

DX11B

ON LINE EXERCISER
CZDXHD0

AH-8773D-MC

COPYRIGHT 72-78

FICHE 1 OF 1

DEC 1978

digital

MADE IN USA

This microfiche card contains a grid of frames. The first column of frames contains text, likely instructions or data labels. The subsequent columns contain various types of data, including tables, diagrams, and lists. The frames are arranged in a regular grid pattern, typical of microfiche format.

B 1
IDENTIFICATION

SEQ 0001

PRODUCT CODE: AC-8772D-MC
PRODUCT NAME: CZDXHDO DX11B ONLINE EXER
PRODUCT DATE: JULY ,1978
MAINTAINER: DIAGNOSTIC ENGINEERING

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1972,1978 BY DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

THE FUNCTION OF THE DX11B DIAGNOSTICS IS TO VERIFY THAT THE DX11B IMPLEMENTS THE FUNCTIONAL FLOW DIAGRAMS ILLUSTRATED IN THE DX11B PRINT SET. THE DX11 DIAGNOSTIC PACKAGE CONSISTS OF FOUR TAPES

1. DZDXA-[REV] MAINTENANCE CLOCK #1
2. DXDXF-[REV] MAINTENANCE CLOCK #2
3. DZDXG-[REV] DX OFFLINE DIAGNOSTIC EXERCISER
4. CZDXH-[REV] DX11B ONLINE EXER

THE DIAGNOSTICS WERE DIVIDED INTO FOUR TAPES BECAUSE OF THE 8K WORD MEMORY LIMIT REQUIRED TO SUPPORT MINIMUM SYSTEMS AND FOR FUNCTIONAL SAFEGUARDS. IT WAS FELT THAT SAFEGUARDS SHOULD BE TAKEN TO INSURE THAT NO ONE INADVERTENTLY RAN THE ONLINE-MAINTENANCE-CABLE EXERCISER WHILE CONNECTED ONLINE TO IBM. IT WAS ALSO FELT THAT THE FUNCTIONAL SEPARATION OF TESTS WOULD FACILITATE ADAPTION TO ACT11 AND DDP TESTING. THERE ARE ALSO TWO OTHER MAINDEC'S SUPPORTED BY DIAGNOSTICS THAT RUN THE DX11B:

1. COMMUNICATION TEST PROGRAM (CTP)
2. GENERAL TEST PROGRAM (GTP) WITH DX OVERLAY
3. DEC/X11 WITH DX SOFTWARE MODULE

THESE TESTS OPERATE IN THE MAINTENANCE MODE AND WERE DESIGNED TO DETECT UNIBUS DEVICE INTERACTION PROBLEMS. ADDITIONALLY CTP HAS A 'RESPONDER' MODE SO THAT INTERACTION PROBLEMS MAY BE DETECTED WHILE RUNNING ONLINE.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11 (MINIMUM 8K WORDS MEMORY)
ASR-33 (OR EQUIVALENT)
DX11B

2.2 STORAGE

ALL PROGRAMS LOAD IN 8K OF MEMORY

2.3 OTHER

A WORKING KNOWLEDGE OF ODT VERSION V006A.ODT IS NECESSARY

3. LOADING PROCEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED USING THE ABSOLUTE LOADER.

ABSOLUTE LOADER START ADDRESS *500

MEMORY *
SIZE

4K	17
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

3.1.1 LOAD ADDRESS OF ABS LOADER INTO SWITCHES

3.1.2 DEPRESS 'LOAD ADDRESS' KEY ON CONSOLE

3.1.3 DEPRESS 'START' KEY ON CONSOLE

4. STARTING PROCEDURE

 ONLINE DIAGNOSTIC REQUIRES THAT IT BE MAINTENANCE
 CABLED -SEE MANUAL FOR DETAILS

 NEVER NEVER NEVER RUN THE ONLINE- MAINTENANCE CABLED
 EXERCISER WHILE CONNECTED TO IBM

- A. SET SWITCH REGISTER TO 000200
- B. DEPRESS 'LOAD ADDRESS' KEY
- C. DEPRESS START

THE PROGRAM WILL JUMP TO THE DIAGNOSTIC MONITOR AND
 TYPE OUT THE OPERATING INSTRUCTIONS. THIS IS ONCE ONLY
 CODE. TO RETYPE THE OPERATING INSTRUCTION THE OPERATOR
 MAY EITHER RELOAD THE PROGRAM OR LOAD THE ADDRESS
 'MONITOR' IN THE SWITCH REGISTER AND DEPRESS START.

4.1 CONTROL SWITCH SETTINGS

SR 15	HALT ON ERROR
SR 14	SCOPE ON TEST OR ERROR
SR 13	INHIBIT PRINTING
SR 12	TYPE SHORT ERROR REPORT
SR 11	INHIBIT INTERATIONS
SR 10	CONTROL MAINTENANCE CLOCK (MAINT. CLK. TEST ONLY)
SR 9	ODT TRAP ON ERROR
SR 2	MULTIPLEXER CHANNEL
SR 1	SET BUSY ENABLE

4.2 STARTING ADDRESSES

ADDRESSES	COMMENT
000200	NORMAL START WITH 200 LEFT IN THE SWITCHES THE PROGRAMS TYPE OUT FULL INSTRUCTIONS ONCE AND ABBREVIATED INSTRUCTIONS THEREAFTER. WITH THE SWITCHES ZERO THE PROGRAMS SET UP EITHER THE DEFAULT OR PREVIOUSLY SELECTED PARAMETERS AND IMMEDIATELY ASKS FOR THE DYNAMIC SWITCH SETTINGS
MONITOR	RELOAD TAPE FOR RETYPING OF INSTRUCTIONS
000042	IF THIS LOCATION IS NONZERO THE PROGRAM ASSUMES IT IS RUNNING UNDER ACT11 OR DDP AND USES THE DEFAULT PARAMETERS
0.ODT	ENTRANCE TO ODT-11X VERSION V006A.ODT MAY START THE PROGRAM BY TYPE 200;G <CR>. (MAINTENANCE CLOCK TESTS ONLY)

NOTICE: HE WHO USES ODT IN A MEANS OTHER THAN EXPLICITLY
DIRECTED BY THIS DOCUMENT DOES SO AT HIS OWN RISK.

5. OPERATING PROCEDURE

STARTING FROM 200 WITH SR<07> UP CAUSES THE FOLLOWING GENERAL
TYPEOUT:

AC-XXXX[REV]-MC (TEST DESCRIPTION)

TYPE: <D>, FOR DEFAULT PARAMETERS
<P>, FOR PREVIOUS PARAMETERS
<S>, FOR SELECT PARAMETERS
<N>, FOR START WITH THIS TEST NUMBER

(5. CONT'D)

D, P, S, N?

IN RESPONSE TO THIS LAST QUESTION THE OPERATOR IS REQUIRED
TO TYPE ONE OF THE LETTERS IN THE STRING. AT AUTO START
TIME THE PROGRAM FIRST SETS UP ALL THE DEFAULT PARAMETERS
'DEFAULT PARAMETERS' MEANS THE SET OF OPERATING VARIABLES

SELECTED AT THE FACTORY. FOR EXAMPLE, THE DEFAULT ADDRESS IS 176200, THE DEFAULT VECTOR ADDRESS IS 300. THEREFORE, AT AUTO START TYPING 'P' FOR PREVIOUSLY SELECTED PARAMETERS IS EQUIVALENT TO TYPING 'D' FOR DEFAULT PARAMETERS.

IF ANY CHARACTER OTHER THAN ONE IN THE STRING IS TYPE THE MONITOR WILL REJECT THE CHARACTER AND RETYPE THE STRING.

IF, IN RESPONSE TO THE STRING, THE OPERATOR TYPES AN 'S' THE SELECTION SEQUENCE IS ENTERED AND THE FOLLOWING DIALOGUE TAKES PLACE.

NOTE: THESE ARE THE DEFAULT PARAMETERS. TYPING <D> IS EQUIVALENT TO TYPING THE DEFAULT PARAMETERS.

TEST NUMBER: 1
 BASE ADDRESS: 176200
 VECTOR ADDRESS: 300
 DX PRIORITY LEVEL: 4
 TYPE CU ADRS'S IN HEX <CR><LF>; <CR><CR> TERMINATES LIST
 ADRS: 10 (THIS IS IN HEX)
 DEVICES PER CU: 20 (THIS IS IN OCTAL)
 LIST ALL LEGAL COMMANDS
 COMMAND:
 SET SWITCHES

AT ANY TIME DURING THE 'SE'LECTION SEQUENCE A CONTROL C MAY BE TYPED AND THE MONITOR WILL ASK AGAIN 'D,P,S,N?'.

'TEST NUMBER:'

HERE THE MONITOR IS ASKING FOR THE NUMBER OF THE FIRST TEST IN THE CHAINING SEQUENCE. THE DEFAULT ANSWER IS '1' ONE, THE FIRST TEST IN THE CHAIN. IT MAY BE THAT THE OPERATOR IS ONLY INTERESTED IN THE LAST FEW TESTS AND THEREFORE WOULD TYPE 22 OR WHATEVER. AT THIS WRITING THERE IS NO CHECK TO SEE IF THE OPERATOR SELECTED A NONEXISTANT TEST NUMBER (E.G. PI,-2,4 MEG). SEE TABLE OF CONTENTS IN BEGINNING OF LISTING.
 -TYPING <CR> WILL DEFAULT THIS PARAMETER

(5. CONT'D)

'BASE ADDRESS: 176200'

THIS IS THE BASE ADDRESS FOR THE DX11 AND IS ALSO THE ADDRESS OF THE DXDS.
 -TYPING <CR> WILL DEFAULT THIS PARAMETER

'VECTOR ADDRESS: 300'

THE DX11 IS CUT TO INTERRUPT TO ADDRESS 300 AT THE FACTORY. ON SITE THE DX FOLLOWS, DC'S KL'S DP'S, DM'S DN'S, DMBB'S, DR11'S, DR11A, DR11B, TYPESETTING AND BUS SWITCHES.

-TYPING <CR> WILL DEFAULT THIS PARAMETER

'TYPE CU ADRS'S IN HEX <CR><LF>;<CR><CR> TERMINATES LIST

ADRS: 10 <CR><LF>
ADRS: 20 <CR><CR>

THIS REQUEST IS FOR THE CONTROL UNIT'S HEXIDECIMAL ADDRESS OR ADDRESSES. CAUTION:::::::::::
DO NOT EXCEED 16 ENTRIES OF CU ADDRESSES ..THE PROGRAM MAY SELF DESTRUCT....IF THE SYSTEM REQUIRES THAT THERE BE MORE THAN 16 CU ADDRESSES THEN THE DIAGNOSTICS MUST BE RUN AGAIN FOR THOSE EXCEEDING 16 CAUTION:::::
IN MAINTENANCE CLOCK 1 DIAGNOSTIC THE M908 MUST NOT BE CUT FOR MORE THAN 16 CU ADDRESSES

THE IBM CONTROL UNITS ADDRESSES ARE SPECIFIED IN HEXADECIMAL. FOR CONTROL UNIT 010(16) THE RESPONSE TO ADRS: IS 10(HEX) WHICH IS 00010000(2). THE DX11 CAN EMULATE UP TO 128(10) CONTROL UNITS WITH 1 DEVICE EACH OR 1 CONTROL UNIT WITH 128(10) DEVICES OR AS IS THE DEFAULT CASE 1 CONTROL UNIT WITH 16(10) DEVICES. THE ADRS: AND RESPONSE WILL CONTINUE INDEFINITELY AS LONG AS <CR><LF> IS TYPED FOLLOWING THE CU ADDRESS. THE LIST IS TERMINATED BY TYPING <CR><CR>. NOTE:::: TYPING <CR><CR> IN RESPONSE TO THE FIRST ADRS: WILL DEFAULT THE CU ADDRESS TO 00 AND WILL ALSO TERMINATE THE LIST.(DEFAULT=10 HEX). THE ACTUAL # MUST BE TYPED IN

'DEVICES PER CU: 20''

THE RESPONSE TO THIS INPUT REQUEST IS IN OCTAL AND REPRESENTS THE NUMBER OF DEVICES THIS CONTROL UNIT SERVICES. A DX11 EMULATED CONTROL UNIT CAN SERVICE FROM 1 TO 200(8) DEVICES. NOTE:::: TYPING <CR> IN RESPONSE TO DEVICES PER CU: WILL DEFAULT TO 0. THEREBY CAUSING AN ILLEGAL NUMBER OF DEVICES PER CU MESSAGE...THE ACTUAL # MUST BE TYPED IN.(DEFAULT=20 OCTAL)
THIS DIAGNOSTIC WILL REJECT <1 AND >20 DEVICES PER CU

(5. CONT'D)

A CHECK IS MADE HERE TO INSURE THAT THE OPERATOR DID NOT ASSIGN AN IMPOSSIBLE NUMBER OF DEVICES FOR EACH CONTROL UNIT.

TYPE CU ADRS'S IN HEX <CR><LF>; <CR><CR> TERMINATES LIST
ADRS: F0
DEVICES PER CU: 0
ILLEGAL NUMBER OF DEVICES PER CU
DEVICES PER CU: 4
LIST ALL LEGAL COMMANDS
COMMAND:

WHEN A '4' WAS TYPED IN RESPONSE TO DEVICES PER CU:, THE NUMBER WAS ACCEPTED AND THE MONITOR CONTINUED.

H 1

NOTICE: OFFLINE & ONLINE DIAGNOSTICS REQUIRE AT LEAST TWO CU DEVICE ADDRESSES FOR TESTING MULTIPLEXOR FUNCTIONS...THE M908 MUST ALSO BE STRAPPED FOR >1 SEQ 0007

'LIST ALL LEGAL COMMANDS'
COMMAND: 400<CR>
STATUS: 0 <CR><LF> TO CONTINUE LIST
<CR><CR> TO TERMINATE LIST

THIS FACILITY WAS BUILT INTO THE DIAGNOSTIC TO ENABLE THE OPERATOR TO BUILD HIS OWN DEVICE STATUS TABLE (DST). A <CR> IN RESPONSE TO COMMAND: ASSUMES THE DEFAULT DST. THE FIRST ENTRY MUST BE NONZERO. THEREFORE IF YOU WISH YOUR FIRST COMMAND TO BE A TIO=0 YOU MUST TYPE IT IN WITH PARITY (E.G. 400). FOLLOWING THE COMMAND THE MONITOR WILL ASK FOR THE CORRESPONDING STATUS. -TYPING <CR> WILL DEFAULT THIS PARAMETER

'SET SWITCHES'

HERE THE MONITOR ASKS FOR THE CONSOLE SWITCH SETTINGS.

SW<15> HALT ON ERROR
SW<14> SCOPE LOOP
<SW13> INHIBIT ERROR PRINTOUT
SW<12> SHORT ERROR REPORT
SW<11> INHIBIT ITERATIONS
SW<10> MAINTENANCE CLOCK CONTROL (MAINTENANCE TESTS ONLY)
SW<9> ODT TRAP ON ERROR
SW<2> MUX MODE
SW<3> BSYEN MODE

LOAD THE SWITCH REGISTER WITH THE APPROPRIATE FUNCTION AND TYPE <CR>.

5.2 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

1. HALT ON ERROR
WHEN AN ERROR HALT OCCURS
2. CLEAR SW<15>
3. SET SW<14>, SCOPE
4. TYPE <P> FOR PROCEED IF ODT WAS SELECTED
(SW9=1). OR PRESS CONTINUE ON THE CONSOLE
IF ODT WAS NOT SELECTED SW9=0
IF ERROR IS REPETITIVE.
5. SET SW<13> AND SCOPE ERROR

THE ERROR PC SHOULD BRING THE OPERATOR TO A POINT IN THE LISTING WHERE THE ERROR IS DOCUMENTED. THEN USING THE

I 1

PRINTS AND THE FLOWS THE ERROR CAN BE TRACED TO ITS SOURCE.

SEQ 0008

AT ANY TIME DURING THE INITIALIZATION OR TESTING THE OPERATOR CAN TYPE CONTROL C AND CONTROL WILL BE RETURNED TO THE MONITOR. SOME TESTS ARE 5-10 SECONDS IN DURATION SO THE RESPONSE TO THE CONTROL C WILL NOT BE INSTANTANEOUS.

THE RESTART ADDRESS IS 200. IF THIS ADDRESS IS LEFT IN THE CONSOLE SWITCH WHEN "START" IS PRESSED THE MONITOR WILL TYPE OUT D,P,S,N? IF THE SWITCHES ARE ZEROED THE TYPE WILL BE "SET SWITCHES".

THERE ARE TWO CALIBRATION TESTS (MAINT CLK1) THAT SHOULD BE RUN IN SCOPE MODE(T15 & T16). IT IS QUITE POSSIBLE THAT USING THE STANDARD OPERATIONS PROCEDURE PREVIOUSLY SUGGESTED THAT THE OPERATOR WILL FALL NATURALLY INTO THESE CALIBRATION TESTS. IF THE SYSTEM HAS BEEN BROUGHT UP ONCE BEFORE AND THE OPERATOR WISHES TO CHECK THE CALIBRATION THE FOLLOWING PROCEDURE SHOULD BE FOLLOWED:

1. EXAMINE TABLE OF CONTENTS FOR THE TEST NUMBER (N) OF CALIBRATION ROUTINES.
2. TYPE N IN RESPONSE TO D,P,S,N?
3. PUT SW<14> UP IN RESPONSE TO "SWITCH SETTINGS"
4. TYPE <CR> IF ERROR TYPE OUT OCCURS SET SW<13>.

5.2.1 MAINTENANCE CLOCK CONTROL (MAINTENANCE CLK1 & CLK2 DIAG. ONLY)

WHEN SWITCH 10 IS SELECTED AND A MAINTENANCE CLOCK PROGRAM IS BEING RUN THE EXECUTION OF THE JSR PC, CLK SUBROUTINE WILL CAUSE A BREAK POINT TRAP TO ODT AND A TYPEOUT OF THE FOLLOWING FORMAT WILL OCCUR:

```
AAAAAA B0;NNNNNN  
*
```

THIS INDICATES THAT THE PROGRAM HAS TRAPPED TO ODT AND IS AWAITING THE COMMAND TO 'P'ROCEED BEFORE EXECUTING THE NUMBER OF MAINTENANCE CLOCK PULSES SPECIFIED BY JSR PC, CLK N. UPON TYPING 'P' THE PROGRAM WILL CONTINUE FROM LOCATION AAAAAA.

THIS IS A USEFUL FEATURE IN SEVERAL RESPECTS. FIRST, IT ALLOWS THE OPERATOR TO SINGLE STEP THROUGH THE FLOWS. THE LISTING AIDS HERE ALSO IN THAT IT HIGHLIGHTS THE PHASE AND STATE. IN ADDITION TO WALKING THROUGH THE FLOWS THIS FEATURE ALSO ALLOWS THE OPERATOR TO EXAMINE DONE DISPLAYED DX REGISTERS AND KEY MEMORY LOCATIONS.

IT IS REQUIRED THAT ONLY THE FOLLOWING ODT COMMANDS BE USED

N/	OPENS WORD N
P	PROCEED FROM BREAK POINT
N:G	GOES TO WORD N AND STARTS PROGRAM
<CR>	CLOSES OPEN LOCATION (CARRIAGE RETURN)
<LF>	OPENS NEXT LOCATION (LINE FEED)
^C	CONTROL C, RETURN TO DIAGNOSTIC MONITOR

ANY OTHER COMMANDS ARE USED AT THE OPERATORS OWN RISK.
IF OTHER COMMANDS ARE USED THE DX AND THEREFORE THE DIAG-
NOSTIC MAY BEHAVE STRANGELY. PLEASE RELOAD.

A TYPICAL SEQUENCE

SET SWITCHES

005536 B0:017044	
*P	PROCEED
005640 B0:017044	
*P	PROCEED
006032 B0:017044	
*176204/000500	EXAMINE DXCS
176206 /002000	EXAMINE DXOS
176210 /00300	EXAMINE DXBA
*P	PROCEED
006504 B0:017044	
*	CONTROL C
D,P,S,N?	MONITOR MODE

6. ERRORS

TYPICALLY ERROR REPORTS TAKE THE FOLLOWING FORMAT.

```
ERROR PC: 017274
ERROR IN TEST: 17
CUADRS/MO: 000020
B0:020742
```

THIS INDICATES THAT WHILE EXECUTING TEST #17 ON ERROR STATE WAS DETECTED AND IS DOCUMENTED AT PROGRAM COUNT 017274. THE CONTRL UNIT UNDER TEST OF THE TIME OF ERROR WAS 20(8) AND THE IBM COMMAND WAS A NOP. IN SEVERAL CASES THE COMMAND IS OF NO SIGNIFICANCE.

IF SWITCH 9 IS UP THE ERROR REPORT GENERATOR WILL BREAK TO ODT AS INDICATED BY 'B0:NNNNN'. HERE AGAIN THE POWER OF ODT MAY BE USED TO COLLECT ADDITIONAL DATA CONCERNING THE FAULT.

A TYPICAL APPROACH MIGHT BE (AFTER COLLECTING DATA);
TYPE CONTROL C, RESULTS:

D,P,S,N? N
 TEST NUMBER: 17
 SET SWITCHES

IN RESPONSE TO SWITCHES SET THE FOLLOWING

SR<15>=0 HALT ON ERROR
 SR<14>=1 SCOPE
 TYPE <CR>

IF THE ERROR IS REPEATABLE SET SR<13>. INHIBIT PRINT
 AND GO AT IT.

NOTICE: A TYPE OUT OF THE FORMAT BE;NNNNNN INDICATES
 A BREAK POINT ERROR AT NNNNNN. THIS IS AN ODT ERROR
 AND CAN BE CAUSED BY 1. PLAYING GAMES WITH ODT OR
 2. AN ILLEGAL BREAK TRAP I.E. T BIT SET OR EXECUTE
 A 000003.

(6. CONT'D)

DURING MAINTENANCE CLOCK TESTS THERE EXISTS A SUBROUTINE
 CALLED CHKREG. THIS ROUTINE EXAMINES ALL THE DX11
 REGISTERS AND VERIFIES THAT THEY ARE IN THE EXPECTED STATE.
 CHKREG HAS A SPECIAL ERROR TRAP THAT RESULTS IN THE
 FOLLOWING TEXT:

ERROR PC: 017446
 ERROR IN TEST: 17
 CUADRS/MO: 000020
 ORIGIN OF MAP ERROR 017602
 REGISTER-CONTENTS-MAP

DXMI: 176777 000400 (DXMI IS UNREADABLE IGNORE THIS COMPARE)
 DXCB: 074000 000000 (PHASE AND STATE FLOPS ARE NOT TRACED)
 DXES: 000014 000010 (ERROR CONDITION IS THAT BIT2 IS SET)
 B0:020742

D,P,S,N?

IN THIS REPORT THE REGISTERS ARE NAMED (UNDER REGISTER)
 AND THERE CONTENTS DUMPED (UNDER CONTENTS) SO THAT IT MAY
 BE COMPARED WITH THE EXPECTED STATE IN THE MAP (UNDER MAP).

THERE ARE TWO ANOMALIES HERE:

1. THE DXMI IS OFTEN UNREADABLE THEREFORE IF THE DXMI
 IS ALL ONES OR ALMOST ALL ONES DISREGARD THE COMPARISON
 IT WAS NOT MADE.
2. THE PHASE AND STATES FLOPS ARE NOT COMPARED SO THAT
 CHKREG CAN BE USED IN ROUTINE WITH FREE RUNNING CLOCKS.

THIS MEANS THAT THERE MUST BE A DIFFERENCE BETWEEN

L 1
CONTENTS AND MAP IN A REGISTER OTHER THAN THE DXMI OR BITS
OTHER THAN 074000.

SEQ 0011

THE EXERCISER PROGRAMS DO TUMBLE TABLE TRACING ON INTERRUPT.
IN THE EVENT OF A TRACE ERROR THE PROGRAM WILL TYPE OUT:

TT TRACE ERROR IN TEST: N1
ORIGIN OF LAST TT UPDATE: N2
TT ENTRY WAS: 'WHATEVER'
EXPECTED ENTRY: 'WHATEVER +1'
TT POINTER N3

THEN PROCEED WITH THE NORMAL ERROR REPORT. THE ADDRESS XXXXXX
SPECIFIES THE LOCATION WHERE THE EXPECTED TT ENTRIES WERE
LAST UPDATED.

EXERCISER ERROR REPORTS ALSO INDICATE THE
DX MODE WHEN THE ERROR OCCURED: MULTIPLEXOR OR BUSYEN
(DX ONLINE AND DX OFFLINE EXERCISERS ONLY)

6.2 ERROR RECOVERY

IN THE EVENT THAT THE DX GETS STUCK IN AN UNRECOVERABLE
PHASE AND STATE WHILE MAINTENANCE CLOCK ENABLE IS SET,
DEPRESS HALT AND START. THEN LOAD ADDRESS 200 AND START.

ON BREAK POINT ERRORS RELOAD TAPE

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4.2

7.2 OPERATING RESTRICTIONS

NEVER NEVER NEVER RUN THE ONLINE-MAINTENANCE-CABLED
EXERCISER WHILE CONNECTED TO IBM

8. MISCELLANEOUS

MAINT. CLK1 DIAGNOSTIC ONLY:::::::::::
AT THE END OF THE PROGRAM IT WILL TYPE 'END TEST SET SW3=1?'.
THIS IS TO SIGNIFY THAT SW3 MUST BE SET AT LEAST ONCE DURING THE
USE OF THIS DIAGNOSTIC. IT IS NOT NECESSARY TO LEAVE SW3=1
AS IT CONSUMES TOO MUCH DIAGNOSTIC TIME. BASICALLY THIS OPENS
THE TEST THAT CHECKS THAT YOU HAVE CORRECTLY ANSWERED ALL THE
CU ADDRESSES & DEVICES/CU QUESTIONS CORRECTLY....IF YOU LIED
IT WILL CATCH IT. IE. IF YOU ANSWERED THE DEVICES PER CU

M 1

WITH 10(8) AND IN ACTUALITY THE DEVICES PER CU ARE CUT TO 4 ON THE M908, THIS TEST WILL CATCH THE ERROR.
 WHEN SW3=1 PROGRAM RUN TIME IS GREATLY INCREASED AS IT CHECKS ALL OTHER ADDRESSES FOR ADRECC & ADRECD.

DX ONLINE & DX OFFLINE EXERCISERS::::::::::
 THESE PROGRAMS WILL DEFAULT TO AUTOMATICALLY SETTING SW1=1 & SW2=1 . HOWEVER, TO EXECUTE THEIR FUNCTINS EARLY IN THE PROGRAM RUN TIME- SET THEM BEFORE STRIKING <CR> WHEN THE MONITOR ASKS "SET SWITCHES"
 NOTE: AT LEAST 2 DEVICES/CU MUST BE STRAPPED ON M908

8.1 EXECUTION TIME

THE EXECUTION TIME OF EACH PROGRAM IS VARIABLE AND IS A FUNCTION OF THE PROGRAM LENGTH AND THE CONTROL UNIT ADDRESS STRUCTURE. IN GENERAL THEY RUN 10 TO 20 MINUTES.

9. PROGRAM DESCRIPTION

CONTAINED WITHIN LISTING.

10. LISTING

FOLLOWING

11. FLOW CHARTS

SEE PRINT SET

CZDXHDO DX11B ONLINE EXER
 COPYRIGHT 1972,1978 DIGITAL EQUIPMENT CORP.
 146 MAIN ST. MAYNARD, MA. 01754
 MAINTAINER: DIAGNOSTICS
 AUTHOR: JOHN FRIEDRICH

*
 :***** MOD APR 74 *****
 :*

:* REVISED: BY W. ARMSTRONG

.SBTTL DYNAMIC SWITCH SETTINGS (SWR =1)

:* DYNAMIC SWITCH REGISTER SETTINGS

:*	SWR#	SIGNIFICANCE:
:*	SET = ONE	
:*	SWR 15	'HALT ON ERROR''
:*	SWR 14	'SCOPE LOOP''
:*	SWR 13	'INHIBIT ERROR REPORT''

N 1

SWR 12 'SHORT ERROR REPORT'
SWR 11 'INHIBIT ITERATIONS'

SWR 02 'MUX MODE'
SWR 01 'BSYEN MODE'

'USER CHANGE INFORMATION'
'DUE TO REVISION APR 74'
'PLEASE READ INFO BELOW'

NOTE:

AN OPERATOR RESPONSE OF '0' TO THE PROGRAM
'TTY' REQUEST FOR 'DEVICES PER CU:' IS NO
LONGER DEFAULTED TO 20 (16 DECIMAL). I.E.

DEVICES PER CU: 0 '=ILLEGAL ?'

THE HEADER 'CU CHANNEL ADDRES:' USED ON ERROR
OUTPUT HAS BEEN CHANGED TO 'CUADRS/MO:'. I.E.
IT SIGNIFIES EITHER THE CONTENTS OF THE 'DXMO'
REGISTER OR THE CONTROL UNIT BASE ADDRESS WHERE
MEANINGFULL.

