Will Gasser | wbg0023
COMP 4300
Lab 3

Test Descriptions

1. **dlx_register – Clock High Update:** Tests the register with the input set to "DEADBEEF" while the clock is high. The output updates after a 10 ns delay, confirming proper data propagation.

2. **dlx_register – Clock Low Freeze:** Tests the register's behavior when the clock is held low while the input changes to "CAFEBABE." The output remains unchanged, demonstrating that the register correctly freezes the output when the clock is low.

3. **dlx_register – New Value on Rising Clock:** Tests the register by updating the input to "12345678" with the clock going high again. The waveform shows the output updating after a 10 ns delay, indicating that the register captures the new value properly.

4. **reg_file – Write Operation:** Tests the register file by writing the value "DEADBEEF" into register 3 on a rising clock edge. The immediate storage of the value is visible, ensuring that the write operation works as specified.

5. **reg_file – Read Operation:** Tests reading from register 3 with the read control active. The waveform displays the stored "DEADBEEF" value at the output after a 15 ns delay, confirming the correct retrieval of data.

6. **Two-Way Mux – Select Input 0:** Tests the two-way multiplexer with the select signal set to '0,' routing input_0 to the output. The 5 ns delay between the input and output changes is clearly visible.

7. **Two-Way Mux – Select Input 1:** Tests the multiplexer with the select signal changed to '1,' thereby routing input_1 to the output. The resulting waveform confirms that the mux responds correctly to the control signal.

8. **Three-Way Mux – Valid Select (Code "00"):** Tests the three-way multiplexer by applying a select code "00," which directs input_0 to the output. The waveform shows that input_0 is passed through with the expected propagation delay.

9. **Three-Way Mux – Valid Select (Code "01"):** Tests the multiplexer with a select code "01," routing input_1 to the output. The waveforms verify that the component selects the

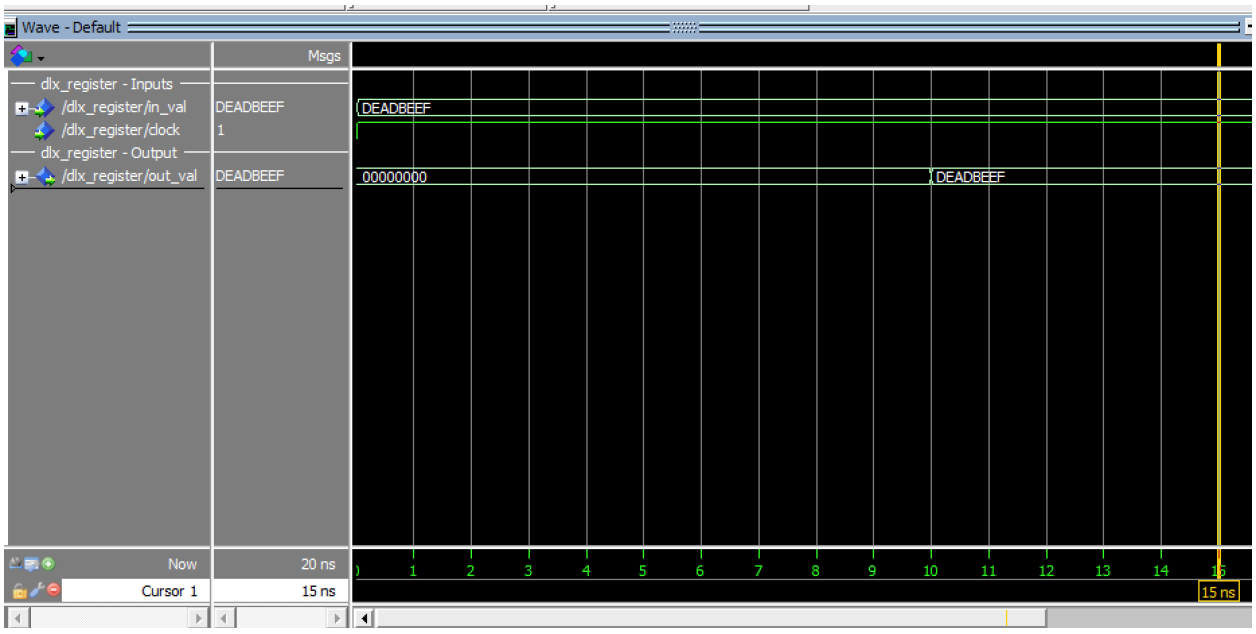
proper input based on the control code.

