

IDENTIFICATION

Product Code: MAINDEC-11-DZDRH-A-D
Product Name: DR11-L, DR11-M Exerciser
Date Created: Sept. 1, 1975
Maintainer: Diagnostic Engineering
Author: Edward C. Badger

Copyright (C) 1975, Digital Equipment Corporation

This software is furnished under a license for use only on a single computer system and may be copied only with the inclusion of the above copyright notice. This software, or any other copies thereof, may not be provided or otherwise made available to any other person except for use on such system and to one who agrees to these license terms. Title to and ownership of the software shall at all times remain in Digital.

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 the use or reliability of its software on equipment which is not supplied by Digital.

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
3.0	LOADING PROCEDURE
4.0	STARTING PROCEDURE
5.0	OPERATING PROCEDURE
6.0	ERRORS
7.0	RESTRICTIONS
8.0	MISCELLANEOUS
9.0	PROGRAM DESCRIPTION
10.0	FLOW CHARTS
11.0	LISTING

1.0 ABSTRACT *****

This program allows the user to checkout or debug the DR11-L or DR11-M module (UNIBUS 2 word input interface or UNIBUS 2 word output interface). To run the logic test, the user must have a maintenance loop-back cable. Logic tests are limited and cannot check the input or output data paths. To check input or output data paths, some means of generating or detecting data changes is needed.

2.0 REQUIREMENTS *****

2.1 Equipment

1. PDP-11/Family computer with 4K of core or more and I/O terminal.
2. DR11L or DR11M (To be tested).
3. For logic test, maintenance loop-back cable. The maintenance loop-back cable is made up of a Berg Connector with 2 wires. One wire connects pin "V" with pin "VV". The other wire connects pin "X" with pin "TT".
4. For auto-slave, a master PDP-11 with DR11-M and this program.
5. For auto-master, a slave CPU (PDP-8 or PDP-11) with input module (DR11-L).
6. For auto-master/slave, a DR11L cabled to a DR11M.
7. For all other tests, some means of generating or detecting data changes.

2.2 Storage

This program occupies core locations 000000-017474.

3.0 LOADING PROCEDURE *****

3.1 Method

Procedure for normal binary tapes should be followed:

1. Absolute Loader must be in memory.
2. Place binary tape in reader.
3. Load address #7500 (*Determined by location of loader).
4. Press "START" (Program will load).

4.0 STARTING PROCEDURE *****

4.1 Control Switch Settings

Starting at SA200 or 210 set all switches as desired. See Section 5.1.

4.2 Starting Addresses

1. Load and start at location 200 for initial questions about SPC modules.
2. Load and start at location 210 to avoid questions if you wish to run the program with the same parameters as set by a start at 200.

NOTE

After loading program into core you
must initially start at location 200.

4.3 Program AND/OR Operator Action

1. Load program into core.
2. Set switch register to starting address.
3. Load address.
4. Set switches to desired settings - See Section 5.1.
5. Press start.
6. The program will type first question.

4.4 Program Questions

4.4.1 TERMINAL CHANGE REQUIRED (Y OR N)?

Respond by typing a Y or (no carriage return required) to represent YES or NO. If the program is started on the CPU's console TTY and you wish to change to a different TTY, type "Y", otherwise type "N".

4.4.2 KEYBOARD CSR ADDR. OF NEW TERMINAL?

Question asked only if YES was the response to 4.4.1. Respond by typing the keyboard CSR address of the new terminal.

4.4.3 ITS VECTOR ADDR.?

Question asked only if YES was the response to 4.4.1. Respond by typing the vector address of the new terminal's keyboard.

4.4.4 PRINTER CSR ADDR. OF NEW TERMINAL?

Question asked only if YES was the response to 4.4.1. Respond by typing the printer CSR address of the new terminal.

4.4.5 CSR ADDR. OF INPUT MODULE?

Respond by typing the CSR address of the word on the input module you wish to exercise. If you do not wish to exercise an input module, type only a carriage return.

4.4.6 CSR ADDR. OF OUTPUT MODULE?

Respond by typing the CSR address of the word on the output module you wish to exercise. If you do not wish to exercise an output module, type only a carriage return.

4.4.7 MODE?

Respond to this question by typing a character from "A" to "M" followed by carriage return. At any time (except logic tests) you may return to this question by typing "tC" (control and the letter "C" typed together).