***** MOD APR 74 *****

2	CLOCK, ISSUE N MAINTENANCE CLOCK PULSES
3	SS, SELECTION MACRO
4	SHORT, SHORT TT TRACE UPDATE AND SELECT
5	DEFINE, EMT DEFINITIONS
6	ESAVE, SAVE REGISTER FOR ERROR PRINT
7	ERSTOR, RESTOR ERROR REGISTERS
8	SAVE, SAVE ARG ON STACK
9	RESTOR, RESTOR ARG FROM STACK
10	SCOPELOOP, SUBROUTINE TO EXECUTE SCOPE CODE
11	CLEAR, CLEAR FROM ARG1, ARG2 WORDS
12	CLRSUB, SUBROUTINE TO CLEAR FROM ARG1, ARG2 WORDS
13	DUMP, OCTAL DUMP OF ARG
14	SDUMP, OCTAL DUMP OF ARG, LEADING ZEROS SUPPRESSED
15	NUMBER, TEST NUMBER INCREMENTER
16	SCOPEM, SCOPE
17	ERCALL, ERROR CALL EMT
18	STEPTSSF, SINGLE STEP TSSF
19	CHECKFOR, CHECK FOR PHASE ARG
20	CHECK, CHECK FOR PHASE.STATE ARG
21	SNAPSHOTPH, ?
22	LDNLK, LOAD AND LOOK MCLK MACRO
23	CLKCHK, CLOCK AND CHECK PHASE+STATE
24	LOAD, LOAD BIT IN REGISTER + MAP
25	REMOV, REMOVE BIT FROM REGISTER + MAP
191	MISCELLANEOUS DEFINITIONS
293	TRAP DEFINITIONS
357	DX REGISTERS
457	POWER FAIL
511	STATUS POINTER WORD TABLE
823	TUMBLE TABLE
832	T1 TEST I/O COMMAND
857	T2 ILLEGAL COMMAND
885	T3 SENSE COMMAND
937	T4 READ COMMAND (PDP OUTPUT)
1065	T5 WRITE COMMAND (PDP INPUT)
1193	T6 END OF TEST STRING
1386	SETUP SELECTED PARAMETERS
1416	STATUS PRESENTATION
1449	DATA TRANSFER ROUTINES
1507	ONLINE ROUTINE
1559	INITIAL SELECTION SEQUENCE
1752	HIO SUBROUTINE
1828	CUIS SUBROUTINE
2124	MONITOR
2393	MONITOR FILES
2616	MONITOR SUBROUTINES
3002	TTY ASCII OUTPUT ROUTINE
3037	SAVE AND RESTORE REGISTERS
3067	OCTAL DUMP ROUTINE
3394	MESSAGES
3402	DATA BUFFERS

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

:DXDS, DX DEVICE STATUS BITS

100000	PARER =100000	:ERRORS
040000	NXM = 40000	:NONEXISTANT MEMORY REFERENCE
020000	SELRST= 20000	:IBM RESETS; SELECTIVE RESET
010000	SYSRST= 10000	:SYSTEM RESET
004000	INF DSC= 4000	:INTERFACE DISCONNECT
034000	IBMRST= SELRST!SYSRST!INF DSC	
002000	UCHKS = 2000	:STATUS FLAGS
001000	CHENDS= 1000	:CHANNEL END SENT
000400	BSYS = 400	:BUSY SENT
000200	CHIS = 200	:CHANNEL INITIATED SELECTION
000100	ESEND = 100	:ENDING STATUS END
000040	CHDEND= 40	:CH DATA END
000020	CJDEND= 20	:CU DATA END
000010	ISSREJ= 10	:ISS REJECT
000004	CMDCHN= 4	:COMMAND CHAINING
000002	STKSTB= 2	:STACKED STATUS B
000001	CMDREJ= 1	:COMMAND REJECT

:DXCS,DX CONTROL UNIT STATUS BITS

100000	PARSTP=100000	:STOP ON BUSO PARITY ERROR
040000	CUFBM = 40000	:SELECT FORCED BURST
020000	ENDEN = 20000	: 'CUEND'
010000	CS12 = 10000	:NOT USED
004000	BSYEN = 4000	:ENABLE SET 'CUBSY'
002000	CS10 = 2000	:NOT USED
001000	ONLINA= 1000	:ONLINE A
000400	CUBSY = 400	:CU BUSY
000200	DONE = 200	:FUNCTION DONE
000100	INTEN = 100	:INTERRUPT
000040	STKSTA= 40	:STACKED STATUS
000030	XBA = 30	:EXTENDED BASE ADDRESS
000006	FCTN = 6	
000001	DXFRS = 1	:FCTN + GO
000003	DXFI = 3	:READ (INPUT)
000005	DXFO = 5	:WRITE (OUTPUT)
000007	DXFST = 7	:STATUS
000001	GO = 1	:BEGIN FUNCTION

:DXOS DX OFFSET (CUOR) AND STATUS (CUSR) BITS

000200	ATTEN =200	:ATTENTION
000100	STAMOD=100	:STATUS MODIFIER
000040	CUEND = 40	:CU END
000020	BSY = 20	:BUSY
000010	CHEND = 10	:CH END
000004	DEVEND= 4	:DEVICE END
000002	UCHECK= 2	:UNIT CHECK
000001	UEXCEP= 1	:UNIT EXCEPT

:DXMO DX MAINTENANCE-OUT BITS


```
57                                     ;SELECTION CONTROL INES
58      100000                        OPLO =100000      ;OPERATIONAL OUT
59      040000                        HLDO = 40000      ;HOLD OUT
60      020000                        SELO = 20000      ;SELECT OUT
61      010000                        SUPO = 10000      ;SUPPRESS OUT
62
63                                     ;TAG LINES
64      004000                        ADRO = 4000      ;ADDRESS OUT
65      002000                        CMDO = 2000      ;COMMAND OUT
66      001000                        SRVO = 1000      ;SERVICE OUT
67      000400                        PARO = 400      ;PARITY OF/FOR BUS OUT
68                                     ;DXMI DX MAINTENANCE-IN BITS
69
70                                     ;SELECTION CONTROL LINES
71      100000                        OPLI =100000     ;OPERATIONAL IN
72      040000                        SELI = 40000      ;SELECT IN
73      020000                        REQI = 20000      ;REQUEST IN
74
75                                     ;TAG LINES
76      010000                        ADRI = 10000     ;ADDRESS IN
77      004000                        STAI = 4000      ;STATUS IN
78      002000                        SRVI = 2000      ;SERVICE IN
79      001000                        CLKO = 1000      ;OK TO GO ONLINE (RO)
80      000400                        PARI = 400      ;BUSI PARITY (RO)
81
82                                     ;DXCB DX CONTROL BITS
83
84      100000                        LOCKO =100000    ;LOCK OUT
85      074000                        PHS =074000     ;PHASE - STATE BITS
86      002000                        FASTCU= 2000     ;FAST CU
87      001000                        SYNC = 1000      ;SYNCHRONIZATION
88      000400                        CUDX = 400      ;CU DATA CONTROL
89      000200                        IOD = 200      ;INPUT OUTPUT DONE
90
91                                     ;NPR CONTROLS
92      000100                        BYPAS = 100     ;BYPASS
93      000040                        NPRX = 40      ;NPR CONTROL SWITCH
94      000020                        NPRT = 20      ;NPR TRANSFER DIRECTION
95      000010                        BALF = 10      ;BUFFERED ALTERNATOR FLOP
96      000004                        ONLINB= 4      ;ON LINE TO IBM
97      000002                        ADRECC= 2      ;ADDRESS RECOGNITION (CU)
98      000001                        ADRECD= 1      ;ADDRESS RECOGNITION (DEVICE)
99
100                                    ;DXES DX EXTRA SIGNALS
101
102      000001                        MCLKP=1      ;MAINTENANCE CLOCK PULSE
103      000002                        MCLKEN=2     ;MAINT. CLK ENABLE
104      000004                        SOSIEN=4     ;SRVO-SRVI ENABLE
105      000010                        TIMDIS=10    ;TIMER(5 SEC) DISABLE
106      000020                        DXTO=20     ;DX TIMEOUT (5 SEC)
107      000040                        NPRTO=40    ;NPR TIMEOUT (8 MICROSEC)
108      000200                        INTREQ=200   ;INTERRUPT REQUEST
109
110                                    ;DXES1 DX EXTRA EXTRA SIGNALS
111
112      000001                        IRS =1      ;IBM RESET STORED
```

113 000002 DSCRSP =2 ;DISCONNECT RESPONSE
114 000004 SELIBR =4 ;SELECT IN BUFFERED

;DEFINE REGISTER MAP INDICES

118
119 000000 DS= 00
120 000002 CA= 02
121 000004 CS= 04
122 000006 OS= 06
123 000010 BA= 10
124 000012 BC= 12
125 000014 MO= 14
126 000016 MI= 16
127 000020 CB= 20
128 000022 ND= 22
129 000024 ES= 24

;PHASE CONTROL FLOPS OF DXCB

130
131
132
133 000000 PHASE0=00000
134 010000 PHASE1=10000
135 020000 PHASE2=20000
136 030000 PHASE3=30000
137 040000 PHASE4=40000
138 050000 PHASE5=50000
139 060000 PHASE6=60000
140 070000 PHASE7=70000

;TIME STATE FLOP AND STATE DEFINATION

141
142
143
144 004000 TSSF=4000
145 004000 TS1=4000
146 000000 TS2=0000

;PHASE AND STATE DEFINITIONS

147
148
149
150 004000 PHS01= PHASE0! TS1
151 000000 PHS02= PHASE0! TS2
152 014000 PHS11= PHASE1! TS1
153 010000 PHS12= PHASE1! TS2
154 024000 PHS21= PHASE2! TS1
155 020000 PHS22= PHASE2! TS2
156 034000 PHS31= PHASE3! TS1
157 030000 PHS32= PHASE3! TS2
158 044000 PHS41= PHASE4! TS1
159 040000 PHS42= PHASE4! TS2
160 054000 PHS51= PHASE5! TS1
161 050000 PHS52= PHASE5! TS2
162 064000 PHS61= PHASE6! TS1
163 060000 PHS62= PHASE6! TS2
164 074000 PHS71= PHASE7! TS1
165 070000 PHS72= PHASE7! TS2

.SBTTL MISCELLANEOUS DEFINITIONS

166
167
168

```
169      104400      SCOPE=TRAP      ;SCOPE LOOP TRAP
170
171      100000      BIT15=100000
172      040000      BIT14=40000
173      C20000      BIT13=20000
174      010000      BIT12=10000
175      004000      BIT11=4000
176      002000      BIT10=2000
177      001000      BIT9=1000
178      000400      BIT8=400
179      000200      BIT7=200
180      000100      BIT6=100
181      000040      BIT5=40
182      000020      BIT4=20
183      000010      BIT3=10
184      000004      BIT2=4
185      000002      BIT1=2
186      000001      BIT0=1
187      000000      HERE=0
188
189      ;CHANNEL COMMANDS WITH PARITY
190
191      000400      TI0C=400      ;TEST I/O
192      000001      WRITEC=001    ;WRITE
193      000002      READC=002    ;READ
194      000403      NOPC=403     ;NOP
195      000004      SENSEC=4     ;SENSE
196      000405      ILLC=405    ;ILLEGAL COMMAND
197
198      ;UTILITY FLAGS
199
200      100000      INTOK=100000
201      000002      DOFLIN=2     ;SPW BIT FOR NO DST !
202
203      ;CHANNEL STATUS
204
205      000010      CE=10        ;CH END
206      000004      DE=4        ;DEVICE END
207      000002      UC=2        ;UNIT CHECK
208      000200      ATTN=200    ;ATTENTION
209      000100      SM=100      ;STATUS MODIFIER
210      000040      CUE=40     ;CU END
211      000020      BSY=20     ;BUSY
212
213      ;SWITCH DEFINITIONS
214
215      100000      HLT SW=BIT15  ;HALT ON ERROR
216      040000      LOP SW=BIT14 ;LOOP ON ERROR
217      020000      PNT SW=BIT13 ;INHIBIT PRINT
218      010000      SESW=BIT12  ;SHORT ERROR SWITCH
219      004000      IISW=BIT11  ;INHIBIT ITERATIONS
220      002000      MCCSW=BIT10 ;MAINTENANCE CLOCK CONTROL
221
222      ;PROCESSOR PRIORITY LEVELS
223
224      000000      LEVEL0= 000
```

225 000040 LEVEL1= 040
226 000100 LEVEL2= 100
227 000140 LEVEL3= 140
228 000200 LEVEL4= 200
229 000240 LEVEL5= 240
230 000300 LEVEL6= 300
231 000340 LEVEL7= 340

;REGISTER DEFINITIONS

235 000000 R0=%0
236 000001 R1=%1
237 000002 R2=%2
238 000003 R3=%3
239 000004 R4=%4
240 000005 R5=%5
241 000005 TTY=%5
242 000006 R6=%6
243 000006 SP=%6 ;STACK POINTER
244 000007 PC=%7 ;PROGRAM COUNTER

246 000004 TYPE=IOT
247 000240 NOP=240
248 177776 PS=177776 ;PROCESSOR STATUS
249 177570 SWR=177570
250 177570 SR=177570 ;SWITCH REGISTER

252 000000 E=0
253 012024 EMTABLE=EMTAG

;EMT DEFINITIONS

257 001004 104000 ERROR ;TRAPS TO T.ERROR
258 001006 104001 MAPERR ;TRAPS TO T.MAPERR
259 001010 104002 TRACER ;TRAPS TO T.TRACER
260 001012 104003 SAVRG ;TRAPS TO T.SAVRG
261 001014 104004 RSTRG ;TRAPS TO T.RSTRG
262 001016 104005 ACCEPTO ;TRAPS TO T.ACCEPTO
263 001020 104006 KEY.TO.R0 ;TRAPS TO T.KEY.TO.R0
264 001022 104007 PARITY ;TRAPS TO T.PARITY
265 001024 104010 PCH1 ;TRAPS TO T.PCH1
266 001026 104011 PCH2 ;TRAPS TO T.PCH2
267 001030 104012 PCH3 ;TRAPS TO T.PCH3

.SBTTL TRAP DEFINITIONS

;TRAP INITIALIZATION

273 000014 000014 000340 .=14 ;BREAK TRAP
274 000014 006674 000340 O.BRK,LEVEL7
276 000020 000020 000340 .=20 ;TTY OUTPUT TRAP,LEVEL 7
277 000020 016614 000340 .IOT,LEVEL7
279 000024 000024 000340 .=24 ;POWER FAIL TRAP
280 000024 001444 000340 PFAIL,LEVEL7

```
281  
282      000030 000030      . = 30  
283      000030 011762 000340      EMTDECODER,LEVEL7      ;EMT DECODER TRAP,LEVEL 7  
284  
285      000034 000034      . = 34  
286      000034 011530 000340      SCOPEC,LEVEL7      ;SCOPE TRAP  
287      000046 000046      . = 46  
288      000046 014020      LOGICAL      ;ACT11  
289  
290      000200      . = 200  
291  
292      000200 000137 001100      START: JMP @#BEGIN      ;GO TO BEGINNING OF PROGRAM  
293  
294      001100      . = 1100  
295  
296      001100 012706 001100      BEGIN: MOV #BEGIN,SP      ;SET UP STACK POINTER  
297      001104 012737 000340 177776      MOV #LEVEL7,PS      ;PRIORITY LEVEL 7  
298  
299      ;***** MOD APR 74 *****  
300      ;* 11/40.11/45 TRACE TRAP  
301  
302      001112 012737 000002 006706      MOV #2,RTX  
303      001120 012737 001154 000010      MOV #INITB,@#10  
304      001126 012737 000340 000012      MOV #340,@#12  
305      001134 005046      CLR -(SP)  
306      001136 012746 001144      MOV #INITZ,-(SP)  
307      001142 000006      RTT  
308      001144 012737 000006 006706      INITZ: MOV #6,RTX  
309      001152 000402      BR INITC  
310      001154 062706 000010      INITB: ADD #10,SP  
311      001160 013737 006706 006704      INITC: MOV RTX,YESRTI  
312      001166 012737 000012 000010      MOV #12,@#10  
313      001174 005037 000012      CLR @#12  
314  
315      ;***** MOD APR 74 *****  
316  
317      001200 005737 000042      TST @#42      ;ACT11  
318      001204 001404      BEQ BGN0      ;BR IF NO  
319      ; JSR PC,MONDFLT      ;INSERT DEFAULT PARAMETERS  
320      001206 005037 014206      CLR @#ONESHOT      ;DO NOT EXECUTE TIME CONSUMING TESTS  
321      001212 000137 013576      JMP @#MON11      ;  
322      001216 005327 000001      BGN0: DEC #1  
323      001222 001002      BNE BGN1  
324      001224 000137 012654      JMP @#MONITOR  
325      001230 032737 000200 177570      BGN1: BIT #200,SR      ;TEST FOR FAST START  
326      001236 001402      BEQ BGN2      ;BRANCH IF FAST START  
327      001240 000137 012724      JMP @#MON1.0  
328      001244 012706 001100      BGN2: MOV #BEGIN,SP  
329      001250 012737 000340 177776      MOV #LEVEL7,PS  
330      001256 000137 013544      JMP @#MON10      ;USE PREVIOUS PARAMETERS  
331  
332  
333      .SBTTL DX REGISTERS  
334  
335      001262 176200      DXBASE: 176200  
336      001264 000300      DXIV: 300      ;DX INTERRUPT VECTOR ADRS
```

337	001266	000302	DXIS:	302	:DX INTERRUPT STATUS
338	001270	000200	DXPRT:	LEVEL4	:INT PRIORITY ADRS
339	001272	000140	LESS1:	LEVEL3	:DX PRIORITY MINUS ONE
340	001274	176200	DXDS:	176200	:DEVICE STATUS ->TT
341	001276	176202	DXCA:	176202	:COMMAND AND ADDRESS ->TT
342	001300	176204	DXCS:	176204	:CONTROL UNIT STATUS
343	001302	176206	DXOS:	176206	:OFFSET AND STATUS
344	001304	176210	DXBA:	176210	:BUS ADDRESS FOR NPR'S
345	001306	176212	DXBC:	176212	:BYTE COUNT
346	001310	176214	DXMO:	176214	:MAINTENANCE OUT
347	001312	176216	DXMI:	176216	:MAINTENANCE IN
348	001314	176220	DXCB:	176220	:CONTROL BITS
349	001316	176222	DXND:	176222	:NPR DATA
350	001320	176224	DXES:	176224	:EXTRA SIGNALS
351	001322	176226	DXMOB:	176226	:MAINTENANCE OUT BUFFERED
352	001324	176230	DXES1:	176230	:EXTRA EXTRA SIGNALS
353					
354			:BYTE REGISTERS		
355					
356			:DXCA		
357					
358	001326	176202	CUAR:	176202	:CU ADDRESS REGISTER
359	001330	176203	CUCR:	176203	:CU COMMAND REGISTER
360					
361			:DXOS		
362					
363	001332	176206	CUSR:	176206	:CU STATUS REGISTER
364	001334	176207	CUOR:	176207	:CU OFFSET REGISTER
365					
366			:DXMO		
367					
368	001336	176214	BUSO:	176214	:IBM BUS OUT
369	001340	176215	CONO:	176215	:CONTROL LINES OUT
370					
371			:DXMI		
372					
373	001342	176216	BUSI:	176216	:IBM BUS IN
374	001344	176217	CONI:	176217	:CONTROL LINES IN
375					
376			:DXES		
377	001346	176224	MISC:	176224	:MISCELLANEOUS BITS
378	001350	176225	TTNDX:	176225	:TUMBLE TABLE INDEX REG
379					
380			:DXMOB		
381					
382	001352	176226	BUSOB:	176226	:BUS OUT BUFFERED
383	001354	176227	CONOB:	176227	:CONTROL OUT BUFFERED
384					
385			:REGISTER ADDRESSES		
386					
387	001356	177700	REG0:	177700	
388	001360	177701	REG1:	177701	
389	001362	177702	REG2:	177702	
390	001364	177703	REG3:	177703	
391	001366	177704	REG4:	177704	
392	001370	177705	REG5:	177705	

393 001372 177706
394 001374 177707
395
396
397
398 001376 177560
399 001400 177562
400 001402 177564
401 001404 177566
402
403
404

REG6: 177706
REG7: 177707
:TTY ADDRESSES
TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566

405 001406
406 001406 000000
407 001410 000000
408 001412 000000
409 001414 000000
410 001416 000000
411 001420 000000
412 001422 000000
413 001424 000000
414 001426 000000
415 001430 000000
416 001432 000000
417 001434 000000
418
419
420
421
422 001436 002000
423
424
425
426 001440 003000
427
428
429
430 001442 017400
431
432
433
434
435
436
437
438
439

:REGISTER TRACE TABLE

REGTT: :REGISTER TRACE TABLE
TDXDS: 0 :DEVICE STATUS TRACE
TDXCA: 0 :COMMAND AND ADDRESS TRACE
TDXCS: 0 :CU STATUS TRACE
TDXOS: 0 :OFFSET AND STATUS TRACE
TDXBA: 0 :BUS ADDRESS TRACE
TDXMO: 0 :MAINTENANCE-OUT TRACE
TDXMI: 0 :MAINTENANCE-IN TRACE
TDXCB: 0 :CONTROL BIT TRACE
TDXND: 0 :NPR DATA TRACE
TDXES: 0 :EXTRA SIGNAL TRACE
TDXES1: 0 :EXTRA SIGNAL TRACE 1
TTNDX: 0 :TTNDX TRACE

:STATUS POINTER WORD ADDRESS

SPW: 2000

:TUMBLE TABLE ADDRESS

TT: 3000

:DEVICE STATUS TABLE ADDRESS

DST: DSTADRS ;DST MUST BE MOD(400)

.SBTTL POWER FAIL

:POWER FAIL ROUTINE
:IF SELECTED VERIFY STATUS-IN IS UP
:AND CE AND DE ARE PRESENTED AS STATUS

440 001444 104003
441 001446 010637 001606
442 001452 012737 001530 000024
443 001460 032777 020000 177622
444 001466 001417
445 001470 032777 100000 177614
446 001476 001001
447 001500 104000
448 001502 032777 004000 177602

PFAIL: SAVRG
MOV R6,SAVR6
MOV #PWRUP,24
BIT #SELO,@DXMO
BEQ 1\$
BIT #OPLI,@DXMI
BNE .+4 ;BRANCH IF NO ERROR CONDITION
ERROR
BIT #STAI,@DXMI

```

449 001510 001001          BNE      .+4          ;BRANCH IF NO ERROR CONDITION
450 001512 104000          ERROR
451 001514 122777 000014 177620  CMPB    #CE!DE,@BUSI
452 001522 001401          BEQ     .+4          ;BRANCH IF NO ERROR CONDITION
453 001524 104000          ERROR
454 001526 000000          1$:     HALT
455
456          ;POWER UP ROUTINE
457
458 001530 000240          PWRUP:  NOP          ;PATCH ANYONE?
459 001532 013706 001606          MOV     SAVR6,R6
460 001536 104004          RSTRG
461 001540 012737 001444 000024  MOV     #PFAIL,24      ;RESTORE POWER FAIL VECTOR
462 001546 013777 001436 177526  MOV     SPW,@DXOS      ;RESTORE OFFSET REG
463 001554 004737 015614          JSR    PC,RESRES      ;RESET AND RESTORE
464 001560 005027          CLR    (PC)+          ;STALL FOR MECHANICS
465 001562 000000          0
466 001564 005337 001562          DEC     .-2
467 001570 001375          BNE     .-4
468 001572 000004 001610          TYPE   .PFLD          ;POWER FAILED
469
470
471
472
473
474
475 001576 012637 177776          MOV     (SP)+,PS
476 001602 000177 010052          JMP     @RETURN
477 001606 000000          SAVR6: 0
478 001610 050137 053517 051105  PFLD:  .ASCIZ  "_POWER FAILED_"
479 001616 043040 044501 042514
480 001624 057504 000          .EVEN
481 001630
482
483
484
485
486
487          .SBTTL STATUS POINTER WORD TABLE
488          001630          ENDSTR=.          ;DEFINE END OF START CODE
489          002000          .=2000
490          ;DEFAULT STATUS POINTER WORD (SPW)
491          ;DEFAULT EMULATION IS OF ONE CONTROL UNIT
492          ;WITH CAPACITY OF 16 DEVICES
493          000000          N=0
494
495          ;STATUS POINTER WORDS FOR CU 0
496
497 002000 016400          ERRDST          ;DEVICE STATUS TABLE IS AT ERRDST
498 002002 016400          ERRDST          ;DEVICE STATUS TABLE IS AT ERRDST
499 002004 016400          ERRDST          ;DEVICE STATUS TABLE IS AT ERRDST
500 002006 016400          ERRDST          ;DEVICE STATUS TABLE IS AT ERRDST
501 002010 016400          ERRDST          ;DEVICE STATUS TABLE IS AT ERRDST
502 002012 016400          ERRDST          ;DEVICE STATUS TABLE IS AT ERRDST
503 002014 016400          ERRDST          ;DEVICE STATUS TABLE IS AT ERRDST
504 002016 016400          ERRDST          ;DEVICE STATUS TABLE IS AT ERRDST
    
```


505	002020	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
506	002022	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
507	002024	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
508	002026	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
509	002030	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
510	002032	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
511	002034	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
512	002036	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

513
514
515

:STATUS POINTER WORDS FOR CU 1

516	002040	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
517	002042	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
518	002044	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
519	002046	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
520	002050	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
521	002052	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
522	002054	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
523	002056	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
524	002060	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
525	002062	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
526	002064	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
527	002066	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
528	002070	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
529	002072	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
530	002074	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
531	002076	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

532
533
534

:STATUS POINTER WORDS FOR CU 2

535	002100	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
536	002102	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
537	002104	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
538	002106	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
539	002110	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
540	002112	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
541	002114	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
542	002116	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
543	002120	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
544	002122	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
545	002124	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
546	002126	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
547	002130	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
548	002132	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
549	002134	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
550	002136	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

551
552
553

:STATUS POINTER WORDS FOR CU 3

554	002140	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
555	002142	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
556	002144	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
557	002146	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
558	002150	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
559	002152	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
560	002154	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

561	002156	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
562	002160	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
563	002162	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
564	002164	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
565	002166	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
566	002170	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
567	002172	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
568	002174	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
569	002176	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 4

573	002200	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
574	002202	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
575	002204	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
576	002206	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
577	002210	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
578	002212	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
579	002214	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
580	002216	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
581	002220	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
582	002222	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
583	002224	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
584	002226	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
585	002230	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
586	002232	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
587	002234	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
588	002236	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 5

592	002240	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
593	002242	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
594	002244	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
595	002246	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
596	002250	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
597	002252	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
598	002254	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
599	002256	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
600	002260	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
601	002262	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
602	002264	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
603	002266	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
604	002270	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
605	002272	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
606	002274	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
607	002276	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 6

611	002300	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
612	002302	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
613	002304	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
614	002306	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
615	002310	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
616	002312	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

617	002314	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
618	002316	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
619	002320	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
620	002322	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
621	002324	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
622	002326	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
623	002330	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
624	002332	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
625	002334	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
626	002336	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 7

630	002340	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
631	002342	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
632	002344	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
633	002346	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
634	002350	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
635	002352	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
636	002354	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
637	002356	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
638	002360	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
639	002362	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
640	002364	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
641	002366	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
642	002370	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
643	002372	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
644	002374	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
645	002376	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 10

649	002400	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
650	002402	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
651	002404	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
652	002406	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
653	002410	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
654	002412	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
655	002414	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
656	002416	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
657	002420	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
658	002422	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
659	002424	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
660	002426	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
661	002430	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
662	002432	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
663	002434	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
664	002436	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 11

668	002440	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
669	002442	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
670	002444	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
671	002446	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
672	002450	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

673	002452	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
674	002454	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
675	002456	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
676	002460	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
677	002462	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
678	002464	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
679	002466	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
680	002470	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
681	002472	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
682	002474	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
683	002476	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 12

684				
685				
686				
687	002500	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
688	002502	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
689	002504	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
690	002506	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
691	002510	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
692	002512	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
693	002514	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
694	002516	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
695	002520	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
696	002522	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
697	002524	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
698	002526	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
699	002530	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
700	002532	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
701	002534	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
702	002536	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 13

703				
704				
705				
706	002540	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
707	002542	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
708	002544	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
709	002546	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
710	002550	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
711	002552	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
712	002554	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
713	002556	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
714	002560	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
715	002562	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
716	002564	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
717	002566	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
718	002570	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
719	002572	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
720	002574	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
721	002576	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 14

722				
723				
724				
725	002600	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
726	002602	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
727	002604	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
728	002606	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

729	002610	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
730	002612	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
731	002614	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
732	002616	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
733	002620	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
734	002622	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
735	002624	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
736	002626	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
737	002630	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
738	002632	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
739	002634	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
740	002636	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 15