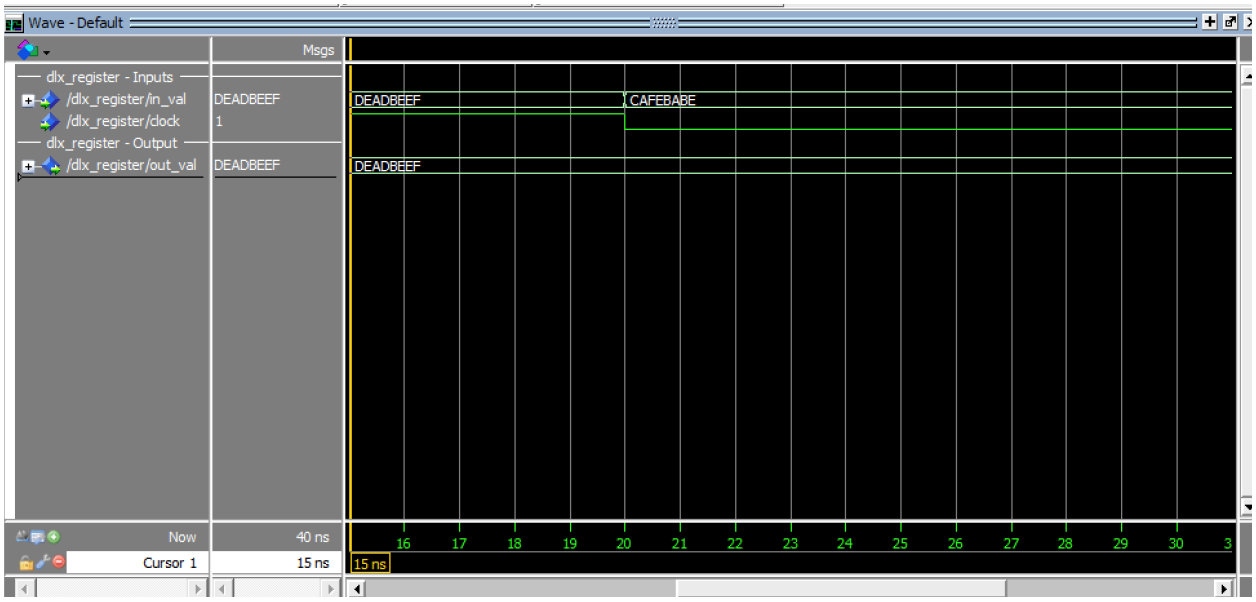
10. **Three-Way Mux – Valid Select (Code “10”):** Tests the multiplexer using a select code “10,” which selects input_2. The output waveform clearly reflects input_2 with the appropriate delay.

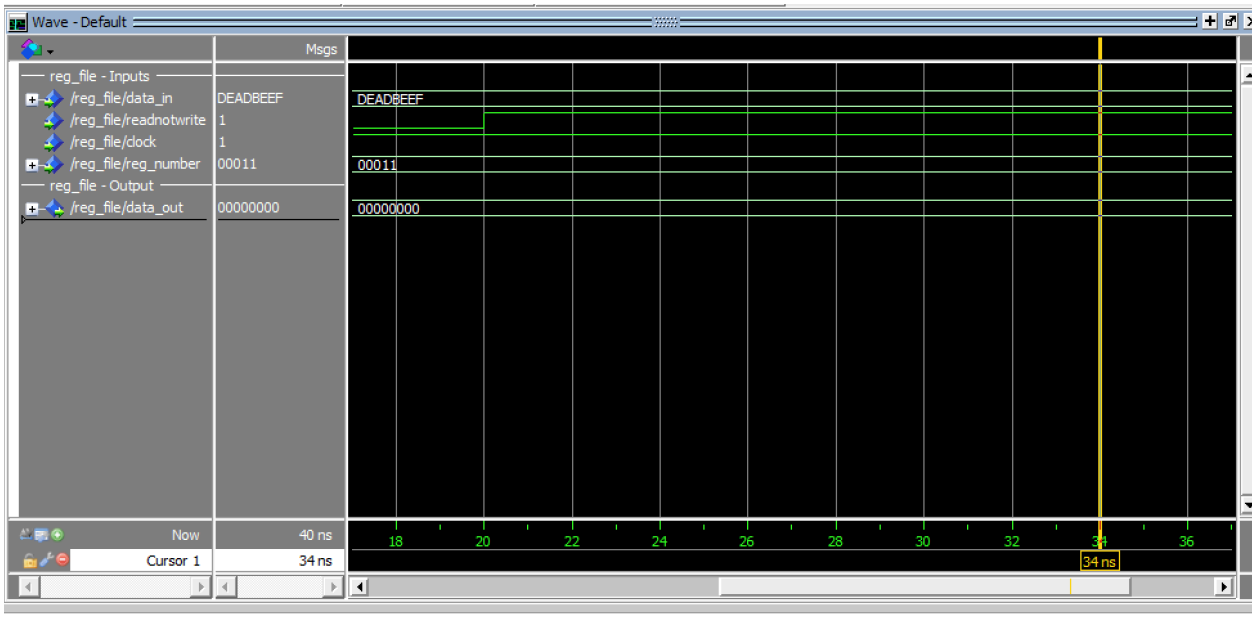
11. **Three-Way Mux – Invalid Select (Code “11”):** Tests the multiplexer’s default behavior when an invalid code “11” is applied, where input_0 is routed to the output. The waveform confirms that the component handles the error case as expected.

