON STACK
1909 010076 000002      RTI      ;;POP NEW PC AND PS
1910 010100      74$:
1911 010100 012714 177660      MOV      #-80,@CDC      ;;SET UP COLUMN COUNT TO READ 80 COLUMNS
1912 010104 012715 045442      MOV      #8UFBEG,@CDA      ;;SET UP BUS ADDRESS
1913 010110 012713 000101      MOV      #101,@CDS      ;;SET INTERRUPT ENABLE AND READ
1914 010114 105713      1$:      TSTB     @CDS      ;;WAIT FOR CONTROLLER READY
1915 010116 100376      BPL      1$
1916      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
1917 010120 016246 000002      MOV      2(ADINT),-(SP)      ;;PUT NEW PS ON STACK
1918 010124 012746 010132      MOV      #755,-(SP)      ;;PUT NEW PC ON STACK
1919 010130 000002      RTI      ;;POP NEW PC AND PS
1920 010132      75$:
1921 010132 005013      CLR      @CDS      ;;DISABLE INTERRUPTS
1922 010134 012712 000232      MOV      #232,@ADINT      ;;CHANGE INTERRUPT RETURN ADDRESS
1923 010140 005037 000232      CLR      @#232      ;;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1924 010144 104041      ERROR   +41      ;;NO INTERRUPT WITH PROCESSOR AT LEVEL 0
1925 010146 000433      BR      TST20
1926 010150 105713      TINT17: TSTB     @CDS      ;;MAKE SURE CONTROLLER READY IS SET
1927 010152 100401      BMI     1$      ;;BRANCH IF SET
1928 010154 104023      ERROR   +23      ;;CONTROLLER READY WASN'T SET
1929 010156 005013      1$:      CLR      @CDS      ;;DISABLE FURTHER INTERRUPTS
1930 010160 012712 000232      MOV      #232,@ADINT      ;;CHANGE INTERRUPT RETURN ADDRESS
1931 010164 005037 000232      CLR      @#232      ;;TO CAUSE A HALT IF AN INTERRUPT OCCURS
1932 010170 027626      CMP      (SP)+,(SP)+      ;;RESTORE STACK POINTER
1933 010172 005737 001252      TST     INTFLG      ;;CHECK FOR PREVIOUS FLAG
1934 010176 100412      BMI     SET1      ;;BRANCH IF FLAG SET
1935 010200 012737 100001 001252      MOV      #100001,INTFLG      ;;SET FLAG AND LEVEL
1936 010206 104400 031741      TYPE,   MSG4      ;;PRINT MESSAGE "THE INTERRUPT LEVEL WAS"
1937 010212 012746 000001      MOV      #1,-(SP)
1938 010216 104402      TYPOS
1939 010220      .BYTE  1
1940 010221      .BYTE  0
1941 010222 000405      BR      TST20
1942 010224 023727 001252 100001 SET1: CMP     INTFLG, #100001 ;CHECK PREVIOUS LEVEL
1943 010232 100001      BPL     TST20
1944 010234 104024      ERROR   +24      ;;INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL

```



```

1945
1946
1947
1948
1949 010236 000004
1950 010240 004737 025744
1951 010244 012712 010452
1952
1953 010250 005046
1954 010252 013746 000034
1955 010256 012737 010266 000034
1956 010264 104400
1957 010266 016666 000002 000006 64$:
1958 010274 012716 010302
1959 010300 000002
1960 010302 012637 000034 55$:
1961 010306 012637 001214
1962 010312 052737 000340 001214
1963 010320 013746 001214
1964 010324 012746 010332
1965 010330 000002
1966 010332
1967
1968 010332 005046
1969 010334 013746 000034
1970 010340 012737 010350 000034
1971 010346 104400
1972 010350 016666 000002 000006 67$:
1973 010356 012716 010364
1974 010362 000002
1975 010364 012637 000034 68$:
1976 010370 012662 000002
1977 010374 013746 000000
1978 010400 012746 010406
1979 010404 000002
1980 010406
1981 010406 012714 177777
1982 010412 012715 045442
1983 010416 012713 000100
1984 010422 005037 001250
1985 010426 005237 001250
1986 010432 001375
1987
1988 010434 016246 000002
1989 010440 012746 010446
1990 010444 000002
1991 010446
1992 010446 005013
1993 010450 000403
1994 010452 104021
1995 010454 022626
1996 010456 005013
1997 010460 005037 000232
1998 010464 012712 000232

```

```

;*****
;*TEST 20 TEST NO INTERRUPT WITH IE SET & REST CLEARED
;*****
TST20: SCOPE
JSR PC INIT ;INITIALIZE CSR TO ZERO
MOV #TINT20,ADINT ;SETUP RETURN ADDRESS
;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

66$:
;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
CLR -(SP)
MOV 34, -(SP) ;SAVE CURRENT TRAP VECTOR
MOV #67$,34 ;SETUP NEW TRAP VECTOR
TRAP ;PUSH OLD PSW AN PCON STACK
MOV 2(SP),6(SP) ;
MOV #68$, (SP) ;REPLACE OLD PC WITH NEW
RTI ;RESTORE PSW
MOV (SP)+,34 ;RESTORE OLD TRAP VECTOR
MOV (SP)+,2(ADINT)
MOV PRO,-(SP) ;PUT NEW PS ON STACK
MOV #69$,-(SP) ;PUT NEW PC ON STACK
RTI ;POP NEW PC AND PS

69$:
MOV #-1, @CDC ;SET UP COLUMN COUNT TO READ 1 COLUMN
MOV #BUFBEQ,@CDA ;SET UP BUS ADDRESS
MOV #100,@CDS ;ENABLE INTERRUPTS
CLR COUNT ;INITIALIZE COUNTER
INC COUNT ;WAIT AWHILE
BNE 1$

1$:
;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
MOV 2(ADINT),-(SP) ;PUT NEW PS ON STACK
MOV #70$,-(SP) ;PUT NEW PC ON STACK
RTI ;POP NEW PC AND PS

70$:
CLR @CDS ;DISABLE FURTHER INTERRUPTS
BR CONT20
TINT20: ERROR +21 ;AN INTERRUPT OCCURRED
CMP (SP)+, (SP)+ ;RESTORE STACK
CLR @CDS ;DISABLE FURTHER INTERRUPTS
CONT20: CLR @#232 ;CHANGE INTERRUPT RETURN ADDRESS TO
MOV #232, @ADINT ;CAUSE A HALT IF AN INTERRUPT OCCURS

```

```

1999
2000 ;*****
2001 ;*TEST 21 SIMULTANEOUS INTERRUPTS AT MORE THAN 1 LEVEL
2002 ;*****
2003 010470 000004          ;ST21: SCOPE
2004 010472 004737 025744 JSR PC INIT ;INITIALIZE CSR TO ZERO
2005 010476 012712 010726 MOV #TINT21,ADINT ;SETUP RETURN ADDRESS
2006 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
2007 010502 005046 CLR -(SP) ;;
2008 010504 013746 000034 MOV 34,-(SP) ;;SAVE CURRENT TRAP VECTOR
2009 010510 012737 010520 000034 MOV #64$,34 ;SETUP NEW TRAP VECTOT
2010 010516 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
2011 010520 016666 000002 000006 64$: MOV 2(SP),6(SP) ;;
2012 010526 012716 010534 MOV #65$, (SP) ;;REPLACE OLD PC WITH NEW
2013 010532 000002 RTI ;;RESTORE PSW
2014 010534 012637 000034 65$: MOV (SP)+,34 ;;RESTORE OLD TRAP VECTOR
2015 010540 012637 001214 MOV (SP)+,$TMP6
2016 010544 052737 000340 001214 BIS #340,$TMP6
2017 010552 013746 001214 MOV $TMP6,-(SP) ;;PUT NEW PS ON STACK
2018 010556 012746 010564 MOV #66$,-(SP) ;;PUT NEW PC ON STACK
2019 010562 000002 RTI ;;POP NEW PC AND PS
2020 010564 66$:
2021 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
2022 010564 005046 CLR -(SP) ;;
2023 010566 013746 000034 MOV 34,-(SP) ;;SAVE CURRENT TRAP VECTOR
2024 010572 012737 010602 000034 MOV #67$,34 ;SETUP NEW TRAP VECTOT
2025 010600 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
2026 010602 016666 000002 000006 67$: MOV 2(SP),6(SP) ;;
2027 010610 012716 010616 MOV #68$, (SP) ;;REPLACE OLD PC WITH NEW
2028 010614 000002 RTI ;;RESTORE PSW
2029 010616 012637 000034 68$: MOV (SP)+,34 ;;RESTORE OLD TRAP VECTOR
2030 010622 012662 000002 MOV (SP)+,2(ADINT)
2031 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
2032 010626 005046 CLR -(SP) ;;
2033 010630 013746 000034 MOV 34,-(SP) ;;SAVE CURRENT TRAP VECTOR
2034 010634 012737 010644 000034 MOV #69$,34 ;SETUP NEW TRAP VECTOT
2035 010642 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
2036 010644 016666 000002 000006 69$: MOV 2(SP),6(SP) ;;
2037 010652 012716 010660 MOV #70$, (SP) ;;REPLACE OLD PC WITH NEW
2038 010656 000002 RTI ;;RESTORE PSW
2039 010660 012637 000034 70$: MOV (SP)+,34 ;;RESTORE OLD TRAP VECTOR
2040 010664 012637 001214 MOV (SP)+,$TMP6
2041 010670 042737 000340 001214 BIC #340,$TMP6
2042 010676 013746 001214 MOV $TMP6,-(SP) ;;PUT NEW PS ON STACK
2043 010702 012746 010710 MOV #71$,-(SP) ;;PUT NEW PC ON STACK
2044 010706 000002 RTI ;;POP NEW PC AND PS
2045 010710 71$:
2046 010710 012714 177777 MOV #-1, @CDC ;SET UP COLUMN COUNT TO READ 1 COLUMN
2047 010714 012715 045442 MOV #BUFBEQ,@CDA ;SET UP BUS ADDRESS
2048 010720 012713 000101 MOV #101, @CDS ;SET INTERRUPT ENABLE AND READ
2049 010724 000777 BR ;WAIT FOR INTERRUPT
2050 010726 022626 TINT21: CMP (SP)+, (SP)+ ;RESTORE STACK POINTER
2051 010730 012712 010764 MOV #TINA21,@ADINT ;CHANGE RETRUN ADDRESS
2052 010734 013746 000000 MOV PRO,-(SP) ;;PUT NEW PS ON STACK

```

```

2053 010740 012746 010746      MOV      #64$,-(SP)      ;;PUT NEW PC ON STACK
2054 010744 000002              RTI                    ;; POP NEW PC AND PS
2055 010746                      64$:
2056 010746 000240              NOP                    ;WAIT
2057                      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT),PS"
2058 010750 016246 000002      MOV      2(ADINT),-(SP) ;;PUT NEW PS ON STACK
2059 010754 012746 010762      MOV      #65$,-(SP)    ;;PUT NEW PC ON STACK
2060 010760 000002              RTI                    ;; POP NEW PC AND PS
2061 010762                      65$:
2062 010762 000402      BR      CONT21
2063 010764 022626      TINA21: CMP      (SP)+, (SP)+ ;RESTORE STACK
2064 010766 104025      ERROR +25 ;THE INTERRUPT OCCURRED AT 2 LEVELS
2065 010770 005013      CONT21: CLR      @CDS ;DISABLE INTERRUPTS
2066 010772 005037 000232      CLR      @#232 ;CHANGE INTERRUPT RETURN ADDRESS TO
2067 010776 012712 000232      MOV      #232, @ADINT ;CAUSE A HALT IF AN INTERRUPT OCCURS
2068
2069 ;*****
2070 ;*TEST 22 NON-EXISTANT MEMORY DETECTION
2071 ;*****
2072 011002 000004      †ST22: SCOPE
2073 011004 004737 025744      JSR      PC, INIT ;INITIALIZE CSR TO ZERO
2074 011010 012712 011240      MOV      #TINT22,@ADINT ;SETUP RETURN ADDRESS
2075                      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
2076 011014 005046      CLR      -(SP)
2077 011016 013746 000034      MOV      34, -(SP) ;SAVE CURRENT TRAP VECTOR
2078 011022 012737 011032 000034      MOV      #64$,34 ;SETUP NEW TRAP VECTOT
2079 011030 104400      TRAP ;PUSH OLD PSW AN PCON STACK
2080 011032 016666 000002 000006 64$: MOV      2(SP), 6(SP)
2081 011040 012716 011046      MOV      #65$, (SP) ;
2082 011044 000002              RTI                    ;;RESTORE PSW
2083 011046 012637 000034 65$: MOV      (SP)+, 34 ;;RESTORE OLD TRAP VECTOR
2084 011052 012637 001214      MOV      (SP)+, $TMP6
2085 011056 052737 000340 001214      BIS      #340, $TMP6
2086 011064 013746 001214      MOV      $TMP6, -(SP) ;;PUT NEW PS ON STACK
2087 011070 012746 011076      MOV      #66$, -(SP) ;;PUT NEW PC ON STACK
2088 011074 000002              RTI                    ;; POP NEW PC AND PS
2089 011076                      66$:
2090                      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
2091 011076 005046      CLR      -(SP)
2092 011100 013746 000034      MOV      34, -(SP) ;;SAVE CURRENT TRAP VECTOR
2093 011104 012737 011114 000034      MOV      #67$, 34 ;;SETUP NEW TRAP VECTOT
2094 011112 104400      TRAP ;;PUSH OLC PSW AN PCON STACK
2095 011114 016666 000002 000006 67$: MOV      2(SP), 6(SP)
2096 011122 012716 011130      MOV      #68$, (SP) ;;REPLACE OLD PC WITH NEW
2097 011126 000002              RTI                    ;;RESTORE PSW
2098 011130 012637 000034 68$: MOV      (SP)+, 34 ;;RESTORE OLD TRAP VECTOR
2099 011134 012662 000002      MOV      (SP)+, 2(ADINT)
2100                      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
2101 011140 005046      CLR      -(SP)
2102 011142 013746 000034      MOV      34, -(SP) ;;SAVE CURRENT TRAP VECTOR
2103 011146 012737 011156 000034      MOV      #69$, 34 ;;SETUP NEW TRAP VECTOT
2104 011154 104400      TRAP ;;PUSH OLD PSW AN PCON STACK
2105 011156 016666 000002 000006 69$: MOV      2(SP), 6(SP)
2106 011164 012716 011172      MOV      #70$, (SP) ;;REPLACE OLD PC WITH NEW

```

```

2107 011170 000002          RTI          ;;RESTORE PSW
2108 011172 012637 000034 70$: MOV (SP)+,34          ;;RESTORE OLD TRAP VECTOR
2109 011176 012637 001214          MOV (SP)+,$TMP6
2110 011202 042737 000340 001214 BIC #340,$TMP6
2111 011210 013746 001214          MOV $TMP6,-(SP)          ;;PUT NEW PS ON STACK
2112 011214 012746 011222          MOV #71$,-(SP)          ;;PUT NEW PC ON STACK
2113 011220 000002          RTI          ;;POP NEW PC AND PS
2114 011222
2115 011222 012714 177773          MOV #-5, @CDC          ;;SET UP COLUMN COUNT TO READ 1 COLUMN
2116 011226 012715 160000          MOV #160000, @CDA          ;;SET UP BUS ADDRESS TO NON-EXISTANT MEMORY
2117 011232 012713 000161          MOV #161, @CDS          ;;SET INTERRUPT ENABLE AND READ, X MEM BITS SET
2118 011236 000777          BR          ;;WAIT FOR INTERRUPT
2119 011240 022626          TINT2$: CMP (SP)+, (SP)+          ;;RESTORE STACK
2120 011242 005037 000232          CLR @232          ;;CHANGE INTERRUPT RETURN ADDRESS TO
2121 011246 012712 000232          MOV #232, @ADINT          ;;CAUSE A HALT IF AN INTERRUPT OCCURS
2122 011252 105713          TSTB @CDS          ;;CHECK FOR CONTROLLER READY
2123 011254 100401          BMI 1$          ;;BRANCH IF SET OK
2124 011256 104023          ERROR +23          ;;CONTROLLER READY DIDN'T SET
2125
2126 011260 005713          1$: TST @CDS          ;;CHECK FOR ERROR (BIT 15)
2127 011262 100401          BMI 2$          ;;BRANCH IF SET OK
2128 011264 104026          ERROR +26          ;;ERROR BIT 15 NOT SET
2129
2130 011266 032713 001000          2$: BIT #1000, @CDS          ;;CHECK FOR NON-EXISTANT MEMORY (BIT 9)
2131 011272 001001          BNE 3$          ;;BRANCH IF SET OK
2132 011274 104027          ERROR +27          ;;BIT 9 NOT SET
2133
2134 011276 032713 000040          3$: BIT #40, @CDS          ;;CHECK FOR EXTENDED MEMORY BIT 17 SET
2135 011302 001001          BNE 4$          ;;BRANCH IF SET OK
2136 011304 104030          ERROR +30          ;;EX-MEM BIT 17 GOT CLEARED
2137
2138 011306 032713 000020          4$: BIT #20, @CDS          ;;CHECK FOR EX-MEM (BIT 4)
2139 011312 001001          BNE 5$          ;;BRANCH IF SET OK
2140 011314 104031          ERROR +31          ;;EX-MEM (BIT 4) GOT CLEARED
2141
2142 011316 032713 076417          5$: BIT #076417, @CDS          ;;CHECK FOR ANY OTHER BITS
2143 011322 001401          BEQ 6$          ;;BRANCH IF OK
2144 011324 104032          ERROR +32          ;;EXTRA ERROR BITS SET
2145
2146 011326 022715 160002          6$: CMP #160002, @CDA          ;;CHECK ADDRESS BUFFER
2147 011332 001404          BEQ 7$          ;;BRANCH IF OK
2148 011334 012737 160002 001210          MOV #160002, $TMP4          ;;CORRECT 'CDA' CONTENTS FOR ERROR REPORT
2149 011342 104033          ERROR +33          ;;BUS ADDRESS REG CHANGED
2150
2151 011344 022714 177774          7$: CMP #-4, @CDC          ;;CHECK COLUMN COUNT REG
2152 011350 001404          BEQ 10$          ;;BRANCH IF OK
2153 011352 012737 177774 001210          MOV #-4, $TMP4          ;;CORRECT 'CDC' CONTENTS FOR ERROR REPORT
2154 011360 104034          ERROR +34          ;;COLUMN COUNT REG CHANGED
2155
2156 011362          10$:
2157          ;*****
2158          ;*TEST 23 EXECUTE DATA, DATAB(Low BYTE) LOAD ON COLUMN COUNT
2159          ;*****
2160          †T23: SCOPE

```

```

2161 011364 005014          CLR      @CDC          ;CLEAR COLUMN COUNT REGISTER
2162                                     ;PRIOR TO BYTE LOADING
2163 011366 005114          COM      @CDC
2164 011370 012700 000152  MOV      #152,R0
2165 011374 110014          MOVB    R0,@CDC          ;ATTEMPT TO LOAD LOWER BYTE
2166                                     ;OF COLUMN COUNT REGISTER
2167 011376 020014          CMP      R0,@CDC          ;DID HIGH BYTE GET LOADED AS
2168                                     ;WELL ON A LOW BYTE LOADING?
2169 011400 001403          BEQ      1$              ;BRANCH IF YES
2170 011402 010037 001210  MOV      R0,$TMP4        ;STORE GOOD DATA - SHOULD BE
2171 011406 104034          ERROR +34              ;HIGH BYTE NOT LOADED ON A LOW
2172                                     ;BYTE LOAD OF COLUMN COUNT REGISTER

```

1\$:

```

;*****
;*TEST 24 EXECUTE DATI,DATOB(HIGH BYTE) LOAD ON COLUMN COUNT
;*****

```

```

†ST24: SCOPE
2178 011410 000004          CLR      @CDC          ;CLEAR COLUMN COUNT REGISTER
2179 011412 005014          ;PRIOR TO BYTE LOADING
2180                                     ;SET VALUE FOR BYTE LOAD FROM R0
2181 011414 012700 125252  MOV      #125252,R0    ;ATTEMPT TO LOAD HIGH BYTE
2182 011420 110064 000001  MOVB    R0,+1(R4)      ;OF COLUMN COUNT REGISTER
2183                                     ;DID LOW BYTE GET LOADED AS
2184 011424 005714          TST      @CDC          ;WELL ON HIGH BYTE LOADING?
2185                                     ;BRANCH IF NO
2186 011426 001403          BEQ      1$              ;STORE GOOD DATA - SHOULD BE
2187 011430 005037 001210  CLR      $TMP4        ;LOW BYTE LOADED ON A HIGH
2188 011434 104034          ERROR +34              ;BYTE LOAD OF COLUMN COUNT REGISTER

```

1\$:

```

;*****
;*TEST 25 EXECUTE DATI,DATIP ON COLUMN COUNT REGISTER
;*****

```

```

†ST25: SCOPE
2195 011436 000004          CLR      @CDC          ;CLEAR COLUMN COUNT REGISTER
2196 011440 005014          ;PRIOR TO 'DATIP' PROCESS
2197                                     ;SET A KNOWN VALUE INTO COLUMN
2198 011442 012714 100000  MOV      #100000,@CDC  ;COUNT REGISTER
2199                                     ;PERFORM DATIP, ON COLUMN COUNT
2200 011446 005414          NEG      @CDC          ;REGISTER
2201                                     ;IS CONTENTS = 100000?
2202 011450 022714 100000  CMP      #100000,@CDC  ;BRANCH IF YES
2203 011454 001404          BEQ      1$              ;STORE GOOD DATA - SHOULD BE
2204 011456 012737 100000 001210  MOV      #100000,$TMP4 ;CONTENTS OF COLUMN COUNT REGISTER
2205 011464 104034          ERROR +34              ;INCORRECT

```

1\$:

```

;*****
;*TEST 26 EXECUTE DATI,DATOB(LOW BYTE) LOAD ON BUS ADDRESS
;*****

```

```

†ST26: SCOPE
2212 011466 000004          CLR      @CDA          ;CLEAR BUS ADDRESS REGISTER
2213 011470 005015          ;PRIOR TO BYTE LOADING
2214

```

```

2215 011472 005115          COM      @CDA
2216 011474 012700 000152  MOV      #152,R0
2217 011500 110015          MOVB     R0,@CDA
2218
2219 011502 020015          CMP      R0,@CDA
2220
2221 011504 001403          BEQ      15
2222 011506 010037 001210  MOV      R0,$TMP4
2223 011512 104033          ERROR   +33
2224
2225 011514

```

; ATTEMPT TO LOAD LOWER BYTE
; OF BUS ADDRESS REGISTER
; DID HIGH BYTE GET LOADED AS
; WELL ON A LOW BYTE LOADING?
; BRANCH IF YES
; STORE GOOD DATA - SHOULD BE
; HIGH BYTE NOT LOADED ON A LOW
; BYTE LOAD OF BUS ADDRESS REGISTER

```

2226 011514
2227
2228
2229
2230 011514 007004          *TEST 27 EXECUTE DATI,DATOB(HIGH BYTE) LOAD ON BUS ADDRESS
2231 011516 005015          *****
†ST27: SCOPE
2232          CLR      @CDA
2233          MOV      #125252,R0
2234 011520 012700 125252  MOVB     R0,+1(R5)
2235 011524 110065 000001          ; CLEAR BUS ADDRESS REGISTER
2236          TST      @CDA
2237          ; PRIOR TO BYTE LOADING
2238 011530 005715          ; SET VALUE FOR BYTE LOAD FROM R0
2239 011532 001403          ; ATTEMPT TO LOAD HIGH BYTE
2240 011534 005037 001210  ; OF BUS ADDRESS REGISTER
2241 011540 104033          ; DID LOW BYTE GET LOADED AS
2242          ; WELL ON HIGH BYTE LOADING?
2243          ; BRANCH IF NO
2244          BEQ      15
2245          CLR      $TMP4
2246          ; STORE GOOD DATA - SHOULD BE
2247          ; LOW BYTE LOADED ON A HIGH
2248          ; BYTE LOAD OF BUS ADDRESS REGISTER
2249          ERROR   +33

```

```

2250 011542
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264 011542 000004          *TEST 30 EXECUTE DATI,DATIP ON BUS ADDRESS REGISTER
2265 011544 005015          *****
†ST30: SCOPE
2266          CLR      @CDA
2267          MOV      #100000,@CDA
2268          NEG      @CDA
2269          MOV      #100000,R0
2270          CMP      #100000,@CDA
2271          BEQ      15
2272          MOV      #100000,$TMP4
2273          ; CLEAR BUS ADDRESS REGISTER
2274          ; PRIOR TO 'DATIP' PROCESS
2275          ; SET A KNOWN VALUE INTO BUS
2276          ; ADDRESS REGISTER
2277          ; PERFORM DATIP, ON BUS ADDRESS
2278          ; REGISTER
2279          ; IS CONTENTS = 100000?
2280          ; BRANCH IF YES
2281          ; STORE GOOD DATA - SHOULD BE
2282          ; CONTENTS OF BUS ADDRESS REGISTER
2283          ; INCORRECT
2284          ERROR   +33

```

```

2285 011572
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304 011572 000004          *TEST 31 WORD COUNT OVERFLOW TO 2ND CARD
2305 011574 004737 025744  *****
2306 011600 012712 012146  †ST31: SCOPE
2307          JSR      PC INIT
2308          MOV      $TINT31,@DINT
2309          ; INITIALIZE
2310          ; SET UP A RETURN ADDRESS.
2311          ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
2312          CLR      -(SP)
2313          ;;

```

```

2269 011606 013746 000034      MOV      34,-(SP)      ;;SAVE CURRENT TRAP VECTOR
2270 011612 012737 011622 000034      MOV      #64$,34      ;;SETUP NEW TRAP VECTOT
2271 011620 104400      TRAP      ;;PUSH OLD PSW AN PCON STACK
2272 011622 016666 000002 000006 64$:      MOV      2(SP),6(SP)  ;;
2273 011630 012716 011636      MOV      #65$, (SP)  ;;REPLACE OLD PC WITH NEW
2274 011634 000002      RTI      ;;RESTORE PSW
2275 011636 012637 000034      MOV      (SP)+,34    ;;RESTORE OLD TRAP VECTOR
2276 011642 012637 001214      MOV      (SP)+,$TMP6
2277 011646 052737 000340 001214      BIS      #340,$TMP6
2278 011654 013746 001214      MOV      $TMP6,-(SP) ;;PUT NEW PS ON STACK
2279 011660 012746 011666      MOV      #66$,-(SP) ;;PUT NEW PC ON STACK
2280 011664 000002      RTI      ;;POP NEW PC AND PS
2281 011666      66$:
2282      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV F3,2(ADINT)"
2283      CLR      -(SP)
2284 011670 013746 000034      MOV      34,-(SP)  ;;SAVE CURRENT TRAP VECTOR
2285 011674 012737 011704 000034      MOV      #67$,34  ;;SETUP NEW TRAP VECTOT
2286 011702 104400      TRAP      ;;PUSH OLD PSW AN PCON STACK
2287 011704 016666 000002 000006 67$:      MOV      2(SP),6(SP)  ;;
2288 011712 012716 011720      MOV      #68$, (SP)  ;;REPLACE OLD PC WITH NEW
2289 011716 000002      RTI      ;;RESTORE PSW
2290 011720 012637 000034      MOV      (SP)+,34    ;;RESTORE OLD TRAP VECTOR
2291 011724 012662 000002      MOV      (SP)+,2(ADINT)
2292      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
2293      CLR      -(SP)
2294 011732 013746 000034      MOV      34,-(SP)  ;;SAVE CURRENT TRAP VECTOR
2295 011736 012737 011746 000034      MOV      #69$,34  ;;SETUP NEW TRAP VECTOT
2296 011744 104400      TRAP      ;;PUSH OLD PSW AN PCON STACK
2297 011746 016666 000002 000006 69$:      MOV      2(SP),6(SP)  ;;
2298 011754 012716 011762      MOV      #70$, (SP)  ;;REPLACE OLD PC WITH NEW
2299 011760 000002      RTI      ;;RESTORE PSW
2300 011762 012637 000034      MOV      (SP)+,34    ;;RESTORE OLD TRAP VECTOR
2301 011766 012637 001214      MOV      (SP)+,$TMP6
2302 011772 042737 000340 001214      BIC      #340,$TMP6
2303 012000 013746 001214      MOV      $TMP6,-(SP) ;;PUT NEW PS ON STACK
2304 012004 012746 012012      MOV      #71$,-(SP) ;;PUT NEW PC ON STACK
2305 012010 000002      RTI      ;;POP NEW PC AND PS
2306 012012      71$:
2307 012012 005046      CLR      -(SP)
2308 012014 013746 000034      MOV      34,-(SP)  ;;SAVE CURRENT TRAP VECTOR
2309 012020 012737 012030 000034      MOV      #72$,34  ;;SETUP NEW TRAP VECTOT
2310 012026 104400      TRAP      ;;PUSH OLD PSW AN PCON STACK
2311 012030 016666 000002 000006 72$:      MOV      2(SP),6(SP)  ;;
2312 012036 012716 012044      MOV      #73$, (SP)  ;;REPLACE OLD PC WITH NEW
2313 012042 000002      RTI      ;;RESTORE PSW
2314 012044 012637 000034      MOV      (SP)+,34    ;;RESTORE OLD TRAP VECTOR
2315 012050 012637 001214      MOV      (SP)+,$TMP6
2316 012054 052737 000000 001214      BIS      #000,$TMP6
2317 012062 013746 001214      MOV      $TMP6,-(SP) ;;PUT NEW PS ON STACK
2318 012066 012746 012074      MOV      #74$,-(SP) ;;PUT NEW PC ON STACK
2319 012072 000002      RTI      ;;POP NEW PC AND PS
2320 012074      74$:
2321 012074 012714 177657      MOV      #-81.,@CDC  ;;SET COLUMN COUNT TO READ 81. COLUMNS
2322      ;;I.E. - WRAP AROUND INTO 2ND CARD

```

```

2323
2324 012100 012715 045442
2325
2326 012104 012713 000.01
2327 012110 105713
2328 012112 100376
2329
2330 012114 016246 000002
2331 012120 012746 012126
2332 012124 000002
2333 012126
2334
2335
2336 012126 005013
2337 012130 012712 000232
2338 012134 005037 000232
2339
2340 012140 104041
2341
2342 012142 000137 012206
2343 012146 105713
2344 012150 100401
2345 012152 104023
2346 012154 005013
2347 012156 012712 000232
2348 012162 005037 000232
2349
2350 012166 022626
2351 012170 022715 045704
2352 012174 001404
2353 012176 012737 045704 001210
2354 012204 104033
2355
2356 012206
2357
2358
2359
2360
2361 012206 000004
2362 012210 004737 025744
2363 012214 012714 177777
2364 012220 012715 045443
2365
2366 012224 012737 012250 000010
2367 012232 012737 000340 000012
2368 012240 005213
2369 012242 105713
2370 012244 100376
2371 012246 000402
2372 012250 104070
2373
2374 012252 022626
2375
2376 012254 012737 000012 000010

```

```

MOV #BUFBEQ, @CDA
MOV #101, @CDS
TSTB @CDS
BPL 1$
MOV 2(ADINT), -(SP)
MOV #75$, -(SP)
RTI
CLR @CDS
MOV #232, @ADINT
CLR @#232
ERROR +41
TINT31: JMP T31END
TSTB @CDS
BMI 1$
ERROR +23
1$: CLR @CDS
MOV #232, @ADINT
CLR @#232
CMP (SP)+, (SP)+
CMP #BUFBEQ+242, @CDA
BEQ T31END
MOV #BUFBEQ+242, $TMP4
ERROR +33
T31END:

```

```

; AUTOMATICALLY
; SET UP STARTING BUFFER ADDRESS
; FOR DUMP OF CARD/S CONTENTS
; SET INTERRUPT ENABLE & READ
; WE COME HERE AWAITING THE INTERRUPT
; AWAIT INTERRUPT
; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV 2(ADINT), PS"
; PUT NEW PS ON STACK
; PUT NEW PC ON STACK
; POP NEW PC AND PS
; AND GIVE CONTROL BACK TO THE
; PROCESSOR
; DISABLE INTERRUPTS
; RESTORE INTERRUPT RETURN ADDRESS
; TO 'HALT' IF AN UNEXPECTED INTERRUPT
; OCCURS
; NO INTERRUPT WITH PROCESSOR AT
; LEVEL 0
; GO TO NEXT TEST
; IS CONTROLLER READY SET?
; BRANCH IF YES
; INDICATE CONTROLLER READY NOT SET
; DISABLE FURTHER INTERRUPTS
; RESTORE INTERRUPT RETURN ADDRESS
; TO 'HALT' IF AN UNEXPECTED INTERRUPT
; OCCURS
; RESET STACK FROM THE INTERRUPT
; WERE 2 CARDS READ??
; BRANCH IF APPARENTLY SO
; STORE GOOD DATA - SHOULD BE
; BUS ADDRESS DOESN'T CORRELATE
; TO HAVING READ 2 CARDS
; *****
; *TEST 32 BUS ADDRESS ODD & TRANSFER IN NON-PACK MODE
; *****
TST32: SCOPE
JSR PC, INIT
MOV #-1, @CDC
MOV #BUFBEQ+1, @CDA
MOV #2$, RESVEC
MOV #340, RESVEC+2
INC @CDS
1$: TSTB @CDS
BPL 1$
BR 3$
2$: ERROR +70
CMP (SP)+, (SP)+
3$: MOV #12, 10
; INITIALIZE
; SET COLUMN COUNT TO READ 1 COLUMN
; SET BUFFER ADDRESS, FOR COLUMN DUMP,
; TO HIGH BYTE OF WORD
; SET RETURN FOR ILLEGAL INSTRUCTION
; SET PS FOR ILLEGAL INSTRUCTION
; READ A CARD
; WAIT FOR CONTROLLER READY
; WAIT TILL DONE!
; GO TO NEXT TEST
; ODD BUS ADDRESS CAUSED A TRAP IN
; NON-PACK MODE
; RESET STACK FROM ILLEGAL INSTRUCTION
; TRAP
; RESTORE TRAPCATCHER AREA FOR

```



```

2431 012446 001366          BNE      NOTRP1      ;BRANCH IF SET
2432 012450 013746 000360    MOV      TRACE,-(SP) ;PUT NEW PS ON STACK
2433 012454 012746 012462    MOV      #64$,-(SP) ;PUT NEW PC ON STACK
2434 012460 000002          RTI                    ;POP NEW PC AND PS
2435 012462
2436 012462 004737 025744    64$: DCNT1: JSR      PC,   INIT ;INITIALIZE CARD READER STATUS REGISTER
2437
2438          ;SET UP INTERRUPT SERVICING, AND START READING
2439 012466 012712 012730    MOV      #SRVC, ADINT ;SETUP RETURN POINTER
2440          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
2441 012472 005046          CLR      -(SP)        ;
2442 012474 013746 000034    MOV      34,-(SP)     ;SAVE CURRENT TRAP VECTOR
2443 012500 012737 012510 000034    MOV      #64$,34     ;SETUP NEW TRAP VECTOT
2444 012506 104400          TRAP                    ;PUSH OLD PSW AN PCON STACK
2445 012510 016666 000002 000006 64$: MOV      2(SP), 6(SP) ;
2446 012516 012716 012524    MOV      #65$, (SP)  ;
2447 012522 000002          RTI                    ;RESTORE PSW
2448 012524 012637 000034    65$: MOV      (SP)+, 34 ;RESTORE OLD TRAP VECTOR
2449 012530 012637 001214    MOV      (SP)+, $TMP6
2450 012534 042737 000340 001214    BIC      #340, $TMP6
2451 012542 013746 001214    MOV      $TMP6,-(SP) ;PUT NEW PS ON STACK
2452 012546 012746 012554    MOV      #66$,-(SP) ;PUT NEW PC ON STACK
2453 012552 000002          RTI                    ;POP NEW PC AND PS
2454 012554
2455          66$: ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
2456 012554 005046          CLR      -(SP)        ;
2457 012556 013746 000034    MOV      34,-(SP)     ;SAVE CURRENT TRAP VECTOR
2458 012562 012737 012572 000034    MOV      #67$,34     ;SETUP NEW TRAP VECTOT
2459 012570 104400          TRAP                    ;PUSH OLD PSW AN PCON STACK
2460 012572 016666 000002 000006 67$: MOV      2(SP), 6(SP) ;
2461 012600 012716 012606    MOV      #68$, (SP)  ;
2462 012604 000002          RTI                    ;RESTORE PSW
2463 012606 012637 000034    68$: MOV      (SP)+, 34 ;RESTORE OLD TRAP VECTOR
2464 012612 012662 000002    MOV      (SP)+, 2(ADINT)
2465 012616 013701 015326    MOV      TSTART, R1   ;SET UP TABLE POINTER
2466 012622 012700 045442    MOV      #BUFBEQ, RO  ;SET UP BUFFER POINTER
2467 012626 012737 177660 015314    MOV      #-120, SIZE ;SET UP "SIZE"
2468 012634 012737 177660 015316    MOV      #-120, OFFSET
2469 012642 013714 015314    MOV      SIZE, @CDC   ;SET UP COLUMN COUNT
2470 012646 010015          MOV      RO, @CDA    ;SET UP ADDRESS REG
2471 012650 012713 000100    MOV      #100, @CDS  ;ENABLE INTERRUPTS
2472 012654 032777 000010 166254    BIT      #10, @SWR   ;CHECK FOR PACK MODE ONLY
2473 012662 001406          BEQ      CDREAD     ;BRANCH IF NOT SET
2474 012664 032777 000004 166244    BIT      #4, @SWR   ;CHECK FOR IMAGE MODE ONLY
2475 012672 001002          BNE      CDREAD     ;BRANCH IF SET
2476 012674 004737 014522    JSR      PC,   PAKSET ;SET UP FOR PACKING MODE
2477 012700 005213          CDREAD: INC @CDS   ;RFAD
2478 012702 032713 004000    BKGND: BIT      #4000, @CDS ;CHECK FOR DATA ERROR
2479 012706 001775          BEQ      BKGND
2480 012710 011437 015340    BKGND1: MOV      @CDC, DERCNT ;SAVE THE COLUMN COUNT
2481 012714 032713 004000    BIT      #4000, @CDS ;CHECK FOR DATA ERROR
2482 012720 001375          BNE      BKGND1    ;BRANCH IF SET
2483 012722 005037 015340    CLR      DERCNT    ;CLR COLUMN COUNT SAVER
2484 012726 000765          BR

```

```

2485 ; INTERRUPT SERVICE ROUTINE WHICH RUNS DATA RELIABILITY TEST
2486 SRVC: TSTB  @CDS ; CHECK CONTROLLER READY
2487 012730 105713 BMI 1$ ; BRANCH IF SET
2488 012732 100401 ERROR +23 ; CONTROLLER READY NOT SET
2489 012734 104023 1$: BIT #2 @CDS ; CHECK FOR DATA PACK MODE
2490 012736 032713 000002 BEQ ISR ; BRANCH IF IMAGE MODE
2491 012742 001402 JMP PSR ; JUMP TO PACK MODE ROUTINE
2492 012744 000137 013640
2493
2494 012750 032713 177477 ISR: BIT #177477, @CDS ; CHECK ALL BITS EXCEPT 6 AND 7
2495 012754 001001 BNE 1$ ; BRANCH TO ERROR ROUTINE
2496 012756 000402 BR 2$ ; OTHERWISE, CONTINUE ON
2497 012760 000137 013510 1$: JMP ISRR ; GO TO ERROR ROUTINE
2498 012764 005714 2$: TST @CDC ; CHECK COLUMN COUNT
2499 012766 001403 BEQ 3$ ; BRANCH IF OK
2500 012770 005037 001210 CLR $TMP4 ; CORRECT 'CDC' CONTENTS FOR ERROR REPORT
2501 012774 104034 ERROR +34 ; COLUMN COUNT REGISTER NOT 0
2502
2503 012776 010037 015320 3$: MOV R0, BUFEND
2504 013002 163737 015314 015320 SUB SIZE, BUFEND
2505 013010 163737 015314 015320 SUB SIZE, BUFEND
2506 013016 023715 015320 CMP BUFEND, @CDA
2507 013022 001404 BEQ ISRNC
2508 013024 013737 015320 001210 MOV BUFEND, $TMP4 ; CORRECT 'CDA' CONTENTS FOR ERROR REPORT
2509 013032 104033 ERROR +33
2510
2511 013034 013737 015314 001250 ISRNC: MOV SIZE, COUNT ; SET UP COLUMN COUNTER
2512 013042 005777 166172 ISRLP: TST @CDD8 ; ARE WE ON A CARD READER WITH
2513 ; ECO #14 INSTALLED?
2514 013046 100411 BMI 1$ ; BRANCH IF YES
2515 013050 012137 001200 MOV (R1)+, $TMP0 ; GET THE TABLE VALUE
2516 013054 042737 170000 001200 BIC #170000, $TMP0 ; STRIP OFF TOP 4 BITS BEFORE COMPARISON
2517 013062 023720 001200 CMP $TMP0, (R0)+ ; IS VALUE SAME AS DUMPED INTO MEMORY?
2518 013066 001043 BNE ISRDE ; BRANCH IF NOT THE SAME
2519 013070 000402 BR ISRRT ; OTHERWISE, CONTINUE ON
2520 013072 022021 1$: CMP (R0)+, (R1)+ ; TEST THE DATA
2521 013074 001040 BNE ISRDE ; BRANCH IF DATA ERROR
2522 013076 020137 015330 ISRRT: CMP R1, TEND ; CHECK FOR END OF TABLE
2523 013102 100402 BMI 1$ ; BRANCH IF NO
2524 013104 013701 015326 MOV TSTART, R1 ; MOVE POINTER TO TOP OF TABLE
2525 013110 005237 001250 1$: INC COUNT ; CHECK FOR END OF BUFFER
2526 013114 001415 BEQ ISRBE ; BRANCH IF BUFFER END
2527 013116 005237 015334 INC CLCNT ; KEEP TRACK OF COLUMNS
2528 013122 062737 000002 001212 ADD #2, $TMP5 ; UPDATE TABLE OFFSET FOR CARD #1 OF DECK
2529 013130 023727 015334 000120 CMP CLCNT, #120 ; CHECK FOR END OF CARD
2530 013136 001341 BNE ISRLP ; BRANCH IF NOT END OF CARD
2531 013140 004737 014710 JSR PC, NXCRD ; INC TO NEXT CARD
2532 013144 005721 TST (R1)+ ; UPDATE TABLE POINTER FOR NEXT CARD
2533 013146 000735 BR ISRLP
2534
2535 013150 004737 014710 ISRBE: JSR PC, NXCRD ; GO TO NEXT CARD
2536 013154 005721 ISRNX: TST (R1)+
2537 013156 032777 000004 165752 BIT #4, @SWR ; CHECK FOR IMAGE MODE ONLY
2538 013164 001002 BNE ISRNX1 ; BRANCH IF SET

```

F10

MAINDEC - 11 - UZCDB-B MACY11 27(654) 1-JUL-77 08:39 PAGE 50
UZCDB.P11 DATA RELIABILITY TEST

SEQ 0.22

| Address | PC | PAKSET | Comments |
|---------|--------|--------|----------------------------------------------------------------------------|
| 2539 | 013166 | 004737 | 014522 JSR PC, PAKSET ; SET UP FOR PACKING MODE |
| 2540 | 013172 | 000137 | 014436 JSRNX1: JMP SRRETRN ; CALCULATE "SIZE" AND RETURN |
| 2541 | | | |
| 2542 | | | : DATA ERROR WAS DETECTED, OUTPUT ERROR PRINTOUT |
| 2543 | 013176 | 005737 | 015332 JSRDE: TST CDCNT ; CHECK FOR FIRST CARD |
| 2544 | 013202 | 001102 | BNE ISR02 ; BRANCH IF NOT |
| 2545 | 013204 | 005740 | ISR01: TST -(R0) ; SUB 2 FROM POINTER |
| 2546 | 013206 | 005237 | 015332 INC CDCNT |
| 2547 | 013212 | 005777 | 166022 TST @C0DB ; ARE WE ON A CARD READER WITH |
| 2548 | | | ECO #14 INSTALLED? |
| 2549 | 013216 | 100411 | BMI 1\$; BRANCH IF YES |
| 2550 | 013220 | 012137 | 001200 MOV (R1)+, \$TMP0 ; GET THE TABLE VALUE |
| 2551 | 013224 | 042737 | 170000 001200 BIC #170000, \$TMP0 ; STRIP OFF TOP 4 BITS BEFORE COMPARISON |
| 2552 | 013232 | 023720 | 001200 CMP \$TMP0, (R0)+ ; DOES VALUE MATCH THAT DUMPED INTO MEMORY? |
| 2553 | 013236 | 001051 | BNE 6\$; BRANCH IF NO |
| 2554 | 013240 | 000402 | BR 2\$; OTHERWISE, CONTINUE ON |
| 2555 | 013242 | 022021 | 1\$: CMP (R0)+, (R1)+ ; TEST THE DATA |
| 2556 | 013244 | 001046 | BNE 6\$; BRANCH IF NOT THE SAME |
| 2557 | 013246 | 062701 | 000042 2\$: ADD #42, R1 ; ADD THE MAGIC NUMBER |
| 2558 | 013252 | 020137 | 015330 CMP R1, TEND ; CHECK FOR RAP AROUND |
| 2559 | 013256 | 003402 | BLE 3\$; BRANCH IF NOT |
| 2560 | 013260 | 162701 | 000240 SUB #240, R1 ; RAP AROUND |
| 2561 | 013264 | 005777 | 165750 3\$: TST @C0DB ; ARE WE ON A CARD READER WITH |
| 2562 | | | ECO #14 INSTALLED? |
| 2563 | 013270 | 100412 | BMI 7\$; BRANCH IF YES |
| 2564 | 013272 | 011137 | 001200 MOV (R1), \$TMP0 ; GET THE TABLE VALUE |
| 2565 | 013276 | 042737 | 170000 001200 BIC #170000, \$TMP0 ; STRIP OFF TOP 4 BITS BEFORE COMPARISON |
| 2566 | 013304 | 026037 | 000042 001200 CMP 42(R0), \$TMP0 ; DOES VALUE MATCH FOR A DCUBLE? |
| 2567 | 013312 | 001014 | BNE 5\$; BRANCH IF NO |
| 2568 | 013314 | 000403 | BR 4\$; OTHERWISE, CONTINUE ON |
| 2569 | 013316 | 026011 | 000042 7\$: CMP 42(R0), (R1) ; CHECK FOR DOUBLE MATCH |
| 2570 | 013322 | 001010 | BNE 5\$; BRANCH IF NOT |
| 2571 | 013324 | 162701 | 000042 4\$: SUB #42, R1 ; SUBTRACT THE MAGIC NUMBER |
| 2572 | 013330 | 020137 | 015326 CMP R1, TSTART ; CHECK FOR RAP AROUND |
| 2573 | 013334 | 003260 | BGT ISRRT ; BRANCH IF NOT |
| 2574 | 013336 | 062701 | 000240 ADD #240, R1 ; RAP AROUND |
| 2575 | 013342 | 000655 | BR ISRRT ; GO CHECK REST OF DATA |
| 2576 | | | |
| 2577 | 013344 | 162701 | 000042 5\$: SUB #42, R1 ; SUBTRACT MAGIC NUMBER |
| 2578 | 013350 | 020137 | 015326 CMP R1, TSTART ; CHECK FOR RAP AROUND |
| 2579 | 013354 | 003002 | BGT 6\$; BRANCH IF NOT |
| 2580 | 013356 | 062701 | 000240 ADD #240, R1 ; RAP AROUND |
| 2581 | 013362 | 020137 | 015330 6\$: CMP R1, TEND |
| 2582 | 013366 | 001306 | BNE ISR01 |
| 2583 | 013370 | 013701 | 015326 MOV TSTART, R1 ; OBTAIN THE 1ST TABLE ENTRY |
| 2584 | 013374 | 063701 | 001212 ADD \$TMP5, R1 ; ADD TABLE OFFSET FOR CARD #1 OF DECK |
| 2585 | 013400 | 062701 | 000002 ADD #2, R1 ; GO AHEAD ONE TABLE POSITION FOR ERROR REPORT |
| 2586 | 013404 | 005037 | 015332 CLR CDCNT ; RESET CARD COUNTER |
| 2587 | 013410 | 032777 | 020000 165520 ISR02: BIT #20000, \$SWR ; CK SW13 FOR INHIBIT PRINTOUT |
| 2588 | 013416 | 001026 | BNE ISRDE4 ; BRANCH IF SET |
| 2589 | 013420 | 004737 | 014554 JSR PC, TYHEAD ; TYPE HEADING, DECK, CDCNT, CLCNT |
| 2590 | 013424 | 005777 | 165610 TST @C0DB ; ARE WE ON A CARD READER WITH |
| 2591 | | | ECO #14 INSTALLED? |
| 2592 | 013430 | 100410 | BMI 1\$; BRANCH IF YES |