744	002640	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
745	002642	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
746	002644	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
747	002646	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
748	002650	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
749	002652	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
750	002654	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
751	002656	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
752	002660	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
753	002662	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
754	002664	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
755	002666	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
756	002670	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
757	002672	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
758	002674	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
759	002676	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 16

763	002700	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
764	002702	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
765	002704	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
766	002706	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
767	002710	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
768	002712	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
769	002714	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
770	002716	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
771	002720	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
772	002722	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
773	002724	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
774	002726	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
775	002730	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
776	002732	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
777	002734	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
778	002736	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

:STATUS POINTER WORDS FOR CU 17

782	002740	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
783	002742	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
784	002744	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST

785	002746	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
786	002750	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
787	002752	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
788	002754	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
789	002756	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
790	002760	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
791	002762	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
792	002764	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
793	002766	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
794	002770	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
795	002772	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
796	002774	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
797	002776	016400	ERRDST	:DEVICE STATUS TABLE IS AT ERRDST
798				
799			.SBTTL TUMBLE TABLE	
800		003000	.=. ;START OF TUMBLE TABLE	
801				
802	003000	000400	.BLKW 256. ;RESERVE 265. WORDS FOR TT	
803				
804		004000	ENDTT=.	
805				
806				
807				

```
808 : *****  
809 : TEST 1 TEST I/O COMMAND  
810 : *****  
811 004000 104400 TST1: SCOPE  
812 004002 012737 000200 011654 MOV #200,@#ICOUNT ;ITERATION COUNT  
813 004010 012737 000001 012636 MOV #1,@#ERTSTN ;SAVE TEST # FOR ERROR REPORT  
814 004016 012737 004024 011660 MOV #SCP1,@#RETURN ;SCOPE LOOP RETURN ADRS  
815 004024 SCP1:  
816  
817 .REM *  
818 THE FUNCTION OF THIS TEST IS TO VERIFY THE DX11 CAN RESPOND TO A  
819 TEST I/O COMMAND.  
820  
821 *  
822  
823  
824 004024 012737 000400 014216 MOV #TIOC,@#CMD ;LOAD COMMAND  
825 004032 004737 006542 JSR PC,@#TRAIT ;TT TRACE TRACE INIT  
826 004036 012737 000200 006366 MOV #CHIS,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1  
827 004044 013737 001272 177776 MOV @#LESS1,PS ;LOWER PROCESSOR STATUS  
828 004052 012737 004052 012106 MOV #.,@#TERPC ;ORIGIN OF TRAP ERROR  
829 004060 004737 007446 JSR PC,ISS.SUB ;SELECT  
830  
831
```

```

832 : *****
833 : TEST 2 ILLEGAL COMMAND
834 : *****
835 004064 104400 TST2: SCOPE
836 004066 012737 000200 011654 MOV #200,@#ICOUNT ;ITERATION COUNT
837 004074 012737 000002 012636 MOV #2,@#VERTSTN ;SAVE TEST # FOR ERROR REPORT
838 004102 012737 004110 011660 MOV #SCP2,@#RETURN ;SCOPE LOOP RETURN ADRS
839 004110 SCP2:
840
841 .REM *
842 THE FUNCTION OF THIS TEST IS TO VERIFY THE DX11 RESPONDS CORRECTLY
843 TO AN ILLEGAL COMMAND (I.E. PRESENTS UNIT CHECK).
844 *
845 004110 042737 000400 010222 BIC #400,ISSCRJ ;ILLC PRESENTS UNIT CHECK STATUS
846 ;THIS CAUSES CMDREJ
847 ;THEREFORE CHANGE ISS DIAGNOSTIC
848 ;TO LOOK FOR CMDREJ UP
849 004116 012737 000405 014216 MOV #ILLC,@#CMD ;LOAD COMMAND
850 004124 004737 006542 JSR PC,@#TRAIT ;TT TRACE TRACE INIT
851 004130 012737 002201 006366 MOV #UCHKS!CHIS!CMDREJ,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1
852 004136 013737 001272 177776 MOV @#LESS1,PS ;LOWER PROCESSOR STATUS
853 004144 012737 004144 012106 MOV #.,@#TERPC ;ORIGIN OF TRAP ERROR
854 004152 004737 007446 JSR PC,ISS.SUB ;SELECT
855
856 004156 052737 000400 010222 BIS #400,ISSCRJ ;RESTORE ISS TO LOOK FOR NOT CMDREJ
857
858
  
```



```
859 : *****
860 : TEST 3 SENSE COMMAND
861 : *****
862 004164 104400 TST3: SCOPE
863 004166 012737 000200 011654 MOV #200,@#ICOUNT ;ITERATION COUNT
864 004174 012737 000003 012636 MOV #3,@#ERTSTN ;SAVE TEST # FOR ERROR REPORT
865 004202 012737 004210 011660 MOV #SCP3,@#RETURN ;SCOPE LOOP RETURN ADRS
866 004210 SCP3:
867
868
869
870 004210 012737 000004 014216 MOV #SENSEC,@#CMD ;LOAD COMMAND
871 004216 004737 006542 JSR PC,@#TRAIT ;TT TRACE TRACE INIT
872 004222 012737 000200 006366 MOV #CHIS,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1
873 004230 013737 001272 177776 MOV @#LESS1,PS ;LOWER PROCESSOR STATUS
874 004236 012737 004236 012106 MOV #.,@#TERPC ;ORIGIN OF TRAP ERROR
875 004244 004737 007446 JSR PC,ISS.SUB ;SELECT
876
877
878 004250 012737 000001 014252 MOV #1,COUNT ;INIT SOFTWARE BYTE COUNTER
879 004256 012777 177777 175022 MOV #-1,@DXBC ;SET UP DX BYTE COUNT
880 004264 012777 014240 175012 MOV #SSTAT,@DXBA ;DX BASE ADDRESS
881 004272 052777 000005 175000 BIS #DXFO!GO,@DXCS ;FUNCTION OUTPUT & GO
882 ;SERVICE-IN SHOULD RISE AND
883 ;(SSTAT) SHOULD BE ON BUS-IN
884 004300 012737 000200 006366 MOV #CHIS,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1
885 004306 113737 014214 006426 MOVB @#DEV,@#ENTRY2 ;SECOND TT ENTRY = DXCA
886 004314 113737 014216 006427 MOVB @#CMD,@#ENTRY2+1 ;
887 004322 113777 014214 174776 MOVB @#DEV,@CUAR ;LOAD DEV ADRS
888 004330 113777 014216 174772 MOVB @#CMD,@CUCR ;LOAD COMMAND IN CUCR
889 004336 012737 004336 012106 MOV #.,@#TERPC ;DEFINE ORIGIN OF TRACE ERROR
890 004344 004737 011020 JSR PC,CUIS.SUB ;EXECUTE SELECTION
891
892 004350 012737 000020 006366 MOV #CUDEND,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1
893 004356 113737 014214 006426 MOVB @#DEV,@#ENTRY2 ;SECOND TT ENTRY = DXCA
894 004364 112737 000001 006427 MOVB #1,@#ENTRY2+1 ;CUCR OF DXCA TT ENTRY
895 004372 113777 014214 174726 MOVB @#DEV,@CUAR ;LOAD DEV ADRS
896 004400 012737 104400 012106 MOV #.,@#TERPC ;DEFINE ORIGIN OF TRACE ERROR
897 004406 004737 007142 JSR PC,TRANSFER ;DO TRANSFER IF REQUIRED
898 004412 012737 001100 006366 MOV #CHENDS!ESND,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1
899 004420 113737 014214 006426 MOVB @#DEV,@#ENTRY2 ;SECOND TT ENTRY = DXCA
900 004426 032737 000004 013602 BIT #BIT2,@#PARA ;TEST FOR MUX
901 004434 001404 BEQ 1$ ;BR IF SELECTOR
902 004436 112737 000000 006427 MOVB #0,@#ENTRY2+1 ;MUX TT CUCR
903 004444 000403 BR 2$ ;GO
904 004446 112737 000001 006427 1$: MOVB #1,@#ENTRY2+1 ;CUCR OF DXCA TT ENTRY
905 004454 113777 014214 174644 2$: MOVB @#DEV,@CUAR ;LOAD DEV ADRS
906 004462 012737 004462 012106 MOV #.,@#TERPC ;DEFINE ORIGIN OF TRACE ERROR
907 004470 004737 007064 JSR PC,STATUSPRESENTATION ;PRESENT STATUS
908
909
```

```
910 : *****  
911 : TEST 4 READ CMMAND (PDP OUTPUT)  
912 : *****  
913 004474 104400 TST4: SCOPE  
914 004476 012737 000100 011654 MOV #100,@#ICOUNT ;ITERATION COUNT  
915 004504 012737 000004 012636 MOV #4,@#ERTSTN ;SAVE TEST # FOR ERROR REPORT  
916 004512 012737 004520 011660 MOV #SCP4,@#RETURN ;SCOPE LOOP RETURN ADRS  
917 004520 SCP4:  
918  
919  
920 .REM *  
921  
922 THE FUNCTION OF THIS TEST IS TO VERIFY THAT THE DX11 CAN PROPERLY  
923 EXECUTE A READ COMMAND AND SUPPRESS DATA DURING THIS TRANSFER.  
924 THIS IS ACCOMPLISHED BY EXECUTING A CHANNEL INITIATED SELECTION  
925 FOLLOWED BY A CONTROL UNIT INITIATED SELECTION IF THE DX11'S SIM-  
926 ULATOR IS MIMICING A MULTIPLEXOR CHANNEL. FOLLOWING THE COMPLETION  
927 OF A SUCCESSFUL SELECTION THE DX'S BYTE COUNT IS SET TO 256. AND  
928 ITS BUS ADDRESS REGISTER IS LOADED TO POINT TO 'DATA' A FILE OF  
929 128. WORD OF FLOATING 1'S AND FLOATING 0'S. THIS INITIALIZATION  
930 IS FOLLOWED BY SETTING THE DX FUNCTION BITS TO A FUNCTION-OUTPUT  
931 AND GO.  
932 DURING THE ENTIRE TEST ALL TUMBLE TABLE ENTRIES ARE TRACED. PRIOR  
933 TO EACH ANTICIPATED 'DONE' THE EXPECTED TUMBLE TABLE ENTRIES (DXDS  
934 AND DXCA) ARE LOADED INTO ENTRY1 AND ENTRY2 RESPECTIVELY THEN UPON  
935 INTERRUPT THESE EXPECTED ENTRIES ARE COMPARED WITH THE ACTUAL TT  
936 ENTRIES ALONG WITH THE EXPECTED TTNDX CONTENTS. IF A DISCREPANCY  
937 OCCURS IT IS NOTED BY AN ERROR PC AND A TRACE ERROR ORIGIN PC.  
938 THE ORIGIN OF THE TRACE ERROR IS CONTAINED IN TERPC WHICH IS UP-  
939 DATED PRIOR TO EACH ANTICIPATED 'DONE' BY A MOV #.,@#TERPC.  
940  
941 THE TRANSFER OF DATA FROM THE DX11 TO THE CHANNEL SIMULATOR IS  
942 CONTROLLED BY THE SRV0-SRV1 SEQUENCE. DURING THIS SEQUENCE DATA  
943 CHECKS ARE MADE AND SUPPRESS DATA IS CHECKED BY RAISING SUPO IN  
944 RESPONSE TO SRV1 AND THEN RATTLING SRV0. NO DATE TRANSFER SHOULD  
945 TAKE PLACE IF SUPO IS RAISED AFTER THE FIRST BYTE.  
946  
947 *  
948 004520 012737 000002 014216 MOV #READC,@#CMD ;LOAD COMMAND  
949 004526 004737 006542 JSR PC,@#TRAIT ;TT TRACE TRACE INIT  
950 004532 012737 000200 006366 MOV #CHIS,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1  
951 004540 013737 001272 177776 MOV @#LESS1,PS ;LOWER PROCESSOR STATUS  
952 004546 012737 004546 012106 MOV #.,@#TERPC ;ORIGIN OF TRAP ERROR  
953 004554 004737 007446 JSR PC,ISS.SUB ;SELECT  
954  
955  
956 004560 012737 000400 014252 MOV #256.,COUNT ;SOFTWARE BYTE COUNT  
957 004566 012777 177400 174512 MOV #-256.,@DXBC ;DX BYTE COUNT  
958 004574 012701 022042 MOV #DATA,R1 ;BASE ADRS OF TEST DATA  
959 004600 012777 022042 174476 MOV #DATA,@DXBA ;FLOAT 1'S, 0'S OUTPUT DATA  
960 004606 052777 000005 174464 BIS #DXFO,@DXCS ;FUNCTION OUTPUT AND GO  
961 ;SERVICE-IN SHOULD RAISE AND  
962 ;DATA SHOULD BE ON BUS-IN  
963 004614 012737 000200 006366 MOV #CHIS,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1  
964 004622 113737 014214 006426 MOV @#DEV,@#ENTRY2 ;SECOND TT ENTRY = DXCA  
965 004630 113737 014216 006427 MOV @#CMD,@#ENTRY2+1 ;
```

```
966 004636 113777 014214 174462      MOVB    @#DEV,@CUAR      ;LOAD DEV ADRS
967 004644 113777 014216 174456      MOVB    @#CMD,@CUCR      ;LOAD COMMAND IN CUCR
968 004652 012737 004652 012106      MOV     #,@#TERPC        ;DEFINE ORIGIN OF TRACE ERROR
969 004660 004737 011020                JSR     PC,CUIS.SUB       ;EXECUTE SELECTION
970
971 004664 112737 000001 006427      MOVB    #1,@#ENTRY2+1
972 004672 012737 000020 006366      MOV     #CUDEND,@#ENTRY1 ;LOAD TT ENTRY
973 004700 012737 004700 012106      MOV     #,@#TERPC        ;READ DONE TT TRACE ERROR PC
974 004706 032777 002000 174376      RC0:   BIT     #SRVI,@DXMI ;WAIT FOR SERVICE-IN
975 004714 001001                BNE     .+4              ;BRANCH IF NO ERROR CONDITION
976 004716 104000                ERROR   ;SRVI NOT SET
977 004720 127711 174416      CMPB    @BUSI,@R1        ;CHECK DATA TRANSMITTED TO 360
978 004724 001401                BEQ     .+4              ;BRANCH IF NO ERROR CONDITION
979 004726 104000                ERROR   ;MEMORY TO BUSI TRANSFER ERROR
980 004730 105721                TSTB   (R1)+            ;INC TO NEXT BYTE
981 004732 052777 001000 174350      BIS     #SRVO,@DXMO
982 004740 032777 001000 174342      BIT     #SRVO,@DXMO
983 004746 001001                BNE     .+4              ;BRANCH IF NO ERROR CONDITION
984 004750 104000                ERROR   ;SRVO NOT SET
985 004752 032737 000004 013602      BIT     #BIT2,PARA      ;TEST FOR MUX MODE
986 004760 001005                BNE     RC1
987
988                ;WHEN ON MUX CH OPLI WILL DROP WHEN TRANSMISSION IS FINISHED
989                ;THIS WILL CAUSE DXMI TO BE UNREADABLE
990 004762 032777 002000 174322      BIT     #SRVI,@DXMI      ;SERVICE-IN SHOULD DROP
991 004770 001401                BEQ     .+4              ;BRANCH IF NO ERROR CONDITION
992 004772 104000                ERROR   ;SRVI STUCK HIGH
993 004774 052777 010000 174306      RC1:   BIS     #SUPO,@DXMO ;SET SUPPRESS-OUT
994 005002 032777 010000 174300      BIT     #SUPO,@DXMO      ;VERIFY SUPO SET
995 005010 001001                BNE     .+4              ;BRANCH IF NO ERROR CONDITION
996 005012 104000                ERROR   ;SUPO NOT SET
997 005014 012727 000010                MOV     #10,(PC)+        ;SRVO COUNT
998 005020 000000                RC2:   0                ;SRVO COUNT
999 005022 052777 001000 174260      RC3:   BIS     #SRVO,@DXMO ;KEEP SRVO CHANGING
1000 005030 032777 001000 174252      BIT     #SRVO,@DXMO      ;VERIFY SRVO SET
1001 005036 001001                BNE     .+4              ;BRANCH IF NO ERROR CONDITION
1002 005040 104000                ERROR   ;SRVO DID NOT SET
1003 005042 042777 001000 174240      BIC     #SRVO,@DXMO      ;
1004 005050 032777 001000 174232      BIT     #SRVO,@DXMO      ;VERIFY SRVO CLEAR
1005 005056 001401                BEQ     .+4              ;BRANCH IF NO ERROR CONDITION
1006 005060 104000                ERROR   ;SRVO NOT 0
1007 005062 032737 000004 013602      BIT     #BIT2,PARA      ;TEST FOR MUX MODE
1008 005070 001005                BNE     RC4
1009                ;WHEN ON MUX CH OPLI WILL DROP WHEN TRANSMISSION IS FINISHED
1010                ;THIS WILL CAUSE DXMI TO BE UNREADABLE
1011 005072 032777 002000 174212      BIT     #SRVI,@DXMI      ;SUPO SHOULD SUPPRESS SRVI SETTING
1012 005100 001401                BEQ     .+4              ;BRANCH IF NO ERROR CONDITION
1013 005102 104000                ERROR   ;SRVI WAS NOT SUPPRESSED BY SUPO
1014 005104 005337 005020                RC4:   DEC     RC2        ;DEC SRVO COUNT
1015 005110 001344                BNE     RC3              ;BRANCH IF SRVO NOT DONE
1016 005112 042777 010000 174170      BIC     #SUPO,@DXMO      ;DROP SUPPRESS-OUT
1017 005120 032777 010000 174162      BIT     #SUPO,@DXMO      ;VERIFY SUPO CLEAR
1018 005126 001401                BEQ     .+4              ;BRANCH IF NO ERROR CONDITION
1019 005130 104000                ERROR   ;SUPO STUCK HIGH
1020 005132 032777 001000 174150      BIT     #SRVO,@DXMO
1021 005140 001401                BEQ     .+4              ;BRANCH IF NO ERROR CONDITION
1022 005142 104000                ERROR   ;SRVO STUCK HIGH
```

Address	Op Code	Operand 1	Operand 2	Operand 3	Operand 4	Comment
1022	005144	005337	014252			DEC COUNT
1023	005150	001256				BNE RCO
1024	005152	012737	001100	006366		MOV #CHENDS!ESND,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1
1025	005160	113737	014214	006426		MOVB @#DEV,@#ENTRY2 ;SECOND TT ENTRY = DXCA
1026	005166	032737	000004	013602		BIT #BIT2,@#PARA ;TEST FOR MUX
1027	005174	001404				BEQ 1\$;BR IF SELECTOR
1028	005176	112737	000000	006427		MOVB #0,@#ENTRY2+1 ;MUX TT CUCR
1029	005204	000403				BR 2\$;GO
1030	005206	112737	000001	006427	1\$:	MOVB #1,@#ENTRY2+1 ;CUCR OF DXCA TT ENTRY
1031	005214	113777	014214	174104	2\$:	MOVB @#DEV,@CUAR ;LOAD DEV ADRS
1032	005222	012737	005222	012106		MOV #,@#TERPC ;DEFINE ORIGIN OF TRACE ERROR
1033	005230	004737	007064			JSR PC,STATUSPRESENTATION ;PRESENT STATUS
1034						
1035						
1036						

```
1037 : *****
1038 : TEST 5 WRITE COMMAND (PDP INPUT)
1039 : *****
1040 005234 104400 TST5: SCOPE
1041 005236 012737 000100 011654 MOV #100,@#ICOUNT ;ITERATION COUNT
1042 005244 012737 000005 012636 MOV #5,@#ERTSTN ;SAVE TEST # FOR ERROR REPORT
1043 005252 012737 005260 011660 MOV #SCP5,@#RETURN ;SCOPE LOOP RETURN ADRS
1044 005260 SCP5:
1045
1046
1047 .REM *
1048
1049 THE FUNCTION OF THIS TEST IS TO VERIFY THAT THE DX11 CAN EXECUTE
1050 WRITE COMMANDS FROM THE CHANNEL SIMULATOR. THIS TEST IS IMPL-
1051 MENTED MUCH LIKE THE READ TEST IN THAT THE TRANSFER IS 256. BYTES,
1052 TT TRACING IS DONE AND SUPPRESS DATA IS ALSO CHECKED. TT DIFFERS
1053 IN; THE DIRECTION AND TYPE OF DATA.
1054
1055 *
1056
1057 005260 012737 000001 014216 MOV #WRITEC,@#CMD ;LOAD COMMAND
1058 005266 004737 006542 JSR PC,@#TRAIT ;TT TRACE TRACE INIT
1059 005272 012737 000200 006366 MOV #CHIS,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1
1060 005300 013737 001272 177776 MOV @#LESS1,PS ;LOWER PROCESSOR STATUS
1061 005306 012737 005306 012106 MOV #,@#TERPC ;ORIGIN OF TRAP ERROR
1062 005314 004737 007446 JSR PC,ISS.SUB ;SELECT
1063
1064
1065 005320 012737 000400 014252 MOV #256.,COUNT ;SOFTWARE COUNTER
1066 005326 012777 177400 173752 MOV #-256.,@DXBC ;DX BYTE COUNT
1067 005334 012777 023442 173742 MOV #NPRDATA,@DXBA ;ADRS FOR DATA FROM 360 SIM
1068 005342 012702 023442 MOV #NPRDATA,R2 ;INPUT FILE FOR NPR DATA
1069 005346 012701 022442 MOV #WDATA,R1 ;WRITE DATA FOR 360 SIM
1070 005352 052777 000003 173720 BIS #DXFI!GO,@DXCS ;FUNCTION INPUT & GO (360 WRITE)
1071 005360 012737 000200 006366 MOV #CHIS,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1
1072 005366 113737 014214 006426 MOV @#DEV,@#ENTRY2 ;SECOND TT ENTRY = DXCA
1073 005374 113737 014216 006427 MOV @#CMD,@#ENTRY2+1 ;
1074 005402 113777 014214 173716 MOV @#DEV,@CUAR ;LOAD DEV ADRS
1075 005410 113777 014216 173712 MOV @#CMD,@CUCR ;LOAD COMMAND IN CUCR
1076 005416 012737 005416 012106 MOV #,@#TERPC ;DEFINE ORIGIN OF TRACE ERROR
1077 005424 004737 011020 JSR PC,CUIS.SUB ;EXECUTE SELECTION
1078
1079
1080 005430 012737 000020 006366 WCO: MOV #CUDEND,@#ENTRY1 ;TT ENTRY 1
1081 005436 112737 177777 006427 MOVB #-1,@#ENTRY2+1 ;FAKE CUCR OF ENTRY 2
1082 005444 012737 005444 012106 MOV #,@#TERPC ;WRITE TRACE ERROR
1083 005452 032777 002000 173632 BIT #SRVI,@DXMI ;WAIT FOR SERVICE-IN
1084 005460 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1085 005462 104000 ERROR ;SRVI NOT SET
1086 005464 051177 173646 BIS @R1,@BUSO ;PUT DATA ON BUS-OUT
1087 005470 121177 173642 CMPB @R1,@BUSO ;VERIFY LOAD
1088 005474 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1089 005476 104000 ERROR ;BUSO INTO MEMORY DATA TRANSFER ERROR
1090 005500 052777 001000 173602 BIS #SRVO,@DXMO
1091 005506 032777 001000 173574 BIT #SRVO,@DXMO
1092 005514 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
```

```

1093 005516 104000          ERROR          ;SRVO NOT SET
1094 005520 032737 000004 013602  BIT          #BIT2,FARA          ;CAN'T LOOK AT SRVI ON MUX CH
1095 005526 001005          BNE          WC00
1096 005530 032777 002000 173554  BIT          #SRVI,@DXMI
1097 005536 001401          BEQ          .+4          ;BRANCH IF NO ERROR CONDITION
1098 005540 104000          ERROR          ;SRVI DID NOT DROP
1099 005542 005721          TST          (R1)+
1100 005544 105722          TSTB         (R2)+
1101 005546 032702 000001          BIT          #BIT0,R2          ;TEST FOR EVEN BOUNDRY
1102 005552 001016          BNE          WC1
1103
1104 005554 162701 000004          SUB          #4,R1
1105 005560 162702 000002          SUB          #2,R2
1106 005564 121112          CMPB        @R1,@R2          ;VERIFY 1ST DATA BYTE TRANSFER
1107 005566 001401          BEQ          .+4          ;BRANCH IF NO ERROR CONDITION
1108 005570 104000          ERROR          ;BUS0 INTO MEMORY DATA TRANSFER ERROR
1109 005572 005721          TST          (R1)+
1110 005574 105722          TSTB         (R2)+
1111 005576 121112          CMPB        @R1,@R2          ;VERIFY 2ND BYTE TRANSFER
1112 005600 001401          BEQ          .+4          ;BRANCH IF NO ERROR CONDITION
1113 005602 104000          ERROR          ;SECOND DATA BYTE INTO MEMORY
1114
1115 005604 005721          TST          (R1)+
1116 005606 105722          TSTB         (R2)+
1117 005610 052777 010000 173472  WC1:  BIS          #SUPO,@DXMO          ;SET SUPPRESS-OUT
1118 005616 032777 010000 173464  BIT          #SUPO,@DXMO          ;VERIFY SUPO SET
1119 005624 001001          BNE          .+4          ;BRANCH IF NO ERROR CONDITION
1120 005626 104000          ERROR          ;SUPO NOT SET
1121 005630 012727 000010          MOV          #10,(PC)+          ;SRVO COUNT
1122 005634 000000          0          ;SRVO COUNT
1123 005636 052777 001000 173444  WC3:  BIS          #SRVO,@DXMO          ;KEEP SRVO CHANGING
1124 005644 032777 001000 173436  BIT          #SRVO,@DXMO          ;VERIFY SRVO SET
1125 005652 001001          BNE          .+4          ;BRANCH IF NO ERROR CONDITION
1126 005654 104000          ERROR          ;SRVO DID NOT SET
1127 005656 042777 001000 173424  BIC          #SRVO,@DXMO          ;CLEAR SRVO
1128 005664 032777 001000 173416  BIT          #SRVO,@DXMO          ;VERIFY SRVO CLEAR
1129 005672 001401          BEQ          .+4          ;BRANCH IF NO ERROR CONDITION
1130 005674 104000          ERROR          ;SRVO NOT 0
1131 005676 032737 000004 013602  BIT          #BIT2,PARA          ;TEST FOR MUX MODE
1132 005704 001005          BNE          WC4
1133          ;WHEN ON MUX CH OPLI WILL DROP WHEN TRANSMISSION IS FINISHED
1134          ;THIS WILL CAUSE DXMI TO BE UNREADABLE
1135 005706 032777 002000 173376  BIT          #SRVI,@DXMI          ;SUPO SHOULD SUPPRESS SRVI SETTING
1136 005714 001401          BEQ          .+4          ;BRANCH IF NO ERROR CONDITION
1137 005716 104000          ERROR          ;SRVI WAS NOT SUPPRESSED BY SUPO
1138 005720 005337 005634          WC4:  DEC          WC2          ;DEC SRVO COUNT
1139 005724 001344          BNE          WC3          ;BRANCH IF SRVO NOT DONE
1140 005726 042777 010000 173354  BIC          #SUPO,@DXMO          ;DROP SUPPRESS-OUT
1141 005734 032777 010000 173346  BIT          #SUPO,@DXMO          ;VERIFY SUPO CLEAR
1142 005742 001401          BEQ          .+4          ;BRANCH IF NO ERROR CONDITION
1143 005744 104000          ERROR          ;SUPO STUCK HIGH
1144 005746 032777 001000 173334  BIT          #SRVO,@DXMO
1145
1146 005754 042777 001777 173326  BIC          #SRVO!777,@DXMO          ;CLR SRVO AND BUS0+PARITY
1147 005762 032777 001000 173320  BIT          #SRVO,@DXMO
1148 005770 001401          BEQ          .+4          ;BRANCH IF NO ERROR CONDITION
  
