Department of Electrical, Computer & Biomedical Engineering
Faculty of Engineering & Architectural Science

Programs: Computer Engineering

Course Number	**COE608**
Course Title	**Computer Organization and Architecture**
Semester/Year	**Summer 2020**
Instructor	**Patrick Siddavaatam**

Lab Report No.	**3a**

Lab Title	**32-bit ALU Design and Simulation**

Section No.	**011**
Group No.	
Submission Date	**13 July 2020**
Due Date	**17 July 2020**

Name	Student ID	Signature*
Duanwei Zhang	*500824903*	DZ
Xinyu Hadrian Hu	500194233	XHH

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:*
www.ryerson.ca/senate/current/pol60.pdf.

Name: Xinyu Hadrian Hu, and Duan Wei Zhang

Student Number: 500194233, and 500824903

Date: June 26, 2020

COE 608: Computer Architecture and Design

COE 608: Lab 3, Part 1 Report

Purpose

The purpose of this lab is to generate the components of a 32-bit ALU, with six different operations. The addition and subtraction units must be done in structural form of VHDL.

Design and Implementation

The goal of the 32-bit ALU is to perform the following operations:

32-bit ALU			
OP Name	**Neg/Tsel**	**ALU-Select**	**Operation Performed**
AND (logical)	0	0 0	Result <= a AND b
OR (logical)	0	0 1	Result <= a OR b
ADD	0	1 0	Result <= a + b
SUB	1	1 0	Result <= a - b
ROL	1	0 0	Result <= a << 1
ROR	1	0 1	Result <= a >> 1

Figure 1: ALU Operations

The first step is to create the adder, since it is not possible to use behavioral model in this lab to create the adder circuit.

Full Adder Truth Table				
A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1
		Sum	A xor B xor C	
		Cout	(A and (B or C)) or (B and C)	

Figure 2: Full Adder Circuit with logical functions

Next, the operation of the two-to-one multiplexer is illustrated below:

2 to 1 Multiplexer					
A	B	X	A and not X	B and X	A and not X or (B and X
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1

Figure 3: Two-to-one multiplexer

Next, the binary-coded-decimal to seven-segment converter is shown below. It is assumed that the seven-segment display is cathode-active.

BCD to 7 segment display		
BCD Code	**7seg Code**	**Symbol**
0 0 0 0	0 0 0 0 0 0 1	0
0 0 0 1	1 0 0 1 1 1 1	1
0 0 1 0	0 0 1 0 0 1 0	2
0 0 1 1	0 0 0 0 1 1 0	3
0 1 0 0	1 0 0 1 1 0 0	4
0 1 0 1	0 1 0 0 1 0 0	5
0 1 1 0	0 1 0 0 0 0 0	6
0 1 1 1	0 0 0 1 1 1 1	7
1 0 0 0	0 0 0 0 0 0 0	8
1 0 0 1	0 0 0 0 1 0 0	9
1 0 1 0	0 0 0 1 0 0 0	A
1 0 1 1	1 1 0 0 0 0 0	B
1 1 0 0	0 1 1 0 0 0 1	C
1 1 0 1	1 0 0 0 0 1 0	D
1 1 1 0	0 1 1 0 0 0 0	E
1 1 1 1	0 1 1 1 0 0 0	F