MODE	MODULE AFFECTED	FUNCTION
A	output	Continuous output. Load data to the output module without checking for any flag.
B	output	Flag output. Check for output modules's "Ready" flag (bit7) and loads DBR with data.
C	output	Interrupt output. Program sets output module's interrupt enable. On interrupt it loads output modules DBR with data.
D	output	Delayed output. The program clears output modules CSR bit0; loads DBR with data; and then sets CSR bit0.
E	output	Start Auto Mode Master. This program may be used to generate data a master to send to a slave CPU running slave mode. A table of data is outputted in modes A-D automatically. Mode "K" of program MD-11-DZDRH (for PDP-11's) or MD-08-DHDRH (for PDP-8's) must have been selected. Output word must be cabled to input word of DR11-M or DR8-ED. "Start Mode" is typed at the beginning of this mode. The program stalls waiting to link up with the input module its connected to. After link-up, "Immediate Send Mode" is typed. In this mode data from the table is sent directly out as soon as the input module

is ready to receive it. If the input module detects an error in the data it receives, it can request a re-transmit of the previous data by setting the stat out bit thus setting the input module's stat in bit. If this occurs "Resending data xxxxx" will be typed. After this mode, "Burst Send Mode" is typed. In this mode the output module is allowed to interrupt for table data. "END PASS" is typed after completion.

Warning: Before starting the input or output module into "Start Mode", both CPU's should be running and waiting at "4.4.2 Mode?". At no time should the program be allowed to enter "Start Mode" and the other CPU initialized.

This mode was not designed to diagnose this module. Only data errors are checked. If any of the control signals fail the program could "hang" waiting for the signal. Maximum time between timeouts should not exceed 3 minutes - if it does, halt the CPU and determine the PC the program is at. Go to the listing for more information. Prints mentioned in the listing can be coordinated against the flow chart.

- F output Output Module Logic Test.
The program performs a brief logic check of the outputs module's CSR and interrupt capacities. A Maintenance loop-back must be installed in the word under test.
- G input Continuous input.
The program continuously reads the input module's DBR.
- H input Flag input.
The program waits for the "Ready" flag (CSR bit7) and then does a DBR read when bit7=1.
- I input Interrupt input.
The program sets interrupt enable and waits until the input module interrupts to read the module's DBR.
- J input Delayed input.
The program clears CSR bit0; reads the DBR; and then sets CSR bit0, thus delaying Data accept from returning.
- K input Start Auto Slave
The program may be used to receive data as a slave to receive data incoming from master CPU running in Master Mode (see Mode E), as data is received it is compared against a table of data. Modes G-J are automatically executed. If data received does not compare to the table, an error is reported.
- Mode "E" of MD-11-DZDRH must be selected in Master CPU. Input word must be cabled to the output word of a DR11-L.
- "Start Mode" is typed at the beginning of this mode. The program stalls waiting to link up with the output module it's connected to. After link-up, "Immediate Send Mode" is typed. Each time data is ready, the data is read and compared against table data. "Burst Send Mode" is next. Here the input module is allowed to interrupt each time it receives data. "End Pass" is typed at completion.
- Warning: Before starting the input or output module into "Start Mode" both CPU's should be

running and waiting at "4.4.2 Mode?". At no time should the program be allowed to enter "Start Mode" and the other CPU be initialized.

This mode was not designed to diagnose this module. Only data errors are checked. If any of the control signals fail the program could "hang" waiting for the signal. Maximum time between timeouts should not exceed 3 minutes - if it does, halt the CPU and determine the PC the program is at. Go to the listing for more information. Points mentioned in the listing can be coordinated against the flow chart.

- | | | |
|---|--------------|--|
| L | input | <p>Input Module Logic Test.</p> <p>The program performs a brief logic check of the input modules CSR and interrupt capacities. A Maintenance loop back connector must be installed in the word under test.</p> |
| M | input-output | <p>Start Auto Master-Slave.</p> <p>Performs Modes E and K on an input and output mode located in the same CPU. The input word on the DR11-M must be cabled to the output word on the DR11-L.</p> |

4.4.8 SOURCE OF DATA?

This question is typed when an output Mode A-D is selected. What the program wants to know is where do you want to get the data that you want to output. Respond by typing a character from A-D followed by a carriage return.

- A Data from the Switch Register
- B Data from the TTY.
In this case, any future octal digits typed on the Keyboard will be assembled as data and each time a carriage return is typed the new data typed will be sent out.
- C Program generated pattern.
The program will automatically generate the data to be outputted in one of 4 ways specified by the operator by question 4.4.10.
- D Data is outputted from a table in core.
Data to be outputted can be found by finding the address of TABLE and looking in the listing.