```

Address	OpCode	Operand 1	Operand 2	Operand 3	Operand 4	Comment
1149	005772	104000				ERROR ;SRVG DID NOT DROP
1150	005774	005337	014252			DEC COUNT
1151	006000	001213				BNE WCO
1152	006002	012737	001100	006366		MOV #CHENDS!ESEND,@#ENTRY1 ;LOAD EXPECTED TT ENTRY 1
1153	006010	113737	014214	006426		MOVB @#DEV,@#ENTRY2 ;SECOND TT ENTRY = DXCA
1154	006016	032737	000004	013602		BIT #BIT2,@#PARA ;TEST FOR MUX
1155	006024	001404				BEQ 1\$;BR IF SELECTOR
1156	006026	112737	000000	006427		MOVB #0,@#ENTRY2+1 ;MUX TT CUCR
1157	006034	000403				BR 2\$;GO
1158	006036	112737	177777	006427	1\$:	MOVB #-1,@#ENTRY2+1 ;CUCR OF DXCA TT ENTRY
1159	006044	113777	014214	173254	2\$:	MOVB @#DEV,@CUAR ;LOAD DEV ADRS
1160	006052	012737	006052	012106		MOV #.,@#TERPC ;DEFINE ORIGIN OF TRACE ERROR
1161	006060	004737	007064			JSR PC,STATUSPRESENTATION ;PRESENT STATUS
1162						
1163						

```
1164 : *****
1165 : TEST 6 END OF TEST STRING
1166 : *****
1167 006064 104400 TST6: SCOPE
1168 006066 012737 000001 011654 MOV #1,@#ICOUNT ;ITERATION COUNT
1169 006074 012737 000006 012636 MOV #6,@#ERTSTN ;SAVE TEST # FOR ERROR REPORT
1170 006102 012737 006110 011660 MOV #SCP6,@#RETURN ;SCOPE LOOP RETURN ADRS
1171 006110 SCP6:
1172
1173
1174 .REM *
1175
1176 THIS TEST FUNCTIONS AS A TERMINATOR FOR THE CHAINABLE TEST STRING.
1177 AS SUCH IT TRANSFERS CONTROL TO THE LOOP CONTROL SUBROUTINE.
1178 *
1179
1180
1181
1182 006110 000137 013666 JMP @#LPCNTL
1183
1184
1185
1186 ;INTERRUPT HANDLERS
1187
1188 006114 104000 FALSE: ERROR ;FALSE OR UNEXPECTED INTERRUPT
1189 006116 000002 RTI
1190
1191 006120 032777 000200 173152 INTR: BIT #DONE,@DXCS ;TEST DONE
1192 006126 001001 BNE .+4 ;BRANCH ON DONE
1193 006130 104000 ERROR ;FALSE INTERRUPT
1194 006132 042777 000200 173140 BIC #DONE,@DXCS ;CLEAR INT CONDION
1195 006140 052737 100000 006150 BIS #INTOK,INTPAS ;SET INT PASS FLAG
1196 006146 000002 RTI
1197
1198 006150 000000 INTPAS: 0 ;INTERRUPT PASS FLAG
1199
1200
1201 ;INTERR, ROUTINE TO TEST FOR SUCCESSFUL INTERRUPT
1202
1203 006152 INTERR:
1204 006152 032737 100000 006150 BIT #INTOK,INTPAS ;DID INTERRUPT OCCUR
1205 006160 001405 BEQ IRR ;BRANCH IF NOT
1206 006162 062716 000002 006150 IRR2: ADD #2,@SP ;INC RETURN PC
1207 006166 042737 100000 006150 BIC #INTOK,INTPAS ;CLEAR PASS FLAG
1208 006174 000207 IRR: RTS PC
1209
1210 ;ZEROTT, ROUTINE TO ZERO TUMBLE TABLE
1211
1212 006176 ZEROTT:
1213 006176 010146 MOV R1,-(SP)
1214 006200 010246 MOV R2,-(SP)
1215 006202 013701 001440 MOV TT,R1
1216 006206 012702 000400 MOV #256.,R2
1217 006212 005021 ZTT1: CLR (R1)+
1218 006214 005302 DEC R2
1219 006216 001375 BNE ZTT1
```



```
1220 006220 012602      MOV      (SP)+,R2
1221 006222 012601      MOV      (SP)+,R1
1222 006224 000207      RTS      PC
1223
1224      ;TTZERO, ROUTINE TO VERIFY TT ZERO
1225
1226 006226      TTZERO:
1227 006226 010146      MOV      R1,-(SP)
1228 006230 010246      MOV      R2,-(SP)
1229 006232 013701 001440      MOV      TT,R1
1230 006236 012702 000400      MOV      #256.,R2
1231 006242 005721      TTZ1:  TST      (R1)+
1232 006244 001401      BEQ      .+4      ;BRANCH IF NO ERROR CONDITION
1233 006246 104000      ERROR    ;ILLEGAL TT ENTRY
1234 006250 005302      DEC      R2
1235 006252 001373      BNE      TTZ1
1236 006254 012602      MOV      (SP)+,R2
1237 006256 012601      MOV      (SP)+,R1
1238 006260 000207      RTS      PC
1239
1240      ;ISS DONE INTERRUPT SERVICE
1241 006262      SEL.DONE:
1242 006262 032777 000200 173010      BIT      #DONE,@DXCS      ;CHECK DONE
1243 006270 001001      BNE      .+4      ;BRANCH IF NO ERROR CONDITION
1244 006272 104000      ERROR    ;FALSE INTERRUPT
1245 006274 042777 000200 172776      BIC      #DONE,@DXCS      ;CLEAR DONR
1246 006302 032777 000200 172770      BIT      #DONE,@DXCS
1247 006310 001401      BEQ      .+4      ;BRANCH IF NO ERROR CONDITION
1248 006312 104000      ERROR    ;DONE NOT CLEAR
1249 006314 012746 006540      MOV      #SEL.X,-(SP)      ;FAKE A JSR PC,TT.TRACE
1250
1251      ;TT.TRACE, ROUTINE TO TRACE TUMBLE TABLE ENTRIES
1252      ;AND THE TTNDX
1253 006320      TT.TRACE:
1254
1255 006320 010146      MOV      R1,-(SP)
1256 006322 013701 006526      MOV      @#TTTRACE,R1      ;LOAD R1 WITH SOFTWARE IT
1257 006326 020137 001440      CMP      R1,@#TT      ;CHECK FOR BOTTOM OF TABLE
1258 006332 001005      BNE      1$      ;BRANCH IF NOT BOTTOM
1259 006334 005737 003776      TST      @#3776      ;LOOK AT TOP OF TT
1260 006340 001401      BEQ      .+4      ;BRANCH IF NO RAP AROUND
1261 006342 104002      TRACER   ;REPORT TT TRACE ERROR
1262 006344 000404      BR      2$
1263 006346 005741      1$:  TST      -(R1)      ;CHECK FOR TT OVERFLOW
1264 006350 001401      BEQ      .+4      ;BRANCH IF NO RAP AROUND
1265 006352 104002      TRACER   ;TT OVERFLOW ERROR
1266 006354 005721      TST      (R1)+      ;INC TO ENTRY
1267 006356 011127      2$:  MOV      (R1),(PC)+      ;SAVE ENTRY ONE
1268 006360 000000      SENRY1: 0      ;HERE
1269 006362 023727 006360      ENTRY1: 0      ;COMPARE SAVED ENTRY WITH
1270 006366 000000      ENTRY1: 0      ;EXPECTED ENTRY
1271 006370 001407      BEQ      1$      ;BRANCH IF DXDS ENTRY OK
1272 006372 013737 006360 006534      MOV      @#SENRY1,@#TTWAS
1273 006400 013737 006366 006536      MOV      @#ENTRY1,@#TTSHOULD
1274 006406 104002      TRACER   ;REPORT TT TRACE ERROR
1275 006410 005037 006366      1$:  CLR      ENTRY1
```

```
1276 006414 005021          CLR      (R1)+      ;CLEAR ENTRY AND ADVANCE POINTER
1277 006416 011127          MOV      (R1),(FC)+ ;SAVE ENTRY TWO
1278
1279 006420 000000          SENRY2: 0           ;HERE
1280 006422 023727 006420          CMP      @#SENRY2,(PC)+ ;COMPARE SAVED ENTRY WITH
1281 006426 000000          ENTRY2: 0          ;EXPECTED ENTRY
1282 006430 001411          TT.T0:  BEQ      2$           ;BRANCH IF DXCA ENTRY OK
1283 006432 013737 006420 006534          MOV      @#SENRY2,@#TTWAS
1284 006440 013737 006426 006536          MOV      @#ENTRY2,@#TTSHOULD
1285 006446 104002          TRACER          ;REPORT TT TRACE ERROR
1286 006450 005037 006426          CLR      ENTRY2
1287 006454 005021          2$:          CLR      (R1)+      ;CLEAR
1288 006456 022701 004000          CMP      #TST1,R1    ;CHECK FOR SOFTWARE
1289 006462 001002          BNE      TT.T1       ;BRANCH IF NO OVERFLOW
1290 006464 013701 001440          MOV      @#TT,R1
1291 006470 005037 006500          TT.T1:  CLR      @#TT.T2
1292 006474 117727 172650          MOVVB   @TTNDX,(PC)+ ;SAVE TTNDX
1293 006500 000000          TT.T2:  0           ;HERE
1294 006502 006337 006500          ASL      TT.T2       ;SCALE MOD(2)
1295 006506 063737 001440 006500          ADD      @#TT,@#TT.T2 ;ADD BASE OF TT TO INDEX
1296 006514 123701 006500          CMPB    @#TT.T2,R1   ;COMPARE TT POINTERS
1297 006520 001401          BEQ      .+4         ;BRANCH IF HARDWARE AND SOFTWARE TT POINTERS MATCH
1298 006522 104002          TRACER          ;REPORT TT TRACE ERROR
1299 006524 010127          MOV      R1,(PC)+   ;SAVE TT TRACE
1300 006526 000000          TTRACE: 0          ;HERE
1301 006530 012601          MOV      (SP)+,R1
1302 006532 000207          RTS      PC          ;RETURN
1303 006534 000000          TTWAS:  0           ;ACTUAL CONTENTS OF TT
1304 006536 000000          TTSHOULD: 0        ;EXPECTED CONTENTS OF TT
1305
1306 006540 000002          SEL.X:  RTI
1307
1308          ;SUBROUTINE TO INIT DX FOR TT TRACINR
1309
1310 006542 042777 077777 172540          TRRAINT: BIC      #77777,@DXMO ;DESELECT
1311 006550 112777 000100 172664          MOVVB   #SM,@DST    ;RESPONSE TO TEST IO
1312 006556 032737 000002 013602          BIT     #BIT1,@#PARA ;CHECK FOR CUBSY MODE
1313 006564 001410          BEQ      TI.0       ;BRANCH IF NOT CU BUSY
1314 006566 052777 004000 172504          BIS     #BSYEN,@DXCS ;SET BSYEN
1315 006574 032777 004000 172476          BIT     #BSYEN,@DXCS ;VERIFY SET
1316 006602 001001          BNE      .+4         ;BRANCH IF NO ERROR CONDITION
1317 006604 104000          ERROR          ;BSYEN NOT SET
1318 006606 113737 014214 006426          TI.0:  MOVVB   @#DEV,@#ENTRY2 ;SECOND TT ENTRY = DXCA
1319 006614 113737 014216 006427          MOVVB   @#CMD,@#ENTRY2+1 ;
1320 006622 113777 014216 172500          MOVVB   @#CMD,@CUAR   ;LOAD COMMAND
1321 006630 113777 014214 172470          MOVVB   @#DEV,@CUAR  ;LOAD COMMAND IN CUAR
1322 006636 052777 000100 172434          BIS     #INTEN,@DXCS ;SET INTERRUPT ENABLE
1323 006644 032777 000100 172426          BIT     #INTEN,@DXCS ;VERIFY SET
1324 006652 001001          BNE      .+4         ;BRANCH IF NO ERROR CONDITION
1325 006654 104000          ERROR          ;INTEN NOT SET
1326 006656 013777 001270 172402          MOV     @#DXPRT,@DXIS ;LOAD INT STATUS
1327 006664 012777 006262 172372          MOV     #SEL.DONE,@DXIV ;LOAD INT VECTOR
1328 006672 000207          RTS      PC
1329
1330
1331
```

1332 006674 104000
 1333 006676 022626
 1334 006700 000137 012724
 1335
 1336
 1337
 1338
 1339 006704 000002
 1340 006706 000002
 1341
 1342
 1343
 1344
 1345 006710 000020
 1346 006712 013727 006710
 1347 006716 000000
 1348 006720 005337 006716
 1349 006724 001375
 1350 006726 000207
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1360
 1361
 1362
 1363 006730 000240
 1364 006732 042777 000200 172340
 1365 006740 042777 060000 172342
 1366 006746 042777 177000 172336
 1367 006754 052777 000010 172336
 1368 006762 042737 100000 006150
 1369 006770 032737 000001 013602
 1370 006776 001402
 1371 007000 004737 007414
 1372 007004 112777 000100 172430
 1373 007012 032737 000002 013602
 1374 007020 001405
 1375 007022 052777 004000 172250
 1376 007030 105077 172406
 1377 007034 012777 006120 172222
 1378 007042 013777 001270 172216
 1379 007050 052777 000100 172222
 1380 007056 005037 177776
 1381 007062 000207
 1382
 1383
 1384
 1385
 1386
 1387 007064

```

O.BRK: ERROR ;BREAK TRAP IS ILLEGAL
        CMP (SP)+,(SP)+ ;RESTORE STACK
        JMP @MON1.0
:***** MOD APR 74 *****
:*      11/40.11/45 RTT
:*      TRACE TRAP MOD

YESRTI: RTI
RTX:    RTI ;MODIFIED FOR 11/40.11/45 TO RTT
:***** MOD APR 74 *****
:*      TIMING
:*      MOS.BIPOLAR RESPONSE MOD

DELAY:  20
RDLAY:  MOV    DELAY,(PC)+
1$:     0
2$:     DEC    1$ ;DELAY FOR BIPOLAR MEMORY
        BNE   2$
        RTS   PC

:***** MOD APR 74 *****
:

.SBTTL  SETUP SELECTED PARAMETERS

;SETUP, ROUTINE TO CLEAR DX AND SETUP THE FOLLOWING
:
:      BIT2= MUX CH
:      BIT1= BSYEN
:      BIT0= ON-LINE

SETUP:  NOP
        BIC   #DONE,@DXCS ;CLR'DONE'+ THEREFORE 'LOCKO'
        BIC   #HLDO!SELO,@DXMO ;GET READY FOR NEXT ISS
        BIC   #177000,@DXMI
        BIS   #TIMDIS,@DXES ;SET DXTO DISABLE
        BIC   #INTOK,INTPAS ;CLEAR INTERRUPT PASS FLAG
        BIT   #BIT0,PARA ;TEST FOR ONLINE
        BEQ   SP1 ;BRANCH IF NOT ONLINE
        JSR   PC,FOL ;DO FAST ONLINE
        MOVB  #SM,@DST ;RESPONSE TO TIO
        BIT   #BIT1,PARA ;WAS BSYEN SELECTED
        BEQ   SP2 ;BRANCH IF NOT
        BIS   #BSYEN,@DXCS ;SET BUSY ENABLE
        CLRB  @DST ;TIO GETS STATUS BY HARDWARE
        MOV   #INTR,@DXIV ;SET UP DX INTERRUPT VECTOR
        MOV   DXPRT,@DXIS ;SET UP DX INTERRUPT STATUS
        BIS   #INTEN,@DXCS ;SET INT ENABLE
        CLR   PS ;CLEAR PROCESSOR STATUS
        RTS   PC ;RETURN
  
```

.SBTTL STATUS PRESENTATION
 STATUSPRESENTATION:

```

1388 007064 042777 000200 172206      BIC      #DONE,@DXCS      ;CLEAR DONE AND LOCKO
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401 007072 112777 000014 172232      MOVB     #CE+DE,@CUSR    ;LOAD CE/DE STATUS
1402
1403 007100 052777 000007 172172      BIS      #DXFST!GO,@DXCS ;SET FCTN/STATUS/GO
1404                                     ;STATUS-IN SHOULD RISE
1405
1406 007106 004737 011020      JSR      %7,CUIS.SUB     ;EXECUTE CUIS
1407 007112 042777 060000 172170      BIC      #HLDO!SELO,@DXMO ;DROP SELO!HLDO FOR NONZERO STATUS
1408 007120 052777 001000 172162      BIS      #SRVO,@DXMO     ;SERVICE-OUT UP
1409                                     ;STATUS-IN SHOULD FALL
1410
1411
1412 007126 042777 001000 172154      BIC      #SRVO,@DXMO     ;DROP SERVICE-OUT,
1413                                     ;THIS ENDS ES -- ESEND INT SHOULD
1414                                     ;BE UP AND CUBSY CLEAR
1415                                     ;CHECK FOR INTERRUPTS
1416 007134 005077 172152      CLR      @DXMI          ;CLEAR DXMI
1417 007140 000207      RTS      %7
1418
1419                                     .SBTTL DATA TRANSFER ROUTINES
1420 007142                                     TRANSFER:
1421 007142 032737 000001 014216      BIT      #1,CMD          ;IS COMMAND READ OR WRITE?
1422 007150 001012      BNE     TR.WRITE        ;WRITE
1423
1424                                     ;OUTPUT HERE
1425 007152                                     TR.READ:
1426 007152 052777 001000 172130      BIS      #SRVO,@DXMO     ;SERVICE-OUT UP
1427                                     ;SERVICE-IN SHOULD FALL
1428 007160 042777 001000 172122      BIC      #SRVO,@DXMO     ;SERVICE-OUT DOWN
1429                                     ;SERVICE-IN UP AGAIN IF MORE
1430 007166 005337 014252      DEC     COUNT           ;LOOP TILL DONE
1431 007172 001367      BNE     TR.READ         ;...
1432 007174 000416      BR      TR.OUT
1433
1434                                     ;INPUT HERE
1435 007176                                     TR.WRITE:
1436 007176 052177 172134      BIS      (R1)+,@BUSO     ;PUT DATA ON BUS
1437 007202 052777 001000 172100      BIS      #SRVO,@DXMO     ;SERVICE-OUT UP
1438                                     ;SERVICE-IN SHOULD FALL
1439 007210 042777 001000 172072      BIC      #SRVO,@DXMO     ;SERVICE-OUT DOWN
1440 007216 042777 000777 172112      BIC      #777,@BUSO     ;TAKE DATA OFF BUS (SERVICE-IN UP)
1441 007224 005337 014252      DEC     COUNT
1442 007230 001362      BNE     TR.WRITE
1443

```

1444 007232 000207 TR.OUT: RTS %7

1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499

.SBTTL ONLINE ROUTINE

: *****
:ONLIN, ROUTINE TO SET DX ONLINE
:AND WAIT FOR RELAY TO PICK
:

007234	012737	010000	007412	ONLIN:	MOV	#10000,CNT	:WAIT FOR RELAY TO SETTLE
007242	005337	007412		OL4:	DEC	CNT	
007246	001375				BNE	OL4	
007250	013777	001436	172024		MOV	SPW,@DXOS	:LOAD DX OFFSET
007256	023777	001436	172016		CMP	SPW,@DXOS	:VERIFY OFFSET IS LOADED
007264	001401				BEQ	+.4	:BRANCH IF NO ERROR CONDITION
007266	104000				ERROR		:DXOS LOAD ERROR
007270	052777	001000	172002		BIS	#ONLINA,@DXCS	:SET ONLINE BIT
007276	032777	001000	171774		BIT	#ONLINA,@DXCS	:ONLINE REQUEST SET
007304	001001				BNE	+.4	:BRANCH IF NO ERROR CONDITION
007306	104000				ERROR		:ONLINA DID NOT SET
007310	012737	010000	007412		MOV	#10000,CNT	:SET UP STALL FOR ONLINE RELAY
007316	005337	007412		OL2:	DEC	CNT	
007322	001375				BNE	OL2	
007324	032777	000004	171762		BIT	#ONLINB,@DXCB	:IS DX ONLINE
007332	001001				BNE	+.4	:BRANCH IF NO ERROR CONDITION
007334	104000				ERROR		:ONLINB DID NOT SET

```

1500 007336 032777 100000 171744 BIT #OPLO,@DXMO ;
1501 007344 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1502 007346 104000 ERROR ;ARE YOU REALLY MAINTENANCED CABLED?
1503 007350 052777 001000 171732 OL3: BIS #SRVO,@DXMO ;
1504 007356 032777 001000 171724 BIT #SRVO,@DXMO ;CHECK SERVICE OUT
1505 007364 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1506 007366 104000 ERROR ;SRVO NOT SET
1507 007370 042777 001000 171712 BIC #SRVO,@DXMO ;
1508 007376 032777 001000 171704 BIT #SRVO,@DXMO ;CHECK SERVICE OUT
1509 007404 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1510 007406 104000 ERROR ;SRVO NOT ZERO
1511
1512 007410 000207 RTS PC ;
1513 007412 000000 CNT: 0 ;STALL
1514
1515
1516
1517 ;FAST ONLINE ROUTINE
1518
1519 007414 052777 001000 171656 FOL: BIS #ONLINA,@DXCS ;SET ONLINE BIT
1520 007422 032777 001000 171650 BIT #ONLINA,@DXCS ;ONLINE REQUEST SET
1521 007430 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1522 007432 104000 ERROR ;ONLINA NOT SET
1523 007434 032777 000004 171652 FOL1: BIT #ONLINB,@DXCB ;IS DX ONLINE
1524 007442 001774 BEQ FOL1 ;WAIT FOR RELAY TO PICK
1525 007444 000207 RTS PC
1526
1527
1528
1529 .SBTTL INITIAL SELECTION SEQUENCE
1530
1531 ;INITIAL SELECTION SEQUENCE
1532
1533 007446 ISS.SUB:
1534 007446 032777 100000 171636 BIT #OPLI,@DXMI ;IS OPRI CLEAR
1535 007454 001401 BEQ 1$ ;IF YES CONTINUE
1536 007456 104000 ERROR ;ERROR-OPLI NOT RESET
1537 007460 052777 060400 171622 1$: BIS #PARO!HLDO!SELO,@DXMO ;
1538 007466 032777 001000 171616 BIT #CLKO,@DXMI ;IS CLKO CLEAR
1539 007474 001001 BNE 2$ ;IF NO CONTINUE
1540 007476 104000 ERROR ;ERROR-CLKO DID NOT SET
1541 007500 032777 040000 171604 2$: BIT #SELI,@DXMI ;IS SELI CLEAR
1542 007506 001001 BNE 3$ ;IF NO CONTINUE
1543 007510 104000 ERROR ;ERROR-SELI DID NOT SET
1544 007512 032777 000004 171604 3$: BIT #SELIBR,@DXES1 ;IS SELIBR SET
1545 007520 001001 BNE 4$ ;IF NO CONTINUE
1546 007522 104000 ERROR ;ERROR-SELIBR DID NOT SET
1547 007524 042777 060400 171556 4$: BIC #PARO!HLDO!SELO,@DXMO ;
1548 007532 032777 001000 171552 BIT #CLKO,@DXMI ;IS CLKO CLEAR
1549 007540 001401 BEQ 5$ ;IF YES CONTINUE
1550 007542 104000 ERROR ;ERROR-CLKO NOT RESET
1551 007544 032777 040000 171540 5$: BIT #SELI,@DXMI ;IS SELI CLEAR
1552 007552 001401 BEQ 6$ ;IF YES CONTINUE
1553 007554 104000 ERROR ;ERROR-SELI NOT RESET
1554 007556 032777 000004 171540 6$: BIT #SELIBR,@DXES1 ;IS SELIBR SET
1555 007564 001401 BEQ 7$ ;IF YES CONTINUE

```

```
1556 007566 104000          ERROR          ;ERROR-SELIBR NOT RESET
1557 007570 052777 040000 171512 7$: BIS #HLDO,@DXMO ;SET HLDO
1558 007576 032777 020000 171504 BIT #SELO,@DXMO ;SELO SHOULD NOT SET
1559 007604 001401          BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1560 007606 104000          ERROR          ;SELO SHOULD NOT BE SET
1561 007610 042777 040000 171472 8$: BIC #HLDO,@DXMO ;CLEAR HLDO
1562 007616 053777 014214 171464 BIS DEV,@DXMO ;LOAD BUSO WITH DEVICE ADRS
1563 007624 123777 014214 171504 CMPB DEV,@BUSO ;TEST BUSO FOR CORRECT ADRS
1564 007632 001401          BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1565 007634 104000          ERROR          ;BUSO DID NOT LOAD
1566                                     ;***** MOD APR 74 *****
1567                                     ;* INITIAL SELECTION CLEANUP MOD
1568 007636 113777 014214 171462 MOVB DEV,@CUAR
1569 007644 123777 014214 171454 CMPB DEV,@CUAR
1570 007652 001401          BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1571 007654 104000          ERROR          ;CUAR BITS 7 THRU 0 FAILURE
1572                                     ;***** MOD APR 74 *****
1573
1574 007656 052777 004000 171424 BIS #ADRO,@DXMO ;SET ADRS-OUT
1575 007664 132777 000010 171446 BITB #10,@CONO ;TEST ADRO UP
1576 007672 001001          BNE .+4 ;BRANCH IF NO ERROR CONDITION
1577 007674 104000          ERROR          ;ADRO DID NOT SET
1578
1579 007676 052777 040000 171404 BIS #HLDO,@DXMO ;SET HOLD-OUT
1580 007704 032777 040000 171376 BIT #HLDO,@DXMO ;TEST FOR HOLD-OUT UP
1581 007712 001001          BNE .+4 ;BRANCH IF NO ERROR CONDITION
1582 007714 104000          ERROR          ;HOLD OUT DID NOT SET
1583
1584 007716 052777 020000 171364 BIS #SELO,@DXMO ;SET SELECT-OUT
1585 007724 132777 000040 171406 BITB #40,@CONO ;TEST FOR SELECT-OUT
1586 007732 001001          BNE .+4 ;BRANCH IF NO ERROR CONDITION
1587 007734 104000          ERROR          ;SELECT-OUT DID NOT SET
1588
1589
1590 007736 032777 000002 171350 BIT #ADRECC,@DXCB ;WAS CU ADRS RECOGNISED
1591                                     ;IS DEV A VALID CU ADRS
1592 007744 001001          BNE .+4 ;BRANCH IF NO ERROR CONDITION
1593 007746 104000          ERROR          ;DX DID NOT RECOGNISE CU ADRS
1594                                     ;IF YES CHECK JUMPERS
1595                                     ;ADRECC LOGIC
1596
1597
1598 007750 032777 000010 171316 BIT #ISSREJ,@DXDS ;TEST FOR ISS REJECT
1599 007756 001401          BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1600 007760 104000          ERROR          ;ISS REJECT SET
1601 007762 032777 100000 171322 BIT #OPLI,@DXMI ;OPLI SHOULD BE UP
1602 007770 001001          BNE .+4 ;BRANCH IF NO ERROR CONDITION
1603 007772 104000          ERROR          ;OPLI DID NOT SET
1604                                     ;IS DEV A VALID DEVICE ON CU?
1605
1606 007774 042777 004000 171306 BIC #ADRO,@DXMO ;DROP ADRS-OUT
1607 010002 132777 000010 171330 BITB #10,@CONO ;IS ADRO DOWN
1608 010010 001401          BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1609 010012 104000          ERROR          ;ADRO DID NOT DROP
1610 010014 032737 000004 013602 BIT #BIT2,PARA ;TEST FOR MUX CH
1611 010022 001415          BEQ ISS3 ;BRANCH IF NOT MUX
```