```

2593 013432 014137 001200      MOV      -(R1),STMPD      ;GET THE TABLE VALUE
2594 013436 042737 170000 001200  BIC      #170000,STMPD    ;STRIP OFF TOP 4 BITS BEFORE PRINTOUT
2595 013444 013746 001200      MOV      STMPD,-(SP)      ;PLACE VALUE ON STACK FOR PRINTOUT
2596 013450 000401                BR       2$              ;GO TO PRINT OUT VALUE
2597 013452 014146                1$:     MOV      -(R1),-(SP) ;PUSH SHOULD BE DATA ONTO STACK
2598 013454 104402                2$:     TYPOS                      ;SYSMAC ROUTINE
2599 013456 006                          .BYTE   6                ;FOR
2600 013457 001                          .BYTE   1                ;ERROR TYPEOUT
2601 013460 104400 030545      TYPE,   SPACE            ;PUSH WAS DATA
2602 013464 014046                MOV      -(R0),-(SP)    ;ONTO STACK
2603 013466 104402                TYPOS                      ;FOR
2604 013470 006                          .BYTE   6                ;ERROR TYPEOUT
2605 013471 001                          .BYTE   1                ;FOR
2606 013472 022021                CMP      (R0)+, (R1)+    ;RESET POINTERS
2607 013474 005777 165436      ISRDE4: TST      #SWR      ;CHECK FOR HALT ON ERROR
2608 013500 100001                BPL     1$              ;BRANCH IF HALT ON ERROR NOT SET
2609 013502 000000                HALT                               ;HALT ON ERROR SET
2610 013504 000137 013076      1$:     JMP      ISRRT
2611
2612                ; INTERRUPT DUE TO SOME KIND OF ERROR
2613                ; THESE ERRORS ARE DISASTEROUS, THEREFORE THE DATA TEST IS RESTARTED
2614 013510 100402      ISRER:  BMI     ISRE1      ;BRANCH ON ERROR BIT 15
2615 013512 104026                ERROR +26                ;ERROR BIT 15 NOT SET
2616 013514 000447                BR       ISRST
2617
2618 013516 032713 010000      ISRE1:  BIT     #10000, @CDS ;CHECK FOR OFF-LINE
2619 013522 001414                BEQ     ISRE2
2620 013524 032713 040000      BIT     #40000, @CDS    ;CHECK FOR CARD READER ERROR
2621 013530 001002                BNE     1$              ;BRANCH IF SET
2622 013532 104035                ERROR +35                ;OFF-LINE BUT NOT CARD READER ERROR
2623 013534 000413                BR       ISRE3
2624
2625 013536 004737 014650      1$:     JSR     PC, LASTCD ;CHECK FOR LAST CARD
2626 013542 002002                BGE     2$              ;BRANCH IF BOTH CARD
2627 013544 104057                ERROR +57                ;CARD READER ERROR BUT NOT BOTH CARD
2628 013546 000432                BR       ISRST
2629 013550 000137 013034      2$:     JMP     ISRNC      ;IF BOTH CARD - GO HERE!
2630
2631 013554 032713 040000      ISRE2:  BIT     #40000, @CDS ;CHECK FOR CARD READER ERROR
2632 013560 001401                BEQ     ISRE3            ;BRANCH IF NOT
2633 013562 104060                ERROR +60                ;CARD READER ERROR BUT NOT OFF LINE
2634
2635 013564 032713 020000      ISRE3:  BIT     #20000, @CDS
2636 013570 001401                BEQ     1$
2637 013572 104061                ERROR +61                ;END OF FILE ERROR (M1200 ONLY)
2638
2639 013574 032713 004000      1$:     BIT     #4000, @CDS
2640 013600 001401                BEQ     2$
2641 013602 104062                ERROR +62                ;DATA ERROR
2642
2643 013604 032713 002000      2$:     BIT     #2000, @CDS
2644 013610 001401                BEQ     3$
2645 013612 104063                ERROR +63                ;DATA LATE ERROR
2646

```

```

2647 013614 032713 001000 3S: BIT #1000, @CDS
2648 013620 001401 BEQ 4S
2649 013622 104064 ERROR +64 ;NON-EXISTANT MEMORY ERROR
2650 013624 032713 077000 4S: BIT #077000, @CDS ;CHECK ALL ERROR BITS
2651 013630 001001 BNE ISRST ;BRANCH IF AT LEAST ONE
2652 013632 104037 ERROR +37 ;NONE OF THE ERROR BITS SET
2653 013634 000137 015306 ISRST: JMP DATRST ;RESTART THE ENTIRE DATA TEST
2654
2655 013640 032713 177475 PSR: BIT #177475, @CDS ;CHECK ALL BITS EXCEPT 1,6 AND 7
2656 013644 001402 BEQ 1S ;BRANCH IF OK:
2657 013646 000137 014256 JMP PSRER ;OTHERWISE, GO TO REPORT ERROR
2658 013652 005714 1S: TST @CDC ;CHECK COLUMN COUNT REG.
2659 013654 001403 BEQ 2S ;BRANCH IF OK
2660 013656 005037 071210 CLR $TMP4 ;CORRECT 'CDC' CONTENTS FOR ERROR REPORT
2661 013662 104034 ERROR +34 ;
2662 013664 010037 015320 2S: MOV RO, BUFEND
2663 013670 163737 015314 015320 SUB SIZE, BUFEND
2664 013676 023715 015320 CMP BUFEND, @CDA
2665 013702 001404 BEQ PSRNC
2666 013704 013737 015320 001210 MOV BUFEND, $TMP4 ;CORRECT 'CDA' CONTENTS FOR ERROR REPORT
2667 013712 104033 ERROR +33
2668 013714 013737 015314 001250 PSRNC: MOV SIZE, COUNT ;SET UP COLUMN COUNTER
2669 013722 122021 PSRLP: CMPB (RO)+, (R1)+ ;TEST THE DATA
2670 013724 001052 BNE PSRDE ;BRANCH IF DATA ERROR
2671 013726 020137 015330 PSRRT: CMP R1, TEND ;CHECK FOR END OF TABLE
2672 013732 100402 BMI 1S ;BRANCH IF NOT
2673 013734 013701 015326 MOV TSTART, R1 ;MOVE POINTER TO TOP OF TABLE
2674 013740 005237 001250 1S: INC COUNT ;CHECK FOR END OF BUFFER
2675 013744 001415 BEQ PSRBE ;BRANCH IF BUFFER END
2676 013746 005237 015334 INC CLCNT ;KEEP TRACK OF COLUMNS
2677 013752 062737 000001 001210 ADD #1, $TMP4 ;UPDATE TABLE OFFSET FOR CARD #1 OF DECK
2678 013760 023727 015334 000120 CMP CLCNT, #120 ;CHECK FOR END OF CARD
2679 013766 001355 BNE PSRLP ;BRANCH IF NOT END OF CARD
2680 013770 004737 014710 JSR PC, NXCRD ;GO TO NEXT CARD
2681 013774 105721 TSTB (R1)+ ;UPDATE TABLE POINTER FOR NEXT CARD
2682 013776 000751 BR PSRLP
2683
2684 014000 004737 014710 PSRBE: JSR PC, NXCRD ;GO TO NEXT CARD
2685 014004 105721 PSRNX: TSTB (R1)+
2686 014006 032777 000010 165122 BIT #10, @SWR
2687 014014 001014 BNE PSRNX1
2688 014016 162737 000240 015326 SUB #160., TSTART ;MOVE TABLE POINTER TO IMAGE TABLE
2689 014024 162737 000120 015330 SUB #80., TEND
2690 014032 162701 000240 SUB #160., R1 ;UPDATE TABLE POINTER
2691 014036 063701 015332 ADD CDCNT, R1 ;COMPENSATE FOR BYTES
2692 014042 042713 000002 BIC #2, @CDS ;CLR PACKING MODE BIT
2693 014046 000137 014436 PSRNX1: JMP SRTRN ;CALCULATE "SIZE" AND READ MORE CARDS
2694
2695 ;DATA ERROR WAS DETECTED. OUTPUT ERROR PRINTOUT
2696 014052 005737 015332 PSRDE: TST CDCNT
2697 014056 001051 BNE PSRD2
2698 014060 105740 PSRD1: TSTB -(RO) ;SUB 1 FROM POINTER
2699 014062 005237 015332 INC CDCNT
2700 014066 122021 CMPB (RO)+, (R1)+ ;TEST THE DATA

```

```

2701 014070 001031 BNE 1$ ; BRANCH IF NOT THE SAME
2702 014072 062701 000021 ADD #21, R1 ; ADD THE MAGIC NUMBER
2703 014076 020137 015330 CMP R1, TEND ; CHECK FOR RAP AROUND
2704 014102 003402 BLE 2$ ; BRANCH IF NOT
2705 014104 162701 000120 SUB #120, R1 ; RAP AROUND
2706 014110 126011 000021 2$: CMPB 21(R0), (R1) ; CHECK FOR DOUBLE MATCH
2707 014114 001010 BNE 3$ ; BRANCH IF NOT
2708 014116 162701 000021 SUB #21, R1 ; SUBTRACT THE MAGIC NUMBER
2709 014122 020137 015326 CMP R1, TSTART ; CHECK FOR RAP AROUND
2710 014126 003277 BGT PSRR1 ; BRANCH IF NOT
2711 014130 062701 000120 ADD #120, R1 ; RAP AROUND
2712 014134 000674 BR PSRR1 ; GO CHECK REST OF DATA
2713
2714 014136 162701 000021 3$: SUB #21, R1 ; SUBTRACT MAGIC NUMBER
2715 014142 020137 015326 CMP R1, TSTART ; CHECK FOR RAP AROUND
2716 014146 003002 BGT 1$ ; BRANCH IF NOT
2717 014150 062701 000120 ADD #120, R1 ; RAP AROUND
2718 014154 020137 015330 1$: CMP R1, TEND
2719 014160 001337 BNE PSRD1
2720 014162 013701 015326 MOV TSTART,R1 ; OBTAIN THE 1ST TABLE ENTRY
2721 014166 063701 00.212 ADD $TMP5,R1 ; ADD TABLE OFFSET FOR CARD #1 OF DECK
2722 014172 062701 000001 ADD #1,R1 ; GO AHEAD ONE TABLE POSITION FOR ERROR REPORT
2723 014176 005037 015332 CLR CDCNT ; RESET CARD COUNTER
2724 014202 032777 020000 164726 PSRD2: BIT #20000, $SWR ; CK SW13 FOR INHIBIT PRINTOUT
2725 014210 001015 BNE PSRDE3 ; BRANCH IF SET
2726 014212 004737 014554 JSR PC, TYHEAD ; TYPE HEADING, DECK, CDCNT, CLCNT
2727 014216 104400 030545 TYPE, SPACE
2728 014222 114146 MOVB -(R1), -(SP) ; PUSH SHOULD BE DATA
2729 014224 104402 TYPOS ; ONTO STACK
2730 014226 003 .BYTE 3 ; FOR
2731 014227 000 .BYTE 0 ; ERROR TYPEOUT
2732 014230 104400 030542 TYPE, SPACE-3
2733 014234 114046 MOVB -(R0), -(SP) ; PUSH WAS DATA
2734 014236 104402 TYPOS ; ONTO STACK
2735 014240 003 .BYTE 3 ; FOR
2736 014241 000 .BYTE 0 ; ERROR TYPEOUT
2737 014242 122021 CMPB (R0)+, (R1)+ ; RESET POINTERS
2738 014244 005777 164666 PSRDE3: TST $SWR ; CHECK FOR HALT ON ERROR
2739 014250 100001 BPL 1$ ; BRANCH IF HALT ON ERROR NOT SET
2740 014252 000000 HALT ; HALT ON ERROR SET
2741 014254 000624 1$: BR PSRR1
2742
2743 ; INTERRUPT DUE TO SOME KIND OF ERROR
2744 014256 100402 PSRR1: BMI PSRE1 ; BRANCH ON ERROR BIT 15
2745 014260 104026 ERROR +26 ; ERROR BIT 15 NOT SET
2746 014262 000463 BR PSRST
2747
2748 014264 032713 004000 PSRE1: BIT #4000, $CDS
2749 014270 001414 BEQ PSRE2 ; BRANCH IF NOT
2750 014272 032713 000002 BIT #2, $CDS
2751 014276 001001 BNE 1$
2752 014300 104065 ERROR +65
2753 014302 032777 000020 164626 1$: BIT #20, $SWR
2754 014310 001001 BNE 2$ ; BRANCH IF BINARY DECK

```

```

2755 014312 104066          ERROR +66
2756 014314 012737 177660 001250 2$: MOV    #-120, COUNT ; ONLY READ ONE CARD
2757 014322 032713 010000 PSRC2: BIT    #10000, @CDS ; CHECK FOR OFF-LINE
2758 014326 001415          BEQ    PSRE3
2759 014330 032713 040000          BIT    #40000, @CDS ; CHECK FOR CARD READER ERROR
2760 014334 001002          BNE    1$ ; BRANCH IF SET
2761 014336 104035          ERROR +35 ; OFF-LINE BUT NOT CARD READER ERROR
2762 014340 000414          BR     PSRE4
2763
2764 014342 004737 014650          1$: JSR    PC, LASTCD ; CHECK FOR LAST CARD
2765 014346 002402          BLT    2$ ; BRANCH IF NOT
2766 014350 000137 013714          JMP    PSRNC ; BRANCH IF BOTH CARD
2767 014354 104057          2$: ERROR +57 ; CARD READER ERROR BUT NOT BOTH CARD
2768 014356 000137 015306          JMP    DATRST ; RESTART THE ENTIRE TEST
2769
2770 014362 032713 040000 PSRE3: BIT    #40000, @CDS ; CHECK FOR CARD READER ERROR
2771 014366 001401          BEQ    PSRE4 ; BRANCH IF NOT
2772 014370 104060          ERROR +60 ; CARD READER ERROR BUT NOT OFF LINE
2773
2774 014372 032713 020000 PSRE4: BIT    #20000, @CDS
2775 014376 001401          BEQ    1$
2776 014400 104061          ERROR +61 ; END OF FILE ERROR (M1200 ONLY)
2777
2778 014402 032713 002000          1$: BIT    #2000, @CDS
2779 014406 001401          BEQ    2$
2780 014410 104063          ERROR +63 ; DATA LATE ERROR
2781
2782 014412 032713 001000          2$: BIT    #1000, @CDS
2783 014416 001401          BEQ    3$
2784 014420 104064          ERROR +64 ; NON-EXISTANT MEMORY ERROR
2785 014422 032713 077000          3$: BIT    #077000, @CDS ; CHECK ALL ERROR BITS
2786 014426 001001          BNE    PSRST ; BRANCH IF AT LEAST ONE
2787 014430 104037          ERROR +37 ; NONE OF THE ERROR BITS SET
2788 014432 000137 013722 PSRST: JMP    PSRLP ; GO CHECK THE DATA
2789
2790          ; RETURN PORTION OF INTERRUPT SERVICE ROUTINE
2791          ; CALCULATES A NEW "SIZE" (NUMBER OF COLUMNS TO BE READ)
2792          ; SETS UP THE CARD READER BUFFERS, AND ISSUES THE READ COMMAND
2793          ; THEN DOES AN RTI TO THE BACKGROUND ROUTINE
2794 014436 063737 015316 015314 SRETRN: ADD    OFFSET, SIZE
2795 014444 100404          BMI    SRETR1
2796 014446 012737 177660 015316          MOV    #-120, OFFSET
2797 014454 000770          BR     SRETRN
2798 014456 032737 001000 015314 SRETR1: BIT    #001000, SIZE
2799 014464 001004          BNE    SRETR4
2800 014466 012737 000120 015316 SKETR3: MOV    #120, OFFSET
2801 014474 000760          BR     SRETRN
2802 014476 004737 014650          SRETR4: JSR    PC, LASTCD ; CHECK FOR MORE THAN 80 CARDS
2803 014502 003371          BGT    SRETR3 ; BRANCH IF GREATER
2804 014504 013714 015314          MOV    SIZE, @CDC ; SET UP COLMN COUNT
2805 014510 012700 045442          MOV    #BUFBEG, @CDA ; RESET TABLE POINTER
2806 014514 010015          MOV    @CDA ; SET UP ADDRESS REG
2807 014516 005213          INC   @CDS ; READ
2808 014520 000002          RTI

```



```

2809
2810
2811 ;SUBROUTINE TO SET PACKING MODE AND MOVE THE POINTERS FOR THE DATA.
2812 014522 062737 000240 015326 PAKSET: ADD #160., TSTART ;MOVE TABLE POINTER TO PACKED TABLE
2813 014530 062737 000120 015330 ADD #80., TEND
2814 014536 062701 000240 ADD #160., R1 ;UPDATE TABLE POINTER
2815 014542 163701 015332 SUB CDCNT, R1 ;COMPENSATE FOR BYTES
2816 014546 052713 000602 BIS #2, @CDS ;SET PACKING MODE BIT
2817 014552 000207 RTS PC
2818
2819 ;SUBROUTINE TO TYPE HEADING, TYPE OF DECK, CARD COUNT, AND COLUMN COUNT
2820 014554 005737 001260 TYHEAD: TST ERFLG ;CHECK FOR FIRST ERROR
2821 014560 001004 BNE NOHEAD ;BRANCH IF NOT
2822 014562 005237 001260 INC ERFLG ;SET FLAG
2823 014566 104400 033053 TYPE, MSG13 ;TYPE HEADING FOR DATA ERRORS
2824 014572 104400 NOHEAD: TYPE ;OUTPUT TYPE OF DECK
2825 014574 000000 DECK: DUMMY ;POINTER TO DECK TITLE
2826 014576 104400 030545 TYPE, SPACE
2827 014602 005237 015332 INC CDCNT ;ADJUST CADR COUNT
2828 014604 013746 015332 MOV CDCNT, -(SP)
2829 014612 104404 TYPDS
2830 014614 005337 015332 DEC CDCNT ;READJUST CADR COUNT
2831 014620 104400 030540 TYPE, SPACE-5
2832 014624 005237 015334 INC CLCNT ;ADJUST COLUMN COUNT
2833 014630 013746 015334 MOV CLCNT, -(SP)
2834 014634 104404 TYPDS
2835 014636 005337 015334 DEC CLCNT ;READJUST COLUMN COUNT
2836 014642 104400 030543 TYPE, SPACE-2
2837 014646 000207 RTS PC
2838
2839 ;SUBROUTINE TO CHECK FOR LAST CARD
2840 014650 013737 015314 015322 LASTCD: MOV SIZE, TEMP1
2841 014656 013737 015332 015324 MOV CDCNT, TEMP2
2842 014664 005237 015324 LSTCD1: INC TEMP2
2843 014670 062737 000120 015322 ADD #120, TEMP1
2844 014676 100772 BMI LSTCD1
2845 014700 023727 015324 000120 CMP TEMP2, #80.
2846 014706 000207 RTS PC
2847
2848 ;SUBROUTINE TO KEEP TRACK OF CARDS
2849 014710 005037 015334 NXCRD: CLR CLCNT ;RESET COLUMN COUNT
2850 014714 005037 001212 CLR STMP5
2851 014720 005237 015332 INC CDCNT ;KEEP TRACK OF CARDS
2852 014724 023727 015332 000120 CMP CDCNT, #120 ;CHECK FOR BOTH CARD
2853 014732 002001 BGE SEOP
2854 014734 000207 RTS PC ;RETURN
2855
2856 ;*****
2857
2858 .SBTTL END OF PASS ROUTINE
2859
2860 ;*INCREMENT THE PASS NUMBER ($PASS)
2861 ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
2862

```

```

2863 ;*IF SW12=1 INHIBIT TRACE TRAP
2864 ;*IF THERES A MONITOR GO TO IT
2865 ;*IF THERE ISN'T JUMP TO HOOK1
2866
2867 SEOP:
2868 TST (SP)+ ;CORRECT STACK POINTER TO REPLACE 'RTS'
2869 CMP (SP)+,(SP)+ ;CORRECT STACK POINTER TO REPLACE 'RTI'
2870 CLR $STNM ;ZERO THE TEST NUMBER
2871 CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
2872 INC $PASS ;INCREMENT THE PASS NUMBER
2873 BIC #10000,$PASS ;DON'T ALLOW A NEG. NUMBER
2874 DEC (PC)+ ;LOOP?
2875 SEOPCT: .WORD 1
2876 BGT $DOAGN ;YES
2877 MOV (PC)+,$(PC)+ ;RESTORE COUNTER
2878 SENDCT: .WORD 1
2879 SEOPCT
2880 TYPE $SENDMG ;TYPE "END PASS #"
2881 MOV $PASS,-(SP) ;SAVE $PASS FOR TYPEOUT
2882 TYPDS ;GO TYPE--DECIMAL ASCII WITH SIGN
2883 TYPE , $ENULL ;TYPE A NULL CHARACTER
2884
2885 $GET42:
2886 MOV #42,R0 ;GET MONITOR ADDRESS
2887 BEQ $DOAGN ;BRANCH IF NO MONITOR
2888 RESET ;MONITOR,CLEAR WORLD
2889 CLR -(SP) ;INSURE THE "T" BIT IS CLEAR
2890 MOV $SENDAD,-(SP) ;SETUP FOR AN RTI OR RTT
2891 BR $RTRN ;GO DO AN RTI OR RTT TO LOAD THE PSW
;WITH A CLEARED "T" BIT
2892
2893 MOV #42,R0 ;GET MONITOR ADDRESS
2894 BEQ $DOAGN ;BRANCH IF NO MONITOR
2895 RESET ;CLEAR THE WORLD
2896 JSR PC,(R0) ;GO TO MONITOR
2897 NOP ;SAVE ROOM
2898 NOP ;FOR
2899 NOP ;ACT11
2900
2901 $DOAGN:
2902 CLR -(SP) ;RESERVE A STACK LOC. FOR THE PS
2903 MOV #34,-(SP) ;SETUP THE TRAP VECTOR
2904 MOV #15,$34 ;TO GET THE PS
2905 TRAP
2906
2907 15:
2908 TST (SP)+ ;CLEAN OFF THE USED PC
2909 MOV (SP)+,$(SP) ;SAVE OFF THE PS
2910 MOV (SP)+,$34 ;RESTORE TRAP VECTOR
2911 BIC #20,(SP) ;CLEAR THE "T" BIT
2912 BIT #BIT12,$SWR ;RUN WITH TRACE TRAP?
2913 BNE 25 ;BR IF NO
2914 COM $TBIT ;IS IT TIME FOR TRACE TRAP
2915 BMI 25 ;BR IF NO
2916 BIS #20,(SP) ;SET TRACE TRAP
2917 MOV $SLOOP,-(SP) ;JUMP TO START OF TEST
2918 SRTRN: RTI ;RETURN--THIS IS CHANGED TO
;AN "RTT" IF "RTT" IS LEGAL

```

```

2917                                     ;; INSTRUCTION
2918 015142 SLOOP: JMP #HOOK1 ;; RETURN
2919 015142 000137 015170 $TBIT: 0
2920 015146 000000 $ENDMG: .ASCIZ <15><12>/END PASS #/
2921 015150 005015 047105 020104
2922 015156 040520 051523 021440
2923 015164 000
2924 015165 377 377 000 $ENULL: .BYTE -1,-1,0 ;; NULL CHARACTER STRING
2925
2926 015170 032777 000040 163740 HOOK1: BIT #40, $SWR ;CHECK $SWR FOR HALT AT END OF DECK
2927 015176 001402 BEQ ONLINE ;CONTINUE IF NOT SET
2928 015200 070000 HALT ;END OF DECK, $WS SET
2929 015202 000427 BR DECKCK
2930
2931 015204 032713 010000 ONLINE: BIT #10000, $CDS ;CHECK FOR OFF-LINE
2932 015210 001424 BEQ DECKCK ;BRANCH IF NOT
2933 015212 005713 TST $CDS ;CHECK FOR ERROR (BIT 15)
2934 015214 100401 BMI 15 ;BRANCH IF SET OK
2935 015216 104026 ERROR +26 ;ERROR BIT 15 NOT SET
2936
2937 015220 032713 040000 1$: BIT #40000, $CDS ;CHECK FOR CARD READER ERROR
2938 015224 001001 BNE 25 ;BRANCH IF SET OK
2939 015226 104035 ERROR +35 ;OFF-LINE NOT DUE TO CARD READER ERROR
2940
2941 015230 032713 023471 2$: BIT #023471, $CDS ;CHECK FOR EXTRA BITS SET
2942 015234 001401 BEQ 35 ;BRANCH IF OK
2943 015236 104015 ERROR +15 ;EXTRA ERROR BITS SET
2944
2945 015240 012712 015250 3$: MOV #ONINT, $ADINT ;SET UP INTERRUPT VECTOR
2946 015244 000001 4$: WAIT ;WAIT FOR AN INTERRUPT
2947 015246 000776 BR 45 ;WAIT ON TRACE TRAPS
2948
2949 015250 032713 000010 ONINT: BIT #10, $CDS ;CHECK FOR TRANSITION TO ON LINE
2950 015254 001001 BNE 15 ;BRANCH IF SET OK
2951 015256 104036 ERROR +36 ;INTERRUPT BY OTHER THAN BIT 3 SETTING
2952
2953 015260 022626 1$: CMP (SP)+, (SP)+ ;RESTORE THE STACK
2954 ;WHEN CONTINUING FROM ONE DECK TO ANOTHER, CHECK $W6 FOR TYPE
2955 ;OF TESTING TO BE PERFORMED
2956 015262 005137 001254 DECKCK: COM TRFLG ;TOGGLE TRACE FLAG
2957 015266 032777 000100 163642 BIT #100, $SWR ;CHECK $W6
2958 015274 001402 BEQ 15 ;BRANCH IF NOT SET
2959 015276 000137 002534 JMP RESTR ;RERUN COMBINED INSTRUCTION AND DATA TEST
2960 015302 000137 012314 1$: JMP DATST
2961
2962 015306 022626 DATRST: CMP (SP)+, (SP)+ ;RESTORE THE STACK
2963 015310 000137 012314 JMP DATST ;RESTART DATA TEST
2964
2965 015314 177660 SIZE: -120
2966 015316 177660 OFFSET: -120
2967 015320 000000 BUFEND: 0
2968 015322 000000 TEMP1: 0
2969 015324 000000 TEMP2: 0
2970 015326 000000 TSTART: 0 ;STARTING ADDRESS OF DATA TABLE

```

N10

MAINDEC 11 - DZCDB-B MACY11 27(654) 1-JUL-77 08:39 PAGE 58
DZCDB.P11 END OF PASS ROUTINE

SEQ 0130

2971 015330 000000
2972 015332 000000
2973 015334 000000
2974 015336 000000
2975 015340 000000

TEND: 0
CDCNT: 0
CLCNT: 0
PTOFF: 0
DERCNT: 0

: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

```

2976
2977
2978 015342
2979 015342 012706 001100
2980 015346 005026
2981 015350 022706 001126
2982 015354 001374
2983 015356 012706 001100
2984 015362 012737 026326 000020
2985 015370 012737 000340 000022
2986 015376 012737 026152 000030
2987 015404 012737 000340 000032
2988 015412 012737 027374 000034
2989 015420 012737 000340 000036
2990 015426 012737 027430 000024
2991 015434 012737 000340 000026
2992 015442 013737 014774 014766
2993 015450 005037 001220
2994 015454 012737 015140 000014
2995 015462 012737 000340 000016
2996 015470 012737 000002 015140
2997 015476 012737 015524 000010
2998 015504 005046
2999 015506 012746 015514
3000 015512 000006
3001 015514 012737 000006 015140 64$:
3002 015522 000402
3003 015524 062706 000010 65$:
3004 015530 012737 000012 000010 66$:
3005 015536 005037 015146
3006 015542 012737 015542 001106
3007 015550 013746 000004
3008 015554 013746 000006
3009 015560 012737 015574 000004
3010 015566 005777 163344
3011 015572 000407
3012 015574 012737 000176 001136 67$:
3013 015602 012737 000174 001140
3014 015610 022626
3015 015612 012637 000006 68$:
3016 015616 012637 000004
3017 015622 104400 030550
3018 015626 104400 031333
3019
3020 015632 005037 001266
3021 015636 000137 016134
3022 015642
3023 015642 012706 001100
3024 015646 005026
3025 015650 022706 001126
3026 015654 001374
3027 015656 012706 001100
3028 015662 012737 026326 000020
3029 015670 012737 000340 000022

```

: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 #SRTN,#TBITVEC ;:SET "T" BIT VECTOR TO SRTN
MOV #340,#TBITVEC+2 ;:LEVEL 7
MOV #RTI,SRTN ;:SET SRTN TO A RTI
MOV #65S,#RESVEC ;:TRY TO DO A RTT
CLR -(SP) ;:DUMMY PS
MOV #64S,-(SP) ;:AND PC
RTT ;:TRY THE RTT
MOV #RTT,SRTN ;:RTT IS LEGAL--SET SRTN TO A RTT
BR 66S
ADD #10,SP ;:RTT ILLEGAL--CLEAN OFF THE STACK
MOV #RESVEC+2,#RESVEC ;:RESTORE TRAP CATCHER
CLR $TBIT ;:CLEAR "T" BIT SWITCH
MOV #,$LPRDR ;: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 68S ;: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,CRLF-3 ;:MOVE MESSAGE UP ON TTY
TYPE,M1200E ;:INDICATE "ENTERING ERROR FUNCTION
;:TESTING OF AN M-1200"
;:CARD READER IS M-1200
CLR C01000
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

```



```

3084 01620C 012714 177701      MOV      #-77,  @CDC      ;SET UP COLUMN COUNT
3085 016204 012715 045442      MOV      #BUFBEQ, @CDA   ;SET UP BUS ADDRESS
3086 016210 000000                HALT                       ;
3087 016212 005213                INC      @CDS              ;START READING
3088 016214 105713                TSTB    @CDS              ;CHECK FOR CONTROLLER READY
3089 016216 001001                BNE     1$                ;BRANCH IF SET OK
3090 016220 104023                ERROR  +23                ;CONTROLLER READY FAILED TO SET
3091
3092 016222 005713                1$:   TST      @CDS        ;CHECK FOR ERROR ( BIT 15)
3093 016224 001001                BNE     2$                ;BRANCH IF SET OK
3094 016226 104026                ERROR  +26                ;ERROR BIT 15 NOT SET
3095
3096 016230 032713 002000                2$:   BIT      #2000, @CDS ;CHECK FOR DATA LATE ERROR (BIT 10)
3097 016234 001001                BNE     3$                ;BRANCH IF SET OK
3098 016236 104042                ERROR  +42                ;DATA LATE BIT 10 NOT SET
3099
3100 016240 032713 075577                3$:   BIT      #075577, @CDS ;CHECK FOR ANY OTHER BITS
3101 016244 001401                BEQ     4$                ;BRANCH IF OK
3102 016246 104004                ERROR  +4                  ;EXTRA BITS SET IN STATUS WORD
3103
3104 016250                4$:
3105 ;*****
3106 ;*TEST 34      TEST ERROR AND OFF LINE BITS
3107 ;*****
3108 016250 000004                †ST34: SCOPE
3109 ;THE CARD READER GOING OFF-LINE SHOULD SET ERROR (BIT 15)
3110 ;AND OFF-LINE (BIT 12)
3111 ;GOING BACK ON LINE SHOULD SET "TRANSITION TO ON-LINE" (BIT 3)
3112 016252 004737 025744                JSR     PC, INIT          ;INITIALIZE STATUS REGISTER
3113 016256 104400 035675                TYPE,  MSG33             ;TST34 OFF-LINE TEST
3114 016262 104400 031706                TYPE,  MSG3              ;"PRESS CARD READER 'STOP'"
3115 016266 104400 031632                TYPE,  MSG2              ;"THEN HIT 'CONTINUE' ON THE CONSOLE"
3116 016272 104400 030550                TYPE,  CRLF-3           ;MOVE MESSAGE UP ON TTY
3117 016276 000000                HALT
3118 016300 032713 010000                BIT     #10000, @CDS     ;CHECK BIT 12
3119 016304 001001                BNE     1$                ;BRANCH IF SET
3120 016306 104043                ERROR  +43                ;OFF-LINE (BIT 12) WASN'T SET
3121
3122 016310 005713                1$:   TST      @CDS        ;CHECK BIT 15
3123 016312 100401                BNE     2$                ;BRANCH IF SET
3124 016314 104026                ERROR  +26                ;ERROR (BIT 15) WASN'T SET
3125
3126 016316 031327 067577                2$:   BIT      @CDS, #067577 ;CHECK FOR EXTRA BITS
3127 016322 001401                BEQ     3$                ;BRANCH IF OK
3128 016324 104015                ERROR  +15                ;STATUS WORD ERROR
3129
3130 016326 104400 031576                3$:   TYPE,    MSG1          ;"PRESS CARD READER 'RESET'";
3131 016332 104400 031632                TYPE,    MSG2          ;"THEN HIT 'CONTINUE' ON THE CONSOLE"
3132 016336 104400 030550                TYPE,    CRLF-3       ;MOVE MESSAGE UP ON TTY
3133 016342 000000                HALT
3134
3135 016344 032713 000010                BIT     #10, @CDS       ;CHECK FOR TRANSITION TO ON-LINE(BIT 3)
3136 016350 001001                BNE     4$                ;BRANCH IF SET OK
3137 016352 104036                ERROR  +36                ;TRANSITION TO ON-LINE FAILED TO SET

```

```

3138
3139 016354 032713 010000 4$: BIT #10000, @CDS ;CHECK FOR OFF-LINE
3140 016360 001401 BEQ 5$ ;BRANCH IF OK
3141 016362 104044 ERROR +44 ;OFF-LINE STILL SET
3142
3143 016364 005713 5$: TST @CDS ;CHECK ERROR (BIT 15)
3144 016366 100401 BMI 6$ ;BRANCH IF STILL SET
3145 016370 104026 ERROR +26 ;ERROR BIT 15 CLEARED
3146
3147 016372 032713 077567 6$: BIT #077567, @CDS ;CHECK FOR EXTRA BITS
3148 016376 001401 BEQ 7$ ;BRANCH IF OK
3149 016400 104015 ERROR +15 ;EXTRA STATUS BITS SET
3150
3151 016402 7$:
3152 *****
3153 016402 000004 †ST35: SCOPE
3154 ;TRYING TO READ WHEN CARD READER IS OFF-LINE SHOULD CAUSE AN INTERRUPT
3155 ;CHECK THAT AN INTERRUPT OCCURS WHEN THE CARD READER COMES ON LINE
3156 016404 004737 025744 JSR PC INIT ;INITIALIZE STATUS REGISTER
3157 016410 012712 016652 MOV #TINTC, @ADINT ;LOAD RETURN POINTER
3158 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3159 016414 005046 CLR -(SP) ;;
3160 016416 013746 000034 MOV 34, -(SP) ;;SAVE CURRENT TRAP VECTOR
3161 016422 012737 016432 000034 MOV #64$, 34 ;;SETUP NEW TRAP VECTOR
3162 016430 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
3163 016432 016666 000002 000006 64$: MOV 2(SP), 6(SP) ;;
3164 016440 012716 016446 MOV #65$, (SP) ;;REPLACE OLD PC WITH NEW
3165 016444 000002 RTI ;;RESTORE PSW
3166 016446 012637 000034 65$: MOV (SP)+, 34 ;;RESTORE OLD TRAP VECTOR
3167 016452 012637 001214 MOV (SP)+, $TMP6 ;;
3168 016456 052737 000340 001214 BIS #340, $TMP6 ;;PUT NEW PS ON STACK
3169 016464 013746 001214 MOV $TMP6, -(SP) ;;PUT NEW PC ON STACK
3170 016470 012746 016476 MOV #66$, -(SP) ;;POP NEW PC AND PS
3171 016474 000002 RTI ;;
3172 016476 66$:
3173 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(@DINT)"
3174 016476 005046 CLR -(SP) ;;
3175 016500 013746 000034 MOV 34, -(SP) ;;SAVE CURRENT TRAP VECTOR
3176 016504 012737 016514 000034 MOV #67$, 34 ;;SETUP NEW TRAP VECTOR
3177 016512 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
3178 016514 016666 000002 000006 67$: MOV 2(SP), 6(SP) ;;
3179 016522 012716 016530 MOV #68$, (SP) ;;REPLACE OLD PC WITH NEW
3180 016526 000002 RTI ;;RESTORE PSW
3181 016530 012637 000034 68$: MOV (SP)+, 34 ;;RESTORE OLD TRAP VECTOR
3182 016534 012662 000002 MOV (SP)+, 2(@DINT) ;;
3183 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3184 016540 005046 CLR -(SP) ;;
3185 016542 013746 000034 MOV 34, -(SP) ;;SAVE CURRENT TRAP VECTOR
3186 016546 012737 016556 000034 MOV #69$, 34 ;;SETUP NEW TRAP VECTOR
3187 016554 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
3188 016556 016666 000002 000006 69$: MOV 2(SP), 6(SP) ;;
3189 016564 012716 016572 MOV #70$, (SP) ;;REPLACE OLD PC WITH NEW
3190 016570 000002 RTI ;;RESTORE PSW
3191 016572 012637 000034 70$: MOV (SP)+, 34 ;;RESTORE OLD TRAP VECTOR

```


| | | | | | | | |
|------|--------|--------|--------|--------|-------|-----------------|-----------------------------------------------------------------|
| 3192 | 016576 | 012637 | 001214 | | MOV | (SP)+,\$TMP6 | |
| 3193 | 016602 | 042737 | 000340 | 001214 | BIC | #340,\$TMP6 | |
| 3194 | 016610 | 013746 | 001214 | | MOV | \$TMP6,-(SP) | :::PUT NEW PS ON STACK |
| 3195 | 016614 | 012746 | 016622 | | MOV | #71\$,-(SP) | :::PUT NEW PC ON STACK |
| 3196 | 016620 | 000002 | | | RTI | | :::POP NEW PC AND PS |
| 3197 | 016622 | | | 71\$: | | | |
| 3198 | 016622 | 012713 | 000100 | | MOV | #100,\$CDS | :::SET INTERRUPT ENABLE |
| 3199 | 016626 | 104400 | 035724 | | TYPE, | MSG34 | :::TST35 SPECIAL INT. COND. TEST |
| 3200 | 016632 | 104400 | 031706 | | TYPE, | MSG3 | :::"PRESS CARD READER 'STOP'" |
| 3201 | 016636 | 104400 | 030550 | | TYPE, | CRLF-3 | :::MOVE MESSAGE UP ON TTY |
| 3202 | 016642 | 032713 | 010000 | TLOPC: | BIT | #10000,\$CDS | :::WAIT FOR OFF-LINE TO SET |
| 3203 | 016646 | 001775 | | | BEQ | TLOPC | |
| 3204 | 016650 | 000402 | | | BR | CONTC | :::SKIP INTERRUPT HANDLER |
| 3205 | | | | | | | |
| 3206 | 016652 | 104021 | | TINTC: | ERROR | +21 | :::'STOP' SHOULDN'T CAUSE AN INTERRUPT |
| 3207 | 016654 | 000002 | | | RTI | | :::RETURN FROM THE INTERRUPT |
| 3208 | | | | | | | |
| 3209 | 016656 | 105713 | | CONTC: | TSTB | \$CDS | :::CHECK CONTROLLER READY BIT 7 |
| 3210 | 016660 | 100401 | | | BMI | 1\$ | :::BRANCH IF OK |
| 3211 | 016662 | 104023 | | | ERROR | +23 | :::CU READY DIDN'T SET YET |
| 3212 | | | | | | | |
| 3213 | 016664 | 005713 | | 1\$: | TST | \$CDS | :::CHECK ERROR BIT |
| 3214 | 016666 | 100401 | | | BMI | 2\$ | :::BRANCH IF SET |
| 3215 | 016670 | 104026 | | | ERROR | +26 | :::ERROR (BIT 15) NOT SET |
| 3216 | | | | | | | |
| 3217 | 016672 | 032713 | 067477 | 2\$: | BIT | #067477,\$CDS | :::CHECK FOR EXTRA BITS |
| 3218 | 016676 | 001401 | | | BEQ | 3\$ | :::BRANCH IF OK |
| 3219 | 016700 | 104004 | | | ERROR | +4 | :::STATUS WORD ERROR |
| 3220 | | | | | | | |
| 3221 | 016702 | 012712 | 017120 | 3\$: | MOV | #TINTCA,\$ADINT | :::LOAD RETURN POINTER |
| 3222 | | | | | | | :::THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS" |
| 3223 | 016706 | 005046 | | | CLR | -(SP) | ::: |
| 3224 | 016710 | 013746 | 000034 | | MOV | 34,-(SP) | :::SAVE CURRENT TRAP VECTOR |
| 3225 | 016714 | 012737 | 016724 | 000034 | MOV | #64\$,34 | :::SETUP NEW TRAP VECTOR |
| 3226 | 016722 | 104400 | | | TRAP | | :::PUSH OLD PSW AN PCON STACK |
| 3227 | 016724 | 016666 | 000002 | 000006 | 64\$: | MOV | 2(SP),6(SP) |
| 3228 | 016732 | 012716 | 016740 | | MOV | #65\$, (SP) | ::: |
| 3229 | 016736 | 000002 | | | RTI | | :::REPLACE OLD PC WITH NEW |
| 3230 | 016740 | 012637 | 000034 | 65\$: | MOV | (SP)+,34 | :::RESTORE PSW |
| 3231 | 016744 | 012637 | 001214 | | MOV | (SP)+,\$TMP6 | :::RESTORE OLD TRAP VECTOR |
| 3232 | 016750 | 052737 | 000340 | 001214 | BIS | #340,\$TMP6 | |
| 3233 | 016756 | 013746 | 001214 | | MOV | \$TMP6,-(SP) | :::PUT NEW PS ON STACK |
| 3234 | 016762 | 012746 | 016770 | | MOV | #66\$,-(SP) | :::PUT NEW PC ON STACK |
| 3235 | 016766 | 000002 | | | RTI | | :::POP NEW PC AND PS |
| 3236 | 016770 | | | 66\$: | | | |
| 3237 | | | | | | | :::THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)" |
| 3238 | 016770 | 005046 | | | CLR | -(SP) | ::: |
| 3239 | 016772 | 013746 | 000034 | | MOV | 34,-(SP) | :::SAVE CURRENT TRAP VECTOR |
| 3240 | 016776 | 012737 | 017006 | 000034 | MOV | #67\$,34 | :::SETUP NEW TRAP VECTOR |
| 3241 | 017004 | 104400 | | | TRAP | | :::PUSH OLD PSW AN PCON STACK |
| 3242 | 017006 | 016666 | 000002 | 000006 | 67\$: | MOV | 2(SP),6(SP) |
| 3243 | 017014 | 012716 | 017022 | | MOV | #68\$, (SP) | ::: |
| 3244 | 017020 | 000002 | | | RTI | | :::REPLACE OLD PC WITH NEW |
| 3245 | 017022 | 012637 | 000034 | 68\$: | MOV | (SP)+,34 | :::RESTORE PSW |
| | | | | | | | :::RESTORE OLD TRAP VECTOR |

```

3246 017026 012662 000002      MOV      (SP)+,2(ADINT)
3247      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3248 017032 005046      CLR      -(SP)
3249 017034 013746 000034      MOV      34,-(SP)
3250 017040 012737 017050 000034      MOV      #69$,34
3251 017046 104400      TRAP
3252 017050 016666 000002 000006 69$:      MOV      2(SP),6(SP)
3253 017056 012716 017064      MOV      #70$,1(SP)
3254 017062 000002      RTI
3255 017064 012537 000034 70$:      MOV      (SP)+,34
3256 017070 012637 001214      MOV      (SP)+,$TMP6
3257 017074 042737 000340 001214      BIC      #340,$TMP6
3258 017102 013746 001214      MOV      $TMP6,-(SP)
3259 017106 012746 017114      MOV      #71$,-(SP)
3260 017112 000002      RTI
3261 017114 71$:
3262 017114 005213      INC      @CDS
3263 017116 000777      BR
3264
3265 017120 022626      TINTCA: CMP      (SP)+, (SP)+
3266 017122 105713      TSTB     @CDS
3267 017124 100401      BMI     1$
3268 017126 104023      ERROR +23
3269
3270 017130 032713 010000 1$:      BIT      #10000, @CDS
3271 017134 001001      BNE     2$
3272 017136 104043      ERROR +43
3273
3274 017140 005713 2$:      TST      @CDS
3275 017142 100401      BMI     3$
3276 017144 104026      ERROR +26
3277
3278 017146 032713 067477 3$:      BIT      #067477,@CDS
3279 017152 001401      BEQ     4$
3280 017154 104004      ERROR +4
3281
3282 017156 012712 017402 4$:      MOV      #TINTCB,@ADINT
3283      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3284 017162 005046      CLR      -(SP)
3285 017164 013746 000034      MOV      34,-(SP)
3286 017170 012737 017200 000034      MOV      #64$,34
3287 017176 104400      TRAP
3288 017200 016666 000002 000006 64$:      MOV      2(SP),6(SP)
3289 017206 012716 017214      MOV      #65$,1(SP)
3290 017212 000002      RTI
3291 017214 012637 000034 65$:      MOV      (SP)+,34
3292 017220 012637 001214      MOV      (SP)+,$TMP6
3293 017224 052737 000340 001214      BIS      #340,$TMP6
3294 017232 013746 001214      MOV      $TMP6,-(SP)
3295 017236 012746 017244      MOV      #66$,-(SP)
3296 017242 000002      RTI
3297 017244 66$:
3298      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
3299 017244 005046      CLR      -(SP)
    
```

```

3300 017246 013746 000034      MOV      34,-(SP)      ;;SAVE CURRENT TRAP VECTOR
3301 017252 012737 017262 000034      MOV      #67$,34      ;;SETUP NEW TRAP VECTOT
3302 017260 104400      TRAP                      ;;PUSH OLD PSW AN PCON STACK
3303 017262 016666 000002 000006 67$:      MOV      2(SP),6(SP)  ;;
3304 017270 012716 017276      MOV      #68$, (SP)  ;;REPLACE OLD PC WITH NEW
3305 017274 000002      RTI                      ;;RESTORE PSW
3306 017276 012637 000034 68$:      MOV      (SP)+,34      ;;RESTORE OLD TRAP VECTOR
3307 017302 012663 000002      MOV      (SP)+,2(ADINT)
3308      ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3309 017306 005046      CLR      -(SP)          ;;
3310 017310 013746 000034      MOV      34,-(SP)      ;;SAVE CURRENT TRAP VECTOR
3311 017314 012737 017324 000034      MOV      #69$,34      ;;SETUP NEW TRAP VECTOT
3312 017322 104400      TRAP                      ;;PUSH OL' PSW AN PCON STACK
3313 017324 016666 000002 000006 69$:      MOV      2(SP),6(SP)  ;;
3314 017332 012716 017340      MOV      #70$, (SP)  ;;REPLACE OLD PC WITH NEW
3315 017336 000002      RTI                      ;;RESTORE PSW
3316 017340 012637 000034 70$:      MOV      (SP)+,34      ;;RESTORE OLD TRAP VECTOR
3317 017344 012637 001214      MCV      (SP)+,$TMP6
3318 017350 042737 000340 001214      BIC      #340,$TMP6
3319 017356 013746 001214      MOV      $TMP6,-(SP)  ;;PUT NEW PS ON STACK
3320 017362 012746 017370      MOV      #71$,-(SP)  ;;PUT NEW PC ON STACK
3321 017366 000002      RTI                      ;;POP NEW PC AND PS
3322 017370 71$:
3323 017370 104400 031576      TYPE,    MSG1          ;"PRESS CARD READER 'RESET'"
3324 017374 104400 030550      TYPE,    CRLF-3       ;MOVE MESSAGE UP ON TTY
3325 017400 000777      BR      .              ;WAIT FOR THE INTERRUPT
3326
3327 017402 022626      TINTCB: CMP      (SP)+, (SP)+  ;;RESTORE THE STACK
3328 017404 032713 000010      BIT      #10, @CDS     ;;CHECK FOR TRANSITION TO ON-LINE(BIT 3)
3329 017410 001001      BNE      1$           ;;BRANCH IF SET OK
3330 017412 104036      ERROR +36           ;;TRANSITION TO ON-LINE FAILED TO SET
3331
3332 017414 032713 010000 1$:      BIT      #10000, @CDS  ;;CHECK FOR OFF-LINE
3333 017420 001401      BEQ      2$           ;;BRANCH IF OK
3334 017422 104044      ERROR +44           ;;OFF-LINE STILL SET
3335
3336 017424 005713 2$:      TST      @CDS         ;;CHECK ERROR (BIT 15)
3337 017426 100401      BMI      3$           ;;BRANCH IF STILL SET
3338 017430 104026      ERROR +26           ;;ERROR BIT 15 CLEARED
3339
3340 017432 032713 077467 3$:      BIT      #077467,@CDS  ;;CHECK FOR EXTRA BITS
3341 017436 001401      BEQ      4$           ;;BRANCH IF OK
3342 017440 104015      ERROR +15           ;;EXTRA STATUS BITS SET
3343
3344 017442 4$:
3345      ;*****
3346      ;*TEST 36 TEST INPUT HOPPER EMPTY
3347      ;*****
3348 017442 000004      †ST36: SCOPE
3349      ;INPUT HOPPER EMPTY SHOULD SET SPECIAL CONDITION
3350      ;CHECK THAT INTERRUPTS OCCUR WHEN THE CARD READER COMES ON LINE
3351 017444 004737 025744      JSR      PC INIT      ;INITIALIZE STATUS REGISTER
3352 017450 104400 035765      TYPE,    MSG35        ;TST36 HOPPER EMPTY TEST
3353 017454 104400 031774      TYPE,    MSG5         ;"REMOVE ALL CARDS FROM THE INPUT HOPPER"