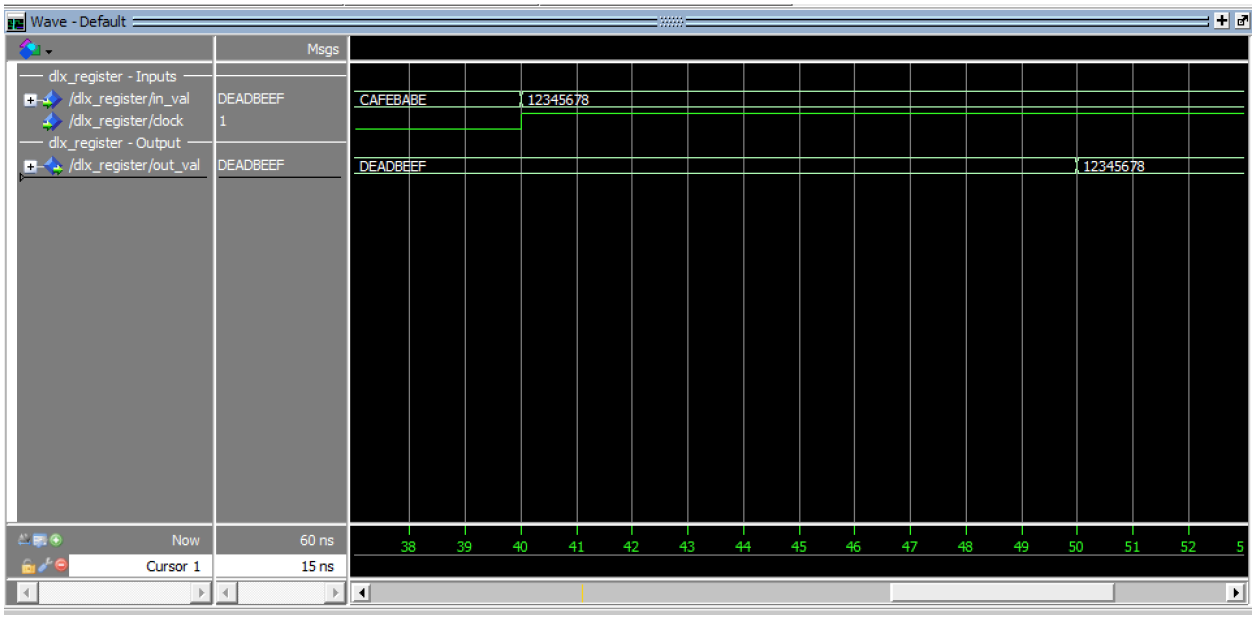
12. **Pc Incrementer– Increment from Zero:** Tests the incrementer unit by applying an input of “00000000” with the clock high. The output and current_pc signals show that the value is incremented from 0 to 1 after the 15 ns delay.