Figure 4: BCD to 7 segment display Truth Table

One-bit Adder: Functional and Timing Waveforms

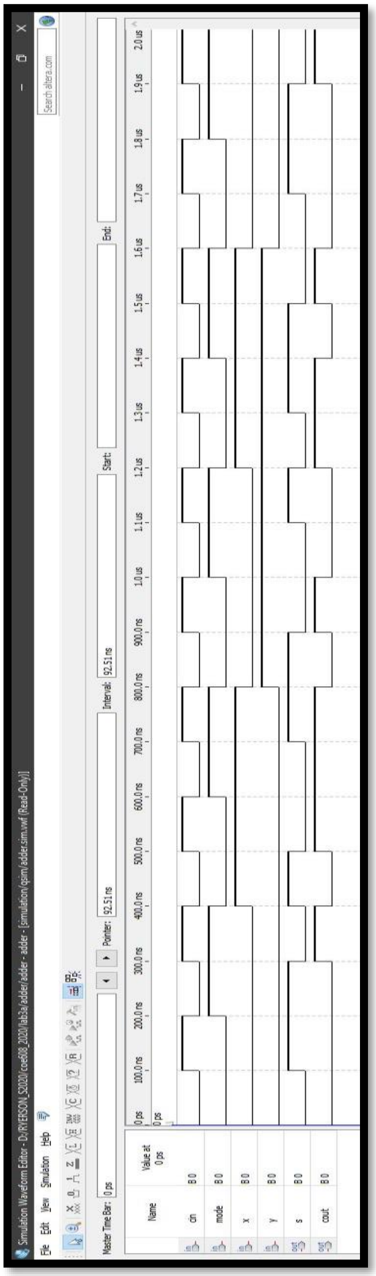


Figure 5: 1-bit Adder Functional Waveform

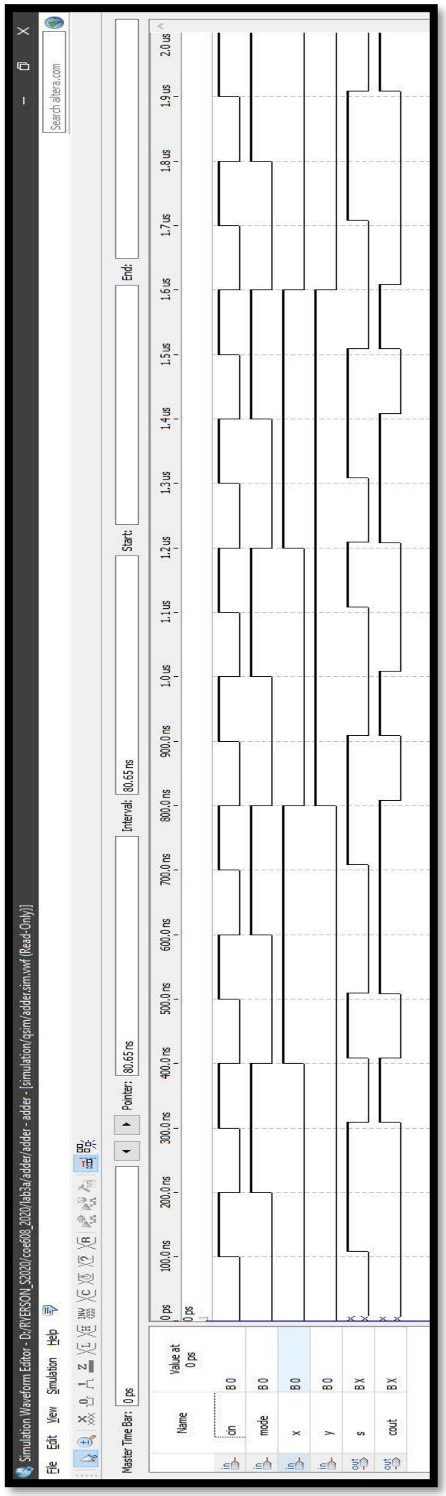


Figure 6: 1-bit Adder Timing Waveform

32-bit Adder: Functional and Timing Waveforms

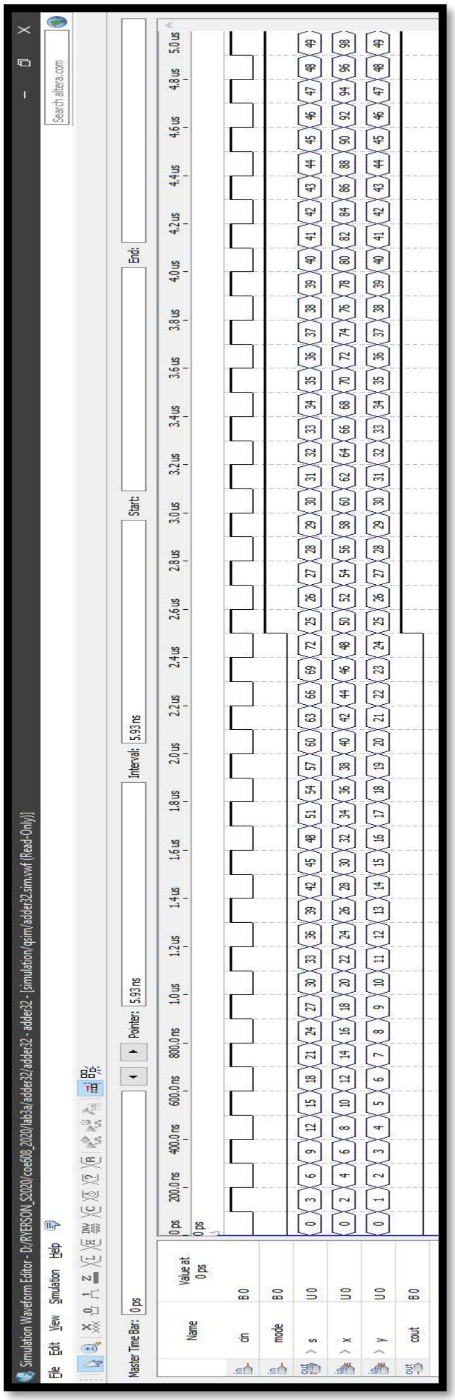


Figure 7: 32-bit Adder Functional Waveform

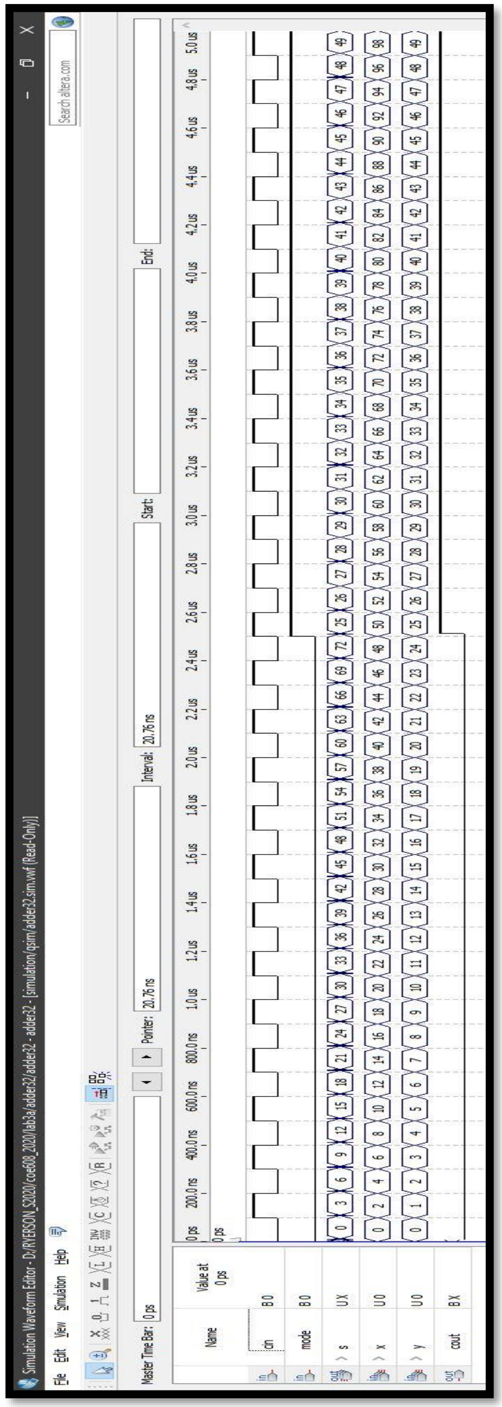