```

```

3354 017460 104400 031632 TYPE, MSG2 ;"THEN HIT 'CONTINUE' ON THE CONSOLE"
3355 017464 104400 030550 TYPE, CRLF-3 ;MOVE MESSAGE UP ON TTY
3356 017470 000000 HALT
3357 017472 032713 010000 BIT #10000, @CDS ;CHECK BIT 12
3358 017476 001001 BNE 15 ;BRANCH IF SET
3359 017500 104043 ERROR +43 ;OFF-LINE (BIT 12) WASN'T SET
3360
3361 017502 005713 15: TST @CDS ;CHECK ERROR BIT
3362 017504 100401 BMI 25 ;BRANCH IF SET
3363 017506 104026 ERROR +26 ;ERRGR (BIT 15) NOT SET
3364
3365 017510 032713 040000 25: BIT #40000, @CDS ;CHECK FOR CARD READER ERROR
3366 017514 001001 BNE 35 ;BRANCH IF SET
3367 017516 104035 ERROR +35 ;CARD READER ERROR BIT 14 NOT SET
3368
3369 017520 032713 027573 35: BIT #027573, @CDS ;CHECK FOR EXTRA BITS
3370 017524 001401 BEQ 45 ;BRANCH IF OK
3371 017526 104015 ERROR +15 ;STATUS WORD ERROR
3372
3373 017530 012712 017764 45: MOV #TINTD, @ADINT ;LOAD RETURN POINTER
3374 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340, PS"
3375 017534 005046 CLR -(SP) ;;
3376 017536 013746 000034 MOV 34, -(SP) ;;SAVE CURRENT TRAP VECTOR
3377 017542 012737 017552 000034 MOV #64$, 34 ;;SETUP NEW TRAP VECTOT
3378 017550 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
3379 017552 016666 000002 000006 64$: MOV 2(SP), 6(SP) ;;
3380 017560 012716 017566 MOV #65$, (SP) ;;REPLACE OLD PC WITH NEW
3381 017564 000002 RTI ;;RESTORE PSW
3382 017566 012637 000034 65$: MOV (SP)+, 34 ;;RESTORE OLD TRAP VECTOR
3383 017572 012637 001214 MOV (SP)+, $TMP6 ;;
3384 017576 052737 000340 C01214 BIS #340, $TMP6 ;;PUT NEW PS ON STACK
3385 017604 013746 001214 MOV $TMP6, -(SP) ;;PUT NEW PC ON STACK
3386 017610 012746 017616 MOV #66$, -(SP) ;;POP NEW PC AND PS
3387 017614 000002 RTI ;;
3388 017616 66$:
3389 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, 2(ADINT)"
3390 017616 005046 CLR -(SP) ;;
3391 017620 013746 000034 MOV 34, -(SP) ;;SAVE CURRENT TRAP VECTOR
3392 017624 012737 017634 000034 MOV #67$, 34 ;;SETUP NEW TRAP VECTOT
3393 017632 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
3394 017634 016666 000002 000006 67$: MOV 2(SP), 6(SP) ;;
3395 017642 012716 017650 MOV #68$, (SP) ;;REPLACE OLD PC WITH NEW
3396 017646 000002 RTI ;;RESTORE PSW
3397 017650 012637 000034 68$: MOV (SP)+, 34 ;;RESTORE OLD TRAP VECTOR
3398 017654 012662 000002 MOV (SP)+, 2(ADINT) ;;
3399 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340, FS"
3400 017660 005046 CLR -(SP) ;;
3401 017662 013746 000034 MOV 34, -(SP) ;;SAVE CURRENT TRAP VECTOR
3402 017666 012737 017676 000034 MOV #69$, 34 ;;SETUP NEW TRAP VECTOT
3403 017674 104400 TRAP ;;PUSH OLD PSW AN PCON STACK
3404 017676 016666 000002 000006 69$: MOV 2(SP), 6(SP) ;;
3405 017704 012716 017712 MOV #70$, (SP) ;;REPLACE OLD PC WITH NEW
3406 017710 000002 RTI ;;RESTORE PSW
3407 017712 012637 000034 70$: MOV (SP)+, 34 ;;RESTORE OLD TRAP VECTOR
    
```

```

3408 017716 012637 001214      MOV      (SP)+,$TMP6
3409 017722 042737 000340 001214      BIC      #340,$TMP6
3410 017730 013746 001214      MOV      $TMP6,-(SP)      ;; PUT NEW PS ON STACK
3411 017734 012746 017742      MOV      #71$,-(SP)      ;; PUT NEW PC ON STACK
3412 017740 000002      RTI      ;; POP NEW PC AND PS
3413 017742      71$:
3414 017742 012713 000100      MOV      #100, 2CDS      ;; SET INTERRUPT ENABLE
3415 017746 104400 032045      TYPE,    MSG6            ;; "RESTORE CARDS TO THE INPUT HOPPER"
3416 017752 104400 031576      TYPE,    MSG1            ;; "PRESS CARD READER 'RESET'"
3417 017756 104400 030550      TYPE,    CRLF-3         ;; MOVE MESSAGE UP ON TTY
3418 017762 000777      BR      .                ;; WAIT FOR THE INTERRUPT
3419
3420 017764 022626      TINTD:  CMP      (SP)+,(SP)+      ;; RESTORE THE STACK
3421 017766 012712 020214      MOV      #TINTDA,2ADINT  ;; LOAD RETURN POINTER
3422      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3423 017772 005046      CLR      -(SP)           ;;
3424 017774 013746 000034      MOV      34,-(SP)        ;; SAVE CURRENT TRAP VECTOR
3425 020000 012737 020010 000034      MOV      #64$,34        ;; SETUP NEW TRAP VECTOR
3426 020006 104400      TRAP     ;; PUSH OLD PSW AN PCON STACK
3427 020010 016666 000002 000006 64$:      MOV      2(SP),6(SP)     ;;
3428 020016 012716 020024      MOV      #65$, (SP)     ;;
3429 020022 000002      RTI      ;; REPLACE OLD PC WITH NEW
3430 020024 012637 000034      MOV      (SP)+,34        ;; RESTORE PSW
3431 020030 012637 001214      MOV      (SP)+,$TMP6     ;; RESTORE OLD TRAP VECTOR
3432 020034 052737 000340 001214      BIS      #340,$TMP6
3433 020042 013746 001214      MOV      $TMP6,-(SP)     ;; PUT NEW PS ON STACK
3434 020046 012746 020054      MOV      #66$,-(SP)     ;; PUT NEW PC ON STACK
3435 020052 000002      RTI      ;; POP NEW PC AND PS
3436 020054      66$:
3437      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
3438 020054 005046      CLR      -(SP)           ;;
3439 020056 013746 000034      MOV      34,-(SP)        ;; SAVE CURRENT TRAP VECTOR
3440 020062 012737 020072 000034      MOV      #67$,34        ;; SETUP NEW TRAP VECTOR
3441 020070 104400      TRAP     ;; PUSH OLD PSW AN PCON STACK
3442 020072 016666 000002 000006 67$:      MOV      2(SP),6(SP)     ;;
3443 020100 012716 020106      MOV      #68$, (SP)     ;;
3444 020104 000002      RTI      ;; REPLACE OLD PC WITH NEW
3445 020106 012637 000034      MOV      (SP)+,34        ;; RESTORE PSW
3446 020112 012662 000002      MOV      (SP)+,2(ADINT)  ;; RESTORE OLD TRAP VECTOR
3447      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3448 020116 005046      CLR      -(SP)           ;;
3449 020120 013746 000034      MOV      34,-(SP)        ;; SAVE CURRENT TRAP VECTOR
3450 020124 012737 020134 000034      MOV      #69$,34        ;; SETUP NEW TRAP VECTOR
3451 020132 104400      TRAP     ;; PUSH OLD PSW AN PCON STACK
3452 020134 016666 000002 000006 69$:      MOV      2(SP),6(SP)     ;;
3453 020142 012716 020150      MOV      #70$, (SP)     ;;
3454 020146 000002      RTI      ;; REPLACE OLD PC WITH NEW
3455 020150 012637 000034      MOV      (SP)+,34        ;; RESTORE PSW
3456 020154 012637 001214      MOV      (SP)+,$TMP6     ;; RESTORE OLD TRAP VECTOR
3457 020160 042737 000340 001214      BIC      #340,$TMP6
3458 020166 013746 001214      MOV      $TMP6,-(SP)     ;; PUT NEW PS ON STACK
3459 020172 012746 020200      MOV      #71$,-(SP)     ;; PUT NEW PC ON STACK
3460 020176 000002      RTI      ;; POP NEW PC AND PS
3461 020200      71$:

```

```

3462 020200 012714 177701          MOV    #-77,  @CDC    ;SET UP COLUMN COUNT
3463 020204 012715 045442          MOV    #BUFBE, @CDA   ;SET UP BUS ADDRESS
3464 020210 005213                    INC    @CDS           ;START READING
3465 020212 000777                    BR     .              ;WAIT FOR AN INTERRUPT
3466
3467 020214 022626                    TINTDA: CMP   (SP)+, (SP)+ ;RESIORE THE STACK
3468 020216 022713 000300          CMP    #000300, @CDS ;CHECK THE CARD READER STATUS
3469 020222 001471                    BEQ    1$             ;BRANCH IF OK
3470 020224 104004                    ERROR  +4             ;CARD READER STATUS ERROR
3471
3472 020226                    1$:
3473 ;*****
3474 ;*TEST 37 TEST OUTPUT STACKER FULL
3475 ;*****
3476 020226 000004          TST37: SCOPE
3477 ;OUTPUT STACKER FULL SHOULD SET BITS 15, 14, 12, 7
3478 020230 004737 025744          JSR    PC, INIT      ;INITIALIZE STATUS REGISTER
3479 020234 104400 036020          TYPE,  MSG36         ;TST37 STACKER FULL TEST
3480 020240 104400 032111          TYPE,  MSG7          ;"PULL OUTPUT STACKER PRESSURE ARM
3481 ;ALL THE WAY DOWN"
3482 020244 104400 031632          TYPE,  MSG2          ;"THEN HIT 'CONTINUE' ON THE CONSOLE"
3483 020250 104400 030550          TYPE,  CRLF-3        ;MOVE MESSAGE UP ON TTY
3484 020254 000000                    HALT
3485 020256 032713 010000          BIT    #10000, @CDS ;CHECK OFF-LINE BIT 12
3486 020262 001001                    BNE    1$             ;BRANCH IF SET
3487 020264 104043                    ERROR  +43           ;OFF-LINE (BIT 12) WASN'T SET
3488
3489 020266 005713                    1$: TST    @CDS        ;CHECK ERROR BIT 15
3490 020270 100401                    BMI    2$             ;BRANCH IF SET
3491 020272 104026                    ERROR  +26           ;ERROR BIT 15 NOT SET
3492
3493 020274 032713 040000          2$: BIT    #40000, @CDS ;CHECK FOR CARD READER ERROR
3494 020300 001001                    BNE    3$             ;BRANCH IF SET
3495 020302 104035                    ERROR  +35           ;CARD READER ERROR BIT 14 NOT SET
3496
3497 020304 032713 027577          3$: BIT    #027577, @CDS ;CHECK FOR EXTRA BITS
3498 020310 001401                    BEQ    4$             ;BRANCH IF OK
3499 020312 104015                    ERROR  +15           ;STATUS WORD ERROR
3500
3501 020314 012712 020544          4$: MOV    #TINTE, @ADINT ;LOAD RETURN POINTER
3502 ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340, PS"
3503 020320 005046                    CLR    -(SP)          ;;
3504 020322 013746 000034          MOV    34, -(SP)     ;;SAVE CURRENT TRAP VECTOR
3505 020326 012737 020336 000034  MOV    #64$, 34      ;;SETUP NEW TRAP VECTOR
3506 020334 104400                    TRAP                          ;;PUSH OLD PSW AN PC ON STACK
3507 020336 016666 000002 000006  64$: MOV    2(SP), 6(SP) ;;
3508 020344 012716 020352          MOV    #65$, (SP)   ;;REPLACE OLD PC WITH NEW
3509 020350 000002                    RTI                          ;;RESTORE PSW
3510 020352 012637 000034          65$: MOV    (SP)+, 34  ;;RESTORE OLD TRAP VECTOR
3511 020356 012637 001214          MOV    (SP)+, $TMP6
3512 020362 052737 000340 001214  BIS    #340, $TMP6
3513 020370 013746 001214          MOV    $TMP6, -(SP)  ;;PUT NEW PS ON STACK
3514 020374 012746 020402          MOV    #66$, -(SP)  ;;PUT NEW PC ON STACK
3515 020400 000002                    RTI                          ;;POP NEW PC AND PS

```

| | | | | | | | | | | |
|------|--------|--------|--------|--------|--------|--|--|--|--|--|
| 3516 | 020402 | | | | 66\$: | | | | | |
| 3517 | | | | | | | | | | |
| 3518 | 020402 | 005046 | | | | | | | | |
| 3519 | 020404 | 013746 | 000034 | | | | | | | |
| 3520 | 020410 | 012737 | 020420 | 000034 | | | | | | |
| 3521 | 020416 | 104400 | | | | | | | | |
| 3522 | 020420 | 016666 | 000002 | 000006 | 67\$: | | | | | |
| 3523 | 020426 | 012716 | 020434 | | | | | | | |
| 3524 | 020432 | 000002 | | | | | | | | |
| 3525 | 020434 | 012637 | 000034 | | 68\$: | | | | | |
| 3526 | 020440 | 012662 | 000002 | | | | | | | |
| 3527 | | | | | | | | | | |
| 3528 | 020444 | 005046 | | | | | | | | |
| 3529 | 020446 | 013746 | 000034 | | | | | | | |
| 3530 | 020452 | 012737 | 020462 | 000034 | | | | | | |
| 3531 | 020460 | 104400 | | | | | | | | |
| 3532 | 020462 | 016666 | 000002 | 000006 | 69\$: | | | | | |
| 3533 | 020470 | 012716 | 020476 | | | | | | | |
| 3534 | 020474 | 000002 | | | | | | | | |
| 3535 | 020476 | 012637 | 000034 | | 70\$: | | | | | |
| 3536 | 020502 | 012637 | 001214 | | | | | | | |
| 3537 | 020506 | 042737 | 000340 | 001214 | | | | | | |
| 3538 | 020514 | 013746 | 001214 | | | | | | | |
| 3539 | 020520 | 012746 | 020526 | | | | | | | |
| 3540 | 020524 | 000002 | | | | | | | | |
| 3541 | 020526 | | | | 71\$: | | | | | |
| 3542 | 020526 | 012713 | 000100 | | | | | | | |
| 3543 | 020532 | 104400 | 031576 | | | | | | | |
| 3544 | 020536 | 104400 | 030550 | | | | | | | |
| 3545 | 020542 | 000777 | | | | | | | | |
| 46 | | | | | | | | | | |
| 3547 | 020544 | 022626 | | | TINTE: | | | | | |
| 3548 | 020546 | 012712 | 020774 | | | | | | | |
| 3549 | | | | | | | | | | |
| 3550 | 020552 | 005046 | | | | | | | | |
| 3551 | 020554 | 013746 | 000034 | | | | | | | |
| 3552 | 020560 | 012737 | 020570 | 000034 | | | | | | |
| 3553 | 020566 | 104400 | | | | | | | | |
| 3554 | 020570 | 016666 | 000002 | 000006 | 64\$: | | | | | |
| 3555 | 020576 | 012716 | 020604 | | | | | | | |
| 3556 | 020602 | 000002 | | | | | | | | |
| 3557 | 020604 | 012637 | 000034 | | 65\$: | | | | | |
| 3558 | 020610 | 012637 | 001214 | | | | | | | |
| 3559 | 020614 | 052737 | 000340 | 001214 | | | | | | |
| 3560 | 020622 | 013746 | 001214 | | | | | | | |
| 3561 | 020626 | 012746 | 020634 | | | | | | | |
| 3562 | 020632 | 000002 | | | | | | | | |
| 3563 | 020634 | | | | 66\$: | | | | | |
| 3564 | | | | | | | | | | |
| 3565 | 020634 | 005046 | | | | | | | | |
| 3566 | 020636 | 013746 | 000034 | | | | | | | |
| 3567 | 020642 | 012737 | 020652 | 000034 | | | | | | |
| 3568 | 020650 | 104400 | | | | | | | | |
| 3569 | 020652 | 016666 | 000002 | 000006 | 67\$: | | | | | |

```

3570 020660 012716 020666      MOV      #68$, (SP)          ;; REPLACE OLD PC WITH NEW
3571 020664 000002              RTI                          ;; RESTORE PSW
3572 020666 012637 000034      68$:  MOV      (SP)+, 34      ;; RESTORE JLD TRAP VECTOR
3573 020672 012662 000002      MOV      (SP)+, 2(ADINT)
3574              ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3575 020676 005046              CLR      -(SP)
3576 020700 013746 000034      MOV      34, -(SP)          ;; SAVE CURRENT TRAP VECTOR
3577 020704 012737 020714 000034  MOV      #69$, 34          ;; SETUP NEW TRAP VECTOT
3578 020712 104400              TRAP
3579 020714 016666 000002 000006 69$:  MOV      2(SP), 6(SP)
3580 02J722 012716 020730      MOV      #70$, (SP)        ;; REPLACE OLD PC WITH NEW
3581 020726 000002              RTI                          ;; RESTORE PSW
3582 020730 012637 000034      70$:  MOV      (SP)+, 34      ;; RESTORE OLD TRAP VECTOR
3583 020734 012637 001214      MOV      (SP)+, $TMP6
3584 020740 042737 000340 001214  BIC      #340, $TMP6
3585 020746 013746 001214      MOV      $TMP6, -(SP)      ;; PUT NEW PS ON STACK
3586 020752 012746 020760      MOV      #71$, -(SP)      ;; PUT NEW PC ON STACK
3587 020756 000002              RTI                          ;; POP NEW PC AND PS
3588 020760
3589 020760 012714 177701      MOV      #-77, @CDC        ; SET UP COLUMN COUNT
3590 020764 012715 045442      MOV      #BUFBEG, @CDA     ; SET UP BUS ADDRESS
3591 020770 005213              INC      @CDS              ; START READING
3592 020772 000777              BR
3593              ; WAIT FOR AN INTERRUPT
3594 020774 022626              TINTER: CMP      (SP)+, (SP)+ ; RESTORE THE STACK
3595 020776 022713 000300      CMP      #000300, @CDS    ; CHECK THE CARD READER STATUS
3596 021002 001401              BEQ      15               ; BRANCH IF OK
3597 021004 104004              ERROR +4                 ; CARD READER STATUS ERROR
3598
3599 021006
3600
3601
3602
3603 021006 000004      15:
3604              ; *****
3605              ; *TEST 40 TEST PICK CHECK ERROR
3606              ; *****
3607      TST40: SCOPE
3608              ; A PICK CHECK ERROR SHOULD SET BIT 15, BIT 14, AND BIT 12
3609              ; THIS ERROR OCCURS WHEN THE FEED MECHANISM FAILS TO DELIVER A CARD TO
3610              ; THE READ STATION WITHIN 400 MS.
3611              ; CAN ALSO BE FORCED BY FOLDING A CARD IN HALF
3612              ; AND PLACING IT INTO CARD READER 'INPUT HOPPER'
3613      JSR      PC, INIT
3614      TYPE,   MSG37          ; TST40 PICK CHECK TEST
3615      TYPE,   MSG5          ; "REMOVE ALL CARDS FROM THE INPUT HOPPER"
3616      TYPE,   MSG2          ; "WHEN HIT 'CONTINUE' ON THE CONSOLE"
3617      TYPE,   MSG8          ; "HOLD DOWN THE SWITCH UNDER THE CAP
3618      TYPE,   MSG1          ; OF THE INPUT HOPPER"
3619      TYPE,   CRLF-3        ; "PRESS CARD READER 'RESET'"
3620      HALT
3621      BIT      #10000, @CDS  ; CHECK FOR OFF-LINE
3622      BNE     15            ; BRANCH IF SET
3623      ERROR +43            ; OFF LINE NOT SET AFTER "CONTINUE"
3624
3625 021056 032713 J00010      15:  BIT      #10, @CDS      ; CHECK FOR "TRANSITION TO ON LINE"
3626 021062 001775      BEQ      15              ; WAIT FOR IT

```


| | | | | | | | | |
|------|--------|--------|--------|--------|------|-----------|----------------|----------------------------------------------------------------|
| 3624 | 021064 | 022713 | 140210 | | | CMP | #140210, 2CDS | ;CHECK FOR CORRECT STATUS BITS |
| 3625 | 021070 | 001401 | | | | BEG | 25 | ;BRANCH IF OK |
| 3626 | 021072 | 104004 | | | | ERROR +" | | ;STATUS NOT EQUAL TO 140210 |
| 3627 | | | | | | | | |
| 3628 | 021074 | 012714 | 17770: | 25: | | MOV | #-77, 2CDC | ;SET UP COLUMN COUNT |
| 3629 | 021100 | 012715 | 045442 | | | MOV | #BUFBEQ, 2CDA | ;SET UP BUS ADDRESS |
| 3630 | 021104 | 005213 | | | | INC | 2CDS | ;READ |
| 3631 | 021106 | 105713 | | 35: | | TSTC | 2CDS | ;CHECK CONTROLLER READY |
| 3632 | 021110 | 100376 | | | | BPL | 35 | ;WAIT FOR CONTROLLER READY |
| 3633 | 021112 | 032713 | 010000 | | | BIT | #10000, 2CDS | ;CHECK BIT 12 |
| 3634 | 021116 | 001001 | | | | BNE | 45 | ;BRANCH IF SET |
| 3635 | 021120 | 104043 | | | | ERROR +43 | | ;OFF-LINE (BIT 12) WASN'T SET |
| 3636 | | | | | | | | |
| 3637 | 021122 | 005713 | | 45: | | TST | 2CDS | ;CHECK SPECIAL CONDITION BIT |
| 3638 | 021124 | 100401 | | | | BMI | 55 | ;BRANCH IF SET |
| 3639 | 021126 | 104026 | | | | ERROR +26 | | ;SPECIAL CONDITION NOT SET |
| 3640 | | | | | | | | |
| 3641 | 021130 | 032713 | 040000 | 55: | | BIT | #40000, 2CDS | ;CHECK FOR CARD READER ERROR |
| 3642 | 021134 | 001001 | | | | BNE | 65 | ;BRANCH IF SET |
| 3643 | 021136 | 104035 | | | | ERROR +35 | | ;CARD READER ERROR BIT 14 NOT SET |
| 3644 | | | | | | | | |
| 3645 | 021140 | 005777 | 160074 | 65: | | TST | 2CDD8 | ;TEST BIT 15 OF STATUS REGISTER #2 |
| 3646 | 021144 | 100005 | | | | BPL | 75 | ;BRANCH IF NOT SET INDICATING |
| 3647 | | | | | | | | ;OLD CD11 CONTROLLER |
| 3648 | 021146 | 032777 | 020000 | 160064 | | BIT | #20000, 2CDD8 | ;IS PICK CHECK INDICATOR SET? |
| 3649 | 021154 | 001001 | | | | BNE | 75 | ;BRANCH IF SET |
| 3650 | 021156 | 104045 | | | | ERROR +45 | | ;PICK CHECK BIT 13 NOT SET |
| 3651 | | | | | | | | |
| 3652 | 021160 | 031327 | 027577 | 75: | | BIT | 2CDS, #027577 | ;CHECK FOR EXTRA BITS |
| 3653 | 021164 | 001401 | | | | BEG | 105 | ;BRANCH IF OK |
| 3654 | 021166 | 104015 | | | | ERROR +15 | | ;STATUS WORD ERROR |
| 3655 | | | | | | | | |
| 3656 | 021170 | 012712 | 021424 | 105: | | MOV | #TINTF, 2ADINT | ;LOAD RETURN POINTER |
| 3657 | | | | | | | | ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340, PS" |
| 3658 | 021174 | 005046 | | | | CLR | -(SP) | |
| 3659 | 021176 | 013746 | 000034 | | | MOV | 34, -(SP) | ;SAVE CURRENT TRAP VECTOR |
| 3660 | 021202 | 012737 | 021212 | 000034 | | MOV | #645, 34 | ;SETUP NEW TRAP VECTOT |
| 3661 | 021210 | 104400 | | | | TRAP | | ;PUSH OLD PSW AN PCON STACK |
| 3662 | 021212 | 016666 | 000002 | 000006 | 645: | MOV | 2(SP), 6(SP) | |
| 3663 | 021220 | 012716 | 021226 | | | MOV | #655, (SP) | ;REPLACE OLD PC WITH NEW |
| 3664 | 021224 | 000002 | | | | RTI | | ;RESTORE PSW |
| 3665 | 021226 | 012637 | 000034 | 655: | | MOV | (SP)+, 34 | ;RESTORE OLD TRAP VECTOR |
| 3666 | 021232 | 012637 | 001214 | | | MOV | (SP)+, \$TMP6 | |
| 3667 | 021236 | 052737 | 000340 | 001214 | | BIS | #340, \$TMP6 | |
| 3668 | 021244 | 013746 | 001214 | | | MOV | \$TMP6, -(SP) | ;PUT NEW PS ON STACK |
| 3669 | 021250 | 012746 | 021256 | | | MOV | #665, -(SP) | ;PUT NEW PC ON STACK |
| 3670 | 021254 | 000002 | | | | RTI | | ;POP NEW PC AND PS |
| 3671 | 021256 | | | 665: | | | | |
| 3672 | | | | | | | | ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, 2(ADINT)" |
| 3673 | 021256 | 005046 | | | | CLR | -(SP) | |
| 3674 | 021260 | 013746 | 000034 | | | MOV | 34, -(SP) | ;SAVE CURRENT TRAP VECTOR |
| 3675 | 021264 | 012737 | 021274 | 000034 | | MOV | #675, 34 | ;SETUP NEW TRAP VECTOT |
| 3676 | 021272 | 104400 | | | | TRAP | | ;PUSH OLD PSW AN PCON STACK |
| 3677 | 021274 | 016666 | 000002 | 000006 | 675: | MOV | 2(SP), 6(SP) | |

| | | | | | | | |
|------|--------|--------|--------|--------|-------|-----------------------------------------------------------------|----------------------------------------|
| 3678 | 021302 | 012716 | 021310 | | | MOV #68\$, (SP) | ;; REPLACE OLD PC WITH NEW |
| 3679 | 021306 | 000002 | | | | RTI | ;; RESTORE PSW |
| 3680 | 021310 | 012637 | 000034 | 68\$: | | MOV (SP)+, 34 | ;; RESTORE OLD TRAP VECTOR |
| 3681 | 021314 | 012662 | 000002 | | | MOV (SP)+, 2(ADINT) | |
| 3682 | | | | | | ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340, PS" | |
| 3683 | 021320 | 005046 | | | | CLR -(SP) | ;; |
| 3684 | 021322 | 013746 | 000034 | | | MOV 34, -(SP) | ;; SAVE CURRENT TRAP VECTOR |
| 3685 | 021326 | 012737 | 021336 | 000034 | | MOV #69\$, 34 | ;; SETUP NEW TRAP VECTOT |
| 3686 | 021334 | 104400 | | | | TRAP | ;; PUSH OLD PSW AN PCON STACK |
| 3687 | 021336 | 016666 | 000002 | 000006 | 69\$: | MOV 2(SP), 6(SP) | ;; |
| 3688 | 021344 | 012716 | 021352 | | | MOV #70\$, (SP) | ;; REPLACE OLD PC WITH NEW |
| 3689 | 021350 | 000002 | | | | RTI | ;; RESTORE PSW |
| 3690 | 021352 | 012637 | 000034 | 70\$: | | MOV (SP)+, 34 | ;; RESTORE OLD TRAP VECTOR |
| 3691 | 021356 | 012637 | 001214 | | | MOV (SP)+, \$TMP6 | |
| 3692 | 021362 | 012737 | 000340 | 001214 | | BIC #340, \$TMP6 | |
| 3693 | 021370 | 013746 | 001214 | | | MOV \$TMP6, -(SP) | ;; PUT NEW PS ON STACK |
| 3694 | 021374 | 012746 | 021402 | | | MOV #71\$, -(SP) | ;; PUT NEW PC ON STACK |
| 3695 | 021400 | 000002 | | | | RTI | ;; POP NEW PC AND PS |
| 3696 | 021402 | | | 71\$: | | | |
| 3697 | 021402 | 012713 | 000100 | | | MOV #100, JCD\$ | ;; SET INTERRUPT ENABLE |
| 3698 | 021406 | 104400 | 032045 | | | TYPE, MSG6 | ;; "RESTORE CARDS TO THE INPUT HOPPER" |
| 3699 | 021412 | 104400 | 031576 | | | TYPE, MSG1 | ;; "PRESS CARD READER 'RESET'" |
| 3700 | 021416 | 104400 | 030550 | | | TYPE, CRLF-3 | ;; MOVE MESSAGE UP ON TTY |
| 3701 | 021422 | 000777 | | | | BR . | ;; WAIT FOR THE INTERRUPT |
| 3702 | | | | | | | |
| 3703 | 021424 | 022626 | | TINTF: | | CMP (SP)+, (SP)+ | ;; RESTORE THE STACK |
| 3704 | 021426 | 012712 | 021654 | | | MOV #TINTFA, JADINT | ;; LOAD RETURN POINTER |
| 3705 | | | | | | ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340, PS" | |
| 3706 | 021432 | 005046 | | | | CLR -(SP) | ;; |
| 3707 | 021434 | 013746 | 000034 | | | MOV 34, -(SP) | ;; SAVE CURRENT TRAP VECTOR |
| 3708 | 021440 | 012737 | 021450 | 000034 | | MOV #64\$, 34 | ;; SETUP NEW TRAP VECTOT |
| 3709 | 021446 | 104400 | | | | TRAP | ;; PUSH OLD PSW AN PCON STACK |
| 3710 | 021450 | 016666 | 000002 | 000006 | 64\$: | MOV 2(SP), 6(SP) | ;; |
| 3711 | 021456 | 012716 | 021464 | | | MOV #65\$, (SP) | ;; REPLACE OLD PC WITH NEW |
| 3712 | 021462 | 000002 | | | | RTI | ;; RESTORE PSW |
| 3713 | 021464 | 012637 | 000034 | 65\$: | | MOV (SP)+, 34 | ;; RESTORE OLD TRAP VECTOR |
| 3714 | 021470 | 012637 | 001214 | | | MOV (SP)+, \$TMP6 | |
| 3715 | 021474 | 052737 | 000340 | 001214 | | BIS #340, \$TMP6 | |
| 3716 | 021502 | 013746 | 001214 | | | MOV \$TMP6, -(SP) | ;; PUT NEW PS ON STACK |
| 3717 | 021506 | 012746 | 021514 | | | MOV #66\$, -(SP) | ;; PUT NEW PC ON STACK |
| 3718 | 021512 | 000002 | | | | RTI | ;; POP NEW PC AND PS |
| 3719 | 021514 | | | 66\$: | | | |
| 3720 | | | | | | ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, 2(ADINT)" | |
| 3721 | 021514 | 005046 | | | | CLR -(SP) | ;; |
| 3722 | 021516 | 013746 | 000034 | | | MOV 34, -(SP) | ;; SAVE CURRENT TRAP VECTOR |
| 3723 | 021522 | 012737 | 021532 | 000034 | | MOV #67\$, 34 | ;; SETUP NEW TRAP VECTOT |
| 3724 | 021530 | 104400 | | | | TRAP | ;; PUSH OLD PSW AN PCON STACK |
| 3725 | 021532 | 016666 | 000002 | 000006 | 67\$: | MOV 2(SP), 6(SP) | ;; |
| 3726 | 021540 | 012716 | 021546 | | | MOV #68\$, (SP) | ;; REPLACE OLD PC WITH NEW |
| 3727 | 021544 | 000002 | | | | RTI | ;; RESTORE PSW |
| 3728 | 021546 | 012637 | 000034 | 68\$: | | MOV (SP)+, 34 | ;; RESTORE OLD TRAP VECTOR |
| 3729 | 021552 | 012662 | 000002 | | | MOV (SP)+, 2(ADINT) | |
| 3730 | | | | | | ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340, PS" | |
| 3731 | 021556 | 005046 | | | | CLR -(SP) | ;; |

```

3732 021560 013746 000034      MOV      34, -(SP)      ;; SAVE CURRENT TRAP VECTOR
3733 021564 012737 021574 000034      MOV      #695, 34      ;; SETUP NEW TRAP VECTOT
3734 021572 104400      TRAP                                ;; PUSH OLD PSW AN PCOM STACK
3735 021574 016666 000002 000006 695:      MOV      2(SP), 6(SP)  ;;
3736 021602 012716 021610      MOV      #705, (SP)    ;;
3737 021606 000002      RTI                                ;; RESTORE PSW
3738 021610 012637 000034 705:      MOV      (SP)+, 34      ;; RESTORE OLD TRAP VECTOR
3739 021614 012637 001214      MOV      (SP)+, $TMP6
3740 021620 042737 000340 001214      BIC      #340, $TMP6
3741 021626 013746 001214      MOV      $TMP6, -(SP)  ;; PUT NEW PS ON STACK
3742 021632 012746 021640      MOV      #715, -(SP)  ;; PUT NEW PC ON STACK
3743 021636 000002      RTI                                ;; POP NEW PC AND PS
3744 021640
3745 021640 012714 177701      MOV      #-77, 2CDC    ;; SET UP COLUMN COUNT
3746 021644 012715 045442      MOV      #BUFBEG, 2CDA ;; SET UP BUS ADDRESS
3747 021650 005213      INC      2CDS          ;; START READING
3748 021652 000777      BR      .             ;; WAIT FOR AN INTERRUPT
3749
3750 021654 022626      TINTFA: CMP      (SP)+, (SP)+  ;; RESTORE THE STACK
3751 021656 022713 000300      CMP      #000370, 2CDS ;; CHECK THE CARD READER STATUS
3752 021662 001401      BEQ      15           ;; BRANCH IF OK
3753 021664 104004      ERROR +4           ;; CARD READER STATUS ERROR
3754
3755 021666
3756
3757
3758
3759 021666 000004
3760
3761
3762
3763 021670 004737 025744      JSR      PC INIT
3764 021674 104400 036104      TYPE,   MSG38
3765 021700 104400 031706      TYPE,   MSG3
3766 021704 104400 032362      TYPE,   MSG9
3767
3768
3769
3770
3771
3772
3773
3774 021710 104400 031576      TYPE,   MSG1
3775 021714 104400 030550      TYPE,   CRLF-3
3776 021720 032713 010000      TLOPG:  BIT      #10000, 2CDS
3777 021724 001775      BEQ      TLOPG
3778 021726 032713 000010      TLOPGA: BIT      #10, 2CDS
3779 021732 001775      BEQ      TLOPGA
3780 021734 022713 100210      CMP      #100210, 2CDS
3781 021740 001401      BEQ      15
3782 021742 104004      ERROR +4           ;; STATUS NOT EQUAL TO 100210
3783
3784 021744 012714 177701      15:      MOV      #-77, 2CDC    ;; SET UP COLUMN COUNT
3785 021750 012715 045442      MOV      #BUFBEG, 2CDA ;; SET UP BUS ADDRESS

```

```

*5:
*****
*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

```

```

;TST41 STACK ERROR TEST
;"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 PHOTOCELL
;IN WHICH CASE, THE PHOTOCELL CAN BE
;BLOCKED BY TEARING OFF A PIECE OF CARD
;AND SLIPPING IT IN FRONT OF THE PHOTOCELL
;"PRESS CARD READER 'RESET'"
;MOVE MESSAGE UP ON TTY
;CHECK FOR OF LINE
;WAIT FOR OFF-LINE
;CHECK FOR "TRANSITION TO ON LINE"
;WAIT FOR IT
;CHECK FOR CORRECT STATUS BITS
;BRANCH IF OK
;STATUS NOT EQUAL TO 100210

```

```

3786 021754 005213
3787 021756 105713
3788 021760 100376
3789 021762 032713 010000
3790 021766 001001
3791 021770 104043
3792
3793 021772 005713
3794 021774 100401
3795 021776 104026
3796
3797 022000 032713 040000
3798 022004 001001
3799 022006 104035
3800
3801 022010 005777 157224
3802 022014 100005
3803
3804 022016 032777 010000 157214
3805 022024 001001
3806 022026 104046
3807
3808 022030 032713 027577
3809 022034 001401
3810 022036 104015
3811
3812 022040 012712 022270
3813
3814 022044 005046
3815 022046 013746 000034
3816 022052 012737 022062 000034
3817 022060 104400
3818 022062 016666 000002 000006
3819 022070 012716 022076
3820 022074 000002
3821 022076 012637 000034
3822 022102 012637 001214
3823 022106 052737 000340 001214
3824 022114 013746 001214
3825 022120 012746 022126
3826 022124 000002
3827 022126
3828
3829 022126 005046
3830 022130 013746 000034
3831 022134 012737 022144 000034
3832 022142 104400
3833 022144 016666 000002 000006
3834 022152 012716 022160
3835 022156 000002
3836 022160 012637 000034
3837 022164 012662 000002
3838
3839 022170 005046

TLOPGB: INC 2CDS ;READ
TSTB 2CDS ;CHECK CONTROLLER READY
BPL TLOPGB ;WAIT FOR CONTROLLER READY
BIT #10000,2CDS ;CHECK BIT12
BNE 15 ;BRANCH IF SET
ERROR +43 ;OFF-LINE (BIT 12) WASN'T SET

15: TST 2CDS ;CHECK SPECIAL CONDITION BIT
BMI 25 ;BRANCH IF SET
ERROR +26 ;SPECIAL CONDITION NOT SET

25: BIT #40000, 2CDS ;CHECK FOR CARD READER ERROR
BNE 35 ;BRANCH IF SET
ERROR +35 ;CARD READER ERROR BIT 14 NOT SET

35: TST 2CDD8 ;TEST BIT15 OF STATUS REGISTER #2
BPL 45 ;BRANCH IF NOT SET INDICATING
;OLD CD11 CONTROLLER
BIT #10000,2CDD8 ;IS STACK CHECK INDICATOR SET?
BNE 45 ;BRANCH IF SET
ERROR +46 ;STACK CHECK BIT12 NOT SET

45: BIT #027577,2CDS ;CHECK FOR EXTRA BITS
BEQ 55 ;BRANCH IF OK
ERROR +15 ;STATUS WORD ERROR

55: MOV #TINTG,2ADINT ;LOAD RETURN POINTER
;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
CLR -(SP) ;;
MOV 34,-(SP) ;;SAVE CURRENT TRAP VECTOR
MOV #645,34 ;;SETUP NEW TRAP VECTOR
TRAP ;;PUSH OLD PSW AN PCOM STACK
MOV 2(SP),6(SP) ;;
MOV #655,(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 #665,-(SP) ;;PUT NEW PC ON STACK
RTI ;;POP NEW PC AND PS

665: ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
CLR -(SP) ;;
MOV 34,-(SP) ;;SAVE CURRENT TRAP VECTOR
MOV #675,34 ;;SETUP NEW TRAP VECTOR
TRAP ;;PUSH OLD PSW AN PCOM STACK
MOV 2(SP),6(SP) ;;
MOV #685,(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) ;;

```

```

3840 022172 013746 000034      MOV      34, -(SP)      ;; SAVE CURRENT TRAP VECTOR
3841 022176 012737 022206 000034      MOV      #69$, 34      ;; SETUP NEW TRAP VECTOT
3842 022204 104400      TRAP                      ;; PUSH OLD PSW AN PCOM STACK
3843 022206 016666 000002 000006 69$:      MOV      2(SP), 6(SP)  ;;
3844 022214 012716 022222      MOV      #70$, (SP)    ;;
                        ;; REPLACE OLD PC WITH NEW
3845 022220 000002      RTI                      ;; RESTORE PSW
                        ;; RESTORE OLD TRAP VECTOR
3846 022222 012637 000034 70$:      MOV      (SP)+, 34
3847 022226 012637 001214      MOV      (SP)+, $TMP6
3848 022232 042737 000340 001214      BIC      #340, $TMP6
3849 022240 013746 001214      MOV      $TMP6, -(SP)  ;; PUT NEW PS ON STACK
3850 022244 012746 022252      MOV      #71$, -(SP)  ;; PUT NEW PC ON STACK
3851 022250 000002      RTI                      ;; POP NEW PC AND PS
3852 022252      71$:
3853 022252 012713 000100      MOV      #100, 3CDS    ;; SET INTERRUPT ENABLE
3854 022256 104400 031576      TYPE,   MSG1          ;; "PRESS CARD READER 'RESET'"
3855 022262 104400 030550      TYPE,   CRLF-3       ;; MOVE MESSAGE UP ON TTY
3856 022266 000777      BR      .              ;; WAIT FOR THE INTERRUPT
3857
3858 022270 022626      TINTG:  CMP      (SP)+, (SP)+  ;; RESTORE THE STACK
3859 022272 012712 022520      MOV      #TINTGA, 3ADINT ;; LOAD RETURN POINTER
3860      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340, PS"
3861 022276 005046      CLR      -(SP)
3862 022300 013746 000034      MOV      34, -(SP)    ;; SAVE CURRENT TRAP VECTOR
3863 022304 012737 022314 000034      MOV      #64$, 34      ;; SETUP NEW TRAP VECTOT
3864 022312 104400      TRAP                      ;; PUSH OLD PSW AN PCOM STACK
3865 022314 016666 000002 000006 64$:      MOV      2(SP), 6(SP)  ;;
3866 022322 012716 022330      MOV      #65$, (SP)    ;;
                        ;; REPLACE OLD PC WITH NEW
3867 022326 000002      RTI                      ;; RESTORE PSW
                        ;; RESTORE OLD TRAP VECTOR
3868 022330 012637 000034 65$:      MOV      (SP)+, 34
3869 022334 012637 001214      MOV      (SP)+, $TMP6
3870 022340 052737 000340 001214      BIS      #340, $TMP6
3871 022346 013746 001214      MOV      $TMP6, -(SP)  ;; PUT NEW PS ON STACK
3872 022352 012746 022360      MOV      #66$, -(SP)  ;; PUT NEW PC ON STACK
3873 022356 000002      RTI                      ;; POP NEW PC AND PS
3874 022360      66$:
3875      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, 2(ADINT)"
3876 022360 005046      CLR      -(SP)
3877 022362 013746 000034      MOV      34, -(SP)    ;; SAVE CURRENT TRAP VECTOR
3878 022366 012737 022376 000034      MOV      #67$, 34      ;; SETUP NEW TRAP VECTOT
3879 022374 104400      TRAP                      ;; PUSH OLD PSW AN PCOM STACK
3880 022376 016666 000002 000006 67$:      MOV      2(SP), 6(SP)  ;;
3881 022404 012716 022412      MOV      #68$, (SP)    ;;
                        ;; REPLACE OLD PC WITH NEW
3882 022410 000002      RTI                      ;; RESTORE PSW
                        ;; RESTORE OLD TRAP VECTOR
3883 022412 012637 000034 68$:      MOV      (SP)+, 34
3884 022416 012662 000002      MOV      (SP)+, 2(ADINT)
3885      : THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340, PS"
3886 022422 005046      CLR      -(SP)
3887 022424 013746 000034      MOV      34, -(SP)    ;; SAVE CURRENT TRAP VECTOR
3888 022430 012737 022440 000034      MOV      #69$, 34      ;; SETUP NEW TRAP VECTOT
3889 022436 104400      TRAP                      ;; PUSH OLD PSW AN PCOM STACK
3890 022440 016666 000002 000006 69$:      MOV      2(SP), 6(SP)  ;;
3891 022446 012716 022454      MOV      #70$, (SP)    ;;
                        ;; REPLACE OLD PC WITH NEW
3892 022452 000002      RTI                      ;; RESTORE PSW
                        ;; RESTORE OLD TRAP VECTOR
3893 022454 012637 000034 70$:      MOV      (SP)+, 34