4.4.9 STARTING PATTERN? (If Source of Data "C" Selected)

Respond by typing the octal representation of the data you wish to start outputting. This data will be modified according to the pattern modifier you select in 4.4.10.

4.4.10 PATTERN MODIFIER? (If source of Data "C" Selected)

Respond by typing a character from "A" to "D" to represent how you want the data that is being outputted to be modified.

<u>MODIFIER</u>	<u>FUNCTION</u>
A	INCREMENT DATA
B	DECREMENT DATA
C	FLOAT ZERO-COMPLEMENT
D	COMPLEMENT

4.4.11 METHOD OF REPORTING DATA?

Respond by typing a character from "A" to "C" to represent the way you wish the data being received by the input module to be reported.

<u>MEANS</u>	<u>METHOD</u>
A	in data lights
B	on TTY-reported each time a change in input data is detected.
C	Error check. An input module is cabled to an output module/data beginning sent to the output module is compared to the data received by the input module-if data is different an error is reported. After each 100 transfers are made, "END PASS" is reported.
D	Error Check. An input module is cabled to an

output module in another CPU. This method only works when the "Source of Data" for the output module is that of the table "D".

On the output module side, you would select "Mode" "B" or "C"; "Source of Data" "D".

On the input module side, you would select "Mode" "H" or "I".

ERRORS detected as in modes "E" and "K".

5.0 OPERATING PROCEDURE LOGIC TEST

5.1 Definition - Switch Register

The program must, at various points in execution, determine what the operator desires to do at these points (such as "halt on error"). Standard procedure was to get this information from the hardware switch register. With the addition of CPU's to this family of computers that have no switch register, a hardware switch register can no longer be assumed. This program uses a software switch register; that is, a location in memory "SWREQ" as the source of "switch register" data.

5.1.1 Loading the Software Switch Register

The software switch register can be altered in two ways:

5.1.2 From the Hardware Switch Register

This way over-rides the other ways. Data loaded into the hardware switch register will be transferred to the software switch register directly each time "switch register" data is needed.

5.1.3 From the User's Terminal

A "ts" (control and letter "S" keys typed simultaneously) will cause the current software switch register to be typed out and allow the user to type in the new value of the switch register.

If a hardware switch register exists, and the user desires to use the software switch register, the user should leave the hardware switch register clear (all switches down).

NOTE 1

When using a pure software switch register "halt on error" is re-defined to "hold on error". If the CPU were allowed to halt, then core data could not be examined and the switch register data could not be altered. "Hold on error" sets in a loop testing software SWR bit 15, allowing TTY interrupts.

NOTE 2

Some means of examining CPL memory must be incorporated into the keyboard handler must be made so that the operator on a switch register - less CPU can examine core locations while in a "hold on error". The program uses a "tE xxxxx" where xxxxx is the location to be examined.

5.1.4 Switch Register Function.

SW15=1 or up Halt on error.
SW14=1 or up Loop on test.
SW13=1 or up Inhibit printout of error.
SW11=1 or up Inhibit subtest iterations.

5.2 Scope Loops

If an error occurs and the user wishes to scope the error, he (or she) should set SW15=1 to halt on error, then when the program halts on error, SW15=0, set SW14=1 to (loop on current test), SW13=1 (to inhibit error printout) and press continue on the CPU's CONSOLE.

5.2 Program AND/OR Operator Action

The first pass through the program will be made with iterations inhibited. Successive passes will enable iterations if SW11=0. "END PASS" is printed at the end of a pass. The vector and priority of each input or output under test will be typed when program is initially run or if the vector or priority is changed during a pass.

6.0 ERRORS *****

6.1 Error Typeout

Print out varies with the error detected. The error PC typed out is the actual location of the error call. Error typeouts are preceded with THE ADDRESS of the error, if information is desired, reference the listing in respect to the PC (ADDRESS) typed out.