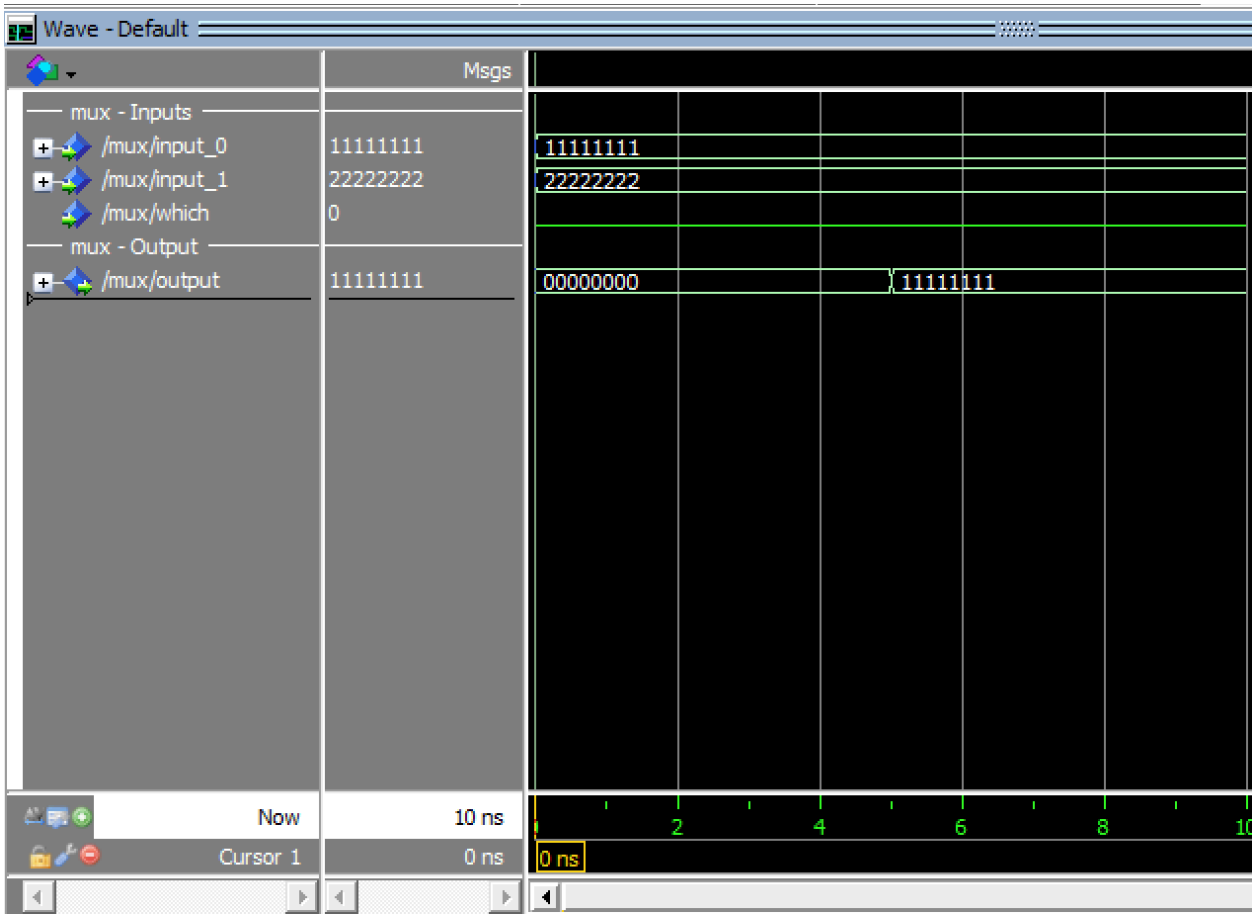
13. **Pc Incrementer– Increment from One:** Tests the incrementer with an input of “00000000” on a rising clock edge. The waveform demonstrates that the output increments from 0 to 1