```

```

3894 022460 012637 001214      MOV      (SP)+, $TMP6
3895 022464 042737 000340 001214      BIC      #340, $TMP6
3896 022472 013.46 001214      MOV      $TMP6, -(SP)      ;: PUT NEW PS ON STACK
3897 022476 012746 022504      MOV      #715, -(SP)      ;: PUT NEW PC ON STACK
3898 022502 000002      RTI      ;: POP NEW PC AND PS
3899 022504      715:
3900 022504 012714 177701      MOV      #-77, @CDC      ;: SET UP COLUMN COUNT
3901 022510 012715 045442      MOV      #BUFBEG, @CDA    ;: SET UP BUS ADDRESS
3902 022514 005213      INC      @CDS      ;: START READING
3903 022516 000777      BR      .      ;: WAIT FOR AN INTERRUPT
3904
3905 022520 022626      TINTGA: CMP      (SP)+, (SP)+ ;: RESTORE THE STACK
3906 022522 022713 000300      CMP      #000300, @CDS    ;: CHECK THE CARD READER STATUS
3907 022526 001401      BEQ      1$      ;: BRANCH IF OK
3908 022530 104004      ERROR +4      ;: CARD READER STATUS ERROR
3909
3910 022532      1$:
3911      ;: ON M-1000/M-200 BIT 13 IS ALWAYS CLEARED
3912      ;: ON M-1200 IF END OF FILE BUTTON IS PRESSED WITH INPUT
3913      ;: HOPPER LOADED THEN WHEN INPUT HOPPER BECOMES EMPTY
3914      ;: HOPPER CHECK INDICATOR LIGHT COMES ON AND BITS
3915      ;: 13 14 AND 15 ARE SET
3916
3917
3918 022532 005737 001266      TST      CD1000      ;: IS READER M1000/M200?
3919 022536 001402      BEQ      TSTM12      ;: BRANCH IF READER IS M-1200
3920 022540 000137 023256      JMP      TSTM10      ;: OUT OF THIS TEST IF M1000/M200
3921
3922 022544      TSTM12:
3923      ;: *****
3924      ;: *TEST 42 TEST 'END OF FILE' AND HOPPER CHECK
3925      ;: *****
3926 022544 000004      TST42: SCOPE
3927 022546 004737 025744      JSR      PC, INIT
3928 022552 104400 036140      TYPE,   MSG39      ;: TST42 EOF & HOPPER CHECK
3929 022556 104400 033506      TYPE,   MSG20      ;: "PUT ANY TWO CARDS IN INPUT HOPPER"
3930 022562 104400 031576      TYPE,   MSG1      ;: "PRESS CARD READER 'RESET'"
3931 022566 104400 031632      TYPE,   MSG2      ;: "THEN HIT 'CONTINUE' ON THE CONSOLE"
3932 022572 104400 030550      TYPE,   CRLF-3     ;: MOVE MESSAGE UP ON TTY
3933 022576 000000      HALT
3934
3935 022600 032713 000010      1$: BIT      #10, @CDS ;: CHECK FOR TRANSITION TO ON LINE
3936 022604 001775      BEQ      1$      ;: WAIT FOR IT
3937 022606 104400 033552      TYPE,   MSG21      ;: "PRESS END OF FILE BUTTON"
3938 022612 104400 031632      TYPE,   MSG2      ;: "THEN HIT 'CONTINUE' ON THE CONSOLE"
3939 022616 104400 030550      TYPE,   CRLF-3     ;: MOVE MESSAGE UP ON TTY
3940 022622 004737 025744      JSR      PC, INIT
3941 022626 000000      HALT
3942
3943
3944 022630 032713 020000      BIT      #20000, @CDS ;: CHECK BIT 13
3945 022634 001401      BEQ      2$      ;: BRANCH IF NOT SET
3946 022636 104047      ERROR +47      ;: EOF SET FROM BEGINNING
3947

```

```

3948
3949 022640 032713 040000      2$: BIT      #40000, @CDS      ;CHECK BIT 14
3950 022644 001401              BEQ      3$                ;BRANCH IF NOT SET
3951 022646 104050              ERROR +50                ;READER CHECK ERROR SET FROM BEGINNING
3952
3953
3954 022650 032713 000004      3$: BIT      #4,      @CDS      ;CHECK BIT 2
3955 022654 001401              BEQ      4$                ;BRANCH IF NOT SET
3956 022656 104051              ERROR +51                ;HOPPER CHECK SET FROM BEGINNING
3957
3958 022660 005713              4$: TST      @CDS          ;CHECK ERROR BIT
3959 022662 100001              BPL      5$                ;BRANCH IF NOT SET
3960 022664 104014              ERROR +14                ;ERROR SET FROM BEGINNING
3961
3962
3963
3964
3965 022666 012712 023120      5$: MOV      #TINTI, @ADINT ;LOAD RETURN POINTER
3966 022672              SECN:
3967              ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS"
3968 022672 005046              CLR      -(SP)            ;;
3969 022674 013746 000034              MOV      34, -(SP)        ;;SAVE CURRENT TRAP VECTOR
3970 022700 012737 022710 000034              MOV      #64$, 34         ;;SETUP NEW TRAP VECTOT
3971 022706 104400              TRAP                    ;;PUSH OLD PSW AN PCON STACK
3972 022710 016666 000002 000006 64$: MOV      2(SP), 6(SP)      ;;
3973 022716 012716 022724              MOV      #65$, (SP)      ;;REPLACE OLD PC WITH NEW
3974 022722 000002              RTI                    ;;RESTORE PSW
3975 022724 012637 000034 65$: MOV      (SP)+, 34        ;;RESTORE OLD TRAP VECTOR
3976 022730 012637 001214              MOV      (SP)+, $TMP6
3977 022734 052737 000340 001214              BIS      #340, $TMP6
3978 022742 013746 001214              MOV      $TMP6, -(SP)    ;;PUT NEW PS ON STACK
3979 022746 012746 022754              MOV      #66$, -(SP)    ;;PUT NEW PC ON STACK
3980 022752 000002              RTI                    ;;POP NEW PC AND PS
3981 022754              66$:
3982              ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS, 2(ADINT)"
3983 022754 005046              CLR      -(SP)            ;;
3984 022756 013746 000034              MOV      34, -(SP)        ;;SAVE CURRENT TRAP VECTOR
3985 022762 012737 022772 000034              MOV      #67$, 34         ;;SETUP NEW TRAP VECTOT
3986 022770 104400              TRAP                    ;;PUSH OLD PSW AN PCON STACK
3987 022772 016666 000002 000006 67$: MOV      2(SP), 6(SP)      ;;
3988 023000 012716 023006              MOV      #68$, (SP)      ;;REPLACE OLD PC WITH NEW
3989 023004 000002              RTI                    ;;RESTORE PSW
3990 023006 012637 000034 68$: MOV      (SP)+, 34        ;;RESTORE OLD TRAP VECTOR
3991 023012 012662 000002              MOV      (SP)+, 2(ADINT)
3992              ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
3993 023016 005046              CLR      -(SP)            ;;
3994 023020 013746 000034              MOV      34, -(SP)        ;;SAVE CURRENT TRAP VECTOR
3995 023024 012737 023034 000034              MOV      #69$, 34         ;;SETUP NEW TRAP VECTOT
3996 023032 104400              TRAP                    ;;PUSH OLD PSW AN PCON STACK
3997 023034 016666 000002 000006 69$: MOV      2(SP), 6(SP)      ;;
3998 023042 012716 023050              MOV      #70$, (SP)      ;;REPLACE OLD PC WITH NEW
3999 023046 000002              RTI                    ;;RESTORE PSW
4000 023050 012637 000034 70$: MOV      (SP)+, 34        ;;RESTORE OLD TRAP VECTOR
4001 023054 012637 001214              MOV      (SP)+, $TMP6

```

| | | | | | | | |
|------|--------|--------|--------|--------|---------|---------------------|---------------------------------------------|
| 4002 | 023063 | 042737 | 000340 | 001214 | BIC | #340, \$TMP6 | |
| 4003 | 023066 | 013746 | 001214 | | MOV | \$TMP6, -(SP) | ;; PUT NEW PS ON STACK |
| 4004 | 023072 | 012746 | 023100 | | MOV | #715, -(SP) | ;; PUT NEW PC ON STACK |
| 4005 | 023076 | 000002 | | | RTI | | ;; POP NEW PC AND PS |
| 4006 | 023100 | | | | 71\$: | | |
| 4007 | 023100 | 012713 | 000100 | | MOV | #100, @CDS | ; SET INTERRUPT ENABLE |
| 4008 | 023104 | 012714 | 177701 | | MOV | #-77, @CDC | ; SET UP COLUMN COUNT |
| 4009 | 023110 | 012715 | 045442 | | MOV | #BUFBEG, @CDA | ; SET UP BUS ADDRESS |
| 4010 | 023114 | 005213 | | | INC | @CDS | ; START READER |
| 4011 | 023116 | 000777 | | | BR | . | ; WAIT FOR AN INTERRUPT |
| 4012 | | | | | | | |
| 4013 | | | | | | | |
| 4014 | 023120 | 022626 | | | TINTI: | CMP (SP)+, (SP)+ | ; RESTORE THE STACK |
| 4015 | | | | | | | |
| 4016 | 023122 | 032713 | 020000 | | BIT | #20000, @CDS | ; CHECK BIT 13 |
| 4017 | 023126 | 001401 | | | BEQ | 1\$ | ; BRANCH IF NOT SET |
| 4018 | 023130 | 104047 | | | ERROR | +47 | ; EOF SET AT END OF ONE CARD |
| 4019 | | | | | | | |
| 4020 | 023132 | 032713 | 040000 | | 1\$: | BIT #40000, @CDS | ; CHECK BIT 14 |
| 4021 | 023136 | 001401 | | | BEQ | 2\$ | ; BRANCH IF NOT SET |
| 4022 | 023140 | 104050 | | | ERROR | +50 | ; READER CHECK ERROR SET AT END OF ONE CARD |
| 4023 | | | | | | | |
| 4024 | 023142 | 005713 | | | 2\$: | TST @CDS | ; CHECK ERROR BIT |
| 4025 | 023144 | 100001 | | | BPL | 3\$ | ; BRANCH IF NOT SET |
| 4026 | 023146 | 104014 | | | ERROR | +14 | ; ERROR SET AT END OF ONE CARD |
| 4027 | | | | | | | |
| 4028 | 023150 | 012712 | 023156 | | 3\$: | MOV #TINTIA, @ADINT | ; LOAD RETURN POINTER |
| 4029 | 023154 | 000646 | | | BR | SECN | ; READ SECOND CARD |
| 4030 | 023156 | 022626 | | | TINTIA: | CMP (SP)+, (SP)+ | ; RESTORE THE STACK |
| 4031 | | | | | | | |
| 4032 | 023160 | 032713 | 020000 | | BIT | #20000, @CDS | ; CHECK BIT 13 |
| 4033 | 023164 | 001001 | | | BNE | 1\$ | ; BRANCH IF SET |
| 4034 | 023166 | 104052 | | | ERROR | +52 | ; EOF NOT SET AT END OF FILE |
| 4035 | | | | | | | |
| 4036 | 023170 | 032713 | 040000 | | 1\$: | BIT #40000, @CDS | ; CHECK BIT 14 |
| 4037 | 023174 | 001001 | | | BNE | 2\$ | ; BRANCH IF SET |
| 4038 | 023176 | 104053 | | | ERROR | +53 | ; READER CHECK NOT SET AT END OF FILE |
| 4039 | | | | | | | |
| 4040 | 023200 | 032713 | 000004 | | 2\$: | BIT #4, @CDS | ; CHECK BIT 2 |
| 4041 | 023204 | 001001 | | | BNE | 3\$ | ; BRANCH IF SET |
| 4042 | 023206 | 104054 | | | ERROR | +54 | ; HOPPER CHECK NOT SET WHEN HOPPER EMPTY |
| 4043 | | | | | | | |
| 4044 | | | | | | | |
| 4045 | 023210 | 005713 | | | 3\$: | TST @CDS | ; CHECK ERROR BIT |
| 4046 | 023212 | 100401 | | | BMI | 4\$ | ; BRANCH IF SET |
| 4047 | 023214 | 104026 | | | ERROR | +26 | ; ERROR BIT NOT SET AT END OF FILE |
| 4048 | | | | | | | |
| 4049 | 023216 | 104400 | 032045 | | 4\$: | TYPE, MSG6 | ; "RESTORE CARDS TO THE INPUT HOPPER" |
| 4050 | 023222 | 104400 | 031576 | | TYPE, | MSG1 | ; "PRESS CARD READER 'RESET'" |
| 4051 | 023226 | 104400 | 031632 | | TYPE, | MSG2 | ; "THEN HIT CONTINUE ON THE CONSOLE" |
| 4052 | 023232 | 104400 | 030550 | | TYPE, | CRLF-3 | ; MOVE MESSAGE UP ON TTY |
| 4053 | 023236 | 000000 | | | HALT | | |
| 4054 | | | | | | | |
| 4055 | 023240 | 032713 | 000010 | | 5\$: | BIT #10, @CDS | ; CHECK TRANSITION TO ON LINE |

MAINDEC - 11 - DZCDB-B
DZCDB.P11 T42

MACY11 27(654) 1-JUL-77 08:39 PAGE 79
TEST 'END OF FILE' AND HOPPER CHECK

SEQ 0151

```

4056 023244 001775          BEQ      5$          ;WAIT FOR IT
4057
4058
4059 023246 032713 020000    BIT      #20000, @CDS ;CHECK BIT 13
4060 023252 001401          BEQ      TSTM10      ;BRANCH IF NOT SET
4061 023254 104055          ERROR +55          ;EOF DIDN'T CLEAR BY TRANSITION TO ON LINE
4062
4063 023256          TSTM10:
4064          ;*****
4065          ;*TEST 43      TEST READ CHECK ERROR
4066          ;*****
4067 023256 000004    TST43: SCOPE
4068          ;A READ CHECK ERROR SHOULD SET BIT 15, BIT 14, AND BIT 12
4069          ;THIS ERROR OCCURS WHEN THE READ ELECTRONICS IN THE CARD
4070          ;READER DISAGREES WITH THE NORMAL UNPUNCHED AREA OF THE CARD
4071 023260 004737 025744    JSR      PC, INIT
4072 023264 104400 036174    TYPE,   MSG40
4073 023270 104400 032706    TYPE,   MSG12
4074
4075 023274 104400 031576    TYPE,   MSG1
4076 023300 104400 030550    TYPE,   CRLF-3
4077 023304 032713 010000    TLOPH:  BIT      #10000, @CDS
4078 023310 001775          BEQ      TLOPH
4079 023312 032713 000010    TLOPHA: BIT      #10, @CDS
4080 023316 001775          BEQ      TLOPHA
4081 023320 022713 140210    CMP      #140210, @CDS
4082 023324 001401          BEQ      1$
4083 023326 104004          ERROR +4
4084
4085 023330 012714 177701    1$:     MOV      #-77, @CDC ;SET UP COLUMN COUNT
4086 023334 012715 045442    MOV      #BUFBEG, @CDA ;SET UP BUS ADDRESS
4087 023340 005213          INC      @CDS ;READ
4088 023342 105713          TLOPHB: TSTB    @CDS ;CHECK CONTROLLER READY
4089 023344 100376          @PL     TLOPHB ;WAIT FOR CONTROLLER READY
4090 023346 032713 010000    BIT      #10000, @CDS ;CHECK BIT12
4091 023352 001001          BNE     1$ ;BRANCH IF SET
4092 023354 104043          ERROR +43 ;OFF-LINE (BIT 12) WASN'T SET
4093
4094 023356 005713          1$:     TST      @CDS ;CHECK SPECIAL CONDITION BIT
4095 023360 100401          BMI     2$ ;BRANCH IF SET
4096 023362 104026          ERROR +26 ;SPECIAL CONDITION NOT SET
4097
4098 023364 032713 040000    2$:     BIT      #40000, @CDS ;CHECK FOR CARD READER ERROR
4099 023370 001001          BNE     3$ ;BRANCH IF SET
4100 023372 104053          ERROR +53 ;CARD READER ERROR BIT 14 NOT SET
4101
4102 023374 005777 155640    3$:     TST      @CDDB ;TEST BIT15 OF STATUS REGISTER #2
4103 023400 100005          EPL     4$ ;BRANCH IF NOT SET INDICATING
4104          ;OLD C011 CONTROLLER
4105 023402 032777 040000 155630    BIT      #40000, @CDDB ;IS READ CHECK INDICATOR SET?
4106 023410 001001          BNE     4$ ;BRANCH IF SET
4107 023412 104056          ERROR +56 ;READ CHECK BIT14 NOT SET
4108
4109 023414 032713 027577    4$:     BIT      #027577, @CDS ;CHECK FOR EXTRA BITS

```

| | | | | | | | | | |
|------|--------|--------|--------|--------|---------|---------------|--------------|--|-----------------------------------------------------------|
| 4110 | 023420 | 001401 | | | BEG | AGAIN | | | ;BRANCH IF OK |
| 4111 | 023422 | 104015 | | | ERROR | +15 | | | ;STATUS WORD ERROR |
| 4112 | 023424 | 104400 | 035373 | | TYPE, | MSG30 | | | ;IS CARD READER RS-1200 |
| 4113 | 023430 | 105777 | 155506 | | WAITA: | TSTB | STKS | | ;TEST TTY FOR READY |
| 4114 | 023434 | 100375 | | | BPL | WAITA | | | ;IF NOT READY WAIT |
| 4115 | 023436 | 117737 | 155502 | 001202 | MOVB | STKB,STMP1 | | | ;MOVE CHARACTER IN |
| 4116 | 023444 | 105777 | 155476 | | WAITB: | TSTB | STPS | | ;TEST TTY FOR DONE |
| 4117 | 023450 | 100375 | | | BPL | WAITB | | | ;WAIT IF NOT READY |
| 4118 | 023452 | 113777 | 001202 | 155470 | MOVB | STMP1,STPB | | | ;ECHO CHARACTER |
| 4119 | 023460 | 042737 | 177600 | 001202 | BIC | #177600,STMP1 | | | ;CLEAN JUNK OFF BYTE |
| 4120 | 023466 | 104400 | 030550 | | TYPE, | CRLF-3 | | | |
| 4121 | 023472 | 123727 | 001202 | 000116 | CMPB | STMP1,#116 | | | ;TEST FOR N |
| 4122 | 023500 | 001452 | | | BEG | BYPASS | | | ;SKIP TEST IF NO |
| 4123 | 023502 | 123727 | 001202 | 000131 | CMPB | STMP1,#131 | | | ;TEST FOR Y |
| 4124 | 023510 | 001403 | | | BEG | SS | | | ;BRANCH IF YES |
| 4125 | 023512 | 104400 | 032570 | | TYPE, | MSG10 | | | ;WRONG RESPONSE TRY AGAIN |
| 4126 | 023516 | 000742 | | | BR | AGAIN | | | ;CHARACTER WAS NOT Y OR N |
| 4127 | | | | | | | | | ;REPEAT QUESTION AGAIN |
| 4128 | | | | | | | | | |
| 4129 | 023520 | 104400 | 036226 | | SS: | TYPE, | MSG41 | | ;TST43A MIS-REGISTERED CARD TEST |
| 4130 | 023524 | 104400 | 034745 | | TYPE, | MSG26 | | | ;TEST FOR VISUAL VERIFICATION OF |
| 4131 | | | | | | | | | ;READ CHECK LIGHT |
| 4132 | 023530 | 104400 | 030553 | | TYPE, | CRLF | | | |
| 4133 | 023534 | 104400 | 035064 | | TYPE, | MSG27 | | | ;PLACE CARD INTO HOPPER |
| 4134 | 023540 | 104400 | 031576 | | TYPE, | MSG1 | | | |
| 4135 | 023544 | 104400 | 030550 | | TYPE, | CRLF-3 | | | |
| 4136 | | | | | | | | | |
| 4137 | 023550 | 032713 | 010000 | | TLOPHC: | JIT | #10000,ACDS | | ;WAIT FOR OFF-LINE |
| 4138 | 023554 | 001775 | | | BEG | TLOPHC | | | ;WAIT FOR OFF-LINE |
| 4139 | 023556 | 032713 | 000010 | | TLOPHC: | BIT | #10,ACDS | | ;CHECK FOR TRANSITION TO ON LINE |
| 4140 | 023562 | 001775 | | | BEG | TLOPHD | | | |
| 4141 | 023564 | 022713 | 140210 | | CMP | #140210,ACDS | | | ;CHECK FOR CORRECT STATUS BITS |
| 4142 | 023570 | 001401 | | | BEG | IS | | | ;BRANCH IF OK |
| 4143 | 023572 | 104004 | | | ERROR | +4 | | | ;STATUS NOT EQUAL 140210 |
| 4144 | | | | | | | | | |
| 4145 | 023574 | 012714 | 177701 | | IS: | MOV | #-77,ACDC | | ;SET UP COLUMN COUNT |
| 4146 | 023600 | 012715 | 045442 | | MOV | #BUFBEQ,ACDA | | | ;SET UP BUS ADDRESS |
| 4147 | 023604 | 005213 | | | INC | ACDS | | | |
| 4148 | | | | | | | | | |
| 4149 | 023606 | 105713 | | | TLOPHE: | TSTB | ACDS | | |
| 4150 | 023610 | 100376 | | | BPL | TLOPHE | | | |
| 4151 | 023612 | 032713 | 010000 | | BIT | #10000,ACDS | | | |
| 4152 | 023616 | 001001 | | | BNE | IS | | | |
| 4153 | 023620 | 104043 | | | ERROR | +43 | | | |
| 4154 | | | | | | | | | |
| 4155 | 023622 | 104400 | 035236 | | IS: | TYPE,MSG28 | | | ;CHECK READ CHECK LIGHT |
| 4156 | 023626 | 012712 | 024064 | | BYPASS: | MOV | #TINTH,ADINT | | ;LOAD RETURN POINTER |
| 4157 | | | | | | | | | ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #340,PS" |
| 4158 | 023632 | 005046 | | | CLR | -(SP) | | | |
| 4159 | 023634 | 013746 | 000034 | | MOV | 34, -(SP) | | | ;SAVE CURRENT TRAP VECTOR |
| 4160 | 023640 | 012737 | 023650 | 000034 | MOV | #645,34 | | | ;SETUP NEW TRAP VECTOT |
| 4161 | 023646 | 104400 | | | TRAP | | | | ;PUSH OLD PSW AN PCON STACK |
| 4162 | 023650 | 016666 | 000002 | 000006 | 645: | MOV | 2(SP),6(SP) | | |
| 4163 | 023656 | 012716 | 023664 | | MOV | #655,(SP) | | | ;REPLACE OLD PC WITH NEW |

```

4164 023662 000002          RTI          ;;RESTORE PSW
4165 023664 012637 000034 65$: MOV      (SP)+,34          ;;RESTORE OLD TRAP VECTOR
4166 023670 012637 001214    MOV      (SP)+,$TMP6
4167 023674 052737 000340 001214    BIS      #340,$TMP6
4168 023702 013746 001214    MOV      $TMP6,-(SP)          ;;PUT NEW PS ON STACK
4169 023706 012746 023714    MOV      #66$,-(SP)          ;;PUT NEW PC ON STACK
4170 023712 000002          RTI          ;;POP NEW PC AND PS
4171 023714          66$:
4172          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "MOV PS,2(ADINT)"
4173 023714 005046          CLR      -(SP)          ;;
4174 023716 013746 000034    MOV      34,-(SP)          ;;SAVE CURRENT TRAP VECTOR
4175 023722 012737 023732 000034    MOV      #67$,34          ;;SETUP NEW TRAP VECTOR
4176 023730 104400          TRAP          ;;PUSH OLD PSW AN PCON STACK
4177 023732 016666 000002 000006 67$: MOV      2(SP),6(SP)          ;;
4178 023740 012716 023746    MOV      #68$, (SP)          ;;REPLACE OLD PC WITH NEW
4179 023744 000002          RTI          ;;RESTORE PSW
4180 023746 012637 000034 68$: MOV      (SP)+,34          ;;RESTORE OLD TRAP VECTOR
4181 023752 012652 000002    MOV      (SP)+,2(ADINT)
4182          ;THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #340,PS"
4183 023756 005046          CLR      -(SP)          ;;
4184 023760 013746 000034    MOV      34,-(SP)          ;;SAVE CURRENT TRAP VECTOR
4185 023764 012737 023774 000034    MOV      #69$,34          ;;SETUP NEW TRAP VECTOR
4186 023772 104400          TRAP          ;;PUSH OLD PSW AN PCON STACK
4187 023774 016666 000002 000006 69$: MOV      2(SP),6(SP)          ;;
4188 024002 012716 024010    MOV      #70$, (SP)          ;;REPLACE OLD PC WITH NEW
4189 024006 000002          RTI          ;;RESTORE PSW
4190 024010 012637 000034 70$: MOV      (SP)+,34          ;;RESTORE OLD TRAP VECTOR
4191 024014 012637 001214    MOV      (SP)+,$TMP6
4192 024020 042737 000340 001214    BIC      #340,$TMP6
4193 024026 013746 001214    MOV      $TMP6,-(SP)          ;;PUT NEW PS ON STACK
4194 024032 012746 024040    MOV      #71$,-(SP)          ;;PUT NEW PC ON STACK
4195 024036 000002          RTI          ;;POP NEW PC AND PS
4196 024040          71$:
4197 024040 012713 000100    MOV      #100, 3CDS          ;;SET INTERRUPT ENABLE
4198 024044 104400 030045    TYPE,   MSG6          ;;"RESTORE CARDS TO THE INPUT HOPPER"
4199 024050 104400 035306    TYPE,   MSG29          ;;"PRESS CARD READER 'RESET'"
4200 024054 104400 030550    TYPE,   CRLF-3          ;;MOVE MESSAGE UP ON TTY
4201 024060 000777          BR          ;;WAIT FOR AN INTERRUPT
4202 024062 000000          HALT
4203 024064 022626          TINTH: CMP      (SP)+, (SP)+          ;;RESTORE THE STACK
4204 024066 000004          SCOPE
4205 024070 104400 001222    TYPE,   $BELL          ;;RING-A-DING
4206 024074 000137 016134    JMP      ER12CD          ;;LOOP BACK TO THE BEGINNING

```

```

*****
;ROUTINE TO LOOP THRU A SINGLE INSTRUCTION TEST OR ERROR FUNCTION TEST
;NOTE THAT SW11 MUST BE DOWN AFTER 2ND HALT
;NOTE THAT SW<14> MUST BE SET.....MUST BE SET.....MUST BE SET.....
*****

```

```

4215 024100          TESTX:
4216 024100 012706 001100    MOV      #SCMTAG,R6          ;;FIRST LOCATION TO BE CLEARED
4217 024104 005026          CLR      (R6)+          ;;CLEAR MEMORY LOCATION

```

| | | | | | | | | |
|------|--------|--------|--------|--------|-------|-------|--------------------|-------------------------------------------------------------|
| 4218 | 024106 | 022706 | 001126 | | | CMP | #SBO DAT, R6 | :: DONE? |
| 4219 | 024112 | 001374 | | | | BNE | .-6 | :: LOOP BACK IF NO |
| 4220 | 024114 | 012706 | 001100 | | | MOV | #STACK, SP | :: SETUP THE STACK POINTER |
| 4221 | 024120 | 012737 | 026326 | 000020 | | MOV | #SCOPE, @IOTVEC | :: IOT VECTOR FOR SCOPE ROUTINE |
| 4222 | 024126 | 012737 | 000340 | 000022 | | MOV | #340, @IOTVEC+2 | :: LEVEL 7 |
| 4223 | 024134 | 012737 | 026152 | 000030 | | MOV | #ERROR, @EMTVEC | :: EMT VECTOR FOR ERROR ROUTINE |
| 4224 | 024142 | 012737 | 000340 | 000032 | | MOV | #340, @EMTVEC+2 | :: LEVEL 7 |
| 4225 | 024150 | 012737 | 027374 | 000034 | | MOV | #STRAP, @TRAPVEC | :: TRAP VECTOR FOR TRAP CALLS |
| 4226 | 024156 | 012737 | 000340 | 000036 | | MOV | #340, @TRAPVEC+2 | :: LEVEL 7 |
| 4227 | 024164 | 012737 | 027430 | 000024 | | MOV | #SPWRON, @PWRVEC | :: POWER FAILURE VECTOR |
| 4228 | 024172 | 012737 | 000340 | 000026 | | MOV | #340, @PWRVEC+2 | :: LEVEL 7 |
| 4229 | 024200 | 013737 | 014774 | 014766 | | MOV | SENDCT, SEOPCT | :: SETUP END-OF-PROGRAM COUNTER |
| 4230 | 024206 | 005037 | 001220 | | | CLR | \$TIMES | :: INITIALIZE NUMBER OF ITERATIONS |
| 4231 | 024212 | 012737 | 015140 | 000014 | | MOV | #SRTN, @TBITVEC | :: SET "T" BIT VECTOR TO \$SRTN |
| 4232 | 024220 | 012737 | 000340 | 000016 | | MOV | #340, @TBITVEC+2 | :: LEVEL 7 |
| 4233 | 024226 | 012737 | 000002 | 015140 | | MOV | #RTI, \$SRTN | :: SET \$SRTN TO A RTI |
| 4234 | 024234 | 012737 | 024262 | 000010 | | MOV | #65\$, @RESVEC | :: TRY TO DO A RTT |
| 4235 | 024242 | 005046 | | | | CLR | -(SP) | :: DUMMY PS |
| 4236 | 024244 | 012746 | 024252 | | | MOV | #64\$, -(SP) | :: AND PC |
| 4237 | 024250 | 000006 | | | | RTT | | :: TRY THE RTT |
| 4238 | 024252 | 012737 | 000006 | 015140 | 64\$: | MOV | #RTT, \$SRTN | :: RTT IS LEGAL--SET \$SRTN TO A RTT |
| 4239 | 024260 | 000402 | | | | BR | 66\$ | |
| 4240 | 024262 | 062706 | 000010 | | 65\$: | ADD | #10, SP | :: RTT ILLEGAL--CLEAN OFF THE STACK |
| 4241 | 024266 | 012737 | 000012 | 000010 | 66\$: | MOV | #RESVEC+2, @RESVEC | :: RESTORE TRAP CATCHER |
| 4242 | 024274 | 005037 | 015146 | | | CLR | \$TBIT | :: CLEAR "T" BIT SWITCH |
| 4243 | 024300 | 012737 | 024300 | 001106 | | MOV | #, \$LPADR | :: INITIALIZE THE LOOP ADDRESS FOR SCOPE |
| 4244 | 024306 | 013746 | 000004 | | | MOV | @#4, -(SP) | :: SAVE ERROR VECTOR |
| 4245 | 024312 | 013746 | 000006 | | | MOV | @#6, -(SP) | |
| 4246 | 024316 | 012737 | 024332 | 000004 | | MOV | #67\$, 4 | :: SET UP TIME OUT VECTOR |
| 4247 | 024324 | 005777 | 154606 | | | TST | @SWR | :: TRY TO REFERENCE HARDWARE SWR |
| 4248 | 024330 | 000407 | | | | BR | 68\$ | :: BRANCH IF NO TIMEOUT TRAP OCCURS |
| 4249 | 024332 | 012737 | 000176 | 001136 | 67\$: | MOV | #SWREG, SWR | :: POINT TO SOFTWARE SWR |
| 4250 | 024340 | 012737 | 000174 | 001140 | | MOV | #DISPREG, DISPLAY | :: POINT TO SOFTWARE DISPLAY REG |
| 4251 | 024346 | 022626 | | | | CMP | (SP)+, (SP)+ | :: RESTORE STACK |
| 4252 | 024350 | 012637 | 000006 | | 68\$: | MOV | (SP)+, @#6 | :: RESTORE ERROR VECTOR |
| 4253 | 024354 | 012637 | 000004 | | | MOV | (SP)+, @#4 | |
| 4254 | 024360 | 004737 | 045366 | | | JSR | PC, SETUP | :: SETUP POINTERS AND FLAGS |
| 4255 | 024364 | 104400 | 034223 | | | TYPE, | MSG23 | :: ASK USER TO LOAD ADDRESS OF DESIRED |
| 4256 | | | | | | | | :: TEST INTO SWITCH REGISTER, THEN |
| 4257 | | | | | | | | :: PRESS CONTINUE |
| 4258 | 024370 | 000000 | | | | HALT | | :: WAIT FOR STARTING ADDRESS |
| 4259 | 024372 | 017737 | 154540 | 024622 | | MOV | @SWR, RETRNX | :: STORE STARTING ADDRESS |
| 4260 | 024400 | 062737 | 000002 | 024622 | | ADD | #2, RETRNX | :: CHANGE TO FIRST ADDRESS AFTER SCOPE INSTRUCTION |
| 4261 | 024406 | 104400 | 034472 | | | TYPE, | MSG24 | :: ASK USER TO SET SWITCH REGISTER |
| 4262 | | | | | | | | :: OPTIONS, THEN PRESS CONTINUE |
| 4263 | 024412 | 000000 | | | | HALT | | :: SET SWR OPTIONS (BIT 11 MUST = 0) |
| 4264 | 024414 | 032777 | 010000 | 154514 | | BIT | #10000, @SWR | :: CHECK SW12 |
| 4265 | 024422 | 001432 | | | | BEQ | 1\$ | :: BRANCH IF NOT SET |
| 4266 | | | | | | | | :: THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIC #20, PS" |
| 4267 | 024424 | 005046 | | | | CLR | -(SP) | |
| 4268 | 024426 | 013746 | 000034 | | | MOV | 34, -(SP) | :: SAVE CURRENT TRAP VECTOR |
| 4269 | 024432 | 012737 | 024442 | 000034 | | MOV | #69\$, 34 | :: SETUP NEW TRAP VECTOT |
| 4270 | 024440 | 104400 | | | | TRAP | | :: PUSH OLD PSW AN PCOM STACK |
| 4271 | 024442 | 016666 | 000002 | 000006 | 69\$: | MOV | 2(SP), 6(SP) | :: |

```

4272 024450 012716 024456      MOV      #70$, (SP)          ;; REPLACE OLD PC WITH NEW
4273 024454 000002      RTI                          ;; RESTORE PSW
4274 024456 012637 000034      MOV      (SP)+, 34          ;; RESTORE OLD TRAP VECTOR
4275 024462 012637 001214      MOV      (SP)+, STMP6
4276 024466 042737 000020 001214      BIC      #20, STMP6
4277 024471 013746 001214      MOV      STMP6, -(SP)      ;; PUT NEW PS ON STACK
4278 024500 012746 024506      MCV      #71$, -(SP)      ;; PUT NEW PC ON STACK
4279 024504 000002      RTI                          ;; POP NEW PC AND PS
4280 024506
4281 024506 000431      BR       2$                  ; SKIP NEXT INSTRUCTION
4282 024510
4283
4284 024510 005046      CLR      -(SP)              ; THE FOLLOWING REPRESENTS THE EQUIVALENT OF "BIS #20, PS"
4285 024512 013746 000034      MOV      34, -(SP)          ; SAVE CURRENT TRAP VECTOR
4286 024516 012737 024526 000034      MOV      #72$, 34          ; SETUP NEW TRAP VECTOR
4287 024524 104400      TRAP
4288 024526 016666 000002 000006 72$:      MOV      2(SP), 6(SP)      ;
4289 024534 012716 024542      MOV      #73$, (SP)        ; REPLACE OLD PC WITH NEW
4290 024540 000002      RTI                          ; RESTORE PSW
4291 024542 012637 000034      MOV      (SP)+, 34          ; RESTORE OLD TRAP VECTOR
4292 024546 012637 001214      MOV      (SP)+, STMP6
4293 024552 052737 000020 001214      BIS      #20, STMP6
4294 024560 013746 001214      MOV      STMP6, -(SP)      ;; PUT NEW PS ON STACK
4295 024564 012746 024572      MCV      #74$, -(SP)      ;; PUT NEW PC ON STACK
4296 024570 000002      RTI                          ;; POP NEW PC AND PS
4297 024572
4298 024572 012737 024600 027576 74$:      MOV      #XLOOP, RETURN    ; SAVE RETURN POINT FOR THIS SECTION
4299
4300 024600 104400 030556      XLOOP:  TYPE,  STMES        ; FOR POWER FAILURE RETURN
4301 024604 104400 031454      TYPE,  MLOOP              ; TYPE MAINDEC TITLE & REV. LEVEL
4302
4303 024610 013737 024622 001106      MOV      RETRNX, $LPADR    ; INDICATE "ENTERING A LOOP ON TEST
4304
4305 024616 000177 000000      JMP      @RETRNX           ; SELECTED BY THE USER"
4306 024622 000000      RETRNX: 0                  ; STORE 1ST ADDRESS OF TEST FOR LOOPING
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325 024624      CKSAME:

```

```

*****
: 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.
*****

```

| | | | | | | | | |
|------|--------|--------|--------|--------|-------|------------------|------------------------------------------|-------------------------------------|
| 4326 | 024624 | 012706 | 001100 | | MOV | #SCMTAG,R6 | :: FIRST LOCATION TO BE CLEARED | |
| 4327 | 024630 | 005026 | | | CLR | (R6)+ | :: CLEAR MEMORY LOCATION | |
| 4328 | 024632 | 022706 | 001126 | | CMP | #SBO DAT,R6 | :: DONE? | |
| 4329 | 024636 | 001374 | | | BNE | .-6 | :: LOOP BACK IF NO | |
| 4330 | 024640 | 012706 | 001100 | | MOV | #STACK,SP | :: SETUP THE STACK POINTER | |
| 4331 | 024644 | 012737 | 026326 | 000020 | MOV | #SCOPE,@IOTVEC | :: IOT VECTOR FOR SCOPE ROUTINE | |
| 4332 | 024652 | 012737 | 000340 | 000022 | MOV | #340,@IOTVEC+2 | :: LEVEL 7 | |
| 4333 | 024660 | 012737 | 026152 | 000030 | MOV | #ERROR,@EMTVEC | :: EMT VECTOR FOR ERROR ROUTINE | |
| 4334 | 024666 | 012737 | 000340 | 000032 | MOV | #340,@EMTVEC+2 | :: LEVEL 7 | |
| 4335 | 024674 | 012737 | 027374 | 000034 | MOV | #TRAP,@TRAPVEC | :: TRAP VECTOR FOR TRAP CALLS | |
| 4336 | 024702 | 012737 | 000340 | 000036 | MOV | #340,@TRAPVEC+2 | :: LEVEL 7 | |
| 4337 | 024710 | 012737 | 027430 | 000024 | MOV | #SPWRON,@PWAVEC | :: POWER FAILURE VECTOR | |
| 4338 | 024716 | 012737 | 000340 | 000026 | MOV | #340,@PWAVEC+2 | :: LEVEL 7 | |
| 4339 | 024724 | 013737 | 014774 | 014766 | MOV | SENOCT,SEOPCT | :: SETUP END-OF-PROGRAM COUNTER | |
| 4340 | 024732 | 005037 | 001220 | | CLR | \$TIMES | :: INITIALIZE NUMBER OF ITERATIONS | |
| 4341 | 024736 | 012737 | 015140 | 000014 | MOV | #SRTN,@TBITVEC | :: SET "T" BIT VECTOR TO SRTN | |
| 4342 | 024744 | 012737 | 000340 | 000016 | MOV | #340,@TBITVEC+2 | :: LEVEL 7 | |
| 4343 | 024752 | 012737 | 000002 | 015140 | MOV | #RTI,SRTN | :: SET SRTN TO A RTI | |
| 4344 | 024760 | 012737 | 025006 | 000010 | MOV | #655,@RESVEC | :: TRY TO DO A RTT | |
| 4345 | 024766 | 005046 | | | CLR | -(SP) | :: DUMMY PS | |
| 4346 | 024770 | 012746 | 024776 | | MOV | #645,-(SP) | :: AND PC | |
| 4347 | 024774 | 000006 | | | RTT | | :: TRY THE RTT | |
| 4348 | 024776 | 012737 | 000006 | 015140 | 645: | MOV | #RTT,SRTN | :: RTT IS LEGAL--SET SRTN TO A RTT |
| 4349 | 025004 | 000402 | | | BR | 665 | | |
| 4350 | 025006 | 062706 | 000010 | | 665: | ADD | #10,SP | :: RTT ILLEGAL--CLEAN OFF THE STACK |
| 4351 | 025012 | 012737 | 000012 | 000010 | 665: | MOV | #RESVEC+2,@RESVEC | :: RESTORE TRAP CATCHER |
| 4352 | 025020 | 005037 | 015146 | | CLR | \$TBIT | :: CLEAR "T" BIT SWITCH | |
| 4353 | 025024 | 012737 | 025024 | 001106 | MOV | #,SLPADR | :: INITIALIZE THE LOOP ADDRESS FOR SCOPE | |
| 4354 | 025032 | 013746 | 000004 | | MOV | @#4,-(SP) | :: SAVE ERROR VECTOR | |
| 4355 | 025036 | 013746 | 000006 | | MOV | @#6,-(SP) | | |
| 4356 | 025042 | 012737 | 025056 | 000004 | MOV | #675,4 | :: SET UP TIME OUT VECTOR | |
| 4357 | 025050 | 005777 | 154062 | | TST | @SWR | :: TRY TO REFERENCE HARDWARE SWR | |
| 4358 | 025054 | 000407 | | | BR | 685 | :: BRANCH IF NO TIMEOUT TRAP OCCURS | |
| 4359 | 025056 | 012737 | 000176 | 001136 | 675: | MOV | #SWREG,SWR | :: POINT TO SOFTWARE SWR |
| 4360 | 025061 | 012737 | 000174 | 001140 | MOV | #DISPREG,DISPLAY | :: POINT TO SOFTWARE DISPLAY REG | |
| 4361 | 025072 | 022626 | | | CMP | (SP)+,(SP)+ | :: RESTORE STACK | |
| 4362 | 025074 | 012637 | 000006 | | 685: | MOV | (SP)+,@#6 | :: RESTORE ERROR VECTOR |
| 4363 | 025100 | 012637 | 000004 | | MOV | (SP)+,@#4 | | |
| 4364 | 025104 | 012737 | 024624 | 027576 | MOV | #CKSAME,RETURN | :: SAVE RETURN POINT FOR THIS SECTION | |
| 4365 | | | | | | | :: FOR POWER FAILURE ROUTINE | |
| 4366 | 025112 | 104400 | 030556 | | TYPE, | \$TIMES | :: TYPE MAINDEC TITLE & REV. LEVEL | |
| 4367 | 025116 | 104400 | 031527 | | TYPE, | MPATS | :: INDICATE "ENTERING SINGLE DATA | |
| 4368 | | | | | | | :: PATTERN TESTING" | |
| 4369 | 025122 | 004737 | 045366 | | JSR | PC,SETUP | :: INITIALIZE POINTERS | |
| 4370 | 025126 | 104400 | 034574 | | TYPE, | MSG25 | :: ASK USER TO LOAD PATTERN VALUE INTO | |
| 4371 | | | | | | | :: SWITCH REGISTER SWS<11:0>, THEN | |
| 4372 | | | | | | | :: PRESS CONTINUE | |
| 4373 | 025132 | 000000 | | | HALT | | :: WAIT FOR CARD IMAGE PATTERN | |
| 4374 | 025134 | 017737 | 153776 | 025736 | MOV | @SWR,CARDIM | :: STORE PATTERN | |
| 4375 | 025142 | 042737 | 170000 | 025736 | BIC | #170000,CARDIM | :: CLEAR UPPER BITS OF PATTERN | |
| 4376 | 025150 | 013737 | 025736 | 025740 | MOV | CARDIM,CDPKO | | |
| 4377 | 025156 | 005037 | 025742 | | CLR | DERFLG | | |
| 4378 | 025162 | 006037 | 025740 | | ROR | CDPKO | | |
| 4379 | 025166 | 106137 | 025741 | | ROLB | CDPK1 | | |

| | | | | | | | |
|------|--------|--------|---------------|----------|-----------|---------------|---------------------------------------------|
| 4434 | 025454 | 032713 | 010000 | CKERR: | BIT | #10000, 2CDS | ;CHECK FOR OFFLINE |
| 4435 | 025460 | 001406 | | | BEQ | CKERR1 | ;BRANCH IF NOT |
| 4436 | 025462 | 104400 | 001222 | | TYPE, | SBELL | ;RING-A-DING |
| 4437 | 025466 | 032713 | 000010 | CKERR3: | BIT | #10, 2CDS | ;CHECK TRANSITION TO ON-LINE |
| 4438 | 025472 | 001775 | | | BEQ | CKERR3 | ;BRANCH IF OFF-LINE |
| 4439 | 025474 | 000666 | | | BR | CKSTR | ;START OVER |
| 4440 | | | | | | | |
| 4441 | 025476 | 032713 | 004000 | CKERR1: | BIT | #4000, 2CDS | ;CHECK FOR DATA ERROR |
| 4442 | 025502 | 001407 | | | BEQ | CKERR2 | |
| 4443 | 025504 | 005737 | 025742 | | TST | DERFLG | |
| 4444 | 025510 | 100004 | | | BFL | CKERR2 | |
| 4445 | 025512 | 122737 | 000001 025742 | | CMPB | #1, DERFLG | |
| 4446 | 025520 | 003323 | | | BGT | CKLOP3 | ;BRANCH IF LEGIT |
| 4447 | 025522 | 104062 | | CKERR2: | ERROR +62 | | ;REAL, LIVE ERROR. |
| 4448 | 025524 | 000662 | | | BR | CKLOOP | |
| 4449 | | | | | | | |
| 4450 | 025526 | 005237 | 025732 | CKFAIL: | INC | TOTERR | ;COUNT ERRORS |
| 4451 | 025532 | 032777 | 020000 153376 | | BIT | #20000, 2SWR | ;CHECK FOR INHIBITING PRINTOUT |
| 4452 | 025540 | 001052 | | | BNE | CKERROR | ;BRANCH AROUND PRINTOUT IF SET |
| 4453 | 025542 | 005737 | 001260 | | TST | ERFLG | ;TEST FLAG TO PRINT HEADING |
| 4454 | 025546 | 001004 | | | BNE | CKNOHD | ;BRANCH IF ALREADY DONE |
| 4455 | 025550 | 005237 | 001260 | | INC | ERFLG | ;PRINT HEADING ONCE ONLY |
| 4456 | 025554 | 104400 | 033442 | | TYPE, | MSG19 | ;OUTPUT HEADING |
| 4457 | 025560 | 104400 | 030553 | CKNOHD: | TYPE, | CRLF | ;OUTPUT CARRIAGE RETURN, LINEFEED |
| 4458 | 025564 | 005237 | 015334 | | INC | CLCNT | ;STEP UP COLUMN COUNT FOR ERROR REPORT |
| 4459 | 025570 | 013746 | 015334 | | MOV | CLCNT, -(SP) | ;PRINT THE COLUMN OF THE CARD |
| 4460 | 025574 | 104404 | | | TYPDS | | ;ON WHICH AN ERROR WAS DETECTED |
| 4461 | 025576 | 104400 | 030545 | | TYPE, | SPACE | |
| 4462 | 025602 | 005337 | 015334 | | DEC | CLCNT | ;DROP COLUMN COUNT BACK AFTER REPORTING |
| 4463 | 025606 | 005737 | 025742 | | TST | DERFLG | |
| 4464 | 025612 | 100007 | | | BPL | CKNOPK | |
| 4465 | 025614 | 104400 | 030545 | | TYPE, | SPACE | |
| 4466 | 025620 | 114046 | | | MOVB | -(RO), -(SP) | ;PRINT THE INCORRECT VALUE |
| 4467 | 025622 | 104402 | | | TYPOS | | ;THAT WAS READ FROM THE |
| 4468 | 025624 | 003 | | | .BYTE | 3 | ;COLUMN INDICATED ABOVE |
| 4469 | 025625 | 000 | | | .BYTE | 0 | ; (PACKED MODE) |
| 4470 | 025626 | 105720 | | | TSTB | (RO)+ | ;CORRECT MEMORY LOCATION AFTER PRINTOUT |
| 4471 | 025630 | 000404 | | | BR | CKOVR1 | |
| 4472 | 025632 | 014046 | | CKNOPK: | MOV | -(RO), -(SP) | ;PRINT THE INCORRECT VALUE THAT |
| 4473 | 025634 | 104402 | | | TYPOS | | ;WAS READ FROM THE COLUMN |
| 4474 | 025636 | 006 | | | .BYTE | 6 | ;INDICATED ABOVE |
| 4475 | 025637 | 001 | | | .BYTE | 1 | ; (IMAGE MODE) |
| 4476 | 025640 | 005720 | | | TST | (RO)+ | ;CORRECT THE MEMORY LOCATION AFTER PRINTOUT |
| 4477 | 025642 | 104400 | 030545 | CKOVR1: | TYPE, | SPACE | |
| 4478 | 025646 | 013746 | 025734 | | MOV | ?OTCRD, -(SP) | ;PRINT THE CARD NUMBER ON WHICH |
| 4479 | 025652 | 104404 | | | TYPDS | | ;THE ERROR WAS FOUND |
| 4480 | 025654 | 104400 | 030545 | | TYPE, | SPACE | |
| 4481 | 025660 | 013746 | 025732 | | MOV | TOTERR, -(SP) | ;PRINT THE TOTAL NUMBER OF ERRORS |
| 4482 | 025664 | 104404 | | | TYPDS | | ;ACCUMULATED TO NOW |
| 4483 | 025666 | 005777 | 153244 | CKERR0P: | TST | 2SWR | ;CHECK SW15 TO HALT ON ERROR |
| 4484 | 025672 | 100001 | | | BPL | 15 | ;BRANCH IF NOT SET |
| 4485 | 025674 | 000000 | | | HALT | | ;HALT ON ERROR |
| 4486 | 025676 | 005237 | 015334 | 15: | INC | CLCNT | ;STEP UP COLUMN COUNT TO BE CORRECT FOR |
| 4487 | | | | | | | ;NEXT POSSIBLE COLUMN TO BE CHECKED |

| | | | | | | | |
|------|--------|--------|--------|--------|-----|---------|-------------|
| 4488 | 025702 | 023727 | 015334 | 000120 | | CMP | CLCNT, #120 |
| 4489 | 025710 | 001002 | | | | BNE | 25 |
| 4490 | 025712 | 000137 | 025272 | | | JMP | CKLOOP |
| 4491 | 025716 | C32777 | 000004 | 153212 | 25: | BIT | #4, 2SWR |
| 4492 | 025724 | 001233 | | | | BNE | CKLOP2 |
| 4493 | 025726 | 000137 | 025370 | | | JMP | CKLOP3 |
| 4494 | | | | | | | |
| 4495 | | | | | | | |
| 4496 | 025732 | 000000 | | | | TOTERR: | 0 |
| 4497 | 025734 | 000000 | | | | TOTCRD: | 0 |
| 4498 | 025736 | 000000 | | | | CARDIM: | 0 |
| 4499 | 025740 | 000 | | | | CDPK0: | .BYTE 0 |
| 4500 | 025741 | 000 | | | | CDPK1: | .BYTE 0 |
| 4501 | 025742 | 000000 | | | | DERFLG: | 0 |

```

: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