Figure 8: 32-bit Adder Timing Waveform

<u>2 to 1 Multiplexer: Functional and Timing Waveforms</u>

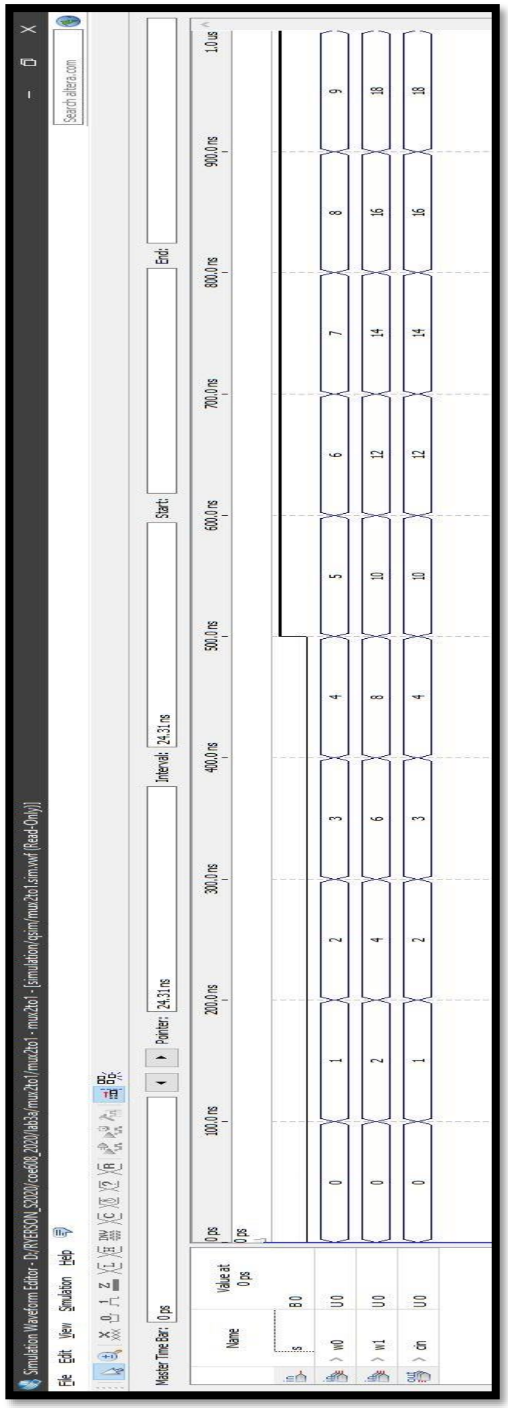


Figure 9: 2-to1 Multiplexer Functional Waveform

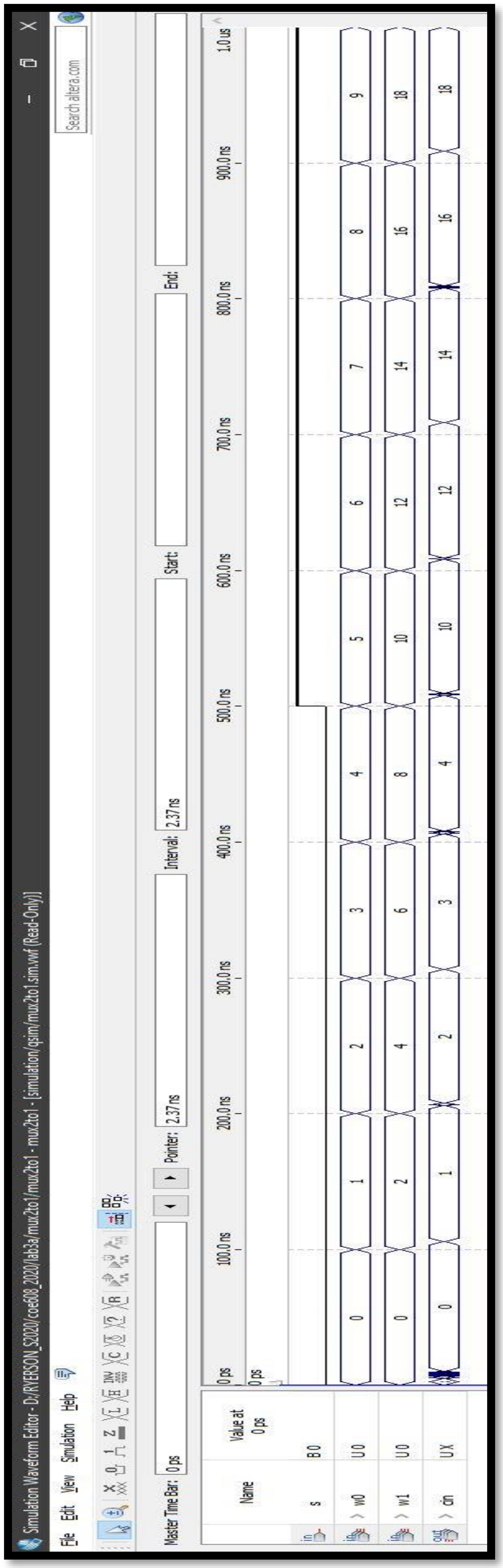


Figure 10: 2-to-1 Multiplexer Timing Waveform

32-bit ALU: Functional and Timing Waveforms

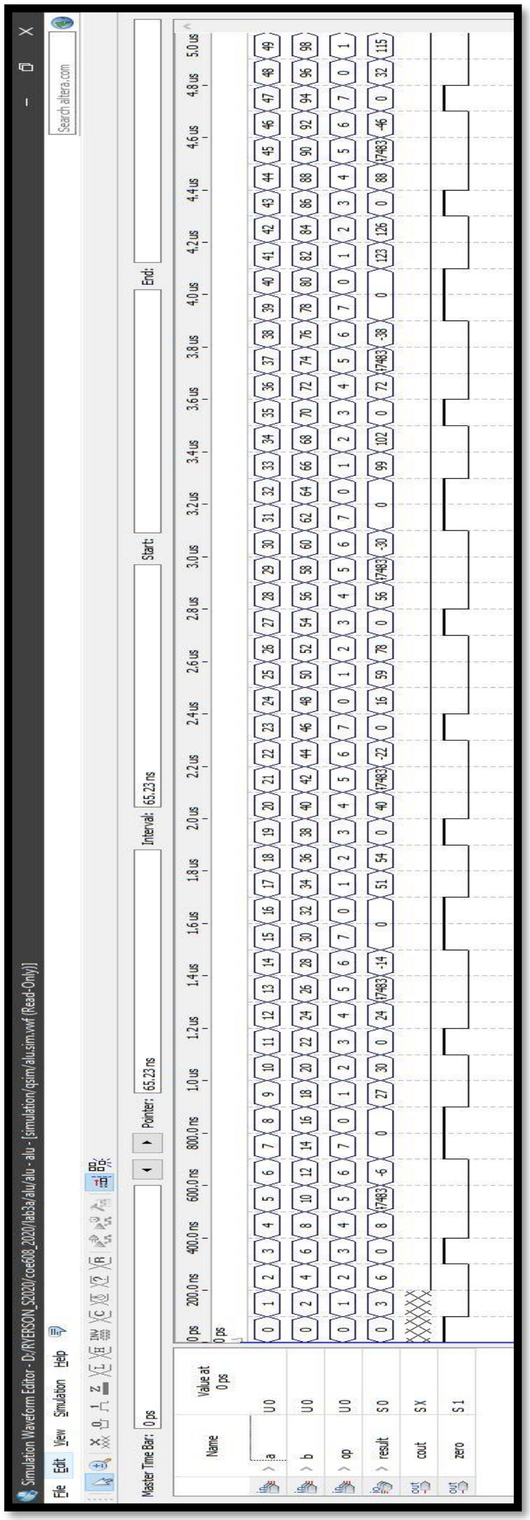


Figure 11: 32-bit ALU Functional Waveform