1612	010024	042777	060000	171256	BIC	#SELO,HLDO,@DXMO	:CLEAR SELECT-OUT, HOLD-OUT
1613	010032	032777	020000	171250	BIT	#SELO,@DXMO	:DID SELO CLEAR
1614	010040	001401			BEQ	+.4	:BRANCH IF NO ERROR CONDITION
1615	010042	104000			ERROR		:REPORT SELO NOT CLEAR
1616	010044	032777	040000	171236	BIT	#HLDO,@DXMO	:DID HLDO CLEAR
1617	010052	001401			BEQ	+.4	:BRANCH IF NO ERROR CONDITION
1618	010054	104000			ERROR		:HLDO DID NOT CLEAR
1619	010056	032777	010000	171226	ISS3: BIT	#ADRI,@DXMI	:ADDRESS-IN SHOULD BE UP
1620	010064	001001			BNE	+.4	:BRANCH IF NO ERROR CONDITION
1621	010066	104000			ERROR		:ADRI DID NOT COME UP
1622							
1623	010070	123777	014214	171244	CMPB	DEV,@BUSI	:ADRS SHOULD BE ON BUSI
1624	010076	001401			BEQ	+.4	:BRANCH IF NO ERROR CONDITION
1625	010100	104000			ERROR		:BUSI DOES NOT CONTAIN
1626							:CORRECT ADRS
1627							
1628	010102	032777	000400	171200	BIT	#PARO,@DXMO	:IS PARO SET
1629	010110	001405			BEQ	OFF	:IF NO ,GO TO OFF
1630	010112	032777	000400	171172	BIT	#PARI,@DXMI	:IS PARI SET
1631	010120	001006			BNE	RET	:IF YES CONTINUE
1632	010122	104000			ERROR		:ERROR-PARI NOT ON
1633	010124	032777	000400	171160	OFF: BIT	#PARI,@DXMI	:IS PARI CLEAR
1634	010132	001401			BEQ	RET	:IF YES CONTINUE
1635	010134	104000			ERROR		:ERROR-PARI NOT CLEAR
1636	010136	043777	014214	171172	RET: BIC	DEV,@BUSO	:REMOVE ADRS FROM BUS-OUT
1637	010144	105777	171166		TSTB	@BUSO	:ADRS REMOVED
1638	010150	001401			BEQ	+.4	:BRANCH IF NO ERROR CONDITION
1639	010152	104000			ERROR		:BUSO DID NOT CLEAR
1640							
1641	010154	053777	014216	171154	BIS	CMD,@BUSO	:BUT COMMAND ON BUSO
1642	010162	123777	014216	171146	CMPB	CMD,@BUSO	:DID COMMAND LOAD OK
1643	010170	001401			BEQ	+.4	:BRANCH IF NO ERROR CONDITION
1644	010172	104000			ERROR		:CMD DID NOT LOAD PROPERLY
1645							
1646	010174	052777	002000	171106	BIS	#CMDO,@DXMO	:SET COMMAND OUT
1647	010202	132777	000004	171130	BITB	#4,@CONO	:DID CMDO SET
1648	010210	001001			BNE	+.4	:BRANCH IF NO ERROR CONDITION
1649	010212	104000			ERROR		:CMDO DID NOT SET
1650							
1651	010214	032777	000001	171052	ISSCRJ: BIT	#CMDREJ,@DXDS	:TEST FOR COMMAND REJECT
1652	010222						
1653	010222	001401			BEQ	+.4	:BRANCH IF NO ERROR CONDITION
1654	010224	104000			ERROR		:COMMAND REJECTED
1655							
1656	010226	032777	010000	171056	BIT	#ADRI,@DXMI	:ADRI SHOULD FALL
1657	010234	001401			BEQ	+.4	:BRANCH IF NO ERROR CONDITION
1658	010236	104000			ERROR		:ADRI DID NOT DROP
1659							
1660	010240	043777	014216	171042	BIC	CMD,@DXMO	:REMOVE CMD FROM BUSG
1661	010246	105777	171064		TSTB	@BUSO	:WAS CMD REMOVED
1662	010252	001401			BEQ	+.4	:BRANCH IF NO ERROR CONDITION
1663	010254	104000			ERROR		:CMD DID NOT CLEAR
1664	010256	042777	002000	171024	BIC	#CMDO,@DXMO	
1665	010264	032777	002000	171016	BIT	#CMDO,@DXMO	
1666	010272	001401			BEQ	+.4	:BRANCH IF NO ERROR CONDITION
1667	010274	104000			ERROR		:CMDO STUCK HIGH


```
1668
1669
1670 010276 032777 004000 171006 BIT #STAI,@DXMI ;TEST FOR STATUS-IN
1671 010304 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1672 010306 104000 ERROR ;STATUS-IN DID NOT RISE
1673 010310 010146 MOV R1,-(SP)
1674 010312 113701 014216 MOVB @#CMD,R1
1675 010316 042701 177400 BIC #177400,R1
1676 010322 063701 001442 ADD @#DST,R1
1677 010326 121177 171000 CMPB @R1,@CUR
1678 010332 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1679 010334 104000 ERROR ;STATUS FROM DST TRANSFER ERROR
1680 010336 121177 171000 CMPB @R1,@BUSI
1681 010342 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1682 010344 104000 ERROR ;CUR TO BUSI TRANSFER ERROR
1683 010346 012601 MOV (SP)+,R1
1684
1685
1686
1687 010350 032737 000004 013602 BIT #BIT2,PARA ;TEST FOR MUX CH
1688 010356 001023 BNE ISS2 ;BRANCH IF MUX
1689 010360 105737 014216 TSTB CMD ;TEST FOR 'TEST I/O' COMMAND
1690 010364 001403 BEQ ISS1 ;CLEAR SELO,HLDO IF T10 CMD
1691 010366 105777 170750 TSTB @BUSI ;TEST BUSI FOR ZERO STATUS
1692 010372 001415 BEQ ISS2 ;DON'T CLEAR SELO,HLDO ON 0 STATUS
1693 010374 042777 060000 170706 ISS1: BIC #SELO!HLDO,@DXMO ;CLEAR SELECT-OUT, HOLD-OUT
1694 010402 032777 020000 170700 BIT #SELO,@DXMO ;DID SELO CLEAR
1695 010410 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1696 010412 104000 ERROR ;REPORT SELO NOT CLEAR
1697 010414 032777 040000 170666 BIT #HLDO,@DXMO ;DID HLDO CLEAR
1698 010422 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1699 010424 104000 ERROR ;HLDO DID NOT CLEAR
1700
1701 010426 052777 001000 170654 ISS2: BIS #SRVO,@DXMO ;SET SERVICE-OUT
1702 010434 032777 001000 170646 BIT #SRVO,@DXMO ;TEST SERVICE OUT
1703 010442 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1704 010444 104000 ERROR ;SERVICE-OUT DID NOT SET
1705 : ***** CAN'T READ DXMI *****
1706 : BIT #STAI,@DXMI ;STATUS-IN SHOULD DROP
1707 : ;STATUS-IN DID NOT DROP
1708 :
1709 : ERCALL BEQ,
1710 : *****
1710 010446 042777 001000 170634 BIC #SRVO,@DXMO ;DROP SERVICE-OUT
1711 010454 032777 001000 170626 BIT #SRVO,@DXMO ;TEST SERVICE-OUT
1712 010462 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1713 010464 104000 ERROR ;SERVICE-OUT DID NOT CLEAR
1714
1715 010466 043777 014216 170614 BIC CMD,@DXMO ;CLEAR COMMAND FROM BUSO
1716 010474 105777 170636 TSTB @BUSO ;TEST FOR BUSO CLEAR
1717 010500 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1718 010502 104000 ERROR ;BUSO DID NOT CLEAR
1719
1720 010504 000207 RTS PC
1721
1722 .SBTTL HIO SUBROUTINE
1723
```

```

1724 010506 032777 070000 170600 HIO.SUB: BIT #70000,@DXCB ;TEST FOR PHASE ZERO
1725 010514 001103 BNE HIO.0 ;BRANCH IF SELECTED
1726
1727 ;SELECT
1728
1729 010516 053777 014214 170564 BIS DEV,@DXMO ;PUT ADRS ON BUSO
1730 010524 123777 014214 170604 CMPB DEV,@BUSO ;ADRS LOAD OK
1731 010532 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1732 010534 104000 ERROR ;ADRS LOAD ERROR
1733
1734 010536 052777 004000 170544 BIS #ADRO,@DXMO ;SET & CHECK ADRS-OUT
1735 010544 032777 004000 170536 BIT #ADRO,@DXMO
1736 010552 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1737 010554 104000 ERROR ;ADRO NOT SET
1738 010556 052777 040000 170524 BIS #HLDO,@DXMO ;SET & CHECK HLDO
1739 010564 032777 040000 170516 BIT #HLDO,@DXMO
1740 010572 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1741 010574 104000 ERROR ;HLDO NOT SET
1742
1743 010576 052777 020000 170504 BIS #SELO,@DXMO ;SET & CHECK SELECT-OUT
1744 010604 032777 020000 170476 BIT #SELO,@DXMO
1745 010612 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
1746 010614 104000 ERROR ;SELO NOT SET
1747
1748 010616 032737 000002 013602 BIT #BIT1,PARA ;TEST FOR BSY SELECT
1749 010624 001406 BEQ HIOS.0
1750 010626 122777 000120 170506 CMPB #STAMOD!BSY,@BUSI ;CHECK STATUS
1751 010634 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1752 010636 104000 ERROR ;BUSI STATUS ERROR
1753 010640 000431 BR HIO.0
1754 010642
1755 010642 123777 014214 170456 HIOS.0: CMPB DEV,@CUAR ;VERIFY DEVICE ADDRESS
1756 010650 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1757 010652 104000 ERROR ;BUSO TO CUAR TRANSFER ERROR
1758
1759 010654 042777 004000 170426 BIC #ADRO,@DXMO ;CLEAR & CHECK
1760 010662 032777 004000 170420 BIT #ADRO,@DXMO
1761 010670 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1762 010672 104000 ERROR ;ADRO SET
1763
1764 010674 123777 014214 170440 CMPB DEV,@BUSI ;IS CORRECT ADRS BEING ECHOED
1765 010702 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1766 010704 104000 ERROR ;BUSI LOAD ERROR
1767 ;CHECK ADRECC,D
1768
1769 010706 043777 014214 170374 BIC DEV,@DXMO ;REMOVE ADRS FROM BUSO
1770 010714 105737 001310 TSTB DXMO ;BRANCH IF CLEAR
1771 010720 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
1772 010722 104000 ERROR ;BUSO DID NOT CLEAR
1773
1774
1775
1776 010724 HIO.0: ;INTERFACE DISCONNECT
1777 010724 042777 060000 170356 BIC #HLDO!SELO,@DXMO ;
1778 010732 032777 040000 170350 BIT #HLDO,@DXMO ;
1779 010740 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
    
```

```

1780 010742 104000          ERROR          ;HLDO SET
1781
1782 010744 032777 020000 170336  BIT      #SELO,@DXMO          ;
1783 010752 001401          BEQ      .+4          ;BRANCH IF NO ERROR CONDITION
1784 010754 104000          ERROR          ;SELO SET
1785
1786 010756 052777 004000 170324  BIS      #ADRO,@DXMO          ;
1787
1788 010764 032777 004000 170316  BIT      #ADRO,@DXMO          ;
1789 010772 001001          BNE      .+4          ;BRANCH IF NO ERROR CONDITION
1790 010774 104000          ERROR          ;ADRO NOT SET
1791
1792 010776 042777 004000 170304  BIC      #ADRO,@DXMO          ;
1793
1794 011004 032777 004000 170276  BIT      #ADRO,@DXMO          ;
1795 011012 001401          BEQ      .+4          ;BRANCH IF NO ERROR CONDITION
1796 011014 104000          ERROR          ;ADRO SET
1797 011016 000207          HIO.2: RTS      PC
1798          .SBTTL  CUIS SUBROUTINE
1799
1800          ;
1801          ;CUIS.SUB, SUBROUTINE TO EXECUTE A
1802          ;CONTROL UNIT INITIATED SELECTION
1803          ;
1804          ;
1805 011020          CUIS.SUB:
1806
1807 011020 032737 000004 013602  BIT      #BIT2,PARA
1808 011026 001510          BEQ      CUIS0
1809
1810 011030          CUIS.GO:
1811
1812 011030 032777 020000 170254  BIT      #REQI,@DXMI          ;TEST FOR REQUEST-IN OP
1813 011036 001001          BNE      .+4          ;BRANCH IF NO ERROR CONDITION
1814 011040 104000          ERROR          ;REQI NOT SET
1815
1816 011042 052777 060000 170240  BIS      #SELO!HLDO,@DXMO     ;SET SELECT-OUT & HOLD-OUT
1817 011050 032777 020000 170232  BIT      #SELO,@DXMO          ;TEST FOR SELECT-OUT
1818 011056 001001          BNE      .+4          ;BRANCH IF NO ERROR CONDITION
1819 011060 104000          ERROR          ;SELO NOT SET
1820
1821 011062 032777 040000 170220  BIT      #HLDO,@DXMO          ;TEST FOR HOLD-OUT
1822 011070 001001          BNE      .+4          ;BRANCH IF NO ERROR CONDITION
1823 011072 104000          ERROR          ;HLDO NOT SET
1824
1825 011074 032777 100000 170210  BIT      #OPLI,@DXMI          ;TEST FOR OPERATIONAL-IN
1826 011102 001001          BNE      .+4          ;BRANCH IF NO ERROR CONDITION
1827 011104 104000          ERROR          ;OPLI NOT SET
1828
1829 011106 032777 020000 170176  BIT      #REQI,@DXMI          ;TEST FOR REQI
1830 011114 001401          BEQ      .+4          ;BRANCH IF NO ERROR CONDITION
1831 011116 104000          ERROR          ;REQI DID NOT DROP
1832
1833 011120 032777 010000 170164  BIT      #ADRI,@DXMI          ;TEST FOR ADDRESS IN
1834 011126 001001          BNE      .+4          ;BRANCH IF NO ERROR CONDITION
1835 011130 104000          ERROR          ;ADRI NOT SET

```

```

1836
1837 011132 123777 014214 170202      CMPB  DEV,@BUSI      ;ADDRESS ON BUSI
1838 011140 001401                      BEQ    .+4           ;BRANCH IF NO ERROR CONDITION
1839 011142 104000                      ERROR  ;BUSI LOAD ERROR
1840
1841 011144 052777 002000 170136      BIS   #CMDO,@DXMO    ;SET COMMAND-OUT
1842 011152 032777 002000 170130      BIT   #CMDO,@DXMO
1843 011160 001001                      BNE   .+4           ;BRANCH IF NO ERROR CONDITION
1844 011162 104000                      ERROR  ;CMDO DID NOT SET
1845
1846 011164 032777 010000 170120      BIT   #ADRI,@DXMI    ;ADRI SHOULD DROP
1847 011172 001401                      BEQ    .+4           ;BRANCH IF NO ERROR CONDITION
1848 011174 104000                      ERROR  ;ADRI DID NOT DROPS
1849
1850 011176 042777 060000 170104      BIC   #SELO,HLDO,@DXMO
1851 011204 032777 040000 170076      BIT   #HLDO,@DXMO    ;TEST HOLD-OUT
1852 011212 001401                      BEQ    .+4           ;BRANCH IF NO ERROR CONDITION
1853 011214 104000                      ERROR  ;HLDO SET
1854
1855 011216 032777 020000 170064      BIT   #SELO,@DXMO    ;TEST SELECT-OUT
1856 011224 001401                      BEQ    .+4           ;BRANCH IF NO ERROR CONDITION
1857 011226 104000                      ERROR  ;SELO SET
1858
1859 011230 042777 002000 170052      BIC   #CMDO,@DXMO    ;CLEAR COMMAND-OUT
1860 011236 032777 002000 170044      BIT   #CMDO,@DXMO    ;TEST FOR CMDO CLEAR
1861 011244 001401                      BEQ    .+4           ;BRANCH IF NO ERROR CONDITION
1862 011246 104000                      ERROR  ;CMDO SET
1863
1864 011250 000207      CUIS0: RTS      PC
1865
1866      ;FAST ISS SELECTOR CH ONLY
1867
1868      FASTISS:
1869 011252 053777 014214 170030      BIS   DEV,@DXMO      ;PUT DEVICE AORS ON OUT TAGS
1870 011260 052777 004000 170022      BIS   #ADRO,@DXMO    ;RAISE ADRS-OUT
1871 011266 052777 060000 170014      BIS   #HLDO!SELO,@DXMO ;RAISE SELECT-OUT, HOLD-OUT
1872 011274 042777 004000 170006      BIC   #ADRO,@DXMO    ;REMOVE ADRS-OUT
1873 011302 043777 014214 170000      BIC   DEV,@DXMO      ;REMOVE ADRS
1874 011310 053777 014216 167772      BIS   CMD,@DXMO      ;PUT COMMAND ON OUT TAGS
1875 011316 052777 002000 167764      BIS   #CMDO,@DXMO    ;RAISE CMD-OUT
1876 011324 043777 014216 167756      BIC   CMD,@DXMO      ;REMOVE CMD
1877 011332 042777 002000 167750      BIC   #CMDO,@DXMO    ;REMOVE CMD-OUT
1878 011340 032737 000004 013602      BIT   #BIT2,PARA     ;TEST FOR MUX CH
1879 011346 001006                      BNE   FISS1          ;BRANCH FI MUX
1880 011350 105737 014216                      TSTB  CMD            ;TEST FOR 'TEST I/O' COMMAND
1881 011354 001403                      BEQ   FISS1          ;CLEAR SELO,HLDO IF TIO CMD
1882 011356 105777 167760                      TSTB  @BUSI          ;TEST BUSI FOR ZERO STATUS
1883 011362 001403                      BEQ   FISS2          ;DON'T CLEAR SELO,HLDO ON 0 STATUS
1884 011364 042777 060000 167716      FISS1: BIC   #HLDO!SELO,@DXMO ;CLEAR SELO AND HLDO
1885 011372 052777 001000 167710      FISS2: BIS   #SRVO,@DXMO ;RELEASE STATUS
1886 011400 042777 001000 167702      BIC   #SRVO,@DXMO
1887 011406 000207      RTS      PC
1888
1889      ;SEL CH ISS
1890
1891 011410      SEL.ISS:
    
```

1892	011410	053777	014214	167710	BIS	DEV,@CUAR	:PRESET COM/ADD REG DEV ADDRESS
1893	011416	042777	000400	167702	BIC	#400,@CUAR	:PARITY RESET
1894	011424	053777	014214	167656	BIS	DEV,@DXMO	:PUT DEVICE AORS ON OUT TAGS
1895	011432	052777	004000	167650	BIS	#ADRO,@DXMO	:RAISE ADRS-OUT
1896	011440	052777	060000	167642	BIS	#HLDO!SELO,@DXMO	:RAISE SELECT-OUT, HOLD-OUT
1897	011446	042777	004000	167634	BIC	#ADRO,@DXMO	:REMOVE ADRS-OUT
1898	011454	043777	014214	167626	BIC	DEV,@DXMO	:REMOVE ADRS
1899	011462	053777	014216	167620	BIS	CMD,@DXMO	:PUT COMMAND ON OUT TAGS
1900	011470	052777	002000	167612	BIS	#CMDO,@DXMO	:RAISE CMD-OUT
1901	011476	043777	014216	167604	BIC	CMD,@DXMO	:REMOVE CMD
1902	011504	042777	002000	167576	BIC	#CMDO,@DXMO	:REMOVE CMD-OUT
1903	011512	052777	001000	167570	BIS	#SRVO,@DXMO	:RELEASE STATUS
1904	011520	042777	001000	167562	BIC	#SRVO,@DXMO	:
1905	011526	000207			RTS	PC	

:SCOPE LOOP AND CONTROL SUBROUTINE

1906							
1907							
1908	011530	105777	167642		SCOPEC:	TSTB @TKS	
1909	011534	100014			BPL	SCOPEH	
1910	011536	017727	167636		MOV	@TKB,(PC)+	
1911	011542	000000			DTMP:	0	
1912	011544	042737	000200	011542	BIC	#200,DTMP	
1913	011552	123727	011542	000003	CMPB	DTMP,#3	
1914	011560	001002			BNE	SCOPEH	
1915	011562	000137	012724		JMP	@MON1.0	
1916	011566	032737	040000	177570	SCOPEH:	BIT #BIT14,SR	:TEST FOR SCOPE
1917	011574	001012			BNE	SCOPEB	:BRANCH IF SCOPE SELECTED
1918	011576	032737	004000	177570	BIT	#BIT11,SR	:TEST FOR ITERATIONS
1919	011604	001020			BNE	SCOPEA	:EXIT IF ITERATIONS INHIBITED
1920	011606	005237	011656		INC	SCOPEF	:INCREMENT ITERATION COUNT
1921	011612	023737	011656	011654	CMP	SCOPEF,ICOUNT	:TEST FOR COMPLETION OF ITERATIONS
1922	011620	001410			BEQ	SCOPEG	:BRANCH IF COMPLETE
1923	011622	012737	177777	014206	SCOPEB:	MOV #-1,ONESHOT	:SO YOU CAN SCOPE ON ONCE ONLY CODE
1924	011630	005726			TST	(SP)+	:POP RETURN PC
1925	011632	012637	177776		MOV	(SP)+,PS	:RESTOR PROCESSOR STATUS
1926	011636	000177	000016		JMP	@RETURN	:
1927	011642	011637	011660		SCOPEG:	MOV @SP,RETURN	:SET UP SCOPE RETURN ADRS
1928	011646	005037	011656		SCOPEA:	CLR SCOPEF	:CLEAR ITERATION COUNT
1929	011652	000002			RTI		
1930	011654	000001			ICOUNT:	1	:NUMBER OF REQUESTED ITERATIONS
1931	011656	000000			SCOPEF:	0	:ITERATION COUNT
1932	011660	004000			RETURN:	TST1	:DEFAULT RETURN
1933	011662				TSTABLE:		:BEGINNING OF TABLE OF TEST ADDRESSES
1934		011762			.=.+100		:TEST ADDRESS LIST
1935							
1936							

:EMULATOR DECODER ROUTINE

1937							
1938							
1939	011762				EMTDECODER:		
1940	011762	011646			MOV	@R6,-(R6)	:DUPLICATE PC ON STACK
1941	011764	162716	000002		SUB	#2,@R6	:POINT PC TO EMT INST.
1942	011770	017616	000000		MOV	@(R6),@R6	:MOV EMT INST ONTO STACK
1943	011774	121627	000024		CMPB	@R6,#20.	:TEST THAT CALL IS WITHIN LIMITS
1944	012000	101401			BLOS	EMTOK	:BRANCH IF WITHIN LIMITS
1945	012002	104000			ERROR		
1946	012004	006116			EMTOK:	ROL @R6	:EMT ARGUMENT X 2.
1947	012006	042716	177001		BIC	#177001,@R6	:CLEAR HIGH BYTE

```

1948 012012 062716 012024      ADD    #EMTAG,@R6      ;FORM ADRS OF ROUTINE ADRS
1949 012016 017616 000000      MOV    @R6,@R6        ;PUT ROUTINE ADRS ON STACK
1950 012022 000136              JMP    @R6+           ;JUMP TO ROUTINE
1951                               ;TAGS FOR EMT CALL
1952
1953 012024              EMTAG:                ;BEGINNING OF EMT TABLE
1954
1955 012024 000024              .BLKW 20.            ;RESERVE 16. WORDS FOR ADRS LIST
1956
1957
1958                               ;ENTRY POINT FOR MAP ERRORS
1959 012074              T.MAPERR:
1960 012074 012737 177777 012602      MOV    #-1,ERFLG      ;FLAG THAT THIS IS MAP ERROR
1961 012102 000137 012244              JMP    @#OEF
1962 012106 000000              TERPC: 0              ;ORIGIN OF TRACE ERROR
1963                               ;TUMBLE TABLE TRACE ERROR TRAP
1964 012110              T.TRACER:
1965 012110 012737 177776 012602      MOV    #-2,ERFLG
1966 012116 000004 021571              TYPE   ,TRCM1
1967 012122 010546              MOV    TTY,-(SP)      ;SAVE TTY
1968 012124 013705 012636              MOV    ERTSTN,TTY     ;TYPE ERTSTN IN OCTAL
1969 012130 004737 017076              JSR    %7,PRINTS      ;AND SUPRESS LEADING ZERO'S
1970 012134 012605              MOV    (SP)+,TTY     ;RESTORE TTY
1971 012136 000004 021623              TYPE   ,TRCM         ;TRACE ERROR AT:
1972 012142 010546              MOV    TTY,-(SP)     ;SAVE TTY
1973 012144 013705 012106              MOV    TERPC,TTY     ;TYPE IN OCTAL
1974 012150 004737 017066              JSR    PC,PRINTR     ;TYPE LEADING ZERO'S
1975 012154 012605              MOV    (SP)+,TTY     ;RESTORE TTY
1976 012156 000004 021723              TYPE   ,TRC1
1977 012162 010546              MOV    TTY,-(SP)     ;SAVE TTY
1978 012164 013705 006534              MOV    TTWAS,TTY     ;TYPE IN OCTAL
1979 012170 004737 017066              JSR    PC,PRINTR     ;TYPE LEADING ZERO'S
1980 012174 012605              MOV    (SP)+,TTY     ;RESTORE TTY
1981 012176 000004 021745              TYPE   ,TRC2
1982 012202 010546              MOV    TTY,-(SP)     ;SAVE TTY
1983 012204 013705 006536              MOV    TTSHOULD,TTY  ;TYPE IN OCTAL
1984 012210 004737 017066              JSR    PC,PRINTR     ;TYPE LEADING ZERO'S
1985 012214 012605              MOV    (SP)+,TTY     ;RESTORE TTY
1986 012216 032701 000002              BIT    #BIT1,R1      ;TEST FOR DXDS OR DXCA
1987 012222 001003              BNE    1$            ;BR IF DXCA
1988 012224 000004 021767              TYPE   ,TTDS
1989 012230 000402              BR     2$
1990 012232 000004 022014              1$: TYPE   ,TTCA
1991 012236              2$:
1992 012236 000402              BR     OEF
1993
1994                               ;ENTRY POINT FOR MOST ERRORS
1995
1996 012240              T.ERROR:
1997 012240 005037 012602      CLR    ERFLG          ;FLAG THIS AS NORMAL ERROR
1998 012244 005237 012634      OEF: INC    ERRCNT    ;INC ERROR COUNT
1999 012250 010037 012640              MOV    R0,E.R0       ;SAVE R0
2000 012254 010137 012642              MOV    R1,E.R1       ;SAVE R1
2001 012260 010237 012644              MOV    R2,E.R2       ;SAVE R2
2002 012264 010337 012646              MOV    R3,E.R3       ;SAVE R3
2003 012270 010437 012650              MOV    R4,E.R4       ;SAVE R4

```

```

2004 012274 010537 012652      MOV      R5,E,R5      ;SAVE R5
2005 012300 032737 020000 177570  BIT      #BIT13,SR    ;TEST FOR INHIBIT PRINT
2006 012306 001063              BNE      PERRPC      ;BRANCH IF INHIBIED
2007 012310 000004 021305      TYPE     ,ERPC
2008 012314 011627              MOV      @SP,(PC)+    ;SAVE ERROR PC +2
2009 012316 000000              ETMPO:  0             ;HERE
2010 012320 162737 000002 012316  SUB      #2,ETMPO     ;CORRECT PC
2011 012326 010546              MOV      TTY,-(SP)    ;SAVE TTY
2012 012330 013705 012316      MOV      @#ETMPO,TTY  ;TYPE IN OCTAL
2013 012334 004737 017066      JSR      PC,PRINTR    ;TYPE LEADING ZERO'S
2014 012340 012605              MOV      (SP)+,TTY    ;RESTORE TTY
2015 012342 032737 010000 177570  BIT      #BIT12,SR    ;TEST FOR SHORT ERROR REPORT
2016 012350 001042              BNE      PERRPC      ;BRANCH IF SHORT SELECTED
2017 012352 022737 177776 012602  CMP      #-2,ERFLG
2018 012360 001410              BEQ      1$
2019 012362 000004 021521      TYPE     ,MSG35      ;ERROR IN TEST:
2020 012366 010546              MOV      TTY,-(SP)    ;SAVE TTY
2021 012370 013705 012636      MOV      ERTSTN,TTY  ;TYPE ERTSTN IN OCTAL
2022 012374 004737 017076      JSR      %7,PRINTS   ;AND SUPRESS LEADING ZERO'S
2023 012400 012605              MOV      (SP)+,TTY    ;RESTORE TTY
2024 012402 000004 021450      1$:      TYPE     ,MSG26
2025 012406 010546              MOV      TTY,-(SP)    ;SAVE TTY
2026 012410 013705 014224      MOV      CUADRS,TTY  ;TYPE IN OCTAL
2027 012414 004737 017066      JSR      PC,PRINTR    ;TYPE LEADING ZERO'S
2028 012420 012605              MOV      (SP)+,TTY    ;RESTORE TTY
2029 012422 000004 021254      TYPE     ,CRLF
2030 012426 032737 000002 013602  BIT      #BIT1,@#PARA ;TEST FOR BUSY ENABLE MODE
2031 012434 001402              BEQ      EMOD1
2032 012436 000004 021664      TYPE     ,ABSYM
2033 012442 032737 000004 013602  EMOD1:  BIT      #BIT2,@#PARA ;TEST FOR MUX CH
2034 012450 001402              BEQ      EMOD2
2035 012452 000004 021703      TYPE     ,AMUXM
2036 012456              EMOD2:
2037 012456 032737 100000 177570  PERRPC: BIT      #HLTSW,SR ;TEST FOR HALT ON ERROR
2038 012464 001401              BEQ      ERRLOP      ;BRANCH IF NO HALT
2039
2040
2041 012466 000000              ERRLOP: HALT          ;HALT ON ERROR
2042 012470
2043 012470 013700 012640      MOV      E.R0,R0     ;RESTORE R0
2044 012474 013701 012642      MOV      E.R1,R1     ;RESTORE R1
2045 012500 013702 012644      MOV      E.R2,R2     ;RESTORE R2
2046 012504 013703 012646      MOV      E.R3,R3     ;RESTORE R3
2047 012510 013704 012650      MOV      E.R4,R4     ;RESTORE R4
2048 012514 013705 012652      MOV      E.R5,R5     ;RESTORE R5
2049 012520 032737 040000 177570  BIT      #LOPSW,SR    ;TEST FOR SCOPE LOOP
2050 012526 001424              BEQ      EXTR1
2051 012530 012706 001100      MOV      #BEGIN,SP   ;REINIT STACK POINTER
2052 012534 012737 177777 014206  MOV      #-1,ONESHOT ;REINIT ONESHOT TEST FLAGS
2053 012542 042777 000002 166550  BIC      #MCLKEN,@DXES ;CLEAR MAINT CLOCK
2054 012550 005077 166534      CLR      @DXMO
2055 012554 052777 000001 166516  BIS      #DXFRS,@DXCS ;DX RESET
2056 012562 004737 015442      JSR      PC,PREI
2057 012566 013737 001272 177776  MOV      LESS1,PS
2058 012574 000177 177060      JMP      @RETURN
2059