```

```

4502
4503 ;ISSUE MESSAGE IF CARD READER IS OFF-LINE
4504 ;WAIT FOR BUSY TO CLEAR IN CASE CARD READER IS STILL READING A CARD
4505 ;INITIALIZE STATUS REGISTER AND USE ERROR HALT IF IT DOESN'T CLEAR PROPERLY
4506 ;NOTE THAT PROGRAM WILL HANG HERE IF BUSY REMAINS SET
4507 025744 094737 025772 INIT: JSR PC CKOFFL ;SEE IF OFF-LINE BIT IS SET
4508 025750 105713 1S: TSTB @CDS ;WAIT FOR CONTROLLER READY, IN CASE
4509 025752 100376 BPL 1$ ;A CARD IS STILL BEING READ
4510 025754 012713 000400 MOV #400, @CDS ;INITIALIZE THE CARD READER
4511 025760 022713 000200 CMP #200, @CDS ;MAKE SURE INITIALIZATION OK
4512 025764 001401 BEQ 2$ ;BRANCH IF ALL BITS ZERO
4513 025766 104067 ERROR +67 ;NOT ALL BITS OF STATUS REGISTER ARE ZERO
4514 025770 000207 2$: RTS PC ;RETURN
4515
4516 ;SUBROUTINE TO CHECK FOR BIT 12 (OFF-LINE) BEING SET IN CARD
4517 ;READER CSR, AND PRINT OUT A MESSAGE IF IT IS
4518 025772 032713 010000 CKOFFL: BIT #10000, @CDS ;CHECK BIT 12
4519 025776 001001 BNE 1$ ;BRANCH IF SET
4520 026000 000207 RTS PC ;RETURN IF NOT SET
4521 026002 104400 033410 1$: TYPE, MSG18 ;"CARD READER OFF-LINE"
4522 026006 104400 033175 TYPE, MSG17 ;"REMEDY THE CONDITION ... ETC."
4523 026012 000000 HALT ;WAIT FOR CONTINUE
4524 026014 000766 BR CKOFFL ;CHECK AGAIN
4525
4526 ;*****
4527
4528 .SBTTL ERROR MESSAGE TYPEOUT ROUTINE
4529
4530 ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
4531 ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
4532 ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
4533
4534 $ERRTYP:
4535 026016 104400 001227 TYPE $SCLF ;;"CARRIAGE RETURN" & "LINE FEED"
4536 026022 010045 MOV RO, -(SP) ;SAVE RC
4537 026024 005000 CLR RO ;PICKUP THE ITEM INDEX
4538 026026 153700 001114 BISB @#$ITEMB, RO
4539 026032 001004 BNE 1$ ;IF ITEM NUMBER IS ZERO, JUST
4540 ;TYPE THE PC OF THE ERROR
4541 026034 013746 001116 MOV $ERRPC, -(SP) ;SAVE $ERRPC FOR TYPEOUT
4542 ;ERROR ADDRESS
4543 ;GO TYPE--OCTAL ASCII(ALL DIGITS)
4544 026040 104401 TYPOC ;GET OUT
4545 026042 000426 BR 6$ ;ADJUST THE INDEX SO THAT IT WILL
4546 026044 005300 1$: DEC RO ;WORK FOR THE ERROR TABLE
4547 026046 006300 ASL RO
4548 026050 006300 ASL RO
4549 026052 006300 ASL RO
4550 026054 062700 001270 ADD #$ERRTB, RO ;FORM TABLE POINTER
4551 026060 012037 026070 MOV (RO)+, 2$ ;PICKUP "ERROR MESSAGE" POINTER
4552 026064 001404 BEQ 3$ ;SKIP TYPEOUT IF NO POINTER
4553 026066 104400 TYPE ;TYPE THE "ERROR MESSAGE"
4554 026070 000000 2$: .WORD 0 ;"ERROR MESSAGE" POINTER GOES HERE
4555 026072 104400 001227 TYPE ,SCLF ;;"CARRIAGE RETURN" & "LINE FEED"

```

```

4556 026076 012037 026106      3$:  MOV      (RD)+,4$      ; PICKUP 'DATA HEADER' POINTER
4557 026102 001404                BEQ      5$           ; SKIP TYPEOUT IF 0
4558 026104 104400                TYPE                    ; TYPE THE "DATA HEADER"
4559 026106 000200                WORD      0           ; "DATA HEADER" POINTER GOES HERE
4560 026110 104400 001227        TYPE      $CRLF       ; "CARRIAGE RETURN" & "LINE FEED"
4561 026114 011000                MOV      (RD),RO      ; PICKUP "DATA TABLE" POINTER
4562 026116 001004                BNE      7$           ; GO TYPE THE DATA
4563 026120 012600                MOV      (SP)+,RO     ; RESTORE RO
4564 026122 104400 001227        TYPE      $CRLF       ; "CARRIAGE RETURN" & "LINE FEED"
4565 026126 000207                RTS      PC           ; RETURN
4566 026130
4567 026130 013046                MOV      2(RD)+,-(SP) ; SAVE 2(RD)+ FOR TYPEOUT
4568 026132 104401                TYPOC                    ; GO TYPE--OCTAL ASCII(ALL DIGITS)
4569 026134 005710                TST      (RO)          ; IS THERE ANOTHER NUMBER?
4570 026136 001770                BEQ      6$           ; BR IF NO
4571 026140 104400 026146        TYPE      8$          ; TYPE TWO(2) SPACES
4572 026144 000771                BR      7$           ; LOOP
4573 026146 020040      000      8$:  .ASCIZ  / /          ; TWO(2) SPACES
4574 026152
4575
4576
4577
4578 .SBTTL  ERROR HANDLER ROUTINE
4579
4580 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
4581 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
4582 ;*AND GO TO $ERRTYP ON ERROR
4583 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4584 ;*SW15=1      HALT ON ERROR
4585 ;*SW13=1      INHIBIT ERROR TYPEOUTS
4586 ;*SW10=1      BELL ON ERROR
4587 ;*CALL
4588 ;*      ERROR  N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
4589
4590 $ERROR:
4591 026152 010637 001174                MOV      SP,$REG6     ; STACK POINTER POSITION
4592 026156 016637 000002 001176    MOV      2(SP),$REG7  ; CONTENTS OF 'PSW'
4593 026164 011337 001200                MOV      2CDS,$TMP0   ; CONTENTS OF DEVICE 'CDS'
4594 026170 017737 153044 001202    MOV      2CDD,$TMP1   ; CONTENTS OF DEVICE 'CDD'
4595 026176 011437 001204                MOV      2CDC,$TMP2   ; CONTENTS OF DEVICE 'CDC'
4596 026202 011537 001206                MOV      2CDA,$TMP3   ; CONTENTS OF DEVICE 'CDA'
4597 026206 105237 001103        7$:  INCB     $ERFLG      ; SET THE ERROR FLAG
4598 026212 001775                BEQ      7$           ; DON'T LET THE FLAG GO TO ZERO
4599 026214 013777 001102 152716    MOV      $TSYMN,$DISPLAY ; DISPLAY TEST NUMBER AND ERROR FLAG
4600 026222 032777 002000 152706    BIT      #BIT10,$SWR   ; BELL ON ERROR?
4601 026230 001402                BEQ      1$           ; NO - SKIP
4602 026232 104400 001222        TYPE      $BELL       ; RING BELL
4603 026236 005237 001112        1$:  INC      $ERTTL      ; COUNT THE NUMBER OF ERRORS
4604 026242 011637 001116                MOV      (SP),$ERRPC  ; GET ADDRESS OF ERROR INSTRUCTION
4605 026246 162737 000002 001116    SUB      #2,$ERRPC    ; STRIP AND SAVE THE ERROR ITEM CODE
4606 026254 117737 152636 001114    MOVB    2$ERRPC,$ITEMB ; SKIP TYPEOUT IF SET
4607 026262 032777 020000 152646    BIT      #BIT13,$SWR  ; SKIP TYPEOUTS
4608 026270 001004                BNE     20$          ; SKIP TYPEOUTS
4609 026272 004737 02601E                JSR     PC,$ERRTYP   ; GO TO USER ERROR ROUTINE

```

```

4610 026276 104400 001227
4611 026302
4612 026302 005777 152630
4613 026306 100006
4614 026310 000000
4615 026312 022737 015046 000042
4616 026320 001001
4617 026322 000000
4618 026324
4619 026324 000002
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634 026326
4635 026326 032777 040000 152602
4636 026334 001055
4637
4638 026336 000416
4639
4640 026340 013746 000004
4641 026344 012737 026364 000004
4642 026352 005737 177060
4643 026356 012637 000004
4644 026362 000436
4645 026364 022626
4646 026366 012637 000004
4647 026372 000436
4648 026374
4649 026374 105737 001103
4650 026400 001404
4651 026402 105037 001103
4652 026406 005037 001220
4653 026412 032777 004000 152516
4654 026420 001011
4655 026422 005737 001100
4656 026426 001406
4657 026430 005237 001104
4658 026434 023737 001220 001104
4659 026442 002012
4660 026444 012737 000001 001104
4661 026452 013737 026504 001220
4662 026460 105237 001102
4663 026464 011637 001106

                                TYPE ,SCRLF
20$:
25: TST      @SWR          ;; HALT ON ERROR
    BPL     @S           ;; SKIP IF CONTINUE
    HALT    @S           ;; HALT ON ERROR!
    CMP     @SENDAD,@#42 ;; ACT-11 AUTO-ACCEPT?
    BNE     @S           ;; BRANCH IF NO
    HALT    @S           ;; YES
35: RTI                ;; RETURN

;*****

.SBTTL SCOPE HANDLER ROUTINE

; *THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
; *AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:C>)
; *AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
; *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
; *SW14=1      LOOP ON TEST
; *SW11=1      INHIBIT ITERATIONS
; *CALL
; *          SCOPE          ;; SCOPE=IOT

$SCOPE:
15: BIT     @BIT14,@SWR    ;; LOOP ON PRESENT TEST?
    BNE     $OVER         ;; YES IF SW14=1
; *****START OF CODE FOR THE XOR TESTER*****
$XTSTR: BR     @S         ;; IF RUNNING ON THE "XOR" TESTER CHANGE
                                ;; THIS INSTRUCTION TO A "NOP" (NOP=240)
                                ;; SAVE THE CONTENTS OF THE ERROR VECTOR
    MOV     @#ERRVEC,-(SP) ;; SET FOR TIMEOUT
    MOV     @S,@#ERRVEC   ;; TIME OUT ON XOR?
    TST     @#177060      ;; RESTORE THE ERROR VECTOR
    MOV     (SP)+,@#ERRVEC ;; GO TO THE NEXT TEST
    BR     $SVLAD         ;; CLEAR THE STACK AFTER A TIME OUT
5$: CMP     (SP)+,(SP)+   ;; RESTORE THE ERROR VECTOR
    MOV     (SP)+,@#ERRVEC ;; LOOP ON THE PRESENT TEST
    BR     $OVER
6$: *****END OF CODE FOR THE XOR TESTER*****
25: TSTB   $ERFLG        ;; HAS AN ERROR OCCURRED?
    BEQ     @S           ;; BR IF NO
45: CLRB   $ERFLG        ;; ZERO THE ERROR FLAG
    CLR     $TIMES       ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
35: BIT     @BIT11,@SWR   ;; INHIBIT ITERATIONS?
    BNE     @S           ;; BR IF YES
    TST     $PASS        ;; IF FIRST PASS OF PROGRAM
    BEQ     @S           ;; INHIBIT ITERATIONS
    INC     $ICNT        ;; INCREMENT ITERATION COUNT
    CMP     $TIMES,$ICNT ;; CHECK THE NUMBER OF ITERATIONS MADE
    BGE     $OVER        ;; BR IF MORE ITERATION REQUIRED
15: MOV     @1,$ICNT     ;; REINITIALIZE THE ITERATION COUNTER
    MOV     @MXCNT,$TIMES ;; SET NUMBER OF ITERATIONS TO DO
$SVLAD: INCB   $TSTNM    ;; COUNT TEST NUMBERS
    MOV     (SP),$LPADR  ;; SAVE SCOPE LOOP ADDRESS

```

```

4664 026470 013777 001102 152442 $OVER: MOV $STNM, @DISPLAY ;; DISPLAY TEST NUMBER
4665 026476 013716 001106 MOV $LPHDR, (SP) ;; FUDGE RETURN ADDRESS
4666 026502 000002 RTI ;; FIXES PS
4667 026504 000001 $MXCNT: 1 ;; MAX. NUMBER OF ITERATIONS
4668
4669 ;*****
4670
4671 .SBTTL TYPE ROUTINE
4672
4673 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
4674 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
4675 ;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
4676 ;*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
4677 ;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
4678 ;*
4679 ;*CALL:
4680 ;*1) USING A TRAP INSTRUCTION
4681 ;* TYPE ,MESADR ;; MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
4682 ;*OR
4683 ;* TYPE
4684 ;* MESADR
4685 ;*
4686
4687 026506 105737 001155 $TYPE: TSTB $TPFLG ;; IS THERE A TERMINAL?
4688 026512 100002 BPL 1$ ;; BR IF YES
4689 026514 000000 HALT ;; HALT HERE IF NO TERMINAL
4690 026516 000407 BR 3$ ;; LEAVE
4691 026520 010046 1$: MOV RO, -(SP) ;; SAVE RO
4692 026522 017600 000002 MOV @2(SP), RO ;; GET ADDRESS OF ASCIZ STRING
4693 026526 112046 2$: MOVB (RO)+, -(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
4694 026530 001005 BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
4695 026532 005726 TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
4696 026534 012600 50$: MOV (SP)+, RO ;; RESTORE RO
4697 026536 062716 000002 3$: ADD #2, (SP) ;; ADJUST RETURN PC
4698 026542 070002 RTI ;; RETURN
4699 026544 122716 000011 4$: CMPB #HT, (SP) ;; BRANCH IF <HT>
4700 026550 001426 BEQ 8$
4701 026552 122716 000200 CMPB #TCRLF, (SP) ;; BRANCH IF NOT <CRLF>
4702 026556 001004 BNE 5$
4703 026560 005726 TST (SP)+ ;; POP <CR><LF> EQUIV
4704 026562 104400 TYPE ;; TYPE A CR AND LF
4705 026564 001227 $CRLF
4706 026566 000757 BR 2$ ;; GET NEXT CHARACTER
4707 026570 004737 026652 5$: JSR PC, $TYPEC ;; GO TYPE THIS CHARACTER
4708 026574 123726 001154 6$: CMPB $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
4709 026600 001352 BNE 2$ ;; IF NO GO GET NEXT CHAR.
4710 026602 013746 001152 MOV $NULL, -(SP) ;; GET # OF FILLER CHARS. NEEDED
4711 AND THE NULL CHAR.
4712 026606 105366 000001 7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
4713 026612 002770 BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
4714 026614 004737 026652 JSR PC, $TYPEC ;; GO TYPE A NULL
4715 026620 105337 026716 DECB $CHARCNT ;; DO NOT COUNT AS A COUNT
4716 026624 000770 BR 7$ ;; LOOP
4717

```

```

4718 ;HORIZONTAL TAB PROCESSOR
4719
4720 026626 112716 000040 8S:   MOVB   #40,(SP)      ;; REPLACE TAB WITH SPACE
4721 026632 004737 026652 9S:   JSR    PC,$TYPEC     ;; TYPE A SPACE
4722 026636 132737 000007 026715 BITB  #7,$SCHARCNT    ;; BRANCH IF NOT AT
4723 026644 001372          BNE    9S             ;; TAB STOP
4724 026646 005726          TST    (SP)+         ;; POP SPACE OFF STACK
4725 026650 000726          BR     2S             ;; GET NEXT CHARACTER
4726 026652 105777 152270 $TYPEC: TSTB  2$TPS        ;; WAIT UNTIL PRINTER IS READY
4727 026656 100375          BPL   $TYPEC
4728 026660 116677 000002 152262 MOVB  2(SP),2$TPB    ;; LOAD CHAR TO BE TYPED INTO DATA REG.
4729 026666 122766 000015 000002 CMFB  #15,2(SP)     ;; BRANCH IF
4730 026674 001003          BNE   1S            ;; NOT <CR>
4731 026676 105037 026716 CLRB  $SCHARCNT
4732 026702 000406          BR    $TYPEX
4733 026704 122766 000012 000002 1S:  CMPB  #12,2(SP)    ;; EXIT
4734 026712 002002          BGE  $TYPEX        ;; BRANCH IF
4735 026714 105227          INCB (PC)+       ;; <LF>
4736 026716 000000          $SCHARCNT: WORD 0 ;; INC SPACE
4737 026720 000207          $TYPEX: RTS   PC  ;; COUNT
4738
4739          ;; EQUATES
4740          THT=11
4741          TCRLF=200

```

```

4742 ;*****
4743
4744 .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
4745
4746 *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
4747 *OCTAL (ASCII) NUMBER AND TYPE IT.
4748 *$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
4749 *CALL:
4750 *      MOV    NUM,-(SP)      ;; NUMBER TO BE TYPED
4751 *      TYPOS          ;; CALL FOR TYPEOUT
4752 *      .BYTE  N           ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
4753 *      .BYTE  M           ;; M=1 OR 0
4754 *                               ;; 1=TYPE LEADING ZEROS
4755 *                               ;; 0=SUPPRESS LEADING ZEROS
4756 *
4757 *$STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
4758 *$TYPOS OR $TYPOC
4759 *CALL:
4760 *      MOV    NUM,-(SP)      ;; NUMBER TO BE TYPED
4761 *      TYPON          ;; CALL FOR TYPEOUT
4762 *
4763 *$STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
4764 *CALL:
4765 *      MOV    NUM,-(SP)      ;; NUMBER TO BE TYPED
4766 *      TYPOC          ;; CALL FOR TYPEOUT
4767 *
4768
4769 026722 017646 000000 027145 $TYPOS: MOV    3(SP),-(SP)  ;; PICKUP THE MODE
4770 026726 116637 000C01          MOVB  1(SP),$DFILL    ;; LOAD ZERO FILL SWITCH
4771 026734 112637 027147          MOVB  (SP)+,$SOMODE+1 ;; NUMBER OF DIGITS TO TYPE

```

```

4772 026740 062716 0000C2 ADD #2,(SP) ;;ADJUST RETURN ADDRESS
4773 026744 000406 BR $TYPON
4774 026746 112737 000001 027145 $TYPON: MOV #1,$OFILL ;;SET THE ZERO FILL SWITCH
4775 026754 112737 000006 027147 MOV #6,$OMODE+1 ;;SET FOR SIX(6) DIGITS
4776 026762 112737 000005 027144 $TYPON: MOV #5,$OCNT ;;SET THE ITERATION COUNT
4777 026770 010346 MOV R3,-(SP) ;;SAVE R3
4778 026772 010446 MOV R4,-(SP) ;;SAVE R4
4779 026774 010546 MOV R5,-(SP) ;;SAVE R5
4780 026776 113704 027147 MOV $OMODF+1,R4 ;;GET THE NUMBER OF DIGITS TO TYPE
4781 027002 005404 NEG R4
4782 027004 062704 000006 ADD #6,R4 ;;SUBTRACT IT FOR MAX. ALLOWED
4783 027010 110437 027146 MOV R4,$OMODE ;;SAVE IT FOR USE
4784 027014 113704 027145 MOV $OFILL,R4 ;;GET THE ZERO FILL SWITCH
4785 027020 016605 000012 MOV 12(SP),R5 ;;PICKUP THE INPUT NUMBER
4786 027024 005003 CLR R3 ;;CLEAR THE OUTPUT WORD
4787 027026 006105 1$: ROL R5 ;;ROTATE MSB INTO "C"
4788 027030 000404 BR 3$ ;;GO DO MSB
4789 027032 006105 2$: ROL R5 ;;FORM THIS DIGIT
4790 027034 006105 ROL R5
4791 027036 006105 ROL R5
4792 027040 010503 MOV R5,R3
4793 027042 006103 3$: ROL R3 ;;GET LSB OF THIS DIGIT
4794 027044 105337 027146 DECB $OMODE ;;TYPE THIS DIGIT?
4795 027050 100016 BPL 7$ ;;BR IF NO
4796 027052 042703 177770 BIC #177770,R3 ;;GET RID OF JUNK
4797 027056 001002 BNE 4$ ;;TEST FOR 0
4798 027060 005704 TST R4 ;;SUPPRESS THIS 0?
4799 027062 001403 BEQ 5$ ;;BR IF YES
4800 027064 005204 4$: INC R4 ;;DON'T SUPPRESS ANYMORE 0'S
4801 027066 052703 000060 BIS #'0,R3 ;;MAKE THIS DIGIT ASCII
4802 027072 052703 000040 5$: BIS #' ,R3 ;;MAKE ASCII IF NOT ALREADY
4803 027076 110337 027142 MOV R3,$$ ;;SAVE FOR TYPING
4804 027102 104400 027142 TYPE $$ ;;GO TYPE THIS DIGIT
4805 027106 105337 027144 7$: DECB $OCNT ;;COUNT BY 1
4806 027112 003347 3GT 2$ ;;BR IF MORE TO DO
4807 027114 002402 BLT 6$ ;;BR IF DONE
4808 027116 005204 INC R4 ;;INSURE LAST DIGIT ISN'T A BLANK
4809 027120 000744 BR 2$ ;;GO DO THE LAST DIGIT
4810 027122 012605 6$: MOV (SP)+,R5 ;;RESTORE R5
4811 027124 012604 MOV (SP)+,R4 ;;RESTORE R4
4812 027126 012603 MOV (SP)+,R3 ;;RESTORE R3
4813 027130 016665 000002 000004 MOV 2(SP),4(SP) ;;SET THE STACK FOR RETURNING
4814 027136 012616 MOV (SP)+,(SP)
4815 027140 000002 RTI ;;RETURN
4816 027142 000 8$: .BYTE 0 ;;STORAGE FOR ASCII DIGIT
4817 027143 000 .BYTE 0 ;;TERMINATOR FOR TYPE ROUTINE
4818 027144 000 $OCNT: .BYTE 0 ;;OCTAL DIGIT COUNTER
4819 027145 000 $OFILL: .BYTE 0 ;;ZERO FILL SWITCH
4820 027146 000000 $OMODE: .WORD 0 ;;NUMBER OF DIGITS TO TYPE

```

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

4821
4822
4823
4824
4825

```

4826
4827
4828
4829
4830
4831
4832
4833
4834
4835 027150
4836 027150 010046
4837 027152 010146
4838 027154 010246
4839 027156 010346
4840 027160 010546
4841 027162 012746 020200
4842 027166 016605 000020
4843 027172 100004
4844 027174 005405
4845 027176 112766 000055 000001
4846 027204 005000
4847 027206 012703 027364
4848 027212 112723 000040
4849 027216 005002
4850 027220 016001 027354
4851 027224 160105
4852 027226 002402
4853 027230 005202
4854 027232 000774
4855 027234 060105
4856 027236 005702
4857 027240 001002
4858 027242 105716
4859 027244 100407
4860 027246 106316
4861 027250 103003
4862 027252 116663 000001 177777
4863 027260 052702 000060
4864 027264 052702 000040
4865 027270 110223
4866 027272 005720
4867 027274 020027 000010
4868 027300 002746
4869 027302 003002
4870 027304 010502
4871 027306 000764
4872 027310 105726
4873 027312 100003
4874 027314 116663 177777 177776
4875 027322 105013
4876 027324 012605
4877 027326 012603
4878 027330 012602
4879 027332 012601

```

```

; *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
; *SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
; *NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
; *BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
; *REPLACED WITH SPACES.
; *CALL:
; *      MOV      NUM,-(SP)      ;; PUT THE BINARY NUMBER ON THE STACK
; *      TYPDS                    ;; GO TO THE ROUTINE
;
$TYPDS:
MOV      R0,-(SP)      ;; PUSH R0 ON STACK
MOV      R1,-(SP)      ;; PUSH R1 ON STACK
MOV      R2,-(SP)      ;; PUSH R2 ON STACK
MOV      R3,-(SP)      ;; PUSH R3 ON STACK
MOV      R5,-(SP)      ;; PUSH R5 ON STACK
MOV      #20200,-(SP)    ;; SET BLANK SWITCH AND SIGN
MOV      20(SP),R5      ;; GET THE INPUT NUMBER
RPL      1$            ;; BR IF INPUT IS POS.
NEG      R5            ;; MAKE THE BINARY NUMBER POS.
MOVB    #'-,1(SP)      ;; MAKE THE ASCII NUMBER NEG.
CLP      R0            ;; ZERO THE CONSTANTS INDEX
MOV      #50BLK,R3     ;; SETUP THE OUTPUT POINTER
MOVB    #'',(R3)+      ;; SET THE FIRST CHARACTER TO A BLANK
CLR      R2            ;; CLEAR THE BCD NUMBER
MOV      $DTBL(R0),R1  ;; GET THE CONSTANT
SUB      R1,R5         ;; FORM THIS BCD DIGIT
BLT     4$            ;; BR IF DONE
INC      R2            ;; INCREASE THE BCD DIGIT BY 1
BR      3$
4$:      ADD      R1,R5  ;; ADD BACK THE CONSTANT
TST     R2            ;; CHECK IF BCD DIGIT=0
BNE     5$            ;; FALL THROUGH IF 0
TSTB   (SP)          ;; STILL DOING LEADING 0'S?
BMI     7$            ;; BR IF YES
ASLB   (SP)          ;; MSD?
BCC     6$            ;; BR IF NO
MOVB   1(SP),-1(R3)  ;; YES--SET THE SIGN
BIS    #'0,R2        ;; MAKE THE BCD DIGIT ASCII
BIS    #' ,R2        ;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
MOVB   R2,(R3)+      ;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
TST    (R0)+         ;; JUST INCREMENTING
CMP    R0,#10        ;; CHECK THE TABLE INDEX
BLT    2$            ;; GO DO THE NEXT DIGIT
BGT    8$            ;; GO TO EXIT
MOV    R5,R2         ;; GET THE LSD
BR     6$            ;; GO CHANGE TO ASCII
TSTB  (SP)+         ;; WAS THE LSD THE FIRST NON-ZERO?
3PL    9$            ;; BR IF NO
MOVB  -1(SP),-2(R3)  ;; YES--SET THE SIGN FOR TYPING
CLRB  (R3)           ;; SET THE TERMINATOR
MOV   (SP)+,R5       ;; POP STACK INTO R5
MOV   (SP)+,R3       ;; POP STACK INTO R3
MOV   (SP)+,R2       ;; POP STACK INTO R2
MOV   (SP)+,R1       ;; POP STACK INTO R1

```



```

4880 027334 012600          MOV      (SP)+,RO      ;; POP STACK INTO RO
4881 027336 104400 027364  TYPE      $DBLK          ;; NOW TYPE THE NUMBER
4882 027342 016666 000002 000004  MOV      2(SP),4(SP)  ;; ADJUST THE STACK
4883 027350 012616          MOV      (SP)+,(SP)
4884 027352 000002          RTI                      ;; RETURN TO USER
4885 027354 023420          $DTBL: 1000.
4886 027356 001750          1000.
4887 027360 000144          100.
4888 027362 000012          10.
4889 027364 000004          $DBLK: .BLKW 4
4890
4891 ;*****
4892 .SBTTL TRAP DECODER
4893
4894 ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
4895 ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
4896 ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
4897 ;*GO TO THAT ROUTINE.
4898
4899 $TRAP: MOV      RO,-(SP)      ;; SAVE RO
4900 027374 010046          MOV      2(SP),RO      ;; GET TRAP ADDRESS
4901 027376 016600 000002  TST      -(RO)         ;; BACKUP BY 2
4902 027402 005740          MOVB    (PO),RO       ;; GET RIGHT BYTE OF TRAP
4903 027404 111000          ASL     RO            ;; POSITION FOR INDEXING
4904 027406 006300          MOV     $TRPAD(RO),RO  ;; INDEX TO TABLE
4905 027410 016000 027416  RTS      RO            ;; GO TO ROUTINE
4906 027414 000200
4907
4908 .SBTTL TRAP TABLE
4909
4910 ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
4911 ;*BY THE "TRAP" INSTRUCTION.
4912
4913 : ROUTINE
4914 : -----
4915 $TRPAD:
4916 027416          $TYPE  ;; CALL=TYPE      TRAP+0(104400) TTY TYPEOUT ROUTINE
4917 027416 026506  $TYPOC ;; CALL=TYPOC     TRAP+1(104401) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
4918 027420 026746  $TYPOS ;; CALL=TYPOS     TRAP+2(104402) TYPE OCTAL NUMBER (NO LEADING ZEROS)
4919 027422 026722  $TYPON ;; CALL=TYPON     TRAP+3(104403) TYPE OCTAL NUMBER (AS PER LAST CALL)
4920 027424 026762  $TYPDS ;; CALL=TYPDS     TRAP+4(104404) TYPE DECIMAL NUMBER (WITH SIGN)
4921 027426 027150
4922
4923 ;*****
4924 .SBTTL POWER DOWN AND UP ROUTINES
4925
4926 : POWER DOWN ROUTINE
4927 $PWDRN: MOV     #SILLUP,2#PWVVEC ;; SET FOR FAST UP
4928 027430 012737 027564 000024  MOV     #340,2#PWVVEC+2 ;; PRIO:7
4929 027436 012737 000340 000026  MOV     RO,-(SP)      ;; PUSH RO ON STACK
4930 027444 010046          MOV     R1,-(SP)     ;; PUSH R1 ON STACK
4931 027446 010146          MOV     R2,-(SP)     ;; PUSH R2 ON STACK
4932 027450 010246          MOV     R3,-(SP)     ;; PUSH R3 ON STACK
4933 027452 010346

```

```

4934 027454 010446      MOV      R4,-(SP)      ;; PUSH R4 ON STACK
4935 027456 010546      MOV      R5,-(SP)      ;; PUSH R5 ON STACK
4936 027460 010637 027570      MOV      SP,$$AVR6     ;; SAVE SP
4937 027464 012737 027476 000024      MOV      #SPWRUP,#PWRVEC ;; SET UP VECTOR
4938 027472 000000      HALT
4939 027474 000776      BR      -2            ;; HANG UP
4940
4941      : POWER UP ROUTINE
4942 027476 013706 027570      $PWRUP: MOV      $$AVR6,SP      ;; GET SP
4943 027502 005037 027570      CLR      $$AVR6        ;; WAIT LOOP FOR THE TTY
4944 027504 005237 027570      IS:     INC      $$AVR6        ;; WAIT FOR THE INC
4945 027512 001375      BNE      IS            ;; OF WORD
4946 027514 012605      MOV      (SP)+,R5      ;; POP STACK INTO R5
4947 027516 012604      MOV      (SP)+,R4      ;; POP STACK INTO R4
4948 027520 012603      MOV      (SP)+,R3      ;; POP STACK INTO R3
4949 027522 012602      MOV      (SP)+,R2      ;; POP STACK INTO R2
4950 027524 012601      MOV      (SP)+,R1      ;; POP STACK INTO R1
4951 027526 012600      MOV      (SP)+,R0      ;; POP STACK INTO R0
4952 027530 012737 027430 000024      MOV      #SPWRDN,#PWRVEC ;; SET UP THE POWER DOWN VECTOR
4953 027536 012737 000340 000026      MOV      #340,#PWRVEC+2 ;; PRIO:7
4954 027544 104400      TYPE
4955 027546 031124      $PWRMG: .WORD  POWMES      ;; REPORT THE POWER FAILURE
4956 027550 012716      MOV      (PC)+,(SP)    ;; POWER FAIL MESSAGE POINTER
4957 027552 027572      $PWRAD: .WORD  DISPATCH    ;; RESTART AT DISPATCH
4958 027554 042766 000020 000002      BIC      #20,2(SP)     ;; RESTART ADDRESS
4959 027562 000002      RTI
4960 027564 000000      $ILLUP: HALT
4961 027566 000776      BR      -2            ;; CLEAR "I" BIT
4962 027570 000000      $$AVR6: 0
4963
4964 027572 000177 000000      DISPATCH: JMP      @RETURN    ;; THE POWER UP SEQUENCE WAS STARTED
4965
4966
4967 027576 000000      RETURN:  .WORD  0        ;; BEING RUN BEFORE POWER FAILURE
4968
4969
4970
4971
4972
4973
4974

```

;; 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:

;DATA TABLES FOR DATA RELIABILITY TESTS

;ALPHANUMERIC DECK DATA TABLE
;CARD IMAGE FORM

| ALPCD: | COLUMN | ASCII | PUNCH |
|--------|--------|-------|-------|
| 4000 | 1 | 0 | 0 |
| 14400 | 2 | 1 | 1 |
| 24200 | 3 | 2 | 2 |
| 34100 | 4 | 3 | 3 |
| 44040 | 5 | 4 | 4 |
| 54020 | 6 | 5 | 5 |
| 64010 | 7 | 6 | 6 |
| 74004 | 8 | 7 | 7 |
| 4002 | 9 | 8 | 8 |
| 4001 | 10 | 9 | 9 |
| 24202 | 11 | 0 | 0 |
| 34102 | 12 | 1 | 1 |
| 44042 | 13 | 2 | 2 |
| 54022 | 14 | 3 | 3 |
| 64012 | 15 | 4 | 4 |
| 74006 | 16 | 5 | 5 |
| 2000 | 17 | 6 | 6 |
| 12400 | 18 | 7 | 7 |
| 22200 | 19 | 8 | 8 |
| 32100 | 20 | 9 | 9 |
| 42040 | 21 | 0 | 0 |
| 52020 | 22 | 1 | 1 |
| 62010 | 23 | 2 | 2 |
| 72004 | 24 | 3 | 3 |
| 2002 | 25 | 4 | 4 |
| 2001 | 26 | 5 | 5 |
| 22202 | 27 | 6 | 6 |
| 32102 | 28 | 7 | 7 |
| 42042 | 29 | 8 | 8 |
| 52022 | 30 | 9 | 9 |
| 62012 | 31 | 0 | 0 |
| 72006 | 32 | 1 | 1 |
| 1000 | 33 | 2 | 2 |
| 11400 | 34 | 3 | 3 |
| 21200 | 35 | 4 | 4 |
| 31100 | 36 | 5 | 5 |
| 41040 | 37 | 6 | 6 |
| 51020 | 38 | 7 | 7 |
| 61010 | 39 | 8 | 8 |
| 71004 | 40 | 9 | 9 |
| 1002 | 41 | 0 | 0 |
| 1001 | 42 | 1 | 1 |
| 21202 | 43 | 2 | 2 |
| 31102 | 44 | 3 | 3 |
| 41042 | 45 | 4 | 4 |
| 51022 | 46 | 5 | 5 |
| 61012 | 47 | 6 | 6 |
| 71006 | 48 | 7 | 7 |

4975
4976
4977
4978
4979
4980
4981 027600 004000
4982 027602 014400
4983 027604 024200
4984 027606 034100
4985 027610 044040
4986 027612 054020
4987 027614 064010
4988 027616 074004
4989 027620 004002
4990 027622 004001
4991 027624 024202
4992 027626 034102
4993 027630 044042
4994 027632 054022
4995 027634 064012
4996 027636 074006
4997 027640 002000
4998 027642 012400
4999 027644 022200
5000 027646 032100
5001 027650 042040
5002 027652 052020
5003 027654 062010
5004 027656 072004
5005 027660 002002
5006 027662 002001
5007 027664 022202
5008 027666 032102
5009 027670 042042
5010 027672 052022
5011 027674 062012
5012 027676 072006
5013 027700 001000
5014 027702 011400
5015 027704 021200
5016 027706 031100
5017 027710 041040
5018 027712 051020
5019 027714 061010
5020 027716 071004
5021 027720 001002
5022 027722 001001
5023 027724 021202
5024 027726 031102
5025 027730 041042
5026 027732 051022
5027 027734 061012
5028 027736 071006

| | | |
|------|--------|--------|
| 5029 | 027740 | 000000 |
| 5030 | 027742 | 010400 |
| 5031 | 027744 | 020200 |
| 5032 | 027746 | 030100 |
| 5033 | 027750 | 040040 |
| 5034 | 027752 | 050020 |
| 5035 | 027754 | 060010 |
| 5036 | 027756 | 070004 |
| 5037 | 027760 | 000002 |
| 5038 | 027762 | 000001 |
| 5039 | 027764 | 020202 |
| 5040 | 027766 | 030102 |
| 5041 | 027770 | 040042 |
| 5042 | 027772 | 050022 |
| 5043 | 027774 | 060012 |
| 5044 | 027776 | 070006 |
| 5045 | 030000 | 004000 |
| 5046 | 030002 | 014400 |
| 5047 | 030004 | 024200 |
| 5048 | 030006 | 034100 |
| 5049 | 030010 | 044040 |
| 5050 | 030012 | 054020 |
| 5051 | 030014 | 064010 |
| 5052 | 030016 | 074004 |
| 5053 | 030020 | 004002 |
| 5054 | 030022 | 004001 |
| 5055 | 030024 | 024202 |
| 5056 | 030026 | 034102 |
| 5057 | 030030 | 044042 |
| 5058 | 030032 | 054022 |
| 5059 | 030034 | 064012 |
| 5060 | 030036 | 074006 |

ALPEND: 74006

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

| | | |
|------|--------|-----|
| 5061 | | |
| 5062 | | |
| 5063 | | |
| 5064 | | |
| 5065 | 030040 | 200 |
| 5066 | 030041 | 201 |
| 5067 | 030042 | 202 |
| 5068 | 030043 | 203 |
| 5069 | 030044 | 204 |
| 5070 | 030045 | 205 |
| 5071 | 030046 | 206 |
| 5072 | 030047 | 207 |
| 5073 | 030050 | 210 |
| 5074 | 030051 | 220 |
| 5075 | 030052 | 212 |
| 5076 | 030053 | 213 |
| 5077 | 030054 | 214 |
| 5078 | 030055 | 215 |
| 5079 | 030056 | 216 |
| 5080 | 030057 | 217 |
| 5081 | 030060 | 100 |
| 5082 | 030061 | 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 |

| | | |
|----|-------|-------|
| 49 | SPACE | BLANK |
| 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 | | |

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

| | | |
|------|--------|-----|
| 5137 | 030150 | 210 |
| 5138 | 030151 | 220 |
| 5139 | 030152 | 212 |
| 5140 | 030153 | 213 |
| 5141 | 030154 | 214 |
| 5142 | 030155 | 215 |
| 5143 | 030156 | 216 |
| 5144 | 030157 | 217 |

| | |
|---------------|-----|
| .BYTE | 210 |
| .BYTE | 220 |
| .BYTE | 212 |
| .BYTE | 213 |
| .BYTE | 214 |
| .BYTE | 215 |
| .BYTE | 216 |
| ALPENP: .BYTE | 217 |

| | | |
|-----|---|----|
| :73 | H | 12 |
| :74 | l | 12 |
| :75 | l | 12 |
| :76 | l | 12 |
| :77 | l | 12 |
| :78 | (| 12 |
| :79 | + | 12 |
| :80 | : | 12 |

;BINARY DECK DATA TABLE

| | | |
|------|--------|--------|
| 5145 | | |
| 5146 | | |
| 5147 | | |
| 5148 | 030160 | 000000 |
| 5149 | 030162 | 000001 |
| 5150 | 030164 | 000002 |
| 5151 | 030166 | 070004 |
| 5152 | 030170 | 060010 |
| 5153 | 030172 | 050020 |
| 5154 | 030174 | 040040 |
| 5155 | 030176 | 030100 |
| 5156 | 030200 | 020200 |
| 5157 | 030202 | 010400 |
| 5158 | 030204 | 001000 |
| 5159 | 030206 | 002000 |
| 5160 | 030210 | 004000 |
| 5161 | 030212 | 171111 |
| 5162 | 030214 | 172222 |
| 5163 | 030216 | 173333 |
| 5164 | 030220 | 174444 |
| 5165 | 030222 | 175555 |
| 5166 | 030224 | 176666 |
| 5167 | 030226 | 177777 |
| 5168 | 030230 | 061010 |
| 5169 | 030232 | 161212 |
| 5170 | 030234 | 171313 |
| 5171 | 030236 | 171414 |
| 5172 | 030240 | 171515 |
| 5173 | 030242 | 171616 |
| 5174 | 030244 | 171717 |
| 5175 | 030246 | 052020 |
| 5176 | 030250 | 172121 |
| 5177 | 030252 | 172323 |
| 5178 | 030254 | 172424 |
| 5179 | 030256 | 172525 |
| 5180 | 030260 | 172626 |
| 5181 | 030262 | 172727 |
| 5182 | 030264 | 173030 |
| 5183 | 030266 | 173131 |
| 5184 | 030270 | 173232 |
| 5185 | 030272 | 173434 |
| 5186 | 030274 | 173535 |
| 5187 | 030276 | 173636 |
| 5188 | 030300 | 173737 |
| 5189 | 030302 | 044040 |
| 5190 | 030304 | 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 | |

| | | | | |
|------|--------|--------|--------|----|
| 5191 | 030306 | 164242 | 164242 | 44 |
| 5192 | 030310 | 174343 | 174343 | 45 |
| 5193 | 030312 | 174545 | 174545 | 46 |
| 5194 | 030314 | 174646 | 174646 | 47 |
| 5195 | 030316 | 174747 | 174747 | 48 |
| 5196 | 030320 | 165050 | 165050 | 49 |
| 5197 | 030322 | 175151 | 175151 | 50 |
| 5198 | 030324 | 165252 | 165252 | 51 |
| 5199 | 030325 | 175353 | 175353 | 52 |
| 5200 | 030330 | 175454 | 175454 | 53 |
| 5201 | 030332 | 175656 | 175656 | 54 |
| 5202 | 030334 | 175757 | 175757 | 55 |
| 5203 | 030336 | 156060 | 156060 | 56 |
| 5204 | 030340 | 176161 | 176161 | 57 |
| 5205 | 030342 | 176262 | 176262 | 58 |
| 5206 | 030344 | 176363 | 176363 | 59 |
| 5207 | 030346 | 176464 | 176464 | 60 |
| 5208 | 030350 | 176565 | 176565 | 61 |
| 5209 | 030352 | 176767 | 176767 | 62 |
| 5210 | 030354 | 177070 | 177070 | 63 |
| 5211 | 030356 | 177171 | 177171 | 64 |
| 5212 | 030360 | 177272 | 177272 | 65 |
| 5213 | 030362 | 177373 | 177373 | 66 |
| 5214 | 030364 | 177474 | 177474 | 67 |
| 5215 | 030366 | 177575 | 177575 | 68 |
| 5216 | 030370 | 177676 | 177676 | 69 |
| 5217 | 030372 | 030101 | 30101 | 70 |
| 5218 | 030374 | 020202 | 20202 | 71 |
| 5219 | 030376 | 130303 | 130303 | 72 |
| 5220 | 030400 | 170404 | 170404 | 73 |
| 5221 | 030402 | 170505 | 170505 | 74 |
| 5222 | 030404 | 170606 | 170606 | 75 |
| 5223 | 030406 | 170707 | 170707 | 76 |
| 5224 | 030410 | 163210 | 163210 | 77 |
| 5225 | 030412 | 170123 | 170123 | 78 |
| 5226 | 030414 | 177654 | 177654 | 79 |
| 5227 | 030416 | 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 |
|------|--------|-----|--------|-------|-------|
| 5233 | 030420 | 000 | 1 | SPACE | BLANK |
| 5234 | 030421 | 020 | 2 | 9 | 9 |
| 5235 | 030422 | 010 | 3 | 8 | 8 |
| 5236 | 030423 | 007 | 4 | 7 | 7 |
| 5237 | 030424 | 006 | 5 | 6 | 6 |
| 5238 | 030425 | 005 | 6 | 5 | 5 |
| 5239 | 030426 | 004 | 7 | 4 | 4 |
| 5240 | 030427 | 003 | 8 | 3 | 3 |
| 5241 | 030430 | 002 | 9 | 2 | 2 |
| 5242 | 030431 | 001 | 10 | 1 | 1 |
| 5243 | 030432 | 040 | 11 | 0 | 0 |
| 5244 | 030433 | 100 | 12 | 0 | 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