6.2 Non-Standard Errors

"TRAPPED TO LOCATION:XXXXXX FROM LOCATION:YYYYYY"

indicates:

1. An INPUT or OUTPUT module interrupted to a wrong vector.
- 2

```

00000000
00000001
00000002
00000003
00000004
00000005
00000006
00000007
00000008
00000009
0000000A
0000000B
0000000C
0000000D
0000000E
0000000F
00000010
00000011
00000012
00000013
00000014
00000015
00000016
00000017
00000018
00000019
0000001A
0000001B
0000001C
0000001D
0000001E
0000001F
00000020
00000021
00000022
00000023
00000024
00000025
00000026
00000027
00000028
00000029
0000002A
0000002B
0000
```

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
7

MAINDEC-11-DZDRH-A MACY11 27(732) 21-SEP-76 09:52 PAGE 4
 DZDRH.P11 BASIC DEFINITIONS

113	004000	BIT11= 4000
114	002000	BIT10= 2000
115	001000	BIT09= 1000
116	000400	BIT08= 400
117	000200	BIT07= 200
118	000100	BIT06= 100
119	000040	BIT05= 40
120		

212					
213	001210	000000	RVT:	.WORD	0
214	001212	000000	RVT2:	.WORD	0
215					
216	001214	000060	TKV:	.WORD	

.SBTTL ERROR POINTER TABLE

;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
 ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
 ;*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
 ;*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
 ;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

K02

MAINDEC-11-DZDRH-A MACY11 27(732) 21-SEP-76 09:52 PAGE 9
 DZDRH.P11 INITIALIZE THE COMMON TAGS

```

364 002012 120027 000131      3$:  CMPB    RO,#'Y      ;IS ANSWER "YES"?
365 002016 001327              BNE     1$      ;IF NOT REASK QUESTIKN
366
367 002020 104401 002026              TYPE    71$      ;;TYPE ASCIZ STRING
368 002024 000402              BR       70$      ;;GET OVER THE ASCIZ
369              ;;71$: .ASCIZ  #YES#
370 
```

```

420
421 002420
422 002420 104401 002426
423 002424 000426
424
425 002502
426 002502 104410
427 002504 012600
428 002506 004737 013264
429 002512 103742
430
431 002514 104401 002522
432 002520 000416
433
434 002556
435 002556 010137 001144
436 002562 062701 000002
```

476

:*THIS SECTION ENTERED FROM START AT 210 OR 1C

```

482
483 003060 012706 001100          RSTART: MOV    #STACK,SP
484 003054 005037 001100          CLR    $PASS
485 003070 005037 001112          CLR    $ERTTL          ;CLEAR ERROR COUNT.
486 003074 005037 001304          CLR    E
```

MAINDEC-11-DZDRH-A MACY11 27(732) 21-SEP-76 09:52 PAGE 12
 DZDRH.P11 PROGRAM RESTART

532	003376	121027	000114	3\$:	CMPB	(RO), #'L	; INPUT MODULE LOGIC TEST "L"?
533	003402	001002			BNE	4\$	
534	003404	000137	010052		JMP	IMLT	; YES-G

```

588      003712      104407      001226      45:      MOV      #TABLE,TEXTPT      ;POINT TO BEGINNING OF TABLE.
589      003712      012600      001236      45:      CLR      TTYPAT      ;CLEAR TTY CHANGE PATTERN.
590      003714      012600      001236      45:      MOV      #0,2XCSRA
591      003716      111037      001226      45:      MOV      #0,2XCSRA
5
```

```

644
645 004106 123727 001224 000102 2$: CMPB PATFR0,'B ;PATTERN FROM TTY?
646 004114 001002 BNE 3$ ;NO-CONTINUE.
647 004116 000137 004042 JMP OUTWR ;YES-DONE AUTOMATICALLY.
648 ;PATTERN FROM THE TABLE
649 004122 017737 175104 001230 3$: MOV JTEXTP,PATRN ;LOAD NEW PATTERN
650 004130 062737 00
```

MAINDEC-11-DZDRH-A MACY11 27,732) 21-SEP-76 09:52 PAGE 15
 DZDRH.P11 OUTPUT MODULE ROUTINES

```

700 004300 005237 001230      INCP:  INC      PATTRN      ; INCREMENT.
701 004304 000773              BR      PMEXT
702
703 004306 005337 001230      DECP:  DEC      PATTRN      ; DECREMENT
704 004312 000770              BR      PMEXT
705
706 004314 005137 001230      COMP:  COM      PATTRN      ; COMPLEMENT
707 004320 000765              BR      PMEXT
708