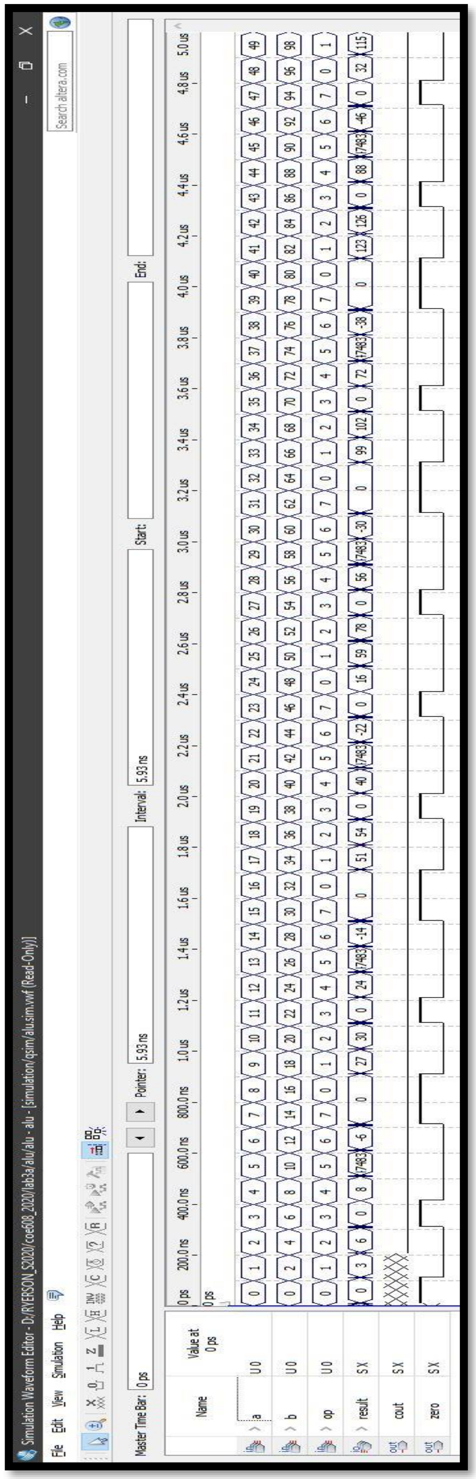


Figure 12: 32-bit ALU Timing Waveform

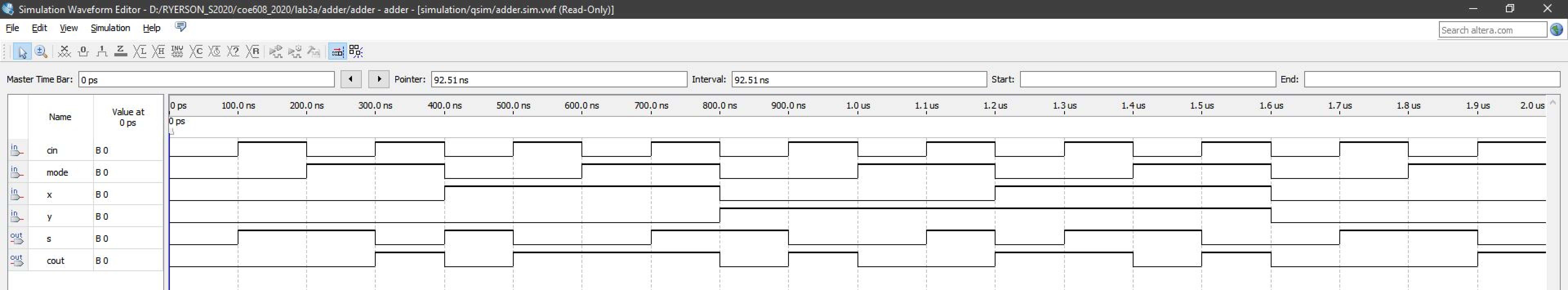
Discussions and Conclusions

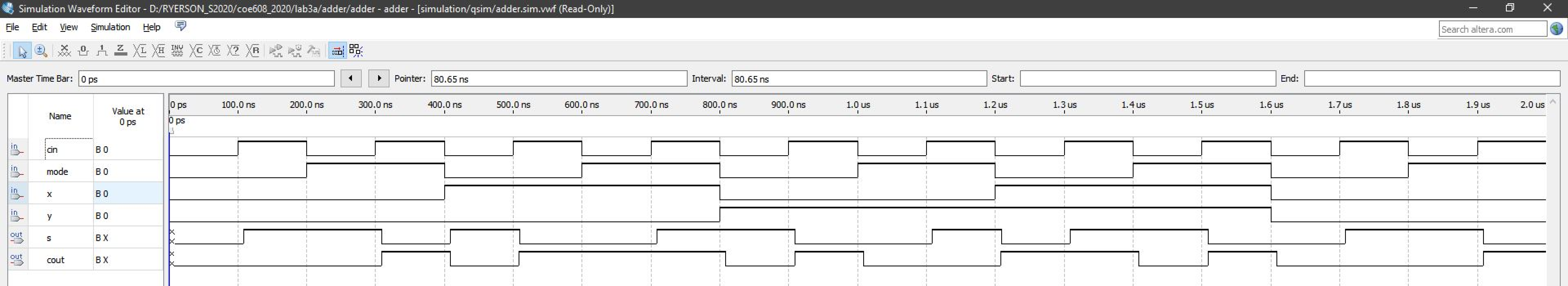
All of the above devices work as expected. The worst-case delay of the 32-bit ALU is 1 ns, for most outputs. The longest delay occurs at the positive or negative signs of the output. The worst delay for the negative output symbol is 40 ns.