1
2
3
4
5
6
7
8
9
10
11
12

| | | | | | | | |
|------|--------|-----|-------|-----|----|---|----|
| 5245 | 030434 | 200 | .BYTE | 200 | 13 | | |
| 5246 | 030435 | 067 | .BYTE | 67 | 14 | 8 | 12 |
| 5247 | 030436 | 117 | .BYTE | 117 | 15 | | |
| 5248 | 030437 | 177 | .BYTE | 177 | 16 | | |
| 5249 | 030440 | 207 | .BYTE | 207 | 17 | | |
| 5250 | 030441 | 267 | .BYTE | 267 | 18 | | |
| 5251 | 030442 | 317 | .BYTE | 317 | 19 | | |
| 5252 | 030443 | 377 | .BYTE | 377 | 20 | | |
| 5253 | 030444 | 046 | .BYTE | 46 | 21 | | |
| 5254 | 030445 | 056 | .BYTE | 56 | 22 | | |
| 5255 | 030446 | 077 | .BYTE | 77 | 23 | | |
| 5256 | 030447 | 047 | .BYTE | 47 | 24 | | |
| 5257 | 030450 | 067 | .BYTE | 67 | 25 | | |
| 5258 | 030451 | 057 | .BYTE | 57 | 26 | | |
| 5259 | 030452 | 077 | .BYTE | 77 | 27 | | |
| 5260 | 030453 | 105 | .BYTE | 105 | 28 | | |
| 5261 | 030454 | 127 | .BYTE | 127 | 29 | | |
| 5262 | 030455 | 137 | .BYTE | 137 | 30 | | |
| 5263 | 030456 | 107 | .BYTE | 107 | 31 | | |
| 5264 | 030457 | 127 | .BYTE | 127 | 32 | | |
| 5265 | 030460 | 117 | .BYTE | 117 | 33 | | |
| 5266 | 030461 | 137 | .BYTE | 137 | 34 | | |
| 5267 | 030462 | 147 | .BYTE | 147 | 35 | | |
| 5268 | 030463 | 167 | .BYTE | 167 | 36 | | |
| 5269 | 030464 | 157 | .BYTE | 157 | 37 | | |
| 5270 | 030465 | 147 | .BYTE | 147 | 38 | | |
| 5271 | 030466 | 167 | .BYTE | 167 | 39 | | |
| 5272 | 030467 | 157 | .BYTE | 157 | 40 | | |
| 5273 | 030470 | 177 | .BYTE | 177 | 41 | | |
| 5274 | 030471 | 204 | .BYTE | 204 | 42 | | |
| 5275 | 030472 | 227 | .BYTE | 227 | 43 | | |
| 5276 | 030473 | 216 | .BYTE | 216 | 44 | | |
| 5277 | 030474 | 237 | .BYTE | 237 | 45 | | |
| 5278 | 030475 | 227 | .BYTE | 227 | 46 | | |
| 5279 | 030476 | 217 | .BYTE | 217 | 47 | | |
| 5280 | 030477 | 237 | .BYTE | 237 | 48 | | |
| 5281 | 030500 | 246 | .BYTE | 246 | 49 | | |
| 5282 | 030501 | 267 | .BYTE | 267 | 50 | | |
| 5283 | 030502 | 256 | .BYTE | 256 | 51 | | |
| 5284 | 030503 | 277 | .BYTE | 277 | 52 | | |
| 5285 | 030504 | 247 | .BYTE | 247 | 53 | | |
| 5286 | 030505 | 257 | .BYTE | 257 | 54 | | |
| 5287 | 030506 | 277 | .BYTE | 277 | 55 | | |
| 5288 | 030507 | 305 | .BYTE | 305 | 56 | | |
| 5289 | 030510 | 327 | .BYTE | 327 | 57 | | |
| 5290 | 030511 | 317 | .BYTE | 317 | 58 | | |
| 5291 | 030512 | 337 | .BYTE | 337 | 59 | | |
| 5292 | 030513 | 307 | .BYTE | 307 | 60 | | |
| 5293 | 030514 | 327 | .BYTE | 327 | 61 | | |
| 5294 | 030515 | 337 | .BYTE | 337 | 62 | | |
| 5295 | 030516 | 347 | .BYTE | 347 | 63 | | |
| 5296 | 030517 | 367 | .BYTE | 367 | 64 | | |
| 5297 | 030520 | 357 | .BYTE | 357 | 65 | | |
| 5298 | 030521 | 377 | .BYTE | 377 | 66 | | |

| | | | | | | | |
|------|--------|--------|--------|--------|---------|-----------------------------------------------------------|-----|
| 5299 | 030522 | 347 | | | .BYTE | 347 | :67 |
| 5300 | 030523 | 367 | | | .BYTE | 367 | :68 |
| 5301 | 030524 | 357 | | | .BYTE | 357 | :69 |
| 5302 | 030525 | 023 | | | .BYTE | 23 | :70 |
| 5303 | 030526 | 012 | | | .BYTE | 12 | :71 |
| 5304 | 030527 | 033 | | | .BYTE | 33 | :72 |
| 5305 | 030530 | 007 | | | .BYTE | 7 | :73 |
| 5306 | 030531 | 027 | | | .BYTE | 27 | :74 |
| 5307 | 030532 | 017 | | | .BYTE | 17 | :75 |
| 5308 | 030533 | 037 | | | .BYTE | 37 | :76 |
| 5309 | 030534 | 146 | | | .BYTE | 146 | :77 |
| 5310 | 030535 | 037 | | | .BYTE | 37 | :78 |
| 5311 | 030536 | 347 | | | .BYTE | 347 | :79 |
| 5312 | 030537 | 237 | | | .BYTE | 237 | :80 |
| 5313 | | | | | | | |
| 5314 | 030540 | 020040 | 020040 | 040 | SPACE: | .ASCIZ / / / | |
| 5315 | 030545 | 040 | 000040 | | | .ASCIZ / / / | |
| 5316 | 030550 | 005012 | 012 | | CRLF: | .ASCIZ <12><12><12> | |
| 5317 | 030553 | 015 | 000012 | | | .ASCIZ <15><12> | |
| 5318 | | | | | | | |
| 5319 | 030556 | 005015 | 046412 | 044501 | STMES: | .ASCIZ <15><12><12>"MAINDEC-11-DZCDB REV. 8"<15><12> | |
| 5320 | 030564 | 042116 | 041505 | 030455 | | | |
| 5321 | 030572 | 026461 | 055104 | 042103 | | | |
| 5322 | 030600 | 020102 | 020040 | 051040 | | | |
| 5323 | 030606 | 053105 | 020056 | 006502 | | | |
| 5324 | 030614 | 000012 | | | | | |
| 5325 | | | | | | | |
| 5326 | 030616 | 005015 | 052123 | 051101 | STADDR: | .ASCII <15><12>"STARTING ADDRESSES ARE:" | |
| 5327 | 030624 | 044524 | 043516 | 040440 | | | |
| 5328 | 030632 | 042104 | 042522 | 051523 | | | |
| 5329 | 030640 | 051505 | 040440 | 042522 | | | |
| 5330 | 030646 | 072 | | | | | |
| 5331 | 030647 | 015 | 031012 | 030060 | | .ASCII <15><12>"200 = INSTRUCTION AND DATA TEST" | |
| 5332 | 030654 | 036440 | 044440 | 051516 | | | |
| 5333 | 030662 | 051124 | 041525 | 044524 | | | |
| 5334 | 030670 | 047117 | 040440 | 042116 | | | |
| 5335 | 030676 | 042040 | 052101 | 020101 | | | |
| 5336 | 030704 | 042524 | 052123 | | | | |
| 5337 | 030710 | 005015 | 030462 | 020060 | | .ASCII <15><12>"210 = ERROR FUNCTION TEST (M-1000/M-200)" | |
| 5338 | 030716 | 020075 | 051105 | 047522 | | | |
| 5339 | 030724 | 020122 | 052506 | 041516 | | | |
| 5340 | 030732 | 044524 | 047117 | 052040 | | | |
| 5341 | 030740 | 051505 | 020124 | 046450 | | | |
| 5342 | 030746 | 030455 | 030060 | 027460 | | | |
| 5343 | 030754 | 026515 | 030062 | 024460 | | | |
| 5344 | 030762 | 005015 | 031062 | 020060 | | .ASCII <15><12>"220 = SINGLE TEST LOOP" | |
| 5345 | 030770 | 020075 | 044523 | 043516 | | | |
| 5346 | 030776 | 042514 | 052040 | 051505 | | | |
| 5347 | 031004 | 020124 | 047514 | 050117 | | | |
| 5348 | 031012 | 005015 | 032062 | 020060 | | .ASCII <15><12>"240 = READ SINGLE DATA PATTERN TEST" | |
| 5349 | 031020 | 020075 | 042522 | 042101 | | | |
| 5350 | 031026 | 051440 | 047111 | 046107 | | | |
| 5351 | 031034 | 020105 | 040504 | 040524 | | | |
| 5352 | 031042 | 050040 | 052101 | 042524 | | | |

| | | | | | |
|------|--------|--------|--------|--------|----------------------------------------------------------------------|
| 5353 | 031050 | 047122 | 052040 | 051505 | |
| 5354 | 031056 | 124 | | | |
| 5355 | 031057 | 015 | 031012 | 030065 | .ASCIZ <15><12>"250 = ERROR FUNCTION TEST (M-1200)" |
| 5356 | 031064 | 036440 | 042440 | 051122 | |
| 5357 | 031072 | 051117 | 043040 | 047125 | |
| 5358 | 031100 | 052103 | 047511 | 020116 | |
| 5359 | 031106 | 042524 | 052123 | 024040 | |
| 5360 | 031114 | 026515 | 031061 | 030060 | |
| 5361 | 031122 | 000051 | | | |
| 5362 | | | | | |
| 5363 | 031124 | 005015 | 050012 | 053517 | POWME: .ASCII <15><12><12>"POWER FAILURE OCCURRED - WILL ATTEMPT TO" |
| 5364 | 031132 | 051105 | 043040 | 044501 | |
| 5365 | 031140 | 052514 | 042522 | 047440 | |
| 5366 | 031146 | 041503 | 051125 | 042522 | |
| 5367 | 031154 | 020104 | 020055 | 044527 | |
| 5368 | 031162 | 046114 | 040440 | 052124 | |
| 5369 | 031170 | 046505 | 052120 | 052040 | |
| 5370 | 031176 | 117 | | | |
| 5371 | 031177 | 015 | 051012 | 051505 | .ASCIZ <15><12>"RESTART SECTION OF PROGRAM FORMERLY IN USE"<15><12> |
| 5372 | 031204 | 040524 | 052122 | 051440 | |
| 5373 | 031212 | 041505 | 044524 | 047117 | |
| 5374 | 031220 | 047440 | 020106 | 051120 | |
| 5375 | 031226 | 043517 | 040522 | 020115 | |
| 5376 | 031234 | 047506 | 046522 | 051105 | |
| 5377 | 031242 | 054514 | 044440 | 020116 | |
| 5378 | 031250 | 051525 | 006505 | 000012 | |
| 5379 | | | | | |
| 5380 | 031256 | 005015 | 047105 | 042524 | MLOGIC: .ASCIZ <15><12>"ENTERING LOGIC TESTS" |
| 5381 | 031264 | 044522 | 043516 | 046040 | |
| 5382 | 031272 | 043517 | 041511 | 052040 | |
| 5383 | 031300 | 051505 | 051524 | 000 | |
| 5384 | | | | | |
| 5385 | 031305 | 015 | 042412 | 052116 | MDATA: .ASCIZ <15><12>"ENTERING DATA TESTS" |
| 5386 | 031312 | 051105 | 047111 | 020107 | |
| 5387 | 031320 | 040504 | 040524 | 052040 | |
| 5388 | 031326 | 051505 | 051524 | 000 | |
| 5389 | | | | | |
| 5390 | 031333 | 015 | 042412 | 052116 | M1200E: .ASCIZ <15><12>"ENTERING M1200 ERROR FUNCTION TESTS" |
| 5391 | 031340 | 051105 | 047111 | 020107 | |
| 5392 | 031346 | 030515 | 030062 | 020060 | |
| 5393 | 031354 | 051105 | 047522 | 020122 | |
| 5394 | 031362 | 052506 | 041516 | 044524 | |
| 5395 | 031370 | 047117 | 052040 | 051505 | |
| 5396 | 031376 | 051524 | 000 | | |
| 5397 | | | | | |
| 5398 | 031401 | 015 | 042412 | 052116 | M1000E: .ASCIZ <15><12>"ENTERING M1000/M200 ERROR FUNCTION TESTS" |
| 5399 | 031406 | 051105 | 047111 | 020107 | |
| 5400 | 031414 | 030515 | 030060 | 027460 | |
| 5401 | 031422 | 031115 | 030060 | 042440 | |
| 5402 | 031430 | 051122 | 051117 | 043040 | |
| 5403 | 031436 | 047125 | 052103 | 047511 | |
| 5404 | 031444 | 020116 | 042524 | 052123 | |
| 5405 | 031452 | 000123 | | | |
| 5406 | | | | | |

| | | | | | |
|------|--------|--------|--------|--------|-------------------------------------------------------------------|
| 5407 | 031454 | 005015 | 047105 | 042524 | ML00P: .ASCIZ <15><12>"ENTERING A LOOP ON TEST SELECTED BY USER" |
| 5408 | 031462 | 044522 | 043516 | 040440 | |
| 5409 | 031470 | 046040 | 047517 | 020120 | |
| 5410 | 031476 | 047117 | 052040 | 051505 | |
| 5411 | 031504 | 020124 | 042523 | 042514 | |
| 5412 | 031512 | 052103 | 042105 | 041040 | |
| 5413 | 031520 | 020131 | 051525 | 051105 | |
| 5414 | 031526 | 000 | | | |
| 5415 | | | | | |
| 5416 | 031527 | 015 | 042412 | 052116 | MPATS: .ASCIZ <15><12>"ENTERING SINGLE DATA PATTERN TESTING" |
| 5417 | 031534 | 051105 | 047111 | 020107 | |
| 5418 | 031542 | 044523 | 043516 | 042514 | |
| 5419 | 031550 | 042040 | 052101 | 020101 | |
| 5420 | 031556 | 040520 | 052124 | 051105 | |
| 5421 | 031564 | 020116 | 042524 | 052123 | |
| 5422 | 031572 | 047111 | 000107 | | |
| 5423 | | | | | |
| 5424 | 031576 | 005015 | 051120 | 051505 | MSG1: .ASCIZ <15><12>/PRESS CARD READER 'RESET' / |
| 5425 | 031604 | 020123 | 040503 | 042122 | |
| 5426 | 031612 | 051040 | 040505 | 042504 | |
| 5427 | 031620 | 020122 | 051047 | 051505 | |
| 5428 | 031626 | 052105 | 000047 | | |
| 5429 | 031632 | 005015 | 044124 | 047105 | MSG2: .ASCIZ <15><12>/THEN CONTINUE DIAGNOSTIC FROM CPU CONSOLE / |
| 5430 | 031640 | 041440 | 047117 | 044524 | |
| 5431 | 031646 | 052516 | 020105 | 044504 | |
| 5432 | 031654 | 043501 | 047516 | 052123 | |
| 5433 | 031662 | 041511 | 043040 | 047522 | |
| 5434 | 031670 | 020115 | 050103 | 020125 | |
| 5435 | 031676 | 047503 | 051516 | 046117 | |
| 5436 | 031704 | 000105 | | | |
| 5437 | 031706 | 005015 | 051120 | 051505 | MSG3: .ASCIZ <15><12>/PRESS CARD READER 'STOP' / |
| 5438 | 031714 | 020123 | 040503 | 042122 | |
| 5439 | 031722 | 051040 | 040505 | 042504 | |
| 5440 | 031730 | 020122 | 051447 | 047524 | |
| 5441 | 031736 | 023520 | 000 | | |
| 5442 | 031741 | 015 | 052012 | 042510 | MSG4: .ASCIZ <15><12>/THE INTERRUPT LEVEL WAS / |
| 5443 | 031746 | 044440 | 052116 | 051105 | |
| 5444 | 031754 | 052522 | 052120 | 046040 | |
| 5445 | 031762 | 053105 | 046105 | 053440 | |
| 5446 | 031770 | 051501 | 000040 | | |
| 5447 | 031774 | 005015 | 042522 | 047515 | MSG5: .ASCIZ <15><12>/REMOVE ALL CARDS FROM THE INPUT HOPPER / |
| 5448 | 032002 | 042526 | 040440 | 046114 | |
| 5449 | 032010 | 041440 | 051101 | 051504 | |
| 5450 | 032016 | 043040 | 047522 | 020115 | |
| 5451 | 032024 | 044124 | 020105 | 047111 | |
| 5452 | 032032 | 052520 | 020124 | 047510 | |
| 5453 | 032040 | 050120 | 051105 | 000 | |
| 5454 | 032045 | 015 | 051012 | 051505 | MSG6: .ASCIZ <15><12>/RESTORE CARDS TO THE INPUT HOPPER / |
| 5455 | 032052 | 047524 | 042522 | 041440 | |
| 5456 | 032060 | 051101 | 051504 | 052040 | |
| 5457 | 032066 | 020117 | 044124 | 020105 | |
| 5458 | 032074 | 047111 | 052520 | 020124 | |
| 5459 | 032102 | 047510 | 050120 | 051105 | |
| 5460 | 032110 | 000 | | | |

| | | | | | | |
|------|--------|--------|--------|--------|------------------------------------------------------------|-----------------------------------------------------------------------|
| 5461 | 032111 | 015 | 030412 | 020056 | MSG7: | ASCII <15><12>/1. PULL OUTPUT STACKER PRESSURE ARM DOWN / |
| 5462 | 032116 | 052520 | 046114 | 047440 | | |
| 5463 | 032124 | 052125 | 052520 | 020124 | | |
| 5464 | 032132 | 052123 | 041501 | 042513 | | |
| 5465 | 032140 | 020122 | 051120 | 051505 | | |
| 5466 | 032146 | 052523 | 042522 | 040440 | | |
| 5467 | 032154 | 046522 | 042040 | 053517 | | |
| 5468 | 032162 | 020118 | | | | |
| 5469 | 032164 | 005015 | 047125 | 044524 | .ASCII <15><12>/UNTIL HOPPER CHECK LIGHT COMES ON/ | |
| 5470 | 032172 | 020114 | 047510 | 050120 | | |
| 5471 | 032200 | 051105 | 041440 | 042510 | | |
| 5472 | 032206 | 045503 | 046040 | 043511 | | |
| 5473 | 032214 | 052110 | 041440 | 046517 | | |
| 5474 | 032222 | 051505 | 047440 | 116 | .ASCIZ <15><12><12>/2. RELEASE STACKER PRESS. ARM/ | |
| 5475 | 032227 | 015 | 005012 | 027062 | | |
| 5476 | 032234 | 051040 | 046105 | 040505 | | |
| 5477 | 032242 | 042523 | 051440 | 040524 | | |
| 5478 | 032250 | 045503 | 051105 | 050040 | | |
| 5479 | 032256 | 042522 | 051523 | 020056 | | |
| 5480 | 032264 | 051101 | 000115 | | | |
| 5481 | 032270 | 005015 | 044012 | 046117 | MSG8: | .ASCII <15><12><12>/HOLD DOWN THE SWITCH UNDER THE CAP OF THE INPUT / |
| 5482 | 032276 | 020104 | 047504 | 047127 | | |
| 5483 | 032304 | 052040 | 042510 | 051440 | | |
| 5484 | 032312 | 044527 | 041524 | 020110 | | |
| 5485 | 032320 | 047125 | 042504 | 020122 | | |
| 5486 | 032326 | 044124 | 020105 | 040503 | | |
| 5487 | 032334 | 020120 | 043117 | 052040 | | |
| 5488 | 032342 | 042510 | 044440 | 050116 | | |
| 5489 | 032350 | 052125 | 040 | | | |
| 5490 | 032353 | 110 | 050117 | 042520 | .ASCIZ /HOPPER/ | |
| 5491 | 032360 | 000122 | | | | |
| 5492 | 032362 | 005015 | 042524 | 051101 | MSG9: | .ASCII <15><12>/TEAR OFF A PIECE OF CARD AND SLIP IT IN/ |
| 5493 | 032370 | 047440 | 043106 | 040440 | | |
| 5494 | 032376 | 050040 | 042511 | 042503 | | |
| 5495 | 032404 | 047440 | 020106 | 040503 | | |
| 5496 | 032412 | 042122 | 040440 | 042116 | | |
| 5497 | 032420 | 051440 | 044514 | 020120 | | |
| 5498 | 032426 | 052111 | 044440 | 116 | | |
| 5499 | 032433 | 015 | 043012 | 047522 | .ASCII <15><12>/FRONT OF THE PHOTOCCELL/ | |
| 5500 | 032440 | 052116 | 047440 | 020106 | | |
| 5501 | 032446 | 044124 | 020105 | 044120 | | |
| 5502 | 032454 | 052117 | 041517 | 046105 | | |
| 5503 | 032462 | 114 | | | | |
| 5504 | 032463 | 015 | 012 | | .ASCII <15><12> | |
| 5505 | 032465 | 015 | 041412 | 052501 | .ASCII <15><12>/CAUTION:MOVE YOUR FINGERS AWAY FROM THIS / | |
| 5506 | 032472 | 044524 | 047117 | 046472 | | |
| 5507 | 032500 | 053117 | 020105 | 047531 | | |
| 5508 | 032506 | 051125 | 043040 | 047111 | | |
| 5509 | 032514 | 042507 | 051522 | 040440 | | |
| 5510 | 032522 | 040527 | 020131 | 051106 | | |
| 5511 | 032530 | 046517 | 052040 | 044510 | | |
| 5512 | 032536 | 020123 | | | | |
| 5513 | 032540 | 051101 | 040505 | 041040 | .ASCIZ /AREA BEFORE CONTINUING!// | |
| 5514 | 032546 | 043105 | 051117 | 020105 | | |

| | | | | | |
|------|--------|--------|--------|--------|------------------------------------------------------------------------|
| 5515 | 032554 | 047503 | 052116 | 047111 | |
| 5516 | 032562 | 044525 | 043516 | 000041 | |
| 5517 | 032570 | 005015 | 020412 | 020441 | MSG10: .ASCIZ <15><12><12>/!!! WRONG RESPONSE TRY AGAIN !!!/ |
| 5518 | 032576 | 053440 | 047522 | 043516 | |
| 5519 | 032604 | 051040 | 051505 | 047520 | |
| 5520 | 032612 | 051516 | 020105 | 051124 | |
| 5521 | 032620 | 020131 | 043501 | 044501 | |
| 5522 | 032626 | 020116 | 020441 | 000041 | |
| 5523 | 032634 | 005015 | 047510 | 042114 | MSG11: .ASCIZ <15><12>/HOLD THE OUTPUT STACKER GATE OPEN. THEN/ |
| 5524 | 032642 | 052040 | 042510 | 047440 | |
| 5525 | 032650 | 052125 | 052520 | 020124 | |
| 5526 | 032656 | 052123 | 041501 | 042513 | |
| 5527 | 032664 | 020122 | 040507 | 042524 | |
| 5528 | 032672 | 047440 | 042520 | 027116 | |
| 5529 | 032700 | 052040 | 042510 | 000116 | |
| 5530 | 032706 | 005015 | 042040 | 051101 | MSG12: .ASCII <15><12>/ DARK-LIGHT CHECK / |
| 5531 | 032714 | 026513 | 044514 | 044107 | |
| 5532 | 032722 | 020124 | 044103 | 041505 | |
| 5533 | 032730 | 020113 | | | |
| 5534 | 032732 | 005015 | 027111 | 027105 | .ASCII <15><12>/I.E. - TEAR A CORNER OFF A CARD AND PLACE IT/ |
| 5535 | 032740 | 026440 | 052040 | 040505 | |
| 5536 | 032746 | 020122 | 020101 | 047503 | |
| 5537 | 032754 | 047122 | 051105 | 047440 | |
| 5538 | 032762 | 043105 | 040440 | 041440 | |
| 5539 | 032770 | 051101 | 020104 | 047101 | |
| 5540 | 032776 | 020104 | 046120 | 041501 | |
| 5541 | 033004 | 020105 | 052111 | | |
| 5542 | 033010 | 005015 | 052101 | 052040 | .ASCIZ <15><12> /AT THE BOTTOM OF THE INPUT STACK/ |
| 5543 | 033016 | 042510 | 041040 | 052117 | |
| 5544 | 033024 | 047524 | 020115 | 043117 | |
| 5545 | 033032 | 052040 | 042510 | 044440 | |
| 5546 | 033040 | 050116 | 052125 | 051440 | |
| 5547 | 033046 | 040524 | 045503 | 000 | |
| 5548 | 033053 | 015 | 042012 | 041505 | MSG13: .ASCIZ <15><12>/DECK CARD NUM CARD COL SHB WAS/ |
| 5549 | 033060 | 020113 | 020040 | 020040 | |
| 5550 | 033066 | 040503 | 042122 | 047040 | |
| 5551 | 033074 | 046525 | 020040 | 020040 | |
| 5552 | 033102 | 041440 | 051101 | 020104 | |
| 5553 | 033110 | 047503 | 020114 | 020040 | |
| 5554 | 033116 | 044123 | 020102 | 020040 | |
| 5555 | 033124 | 020040 | 040527 | 000123 | |
| 5556 | 033132 | 005015 | 046101 | 044120 | MSG14: .ASCIZ <15><12>/ALPHA / |
| 5557 | 033140 | 020101 | 000 | | |
| 5558 | 033143 | 015 | 041012 | 047111 | MSG15: .ASCIZ <15><12>/BINARY/ |
| 5559 | 033150 | 051101 | 000131 | | |
| 5560 | 033154 | 005015 | 044502 | 020124 | MSG16: .ASCIZ <15><12>/BIT 15 WAS SET/ |
| 5561 | 033162 | 032461 | 053440 | 051501 | |
| 5562 | 033170 | 051440 | 052105 | 000 | |
| 5563 | 033175 | 015 | 051012 | 046505 | MSG17: .ASCII <15><12>/REMEDY THE CONDITION BY RELOADING INPUT HOPPER/ |
| 5564 | 033202 | 042105 | 020131 | 044124 | |
| 5565 | 033210 | 020105 | 047503 | 042116 | |
| 5566 | 033216 | 052111 | 047511 | 020116 | |
| 5567 | 033224 | 054502 | 051040 | 046105 | |
| 5568 | 033232 | 040517 | 044504 | 043516 | |

| | | | | | |
|------|--------|--------|--------|--------|--------------------------------------------------------------------|
| 5569 | 033240 | 044440 | 050116 | 052125 | |
| 5570 | 033246 | 044040 | 050117 | 042520 | |
| 5571 | 033254 | 122 | | | |
| 5572 | 033255 | 015 | 053412 | 052111 | .ASCII <15><12>/WITH CARD DECK - PRESS 'RESET' BUTTON/ |
| 5573 | 033262 | 020110 | 040503 | 042122 | |
| 5574 | 033270 | 042040 | 041505 | 020113 | |
| 5575 | 033276 | 020055 | 051120 | 051505 | |
| 5576 | 033304 | 020123 | 051047 | 051505 | |
| 5577 | 033312 | 052105 | 020047 | 052502 | |
| 5578 | 033320 | 052124 | 047117 | | |
| 5579 | 033324 | 005015 | 047117 | 041440 | .ASCIZ <15><12>/ON CARD READER AND 'CONTINUE' SWITCH ON CPU PANEL/ |
| 5580 | 033332 | 051101 | 020104 | 042522 | |
| 5581 | 033340 | 042101 | 051105 | 040440 | |
| 5582 | 033346 | 042116 | 023440 | 047503 | |
| 5583 | 033354 | 052116 | 047111 | 042525 | |
| 5584 | 033362 | 020047 | 053523 | 052111 | |
| 5585 | 033370 | 044103 | 047440 | 020116 | |
| 5586 | 033376 | 050103 | 020125 | 040520 | |
| 5587 | 033404 | 042516 | 000114 | | |
| 5588 | 033410 | 005015 | 040503 | 042122 | MSG18: .ASCIZ <15><12>/CARD READER IS OFF-LINE/ |
| 5589 | 033416 | 051040 | 040505 | 042504 | |
| 5590 | 033424 | 020122 | 051511 | 047440 | |
| 5591 | 033432 | 043106 | 046055 | 047111 | |
| 5592 | 033440 | 000105 | | | |
| 5593 | 033442 | 005015 | 020040 | 041440 | MSG19: .ASCIZ <15><12>/ COL. WAS CARD NUM. ERRORS/ |
| 5594 | 033450 | 046117 | 020056 | 020040 | |
| 5595 | 033456 | 040527 | 020123 | 020040 | |
| 5596 | 033464 | 040503 | 042122 | 047040 | |
| 5597 | 033472 | 046525 | 020056 | 042440 | |
| 5598 | 033500 | 051122 | 051117 | 000123 | |
| 5599 | 033506 | 005015 | 052520 | 020124 | MSG20: .ASCIZ <15><12>/PUT ANY TWO CARDS IN INPUT HOPPER/ |
| 5600 | 033514 | 047101 | 020131 | 053524 | |
| 5601 | 033522 | 020117 | 040503 | 042122 | |
| 5602 | 033530 | 020123 | 047111 | 044440 | |
| 5603 | 033536 | 050116 | 052125 | 044040 | |
| 5604 | 033544 | 050117 | 042520 | 000122 | |
| 5605 | 033552 | 005015 | 051120 | 051505 | MSG21: .ASCIZ <15><12>/PRESS END OF FILE BUTTON/ |
| 5606 | 033560 | 020123 | 047105 | 020104 | |
| 5607 | 033566 | 043117 | 043040 | 046111 | |
| 5608 | 033574 | 020105 | 052502 | 052124 | |
| 5609 | 033602 | 047117 | 000 | | |
| 5610 | 033605 | 015 | 030412 | 020056 | MSG22: .ASCII <15><12>/1. PUT 'HALT'SW. DOWN AND PRESS/ |
| 5611 | 033612 | 052520 | 020124 | 044047 | |
| 5612 | 033620 | 046101 | 023524 | 053523 | |
| 5613 | 033626 | 020056 | 047504 | 047127 | |
| 5614 | 033634 | 040440 | 042116 | 050040 | |
| 5615 | 033642 | 042522 | 051523 | | |
| 5616 | 033646 | 005015 | 020040 | 041440 | .ASCII <15><12>/ CONTINUE ON THE CONSOLE UNTIL/ |
| 5617 | 033654 | 047117 | 044524 | 052516 | |
| 5618 | 033662 | 020105 | 047117 | 052040 | |
| 5619 | 033670 | 042510 | 041440 | 047117 | |
| 5620 | 033676 | 047523 | 042514 | 052440 | |
| 5621 | 033704 | 052116 | 046111 | | |
| 5622 | 033710 | 005015 | 020040 | 047440 | .ASCII <15><12>/ ONE OR MORE CARDS ARE READ./ |

| | | | | |
|------|--------|--------|--------|--------|
| 5623 | 033716 | 042516 | 047440 | 020122 |
| 5624 | 033724 | 047515 | 042522 | 041440 |
| 5625 | 033732 | 051101 | 051504 | 040440 |
| 5626 | 033740 | 042522 | 051040 | 040505 |
| 5627 | 033746 | 027104 | | |
| 5628 | 033750 | 005015 | 020040 | 024040 |
| 5629 | 033756 | 047508 | 020122 | 044124 |
| 5630 | 033764 | 020105 | 030461 | 031457 |
| 5631 | 033772 | 020064 | 051120 | 051505 |
| 5632 | 034000 | 020123 | 046110 | 027524 |
| 5633 | 034006 | 051523 | 051 | |
| 5634 | 034011 | 015 | 005012 | 027062 |
| 5635 | 034016 | 052040 | 042510 | 020116 |
| 5636 | 034024 | 042522 | 052123 | 051117 |
| 5637 | 034032 | 020105 | 040510 | 052114 |
| 5638 | 034040 | 051440 | 027127 | 052040 |
| 5639 | 034047 | 020117 | 047516 | 046522 |
| 5640 | 034054 | 046101 | 050040 | 051517 |
| 5641 | 034062 | 052111 | 047511 | 116 |
| 5642 | 034067 | 015 | 020012 | 020040 |
| 5643 | 034074 | 047101 | 020104 | 047503 |
| 5644 | 034102 | 052116 | 047111 | 042525 |
| 5645 | 034110 | 042040 | 040511 | 047107 |
| 5646 | 034116 | 051517 | 044524 | 020103 |
| 5647 | 034124 | 051106 | 046517 | 041440 |
| 5648 | 034132 | 052520 | 041440 | 047117 |
| 5649 | 034140 | 047523 | 042514 | |
| 5650 | 034144 | 005015 | 020040 | 024040 |
| 5651 | 034152 | 030461 | 031457 | 020064 |
| 5652 | 034160 | 051120 | 051505 | 020123 |
| 5653 | 034166 | 047103 | 051124 | 020114 |
| 5654 | 034174 | 020046 | 047503 | 052116 |
| 5655 | 034202 | 051440 | 046511 | 046125 |
| 5656 | 034210 | 040524 | 042516 | 052517 |
| 5657 | 034216 | 046123 | 024531 | 000 |
| 5658 | | | | |
| 5659 | 034223 | 015 | 046012 | 040517 |
| 5660 | 034230 | 020104 | 044124 | 020105 |
| 5661 | 034236 | 042101 | 051104 | 051505 |
| 5662 | 034244 | 020123 | 043117 | 052040 |
| 5663 | 034252 | 042510 | 042040 | 051505 |
| 5664 | 034260 | 051111 | 042105 | 052040 |
| 5665 | 034266 | 051505 | 020124 | 047111 |
| 5666 | 034274 | 047524 | 052040 | 042510 |
| 5667 | 034302 | 005015 | 053523 | 052111 |
| 5668 | 034310 | 044103 | 051040 | 043505 |
| 5669 | 034316 | 051511 | 042524 | 020122 |
| 5670 | 034324 | 020055 | 027111 | 027105 |
| 5671 | 034332 | 052040 | 042510 | 040440 |
| 5672 | 034340 | 042104 | 042522 | 051523 |
| 5673 | 034346 | 047440 | 020106 | 044124 |
| 5674 | 034354 | 020105 | 041523 | 050117 |
| 5675 | 034362 | 105 | | |
| 5676 | 034363 | 015 | 044412 | 051516 |

.ASCII <15><12>" (FOR THE 11/34 PRESS HLT/SS)"

.ASCII <15><12><12>/2. THEN RESTORE HALT SW. TO NORMAL POSITION/

.ASCII <15><12>/ AND CONTINUE DIAGNOSTIC FROM CPU CONSOLE/

.ASCIZ <15><12>" (11/34 PRESS CNTRL & CONT SIMULTANEOUSLY)"

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

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

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

| | | | | |
|------|--------|--------|--------|--------|
| 5677 | 034370 | 051124 | 041525 | 044524 |
| 5678 | 034376 | 047117 | 040440 | 020124 |
| 5679 | 034404 | 044124 | 020105 | 042502 |
| 5680 | 034412 | 044507 | 047116 | 047111 |
| 5681 | 034420 | 020107 | 043117 | 052040 |
| 5682 | 034426 | 042510 | 052040 | 051505 |
| 5683 | 034434 | 124 | | |
| 5684 | 034435 | 015 | 052012 | 042510 |
| 5685 | 034442 | 020116 | 051120 | 051505 |
| 5686 | 034450 | 020123 | 047503 | 052116 |
| 5687 | 034456 | 047111 | 042525 | 051440 |
| 5688 | 034464 | 044527 | 041524 | 000110 |
| 5689 | | | | |
| 5690 | 034472 | 005015 | 042523 | 020124 |
| 5691 | 034500 | 042504 | 044523 | 042522 |
| 5692 | 034506 | 020104 | 053523 | 052111 |
| 5693 | 034514 | 044103 | 051040 | 043505 |
| 5694 | 034522 | 051511 | 042524 | 020122 |
| 5695 | 034530 | 050117 | 044524 | 047117 |
| 5696 | 034536 | 123 | | |
| 5697 | 034537 | 015 | 052012 | 042510 |
| 5698 | 034544 | 020116 | 051120 | 051505 |
| 5699 | 034552 | 020123 | 047503 | 052116 |
| 5700 | 034560 | 047111 | 042525 | 051440 |
| 5701 | 034566 | 044527 | 041524 | 000110 |
| 5702 | | | | |
| 5703 | 034574 | 005015 | 047514 | 042101 |
| 5704 | 034602 | 042040 | 051505 | 051111 |
| 5705 | 034610 | 042105 | 050040 | 052101 |
| 5706 | 034616 | 042524 | 047122 | 053040 |
| 5707 | 034624 | 046101 | 042525 | 044440 |
| 5708 | 034632 | 052116 | 020117 | 044124 |
| 5709 | 034640 | 020105 | 053523 | 052111 |
| 5710 | 034646 | 044103 | | |
| 5711 | 034650 | 005015 | 042522 | 044507 |
| 5712 | 034656 | 052123 | 051105 | 026440 |
| 5713 | 034664 | 044440 | 042456 | 020056 |
| 5714 | 034672 | 047111 | 047524 | 051440 |
| 5715 | 034700 | 051527 | 030474 | 035061 |
| 5716 | 034706 | 037060 | | |
| 5717 | 034710 | 005015 | 044124 | 047105 |
| 5718 | 034716 | 050040 | 042522 | 051523 |
| 5719 | 034724 | 041440 | 047117 | 044524 |
| 5720 | 034732 | 052516 | 020105 | 053523 |
| 5721 | 034740 | 052111 | 044103 | 000 |
| 5722 | | | | |
| 5723 | 034745 | 015 | 052012 | 044510 |
| 5724 | 034752 | 020123 | 042524 | 052123 |
| 5725 | 034760 | 053040 | 051105 | 043111 |
| 5726 | 034766 | 042511 | 020123 | 044124 |
| 5727 | 034774 | 020105 | 041101 | 046111 |
| 5728 | 035002 | 052111 | 020131 | 043117 |
| 5729 | 035010 | 052040 | 042510 | |
| 5730 | 035014 | 005015 | 051522 | 031061 |

.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>"

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

MSG26: .ASCII <15><12>/THIS TEST VERIFIES THE ABILITY OF THE/

.ASCIZ <15><12>/RS1200 TO DETECT MIS-REGISTERED CARDS/

| | | | | |
|------|--------|--------|--------|--------|
| 5731 | 035022 | 030060 | 052040 | 020117 |
| 5732 | 035030 | 042504 | 042524 | 052103 |
| 5733 | 035036 | 046440 | 051511 | 051055 |
| 5734 | 035044 | 043505 | 051511 | 042524 |
| 5735 | 035052 | 042522 | 020104 | 040503 |
| 5736 | 035060 | 042122 | 000123 | |
| 5737 | | | | |
| 5738 | 035064 | 005015 | 040515 | 042513 |
| 5739 | 035072 | 040440 | 042116 | 052040 |
| 5740 | 035100 | 042510 | 020116 | 046120 |
| 5741 | 035106 | 041501 | 020105 | 020101 |
| 5742 | 035114 | 050123 | 041505 | 040511 |
| 5743 | 035122 | 020114 | 040503 | 042122 |
| 5744 | 035130 | 044440 | 052116 | 020117 |
| 5745 | 035136 | 044124 | 020105 | 047111 |
| 5746 | 035144 | 052520 | 124 | |
| 5747 | 035147 | 015 | 044012 | 050117 |
| 5748 | 035154 | 042520 | 027122 | 020040 |
| 5749 | 035162 | 042522 | 042506 | 020122 |
| 5750 | 035170 | 047524 | 051440 | 041505 |
| 5751 | 035176 | 044524 | 047117 | 033040 |
| 5752 | 035204 | 031456 | 047440 | 020106 |
| 5753 | 035212 | 044514 | 052123 | 047111 |
| 5754 | 035220 | 020107 | 047506 | 020122 |
| 5755 | 035226 | 042504 | 040524 | 046111 |
| 5756 | 035234 | 000123 | | |
| 5757 | | | | |
| 5758 | 035236 | 005015 | 051042 | 040505 |
| 5759 | 035244 | 020104 | 044103 | 041505 |
| 5760 | 035252 | 020113 | 044514 | 044107 |
| 5761 | 035260 | 021124 | 051440 | 047510 |
| 5762 | 035266 | 046125 | 020104 | 042502 |
| 5763 | 035274 | 041040 | 044514 | 045516 |
| 5764 | 035302 | 047111 | 000107 | |
| 5765 | | | | |
| 5766 | 035306 | 005015 | 051120 | 051505 |
| 5767 | 035314 | 020123 | 051042 | 051505 |
| 5768 | 035322 | 052105 | 020042 | 047524 |
| 5769 | 035330 | 051040 | 050105 | 040505 |
| 5770 | 035336 | 020124 | 051105 | 047522 |
| 5771 | 035344 | 122 | | |
| 5772 | 035345 | 015 | 043012 | 047125 |
| 5773 | 035352 | 052103 | 047511 | 020116 |
| 5774 | 035360 | 042524 | 052123 | 040440 |
| 5775 | 035366 | 040507 | 047111 | 000 |
| 5776 | 035373 | 015 | 044412 | 020123 |
| 5777 | 035400 | 044124 | 020105 | 040503 |
| 5778 | 035406 | 042122 | 051040 | 040505 |
| 5779 | 035414 | 042504 | 020122 | 047125 |
| 5780 | 035422 | 042504 | 122 | |
| 5781 | 035425 | 015 | 052012 | 051505 |
| 5782 | 035432 | 020124 | 047515 | 042504 |
| 5783 | 035440 | 020114 | 051522 | 030455 |
| 5784 | 035446 | 030062 | 037460 | 054450 |

MSG27: .ASCII <15><12>/MAKE AND THEN PLACE A SPECIAL CARD INTO THE INPUT.