```

```
2060 012600 000002      EXTR1: RTI
2061 012602 000000      ERFLG: 0          ;ERROR CONTROL FLAG -1=MAP ERROR
2062                    ;LIST OF ASCII MESSAGE ADDRESSES
2063
2064 012604 021322      ADRA:  ADXDS
2065 012606 021331          ADXCA
2066 012610 021340          ADXCS
2067 012612 021347          ADXOS
2068 012614 021356          ADXBA
2069 012616 021365          ADXBC
2070 012620 021374          ADXMO
2071 012622 021403          ADXMI
2072 012624 021412          ADXCB
2073 012626 021421          ADXND
2074 012630 021430          ADXES
2075 012632 021437      ADRAE:  ADXES1
2076
2077
2078
2079                    ;ERROR COUNT
2080
2081 012634 000000      ERRCNT: 0
2082 012636 000000      ERTSTN: 0          ;TEST NUMBER
2083
2084                    ;REGISTER STORAGE FOR ERROR REPORTING
2085
2086
2087 012640 000000      E.R0:  0          ;SAVED REGISTERS FOR ERROR REPORTING
2088 012642 000000      E.R1:  0          ;SAVED REGISTERS FOR ERROR REPORTING
2089 012644 000000      E.R2:  0          ;SAVED REGISTERS FOR ERROR REPORTING
2090 012646 000000      E.R3:  0          ;SAVED REGISTERS FOR ERROR REPORTING
2091 012650 000000      E.R4:  0          ;SAVED REGISTERS FOR ERROR REPORTING
2092 012652 000000      E.R5:  0          ;SAVED REGISTERS FOR ERROR REPORTING
2093
2094                    .SBTTL  MONITOR
2095
2096                    ; *****
2097 012654      MONITOR:
2098                    ; *****
2099
2100
2101
2102 012654 012706 001100      MOV      #BEGIN,SP      ;SET UP STACK POINTER
2103 012660 012737 000340 177776      MOV      #LEVEL7,PS     ;MONITOR AT LEVEL 7
2104 012666 004737 014520      JSR      PC,@#MONDFLT   ;SET UP DEFAULT PARAMETERS
2105
2106 012672 012737 017224 000060      MOV      #TTYI,@#60     ;TTY KEYBOARD INT VEC
2107 012700 012737 000200 000062      MOV      #LEVEL4,@#62  ;LEVEL 4
2108 012706 000004 021175          TYPE     ,HOME          ;HOME UP AND ERASE SCREEN
2109
2110 012712 000004 020074          H:      TYPE     ,HEADER
2111 012716 012737 013040 012714      MOV      #RELOD,@#H+2  ;HEADER TEXT GETS WIPED BY NPR'S
2112
2113 012724 000005      MON1.0: RESET
2114 012726 005777 166446          TST     @TKB           ;CLEAR FLAG
2115 012732 052777 000100 166436      BIS     #INTEN,@TKS    ;SET INTERRUPT ENABLE
```



```

2116 012740 012706 001100      MOV      #BEGIN,SP      ;SET UP STACK POINTER
2117 012744 012737 000340 177776  MOV      #LEVEL7,PS     ;MONITOR AT LEVEL 7
2118 012752 000004 020615      TYPE     ,FSTART
2119 012756 104006      KEY.TO.R0
2120 012760 122700 000104      CMPB    #'D,R0         ;D = DEFAULT PARAMETERS
2121 012764 001003      BNE     1$
2122 012766 004737 014520      JSR     PC,@#MONDFLT
2123 012772 000403      BR      2$
2124 012774 122700 000120      1$:    CMPB    #'P,R0         ;P = PREVIOUSLY SELECTED PARAMETERS
2125 013000 001002      BNE     3$
2126 013002 000137 013544      2$:    JMP     @#MON10
2127 013006 122700 000123      3$:    CMPB    #'S,R0         ;S = GO THROUGH AND SELECT PARAMETERS
2128 013012 001426      BEQ     MON1
2129 013014 122700 000116      CMPB    #'N,R0         ;N = START AT THIS TEST #
2130 013020 001341      BNE     MON1.0
2131 013022 000004 020673      TYPE     ,MSG5
2132 013026 104005      ACCEPTO
2133 013030 013737 014232 014234  MOV      OCTNUM,FIRST.TST
2134 013036 000761      BR      2$
2135
2136 013040 051137 046105 040517  RELOD:  .ASCIZ  ''_RELOAD FOR HEADER TEXT''
2137 013046 020104 047506 020122
2138 013054 042510 042101 051105
2139 013062 052040 054105 000124
2140
2141      .EVEN
2142      ;SET UP TEST PARAMETERS
2143 013070 004737 014520      MON1:   JSR     PC,MONDFLT     ;SET UP DEFAULT PARAMETERS
2144 013074 000004 020673      TYPE     ,MSG5         ;FIRST TEST #
2145
2146 013100 104005      ACCEPTO ;ACCEPT TEST NUMBER FROM KEYBOARD
2147
2148
2149 013102 005737 014232      TST     OCTNUM         ;TEST FOR DEFAULT
2150 013106 001403      BEQ     MON3           ;BRANCH ON DEFAULT
2151 013110 013737 014232 014234  MOV      OCTNUM,FIRST.TST ;LOAD FIRST TEST #
2152
2153 013116 000004 020631      MON3:   TYPE     ,MSG2     ;BASE ADDRESS:
2154
2155
2156 013122 104005      ACCEPTO ;ACCEPT BASE ADDRESS FROM KEYBOARD
2157
2158 013124 005737 014232      TST     OCTNUM         ;TEST FOR DEFAULT
2159 013130 001403      BEQ     MON4           ;BRANCH IF DEFAULT
2160 013132 013737 014232 001262  MOV      OCTNUM,DXBASE  ;LOAD NON-DEFAULT ADDRESS
2161
2162 013140 000004 021465      MON4:   TYPE     ,MSG28    ;ACCEPT INTERRUPT VECTOR
2163 013144 104005      ACCEPTO
2164 013146 005737 014232      TST     OCTNUM         ;TEST FOR DEFAULT
2165 013152 001411      BEQ     MON4.1        ;BRANCH IF DEFAULT
2166 013154 013737 014232 001264  MOV      OCTNUM,DXIV    ;LOAD NON-DEFAULT INT VECTOR ADRS
2167 013162 062737 000002 014232  ADD      #2,@#OCTNUM    ;FORM INT STATUS ADRS
2168 013170 013737 014232 001266  MOV      @#OCTNUM,@#DXIS ;INT STATUS ADDRESS
2169
2170
2171 013176 000004 021076      MON4.1: TYPE     ,MSG12    ;PRIORITY

```

2172 013202 104005
2173 013204 005737 014232
2174 013210 001425
2175 013212 006337 014232
2176 013216 006337 014232
2177 013222 006337 014232
2178 013226 006337 014232
2179 013232 006337 014232
2180 013236 013737 014232 001270
2181 013244 005337 014232
2182 013250 042737 000037 014232
2183 013256 013737 014232 001272
2184
2185
2186
2187
2188
2189 013264 000004 020712
2190 013270 012703 014254
2191 013274 000004 021004
2192 013300 004737 015752
2193 013304 104007 016410
2194 013310 013723 016410
2195 013314 104006
2196 013316 122700 000015
2197 013322 001364
2198
2199
2200
2201
2202 013324 013727 016410
2203 013330 000000
2204 013332 042737 000400 013330
2205 013340 023727 013330 000376
2206 013346 003403
2207 013350 000004 021123
2208 013354 000747
2209 013356 012723 177777
2210
2211
2212
2213
2214
2215
2216 013362 000004 020651
2217 013366 104005
2218 013370 005737 014232
2219 013374 001003
2220 013376 000004 021123
2221 013402 000767
2222
2223 013404 013727 014232
2224 013410 000017
2225 013412 005337 013410
2226 013416 013727 013330
2227 013422 000000

ACCEPTO ;ACCEPT DX PRIORITY LEVEL
TST OCTNUM ;TEST FOR DEFAULT
BEQ MON6 ;BRANCH ON DEFAULT
ASL OCTNUM ;SHIFT PRIORITY
ASL OCTNUM ;INTO PROCESSOR
ASL OCTNUM ;PRIORITY BITS OF
ASL OCTNUM ;PROCESSOR STATUS WORD
MOV OCTNUM,DXPRT ;LOAD PRIORITY
DEC OCTNUM
BIC #37,@OCTNUM ;CLEAR TNZVC
MOV OCTNUM,LESS1 ;PRIORITY TO ALLOW DX INTERRUPTS

;GENERATE A LIST OF LEGAL ADDRESSES

MON6: TYPE ,MSG4 ;LEGAL ADDRESS LIST
MOV #LEGAL.ADRS,R3 ;START OF LEGAL ADRS TABLE
MON7: TYPE ,MSG6 ;ADRS:
JSR PC,GETHEX ;GET HEXADECIMAL CU ADDRESS
PARITY ,HEXNUM ;PUT PARITY (ODD) ON ADRS
MOV HEXNUM,(R3)+ ;SAVE LEGAL ADDRESS
KEY.TO.R0
CMPB #CR,R0 ;ALL DONE?
BNE MON7 ;CONTINUE LIST IF NOT <CR>

***** MOD APR 74 *****

;*
;* ADDRESS RESPONSE MOD
;*

MOV HEXNUM,(PC)+
VLUHEX:0
BIC #400,VLUHEX
CMP VLUHEX,#376 ;TEST FOR > FF
BLE 1\$;(OK) BRANCHES
TYPE, MSG13 ;OUTPUT '=' ILLEGAL ?' I.E. > 'FF'
BR MON7 ;TRY AGAIN
1\$: MOV #-1,(R3)+ ;MARK END OF LIST

***** MOD APR 74 *****

;*
;* #DEV/CU MOD
;*

;SET UP MAXIMUM NUMBER OF DEVICES PER CONTROL UNIT
;THIS INFORMATION DETERMINES WHAT THE SPW TABLE LOOKS LIKE

MON5: TYPE ,MSG3 ;MAX # DEVICES/CU
ACCEPTO ;ACCEPT NUMBER OF DEVICES/CU
TST OCTNUM ;USE 16- ON DEFAULT
BNE X1\$
TYPE, MSG13 ;OUTPUT '=' ILLEGAL ?' I.E. = '00'
BR MON5 ;TRY AGAIN

X1\$: MOV OCTNUM,(PC)+
RDXX: 17 ;THIS IS THE DEFAULT VALUE..GETS WIPED BY CONVERSATION
DEC RDXX ;RANGE MODULO 1
MOV VLUHEX,(PC)+
MDXX: 0 ;RANGE MASK

```
2228 013424 063737 013410 013422      ADD    RDXX,MDXX      ;SCALE
2229 013432 105137 013422      COMB   MDXX          ;FORM FINAL
2230 013436 042737 177400 013422      BIC    #177400,MDXX  ;CU PORTION CLR
2231
2232      ;***** MOD APR 74 *****
2233      ;
2234 013444 013737 014232 014236      MOV    OCTNUM,MAX.DEV.CU
2235 013452 004737 015334      MON5.1: JSR   PC,CKCUA      ;CHECK FOR LEGAL NUMBER OF DEV PER CU
2236
2237
2238      ;GET COMMAND LIST
2239
2240      .REM    *
2241
2242      THIS ROUTINE ACCEPTS AN IBM COMMAND LIST FROM THE CONSOL. ALL
2243      COMMANDS MUST BE NON ZERO (I.E. T10 MUST BE TYPED WITH PARITY
2244      400). WITH EACH COMMAND THE MONITOR ASKS FOR ITS ASSOCIATED DST
2245      STATUS.
2246
2247      *
2248
2249 013456 012704 014500      MON8:  MOV    #CMD.STAT,R4
2250 013462 012703 014440      MOV    #CMD.ADRS,R3
2251 013466 000004 021033      TYPE   ,MSG9          ;LEGAL CMD LIST
2252 013472 000004 021064      MON9:  TYPE   ,MSG10      ;CMD:
2253 013476 104005      ACCEPTO      ;ACCEPT LEGAL COMMANDS FROM KEYBOARD
2254 013500 005737 014232      TST    OCTNUM
2255 013504 001417      BEQ    MON10
2256 013506 104007 014232      PARITY ,OCTNUM
2257 013512 013723 014232      MOV    OCTNUM,(R3)+
2258 013516 000004 021507      TYPE   ,MSG31      ;'STATUS: '
2259 013522 104005      ACCEPTO
2260 013524 113724 014232      MOV    OCTNUM,(R4)+
2261 013530 104006      KEY.TO.R0
2262 013532 120027 000015      CMPB   R0,#CR
2263 013536 001355      BNE    MON9
2264 013540 012723 177777      MOV    #-1,(R3)+      ;LOAD TERMINATOR
2265
2266
2267      ;ASK FOR DYNAMIC SWITCH SETTINGS ON CONSOL SWITCHES
2268
2269 013544      MON10:
2270 013544 012737 177777 014206      MOV    #-1,ONESHOT
2271 013552 005037 012634      CLR    ERRCNT
2272 013556 000004 021014      TYPE   ,MSG7          ;SET DYNAMIC SWITCHES
2273 013562 104006      KEY.TO.R0      ;TYPE ANYTHING
2274 013564 122700 000003      CMPB   #3,R0      ;TEST FOR CONTROL C
2275 013570 001002      BNE    MON11      ;GO IF NO ^C
2276 013572 000137 012724      JMP    @MON1.0
2277
2278
2279      ;SET UP TABLES
2280
2281 013576 113727 177570      MON11: MOV    SWR,(PC)+      ;SAVE MODE CONTROL SWITCH SETTINGS
2282 013602 000000      PARA:  0          ;HERE
2283 013604 052737 000001 013602      BIS    #BIT0,@PARA      ;SET ONLINE SELECT FLAG
```

```
2284 013612 004737 015334 JSR PC,CKCUA ;CMP ADRS VS MAX DEV PER CU
2285 013616 004737 015146 JSR PC,SPW.SETUP ;SET UP STATUS POINTER WORDS
2286 013622 004737 014772 JSR PC,TT.CLR ;CLEAR TUMBLE TABLE
2287 013626 004737 015304 JSR PC,DST.SETUP ;SETUP DEVICE STATUS TABLE
2288 013632 004737 015010 JSR PC,ODAT ;SET 360 SIM OUTPUT DATA FILE
2289 013636 004737 015646 JSR PC,REG.SETUP ;SCALE ADDRESSES
2290
2291 013642 LPCSU:
2292 013642 012737 000001 014226 MOV #1,DEV CNT ;INIT DEVICE COUNT
2293 013650 012737 014254 014222 MOV #LEGAL.ADRS,ACUA ;ADRS OF CU ADRS
2294 013656 117737 000340 014224 MOVB @ACUA,CUADRS ;CU ADDRESS
2295 013664 000472 BR LPC1
2296
2297 013666 000004 021200 LPCNTL: TYPE ,BELL
2298
2299 013672 062737 000001 014224 ADD #1,CUADRS ;
2300 013700 005237 014226 INC DEV CNT ;INC DEVICE COUNT
2301 013704 023737 014226 014236 CMP DEV CNT,MAX.DEV.CU
2302 013712 003457 BLE LPC1
2303 013714 012737 000001 014226 MOV #1,DEV CNT ;INIT DEVICE COUNT
2304 013722 027727 000274 177777 CMP @ACUA,#-1
2305 013730 001042 BNE LPC2
2306 013732 062737 000002 013602 ADD #2,PARA ;INC TO NEXT PARA COMBINATION
2307 013740 022737 000010 013602 CMP #10,PARA ;HAVE ALL PARA COMBS BEEN TESTED?
2308 013746 002030 BGE LPC5 ;BRANCH IF NOT
2309 013750 042737 177776 013602 BIC #177776,@#PARA ;ALL BUT ONLINE IF SET
2310
2311 ***** MOD APR 74 *****
2312 :
2313 :
2314 : OPLI TIMEOUT RESET MOD
2315 013756 012777 000001 165314 MOV #1,@DXCS ;DX RESET OPLI
2316
2317 ***** MOD APR 74 *****
2318 :
2319 TYPE ,ENDTST
2320 013764 000004 021202 TYPE ,ECM ;ERROR COUNT MESSAGE
2321 013770 000004 021214 MOV TTY,-(SP) ;SAVE TTY
2322 013774 010546 MOV ERR CNT,TTY ;TYPE IN OCTAL
2323 014002 004737 012634 JSR PC,PRINTR ;TYPE LEADING ZERO'S
2324 014006 012605 MOV (SP)+,TTY ;RESTORE TTY
2325
2326 ;THE FOLLOWING CODE IS FOR INTERFACE WITH DDP AND ACT11
2327
2328 014010 013700 000042 MOV @#42,R0 ;IF 42 = 0 REMAIN IN DX DIAGNOSTIC
2329 014014 001405 BEQ LPC5
2330 014016 000005 RESET ;LINK TO DDP OR ACT11
2331 014020 LOGICAL:
2332 014020 004710 JSR PC,@R0
2333 014022 000240 NOP
2334 014024 000240 NOP
2335 014026 000240 NOP
2336 014030 012737 014254 014222 LPC5: MOV #LEGAL.ADRS,ACUA
2337 014036 017737 000160 014224 LPC2: MOV @*CUA,CUADRS
2338 014044 062737 000002 014222 ADD #2,ACUA
2339 014052 104007 014224 LPC1: PARITY ,CUADRS
```

```
2340 014056 013737 014224 014214      MOV      CUADRS,DEV
2341 014064 013737 014214 014220      MOV      DEV,DEV.A      ;MULTI THREAD
2342 014072 023737 014226 014236      CMP      DEVCNT,MAX.DEV.CU
2343 014100 001404                BEQ      LPC3
2344 014102 062737 000001 014220      ADD      #1,DEV.A
2345 014110 000403                BR       LPC4
2346 014112 162737 000001 014220      LPC3:   SUB      #1,DEV.A
2347 014120 104007 014220      LPC4:   PARITY  ,DEV.A
2348
2349
2350 014124 004737 015442                MON12:  JSR      PC,PREI      ;DO PRE INIT
2351 014130 012777 006114 165126      MOV      #FALSE,@DXIV    ;SET UP FALSE INTERRUPT VECTOR TRAP
2352 014136 013777 001270 165122      MOV      DXPRT,@DXIS    ;SET UP INTERRUPT PRIORITY
2353 014144 013700 014234                MOV      FIRST.TST,RO   ;TEST FOR DEFAULT
2354 014150 001002                BNE      MON13          ;BRANCH IF NOT DEFAULT
2355 014152 005237 014234                INC      FIRST.TST      ;DEFAULT TEST NUMBER IS ONE
2356 014156 013737 014234 012636      MON13:  MOV      FIRST.TST,ERTSTN
2357 014164 006300                ASL      RO
2358 014166 016037 011660 011660      MOV      TSTABLE-2(RO),@#RETURN
2359 014174 062737 000024 011660      ADD      #24,@#RETURN
2360 014202 000170 011660      MON14:  JMP      @TSTABLE-2(RO) ;JUMP TO SELECTED TEST
2361
2362
2363      .SBTTL  MONITOR FILES
2364
2365      ; ONE PASS FLAGS
2366
2367      FIVESEC=1                ;5 SEC OPLI TIMER TEST
2368 014206 177777                ONESHOT:  -1            ;ONE PASS FLAGS
2369 014210 000000                CARRY:   0              ;CARRY COUNT
2370 014212 000000                TMP:     0              ;TEMPORARY STORAGE
2371 014214 000020                DEV:     20             ;DEVICE ADDRESS TO SELECT - MUST INCLDE PARITY
2372                                ; ( I.E. 441 IS DEV=1, CU=2)
2373 014216 000403                CMD:     403           ;COMMAND TO PRESET - MUST INCLUDE PARITY
2374                                ; (403 IS BASIC NOP COMMAND)
2375
2376 014220 000421                DEV.A:   421           ;SECOND DEVICE FOR DUAL TESTS
2377
2378
2379 014222 000000                ACUA:    0              ;ADRS OF CU ADRS
2380
2381
2382 014224 000000                CUADRS:  0              ;CU ADRS
2383 014226 000000                DEVCNT:  0              ;DEVICE COUNT
2384
2385 014230 002000                OFFSET: 2000           ;OFFSET TO ADDRESS REGISTER
2386 014232 000000                OCTNUM:   0              ;OCTAL INPUT FROM TTY
2387 014234 000000                FIRST.TST: 0              ;FIRST TEST TO RUN
2388 014236 000000                MAX.DEV.CU: 0           ;MAXIMNM # OF DEVICES/CU
2389
2390      ;DIAGNOSTIC VARIABLES
2391
2392 014240 000777                SSTAT:   777           ;SAVED STATUS
2393 014242 000000                SRCNT:   0              ;SOURCE DATA
2394 014244 000000                DSTCNT:  0              ;DESTINATION DATA
2395 014246 000000                SAVDEV:  0              ;SAVED DEVICE ADDRESS
```

2396 014250 000000 TSSFT: 0 ;TSSF TRACE
2397 014252 000000 COUNT: 0 ;USED BY CH SIM TO COUNT BYTES TRANSFFERED
2398
2399

2400 ;LEGAL ADDRESS LIST
2401

2402 014254 LEGAL.ADRS:
2403

2404	014254	000000	.WORD	0
2405	014256	000000	.WORD	0
2406	014260	000000	.WORD	0
2407	014262	000000	.WORD	0
2408	014264	000000	.WORD	0
2409	014266	000000	.WORD	0
2410	014270	000000	.WORD	0
2411	014272	000000	.WORD	0
2412	014274	000000	.WORD	0
2413	014276	000000	.WORD	0
2414	014300	000000	.WORD	0
2415	014302	000000	.WORD	0
2416	014304	000000	.WORD	0
2417	014306	000000	.WORD	0
2418	014310	000000	.WORD	0
2419	014312	000000	.WORD	0
2420	014314	000000	.WORD	0

2421
2422 014316 SCALD.ADRS:
2423

2424	014316	000000	.WORD	0
2425	014320	000000	.WORD	0
2426	014322	000000	.WORD	0
2427	014324	000000	.WORD	0
2428	014326	000000	.WORD	0
2429	014330	000000	.WORD	0
2430	014332	000000	.WORD	0
2431	014334	000000	.WORD	0
2432	014336	000000	.WORD	0
2433	014340	000000	.WORD	0
2434	014342	000000	.WORD	0
2435	014344	000000	.WORD	0
2436	014346	000000	.WORD	0
2437	014350	000000	.WORD	0
2438	014352	000000	.WORD	0
2439	014354	000000	.WORD	0
2440	014356	000000	.WORD	0

2441
2442
2443 ;LIST OF DEFAULT COMMANDS
2444

2445 014360 DFLT.CMD:
2446

2447	014360	000400	TIOC	;TEST I/O COMMAND
2448	014362	000001	WRITEC	;WRITE COMMAND
2449	014364	000002	READC	;READ COMMAND
2450	014366	000403	NOPC	;NOP COMMAND
2451	014370	000004	SENSEC	;SENSE COMMAND

2452	014372	000405	ILLC	:ILLAGAL COMMAND
2453	014374	177777	-1	:LIST TERMINATOR
2454	014376	177777	-1	:LIST TERMINATOR
2455	014400	177777	-1	:LIST TERMINATOR
2456	014402	177777	-1	:LIST TERMINATOR
2457	014404	177777	-1	:LIST TERMINATOR
2458	014406	177777	-1	:LIST TERMINATOR
2459	014410	177777	-1	:LIST TERMINATOR
2460	014412	177777	-1	:LIST TERMINATOR
2461	014414	177777	-1	:LIST TERMINATOR
2462	014416	177777	-1	:LIST TERMINATOR

2463
2464
2465
2466
2467
2468

:DEFAULT STATUS LIST

2470	014420		DFLT.STAT:	
2471	014420	000	.BYTE	0
2472	014421	000	.BYTE	0
2473	014422	000	.BYTE	0
2474	014423	014	.BYTE	CE!DE
2475	014424	000	.BYTE	0
2476	014425	002	.BYTE	UC
2477	014426	002	.BYTE	UC
2478	014427	002	.BYTE	UC
2479	014430	002	.BYTE	UC
2480	014431	002	.BYTE	UC
2481	014432	002	.BYTE	UC
2482	014433	002	.BYTE	UC
2483	014434	002	.BYTE	UC
2484	014435	002	.BYTE	UC
2485	014436	002	.BYTE	UC
2486	014437	002	.BYTE	UC

2487
2488
2489
2490
2491

:LIST OF LEGAL COMMANDS

2492	014440		CMD.ADRS:	
2493				
2494	014440	000000	.WORD	0
2495	014442	000000	.WORD	0
2496	014444	000000	.WORD	0
2497	014446	000000	.WORD	0
2498	014450	000000	.WORD	0
2499	014452	000000	.WORD	0
2500	014454	000000	.WORD	0
2501	014456	000000	.WORD	0
2502	014460	000000	.WORD	0
2503	014462	000000	.WORD	0
2504	014464	000000	.WORD	0
2505	014466	000000	.WORD	0
2506	014470	000000	.WORD	0
2507	014472	000000	.WORD	0

```
2508 014474 000000 .WORD 0
2509 014476 000000 .WORD 0
2510
2511 ;COMMAND STATUS
2512
2513 014500 CMD.STAT:
2514
2515 014500 000 .BYTE 0
2516 014501 000 .BYTE 0
2517 014502 000 .BYTE 0
2518 014503 000 .BYTE 0
2519 014504 000 .BYTE 0
2520 014505 000 .BYTE 0
2521 014506 000 .BYTE 0
2522 014507 000 .BYTE 0
2523 014510 000 .BYTE 0
2524 014511 000 .BYTE 0
2525 014512 000 .BYTE 0
2526 014513 000 .BYTE 0
2527 014514 000 .BYTE 0
2528 014515 000 .BYTE 0
2529 014516 000 .BYTE 0
2530 014517 000 .BYTE 0
2531
2532
2533
2534
2535
2536
```

;SET UP DEFAULT PARAMETERS

```
2537 014520 MONDFLT:
2538 014520 005037 006366 CLR @WENTRY1 ;TT TRACE ENTRY1
2539 014524 005037 006426 CLR @WENTRY2 ;TT TRACE ENTRY2
2540 014530 013737 001440 006526 MOV @WTT,@WTRACE ;INIT TT TRACE
2541 014536 012737 177777 014206 MOV #-1,ONESHOT ;ONE PASS FLAGS
2542 014544 005037 014234 CLR FIRST.TST ;DEFAULT TEST #
2543 014550 005037 012636 CLR ERTSTN ;ERROR TEST NUMBER
2544 014554 005037 012634 CLR ERRCNT ;ERROR COUNT
2545 014560 013737 014752 011660 MOV @WD.FIRST.TST,@WRETURN ;FIRST TEST
2546 014566 013737 014754 001262 MOV @WD.DXBASE,@WDXBASE ;BASE ADDRESS
2547 014574 013737 014756 001264 MOV @WD.DXIV,@WDXIV ;INT VECTOR ADRS
2548 014602 013737 014770 001266 MOV @WD.DXIS,@WDXIS ;INT STATUS ADRS
2549 014610 013737 014760 001270 MOV @WD.DXPRT,@WDXPRT ;PRIORITY LEVEL
2550 014616 013737 014762 014236 MOV @WD.MAX.DEV.CU,@WMAX.DEV.CU ;MAX DEVICES
2551 014624 013737 014766 014220 MOV @WD.DEV.A,@WDEV.A ;SECOND DEVICE
2552 014632 013737 014764 014254 MOV @WD.LEGAL.ADRS,@WLEGAL.ADRS ;CU ADRS
2553 014640 012737 177777 014256 MOV #-1,LEGAL.ADRS+2
2554 014646 012737 017400 001442 MOV #DSTADRS,@WDST ;INIT DST ADRS
2555 014654 012700 014360 MOV #DFLT.CMD,R0 ;ADRS OF DEFAULT CMD LIST
2556 014660 012701 014440 MOV #CMD.ADRS,R1 ;ADRS OF LEGAL CMD LIST
2557
2558 014664 012021 MON2: MOV (R0)+,(R1)+ ;LOAD DEFAULT CMD LIST
2559 014666 022710 177777 CMP #-1,@R0 ;TEST FOR TERMINATOR
2560 014672 001374 BNE MON2
2561 014674 012721 177777 MOV #-1,(R1)+ ;LOAD TERMINATOR
2562 014700 012727 000020 MOV #16.,(PC)+
2563 014704 000000 MON2.0: 0
```



```

2564 014706 012700 014420          MOV    #DFLT.STAT,R0    ;DEFAULT STATUS
2565 014712 012701 014500          MOV    #CMD.STAT,R1    ;STATUS FOR EACH COMMAND
2566 014716 112021                   MON2.1: MOVB  (R0)+,(R1)+
2567 014720 005337 014704          DEC    MON2.0
2568 014724 001374                   BNE    MON2.1
2569
2570 014726 013737 014230 001436    MOV    OFFSET,SPW      ;LOAD ADRS OF SPW
2571 014734 013737 014230 001440    MOV    OFFSET,TT      ;LOAD ADRS OF TT
2572 014742 062737 001000 001440    ADD    #1000,TT       ;
2573 014750 000207                   MON2.3: RTS    PC
2574                                     ;DEFAULT PARAMETERS
2575
2576 014752 004000                   D.FIRST.TST:  TST1      ;FIRST TEST
2577 014754 176200                   D.DXBASE:    176200    ;BASE ADDRESS
2578 014756 000300                   D.DXIV:      300      ;INT VECTOR ADRS
2579 014760 000200                   D.DXPRT:    LEVEL4   ;DX PRIORITY LEVEL
2580 014762 000020                   D.MAX.DEV.CU: 20      ;MAX # DEVICES PER CU
2581 014764 000020                   D.LEGAL.ADRS: 020    ;DEFAULT CU ADRS
2582 014766 000421                   D.DEV.A:     421
2583 014770 000302                   D.DXIS:     302      ;INT STATUS ADRS
2584
2585
2586                                     .SBTTL MONITOR SUBROUTINES
2587
2588                                     ;TT,CLR, CLEAR TUMBLE TABLE
2589
2590
2591 014772                                     TT.CLR:
2592 014772 013701 001440          CL1:  MOV    TT,R1      ;BOTTOM OF TT
2593 014776 005021                   CLR    (R1)+           ;CLEAR TT
2594 015000 020127 004000          CMP    R1,#ENDTT     ;TEST FOR END OF TT
2595 015004 001374                   BNE    CL1            ;BRANCH IF NOT END
2596 015006 000207                   RTS    PC
2597
2598
2599                                     ;REINIT OUTPUT DATA
2600
2601
2602 015010                                     ODAT:
2603 015010 010046                   MOV    RO,-(SP)
2604 015012 012700 022442          MOV    #WDATA,R0
2605 015016 005027                   CLR    (PC)+
2606 015020 000000                   ODAT1: 0
2607 015022 104007 015020          ODAT2: PARITY ,ODAT1
2608 015026 013720 015020          MOV    ODAT1,(R0)+
2609 015032 042737 000400 015020    BIC    #PARO,ODAT1
2610 015040 105237 015020          INCB  ODAT1
2611 015044 001366                   BNE    ODAT2
2612 015046 012600                   MOV    (SP)+,R0
2613 015050 000207                   RTS    PC
2614
2615
2616
2617
2618
2619