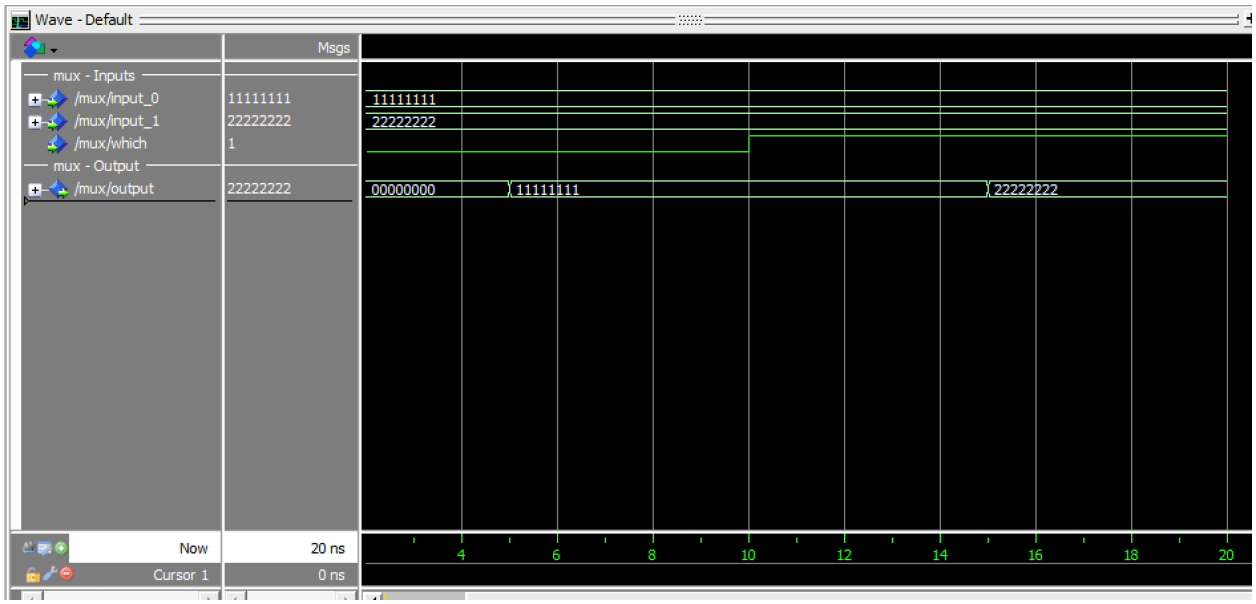

Wave - Default
Msgs
dlx_register - Inputs
/dlx_register/in_val
DEADBEEF
/dlx_register/clock
1
dlx_register - Output
/dlx_register/out_val
DEADBEEF
DEADBEEF
00000000
DEADBEEF
Now
20 ns
Cursor 1
15 ns
15 ns