.ASCIZ <15><12>/HOPPER. REFER TO SECTION 6.3 OF LISTING FOR DETAILS/

MSG28: .ASCIZ <15><12>/"READ CHECK LIGHT" SHOULD BE BLINKING/

MSG29: .ASCII <15><12>/PRESS "RESET" TO REPEAT ERROR/

.ASCIZ <15><12>/FUNCTION TEST AGAIN/

MSG30: .ASCII <15><12>/IS THE CARD READER UNDER/

.ASCIZ <15><12>/TEST MODEL RS-100?(Y OR N)/

| | | | | | |
|------|--------|--------|--------|--------|-------------------------------------------------------------|
| 5785 | 035454 | 047440 | 020122 | 024516 | |
| 5786 | 035462 | 000 | | | |
| 5787 | 035463 | 015 | 040412 | 020124 | MSG31: .ASCII <15><12>/AT THIS TIME SET THE APPROPRIATE/ |
| 5788 | 035470 | 044124 | 051511 | 052040 | |
| 5789 | 035476 | 046511 | 020105 | 042523 | |
| 5790 | 035504 | 020124 | 044124 | 020105 | |
| 5791 | 035512 | 050101 | 051120 | 050117 | |
| 5792 | 035520 | 044522 | 052101 | 105 | |
| 5793 | 035525 | 015 | 051412 | 044527 | .ASCII <15><12>/SWITCHES ON THE SR. (REFER TO THE/ |
| 5794 | 035532 | 041524 | 042510 | 020123 | |
| 5795 | 035540 | 047117 | 052040 | 042510 | |
| 5796 | 035546 | 051440 | 027122 | 024040 | |
| 5797 | 035554 | 042522 | 042506 | 020122 | |
| 5798 | 035562 | 047524 | 052040 | 042510 | |
| 5799 | 035570 | 005015 | 047504 | 052503 | .ASCIZ <15><12>/DOCUMENTATION ON SWITCH REGISTER SETTINGS)/ |
| 5800 | 035576 | 042515 | 052116 | 052101 | |
| 5801 | 035604 | 047511 | 020116 | 047117 | |
| 5802 | 035612 | 051440 | 044527 | 041524 | |
| 5803 | 035620 | 020110 | 042522 | 044507 | |
| 5804 | 035626 | 052123 | 051105 | 051440 | |
| 5805 | 035634 | 052105 | 044524 | 043516 | |
| 5806 | 035642 | 024523 | 000 | | |
| 5807 | 035645 | 015 | 052012 | 052123 | MSG32: .ASCIZ <15><12>/TST33 DATA LATE TEST/ |
| 5808 | 035652 | 031463 | 020040 | 040504 | |
| 5809 | 035660 | 040524 | 046040 | 052101 | |
| 5810 | 035666 | 020105 | 042524 | 052123 | |
| 5811 | 035674 | 000 | | | |
| 5812 | 035675 | 015 | 052012 | 052123 | MSG33: .ASCIZ <15><12>/TST34 OFF-LINE TEST/ |
| 5813 | 035702 | 032063 | 020040 | 043117 | |
| 5814 | 035710 | 026506 | 044514 | 042516 | |
| 5815 | 035716 | 052040 | 051505 | 000124 | |
| 5816 | 035724 | 005015 | 051524 | 031524 | MSG34: .ASCIZ <15><12>/TST35 SPECIAL INT. COND. TEST/ |
| 5817 | 035732 | 020065 | 051440 | 042520 | |
| 5818 | 035740 | 044503 | 046101 | 044440 | |
| 5819 | 035746 | 052116 | 020056 | 047503 | |
| 5820 | 035754 | 042116 | 020056 | 042524 | |
| 5821 | 035762 | 052123 | 000 | | |
| 5822 | 035765 | 015 | 052012 | 052123 | MSG35: .ASCIZ <15><12>/TST36 HOPPER EMPTY TEST/ |
| 5823 | 035772 | 033063 | 020040 | 047510 | |
| 5824 | 036000 | 050120 | 051105 | 042440 | |
| 5825 | 036006 | 050115 | 054524 | 052040 | |
| 5826 | 036014 | 051505 | 000124 | | |
| 5827 | 036020 | 005015 | 051524 | 031524 | MSG36: .ASCIZ <15><12>/TST37 STACKER FULL TEST/ |
| 5828 | 036026 | 020067 | 051440 | 040524 | |
| 5829 | 036034 | 045503 | 051105 | 043040 | |
| 5830 | 036042 | 046125 | 020114 | 042524 | |
| 5831 | 036050 | 052123 | 000 | | |
| 5832 | 036053 | 015 | 052012 | 052123 | MSG37: .ASCIZ <15><12>/TST40 PICK CHECK TEST/ |
| 5833 | 036060 | 030064 | 020040 | 044520 | |
| 5834 | 036066 | 045503 | 041440 | 042510 | |
| 5835 | 036074 | 045503 | 052040 | 051505 | |
| 5836 | 036102 | 000124 | | | |
| 5837 | 036104 | 005015 | 051524 | 032124 | MSG38: .ASCIZ <15><12>/TST41 STACKER ERROR TEST/ |
| 5838 | 036112 | 020061 | 051440 | 040524 | |

| | | | | | |
|------|--------|--------|--------|--------|----------------------------------------------------------|
| 5839 | 036120 | 045503 | 051105 | 042440 | |
| 5840 | 036126 | 051122 | 051117 | 052040 | |
| 5841 | 036134 | 051505 | 000124 | | |
| 5842 | 036140 | 005015 | 051524 | 032124 | MSG39: .ASCIZ <15><12>/TST42 EOF & HOPPER CHECK/ |
| 5843 | 036146 | 020062 | 042440 | 043117 | |
| 5844 | 036154 | 023040 | 044040 | 050117 | |
| 5845 | 036162 | 042520 | 020122 | 044103 | |
| 5846 | 036170 | 041505 | 000113 | | |
| 5847 | 036174 | 005015 | 051524 | 032124 | MSG40: .ASCIZ <15><12>/TST43 DARK-LIGHT CHECK/ |
| 5848 | 036202 | 020063 | 042040 | 051101 | |
| 5849 | 036210 | 026513 | 044514 | 044107 | |
| 5850 | 036216 | 020124 | 044103 | 041505 | |
| 5851 | 036224 | 000113 | | | |
| 5852 | 036226 | 005015 | 051524 | 032124 | MSG41: .ASCIZ <15><12>/TST43A MIS-REGISTERED CARD TEST/ |
| 5853 | 036234 | 040463 | 020040 | 044515 | |
| 5854 | 036242 | 026523 | 042522 | 044507 | |
| 5855 | 036250 | 052123 | 051105 | 042105 | |
| 5856 | 036256 | 041440 | 051101 | 020104 | |
| 5857 | 036264 | 042524 | 052123 | 000 | |
| 5858 | | | | | |
| 5859 | 036272 | | | | .EVEN |
| 5860 | | | | | |
| 5861 | | | | | ;ERROR ITEMS MESSAGE TABLE |
| 5862 | | | | | |
| 5863 | 036272 | 052123 | 052101 | 051525 | EM1: .ASCII "STATUS REGISTER (CDS) BIT07 NOT SET BY" |
| 5864 | 036300 | 051040 | 043505 | 051511 | |
| 5865 | 036306 | 042524 | 020122 | 041450 | |
| 5866 | 036314 | 051504 | 020051 | 044502 | |
| 5867 | 036322 | 030124 | 020067 | 047516 | |
| 5868 | 036330 | 020124 | 042523 | 020124 | |
| 5869 | 036336 | 054502 | | | |
| 5870 | 036340 | 005015 | 047111 | 052111 | .ASCIZ <15><12>"INITIALIZATION PULSE" |
| 5871 | 036346 | 040511 | 044514 | 040532 | |
| 5872 | 036354 | 044524 | 047117 | 050040 | |
| 5873 | 036362 | 046125 | 042523 | 000 | |
| 5874 | | | | | |
| 5875 | 036367 | 103 | 046117 | 046525 | EM2: .ASCII "COLUMN COUNT REGISTER (CDC) NOT CLEARED BY" |
| 5876 | 036374 | 020116 | 047503 | 047125 | |
| 5877 | 036402 | 020124 | 042522 | 044507 | |
| 5878 | 036410 | 052123 | 051105 | 024040 | |
| 5879 | 036416 | 042103 | 024503 | 047040 | |
| 5880 | 036424 | 052117 | 041440 | 042514 | |
| 5881 | 036432 | 051101 | 042105 | 041040 | |
| 5882 | 036440 | 131 | | | |
| 5883 | 036441 | 015 | 044412 | 044516 | .ASCIZ <15><12>"INITIALIZATION PULSE" |
| 5884 | 036446 | 044524 | 046101 | 055111 | |
| 5885 | 036454 | 052101 | 047511 | 020116 | |
| 5886 | 036462 | 052520 | 051514 | 000105 | |
| 5887 | | | | | |
| 5888 | 036470 | 052502 | 020123 | 042101 | EM3: .ASCII "BUS ADDRESS REGISTER (CDA) NOT CLEARED BY" |
| 5889 | 036476 | 051104 | 051505 | 020123 | |
| 5890 | 036504 | 042522 | 044507 | 052123 | |
| 5891 | 036512 | 051105 | 024040 | 042103 | |
| 5892 | 036520 | 024501 | 047040 | 052117 | |

| | | | | | |
|------|--------|--------|--------|--------|---------------------------------------------------------|
| 5893 | 036526 | 041440 | 042514 | 051101 | |
| 5894 | 036534 | 042105 | 041040 | 131 | |
| 5895 | 036541 | 015 | 044412 | 044516 | .ASCIZ <15><12>"INITIALIZATION PULSE" |
| 5896 | 036546 | 044524 | 046101 | 055111 | |
| 5897 | 036554 | 052101 | 047511 | 020116 | |
| 5898 | 036562 | 052520 | 051514 | 000105 | |
| 5899 | | | | | |
| 5900 | 036570 | 052123 | 052101 | 051525 | EM4: .ASCIZ "STATUS REGISTER CONTENTS INCORRECT" |
| 5901 | 036576 | 051040 | 043505 | 051511 | |
| 5902 | 036604 | 042524 | 020122 | 047503 | |
| 5903 | 036612 | 052116 | 047105 | 051524 | |
| 5904 | 036620 | 044440 | 041516 | 051117 | |
| 5905 | 036625 | 042522 | 052103 | 000 | |
| 5906 | | | | | |
| 5907 | 036633 | 103 | 046117 | 046525 | EM5: .ASCII "COLUMN COUNT REGISTER (CDC) NOT ABLE TO" |
| 5908 | 036640 | 020116 | 047503 | 047125 | |
| 5909 | 036646 | 020124 | 042522 | 044507 | |
| 5910 | 036654 | 052123 | 051105 | 024040 | |
| 5911 | 036662 | 042103 | 024503 | 047040 | |
| 5912 | 036670 | 052117 | 040440 | 046102 | |
| 5913 | 036676 | 020105 | 047524 | | |
| 5914 | 036702 | 005015 | 042502 | 046040 | .ASCIZ <15><12>"BE LOADED WITH ALL ONES" |
| 5915 | 036710 | 040517 | 042504 | 020104 | |
| 5916 | 036716 | 044527 | 044124 | 040440 | |
| 5917 | 036724 | 046114 | 047440 | 042516 | |
| 5918 | 036732 | 000123 | | | |
| 5919 | | | | | |
| 5920 | 036734 | 047503 | 052514 | 047115 | EM6: .ASCII "COLUMN COUNT REGISTER (CDC) NOT CLEARED" |
| 5921 | 036742 | 041440 | 052517 | 052116 | |
| 5922 | 036750 | 051040 | 043505 | 051511 | |
| 5923 | 036756 | 042524 | 020122 | 041450 | |
| 5924 | 036764 | 041504 | 020051 | 047516 | |
| 5925 | 036772 | 020124 | 046103 | 040505 | |
| 5926 | 037000 | 042522 | 104 | | |
| 5927 | 037003 | 015 | 041012 | 020131 | .ASCIZ <15><12>"BY POWER CLEAR" |
| 5928 | 037010 | 047520 | 042527 | 020122 | |
| 5929 | 037016 | 046103 | 040505 | 000122 | |
| 5930 | | | | | |
| 5931 | 037024 | 052502 | 020123 | 042101 | EM7: .ASCII "BUS ADDRESS REGISTER (CDA) NOT ABLE TO BE" |
| 5932 | 037032 | 051104 | 051505 | 020123 | |
| 5933 | 037040 | 042522 | 044507 | 052123 | |
| 5934 | 037046 | 051105 | 024040 | 042103 | |
| 5935 | 037054 | 024501 | 047040 | 052117 | |
| 5936 | 037062 | 040440 | 046102 | 020135 | |
| 5937 | 037070 | 047524 | 041040 | 105 | |
| 5938 | 037075 | 015 | 046012 | 040517 | .ASCIZ <15><12>"LOADED WITH ALL ONES" |
| 5939 | 037102 | 042504 | 020104 | 044527 | |
| 5940 | 037110 | 044124 | 040440 | 046114 | |
| 5941 | 037116 | 047440 | 042516 | 000123 | |
| 5942 | | | | | |
| 5943 | 037124 | 052502 | 020123 | 042101 | EM10: .ASCII "BUS ADDRESS REGISTER (CDA) NOT CLEARED" |
| 5944 | 037132 | 051104 | 051505 | 020123 | |
| 5945 | 037140 | 042522 | 044507 | 052123 | |
| 5946 | 037146 | 051105 | 024040 | 042103 | |

MAINDEC - 11 - DZCDB-B
DZCDB.P11

MACY11 27(654) 1-JUL-77 08:39 PAGE 115
DATA TABLES

| | | | | | | |
|------|--------|--------|--------|--------|-------|--------------------------------------------------------|
| 5947 | 037154 | 024501 | 047040 | 052117 | | |
| 5948 | 037162 | 041440 | 042514 | 051101 | | |
| 5949 | 037170 | 042105 | | | | |
| 5950 | 037172 | 005015 | 054502 | 050040 | | .ASCIZ <15><12>"BY POWER CLEAR" |
| 5951 | 037200 | 053517 | 051105 | 041440 | | |
| 5952 | 037206 | 042514 | 051101 | 000 | | |
| 5953 | | | | | | |
| 5954 | 037213 | 103 | 047117 | 051124 | EM11: | .ASCII "CONTROLLER READY DID NOT CLEAR ON" |
| 5955 | 037220 | 046117 | 042514 | 020122 | | |
| 5956 | 037226 | 042522 | 042101 | 020131 | | |
| 5957 | 037234 | 044504 | 020104 | 047516 | | |
| 5958 | 037242 | 020124 | 046103 | 040505 | | |
| 5959 | 037250 | 020122 | 047117 | | | |
| 5960 | 037257 | 005015 | 042522 | 042101 | | .ASCIZ <15><12>"READING A CARD" |
| 5961 | 037262 | 047111 | 020107 | 020101 | | |
| 5962 | 037270 | 040503 | 042122 | 000 | | |
| 5963 | | | | | | |
| 5964 | 037275 | 103 | 047117 | 051124 | EM12: | .ASCII "CONTROLLER READY DID NOT CLEAR BIT00" |
| 5965 | 037302 | 046117 | 042514 | 020122 | | |
| 5966 | 037310 | 042522 | 042101 | 020131 | | |
| 5967 | 037316 | 044504 | 020104 | 047516 | | |
| 5968 | 037324 | 020124 | 046103 | 040505 | | |
| 5969 | 037332 | 020122 | 044502 | 030124 | | |
| 5970 | 037340 | 060 | | | | |
| 5971 | 037341 | 015 | 047412 | 020106 | | .ASCIZ <15><12>"OF STATUS REGISTER (CDS)" |
| 5972 | 037346 | 052123 | 052101 | 051525 | | |
| 5973 | 037354 | 051040 | 043505 | 051511 | | |
| 5974 | 037362 | 042524 | 020122 | 041450 | | |
| 5975 | 037370 | 051504 | 000051 | | | |
| 5976 | | | | | | |
| 5977 | 037374 | 047503 | 052116 | 047522 | EM13: | .ASCIZ "CONTROLLER DID NOT SET WITHIN 1 SECOND" |
| 5978 | 037402 | 046114 | 051105 | 042040 | | |
| 5979 | 037410 | 042111 | 047040 | 052117 | | |
| 5980 | 037416 | 051440 | 052105 | 053440 | | |
| 5981 | 037424 | 052111 | 044510 | 020116 | | |
| 5982 | 037432 | 020061 | 042523 | 047503 | | |
| 5983 | 037440 | 042116 | 000 | | | |
| 5984 | | | | | | |
| 5985 | 037443 | 105 | 051122 | 051117 | EM14: | .ASCIZ "ERROR (BIT15) SET IN STATUS REGISTER (CDS)" |
| 5986 | 037450 | 024040 | 044502 | 030524 | | |
| 5987 | 037456 | 024465 | 051440 | 052105 | | |
| 5988 | 037464 | 044440 | 020116 | 052123 | | |
| 5989 | 037472 | 052101 | 051525 | 051040 | | |
| 5990 | 037500 | 043505 | 051511 | 042524 | | |
| 5991 | 037506 | 020122 | 041450 | 051504 | | |
| 5992 | 037514 | 000051 | | | | |
| 5993 | | | | | | |
| 5994 | 037516 | 044502 | 020124 | 051117 | EM15: | .ASCII "BIT OR BITS SET IN STATUS REGISTER (CDS) THAT" |
| 5995 | 037524 | 041040 | 052111 | 020123 | | |
| 5996 | 037532 | 042523 | 020124 | 047111 | | |
| 5997 | 037540 | 051440 | 040524 | 052524 | | |
| 5998 | 037546 | 020123 | 042522 | 044507 | | |
| 5999 | 037554 | 052123 | 051105 | 024040 | | |
| 6000 | 037562 | 042103 | 024523 | 052040 | | |

| | | | | | | |
|------|--------|--------|--------|--------|--------|-------------------------------------------------------|
| 6001 | 037570 | 040510 | 124 | | | |
| 6002 | 037573 | 015 | 051412 | 047510 | .ASCIZ | <15><12>"SHOULD NOT BE" |
| 6003 | 037600 | 046125 | 020104 | 047516 | | |
| 6004 | 037606 | 020124 | 042502 | 000 | | |
| 6005 | | | | | | |
| 6006 | 037613 | 102 | 051525 | 020131 | EM16: | .ASCII "BUSY SET IN STATUS REGISTER (CDS)" |
| 6007 | 037620 | 042523 | 020124 | 047111 | | |
| 6008 | 037626 | 051440 | 040524 | 052524 | | |
| 6009 | 037634 | 020123 | 042522 | 044507 | | |
| 6010 | 037642 | 052123 | 051105 | 024040 | | |
| 6011 | 037650 | 042103 | 024523 | | | |
| 6012 | 037654 | 005015 | 052111 | 051440 | .ASCIZ | <15><12>"IT SHOULD NOT BE" |
| 6013 | 037662 | 047510 | 046125 | 020104 | | |
| 6014 | 037670 | 047516 | 020124 | 042502 | | |
| 6015 | 037676 | 000 | | | | |
| 6016 | | | | | | |
| 6017 | 037677 | 122 | 051505 | 047524 | EM17: | .ASCII "RESTORING CPU STATUS AFTER READING A CARD" |
| 6018 | 037704 | 044522 | 043516 | 041440 | | |
| 6019 | 037712 | 052520 | 051440 | 040524 | | |
| 6020 | 037720 | 052524 | 020123 | 043101 | | |
| 6021 | 037726 | 042524 | 020122 | 042522 | | |
| 6022 | 037734 | 042101 | 047111 | 020107 | | |
| 6023 | 037742 | 020101 | 040503 | 042122 | | |
| 6024 | 037750 | 005015 | 046103 | 040505 | .ASCIZ | <15><12>"CLEARED CONTROLLER READY IN STATUS REGISTER" |
| 6025 | 037756 | 042522 | 020104 | 047503 | | |
| 6026 | 037764 | 052116 | 047522 | 046114 | | |
| 6027 | 037772 | 051105 | 051040 | 040505 | | |
| 6028 | 040000 | 054504 | 044440 | 020116 | | |
| 6029 | 040006 | 052123 | 052101 | 051525 | | |
| 6030 | 040014 | 051040 | 043505 | 051511 | | |
| 6031 | 040022 | 042524 | 000122 | | | |
| 6032 | | | | | | |
| 6033 | 040026 | 047516 | 044440 | 052116 | EM20: | .ASCIZ "NO INTERRUPT OCCURRED" |
| 6034 | 040034 | 051105 | 052522 | 052120 | | |
| 6035 | 040042 | 047440 | 041503 | 051125 | | |
| 6036 | 040050 | 042522 | 000104 | | | |
| 6037 | | | | | | |
| 6038 | 040054 | 047101 | 044440 | 052116 | EM21: | .ASCIZ "AN INTERRUPT OCCURRED - SHOULD NOT HAVE" |
| 6039 | 040062 | 051105 | 052522 | 052120 | | |
| 6040 | 040070 | 047440 | 041503 | 051125 | | |
| 6041 | 040076 | 042522 | 020104 | 020055 | | |
| 6042 | 040104 | 044123 | 052517 | 042114 | | |
| 6043 | 040112 | 047040 | 052117 | 044040 | | |
| 6044 | 040120 | 053101 | 000105 | | | |
| 6045 | | | | | | |
| 6046 | 040124 | 047111 | 042524 | 051122 | EM22: | .ASCIZ "INTERRUPT ALREADY OCCURRED AT A HIGHER LEVEL" |
| 6047 | 040132 | 050125 | 020124 | 046101 | | |
| 6048 | 040140 | 042522 | 042101 | 020131 | | |
| 6049 | 040146 | 041517 | 052503 | 051122 | | |
| 6050 | 040154 | 042105 | 040440 | 020124 | | |
| 6051 | 040162 | 020101 | 044510 | 044107 | | |
| 6052 | 040170 | 051105 | 046040 | 053105 | | |
| 6053 | 040176 | 046105 | 000 | | | |
| 6054 | | | | | | |

| | | | | | | | |
|------|--------|--------|--------|--------|-------|--------|--------------------------------------------------|
| 6055 | 040201 | 103 | 047117 | 051124 | EM23: | .ASCIZ | "CONTROLLER READY NOT SET" |
| 6056 | 040206 | 046117 | 042514 | 020122 | | | |
| 6057 | 040214 | 042522 | 042101 | 020131 | | | |
| 6058 | 040222 | 047516 | 020124 | 042523 | | | |
| 6059 | 040230 | 000124 | | | | | |
| 6060 | | | | | | | |
| 6061 | 040232 | 047111 | 042524 | 051122 | EM24: | .ASCIZ | "INTERRUPT ALREADY OCCURRED AT A LOWER LEVEL" |
| 6062 | 040240 | 050125 | 020124 | 046101 | | | |
| 6063 | 040246 | 042522 | 042101 | 020131 | | | |
| 6064 | 040254 | 041517 | 052503 | 051122 | | | |
| 6065 | 040262 | 042105 | 040440 | 020124 | | | |
| 6066 | 040270 | 020101 | 047514 | 042527 | | | |
| 6067 | 040276 | 020122 | 042514 | 042526 | | | |
| 6068 | 040304 | 000114 | | | | | |
| 6069 | | | | | | | |
| 6070 | 040306 | 047101 | 044440 | 052116 | EM25: | .ASCIZ | "AN INTERRUPT OCCURRED AT TWO DIFFERENT LEVELS" |
| 6071 | 040314 | 051105 | 052522 | 052120 | | | |
| 6072 | 040322 | 047440 | 041503 | 051125 | | | |
| 6073 | 040330 | 042522 | 020104 | 052101 | | | |
| 6074 | 040336 | 052040 | 047527 | 042040 | | | |
| 6075 | 040344 | 043111 | 042506 | 042522 | | | |
| 6076 | 040352 | 052116 | 046040 | 053105 | | | |
| 6077 | 040360 | 046105 | 000123 | | | | |
| 6078 | | | | | | | |
| 6079 | 040364 | 051105 | 047522 | 020122 | EM26: | .ASCII | "ERROR (BIT15) NOT SET IN STATUS REGISTER (CDS)" |
| 6080 | 040372 | 041050 | 052111 | 032461 | | | |
| 6081 | 040400 | 020051 | 047516 | 020124 | | | |
| 6082 | 040406 | 042523 | 020124 | 047111 | | | |
| 6083 | 040414 | 051440 | 040524 | 052524 | | | |
| 6084 | 040422 | 020123 | 042522 | 044507 | | | |
| 6085 | 040430 | 052123 | 051105 | 024040 | | | |
| 6086 | 040436 | 042103 | 024523 | | | | |
| 6087 | 040442 | 005015 | 052111 | 051440 | | .ASCIZ | <15><12>"IT SHOULD BE" |
| 6088 | 040450 | 047510 | 046125 | 020104 | | | |
| 6089 | 040456 | 042502 | 000 | | | | |
| 6090 | | | | | | | |
| 6091 | 040461 | 116 | 047117 | 042455 | EM27: | .ASCII | "NON-EXISTANT MEMORY (BIT09) NOT SET IN" |
| 6092 | 040466 | 044530 | 052123 | 047101 | | | |
| 6093 | 040474 | 020124 | 042515 | 047515 | | | |
| 6094 | 040502 | 054522 | 024040 | 044502 | | | |
| 6095 | 040510 | 030124 | 024471 | 047040 | | | |
| 6096 | 040516 | 052117 | 051440 | 052105 | | | |
| 6097 | 040524 | 044440 | 116 | | | | |
| 6098 | 040527 | 015 | 051412 | 040524 | | .ASCIZ | <15><12>"STATUS REGISTER (CDS) - IT SHOULD BE" |
| 6099 | 040534 | 052524 | 020123 | 042522 | | | |
| 6100 | 040542 | 044507 | 052123 | 051105 | | | |
| 6101 | 040550 | 024040 | 042103 | 024523 | | | |
| 6102 | 040556 | 026440 | 044440 | 020124 | | | |
| 6103 | 040564 | 044123 | 052517 | 042114 | | | |
| 6104 | 040572 | 041040 | 000105 | | | | |
| 6105 | | | | | | | |
| 6106 | 040576 | 054105 | 042524 | 042116 | EM30: | .ASCII | "EXTENDED MEMORY (BIT05) NOT SET IS STATUS" |
| 6107 | 040604 | 042105 | 046440 | 046505 | | | |
| 6108 | 040612 | 051117 | 020131 | 041050 | | | |

| | | | | | |
|------|--------|--------|--------|--------|-----------------------------------------------------------|
| 6109 | 040620 | 052111 | 032460 | 020051 | |
| 6110 | 040626 | 047516 | 020124 | 042523 | |
| 6111 | 040634 | 020124 | 051511 | 051440 | |
| 6112 | 040642 | 040524 | 052524 | 123 | |
| 6113 | 040647 | 015 | 051012 | 043505 | .ASCIZ <15><12>"REGISTER (CDS)" |
| 6114 | 040654 | 051511 | 042524 | 020122 | |
| 6115 | 040662 | 041450 | 051504 | 000051 | |
| 6116 | | | | | |
| 6117 | 040670 | 054105 | 042524 | 042116 | EM31: .ASCII "EXTENDED MEMORY (BIT04) NOT SET IN STATUS" |
| 6118 | 040676 | 042105 | 046440 | 046505 | |
| 6119 | 040704 | 051117 | 020131 | 041050 | |
| 6120 | 040712 | 052111 | 032060 | 020051 | |
| 6121 | 040720 | 047516 | 020124 | 042523 | |
| 6122 | 040726 | 020124 | 047111 | 051440 | |
| 6123 | 040734 | 040524 | 052524 | 123 | |
| 6124 | 040741 | 015 | 051012 | 043505 | .ASCIZ <15><12>"REGISTER (CDS)" |
| 6125 | 040746 | 051511 | 042524 | 020122 | |
| 6126 | 040754 | 041450 | 051504 | 000051 | |
| 6127 | | | | | |
| 6128 | 040762 | 047503 | 052116 | 047105 | EM32: .ASCII "CONTENTS OF BUS ADDRESS REGISTER (CDA)" |
| 6129 | 040770 | 051524 | 047440 | 020106 | |
| 6130 | 040776 | 052502 | 020123 | 042101 | |
| 6131 | 041004 | 051104 | 051505 | 020123 | |
| 6132 | 041012 | 042522 | 044507 | 052123 | |
| 6133 | 041020 | 051105 | 024040 | 042103 | |
| 6134 | 041026 | 024501 | | | |
| 6135 | 041030 | 005015 | 047111 | 047503 | .ASCIZ <15><12>"INCORRECT" |
| 6136 | 041036 | 051122 | 041505 | 000124 | |
| 6137 | | | | | |
| 6138 | 041044 | 047503 | 052116 | 047105 | EM33: .ASCII "CONTENTS OF COLUMN COUNT REGISTER (CDC)" |
| 6139 | 041052 | 051524 | 047440 | 020106 | |
| 6140 | 041060 | 047503 | 052514 | 047115 | |
| 6141 | 041066 | 041440 | 052517 | 052116 | |
| 6142 | 041074 | 051040 | 043505 | 051511 | |
| 6143 | 041102 | 042524 | 020122 | 041450 | |
| 6144 | 041110 | 041504 | 051 | | |
| 6145 | 041113 | 015 | 044412 | 041516 | .ASCIZ <15><12>"INCORRECT" |
| 6146 | 041120 | 051117 | 042522 | 052103 | |
| 6147 | 041126 | 000 | | | |
| 6148 | | | | | |
| 6149 | 041127 | 122 | 040505 | 042504 | EM34: .ASCII "READER OFF-LINE BUT CARD READER ERROR" |
| 6150 | 041134 | 020122 | 043117 | 026506 | |
| 6151 | 041142 | 044514 | 042516 | 041040 | |
| 6152 | 041150 | 052125 | 041440 | 051101 | |
| 6153 | 041156 | 020104 | 042522 | 042101 | |
| 6154 | 041164 | 051105 | 042440 | 051122 | |
| 6155 | 041172 | 051117 | | | |
| 6156 | 041174 | 005015 | 041050 | 052111 | .ASCIZ <15><12>"(BIT14) NOT SET IN STATUS REGISTER (CDS)" |
| 6157 | 041202 | 032061 | 020051 | 047516 | |
| 6158 | 041210 | 020124 | 042523 | 020124 | |
| 6159 | 041216 | 047111 | 051440 | 040524 | |
| 6160 | 041224 | 052524 | 020123 | 042522 | |
| 6161 | 041232 | 044507 | 052123 | 051105 | |
| 6162 | 041240 | 024040 | 042103 | 024523 | |

| | | | | |
|------|--------|--------|--------|--------|
| 6217 | 041662 | 030524 | 024462 | 051440 |
| 6218 | 041670 | 052105 | 044140 | 020116 |
| 6219 | 041676 | 052123 | 052101 | 051525 |
| 6220 | 041704 | 051040 | 043505 | 051511 |
| 6221 | 041712 | 042524 | 122 | |
| 6222 | 041715 | 015 | 020012 | 020055 |
| 6223 | 041722 | 052111 | 051440 | 047510 |
| 6224 | 041730 | 046125 | 020104 | 047516 |
| 6225 | 041736 | 020124 | 042502 | 000 |
| 6226 | | | | |
| 6227 | 041743 | 120 | 041511 | 020113 |
| 6228 | 041750 | 044103 | 041505 | 020113 |
| 6229 | 041756 | 041050 | 052111 | 031461 |
| 6230 | 041764 | 020051 | 047516 | 020124 |
| 6231 | 041772 | 042523 | 020124 | 047111 |
| 6232 | 042000 | 051440 | 040524 | 052524 |
| 6233 | 042006 | 020123 | 042522 | 044507 |
| 6234 | 042014 | 052123 | 051105 | 021440 |
| 6235 | 042022 | 020062 | 041450 | 042104 |
| 6236 | 042030 | 000051 | | |
| 6237 | | | | |
| 6238 | 042032 | 052123 | 041501 | 020113 |
| 6239 | 042040 | 044103 | 041505 | 020113 |
| 6240 | 042046 | 041050 | 052111 | 031061 |
| 6241 | 042054 | 020051 | 047516 | 020124 |
| 6242 | 042062 | 042523 | 020124 | 047111 |
| 6243 | 042070 | 051440 | 040524 | 052524 |
| 6244 | 042076 | 020123 | 042522 | 044507 |
| 6245 | 042104 | 052123 | 051105 | 021440 |
| 6246 | 042112 | 020062 | 041450 | 042104 |
| 6247 | 042120 | 000051 | | |
| 6248 | | | | |
| 6249 | 042122 | 047105 | 020104 | 043117 |
| 6250 | 042130 | 043040 | 046111 | 020105 |
| 6251 | 042136 | 041050 | 052111 | 031461 |
| 6252 | 042144 | 020051 | 042523 | 020124 |
| 6253 | 042152 | 047111 | 051440 | 040524 |
| 6254 | 042160 | 052524 | 020123 | 042522 |
| 6255 | 042166 | 044507 | 052123 | 051105 |
| 6256 | 042174 | 024040 | 042103 | 024523 |
| 6257 | 042202 | 005015 | 026440 | 044440 |
| 6258 | 042210 | 020124 | 044123 | 052517 |
| 6259 | 042216 | 042114 | 047040 | 052117 |
| 6260 | 042224 | 041040 | 000105 | |
| 6261 | | | | |
| 6262 | 042230 | 042522 | 042101 | 041440 |
| 6263 | 042236 | 042510 | 045503 | 024040 |
| 6264 | 042244 | 044502 | 030524 | 024464 |
| 6265 | 042252 | 051440 | 052105 | 044440 |
| 6266 | 042260 | 020116 | 052123 | 052101 |
| 6267 | 042266 | 051525 | 051040 | 043505 |
| 6268 | 042274 | 051511 | 042524 | 020122 |
| 6269 | 042302 | 041450 | 051504 | 051 |
| 6270 | 042307 | 015 | 020012 | 020055 |

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

EM43: .ASCIZ "PICK CHECK (BIT13) NOT SET IN STATUS REGISTER #2 (CDD)"

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

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"

| | | | | | | |
|------|--------|--------|--------|--------|-------|----------------------------------------------------------------|
| 6271 | 042314 | 052111 | 051440 | 047510 | | |
| 6272 | 042322 | 046125 | 020104 | 047516 | | |
| 6273 | 042330 | 020124 | 042502 | 000 | | |
| 6274 | | | | | | |
| 6275 | 042335 | 110 | 050117 | 042520 | EM47: | .ASCII "HOPPER CHECK (BIT02) SET IN STATUS REGISTER (CDS)" |
| 6276 | 042342 | 020122 | 044103 | 041505 | | |
| 6277 | 042350 | 020113 | 041050 | 052111 | | |
| 6278 | 042356 | 031060 | 020051 | 042523 | | |
| 6279 | 042364 | 020124 | 047111 | 051440 | | |
| 6280 | 042372 | 040524 | 052524 | 020123 | | |
| 6281 | 042400 | 042522 | 044507 | 052123 | | |
| 6282 | 042406 | 051105 | 024040 | 042103 | | |
| 6283 | 042414 | 024523 | | | | |
| 6284 | 042416 | 005015 | 026440 | 044440 | | .ASCIIZ <15><12>" - IT SHOULD NOT BE" |
| 6285 | 042424 | 020124 | 044123 | 052517 | | |
| 6286 | 042432 | 042114 | 047040 | 052117 | | |
| 6287 | 042440 | 041040 | 000105 | | | |
| 6288 | | | | | | |
| 6289 | 042444 | 047105 | 020104 | 043117 | EM50: | .ASCII "END OF FILE (BIT13) OF STATUS REGISTER (CDS) NOT SET" |
| 6290 | 042452 | 043040 | 046111 | 020105 | | |
| 6291 | 042460 | 041050 | 052111 | 031461 | | |
| 6292 | 042466 | 020051 | 043117 | 051440 | | |
| 6293 | 042474 | 040524 | 052524 | 020123 | | |
| 6294 | 042502 | 042522 | 044507 | 052123 | | |
| 6295 | 042510 | 051105 | 024040 | 042103 | | |
| 6296 | 042516 | 024523 | 047040 | 052117 | | |
| 6297 | 042524 | 051440 | 052105 | | | |
| 6298 | 042530 | 005015 | 026440 | 044440 | | .ASCIIZ <15><12>" - IT SHOULD BE" |
| 6299 | 042536 | 020124 | 044123 | 052517 | | |
| 6300 | 042544 | 042114 | 041040 | 000105 | | |
| 6301 | | | | | | |
| 6302 | 042552 | 042522 | 042101 | 041440 | EM51: | .ASCII "READ CHECK (BIT14) OF STATUS REGISTER (CDS) NOT SET" |
| 6303 | 042560 | 042510 | 045503 | 024040 | | |
| 6304 | 042566 | 044502 | 030524 | 024464 | | |
| 6305 | 042574 | 047440 | 020106 | 052123 | | |
| 6306 | 042602 | 052101 | 051525 | 051040 | | |
| 6307 | 042610 | 043505 | 051511 | 042524 | | |
| 6308 | 042616 | 020122 | 041450 | 051504 | | |
| 6309 | 042624 | 020051 | 047516 | 020124 | | |
| 6310 | 042632 | 042523 | 124 | | | |
| 6311 | 042635 | 015 | 020012 | 020055 | | .ASCIIZ <15><12>" - IT SHOULD BE" |
| 6312 | 042642 | 052111 | 051440 | 047510 | | |
| 6313 | 042650 | 046125 | 020104 | 042502 | | |
| 6314 | 042656 | 000 | | | | |
| 6315 | | | | | | |
| 6316 | 042657 | 110 | 050117 | 042520 | EM52: | .ASCII "HOPPER CHECK (BIT12) OF STATUS REGISTER (CDS) NOT SET" |
| 6317 | 042664 | 020122 | 044103 | 041505 | | |
| 6318 | 042672 | 020113 | 041050 | 052111 | | |
| 6319 | 042700 | 031060 | 020051 | 043117 | | |
| 6320 | 042706 | 051440 | 040524 | 052524 | | |
| 6321 | 042714 | 020123 | 042522 | 044507 | | |
| 6322 | 042722 | 052123 | 051105 | 024040 | | |
| 6323 | 042730 | 042103 | 024523 | 047040 | | |
| 6324 | 042736 | 052117 | 051440 | 052105 | | |

M15

| | | | | | |
|------|--------|--------|--------|--------|-----------------------------------------------------------------------|
| 6325 | 042744 | 005015 | 026440 | 044440 | .ASCIZ <15><12>" - IT SHOULD BE" |
| 6326 | 042752 | 020124 | 044123 | 052517 | |
| 6327 | 042760 | 042114 | 041040 | 000105 | |
| 6328 | | | | | |
| 6329 | 042766 | 047105 | 020104 | 043117 | EMS3: .ASCII "END OF FILE (BIT13) OF STATUS REGISTER (CDS) DID NOT" |
| 6330 | 042774 | 043040 | 046111 | 020105 | |
| 6331 | 043002 | 041050 | 052111 | 031461 | |
| 6332 | 043010 | 020051 | 043117 | 051440 | |
| 6333 | 043016 | 040524 | 052524 | 020123 | |
| 6334 | 043024 | 042522 | 044507 | 052123 | |
| 6335 | 043032 | 051105 | 024040 | 042103 | |
| 6336 | 043040 | 024523 | 042040 | 042111 | |
| 6337 | 043046 | 047040 | 052117 | | |
| 6338 | 043052 | 005015 | 046103 | 040505 | .ASCIZ <15><12>"CLEAR WITH TRANSITION TO ON-LINE" |
| 6339 | 043060 | 020122 | 044527 | 044124 | |
| 6340 | 043066 | 052040 | 040522 | 051516 | |
| 6341 | 043074 | 052111 | 047511 | 020116 | |
| 6342 | 043102 | 047524 | 047440 | 026516 | |
| 6343 | 043110 | 044514 | 042516 | 000 | |
| 6344 | | | | | |
| 6345 | 043115 | 122 | 040505 | 020104 | EMS4: .ASCIZ "READ CHECK (BIT14) NOT SET IN STATUS REGISTER #2 (CDD)" |
| 6346 | 043122 | 044103 | 041505 | 020113 | |
| 6347 | 043130 | 041050 | 052111 | 032061 | |
| 6348 | 043136 | 020051 | 047516 | 020124 | |
| 6349 | 043144 | 042523 | 020124 | 047111 | |
| 6350 | 043152 | 051440 | 040524 | 052524 | |
| 6351 | 043160 | 020123 | 042522 | 044507 | |
| 6352 | 043166 | 052123 | 051105 | 021440 | |
| 6353 | 043174 | 020062 | 041450 | 042104 | |
| 6354 | 043202 | 000051 | | | |
| 6355 | | | | | |
| 6356 | 043204 | 040503 | 042122 | 051040 | EMS5: .ASCIZ "CARD READER ERROR BUT NOT BOTH CARD" |
| 6357 | 043212 | 040505 | 042504 | 020122 | |
| 6358 | 043220 | 051105 | 047522 | 020122 | |
| 6359 | 043226 | 052502 | 020124 | 047516 | |
| 6360 | 043234 | 020124 | 030070 | 044124 | |
| 6361 | 043242 | 041440 | 051101 | 000104 | |
| 6362 | | | | | |
| 6363 | 043250 | 040503 | 042122 | 051040 | EMS6: .ASCIZ "CARD READER ERROR BUT NOT OFF-LINE" |
| 6364 | 043256 | 040505 | 042504 | 020122 | |
| 6365 | 043264 | 051105 | 047522 | 020122 | |
| 6366 | 043272 | 052502 | 020124 | 047516 | |
| 6367 | 043300 | 020124 | 043117 | 026506 | |
| 6368 | 043306 | 044514 | 042516 | 000 | |
| 6369 | | | | | |
| 6370 | 043313 | 105 | 042116 | 047440 | EMS7: .ASCIZ "END OF FILE (BIT13) ERROR OF STATUS REGISTER (CDS)" |
| 6371 | 043320 | 020106 | 044506 | 042514 | |
| 6372 | 043326 | 024040 | 044502 | 030524 | |
| 6373 | 043334 | 024463 | 042440 | 051122 | |
| 6374 | 043342 | 051117 | 047440 | 020106 | |
| 6375 | 043350 | 052123 | 052101 | 051525 | |
| 6376 | 043356 | 051040 | 043505 | 051511 | |
| 6377 | 043364 | 042524 | 020122 | 041450 | |
| 6378 | 043372 | 051504 | 000051 | | |

| | | | | | |
|------|--------|--------|--------|--------|-----------------------------------------------------------------|
| 6379 | | | | | EM60: .ASCIZ "DATA ERROR (BIT11) OF STATUS REGISTER (CDS)" |
| 6380 | 043376 | 040504 | 040524 | 042440 | |
| 6381 | 043404 | 051122 | 051117 | 024040 | |
| 6382 | 043412 | 044502 | 030524 | 024461 | |
| 6383 | 043420 | 047440 | 020106 | 052123 | |
| 6384 | 043428 | 052101 | 051522 | 041040 | |
| 6385 | 043436 | 043504 | 051511 | 042524 | |
| 6386 | 043442 | 020122 | 041450 | 051504 | |
| 6387 | 043450 | 000051 | | | |
| 6388 | | | | | |
| 6389 | 043452 | 040504 | 040524 | 046040 | EM61: .ASCIZ "DATA LATE (BIT11) ERROR OF STATUS REGISTER (CDS)" |
| 6390 | 043460 | 052101 | 020105 | 041050 | |
| 6391 | 043466 | 052111 | 030461 | 020051 | |
| 6392 | 043474 | 051105 | 047522 | 020122 | |
| 6393 | 043502 | 043117 | 051440 | 040524 | |
| 6394 | 043510 | 052524 | 020123 | 042522 | |
| 6395 | 043516 | 044507 | 052123 | 051105 | |
| 6396 | 043524 | 024040 | 042103 | 024523 | |
| 6397 | 043532 | 000 | | | |
| 6398 | | | | | |
| 6399 | 043533 | 116 | 047117 | 042455 | EM62: .ASCII "NON-EXISTANT MEMORY (BIT09) ERROR OF STATUS" |
| 6400 | 043540 | 044530 | 052123 | 047101 | |
| 6401 | 043546 | 020124 | 042515 | 047515 | |
| 6402 | 043554 | 054522 | 024040 | 044502 | |
| 6403 | 043562 | 030124 | 024471 | 042440 | |
| 6404 | 043570 | 051122 | 051117 | 047440 | |
| 6405 | 043576 | 020106 | 052123 | 052101 | |
| 6406 | 043604 | 051525 | | | |
| 6407 | 043606 | 005015 | 042522 | 044507 | .ASCIZ <15><12>"REGISTER (CDS)" |
| 6408 | 043644 | 052123 | 051105 | 024040 | |
| 6409 | 043622 | 042103 | 024523 | 000 | |
| 6410 | | | | | |
| 6411 | 043627 | 104 | 051511 | 051501 | EM63: .ASCIZ "DISASTEROUS ERROR BUT NO ERROR BITS SET" |
| 6412 | 043634 | 042524 | 047522 | 051525 | |
| 6413 | 043642 | 042440 | 051122 | 051117 | |
| 6414 | 043650 | 041040 | 052125 | 047040 | |
| 6415 | 043656 | 020117 | 051105 | 047522 | |
| 6416 | 043664 | 020122 | 044502 | 051524 | |
| 6417 | 043672 | 051440 | 052105 | 000 | |
| 6418 | | | | | |
| 6419 | 043677 | 104 | 052101 | 020101 | EM64: .ASCII "DATA PACKING (BIT01) OF STATUS REGISTER (CDS)" |
| 6420 | 043704 | 040520 | 045503 | 047111 | |
| 6421 | 043712 | 020107 | 041050 | 052111 | |
| 6422 | 043720 | 030460 | 020051 | 043117 | |
| 6423 | 043726 | 051440 | 040524 | 052524 | |
| 6424 | 043734 | 020123 | 042522 | 044507 | |
| 6425 | 043742 | 052123 | 051105 | 024040 | |
| 6426 | 043750 | 042103 | 024523 | | |
| 6427 | 043754 | 005015 | 047516 | 020124 | .ASCIZ <15><12>"NOT SET - IT SHOULD BE" |
| 6428 | 043762 | 042523 | 020124 | 020055 | |
| 6429 | 043770 | 052111 | 051440 | 047510 | |
| 6430 | 043775 | 046125 | 020104 | 042502 | |
| 6431 | 044004 | 000 | | | |
| 6432 | | | | | |

| | | | | |
|------|--------|--------|--------|--------|
| 6433 | 044005 | 124 | 047510 | 046125 |
| 6434 | 044012 | 020104 | 042502 | 040440 |
| 6435 | 044020 | 047104 | 042522 | 051523 |
| 6436 | 044026 | 047111 | 020107 | 044502 |
| 6437 | 044034 | 040516 | 054522 | 042040 |
| 6438 | 044042 | 041505 | 113 | |
| 6439 | 044045 | 015 | 050012 | 047522 |
| 6440 | 044052 | 051107 | 046501 | 042040 |
| 6441 | 044060 | 042517 | 020123 | 047516 |
| 6442 | 044066 | 020124 | 043501 | 042522 |
| 6443 | 044074 | 000105 | | |
| 6444 | | | | |
| 6445 | 044076 | 047503 | 052116 | 047105 |
| 6446 | 044104 | 051524 | 047440 | 020106 |
| 6447 | 044112 | 052123 | 052101 | 051525 |
| 6448 | 044120 | 051040 | 043505 | 051511 |
| 6449 | 044126 | 042524 | 020122 | 041450 |
| 6450 | 044134 | 051504 | 020051 | 047111 |
| 6451 | 044142 | 047503 | 051122 | 041505 |
| 6452 | 044150 | 124 | | |
| 6453 | 044151 | 015 | 020012 | 020055 |
| 6454 | 044156 | 044123 | 052517 | 042114 |
| 6455 | 044164 | 041040 | 020105 | 042532 |
| 6456 | 044172 | 047522 | 000 | |
| 6457 | | | | |
| 6458 | 044175 | 117 | 042104 | 041040 |
| 6459 | 044202 | 051525 | 040440 | 042104 |
| 6460 | 044210 | 042522 | 051523 | 041440 |
| 6461 | 044216 | 052501 | 042523 | 020104 |
| 6462 | 044224 | 020101 | 051124 | 050101 |
| 6463 | 044232 | 044440 | 020116 | 047516 |
| 6464 | 044240 | 026516 | 040520 | 045503 |
| 6465 | 044246 | 046440 | 042117 | 000105 |
| 6466 | | | | |
| 6467 | | | | |
| 6468 | | | | |
| 6469 | 044254 | 024040 | 041520 | 020051 |
| 6470 | 044262 | 020040 | 024040 | 050123 |
| 6471 | 044270 | 020051 | 020040 | 041450 |
| 6472 | 044276 | 051504 | 020051 | 020040 |
| 6473 | 044304 | 041450 | 042104 | 020051 |
| 6474 | 044312 | 020040 | 024040 | 051520 |
| 6475 | 044320 | 000051 | | |
| 6476 | 044322 | 024040 | 041520 | 020051 |
| 6477 | 044330 | 020040 | 024040 | 050123 |
| 6478 | 044336 | 020051 | 020040 | 041450 |
| 6479 | 044344 | 051504 | 020051 | 020040 |
| 6480 | 044352 | 041450 | 042104 | 020051 |
| 6481 | 044360 | 020040 | 041450 | 041504 |
| 6482 | 044366 | 020051 | 020040 | 024040 |
| 6483 | 044374 | 051520 | 000051 | |
| 6484 | 044400 | 024040 | 041520 | 020051 |
| 6485 | 044406 | 020040 | 024040 | 050123 |
| 6486 | 044414 | 020051 | 020040 | 041450 |

EM65: .ASCII "SHOULD BE ADDRESSING BINARY DECK"

.ASCIZ <15><12>"PROGRAM DOES NOT AGREE"

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)"

| | | | | | |
|------|--------|--------|--------|--------|-------------------------------------------------------------------------|
| 6541 | 045104 | 020040 | 020040 | 044123 | |
| 6542 | 045112 | 000102 | | | |
| 6543 | 045114 | 024040 | 041520 | 020051 | DN7: .ASCII " (PC) (SP) (COS) (CDD) (CDD) (PS)" |
| 6544 | 045122 | 020040 | 051450 | 024520 | |
| 6545 | 045130 | 020040 | 024040 | 042103 | |
| 6546 | 045136 | 024523 | 020040 | 024040 | |
| 6547 | 045144 | 042103 | 024504 | 020040 | |
| 6548 | 045152 | 024040 | 042103 | 024504 | |
| 6549 | 045160 | 020040 | 020040 | 050050 | |
| 6550 | 045166 | 024523 | | | |
| 6551 | 045170 | 005015 | 020040 | 020040 | .ASCIZ <15><12>" WAS SHB" |
| 6552 | 045176 | 020040 | 020040 | 020040 | |
| 6553 | 045204 | 020040 | 020040 | 020040 | |
| 6554 | 045212 | 020040 | 020040 | 020040 | |
| 6555 | 045220 | 020040 | 040527 | 020123 | |
| 6556 | 045226 | 020040 | 020040 | 044123 | |
| 6557 | 045234 | 000102 | | | |
| 6558 | | | | | .EVEN |
| 6559 | | | | | |
| 6560 | | | | | ;ERROR ITEMS DATA TABLE |
| 6561 | | | | | |
| 6562 | 045236 | 001116 | 001174 | 001200 | DT1: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$REG7, 0 |
| 6563 | 045244 | 001202 | 001176 | 000000 | |
| 6564 | 045252 | 001116 | 001174 | 001200 | DT2: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$REG7, 0 |
| 6565 | 045260 | 001202 | 001204 | 001176 | |
| 6566 | 045266 | 000000 | | | |
| 6567 | 045270 | 001116 | 001174 | 001200 | DT3: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP3, \$REG7, 0 |
| 6568 | 045276 | 001202 | 001206 | 001176 | |
| 6569 | 045304 | 000000 | | | |
| 6570 | 045306 | 001116 | 001174 | 001200 | DT4: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$TMP3, \$REG7, 0 |
| 6571 | 045314 | 001202 | 001204 | 001206 | |
| 6572 | 045322 | 001176 | 000000 | | |
| 6573 | 045326 | 001116 | 001174 | 001200 | DT5: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP2, \$TMP3, \$TMP4, \$REG7, 0 |
| 6574 | 045334 | 001202 | 001204 | 001206 | |
| 6575 | 045342 | 001210 | 001176 | 000000 | |
| 6576 | 045350 | 001116 | 001174 | 001200 | DT6: \$ERRPC, \$REG6, \$TMP0, \$TMP1, \$TMP4, \$REG7, 0 |
| 6577 | 045356 | 001202 | 001210 | 001176 | |
| 6578 | 045364 | 000000 | | | |
| 6579 | | | | | |


```

6580
6581 ;*****
6582 ;SUBROUTINE TO INITIALIZE CSR AND DBR POINTERS
6583 ;*****
6584
6585 045366 013703 001232 SETUP: MOV CDST,CDS ;SET UP STATUS REGISTER POINTER
6586 045372 013704 001234 MOV CDCC,CCC ;SET UP COLUMN COUNT REGISTER POINTER
6587 045376 013705 001236 MOV CDBA,CDA ;SET UP BUS ADDRESS REGISTER POINTER
6588 045402 013702 001244 MOV INTVC,ADINT ;LOAD ADDRESS OF INTERRUPT VECTOR
6589 045406 013712 001246 MOV INTVC+2,(ADINT) ;SET UP CARD READER TRAP VECTOR
6590 045412 005077 133630 CLR @INTVC+2 ;TO HALT
6591 045416 005037 001252 CLR INTFLG ;INITIALIZE INTERRUPT FLAG
6592 045422 005037 001254 CLR TRFLG ;INITIALIZE TRACE FLAG
6593 045426 013746 000340 MOV PR7,-(SP) ;;PUT NEW PS ON STACK
6594 045432 012746 045440 MOV #64$,-(SP) ;;PUT NEW PC ON STACK
6595 045436 000002 RTI ;;POP NEW PC AND PS
6596 045440
6597 045440 000207 64$: RTS PC ;RETURN TO MAINLINE CODE
6598
6599 ;*****