```

2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675

015052
015052 017627 000000
015056 000000
015060 017727 177772
015064 000000
015066 005027
015070 000000
015072 106337 015064
015076 102002
015100 005137 015070
015104 106337 015064
015110 001370
015112 005737 015070
015116 100404
015120 052777 000400 177730
015126 000403
015130 042777 000400 177720
015136 062716 000002
015142 000002
015144 177400
015146
015146 012700 014254
015152 012701 014316
015156 012011
015160 043711 015144
015164 006311
015166 063721 001436
015172 021027 177777
015176 001367
015200 012721 177777
015204 013701 001436

T.PARITY:
SDAPG: 0 MOV @ (SP), (PC)+ ;FETCH ADDRESS OF SOURCE DATA
;SOURCE DATA ADDRESS
TDAT: 0 MOV @SDAPG, (PC)+ ;FETCH SOURCE DATA
;SOURCE DATA
PRTY: 0 CLR (PC)+
PG2: ASLB TDAT
BVC PG3
COM PRTY
PG3: ASLB TDAT
BNE PG2
TST PRTY
BMI PG4
BIS #PARO, @SDAPG ;SET PARITY BIT
BR PG5
PG4: BIC #PARO, @SDAPG ;CLEAR PARITY BIT
PG5: ADD #2, @SP ;ADD 2 TO RETURN PC
RTI
;SETUP SPW TABLE
;***** MOD APR 74 *****
;* ADDRESS RESOLUTION MOD
MARK: 177400
SPW.SETUP:
MOV #LEGAL.ADRS, R0 ;FETCH ADRS OF LEGAL ADRS LINST
MOV #SCALD.ADRS, R1 ;FETCH ADRS OF SCALED LEGAL ADRS LIST
SP.0: MOV (R0)+, @R1 ;MAKE DUPLICATE ADRS LIST
BIC MARK, @R1
;***** MOD APR 74 *****
ASL @R1 ;MAKE INDX MOD(2)
ADD SPW, (R1)+ ;EQUALS REAL SPW ADRS
CMP @R0, #-1 ;TEST FOR TERMINATION
BNE SP.0 ;FETCH NEXT ADRS
MOV #-1, (R1)+ ;MARK END OF SCALED ADRS LIST
MOV SPW, R1 ;ADRS OF SPW

```
2676 015210 013702 001442      MOV     DST,R2      ;ADRS OF DST
2677 015214 012700 014316      SP.1:  MOV     #SCALD.ADRS,R0
2678 015220 020110              SP.2:  CMP     R1,@R0   ;RUN THRU LIST
2679 015222 001407              BEQ     SP.3        ;BRANCH ON LEGAL ADRS
2680 015224 005720              TST     (R0)+
2681 015226 022710 177777      CMP     #-1,@R0    ;TEST FOR END OF LIST
2682 015232 001372              BNE     SP.2        ;BRANCH IF NOT ENT
2683 015234 012721 016400      MOV     #ERRDST,(R1)+ ;LOAD SPW WITH ERROR DST ADRS
2684 015240 000407              BR      SP.6
2685 015242 013727 014236      SP.3:  MOV     MAX.DEV.CU,(PC)+
2686 015246 000000      SP.4:  0
2687 015250 010221      SP.5:  MOV     R2,(R1)+
2688 015252 005337 015246      DEC     SP.4
2689 015256 001374              BNE     SP.5
2690 015260 020137 001440      SP.6:  CMP     R1,TT     ;TEST FOR END OF SPW
2691 015264 002753              BLT     SP.1
2692 015266 001405              BEQ     SP.7
2693 015270 005726              TST     (SP)+      ;POP STACK
2694 015272 000004 015404      TYPE   ,IDVN
2695 015276 000137 012724      JMP     MON1.0     ;GO BACK TO MONITOR
2696 015302 000207      SP.7:  RTS     PC
2697              ;DEVICE STATUS TABLE SETUP
2698
2699
2700              DST.SETUP:
2701 015304 013701 001442      MOV     DST,R1
2702 015310 012727 000020      MOV     #16.,(PC)+
2703 015314 000000      DS.1:  0
2704 015316 012702 014500      MOV     #CMD.STAT,R2
2705 015322 112221      DS.2:  MOVB   (R2)+,(R1)+
2706 015324 005337 015314      DEC     DS.1
2707 015330 001374              BNE     DS.2
2708 015332 000207              RTS     PC
2709
2710              ;SUBROUTINE TO CHECK THAT CU ADDRESS AND THE NUMBER OF DEVICES
2711              ;PER CU IS LEGAL
2712
2713 015334 012700 014254      CKCUA: MOV     #LEGAL.ADRS,R0 ;
2714 015340 005027      CKC1:  CLR     (PC)+
2715 015342 000000      CKC2:  0
2716 015344 111037 015342      MOVB   @R0,@#CKC2      ;FETCH CU ADDRESS
2717              ;***** MOD APR 74 *****
2718              ;*
2719              ;* ADDRESS RANGE MOD
2720              ;*
2721 015350 122737 000020 014236 CKC3:  CMPB   #20,@#MAX.DEV.CU ;CHECK LIMIT 16.
2722              ;***** MOD APR 74 *****
2723              ;*
2724              ;*
2725 015356 103005              BHIS   CKC4      ;BRANCH IF WITHIN LIMITS
2726 015360 000004 015404      TYPE   ,IDVN     ;ILLEGAL NUMBER OF DEVICES PER CU
2727 015364 012716 013362      MOV     #MON5,(SP) ;CHANGE RETURN PC
2728 015370 000207              RTS     PC
2729 015372 005720      CKC4:  TST     (R0)+
2730 015374 021027 177777      CMP     @R0,#-1
2731 015400 001357              BNE     CKC1
```

```

2732 015402 000207
2733 015404 044537 046114 043505 IDVN: RTS PC
2734 015412 046101 021440 047440 .ASCIZ "'_ILLEGAL # OF DEVICES PER CU '"
2735 015420 020106 042504 044526
2736 015426 042503 020123 042520
2737 015434 020122 052503 000040
2738
2739 .EVEN
2740 ;PRE-INIT SUBROUTINE
2741
2742 015442 012737 015610 000004 PREI: MOV #PREITO,4
2743 015450 012737 000340 000006 MOV #LEVEL7,6
2744 015456 005077 163636 CLR @DXES ;CLEAR MAINT CLK
2745 015462 004737 015614 JSR PC,RESRES ;DX RESET AND RESTORE
2746 ;THE FOLLOWING INSTRUCTION GET MODIFIED UPON THE COMPLETION
2747 ;OF THE SYSTEM RESET TEST.IF SCOPE PROBLEMS DEVELOP BEFORE THIS TEST
2748 ;PASSES THIS INST. CAN BE PATCHED TO A TESET,NOP.
2749 015466 004737 015570 PREI.1: JSR PC,@#CLRMO ;MODIFIED TO CLRMO
2750 015472 005077 163602 CLR @DXCS ;CLR DONE,LOCKO
2751 015476 004737 015614 JSR PC,RESRES ;DX RESET AND RESTORE
2752 015502 013777 001436 163572 MOV SPW,@DXOS
2753 015510 023777 001436 163564 CMP SPW,@DXOS
2754 015516 001401 BEQ .+4 ;BRANCH IF NO ERROR CONDITION
2755 015520 104000 ERROR ;
2756 015522 052777 000010 163570 BIS #TIMDIS,@DXES ;TIMER DISABLE
2757 015530 032777 000010 163562 BIT #TIMDIS,@DXES ;
2758 015536 001001 BNE .+4 ;BRANCH IF NO ERROR CONDITION
2759 015540 104000 ERROR ;
2760 015542 012737 000006 000004 MOV #6,4
2761 015550 012737 000000 000006 MOV #HALT,6
2762 015556 004737 006176 JSR PC,ZEROTT ;ZERO TUMBLE TABLE
2763 015562 004737 006226 JSR PC,TTZERO ;VERIFY TT ZERO
2764 015566 000207 RTS PC
2765
2766
2767
2768
2769
2770
2771 015570 005077 163514 CLRMO: CLR @DXMO ;DO SYSTEM RESET
2772 015574 032777 100000 163506 BIT #OPLO,@DXMO ;IS OPLO CLEAR
2773 015602 001401 BEQ NOCLR ;IF YES CONTINUE
2774 015604 104000 ERROR ;OPLO NOT RESET
2775 015606 000207 NOCLR: RTS PC
2776
2777 015610 104000 PREITO: ERROR ;PREINIT TIME OUT ERROR
2778
2779
2780 015612 000002 RTI
2781
2782
2783 ;DX RESET AND RESTORE ROUTINE
2784
2785
2786
2787
  
```

2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843

015614

RESRES:

015614 042777 000200 163456
015622 012777 000001 163450
015630 013737 001440 006526
015636 000240
015640 004737 007234
015644 000207

BIC #DONE,@DXCS ;CLEAR LOCKO
MOV #DXFRS,@DXCS ;DX RESET
MOV @#TT,TTRACE ;RELOAD SOFT TT POINTER
NOP ;INSERT RESET I.E.'5' HERE IF REQUIRED
JSR PC,ONLIN ;GET BACK ONLINE
RTS PC

;REGISTER ADDRESS SETUP ROUTINE

REG.SETUP:

015646
015646 013700 001262
015652 012701 001274
015656 010021
015660 062700 000002
015664 020127 001326
015670 001372

MOV DXBASE,R0 ;FETCH BASE ADRS
MOV #DXDS,R1 ;FETCH ADRS OF DXDS ADRS
RS.1: MOV R0,(R1)+
ADD #2,R0 ;INC TO NEXT DX ADRS
CMP R1,#DXES1+2
BNE RS.1

015672 004537 015726

JSR R5,SBYTE ;SETUP BYTE REF REG'S

015676 001276
015700 001326

DXCA
CUAR

015702 001302
015704 001332

DXOS
CUSR

015706 001310
015710 001336

DXMO
BUSO

015712 001312
015714 001342

DXMI
BUSI

015716 001320
015720 001346

DXES
MISC

015722 177777

-1

015724 000207

RTS PC

SBYTE:

015726
015726 012500
015730 012501
015732 011021
015734 011011
015736 005221

MOV (R5)+,R0
MOV (R5)+,R1
MOV @R0,(R1)+
MOV @R0,@R1
INC (R1)+

```

2844 015740 021527 177777          CMP    @R5,#-1
2845 015744 001370          BNE    SBYTE
2846 015746 005725          TST    (R5)+ ;POP OVER TERMINATOR
2847 015750 000205          RTS    R5
2848
2849
2850          ;ACCEPT HEX NUMBER FROM TTY
2851
2852 015752 005037 016410  GETHEX: CLR    HEXNUM          ;CLEAR HEXADECIMAL NUMBER LOCATION
2853 015756 010246          MOV    R2,-(SP)          ;SAVE R2
2854 015760 010146          MOV    R1,-(SP)          ;SAVE R1
2855 015762 010046          MOV    R0,-(SP)          ;SAVE R0
2856 015764 005001  ACPH:  CLR    R1          ;
2857 015766 104006  ACPH.1: KEY.TO.R0          ;FETCH AN ASCII CHAR FROM KEYBOARD
2858 015770 120027 000003  CMPB   R0,#3          ;CONTROL C?
2859 015774 001002          BNE    AH.2
2860 015776 000137 012724  JMP    @MON1.0
2861 016002 122700 000177  AH.2:  CMPB   #177,R0          ;TEST FOR RUBOUT
2862 016006 001424          BEQ    RUBOUH
2863 016010 122700 000015  CMPB   #15,R0          ;TEST FOR <CR>
2864 016014 001424          BEQ    CARGH
2865 016016 120027 000040  CMPB   R0,#40          ;EXIT IF SPACE
2866 016022 001421          BEQ    CARGH
2867 016024 120027 000000  CMPB   R0,#0          ;TEST FOR VALID HEX NUMBER
2868 016030 002413          BLT    RUBOUH
2869 016032 120027 000071  CMPB   R0,#'9
2870 016036 003021          BGT    AH.3
2871 016040 042700 177760  AH.3:  BIC    #177760,R0          ;CONVERT ASCII TO HEX
2872 016044 006301          ASL    R1
2873 016046 006301          ASL    R1
2874 016050 006301          ASL    R1
2875 016052 006301          ASL    R1
2876 016054 050001          BIS    R0,R1          ;CHALK'M UP
2877 016056 000743          BR    ACPH.1          ;FETCH NEXT CHAR
2878
2879 016060 000004 021240  RUBOUH: TYPE   ,,QUES          ;TYPE?
2880 016064 000737          BR    ACPH
2881 016066 010137 016410  CARGH: MOV    R1,HEXNUM          ;PLACE HEX NUMBER HERE
2882 016072 012600          MOV    (SP)+,R0          ;RESTORE R0
2883 016074 012601          MOV    (SP)+,R1          ;RESTOR R1
2884 016076 012602          MOV    (SP)+,R2          ;RESTORE R2
2885 016100 000207          RTS    PC
2886
2887 016102 005002          AHEX:  CLR    R2
2888 016104 120062 016412  AHEX0: CMPB   R0,ATBL(R2)          ;LOOK THRU ASCII TABLE
2889 016110 001406          BEQ    AHEX1          ;BRANCH ON MATCH
2890 016112 005202          INC    R2
2891 016114 126227 016412 000000  CMPB   ATBL(R2),#0          ;LOOK FOR END OF TABLE
2892 016122 001370          BNE    AHEX0          ;BRANCH IF NOT END
2893 016124 000755          BR    RUBOUH          ;ERROR ON NO MATCH
2894 016126 116200 016422  AHEX1: MOVB   HTBL(R2),R0          ;LOAD BINARY OF FIND
2895 016132 000742          BR    AH.3
2896
2897
2898
2899          ;***** MOD APR 74 *****

```

```
2900      ;
2901      ;
2902      016400      .=.!377+1
2903      ;
2904      ; ILLEGAL OR MALFUNCTIONING CUAR ERROR STATUS TABLE MODULO 8
2905      ;
2906      016400      ERRDST:
2907      016400      002      .BYTE UC      ;UNIT CHECK ENTRIES
2908      016401      002      .BYTE UC      ;UNIT CHECK ENTRIES
2909      016402      002      .BYTE UC      ;UNIT CHECK ENTRIES
2910      016403      002      .BYTE UC      ;UNIT CHECK ENTRIES
2911      016404      002      .BYTE UC      ;UNIT CHECK ENTRIES
2912      016405      002      .BYTE UC      ;UNIT CHECK ENTRIES
2913      016406      002      .BYTE UC      ;UNIT CHECK ENTRIES
2914      016407      002      .BYTE UC      ;UNIT CHECK ENTRIES
2915      ;
2916      016410      000000      HEXNUM: 0      ;HEX NUMBER
2917      016412      041101      042103      043105      ATBL: .ASCII 'ABCDEF'
2918      016420      000000      .WORD 0
2919      016422      012      013      014      HTBL: .BYTE 10.,11.,12.,13.,14.,15.
2920      016425      015      016      017
2921      ;
2922      ;ACCEPT OCTAL NUMBER FROM TTY
2923      016430      T.ACCEPTO:
2924      016430      005037      014232      CLR OCTNUM      ;CLEAR OCTAL NUMBER LOCATION
2925      016434      010146      MOV R1,-(SP)      ;SAVE R1
2926      016436      010046      MOV R0,-(SP)      ;SAVE R0
2927      016440      005001      ACPTO: CLR R1      ;
2928      016442      104006      ACPTO.1: KEY.TO.R0      ;FETCH AN ASCII CHAR FROM KEYBOARD
2929      016444      120027      000003      CMPB R0,#3      ;CONTROL C?
2930      016450      001002      BNE AO.2
2931      016452      000137      012724      JMP @MON1.0
2932      016456      122700      000177      AO.2: CMPB #177,R0      ;TEST FOR RUBOUT
2933      016462      001423      BEQ RUBOUT
2934      016464      122700      000015      CMPB #15,R0      ;TEST FOR <CR>
2935      016470      001423      BEQ CARG
2936      016472      120027      000040      CMPB R0,#40      ;EXIT IF SPACE
2937      016476      001420      BEQ CARG
2938      016500      120027      000000      CMPB R0,#0      ;TEST FOR VALID OCTAL NUMBER
2939      016504      002412      BLT RUBOUT
2940      016506      120027      000067      CMPB R0,#'7
2941      016512      003007      BGT RUBOUT
2942      016514      042700      177770      BIC #177770,R0      ;CONVERT ASCII TO OCTAL
2943      016520      006301      ASL R1
2944      016522      006301      ASL R1
2945      016524      006301      ASL R1
2946      016526      050001      BIS R0,R1      ;CHALK'M UP
2947      016530      000744      BR ACPTO.1      ;FETCH NEXT CHAR
2948      ;
2949      016532      000004      021240      RUBOUT: TYPE .,QUES      ;TYPE?
2950      016536      000740      BR ACPTO
2951      016540      010137      014232      CARG: MOV R1,OCTNUM      ;PLACE OCTAL NUMBER HERE
2952      016544      012600      MOV (SP)+,R0      ;RESTORE R0
2953      016546      012601      MOV (SP)+,R1      ;RESTOR R1
2954      016550      000002      RTI      ;RETURN
2955      ;
```

```

2956          ;FETCH AN ASCII CHARACTER FROM KEYBOARD
2957
2958 016552          T.KEY.TO.R0:
2959 016552 105777 162620          TSTB @TKS          ;TEST FOR DONE
2960 016556 100375          BPL     .-4          ;WAIT FOR KEYBOARD
2961 016560 117700 162614          MOVB @TKB,R0        ;FETCH CHAR
2962 016564 117777 162610 162612 MOVB @TKB,@TPB      ;ECHO
2963 016572 004737 016604          JSR PC,TTYFLG      ;WAIT FOR DONE
2964 016576 042700 177600          BIC #177600,R0    ;7 BIT ASCII
2965 016602 000002          RTI
2966          ;TEST FOR TRANSMITTER DONE
2967
2968 016604          TTYFLG:
2969 016604 105777 162572 2$: TSTB @TPS
2970 016610 100375          BPL 2$
2971 016612 000207          RTS PC
2972          .SBTTL TTY ASCII OUTPUT ROUTINE
2973
2974
2975 016614 032737 020000 177570 .IOT: BIT #BIT13,SR      ;TEST FOR INHIBIT PRINT
2976 016622 001040          BNE .IOTE
2977 016624 010537 016732          MOV TTY,.SAV      ;SAVE TTY
2978 016630 017605 000000          MOV @ (6),TTY    ;GET ADDRESS TO BE TYPED
2979 016634 122715 000044          .MORE: CMPB #'$',(TTY) ;TERMINATOR?
2980 016640 001425          BEQ .TERM
2981 016642 105715          TSTB (TTY)        ;TERMINATOR?
2982 016644 001423          BEQ .TERM
2983 016646 122715 000001          CMPB #1,(TTY)    ;RESTORE OLD SEQUENCE
2984 016652 001416          BEQ .REST
2985 016654 122715 000137          CMPB #'',(TTY)  ;SET UP CR LF
2986 016660 001406          BEQ .CRLF
2987 016662 105777 162514          TSTB @TPS
2988 016666 100375          BPL .-4
2989 016670 112577 162510          MOVB (TTY)+,@TPB
2990 016674 000757          BR .MORE
2991 016676 005205          .CRLF: INC TTY
2992 016700 010546          MOV TTY,-(6)
2993 016702 012705 016734          MOV #.CAR,TTY
2994 016706 000752          BR .MORE
2995 016710 012605          .REST: MOV (6)+,TTY
2996 016712 000750          BR .MORE
2997 016714 004737 016604          .TERM: JSR PC,TTYFLG ;WAIT FOR DONE
2998 016720 013705 016732          MOV .SAV,TTY
2999 016724 062716 000002          .IOTE: ADD #2,(6) ;POP
3000 016730 000002          RTI
3001
3002 016732 000000          .SAV: 0
3003 016734 005015 001002 001002 .CAR: .ASCII <CR><LF><2><2><2><2><2><2><1>
3004 016742 001002          OC1
3005 016746
3006 016746 000000          .EVEN
3007          .TYPE: 0
3008          .SBTTL SAVE AND RESTORE REGISTERS
3009          ;SAVE REGS 0 TO 4 SUBROUTINE.
3009 016750 012637 017006          T.SAVRG: MOV (6)+,SVRPC ;SAVE PC AND PSW.
3010 016754 012637 017010          MOV (6)+,SVRPSW
3011 016760 010546          MOV %5,-(6)

```



```

3012 016762 010446          MOV    %4,-(6) ;SAVE REGS 0 - 4
3013 016764 010346          MOV    %3,-(6) ;IN STACK.
3014 016766 010246          MOV    %2,-(6)
3015 016770 010146          MOV    %1,-(6)
3016 016772 010046          MOV    %0,-(6)
3017 016774 013746 017010   MOV    SVRPSW,-(6) ;RESTORE PC AND PSW.
3018 017000 013746 017006   MOV    SVRPC,-(6)
3019 017004 000002          RTI    ;EXIT.
3020 017006 000000          SVRPC: 0
3021 017010 000000          SVRPSW: 0
3022                          ;RESTORE REGS 0 TO 4 SUBROUTINE.
3023 017012 012637 017050   T.RSTRG: MOV (6)+,RSTPC ;SAVE PC AND PSW.
3024 017016 012637 017052   MOV    (6)+,RSTPSW
3025 017022 012600          MOV    (6)+,%0 ;RESTORE REGS 0 - 4
3026 017024 012601          MOV    (6)+,%1 ;FROM STACK.
3027 017026 012602          MOV    (6)+,%2
3028 017030 012603          MOV    (6)+,%3
3029 017032 012604          MOV    (6)+,%4
3030 017034 012605          MOV    (6)+,%5
3031
3032 017036 013746 017052   MOV    RSTPSW,-(6) ;RESTORE PC AND PSW.
3033 017042 013746 017050   MOV    RSTPC,-(6)
3034 017046 000002          RTI    ;EXIT
3035 017050 000000          RSTPC: 0
3036 017052 000000          RSTPSW: 0
3037                          .SBTTL OCTAL DUMP ROUTINE
3038
3039 017054 000000 000000 000000 PRINT2: .WORD 0,0,0,0
3040 017062 000000
3041 017064 000 000 PRINT3: .BYTE 0,0
3042
3043 017066 112737 000001 017064 PRINTR: MOVB #1,PRINT3 ;SET ZERO FILL SWITCH
3044 017074 000402 BR .+6
3045 017076 005037 017064 PRINTS: CLR PRINT3 ;SUPRESS LEADING ZERO'S
3046 017102 112737 177772 017065 MOVB #-6,PRINT3+1 ;SET COUNT
3047 017110 032737 020000 177570 BIT #BIT13,SR
3048 017116 001041 BNE PRTE
3049 017120 010446 MOV %4,-(6) ;SAVE R4
3050 017122 012704 017054 MOV #PRINT2,%4 ;SET POINTER TO FIRST ASCII CHAR.
3051 017126 105014 CLRB (4) ;CLEAR FIRST BYTE
3052 017130 000405 BR PRINTF ;ROTATE FIRST BIT
3053 017132 105014 PRINTL: CLRB (4) ;CLEAR BYTE OF CHARACTER
3054 017134 006105 ROL TTY ;ROTATE BIT INTO C
3055 017136 106114 ROLB (4) ;PACK IT
3056 017140 006105 ROL TTY ;ROTATE BIT INTO C
3057 017142 106114 ROLB (4) ;PACK IT
3058 017144 006105 PRINTF: ROL TTY ;ROTATE BIT INTO C
3059 017146 106114 ROLB (4) ;PACK IT
3060 017150 105714 TSTB (4)
3061 017152 001402 BEQ .+6
3062 017154 105237 017064 INCB PRINT3
3063 017160 105737 017064 TSTB PRINT3 ;CHECK FILL SWITCH
3064 017164 001402 BEQ .+6
3065 017166 152724 000060 BISB #'0,(4)+ ;MAKE INTO ASCII CHAR
3066 017172 105237 017065 INCB PRINT3+1
3067 017176 001355 BNE PRINTL ;REPEAT

```

```
3068 017200 022704 017054          CMP      #PRINT2,%4
3069 017204 001002          BNE      .+6
3070 017206 112724 000060          MOV      #'0,(4)+
3071 017212 105014          CLRB    (4)
3072 017214 000004 017054          TYPE    ,PRINT2          ;TYPE IT
3073 017220 012604          MOV      (6)+,%4          ;RESTORE R5
3074 017222 000207          PRTE:   RTS      %7
3075          ;TTY WATCH DOG FOR CONTROL C
3076          000003          CNTLC=3          ;ASCII CONTROL C
3077
3078 017224 117727 162150          TTYI:   MOV      @TKB,(PC)+ ;SAVE CHAR
3079 017230 000000          SCHAR:  0          ;HERE
3080 017232 042737 000200 017230          BIC      #200,@#SCHAR      ;SEVEN LEVEL ASCII
3081 017240 122737 000003 017230          CMP      #CNTLC,@#SCHAR    ;CHECK FOR CONTROL C
3082 017246 001004          BNE      TTYI0
3083 017250 000004 017274          TYPE    ,ACLC          ;TYPE CONTROL C
3084 017254 000137 012724          JMP      @#MON1,0
3085 017260 004737 016604          TTYI0:  JSR      PC,TTYFLG
3086 017264 113777 017230 162112          MOV      @#SCHAR,@TPB      ;ECHO CHARACTER
3087 017272 000002          RTI
3088
3089 017274 041536 000          ACLC:   .ASCIIZ <136><103>
3090          017300          .EVEN
3091
3092          017400          .=.!377+1          ;FORM MOD(400) BOUNDRY
3093
3094          017400          DSTADRS=.          ;DEFAULT DST
3095
3096 017400 000          .BYTE  0          ;TIO
3097 017401 000          .BYTE  0          ;WRITE
3098 017402 000          .BYTE  0          ;READ
3099 017403 014          .BYTE  CE!DE      ;NOP
3100 017404 000          .BYTE  0          ;SENSE
3101
3102 017405 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3103 017406 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3104 017407 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3105 017410 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3106 017411 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3107 017412 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3108 017413 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3109 017414 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3110 017415 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3111 017416 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3112 017417 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3113 017420 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3114 017421 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3115 017422 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3116 017423 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3117 017424 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3118 017425 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3119 017426 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3120 017427 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3121 017430 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3122 017431 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
3123 017432 002          .BYTE  UC          ;ILLEGAL ,UNIT CHECK
```