```

E03

MAINDEC-11-DZDRH-A MACY11 27.732) 21-SEP-76 09:52 PAGE 16
 DZDRH.P11 OUTPUT MODULE ROUTINES

```

756 004500 004737 004614 JSR PC,TSTFLG ;CHECK REXMIT REQ.
757
758 004504 005037 177776 CLR PS ;NOW ALLOW INTRs. AGAIN.
759 004510 000767 BR 15
760
761 ;*WILL INTEPRUPT TO HERE
762
763 004512 022626 2$: POPSP2
764
765 004514 013777 001230 174454 MOV PATRN,DXDBRA ;RESET STACK.
766 0045
```

```

812 005020          66$:
813
814 005020 104407          RDLIN          ;GET RESPONSE
815 005022 012600          MOV          (SP)+,RO ;ADDR OF RESPONSE ON STACK
816 005024 011037 001240  MOV          (RO),PATOUT ;SAVE
817 005030 121027 000101  CMPB         (RO),#'A ; OR= TO 'A'
818 005034 002743          BLT          1$
819 005036 121027 000104  CMP
```

```

868 005140 017737 174042 001242 GMODE: MOV      @RDBRA,PATTIN ;READ INPUT MODULE.
869 005146 000137 005102          JMP      INPRD
870
871                                     ;*ROUTINE TO READ WHEN FLAG IS SET.
872                                     ;*>>WARNING<< IF DISPLAYING RESULTS IN DATA LIGHTS, A RESET
873                                     ;*(SYSTEM INITIALIZE) IS USED AND COULD CLEAR THE FLAG.
874
875 005152 105777 174026          HMODE: TSTB@RCSRA          ;FLAG SET?
876 005156 100375          SPL
```

H03

MAINDEC-11-DZDRH-A MACY11 27(732) 21-SEP-76 09:52 PAGE 19
 DZDRH.P11 INPUT MODULE ROUTINES

```

924 005322 001002      BNE      11$
925 005324 000137 005624      JMP      DISTTY      ;YES-GO REPORT
926
927 005330 123727 001240 000104 11$:      CMPB      PATOUT,#'D
928 005336 001002      BNE      2$
929 005340 000137 005700      JMP      CIO
930
931 005344 005237 001104      2$:      INC      $ICNT      ;OUTPUT
```

MAINDEC-11-DZDRH-A MACY11 27(732) 21-SEP-76 09:52 PAGE 20
 DZDRH.F11 INPUT MODULE ROUTINES

```

980 005574 162737 000002 001234 1C3: SUB #2, TABLEP ;POINT TO ERROR DATA
981 005602 000762 BR 8$ ;EXIT
982
983 ;*DISLGH ROUTINE TO DISPLAY DATA
984
985 005604 013700 001242 DISLGH: MOV PATTIN, R0 ;DISPLAY USING DATA LIGHTS
986 005610 104417 RESETC ;DISPLAY
987 005612 052777 000100 1
```

:036
1037

;*STARTS ROUTINE TO START SLAVE AUTO MODE.

```

1041
1042 005770 005737 001204      STARTS: TST      RCSRA      ;ANY INPUT ADDR. ENTERED?
1043 005774 001402              BEQ      1$          ;NO-REPORT ERROR
1044 005776 000137 006312      JMP      SMST      ;YES-START SLAVE
1045
1046 0
```

K03

MAINDEC-11-DZDRH-A MAC111 27(732) 21-SEP-76 09:52 PAGE 22
 DZDRH.P11 SLAVE-MASTER AUTO SEND MODES

```

1092 006232 012737 006262 000014 1S:  MOV  #STTRP,2#14  ;SET TO TRAP TO SLAVE HANDLER
1093 006240 012637 001246          MOV  (SP)+,PTRM  ;SAVE MASTER PC.
1094 006244 012637 001252          MOV  (SP)+,SMSST  ;SAVE MASTER STATUS
1095 006250 013746 001254          MOV  SSLST,-(SP)  ;USE SLAVE STATUS
1096 006254 013746 00125
```


N03

MAINDEC-11-DZDRH-A MACY11 27(732) 21-SEP-76 09:52 PAGE 25
DZDRH.P11 SLAVE BURST SEND MODE

```

1260
1261 007122 104414 SP37: SWCAL
1262 007124 032777 020000 172006 BIT #BIT13,ASWR ;INHIBIT ERROR TYPEOUT?
1263 007132 001042 BNE 15 ;YES-THEN SKIP
1264
1265 007134 104401 007142 TYPE 65$ ;TYPE ASCIZ STRING
1266 007140 000414 BR 64$ ;GET OVER THE ASCIZ
1267 ;:65$: .ASCIZ <
```


J05

MAINDEC-11-DZDRH-A
DZDRH.P11 T31

MACY11 27(732) 21-SEP-76 09:52 PAGE 47
*THAT THE OUTPUT MODULE WON'T INTERRUPT IF CPU IS AT SAME PRIORITY

2475	014274	000002	000002
2476	014276	000004	000004
2477	014300	000010	000010
2478	014302	000020	000020</