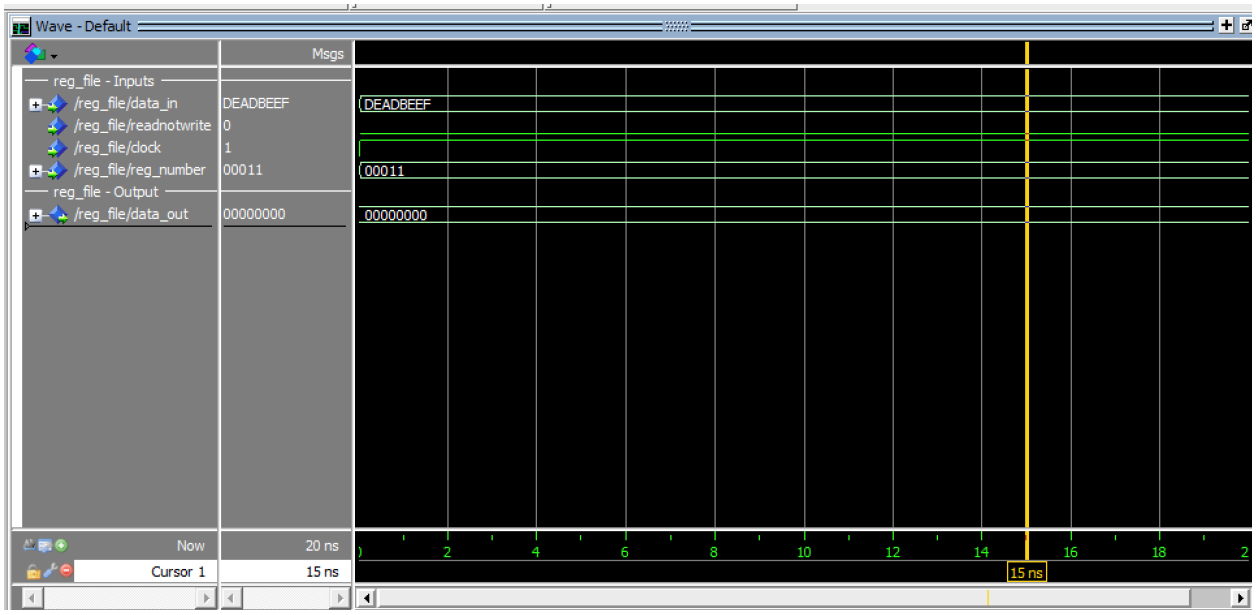

Wave - Default
Msgs
dlx_register - Inputs
/dlx_register/in_val
DEADBEEF
/dlx_register/clock
1
dlx_register - Output
/dlx_register/out_val
DEADBEEF
DEADBEEF
CAFEBABE
DEADBEEF
Now
40 ns
Cursor 1
15 ns
15 ns

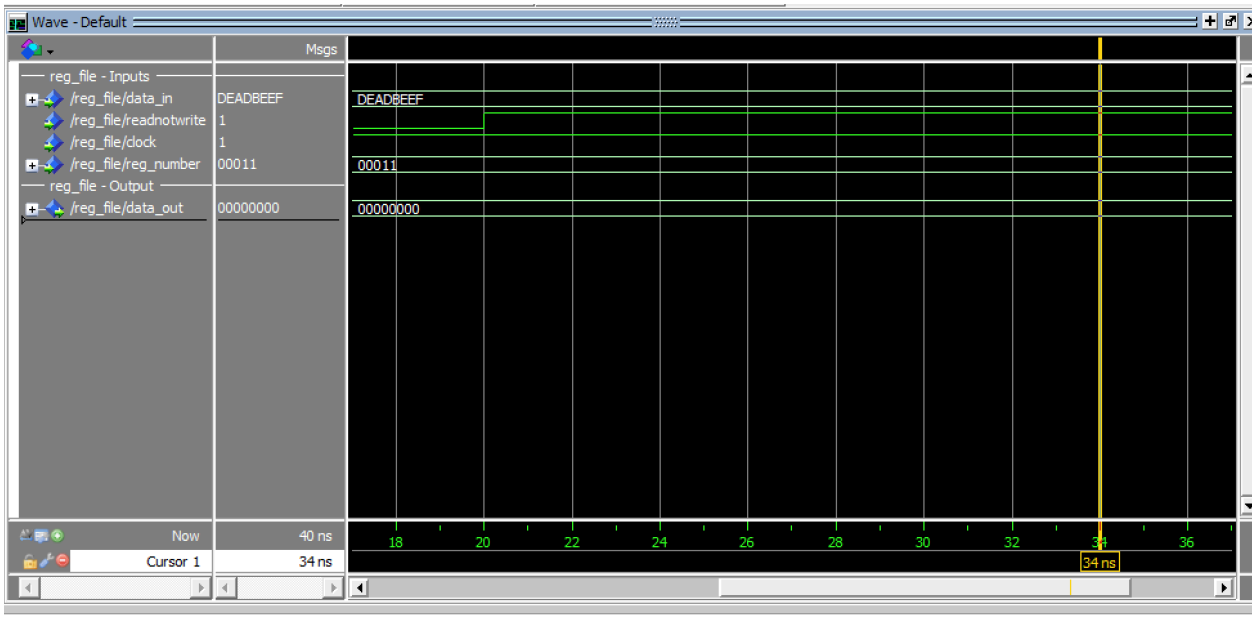

Wave - Default
Msgs
reg_file - Inputs
/reg_file/data_in
/reg_file/readnotwrite
/reg_file/clock
/reg_file/reg_number
reg_file - Output
/reg_file/data_out
DEADBEEF
1
1
00011
00000000
DEADBEEF
00011
00000000
Now
40 ns
Cursor 1
34 ns
18
20
22
24
26
28
30
32
34
36
34 ns