3124	017433	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3125	017434	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3126	017435	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3127	017436	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3128	017437	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3129	017440	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3130	017441	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3131	017442	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3132	017443	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3133	017444	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3134	017445	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3135	017446	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3136	017447	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3137	017450	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3138	017451	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3139	017452	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3140	017453	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3141	017454	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3142	017455	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3143	017456	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3144	017457	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3145	017460	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3146	017461	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3147	017462	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3148	017463	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3149	017464	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3150	017465	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3151	017466	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3152	017467	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3153	017470	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3154	017471	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3155	017472	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3156	017473	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3157	017474	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3158	017475	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3159	017476	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3160	017477	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3161	017500	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3162	017501	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3163	017502	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3164	017503	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3165	017504	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3166	017505	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3167	017506	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3168	017507	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3169	017510	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3170	017511	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3171	017512	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3172	017513	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3173	017514	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3174	017515	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3175	017516	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3176	017517	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3177	017520	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3178	017521	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3179	017522	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK

3180	017523	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3181	017524	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3182	017525	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3183	017526	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3184	017527	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3185	017530	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3186	017531	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3187	017532	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3188	017533	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3189	017534	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3190	017535	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3191	017536	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3192	017537	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3193	017540	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3194	017541	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3195	017542	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3196	017543	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3197	017544	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3198	017545	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3199	017546	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3200	017547	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3201	017550	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3202	017551	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3203	017552	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3204	017553	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3205	017554	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3206	017555	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3207	017556	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3208	017557	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3209	017560	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3210	017561	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3211	017562	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3212	017563	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3213	017564	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3214	017565	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3215	017566	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3216	017567	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3217	017570	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3218	017571	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3219	017572	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3220	017573	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3221	017574	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3222	017575	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3223	017576	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3224	017577	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3225	017600	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3226	017601	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3227	017602	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3228	017603	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3229	017604	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3230	017605	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3231	017606	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3232	017607	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3233	017610	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3234	017611	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK
3235	017612	002	.BYTE	UC	:ILLEGAL	,UNIT	CHECK

3236	017613	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3237	017614	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3238	017615	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3239	017616	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3240	017617	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3241	017620	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3242	017621	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3243	017622	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3244	017623	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3245	017624	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3246	017625	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3247	017626	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3248	017627	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3249	017630	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3250	017631	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3251	017632	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3252	017633	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3253	017634	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3254	017635	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3255	017636	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3256	017637	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3257	017640	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3258	017641	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3259	017642	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3260	017643	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3261	017644	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3262	017645	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3263	017646	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3264	017647	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3265	017650	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3266	017651	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3267	017652	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3268	017653	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3269	017654	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3270	017655	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3271	017656	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3272	017657	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3273	017660	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3274	017661	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3275	017662	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3276	017663	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3277	017664	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3278	017665	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3279	017666	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3280	017667	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3281	017670	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3282	017671	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3283	017672	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3284	017673	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3285	017674	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3286	017675	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3287	017676	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3288	017677	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3289	017700	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3290	017701	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK
3291	017702	002	.BYTE	UC	: ILLEGAL	.UNIT	CHECK

3292	017703	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3293	017704	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3294	017705	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3295	017706	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3296	017707	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3297	017710	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3298	017711	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3299	017712	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3300	017713	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3301	017714	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3302	017715	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3303	017716	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3304	017717	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3305	017720	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3306	017721	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3307	017722	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3308	017723	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3309	017724	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3310	017725	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3311	017726	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3312	017727	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3313	017730	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3314	017731	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3315	017732	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3316	017733	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3317	017734	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3318	017735	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3319	017736	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3320	017737	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3321	017740	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3322	017741	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3323	017742	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3324	017743	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3325	017744	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3326	017745	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3327	017746	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3328	017747	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3329	017750	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3330	017751	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3331	017752	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3332	017753	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3333	017754	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3334	017755	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3335	017756	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3336	017757	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3337	017760	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3338	017761	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3339	017762	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3340	017763	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3341	017764	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3342	017765	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3343	017766	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3344	017767	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3345	017770	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3346	017771	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK
3347	017772	002	.BYTE	UC	:ILLEGAL	.UNIT	CHECK

```
3348 017773 002 .BYTE UC ;ILLEGAL ,UNIT CHECK
3349 017774 002 .BYTE UC ;ILLEGAL ,UNIT CHECK
3350 017775 002 .BYTE UC ;ILLEGAL ,UNIT CHECK
3351 017776 002 .BYTE UC ;ILLEGAL ,UNIT CHECK
3352 017777 002 .BYTE UC ;ILLEGAL ,UNIT CHECK
3353 .EVEN
3354
3355 020000 T.PCH1:
3356 020000 000012 .BLKW 10.
3357
3358 020024 T.PCH2:
3359 020024 000012 .BLKW 10.
3360
3361 020050 T.PCH3:
3362 020050 000012 .BLKW 10.
3363
3364 .SBTTL MESSAGES
3365 LF=12
3366 CR=15
3367 020074 HEADER:
3368 :*
3369 :***** MOD JULY 78 *****
3370 .NLIST BEX
(1) 020074 040537 026503 033470 .ASCII "_AC-8772D-MC CZDXHDO DX11B ONLINE EXER (JUL 78)"
(1) :*
(1) :***** MOD JULY 78 *****
(1) :*
(1) 020153 137 047506 020122 .ASCII "FOR DYNAMIC SWITCH REGISTER SETTINGS"
(1) 020220 051537 042505 050040 .ASCII "SEE PROGRAM LISTING PAGE #1"
(1) 020254 057537 044504 041523 .ASCII "DISCONNECT 'DX11' FROM 360/370 CHANNEL_"
(1) 020325 137 047503 047116 .ASCII "CONNECT MAINTENANCE CABLES"
(1) 020360 052137 051110 053517 .ASCII "THROW ENABLE SWITCH ONLINE"
(1) 020413 137 052137 050131 .ASCII "TYPE: <D>,DEFAULT PARAMETERS"
(1) 020453 137 020040 020040 .ASCII "<P>,PREVIOUS PARAMETERS"
(1) 020513 137 020040 020040 .ASCII "<S>,SELECT PARAMETERS"
(1) 020551 137 020040 020040 .ASCII "<N>,START THIS TEST NUMBER"
(1) :*
(1) :***** MOD APR 74 *****
(1) 020615 137 042137 050054 FSTART: .ASCIIZ "_D,P,S,N?"
(1)
(1) 020631 137 040502 042523 MSG2: .ASCIIZ "BASE ADDRESS: "
(1) 020651 137 042504 044526 MSG3: .ASCIIZ "DEVICES PER CU: "
(1) 020673 137 042524 052123 MSG5: .ASCIIZ "TEST NUMBER: "
(1) 020712 052137 050131 020105 MSG4: .ASCIIZ "TYPE CU ADRS'S IN HEX <CR><LF>; <CR><CR> TERMINATES LIST"
(1) 021004 040537 051104 035123 MSG6: .ASCIIZ "ADRS: "
(1) 021014 051537 052105 051440 MSG7: .ASCIIZ "SET SWITCHES "
(1) 021033 137 044514 052123 MSG9: .ASCIIZ "LIST ALL LEGAL COMMANDS"
(1) 021064 041537 046517 040515 MSG10: .ASCIIZ "COMMAND:"
(1) 021076 042137 020130 051120 MSG12: .ASCIIZ "DX PRIORITY LEVEL: "
(1) 021123 137 036440 044440 MSG13: .ASCIIZ "= ILLEGAL ?"
(1)
(1)
(1) 021141 137 042522 051507 RDH: .ASCIIZ "REGSTR - SHOULD BE - WAS"
(1) 021175 035 000037 HOME: .ASCIIZ "<35><37>"
(1) 021200 000207 BELL: .ASCIIZ "<207>"
(1) 021202 042537 042116 052040 ENDTST: .ASCIIZ "_END TEST"
```

```

(1) 021214 020137 051105 047522 ECM: .ASCIZ " ERRORS DETECTED: "
(1) 021240 057477 000 .QUES: .ASCIZ "? "
(1) 021243 040 020040 020040 SPAC4: .ASCIZ "<40><40><40><40><40><40>"
(1) 021252 000040 SPACE: .ASCIZ "<40>"
(1) 021254 000137 CRLF: .ASCIZ ""
(1) 021256 052137 050131 020105 STALL: .ASCIZ "TYPE IN STALL COUNT: "
(1) 021305 137 042537 051122 ERPC: .ASCIZ " _ERROR PC: "
(1)
(1) 021322 042137 042130 035123 ADXDS: .ASCIZ " DXDS:"
(1) 021331 137 054104 040503 ADXCA: .ASCIZ " DXCA:"
(1) 021340 042137 041530 035123 ADXCS: .ASCIZ " DXCS:"
(1) 021347 137 054104 051517 ADXOS: .ASCIZ " DXOS:"
(1) 021356 042137 041130 035101 ADXBA: .ASCIZ " DXBA:"
(1) 021365 137 054104 041502 ADXBC: .ASCIZ " DXBC:"
(1) 021374 042137 046530 035117 ADXMO: .ASCIZ " DXMO:"
(1) 021403 137 054104 044515 ADXMI: .ASCIZ " DXMI:"
(1) 021412 042137 041530 035102 ADXCB: .ASCIZ " DXCB:"
(1) 021421 137 054104 042116 ADXND: .ASCIZ " DXND:"
(1) 021430 042137 042530 035123 ADXES: .ASCIZ " DXES:"
(1) 021437 137 054104 051505 ADXES1: .ASCIZ " DXES1: "
(1)
(1) ;***** MOD APR 74 *****
(1) ;* ERROR TEXT MOD
(1)
(1) 021450 020137 052503 042101 MSG26: .ASCIZ " CUADRS/MO:"
(1) ;***** MOD APR 74 *****
(1) 021465 137 042526 052103 MSG28: .ASCIZ " VECTOR ADDRESS: "
(1) 021507 137 052123 052101 MSG31: .ASCIZ " STATUS: "
(1) 021521 137 051105 047522 MSG35: .ASCIZ " ERROR IN TEST: "
(1) 021542 047537 044522 044507 MSG36: .ASCIZ " CRIGIN OF MAP ERROR: "
(1) 021571 137 052124 052040 TRCM1: .ASCIZ " TT TRACE ERROR IN TEST: "
(1) 021623 137 051117 043511 TRCM: .ASCIZ " ORIGN OF LAST TT TRACE UPDATE: "
(1) 021664 041040 051525 020131 ABSYM: .ASCIZ " BUSY ENABLE "
(1) 021703 115 046125 044524 AMUXM: .ASCIZ " MULTIPLEXER CH"
(1) 021723 137 047105 051124 TRC1: .ASCIZ " ENTRY WAS : "
(1) 021745 137 047105 051124 TRC2: .ASCIZ " ENTRY SHOULD BE: "
(1) 021767 137 047105 051124 TTDS: .ASCIZ " ENTRY WAS FROM DXDS"
(1) 022014 042537 052116 054522 TTCA: .ASCIZ " ENTRY WAS FROM DXCA"
(1)
(1) .LIST BEX
3371
3372 .SBTTL DATA BUFFERS
3373
3374 ;DATA BUFFER -- CONTAINS 0 - 255 IN ONE BYTE ITEMS
3375
3376 ;DATA FILE FOR CHANNEL SIMULATOR WRITE CMDS
3377 022042 .EVEN
3378
3379 022042 DATA:
3380
3381
3382
3383 N=1
3384 022042 000001 .WORD N
3385 000002 N=N*2
3386 022044 000002 .WORD N
3387 000004 N=N*2
3388 022046 000004 .WORD N

```


3389		000010	N=N*2	
3390	022050	000010	.WORD	N
3391		000020	N=N*2	
3392	022052	000020	.WORD	N
3393		000040	N=N*2	
3394	022054	000040	.WORD	N
3395		000100	N=N*2	
3396	022056	000100	.WORD	N
3397		000200	N=N*2	
3398	022060	000200	.WORD	N
3399		000400	N=N*2	
3400	022062	000400	.WORD	N
3401		001000	N=N*2	
3402	022064	001000	.WORD	N
3403		002000	N=N*2	
3404	022066	002000	.WORD	N
3405		004000	N=N*2	
3406	022070	004000	.WORD	N
3407		010000	N=N*2	
3408	022072	010000	.WORD	N
3409		020000	N=N*2	
3410	022074	020000	.WORD	N
3411		040000	N=N*2	
3412	022076	040000	.WORD	N
3413		100000	N=N*2	
3414	022100	100000	.WORD	N
3415		000000	N=N*2	
3416		000002	W=2	
3417	022102	177775	.WORD	-W-1
3418		000004	W=W+W	
3419	022104	177773	.WORD	-W-1
3420		000010	W=W+W	
3421	022106	177767	.WORD	-W-1
3422		000020	W=W+W	
3423	022110	177757	.WORD	-W-1
3424		000040	W=W+W	
3425	022112	177737	.WORD	-W-1
3426		000100	W=W+W	
3427	022114	177677	.WORD	-W-1
3428		000200	W=W+W	
3429	022116	177577	.WORD	-W-1
3430		000400	W=W+W	
3431	022120	177377	.WORD	-W-1
3432		001000	W=W+W	
3433	022122	176777	.WORD	-W-1
3434		002000	W=W+W	
3435	022124	175777	.WORD	-W-1
3436		004000	W=W+W	
3437	022126	173777	.WORD	-W-1
3438		010000	W=W+W	
3439	022130	167777	.WORD	-W-1
3440		020000	W=W+W	
3441	022132	157777	.WORD	-W-1
3442		040000	W=W+W	
3443	022134	137777	.WORD	-W-1
3444		100000	W=W+W	

3445	022136	077777	.WORD	-W-1
3446		000000	W=W+W	
3447	022140	177777	.WORD	-W-1
3448		000000	W=W+W	
3449		000001	N=1	
3450	022142	000001	.WORD	N
3451		000002	N=N*2	
3452	022144	000002	.WORD	N
3453		000004	N=N*2	
3454	022146	000004	.WORD	N
3455		000010	N=N*2	
3456	022150	000010	.WORD	N
3457		000020	N=N*2	
3458	022152	000020	.WORD	N
3459		000040	N=N*2	
3460	022154	000040	.WORD	N
3461		000100	N=N*2	
3462	022156	000100	.WORD	N
3463		000200	N=N*2	
3464	022160	000200	.WORD	N
3465		000400	N=N*2	
3466	022162	000400	.WORD	N
3467		001000	N=N*2	
3468	022164	001000	.WORD	N
3469		002000	N=N*2	
3470	022166	002000	.WORD	N
3471		004000	N=N*2	
3472	022170	004000	.WORD	N
3473		010000	N=N*2	
3474	022172	010000	.WORD	N
3475		020000	N=N*2	
3476	022174	020000	.WORD	N
3477		040000	N=N*2	
3478	022176	040000	.WORD	N
3479		100000	N=N*2	
3480	022200	100000	.WORD	N
3481		000000	N=N*2	
3482		000002	W=2	
3483	022202	177775	.WORD	-W-1
3484		000004	W=W+W	
3485	022204	177773	.WORD	-W-1
3486		000010	W=W+W	
3487	022206	177767	.WORD	-W-1
3488		000020	W=W+W	
3489	022210	177757	.WORD	-W-1
3490		000040	W=W+W	
3491	022212	177737	.WORD	-W-1
3492		000100	W=W+W	
3493	022214	177677	.WORD	-W-1
3494		000200	W=W+W	
3495	022216	177577	.WORD	-W-1
3496		000400	W=W+W	
3497	022220	177377	.WORD	-W-1
3498		001000	W=W+W	
3499	022222	176777	.WORD	-W-1
3500		002000	W=W+W	

3501	022224	175777	.WORD	-W-1
3502		004000	W=W+W	
3503	022226	173777	.WORD	-W-1
3504		010000	W=W+W	
3505	022230	167777	.WORD	-W-1
3506		020000	W=W+W	
3507	022232	157777	.WORD	-W-1
3508		040000	W=W+W	
3509	022234	137777	.WORD	-W-1
3510		100000	W=W+W	
3511	022236	077777	.WORD	-W-1
3512		000000	W=W+W	
3513	022240	177777	.WORD	-W-1
3514		000000	W=W+W	
3515		000001	N=1	
3516	022242	000001	.WORD	N
3517		000002	N=N*2	
3518	022244	000002	.WORD	N
3519		000004	N=N*2	
3520	022246	000004	.WORD	N
3521		000010	N=N*2	
3522	022250	000010	.WORD	N
3523		000020	N=N*2	
3524	022252	000020	.WORD	N
3525		000040	N=N*2	
3526	022254	000040	.WORD	N
3527		000100	N=N*2	
3528	022256	000100	.WORD	N
3529		000200	N=N*2	
3530	022260	000200	.WORD	N
3531		000400	N=N*2	
3532	022262	000400	.WORD	N
3533		001000	N=N*2	
3534	022264	001000	.WORD	N
3535		002000	N=N*2	
3536	022266	002000	.WORD	N
3537		004000	N=N*2	
3538	022270	004000	.WORD	N
3539		010000	N=N*2	
3540	022272	010000	.WORD	N
3541		020000	N=N*2	
3542	022274	020000	.WORD	N
3543		040000	N=N*2	
3544	022276	040000	.WORD	N
3545		100000	N=N*2	
3546	022300	100000	.WORD	N
3547		000000	N=N*2	
3548		000002	W=2	
3549	022302	177775	.WORD	-W-1
3550		000004	W=W+W	
3551	022304	177773	.WORD	-W-1
3552		000010	W=W+W	
3553	022306	177767	.WORD	-W-1
3554		000020	W=W+W	
3555	022310	177757	.WORD	-W-1
3556		000040	W=W+W	

3557	022312	177737	.WORD	-W-1
3558		000100	W=W+W	
3559	022314	177677	.WORD	-W-1
3560		000200	W=W+W	
3561	022316	177577	.WORD	-W-1
3562		000400	W=W+W	
3563	022320	177377	.WORD	-W-1
3564		001000	W=W+W	
3565	022322	176777	.WORD	-W-1
3566		002000	W=W+W	
3567	022324	175777	.WORD	-W-1
3568		004000	W=W+W	
3569	022326	173777	.WORD	-W-1
3570		010000	W=W+W	
3571	022330	167777	.WORD	-W-1
3572		020000	W=W+W	
3573	022332	157777	.WORD	-W-1
3574		040000	W=W+W	
3575	022334	137777	.WORD	-W-1
3576		100000	W=W+W	
3577	022336	077777	.WORD	-W-1
3578		000000	W=W+W	
3579	022340	177777	.WORD	-W-1
3580		000000	W=W+W	
3581		000001	N=1	
3582	022342	000001	.WORD	N
3583		000002	N=N*2	
3584	022344	000002	.WORD	N
3585		000004	N=N*2	
3586	022346	000004	.WORD	N
3587		000010	N=N*2	
3588	022350	000010	.WORD	N
3589		000020	N=N*2	
3590	022352	000020	.WORD	N
3591		000040	N=N*2	
3592	022354	000040	.WORD	N
3593		000100	N=N*2	
3594	022356	000100	.WORD	N
3595		000200	N=N*2	
3596	022360	000200	.WORD	N
3597		000400	N=N*2	
3598	022362	000400	.WORD	N
3599		001000	N=N*2	
3600	022364	001000	.WORD	N
3601		002000	N=N*2	
3602	022366	002000	.WORD	N
3603		004000	N=N*2	
3604	022370	004000	.WORD	N
3605		010000	N=N*2	
3606	022372	010000	.WORD	N
3607		020000	N=N*2	
3608	022374	020000	.WORD	N
3609		040000	N=N*2	
3610	022376	040000	.WORD	N
3611		100000	N=N*2	
3612	022400	100000	.WORD	N

3613		000000	N=N*2
3614		000002	W=2
3615	022402	177775	.WORD -W-1
3616		000004	W=W+W
3617	022404	177773	.WORD -W-1
3618		000010	W=W+W
3619	022406	177767	.WORD -W-1
3620		000020	W=W+W
3621	022410	177757	.WORD -W-1
3622		000040	W=W+W
3623	022412	177737	.WORD -W-1
3624		000100	W=W+W
3625	022414	177677	.WORD -W-1
3626		000200	W=W+W
3627	022416	177577	.WORD -W-1
3628		000400	W=W+W
3629	022420	177377	.WORD -W-1
3630		001000	W=W+W
3631	022422	176777	.WORD -W-1
3632		002000	W=W+W
3633	022424	175777	.WORD -W-1
3634		004000	W=W+W
3635	022426	173777	.WORD -W-1
3636		010000	W=W+W
3637	022430	167777	.WORD -W-1
3638		020000	W=W+W
3639	022432	157777	.WORD -W-1
3640		040000	W=W+W
3641	022434	137777	.WORD -W-1
3642		100000	W=W+W
3643	022436	077777	.WORD -W-1
3644		000000	W=W+W
3645	022440	177777	.WORD -W-1
3646		000000	W=W+W
3647			
3648		022442	.=. ;INDICATE ADDRESS OF END OF BUFFER
3649	022442	000000	WDATA: 0 ;BEGINNING OF WRITE DATA FILE
3650		023442	NPRDATA=WDATA+512. ;NPR INPUT DATA FILE
3651			;WDATA AND NPRDAT OVER WRITE ASCII TEXT
3652		000001	.END

CUE = 000040	210#																		
CUEND = 000040	48#																		
CUFBM = 040000	26#																		
CUIS.G 011030	1810#																		
CUIS.S 011020	890	969	1077	1406	1805#														
CUISO 011250	1808	1864#																	
CUOR 001334	364#																		
CUSR 001332	363#	1401*	1677	2823															
DATA 022042	958	959	3379#																
DE = 000004	206#	451	1401	2474	3099														
DELAY 006710	1345#	1346																	
DEV 014214	885	887	893	895	899	905	964	966	1025	1031	1072	1074	1153						
	1159	1318	1321	1562	1563	1568	1569	1623	1636	1729	1730	1755	1764						
	1769	1837	1869	1873	1892	1894	1898	2340*	2341	2371#									
DEVCNT 014226	2292*	2300*	2301	2303*	2342	2383#													
DEVEND= 000004	51#																		
DEV.A 014220	2341*	2344*	2346*	2347	2376#	2551*													
DFLT.C 014360	2445#	2555																	
DFLT.S 014420	2470#	2564																	
DOFLIN= 000002	201#																		
DONE = 000200	33#	1191	1194	1242	1245	1246	1364	1388	2796										
DS = 000000	119#																		
DSCRSP= 000002	113#																		
DST 001442	430#	1311*	1372*	1376*	1676	2554*	2676	2701											
DSTADR= 017400	430	2554	3094#																
DSTCNT 014244	2394#																		
DST.SE 015304	2287	2700#																	
DS.1 015314	2703#	2706*																	
DS.2 015322	2705#	2707																	
DTMP 011542	1911#	1912*	1913																
DXBA 001304	344#	880*	959*	1067*															
DXBASE 001262	335#	2160*	2546*	2810															
DXBC 001306	345#	879*	957*	1066*															
DXCA 001276	341#	2819																	
DXCB 001314	348#	1497	1523	1590	1724														
DXCS 001300	342#	881*	960*	1070*	1191	1194*	1242	1245*	1246	1314*	1315	1322*	1323						
	1364*	1375*	1379*	1388*	1403*	1490*	1491	1519*	1520	2055*	2315*	2750*	2796*						
	2797*																		
DXDS 001274	340#	1598	1651	2811															
DXES 001320	350#	1367*	2053*	2744*	2756*	2757	2831												
DXES1 001324	352#	1544	1554	2814															
DXFI = 000003	39#	1070																	
DXFO = 000005	40#	881	960																
DXFRS = 000001	38#	2055	2797																
DXFST = 000007	41#	1403																	
DXIS 001266	337#	1326*	1378*	2168*	2352*	2548*													
DXIV 001264	336#	1327*	1377*	2166*	2351*	2547*													
DXMI 001312	347#	445	448	974	989	1010	1083	1096	1135	1366*	1416*	1534	1538						
	1541	1548	1551	1601	1619	1630	1633	1656	1670	1812	1825	1829	1833						
	1846	2828																	
DXMO 001310	346#	443	981*	982	992*	993	998*	999	1002*	1003	1015*	1016	1019						
	1090*	1091	1117*	1118	1123*	1124	1127*	1128	1140*	1141	1144	1146*	1147						
	1310*	1365*	1407*	1408*	1412*	1426*	1428*	1437*	1439*	1500	1503*	1504	1507*						
	1508	1537*	1547*	1557*	1558	1561*	1562*	1574*	1579*	1580	1584*	1606*	1612*						
	1613	1616	1628	1646*	1660*	1664*	1665	1693*	1694	1697	1701*	1702	1710*						
	1711	1715*	1729*	1734*	1735	1738*	1739	1743*	1744	1759*	1760	1769*	1770						

		1777*	1778	1782	1786*	1788	1792*	1794	1816*	1817	1821	1841*	1842	1850*
		1851	1855	1859*	1860	1869*	1870*	1871*	1872*	1873*	1874*	1875*	1876*	1877*
		1884*	1885*	1886*	1894*	1895*	1896*	1897*	1898*	1899*	1900*	1901*	1902*	1903*
		1904*	2054*	2771*	2772	2825								
DXMOB	001322	351#												
DXND	001316	349#												
DXOS	001302	343#	462*	1486*	1487	2752*	2753	2822						
DXPRT	001270	338#	1326	1378	2180*	2352	2549*							
DXTO =	000020	106#												
D.DEV.	014766	2551	2582#											
D.DXBA	014754	2546	2577#											
D.DXIS	014770	2548	2583#											
D.DXIV	014756	2547	2578#											
D.DXPR	014760	2549	2579#											
D.FIRS	014752	2545	2576#											
D.LEGA	014764	2552	2581#											
D.MAX.	014762	2550	2580#											
E =	000013	252#	257#	258#	259#	260#	261#	262#	263#	264#	265#	266#	267#	
EC =	000001	805#												
ECM	021214	2320	3370#											
EMOD1	012442	2031	2033#											
EMOD2	012456	2034	2036#											
EMTABL=	012052	253#	257#	258#	259#	260#	261#	262#	263#	264#	265#	266#	267#	
EMTAG	012024	253	1948	1953#										
EMTDEC	011762	283	1939#											
EMTOK	012004	1944	1946#											
ENDEN =	020000	27#												
ENDSTR=	001630	488#												
ENDTST	021202	2319	3370#											
ENDTT =	004000	804#	2594											
ENTRY1	006366	826*	851*	872*	884*	892*	898*	950*	963*	972*	1024*	1059*	1071*	1080*
		1152*	1270#	1273	1275*	2538*								
ENTRY2	006426	885*	886*	893*	894*	899*	902*	904*	964*	965*	971*	1025*	1028*	1030*
		1072*	1073*	1081*	1153*	1156*	1158*	1281#	1284	1286*	1318*	1319*	2539*	
ERFLG	012602	1960*	1965*	1997*	2017	2061#								
ERPC	021305	2007	3370#											
ERRCNT	012634	1998*	2081#	2271*	2322	2544*								
ERRDST	016400	497	498	499	500	501	502	503	504	505	506	507	508	509
		510	511	512	516	517	518	519	520	521	522	523	524	525
		526	527	528	529	530	531	535	536	537	538	539	540	541
		542	543	544	545	546	547	548	549	550	554	555	556	557
		558	559	560	561	562	563	564	565	566	567	568	569	573
		574	575	576	577	578	579	580	581	582	583	584	585	586
		587	588	592	593	594	595	596	597	598	599	600	601	602
		603	604	605	606	607	611	612	613	614	615	616	617	618
		619	620	621	622	623	624	625	626	630	631	632	633	634
		635	636	637	638	639	640	641	642	643	644	645	649	650
		651	652	653	654	655	656	657	658	659	660	661	662	663
		664	668	669	670	671	672	673	674	675	676	677	678	679
		680	681	682	683	687	688	689	690	691	692	693	694	695
		696	697	698	699	700	701	702	706	707	708	709	710	711
		712	713	714	715	716	717	718	719	720	721	725	726	727
		728	729	730	731	732	733	734	735	736	737	738	739	740
		744	745	746	747	748	749	750	751	752	753	754	755	756
		757	758	759	763	764	765	766	767	768	769	770	771	772
		773	774	775	776	777	778	782	783	784	785	786	787	788

SHORT	2#	884	963	1071											
SNAPSH	2#														
SPWMM	2#	494	513	532	551	570	589	608	627	646	665	684	703	722	741
	760	779													
SPWMM	2#	494	513	532	551	570	589	608	627	646	665	684	703	722	741
	760	779													
SS	2#	824	849	870	948	1057									
STEPS	2#														
STRM	2#	268													
S.D	2#	1240													
TABLES	2#	486													
TIOCME	2#	806													
TRAPCA	2#														
TTTE	2#	1250													
TYPM	2#	2972													
WRICMD	2#	1034													
ZEROM	2#	1209													

. ABS. 022444 000

ERRORS DETECTED: 0

CZDXHD,CZDXHD/SOL/CRF=CZDXHD.P11
 RUN-TIME: 43 9 1 SECONDS
 RUN-TIME RATIO: 397/55=7.1
 CORE USED: 28K (55 PAGES)