Appendices: VHDL Codes for the ALU and other Supporting Devices

The files are attached as PDF to make this document easier to read.

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity adder is
   port (
      x, y: in std_logic;
      mode: in std_logic;
      cin : in std_logic;
      cout : out std_logic;
      s : out std_logic);
end adder;

architecture behavior of adder is
   signal temp : std_logic;
      begin
         process(mode)
            begin
               if (mode = '1') then
                  temp <= (y xor mode);
               else
                  temp <= y;
               end if;
            s <= (x xor temp xor cin) or (x and temp and cin);
            cout <= (x and temp) or (x and cin) or (cin and temp);
         end process;
end behavior;
```

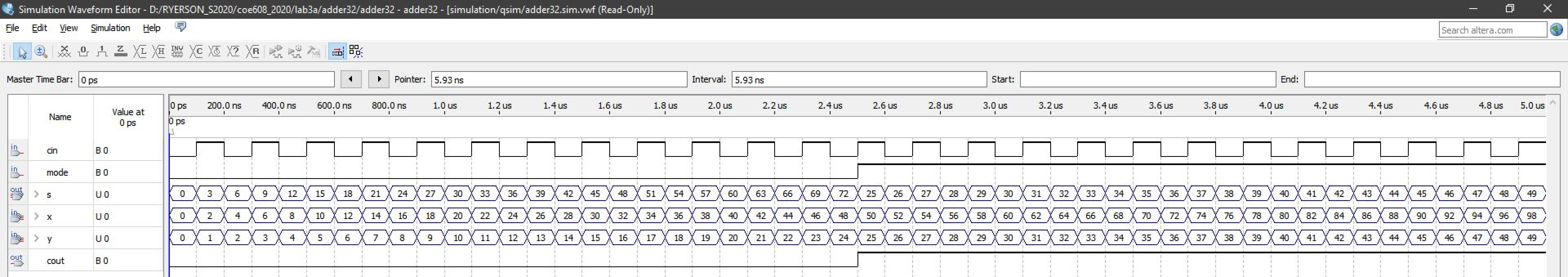

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3a/adder/adder - adder - [simulation/qsim/adder.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 92.51 ns
Interval: 92.51 ns
Start:
End:
Name
Value at 0 ps
0 ps
100.0 ns
200.0 ns
300.0 ns
400.0 ns
500.0 ns
600.0 ns
700.0 ns
800.0 ns
900.0 ns
1.0 us
1.1 us
1.2 us
1.3 us
1.4 us
1.5 us
1.6 us
1.7 us
1.8 us
1.9 us
2.0 us
0 ps
cin B 0
mode B 0
x B 0
y B 0
s B 0
cout B 0

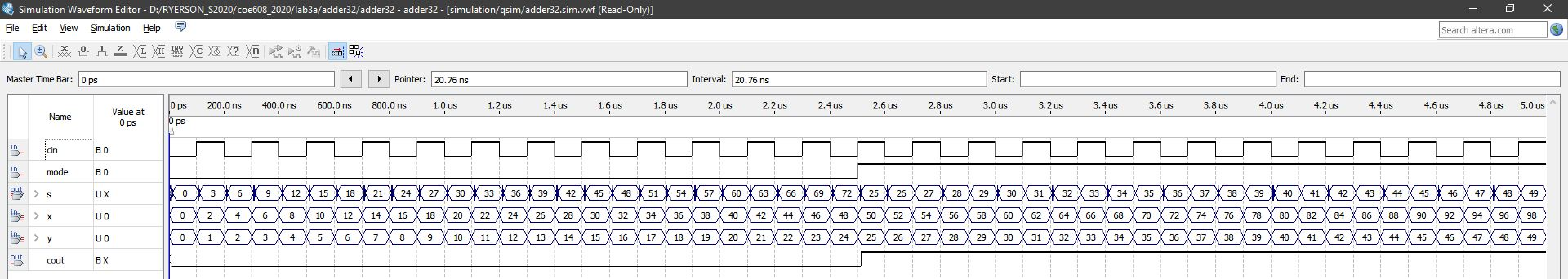


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3a/adder/adder - adder - [simulation/qsim/adder.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 80.65 ns
Interval: 80.65 ns
Start:
End:
Name
Value at 0 ps
0 ps 100.0 ns 200.0 ns 300.0 ns 400.0 ns 500.0 ns 600.0 ns 700.0 ns 800.0 ns 900.0 ns 1.0 us 1.1 us 1.2 us 1.3 us 1.4 us 1.5 us 1.6 us 1.7 us 1.8 us 1.9 us 2.0 us
cin B 0
mode B 0
x B 0
y B 0
s B X
cout B X

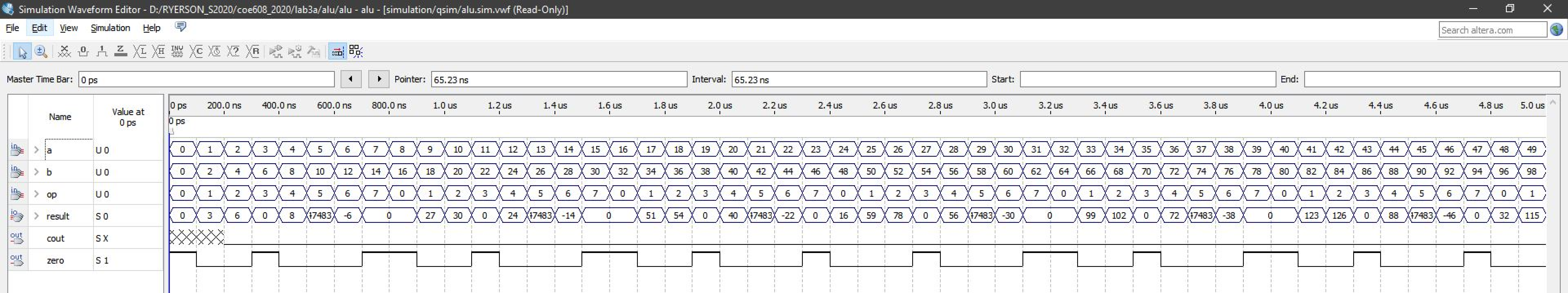
```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