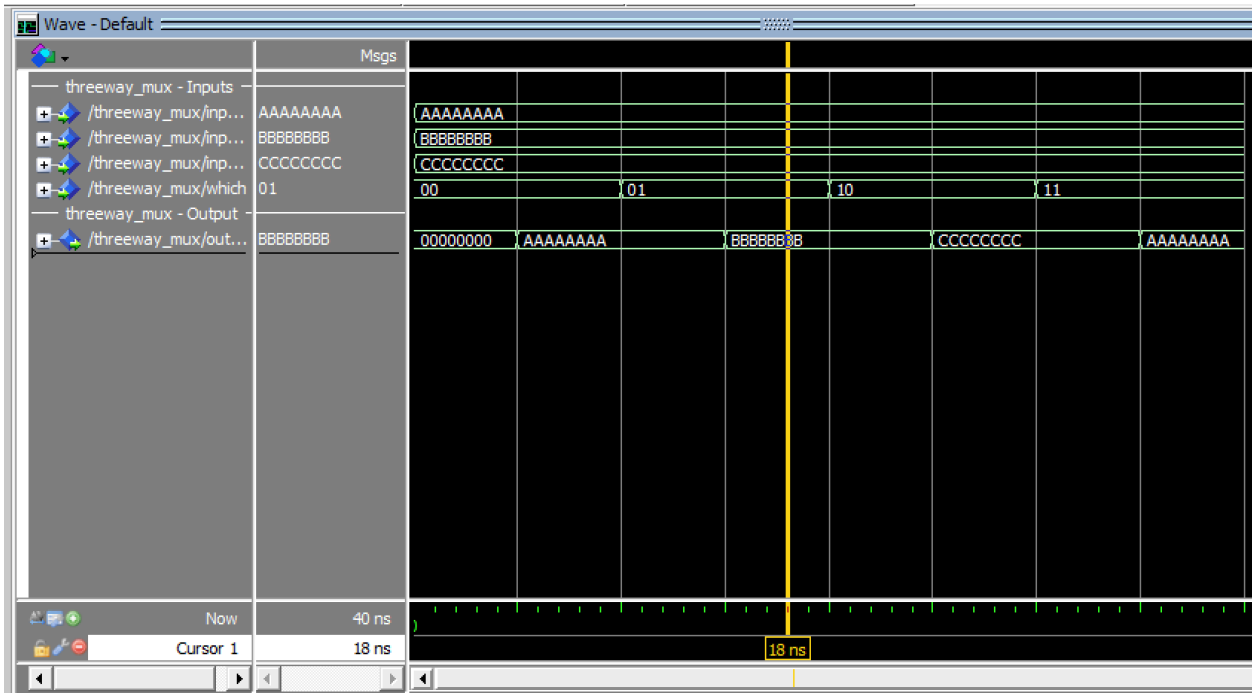

Wave - Default
Msgs
dlx_register - Inputs
/dlx_register/in_val
/dlx_register/clock
dlx_register - Output
/dlx_register/out_val
DEADBEEF
1
DEADBEEF
CAFEBABE
12345678
DEADBEEF
12345678
Now
60 ns
Cursor 1
15 ns
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