```

;

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

;

```

6612 ;*****
6613 ;*****
6614 ;*****
6615 ;*****
6616
6617 045442 000000 BUFBEQ: 0
6618 000001 .END
6619

```


| | | | | | | | | |
|---------|--------|-------|-------|-------|-------|-------|------|-------|
| EM16 | 037613 | 531 | 6006# | | | | | |
| EM17 | 037677 | 539 | 6017# | | | | | |
| EM2 | 036367 | 438 | 5875# | | | | | |
| EM20 | 040026 | 547 | 6033# | | | | | |
| EM21 | 040054 | 554 | 6038# | | | | | |
| EM22 | 040124 | 561 | 6046# | | | | | |
| EM23 | 040201 | 568 | 6055# | | | | | |
| EM24 | 040232 | 575 | 6061# | | | | | |
| EM25 | 040306 | 582 | 6070# | | | | | |
| EM26 | 040364 | 589 | 6079# | | | | | |
| EM27 | 040461 | 597 | 6091# | | | | | |
| EM3 | 036470 | 446 | 5888# | | | | | |
| EM30 | 040576 | 505 | 6106# | | | | | |
| EM31 | 040670 | 613 | 6117# | | | | | |
| EM32 | 040762 | 628 | 6128# | | | | | |
| EM33 | 041044 | 637 | 6138# | | | | | |
| EM34 | 041127 | 646 | 6149# | | | | | |
| EM35 | 041247 | 654 | 6165# | | | | | |
| EM36 | 041353 | 669 | 6178# | | | | | |
| EM37 | 041432 | 678 | 6187# | | | | | |
| EM4 | 036570 | 454 | 5900# | | | | | |
| EM40 | 041501 | 685 | 6195# | | | | | |
| EM41 | 041564 | 693 | 6205# | | | | | |
| EM42 | 041646 | 701 | 6215# | | | | | |
| EM43 | 041743 | 709 | 6227# | | | | | |
| EM44 | 042032 | 717 | 6238# | | | | | |
| EM45 | 042122 | 725 | 6249# | | | | | |
| EM46 | 042230 | 733 | 6262# | | | | | |
| EM47 | 042335 | 741 | 6275# | | | | | |
| EM5 | 036633 | 461 | 5907# | | | | | |
| EM50 | 042444 | 749 | 6289# | | | | | |
| EM51 | 042552 | 757 | 6302# | | | | | |
| EM52 | 042657 | 765 | 6316# | | | | | |
| EM53 | 042766 | 773 | 6329# | | | | | |
| EM54 | 043115 | 782 | 6345# | | | | | |
| EM55 | 043204 | 790 | 6356# | | | | | |
| EM56 | 043250 | 797 | 6363# | | | | | |
| EM57 | 043313 | 804 | 6370# | | | | | |
| EM6 | 036734 | 469 | 5920# | | | | | |
| EM60 | 043376 | 812 | 6380# | | | | | |
| EM61 | 043452 | 819 | 6389# | | | | | |
| EM62 | 043533 | 827 | 6399# | | | | | |
| EM63 | 043527 | 662 | 6411# | | | | | |
| EM64 | 043677 | 835 | 6419# | | | | | |
| EM65 | 044005 | 843 | 6433# | | | | | |
| EM66 | 044076 | 851 | 6445# | | | | | |
| EM67 | 044175 | 859 | 6458# | | | | | |
| EM7 | 037024 | 477 | 5931# | | | | | |
| ENOCK | 012266 | 2380# | | | | | | |
| ERCD11 | 015642 | 315 | 3022# | | | | | |
| ERCD12 | 016146 | 3064 | 3067# | | | | | |
| ERFLG | 001260 | 407# | 2412# | 2820 | 2822* | 4398* | 4453 | 4455* |
| ERRVEC= | 000004 | 261# | 4640 | 4641* | 4643* | 4646* | | |
| ER1200 | 016134 | 3021 | 3064# | 4206 | | | | |

SIZE 015314
SP =%000006

| | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 2467* | 2469 | 2504 | 2505 | 2511 | 2663 | 2668 | 2794* | 2798 | 2804 | 2840 | 2965* | 936* |
| 191* | 873* | 898* | 899* | 893* | 897* | 898* | 904 | 905 | 906 | 925* | 926* | 936* |
| 937* | 1059* | 1060* | 1063* | 1064* | 1066 | 1067 | 1068* | 1069* | 1084* | 1085* | 1116* | 1117* |
| 1120* | 1121* | 1123 | 1124 | 1125* | 1126* | 1136* | 1137* | 1156* | 1157* | 1160* | 1161* | 1163 |
| 1164 | 1166* | 1167* | 1171* | 1172* | 1175* | 1176* | 1178 | 1179 | 1181* | 1182* | 1185* | 1186* |
| 1188 | 1189 | 1191* | 1192* | 1201* | 1202* | 1211 | 1226* | 1227* | 1230* | 1231* | 1233 | 1234 |
| 1236* | 1237* | 1241* | 1242* | 1245* | 1246* | 1248 | 1249 | 1257 | 1275* | 1276* | 1279* | 1280* |
| 1282 | 1283 | 1285* | 1286* | 1290* | 1291* | 1294* | 1295* | 1297 | 1298 | 1300* | 1301* | 1304* |
| 1305* | 1307 | 1308 | 1310* | 1311* | 1314* | 1315* | 1318* | 1319* | 1321 | 1322 | 1324* | 1325* |
| 1334* | 1335* | 1353 | 1358* | 1373* | 1374* | 1377* | 1378* | 1380 | 1381 | 1383* | 1384* | 1388* |
| 1389* | 1392* | 1393* | 1395 | 1396 | 1398* | 1399* | 1402* | 1403* | 1405 | 1406 | 1408* | 1409* |
| 1412* | 1413* | 1416* | 1417* | 1419 | 1420 | 1422* | 1423* | 1432* | 1433* | 1451 | 1456* | 1471* |
| 1472* | 1475* | 1476* | 1478 | 1479 | 1481* | 1482* | 1486* | 1487* | 1490* | 1491* | 1493 | 1494 |
| 1496* | 1497* | 1500* | 1501* | 1503 | 1504 | 1506* | 1507* | 1510* | 1511* | 1514* | 1515* | 1517 |
| 1518 | 1520* | 1521* | 1530* | 1531* | 1549 | 1554* | 1569* | 1570* | 1573* | 1574* | 1576 | 1577 |
| 1579* | 1580* | 1584* | 1585* | 1588* | 1589* | 1591 | 1592 | 1594* | 1595* | 1598* | 1599* | 1601 |
| 1602 | 1604* | 1605* | 1608* | 1609* | 1612* | 1613* | 1615 | 1616 | 1618* | 1619* | 1628* | 1629* |
| 1647 | 1662* | 1663* | 1666* | 1667* | 1669 | 1670 | 1672* | 1673* | 1677* | 1678* | 1681* | 1682* |
| 1684 | 1685 | 1687* | 1688* | 1691* | 1692* | 1694 | 1695 | 1697* | 1698* | 1701* | 1702* | 1705* |
| 1706* | 1708 | 1709 | 1711* | 1712* | 1721* | 1722* | 1740 | 1745* | 1760* | 1761* | 1764* | 1765* |
| 1767 | 1768 | 1770* | 1771* | 1775* | 1776* | 1779* | 1780* | 1782 | 1783 | 1785* | 1786* | 1789* |
| 1790* | 1792 | 1793 | 1795* | 1796* | 1799* | 1800* | 1803* | 1804* | 1806 | 1807 | 1809* | 1810* |
| 1819* | 1820* | 1838 | 1843* | 1858* | 1859* | 1862* | 1863* | 1865 | 1866 | 1868* | 1869* | 1873* |
| 1874* | 1877* | 1878* | 1880 | 1881 | 1883* | 1884* | 1887* | 1888* | 1890 | 1891 | 1893* | 1894* |
| 1897* | 1898* | 1901* | 1902* | 1904 | 1905 | 1907* | 1908* | 1917* | 1918* | 1932 | 1937* | 1953* |
| 1954* | 1957* | 1958* | 1960 | 1961 | 1963* | 1964* | 1968* | 1969* | 1972* | 1973* | 1975 | 1976 |
| 1977* | 1978* | 1988* | 1989* | 1995 | 2007* | 2008* | 2011* | 2012* | 2014 | 2015 | 2017* | 2018* |
| 2022* | 2023* | 2026* | 2027* | 2029 | 2030 | 2032* | 2033* | 2036* | 2037* | 2039 | 2040 | 2042* |
| 2043* | 2050 | 2052* | 2053* | 2058* | 2059* | 2063 | 2076* | 2077* | 2080* | 2081* | 2083 | 2084 |
| 2086* | 2087* | 2091* | 2092* | 2095* | 2096* | 2098 | 2099 | 2101* | 2102* | 2105* | 2106* | 2108 |
| 2109 | 2111* | 2112* | 2119 | 2268* | 2269* | 2272* | 2273* | 2275 | 2276 | 2278* | 2279* | 2283* |
| 2284* | 2287* | 2288* | 2290 | 2291 | 2293* | 2294* | 2297* | 2298* | 2300 | 2301 | 2303* | 2304* |
| 2307* | 2308* | 2311* | 2312* | 2314 | 2315 | 2317* | 2318* | 2330* | 2331* | 2350 | 2374 | 2425* |
| 2426* | 2432* | 2433* | 2441* | 2442* | 2445* | 2446* | 2448 | 2449 | 2451* | 2452* | 2456* | 2457* |
| 2460* | 2461* | 2463 | 2464 | 2595* | 2597* | 2602* | 2728* | 2733* | 2828* | 2833* | 2868 | 2869 |
| 2871* | 2888* | 2889* | 2901* | 2902* | 2905 | 2906* | 2907 | 2908* | 2913* | 2914* | 2953 | 2962 |
| 2983* | 2998* | 2999* | 3003* | 3007* | 3008* | 3014 | 3015 | 3016 | 3027* | 3042* | 3043* | 3047* |
| 3051* | 3052* | 3058 | 3059 | 3060 | 3159* | 3160* | 3163* | 3164* | 3166 | 3167 | 3169* | 3170* |
| 3174* | 3175* | 3178* | 3179* | 3181 | 3182 | 3184* | 3185* | 3188* | 3189* | 3191 | 3192 | 3194* |
| 3195* | 3223* | 3224* | 3227* | 3228* | 3230 | 3231 | 3233* | 3234* | 3238* | 3239* | 3242* | 3243* |
| 3245 | 3246 | 3248* | 3249* | 3252* | 3253* | 3255 | 3256 | 3258* | 3259* | 3265 | 3284* | 3285* |
| 3288* | 3289* | 3291 | 3292 | 3294* | 3295* | 3299* | 3300* | 3303* | 3304* | 3306 | 3307 | 3309* |
| 3310* | 3313* | 3314* | 3316 | 3317 | 3319* | 3320* | 3327 | 3375* | 3376* | 3379* | 3380* | 3382 |
| 3383 | 3385* | 3386* | 3390* | 3391* | 3394* | 3395* | 3397 | 3398 | 3400* | 3401* | 3404* | 3405* |
| 3407 | 3408 | 3410* | 3411* | 3420 | 3423* | 3424* | 3427* | 3428* | 3430 | 3431 | 3433* | 3434* |
| 3438* | 3439* | 3442* | 3443* | 3445 | 3446 | 3448* | 3449* | 3452* | 3453* | 3455 | 3456 | 3458* |
| 3459* | 3467 | 3503* | 3504* | 3507* | 3508* | 3510 | 3511 | 3513* | 3514* | 3518* | 3519* | 3522* |
| 3523* | 3525 | 3526 | 3528* | 3529* | 3532* | 3533* | 3535 | 3536 | 3538* | 3539* | 3547 | 3550* |
| 3551* | 3554* | 3555* | 3557 | 3558 | 3560* | 3561* | 3565* | 3566* | 3569* | 3570* | 3572 | 3573 |
| 3575* | 3576* | 3579* | 3580* | 3582 | 3583 | 3585* | 3585* | 3594 | 3658* | 3659* | 3662* | 3663* |
| 3665 | 3666 | 3668* | 3669* | 3673* | 3674* | 3677* | 3678* | 3680 | 3681 | 3683* | 3684* | 3687* |
| 3688* | 3690 | 3691 | 3693* | 3694* | 3703 | 3706* | 3707* | 3710* | 3711* | 3713 | 3714 | 3716* |
| 3717* | 3721* | 3722* | 3725* | 3726* | 3728 | 3729 | 3731* | 3732* | 3735* | 3736* | 3738 | 3739 |
| 3741* | 3742* | 3750 | 3814* | 3815* | 3818* | 3819* | 3821 | 3822 | 3824* | 3825* | 3829* | 3830* |

| | | | | | | | | | | | | | | |
|----------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|--|
| SW7 = | 000200 | 223# | | | | | | | | | | | | |
| SW8 = | 000400 | 222# | | | | | | | | | | | | |
| SW9 = | 001000 | 221# | | | | | | | | | | | | |
| TBITVE = | 000014 | 263# | 884* | 885* | 2994* | 2995* | 3038* | 3039* | 4231* | 4232* | 4341* | 4342* | | |
| TCRLF = | 000200 | 4701 | 4740# | | | | | | | | | | | |
| TEMP1 | 015322 | 2840* | 2843* | 2968# | | | | | | | | | | |
| TEMP2 | 015324 | 2841* | 2842* | 2845 | 2969# | | | | | | | | | |
| TEND | 015330 | 2416* | 2420* | 2522 | 2558 | 2581 | 2671 | 2689* | 2703 | 2718 | 2813* | 2971# | | |
| TESTX | 024100 | 318 | 4215# | | | | | | | | | | | |
| THT = | 000011 | 4699 | 4739# | | | | | | | | | | | |
| TINA21 | 010764 | 2051 | 2063# | | | | | | | | | | | |
| TINTC | 016652 | 3157 | 3206# | | | | | | | | | | | |
| TINTCA | 017120 | 3221 | 3265# | | | | | | | | | | | |
| TINTCB | 017402 | 3282 | 3327# | | | | | | | | | | | |
| TINTD | 017764 | 3373 | 3420# | | | | | | | | | | | |
| TINTDA | 020214 | 3421 | 3467# | | | | | | | | | | | |
| TINTE | 020544 | 3501 | 3547# | | | | | | | | | | | |
| TINTEA | 020774 | 3548 | 3594# | | | | | | | | | | | |
| TINTF | 021424 | 3656 | 3703# | | | | | | | | | | | |
| TINTFA | 021654 | 3704 | 3750# | | | | | | | | | | | |
| TINTG | 022270 | 3812 | 3858# | | | | | | | | | | | |
| TINTGA | 022520 | 3859 | 3905# | | | | | | | | | | | |
| TINTH | 024064 | 4156 | 4203# | | | | | | | | | | | |
| TINTI | 023120 | 3965 | 4014# | | | | | | | | | | | |
| TINTIA | 023156 | 4028 | 4030# | | | | | | | | | | | |
| TINT10 | 004150 | 1224 | 1256# | | | | | | | | | | | |
| TINT11 | 004556 | 1273 | 1347# | | | | | | | | | | | |
| TINT12 | 005234 | 1371 | 1445# | | | | | | | | | | | |
| TINT13 | 005712 | 1469 | 1543# | | | | | | | | | | | |
| TINT14 | 006370 | 1567 | 1641# | | | | | | | | | | | |
| TINT15 | 007032 | 1660 | 1734# | | | | | | | | | | | |
| TINT16 | 007510 | 1758 | 1832# | | | | | | | | | | | |
| TINT17 | 010150 | 1856 | 1926# | | | | | | | | | | | |
| TINT20 | 010452 | 1951 | 1994# | | | | | | | | | | | |
| TINT21 | 010726 | 2005 | 2050# | | | | | | | | | | | |
| TINT22 | 011240 | 2074 | 2119# | | | | | | | | | | | |
| TINT31 | 012146 | 2266 | 2343# | | | | | | | | | | | |
| TINT7 | 003740 | 1154 | 1208# | | | | | | | | | | | |
| TKVEC = | 000060 | 270# | | | | | | | | | | | | |
| TLOPC | 016642 | 3202# | 3203 | | | | | | | | | | | |
| TLOPG | 021720 | 3776# | 3777 | | | | | | | | | | | |
| TLOPGA | 021726 | 3778# | 3779 | | | | | | | | | | | |
| TLOPG8 | 021756 | 3787# | 3788 | | | | | | | | | | | |
| TLOPH | 023304 | 4077# | 4078 | | | | | | | | | | | |
| TLOPHA | 023312 | 4079# | 4080 | | | | | | | | | | | |
| TLOPH8 | 023342 | 4088# | 4089 | | | | | | | | | | | |
| TLOPHC | 023550 | 4137# | 4138 | | | | | | | | | | | |
| TLOPHD | 023556 | 4139# | 4140 | | | | | | | | | | | |
| TLOPHE | 023606 | 4149# | 4150 | | | | | | | | | | | |
| TOTCRO | 025734 | 4396* | 4411* | 4478 | 4497# | | | | | | | | | |
| TOTERR | 025732 | 4397* | 4450* | 4481 | 4496# | | | | | | | | | |
| TPVEC = | 000064 | 271# | | | | | | | | | | | | |
| TRACE = | 000360 | 281# | 936 | 2432 | | | | | | | | | | |
| TRAPVE = | 000034 | 269# | 878* | 879* | 2988* | 2989* | 3032* | 3033* | 4225* | 4226* | 4335* | 4336* | | |

| | | | | | | | | | | | | | | |
|---------|----------|-------|-------|-------|-------|-------|-------|-------|------|-------|------|------|------|-------|
| TRAPX | 002566 | 923 | 932# | | | | | | | | | | | |
| TRFLG | 001254 | 404# | 922 | 2384* | 2422 | 2956* | 6592* | | | | | | | |
| TRP1 | 012440 | 2423 | 2430# | | | | | | | | | | | |
| TRTRAP | 001242 | 309 | 399# | | | | | | | | | | | |
| TRIVEC= | 000014 | 264# | | | | | | | | | | | | |
| TSTART | 015326 | 2415* | 2419* | 2465 | 2524 | 2572 | 2578 | 2583 | 2673 | 2688* | 2709 | 2715 | 2720 | 2812* |
| | | 2970# | | | | | | | | | | | | |
| TSTM10 | 023256 | 3920 | 4060 | 4063# | | | | | | | | | | |
| TSTM12 | 022544 | 3919 | 3922# | | | | | | | | | | | |
| TST1 | 002620 | 921 | 931 | 944# | | | | | | | | | | |
| TST10 | 003770 | 1222# | | | | | | | | | | | | |
| TST11 | 004166 | 1271# | | | | | | | | | | | | |
| TST12 | 004644 | 1342 | 1344 | 1346 | 1362 | 1364 | 1369# | | | | | | | |
| TST13 | 005322 | 1440 | 1442 | 1444 | 1460 | 1462 | 1467# | | | | | | | |
| TST14 | 006000 | 1538 | 1540 | 1542 | 1558 | 1560 | 1565# | | | | | | | |
| TST15 | 006442 | 1636 | 1638 | 1640 | 1651 | 1653 | 1658# | | | | | | | |
| TST16 | 007120 | 1729 | 1731 | 1733 | 1749 | 1751 | 1756# | | | | | | | |
| TST17 | 007576 | 1827 | 1829 | 1831 | 1847 | 1849 | 1854# | | | | | | | |
| TST2 | 002724 | 982# | | | | | | | | | | | | |
| TST20 | 010236 | 1925 | 1941 | 1943 | 1949# | | | | | | | | | |
| TST21 | 010470 | 2003# | | | | | | | | | | | | |
| TST22 | 011002 | 2072# | | | | | | | | | | | | |
| TST23 | 011362 | 2160# | | | | | | | | | | | | |
| TST24 | 011410 | 2178# | | | | | | | | | | | | |
| TST25 | 011436 | 2195# | | | | | | | | | | | | |
| TST26 | 011466 | 2212# | | | | | | | | | | | | |
| TST27 | 011514 | 2230# | | | | | | | | | | | | |
| TST3 | 002772 | 989 | 1007# | | | | | | | | | | | |
| TST30 | 011542 | 2247# | | | | | | | | | | | | |
| TST31 | 011572 | 2264# | | | | | | | | | | | | |
| TST32 | 012206 | 2361# | | | | | | | | | | | | |
| TST33 | 016152 | 3072# | | | | | | | | | | | | |
| TST34 | 016250 | 3108# | | | | | | | | | | | | |
| TST35 | 016402 | 3153# | | | | | | | | | | | | |
| TST36 | 017442 | 3348# | | | | | | | | | | | | |
| TST37 | 020226 | 3476# | | | | | | | | | | | | |
| TST4 | 003042 | 1030# | | | | | | | | | | | | |
| TST40 | 021006 | 3603# | | | | | | | | | | | | |
| TST41 | 021666 | 3759# | | | | | | | | | | | | |
| TST42 | 022544 | 3926# | | | | | | | | | | | | |
| TST43 | 023256 | 4067# | | | | | | | | | | | | |
| TST5 | 003112 | 1053# | | | | | | | | | | | | |
| TST6 | 003300 | 1081 | 1094 | 1103# | | | | | | | | | | |
| TST7 | 003456 | 1134 | 1152# | | | | | | | | | | | |
| TTY | =%000005 | 280# | | | | | | | | | | | | |
| TYHEAD | 014554 | 2589 | 2726 | 2820# | | | | | | | | | | |
| TYPOS = | 104404 | 2829 | 2834 | 2882 | 4460 | 4479 | 4492 | 4921# | | | | | | |
| TYPE = | 104400 | 912 | 913 | 914 | 915 | 916 | 918 | 919 | 920 | 929 | 930 | 932 | 933 | 1357 |
| | | 1455 | 1553 | 1744 | 1842 | 1936 | 2383 | 2408 | 2601 | 2727 | 2732 | 2823 | 2824 | 2826 |
| | | 2831 | 2836 | 2880 | 2883 | 3017 | 3018 | 3061 | 3067 | 3076 | 3077 | 3078 | 3083 | 3113 |
| | | 3114 | 3115 | 3116 | 3130 | 3131 | 3132 | 3199 | 3200 | 3201 | 3323 | 3324 | 3352 | 3353 |
| | | 3354 | 3355 | 3415 | 3416 | 3417 | 3479 | 3480 | 3482 | 3483 | 3543 | 3544 | 3610 | 3611 |
| | | 3612 | 3613 | 3615 | 3616 | 3698 | 3699 | 3700 | 3764 | 3765 | 3766 | 3774 | 3775 | 3854 |
| | | 3855 | 3928 | 3929 | 3930 | 3931 | 3932 | 3937 | 3938 | 3939 | 4049 | 4050 | 4051 | 4052 |

| | | | | | | | | | | | | | | |
|----------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| \$NULL | 001152 | 362# | 4710 | 4742 | | | | | | | | | | |
| \$NWTST= | 000001 | 941# | 979# | 1004# | 1027# | 1050# | 1100# | 1149# | 1219# | 1269# | 1366# | 1464# | 1562# | 1655# |
| | | 1753# | 1851# | 1946# | 2000# | 2069# | 2157# | 2175# | 2192# | 2209# | 2227# | 2244# | 2261# | 2358# |
| | | 3069# | 3105# | 3152# | 3345# | 3473# | 3600# | 3756# | 3923# | 4064# | | | | |
| \$OCNT | 027144 | 4776* | 4805* | 4818# | | | | | | | | | | |
| \$OMODE | 027146 | 4771* | 4775* | 4780 | 4783* | 4794* | 4820# | | | | | | | |
| \$OVER | 026470 | 4636 | 4647 | 4659 | 4664# | | | | | | | | | |
| \$PASS | 001100 | 339# | 2872* | 2873* | 2881 | 2920 | 4655 | 4668 | | | | | | |
| \$PWRAO | 027552 | 4957# | | | | | | | | | | | | |
| \$PWROD | 027430 | 880 | 2990 | 3034 | 4227 | 4337 | 4928# | 4952 | | | | | | |
| \$PWRMG | 027546 | 4955# | | | | | | | | | | | | |
| \$PWRLP | 027476 | 4937 | 4942# | | | | | | | | | | | |
| \$QJES | 001226 | 386# | 4620 | 4742 | | | | | | | | | | |
| \$ROCHR= | ***** | U | 4922 | | | | | | | | | | | |
| \$RODEC= | ***** | U | 4922 | | | | | | | | | | | |
| \$ROLIN= | ***** | U | 4922 | | | | | | | | | | | |
| \$ROCT= | ***** | U | 4922 | | | | | | | | | | | |
| \$REGAD | 001156 | 366# | | | | | | | | | | | | |
| \$REGO | 001160 | 368# | | | | | | | | | | | | |
| \$REG1 | 001162 | 369# | | | | | | | | | | | | |
| \$REG2 | 001164 | 370# | | | | | | | | | | | | |
| \$REG3 | 001166 | 371# | | | | | | | | | | | | |
| \$REG4 | 001170 | 372# | | | | | | | | | | | | |
| \$REG5 | 001172 | 373# | | | | | | | | | | | | |
| \$REG6 | 001174 | 374# | 4591* | 6562 | 6564 | 6567 | 6570 | 6573 | 6576 | | | | | |
| \$REG7 | 001176 | 375# | 4592* | 6562 | 6564 | 6567 | 6570 | 6573 | 6576 | | | | | |
| \$RTRN | 015140 | 884 | 886* | 891* | 2890 | 2915# | 2994 | 2996* | 3001* | 3038 | 3040* | 3045* | 4231 | 4233* |
| | | 4238* | 4341 | 4343* | 4348* | | | | | | | | | |
| \$R2A = | ***** | U | 4922 | | | | | | | | | | | |
| \$SAVRE= | ***** | U | 4922 | | | | | | | | | | | |
| \$SAVR6 | 027570 | 4936* | 4942 | 4943* | 4944* | 4962# | | | | | | | | |
| \$SCOPE | 026326 | 874 | 2984 | 3028 | 4221 | 4331 | 4634# | | | | | | | |
| \$SETUP= | 000037 | 328# | 874 | 876 | 878 | 880 | 882 | 883 | 884 | 896 | 2870 | 2984 | 2986 | 2988 |
| | | 2990 | 2992 | 2993 | 2994 | 3006 | 3028 | 3030 | 3032 | 3034 | 3036 | 3037 | 3038 | 3050 |
| | | 4221 | 4223 | 4225 | 4227 | 4229 | 4230 | 4231 | 4243 | 4331 | 4333 | 4335 | 4337 | 4339 |
| | | 4340 | 4341 | 4353 | 4615 | | | | | | | | | |
| \$STUP = | 177777 | 328# | | | | | | | | | | | | |
| \$SVLAD | 026460 | 4644 | 4662# | | | | | | | | | | | |
| \$SVPC = | 000204 | 301# | 306 | | | | | | | | | | | |
| \$SWR = | 176000 | 8# | 20 | 384 | 385 | 883 | 884 | 896 | 897 | 945 | 983 | 1008 | 1031 | 1054 |
| | | 1104 | 1153 | 1223 | 1272 | 1370 | 1468 | 1566 | 1659 | 1757 | 1855 | 1950 | 2004 | 2073 |
| | | 2161 | 2179 | 2196 | 2213 | 2231 | 2248 | 2265 | 2362 | 2863 | 2871 | 2885 | 2901 | 2920 |
| | | 2993 | 2994 | 3006 | 3007 | 3037 | 3038 | 3050 | 3051 | 3073 | 3109 | 3154 | 3349 | 3477 |
| | | 3604 | 3760 | 3927 | 4068 | 4230 | 4231 | 4243 | 4244 | 4340 | 4341 | 4353 | 4354 | 4583 |
| | | 4584 | 4585 | 4586 | 4587 | 4600 | 4607 | 4612 | 4618 | 4620 | 4628 | 4629 | 4630 | 4631 |
| | | 4635 | 4647 | 4649 | 4650 | 4651 | 4652 | 4653 | 4664 | 4667 | 4958 | | | |
| \$SWRMK= | 000000 | 4631 | | | | | | | | | | | | |
| \$TBIT | 015146 | 895* | 2911* | 2920# | 3005* | 3049* | 4242* | 4352* | | | | | | |
| \$TIMES | 001220 | 384# | 883# | 2871* | 2993* | 3037* | 4230* | 4340* | 4652* | 4658 | 4561* | 4667 | | |
| \$TKB | 001144 | 359# | 4115 | | | | | | | | | | | |
| \$TKS | 001142 | 358# | 4113 | | | | | | | | | | | |
| \$TMPO | 001200 | 376# | 2515* | 2516* | 2517 | 2550* | 2551* | 2552 | 2564* | 2565* | 2566 | 2593* | 2594* | 2595 |
| | | 4424* | 4425* | 4427 | 4593* | 6562 | 6564 | 6567 | 6570 | 6573 | 6576 | | | |
| \$TMP1 | 001202 | 377# | 4115* | 4118 | 4119* | 4121 | 4123 | 4594* | 6562 | 6564 | 6567 | 6570 | 6573 | 6576 |

| | | | | | | | | | | | | | | |
|----------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| STMP2 | 001204 | 378# | 4595* | 6564 | 6570 | 6573 | | | | | | | | |
| STMP3 | 001206 | 379# | 4596* | 6567 | 6570 | 6573 | | | | | | | | |
| STMP4 | 001210 | 380# | 965* | 974* | 2148* | 2153* | 2170* | 2187* | 2204* | 2222* | 2239* | 2256* | 2353* | 2500* |
| | | 2508* | 2660* | 2666* | 2677* | 6573 | 6576 | | | | | | | |
| STMP5 | 001212 | 381# | 2409* | 2528* | 2584 | 2721 | 2850* | | | | | | | |
| STMP6 | 001214 | 382# | 1164* | 1165* | 1166 | 1189* | 1190* | 1191 | 1234* | 1235* | 1236 | 1283* | 1284* | 1285 |
| | | 1308* | 1309* | 1310 | 1322* | 1323* | 1324 | 1381* | 1382* | 1383 | 1406* | 1407* | 1408 | 1420* |
| | | 1421* | 1422 | 1479* | 1480* | 1481 | 1504* | 1505* | 1506 | 1518* | 1519* | 1520 | 1577* | 1578* |
| | | 1579 | 1602* | 1603* | 1604 | 1616* | 1617* | 1618 | 1670* | 1671* | 1672 | 1695* | 1696* | 1697 |
| | | 1709* | 1710* | 1711 | 1768* | 1769* | 1770 | 1793* | 1794* | 1795 | 1807* | 1808* | 1809 | 1866* |
| | | 1867* | 1868 | 1891* | 1892* | 1893 | 1905* | 1906* | 1907 | 1961* | 1962* | 1963 | 2015* | 2016* |
| | | 2017 | 2040* | 2041* | 2042 | 2084* | 2085* | 2086 | 2109* | 2110* | 2111 | 2276* | 2277* | 2278 |
| | | 2301* | 2302* | 2303 | 2315* | 2316* | 2317 | 2449* | 2450* | 2451 | 3167* | 3168* | 3169 | 3192* |
| | | 3193* | 3194 | 3231* | 3232* | 3233 | 3256* | 3257* | 3258 | 3292* | 3293* | 3294 | 3317* | 3318* |
| | | 3319 | 3383* | 3384* | 3385 | 3408* | 3409* | 3410 | 3431* | 3432* | 3433 | 3456* | 3457* | 3458 |
| | | 3511* | 3512* | 3513 | 3536* | 3537* | 3538 | 3558* | 3559* | 3560 | 3583* | 3584* | 3585 | 3666* |
| | | 3667* | 3668 | 3691* | 3692* | 3693 | 3714* | 3715* | 3716 | 3739* | 3740* | 3741 | 3822* | 3823* |
| | | 3824 | 3847* | 3848* | 3849 | 3869* | 3870* | 3871 | 3894* | 3895* | 3896 | 3976* | 3977* | 3978 |
| | | 4001* | 4002* | 4003 | 4166* | 4167* | 4168 | 4191* | 4192* | 4193 | 4275* | 4276* | 4277 | 4292* |
| | | 4293* | 4294 | | | | | | | | | | | |
| STMP7 | 001216 | 383# | | | | | | | | | | | | |
| STN = | 000044 | 20# | 941 | 945# | 979 | 983# | 1004 | 1008# | 1027 | 1031# | 1050 | 1054# | 1100 | 1104# |
| | | 1149 | 1153# | 1219 | 1223# | 1268 | 1272# | 1366 | 1370# | 1464 | 1466# | 1562 | 1566# | 1655 |
| | | 1659# | 1753 | 1757# | 1851 | 1855# | 1946 | 1950# | 2000 | 2004# | 2069 | 2073# | 2157 | 2161# |
| | | 2175 | 2179# | 2192 | 2196# | 2209 | 2213# | 2227 | 2231# | 2244 | 2248# | 2261 | 2265# | 2358 |
| | | 2362# | 3069 | 3073# | 3105 | 3109# | 3152 | 3154# | 3345 | 3349# | 3473 | 3477# | 3600 | 3604# |
| | | 3756 | 3760# | 3923 | 3927# | 4064 | 4068# | | | | | | | |
| STPB | 001150 | 361# | 4118* | 4728* | 4742 | | | | | | | | | |
| STPFLG | 001155 | 365# | 4687 | 4742 | | | | | | | | | | |
| STPS | 001146 | 360# | 4116 | 4726 | 4742 | | | | | | | | | |
| STRAP | 027374 | 878 | 2988 | 3032 | 4225 | 4335 | 4900# | | | | | | | |
| STRAP = | 000005 | 4908# | 4918# | 4919# | 4920# | 4921# | 4922# | | | | | | | |
| STRPAD | 027416 | 4905 | 4916# | | | | | | | | | | | |
| STSTNM | 001102 | 340# | 2870* | 4599 | 4620 | 4627 | 4662* | 4664 | 4668 | | | | | |
| STYPBN= | ***** U | 4922 | | | | | | | | | | | | |
| STYPOS | 027150 | 4835# | 4921 | | | | | | | | | | | |
| STYPE | 026506 | 4687# | 4908 | 4917 | | | | | | | | | | |
| STYPEC | 026652 | 4707 | 4714 | 4721 | 4726# | 4727 | | | | | | | | |
| STYPEX | 026720 | 4732 | 4734 | 4737# | | | | | | | | | | |
| STYPOC | 026746 | 4774# | 4918 | | | | | | | | | | | |
| STYPON | 026762 | 4773 | 4776# | 4920 | | | | | | | | | | |
| STYPOS | 026722 | 4769# | 4919 | | | | | | | | | | | |
| \$XTSTR | 026336 | 4638# | | | | | | | | | | | | |
| \$OFILL | 027145 | 4770* | 4774* | 4784 | 4819# | | | | | | | | | |
| \$4OCAT= | ***** U | 4609 | 4635 | | | | | | | | | | | |
| | = 045444 | 286# | 290# | 301 | 302# | 304# | 306# | 308# | 314# | 317# | 321# | 325# | 337# | 389 |
| | | 872 | 896 | 2049 | 2118 | 2920 | 2925 | 2982 | 3006 | 3026 | 3050 | 3263 | 3325 | 3418 |
| | | 3465 | 3545 | 3592 | 3701 | 3748 | 3856 | 3903 | 4011 | 4201 | 4219 | 4243 | 4329 | 4353 |
| | | 4574# | 4620 | 4667 | 4668 | 4742 | 4889# | 4939 | 4961 | 5859# | | | | |

K01

SEQ 0217

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| CLR | 4716 | 4725 | 4732 | 4773 | 4788 | 4809 | 4854 | 4871 | 4939 | 4961 | 1156 | 1171 | 1181 | 1215 | 1217 |
| | 870 | 883 | 888 | 895 | 992 | 1059 | 1072 | 1107 | 1114 | 1116 | 1350 | 1352 | 1373 | 1388 | 1398 |
| | 1226 | 1241 | 1258 | 1260 | 1275 | 1290 | 1300 | 1314 | 1338 | 1340 | 1536 | 1546 | 1548 | 1569 | 1584 |
| | 1412 | 1436 | 1438 | 1448 | 1450 | 1471 | 1486 | 1496 | 1510 | 1534 | 1725 | 1727 | 1737 | 1739 | 1760 |
| | 1594 | 1608 | 1632 | 1634 | 1644 | 1646 | 1662 | 1677 | 1687 | 1701 | 1897 | 1921 | 1923 | 1929 | 1931 |
| | 1775 | 1785 | 1799 | 1823 | 1825 | 1835 | 1837 | 1858 | 1873 | 1883 | 2066 | 2076 | 2091 | 2101 | 2120 |
| | 1353 | 1968 | 1984 | 1992 | 1996 | 1997 | 2007 | 2022 | 2032 | 2065 | 2293 | 2307 | 2336 | 2338 | 2346 |
| | 2161 | 2179 | 2187 | 2196 | 2213 | 2231 | 2239 | 2248 | 2268 | 2283 | 2586 | 2660 | 2723 | 2849 | 2850 |
| | 2348 | 2377 | 2409 | 2410 | 2411 | 2412 | 2441 | 2456 | 2483 | 2500 | 3037 | 3042 | 3049 | 3159 | 3174 |
| | 2870 | 2871 | 2888 | 2901 | 2980 | 2993 | 2998 | 3005 | 3020 | 3024 | 3423 | 3438 | 3448 | 3503 | 3518 |
| | 3184 | 3223 | 3238 | 3248 | 3284 | 3299 | 3309 | 3375 | 3390 | 3400 | 3814 | 3829 | 3839 | 3861 | 3876 |
| | 3528 | 3550 | 3565 | 3575 | 3658 | 3673 | 3683 | 3706 | 3721 | 3731 | 4242 | 4267 | 4284 | 4327 | 4340 |
| | 3886 | 3968 | 3983 | 3993 | 4158 | 4173 | 4183 | 4217 | 4230 | 4235 | 4846 | 4849 | 4943 | 6590 | 6591 |
| | 4345 | 4352 | 4377 | 4396 | 4397 | 4398 | 4406 | 4538 | 4652 | 4786 | | | | | |
| | 6592 | | | | | | | | | | | | | | |
| CLRB | 4399 | 4651 | 4731 | 4875 | | | | | | | | | | | |
| CMP | 871 | 904 | 947 | 962 | 970 | 986 | 993 | 999 | 1009 | 1013 | 1022 | 1032 | 1036 | 1045 | 1211 |
| | 1257 | 1343 | 1353 | 1363 | 1441 | 1451 | 1461 | 1539 | 1549 | 1559 | 1637 | 1647 | 1652 | 1730 | 1740 |
| | 1750 | 1828 | 1838 | 1848 | 1932 | 1942 | 1995 | 2050 | 2063 | 2119 | 2146 | 2151 | 2167 | 2202 | 2219 |
| | 2254 | 2350 | 2351 | 2374 | 2506 | 2517 | 2520 | 2522 | 2529 | 2552 | 2555 | 2558 | 2566 | 2569 | 2572 |
| | 2578 | 2581 | 2606 | 2664 | 2671 | 2678 | 2703 | 2709 | 2715 | 2718 | 2845 | 2852 | 2869 | 2953 | 2962 |
| | 2981 | 3014 | 3025 | 3056 | 3265 | 3327 | 3420 | 3467 | 3468 | 3547 | 3594 | 3595 | 3624 | 3703 | 3750 |
| | 3751 | 3780 | 3858 | 3905 | 3906 | 4014 | 4030 | 4081 | 4141 | 4203 | 4218 | 4251 | 4328 | 4361 | 4421 |
| | 4427 | 4430 | 4488 | 4511 | 4615 | 4645 | 4658 | 4867 | | | | | | | |
| CMPB | 2669 | 2700 | 2706 | 2737 | 4121 | 4123 | 4418 | 4445 | 4699 | 4701 | 4708 | 4729 | 4733 | | |
| COM | 2163 | 2215 | 2384 | 2911 | 2956 | | | | | | | | | | |
| COMB | 4405 | | | | | | | | | | | | | | |
| DEC | 1131 | 2830 | 2835 | 2874 | 4390 | 4462 | 4546 | | | | | | | | |
| DECB | 4712 | 4715 | 4794 | 4805 | | | | | | | | | | | |
| EMT | 173 | | | | | | | | | | | | | | |
| HALT | 290 | 917 | 2609 | 2740 | 2928 | 3086 | 3117 | 3133 | 3356 | 3484 | 3617 | 3933 | 3941 | 4053 | 4202 |
| | 4258 | 4263 | 4373 | 4394 | 4485 | 4523 | 4614 | 4617 | 4689 | 4938 | 4960 | 3933 | 3941 | 4053 | 4202 |
| INC | 1073 | 1082 | 1110 | 1985 | 2368 | 2477 | 2525 | 2527 | 2546 | 2674 | 2676 | 2699 | 2807 | 2822 | 2827 |
| | 2832 | 2842 | 2851 | 2872 | 3087 | 3262 | 3464 | 3591 | 3630 | 3747 | 3786 | 3902 | 4010 | 4087 | 4147 |
| | 4388 | 4410 | 4411 | 4420 | 4429 | 4450 | 4455 | 4458 | 4486 | 4603 | 4657 | 4800 | 4808 | 4853 | 4944 |
| INCB | 4597 | 4662 | 4735 | | | | | | | | | | | | |
| IOT | 174 | | | | | | | | | | | | | | |
| JMP | 295 | 315 | 318 | 322 | 326 | 2342 | 2385 | 2492 | 2497 | 2540 | 2610 | 2629 | 2653 | 2657 | 2693 |
| | 2766 | 2768 | 2788 | 2919 | 2959 | 2960 | 2963 | 3021 | 3920 | 4206 | 4305 | 4490 | 4493 | 4964 | |
| JSR | 909 | 945 | 1055 | 1106 | 1153 | 1223 | 1272 | 1370 | 1468 | 1566 | 1659 | 1757 | 1855 | 1950 | 2004 |
| | 2073 | 2265 | 2362 | 2436 | 2476 | 2531 | 2535 | 2539 | 2589 | 2625 | 2680 | 2684 | 2726 | 2764 | 2802 |
| | 2896 | 3066 | 3075 | 3112 | 3156 | 3351 | 3478 | 3609 | 3763 | 3927 | 3940 | 4071 | 4254 | 4369 | 4395 |
| | 4507 | 4609 | 4707 | 4714 | 4721 | | | | | | | | | | |
| MOV | 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 | 925 | 926 | 936 |
| | 937 | 965 | 974 | 998 | 1008 | 1031 | 1056 | 1057 | 1060 | 1061 | 1063 | 1064 | 1066 | 1067 | 1068 |
| | 1069 | 1084 | 1085 | 1108 | 1109 | 1117 | 1118 | 1120 | 1121 | 1123 | 1124 | 1125 | 1126 | 1136 | 1137 |
| | 1154 | 1157 | 1158 | 1160 | 1161 | 1163 | 1164 | 1166 | 1167 | 1172 | 1173 | 1175 | 1176 | 1178 | 1179 |
| | 1182 | 1183 | 1185 | 1186 | 1188 | 1189 | 1191 | 1192 | 1195 | 1196 | 1197 | 1201 | 1202 | 1216 | 1224 |
| | 1227 | 1228 | 1230 | 1231 | 1233 | 1234 | 1236 | 1237 | 1242 | 1243 | 1245 | 1246 | 1248 | 1249 | 1250 |
| | 1251 | 1252 | 1259 | 1273 | 1276 | 1277 | 1279 | 1280 | 1282 | 1283 | 1285 | 1286 | 1291 | 1292 | 1294 |
| | 1295 | 1297 | 1298 | 1301 | 1302 | 1304 | 1305 | 1307 | 1308 | 1310 | 1311 | 1315 | 1316 | 1318 | 1319 |
| | 1321 | 1322 | 1324 | 1325 | 1328 | 1329 | 1330 | 1334 | 1335 | 1339 | 1351 | 1356 | 1358 | 1371 | 1374 |
| | 1375 | 1377 | 1378 | 1380 | 1381 | 1383 | 1384 | 1389 | 1390 | 1392 | 1393 | 1395 | 1396 | 1399 | 1400 |

| | | | | | | | | | | | | | | | |
|--------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | 4231 | 4341 | 4526 | 4908 | | | | | | | | | | | |
| .MCALL | 4 | 5 | 6 | 273 | 897 | 3007 | 3051 | 4244 | 4354 | | | | | | |
| .NLIST | 1 | 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 | 941 | 945 | 979 | 983 |
| | 1004 | 1008 | 1027 | 1031 | 1050 | 1054 | 1100 | 1104 | 1149 | 1153 | 1219 | 1223 | 1268 | 1272 | 1366 |
| | 1370 | 1454 | 1468 | 1562 | 1566 | 1655 | 1659 | 1753 | 1757 | 1851 | 1855 | 1946 | 1950 | 2000 | 2004 |
| | 2069 | 2073 | 2157 | 2161 | 2175 | 2179 | 2192 | 2196 | 2209 | 2213 | 2227 | 2231 | 2244 | 2248 | 2261 |
| | 2265 | 2358 | 2362 | 2388 | 2398 | 2402 | 2870 | 2976 | 3007 | 3051 | 3069 | 3073 | 3105 | 3109 | 3152 |
| | 3154 | 3345 | 3349 | 3473 | 3477 | 3600 | 3604 | 3756 | 3760 | 3923 | 3927 | 4064 | 4068 | 4208 | 4244 |
| | 4309 | 4354 | 4502 | 4615 | 4631 | 4908 | 4917 | 4918 | 4919 | 4920 | 4921 | 4922 | 4975 | 6600 | 6602 |
| | 6604 | 6612 | 6614 | 6616 | | | | | | | | | | | |
| .PAGE | 42 | 330 | 389 | | | | | | | | | | | | |
| .REPT | 290 | 368 | 376 | 2389 | 2395 | | | | | | | | | | |
| .SBTTL | 43 | 138 | 169 | 284 | 294 | 299 | 332 | 412 | 865 | 941 | 979 | 1004 | 1027 | 1050 | 1100 |
| | 1149 | 1219 | 1268 | 1366 | 1464 | 1562 | 1655 | 1753 | 1851 | 1946 | 2000 | 2069 | 2157 | 2175 | 2192 |
| | 2209 | 2227 | 2244 | 2261 | 2358 | 2402 | 2859 | 2976 | 3069 | 3105 | 3345 | 3473 | 3600 | 3756 | 3923 |
| | 4064 | 4208 | 4309 | 4502 | 4529 | 4578 | 4623 | 4671 | 4745 | 4824 | 4893 | 4909 | 4925 | 4975 | |
| .TITLE | 10 | | | | | | | | | | | | | | |
| .WORD | 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 | 2875 | 2878 | 4554 | 4559 | 4736 | 4820 | 4955 |
| | 4957 | 4967 | | | | | | | | | | | | | |

ERRORS DETECTED: 0

002

MAINDEC - 11 - DZCDB-B MACY11 27(654) 1-JUL-77 08:39 PAGE 151
DZCDB.P11

SEQ 0223

*DZCDB,DZCDB/CRF/SOL=DZCDB.P11
RUN-TIME: 27 25 4 SECONDS
CORE USED: 20K

EOF10ZCDBBSEQ

00010000

770920

PDP10 411