ENTITY adder32 IS
   PORT(cin        :IN STD_LOGIC;
        mode       :IN STD_LOGIC; -- 1 = sub , 0 = add
        x          : in std_logic_vector(31 downto 0);
        y          : in std_logic_vector(31 downto 0);
        s          : out std_logic_vector(31 downto 0);
        cout       :OUT STD_LOGIC
   );
end adder32;

ARCHITECTURE description of adder32 IS
   SIGNAL c31,c30,c29,c28,c27,c26,c25,c24,c23,c22,c21,c20,c19,c18,c17,c16,c15,c14,c13,c12,
c11,c10,c9,c8,c7,c6,c5,c4,c3,c2,c1,c0 : STD_LOGIC;
   COMPONENT adder
      port(
      x : in std_logic;
      y : in std_logic;
      mode : in std_logic;
      cin : in std_logic;
      cout : out std_logic;
      s : out std_logic
      );
      END COMPONENT;
BEGIN
      --st0: adder port map (x(0), y(0), mode, cin, c1, s(0));
      st0: adder port map (x(0), y(0), mode, mode, c1, s(0));
      st1: adder port map (x(1), y(1), mode, c1, c2, s(1));
      st2: adder port map (x(2), y(2), mode, c2, c3, s(2));
      st3: adder port map (x(3), y(3), mode, c3, c4, s(3));
      st4: adder port map (x(4), y(4), mode, c4, c5, s(4));
      st5: adder port map (x(5), y(5), mode, c5, c6, s(5));
      st6: adder port map (x(6), y(6), mode, c6, c7, s(6));
      st7: adder port map (x(7), y(7), mode, c7, c8, s(7));
      st8: adder port map (x(8), y(8), mode, c8, c9, s(8));
      st9: adder port map (x(9), y(9), mode, c9, c10, s(9));
      st10: adder port map (x(10), y(10), mode, c10, c11, s(10));
      st11: adder port map (x(11), y(11), mode, c11, c12, s(11));
      st12: adder port map (x(12), y(12), mode, c12, c13, s(12));
      st13: adder port map (x(13), y(13), mode, c13, c14, s(13));
      st14: adder port map (x(14), y(14), mode, c14, c15, s(14));
      st15: adder port map (x(15), y(15), mode, c15, c16, s(15));
      st16: adder port map (x(16), y(16), mode, c16, c17, s(16));
      st17: adder port map (x(17), y(17), mode, c17, c18, s(17));
      st18: adder port map (x(18), y(18), mode, c18, c19, s(18));
      st19: adder port map (x(19), y(19), mode, c19, c20, s(19));
      st20: adder port map (x(20), y(20), mode, c20, c21, s(20));
      st21: adder port map (x(21), y(21), mode, c21, c22, s(21));
      st22: adder port map (x(22), y(22), mode, c22, c23, s(22));
      st23: adder port map (x(23), y(23), mode, c23, c24, s(23));
      st24: adder port map (x(24), y(24), mode, c24, c25, s(24));
      st25: adder port map (x(25), y(25), mode, c25, c26, s(25));
      st26: adder port map (x(26), y(26), mode, c26, c27, s(26));
      st27: adder port map (x(27), y(27), mode, c27, c28, s(27));
      st28: adder port map (x(28), y(28), mode, c28, c29, s(28));
      st29: adder port map (x(29), y(29), mode, c29, c30, s(29));
      st30: adder port map (x(30), y(30), mode, c30, c31, s(30));
      st31: adder port map (x(31), y(31), mode, c31, cout, s(31));
```

```
end description;
```

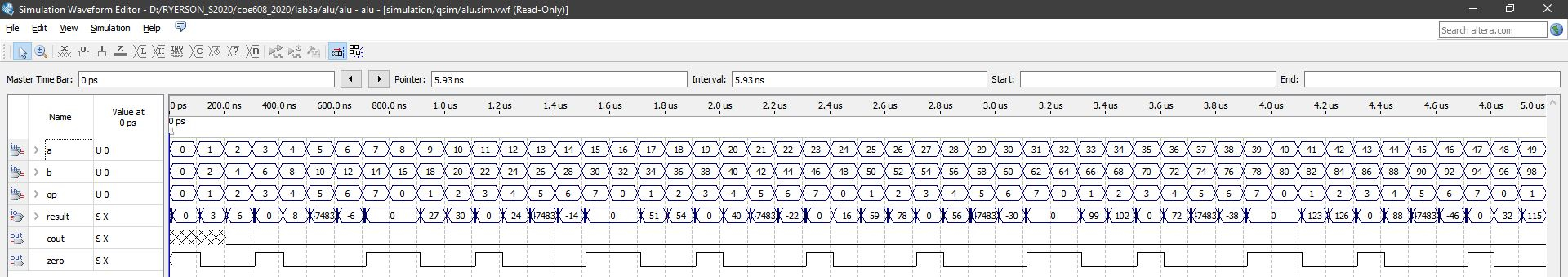

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3a/adder32/adder32 - adder32 - [simulation/qsim/adder32.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 5.93 ns
Interval: 5.93 ns
Start:
End:
Name
Value at 0 ps
0 ps 200.0 ns 400.0 ns 600.0 ns 800.0 ns 1.0 us 1.2 us 1.4 us 1.6 us 1.8 us 2.0 us 2.2 us 2.4 us 2.6 us 2.8 us 3.0 us 3.2 us 3.4 us 3.6 us 3.8 us 4.0 us 4.2 us 4.4 us 4.6 us 4.8 us 5.0 us
cin B 0
mode B 0
s U 0
0 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63 66 69 72 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
x U 0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98
y U 0
0 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
cout B 0


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3a/adder32/adder32 - adder32 - [simulation/qsim/adder32.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 20.76 ns
Interval: 20.76 ns
Start:
End:
Name
Value at 0 ps
0 ps 200.0 ns 400.0 ns 600.0 ns 800.0 ns 1.0 us 1.2 us 1.4 us 1.6 us 1.8 us 2.0 us 2.2 us 2.4 us 2.6 us 2.8 us 3.0 us 3.2 us 3.4 us 3.6 us 3.8 us 4.0 us 4.2 us 4.4 us 4.6 us 4.8 us 5.0 us
cin B 0
mode B 0
s U X 0 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63 66 69 72 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
x U 0 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98
y U 0 0 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
cout B X

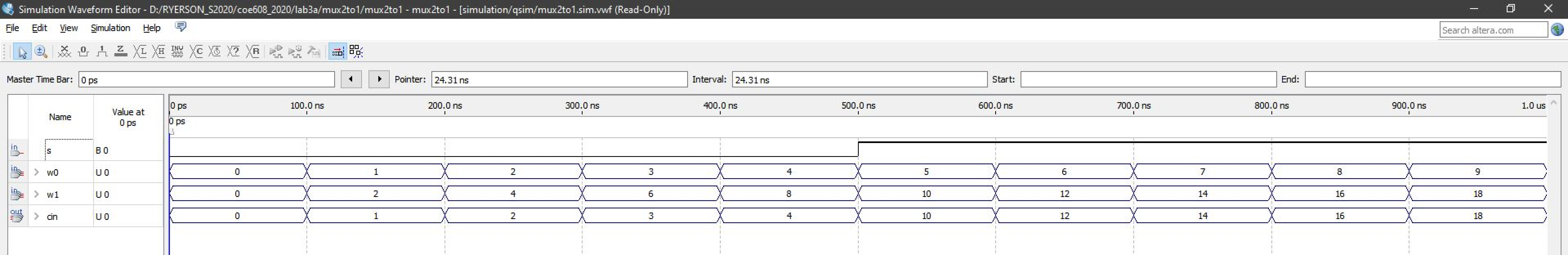
```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use ieee.numeric_std.all;