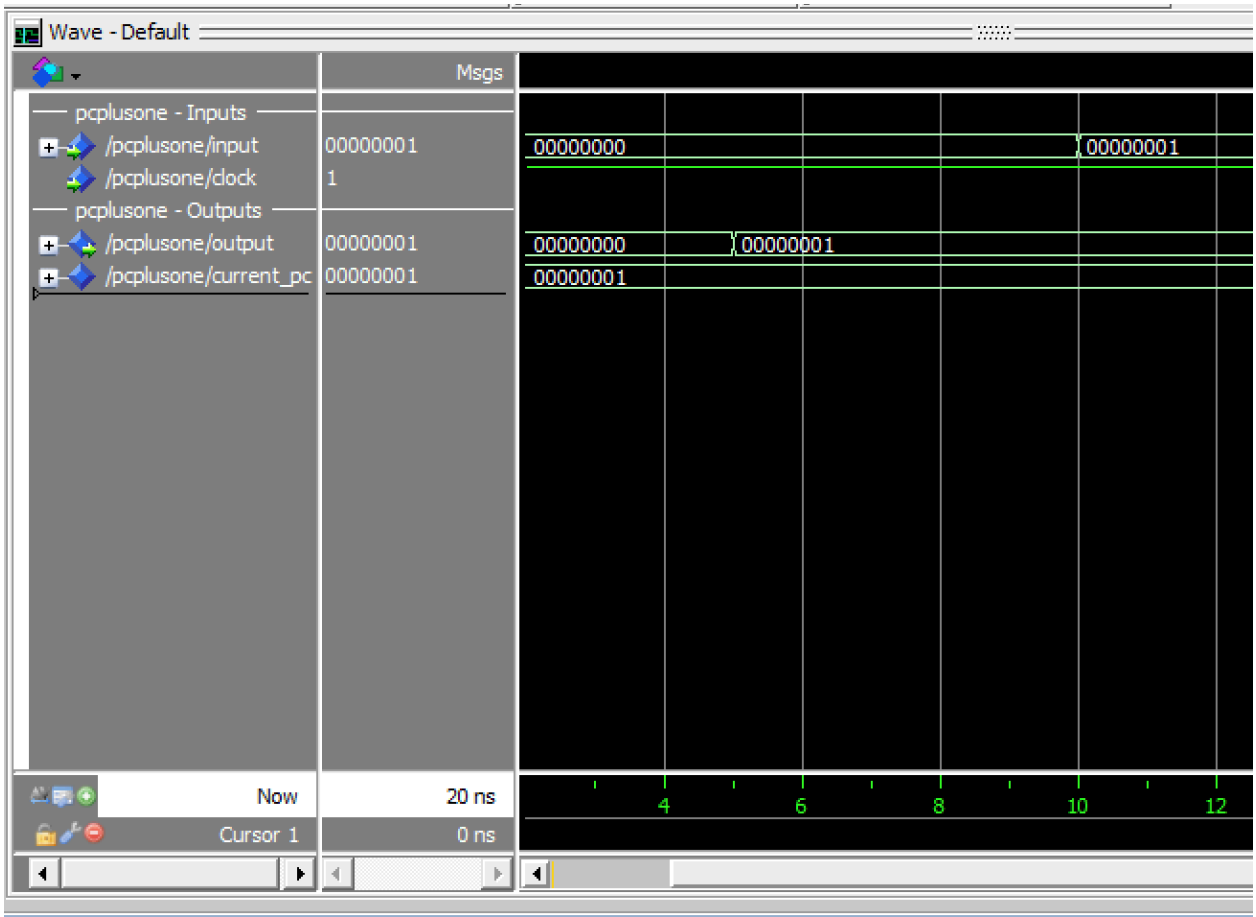

Wave - Default
Msgs
mux - Inputs
/mux/input_0
11111111
/mux/input_1
22222222
/mux/which
0
mux - Output
/mux/output
11111111
00000000
11111111
Now
10 ns
Cursor 1
0 ns
0 ns
2
4
6
8


Wave - Default
Msgs
mux - Inputs
/mux/input_0
11111111
/mux/input_1
22222222
/mux/which
1
mux - Output
/mux/output
22222222
11111111
22222222
00000000
Now
20 ns
Cursor 1
0 ns
4
6
8
10
12
14
16
18
20


Wave - Default
Msgs
reg_file - Inputs
/reg_file/data_in
DEADBEEF
/reg_file/readnotwrite
0
/reg_file/clock
1
/reg_file/reg_number
00011
reg_file - Output
/reg_file/data_out
00000000
DEADBEEF
00011
00000000
Now
20 ns
Cursor 1
15 ns
2
4
6
8
10
12
14
16
18
15 ns


Wave - Default
Msgs
reg_file - Inputs
/reg_file/data_in
DEADBEEF
/reg_file/readnotwrite
1
/reg_file/clock
1
/reg_file/reg_number
00011
reg_file - Output
/reg_file/data_out
00000000
DEADBEEF
00011
00000000
Now
40 ns
Cursor 1
34 ns
18
20
22
24
26
28
30
32
34
36
34 ns


Wave - Default
Msgs
threeway_mux - Inputs
/threeway_mux/inp...
/threeway_mux/inp...
/threeway_mux/inp...
/threeway_mux/which
threeway_mux - Output
/threeway_mux/out...
AAAAAAAA
BBBBBBBB
CCCCCCCC
01
BBBBBBBB
00
01
10
11
00000000
AAAAAAAA
BBBBBBBB
CCCCCCCC
AAAAAAAA
Now
40 ns
Cursor 1
18 ns
18 ns


Wave - Default
Msgs
pcplusone - Inputs
/pcplusone/input
/pcplusone/clock
pcplusone - Outputs
/pcplusone/output
/pcplusone/current_pc
00000001
1
00000001
00000001
00000000
00000001
00000000
00000001
00000001
Now
20 ns
Cursor 1
0 ns
4
6
8
10
12