entity alu is
   port(
      a, b: in std_logic_vector(31 downto 0);
      op: in std_logic_vector(2 downto 0);
      result: inout std_logic_vector(31 downto 0);
      cout, zero: out std_logic);
end alu;

architecture description of alu is
   component adder32 is
      port(
         cin, mode: in std_logic;
         x, y: in std_logic_vector(31 downto 0);
         s : out std_logic_vector(31 downto 0);
         cout: out std_logic);
   end component adder32;
      signal addd: std_logic_vector(31 downto 0);
      signal coutplus: std_logic;
      signal shift: std_logic_vector(31 downto 0);
         begin
            muxadd: adder32 port map (op(2), op(2), a, b, addd, coutplus);
               process (op)
                  begin
                     if op = "000" then
                        result <= a and b;
                     elsif op = "001" then
                        result <= a or b;
                     elsif op = "010" then
                        result <= addd;
                        cout <= coutplus;
                     elsif op = "110" then
                        result <= addd;
                        cout <= coutplus;
                     elsif op = "100" then
                        shift(0) <= a(31);
                        shift(31 downto 1) <= a(30 downto 0);
                        result <= shift;
                     elsif op = "101" then
                        shift(31) <= a(0);
                        shift(30 downto 0) <= a(31 downto 1);
                        result <= shift;
                     else
                        result <= (others => '0');
                     end if;

                     if result = "00000000000000000000000000000000"  then
                        zero <= '1';
                     else
                        zero <= '0';
                     end if;

               end process;

end description;
```



Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3a/alu/alu - alu - [simulation/qsim/alu.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 65.23 ns
Interval: 65.23 ns
Start:
End:
Name
Value at 0 ps
0 ps 200.0 ns 400.0 ns 600.0 ns 800.0 ns 1.0 us 1.2 us 1.4 us 1.6 us 1.8 us 2.0 us 2.2 us 2.4 us 2.6 us 2.8 us 3.0 us 3.2 us 3.4 us 3.6 us 3.8 us 4.0 us 4.2 us 4.4 us 4.6 us 4.8 us 5.0 us
a U 0 0 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
b U 0 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98
op U 0 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1
result S 0 0 3 6 0 8 -17483 -6 0 27 30 0 24 -17483 -14 0 51 54 0 40 -17483 -22 0 16 59 78 0 56 -17483 -30 0 99 102 0 72 -17483 -38 0 123 126 0 88 -17483 -46 0 32 115
cout S X
zero S 1


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3a/alu/alu - alu - [simulation/qsim/alu.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 5.93 ns
Interval: 5.93 ns
Start:
End:
Name
Value at 0 ps
0 ps 200.0 ns 400.0 ns 600.0 ns 800.0 ns 1.0 us 1.2 us 1.4 us 1.6 us 1.8 us 2.0 us 2.2 us 2.4 us 2.6 us 2.8 us 3.0 us 3.2 us 3.4 us 3.6 us 3.8 us 4.0 us 4.2 us 4.4 us 4.6 us 4.8 us 5.0 us
a U 0
0 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
b U 0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98
op U 0
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1
result S X
0 3 6 0 8 -6 0 27 30 0 24 -14 0 51 54 0 40 -22 0 16 59 78 0 56 -30 0 99 102 0 72 -38 0 123 126 0 88 -46 0 32 115
cout S X
zero S X

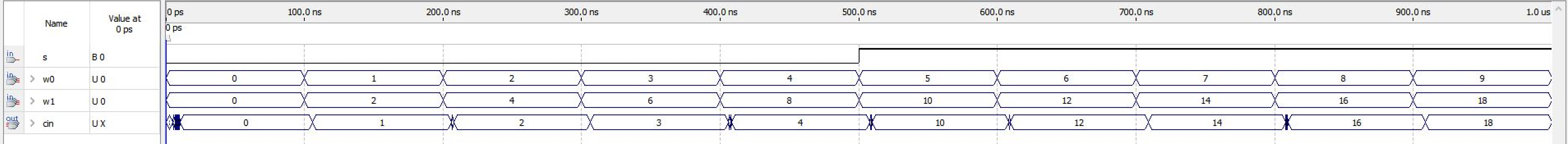
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
   entity mux2too1 is
   port(
      st : in std_logic;
      w0,w1 : in std_logic;
      cin : out std_logic
      );
      end mux2too1;

architecture description of mux2too1 is begin
   --if (st = '0') then -- st if start
      --cin <= '0';
      --elsif(st = '1')then
      --cin <= '1';
   --end if; --st if end
end description;
```


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3a/mux2to1/mux2to1 - mux2to1 - [simulation/qsim/mux2to1.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 24.31 ns
Interval: 24.31 ns
Start:
End:
Name
Value at 0 ps
0 ps
100.0 ns
200.0 ns
300.0 ns
400.0 ns
500.0 ns
600.0 ns
700.0 ns
800.0 ns
900.0 ns
1.0 us
s B 0
w0 U 0 0 1 2 3 4 5 6 7 8 9
w1 U 0 0 2 4 6 8 10 12 14 16 18
cin U 0 0 1 2 3 4 10 12 14 16 18

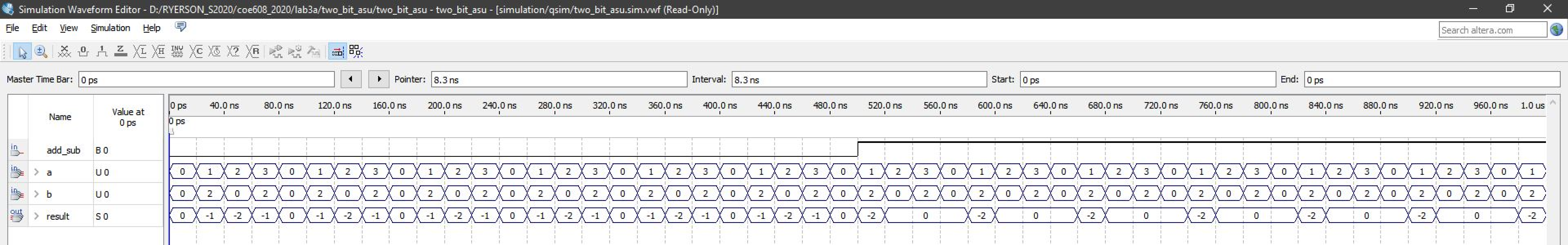
Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3a/mux2to1/mux2to1 - mux2to1 - [simulation/qsim/mux2to1.sim.vwf (Read-Only)]

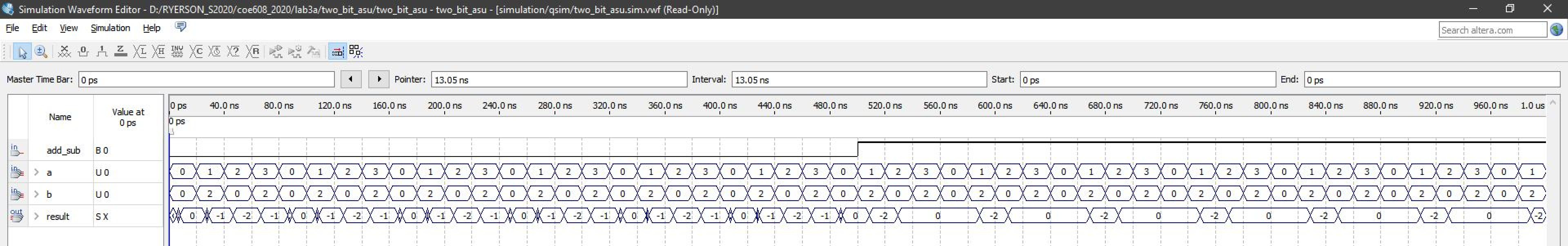
File Edit View Simulation Help

Master Time Bar: 0 ps | Pointer: 2.37 ns | Interval: 2.37 ns | Start: | End:



```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity two_bit_asu is
   port (
   a, b: in std_logic_vector(1 downto 0);
   add_sub: in std_logic;
   result: out std_logic_vector(1 downto 0));
end two_bit_asu;

architecture behavior of two_bit_asu is
   begin
      process(a,b,add_sub)
         begin
            if add_sub = '0' then --adding
               result <= a or b;
            elsif add_sub = '1' then --subtracting
               result <= (a or (not b))+ 1;
            end if;
      end process;
end behavior;
```


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3a/two_bit_asu/two_bit_asu - two_bit_asu - [simulation/qsim/two_bit_asu.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 8.3 ns
Interval: 8.3 ns
Start: 0 ps
End: 0 ps
Name
Value at 0 ps
0 ps 40.0 ns 80.0 ns 120.0 ns 160.0 ns 200.0 ns 240.0 ns 280.0 ns 320.0 ns 360.0 ns 400.0 ns 440.0 ns 480.0 ns 520.0 ns 560.0 ns 600.0 ns 640.0 ns 680.0 ns 720.0 ns 760.0 ns 800.0 ns 840.0 ns 880.0 ns 920.0 ns 960.0 ns 1.0 us
add_sub B 0
a U 0
0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1
b U 0
0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2
result S 0
0 -1 -2 -1 0 -1 -2 -1 0 -1 -2 -1 0 -1 -2 -1 0 -1 -2 -1 0 -1 -2 -1 0 -2 0 -2 0 -2 0 -2 0 -2 0 -2 0 -2


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3a/two_bit_asu/two_bit_asu - two_bit_asu - [simulation/qsim/two_bit_asu.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 13.05 ns
Interval: 13.05 ns
Start: 0 ps
End: 0 ps
Name
Value at 0 ps
add_sub B 0
a U 0
b U 0
result S X

Department of Electrical, Computer & Biomedical Engineering
Faculty of Engineering & Architectural Science

Programs: Computer Engineering

Course Number	**COE608**
Course Title	**Computer Organization and Architecture**
Semester/Year	**Summer 2020**
Instructor	**Patrick Siddavaatam**

Lab Report No.	**3b**

Lab Title	**8-bit ALU Design Implementation and Testing**

Section No.	**011**
Group No.	
Submission Date	**13 July 2020**
Due Date	**17 July 2020**

Name	Student ID	Signature*
Duanwei Zhang	*500824903*	DZ
Xinyu Hadrian Hu	500194233	XHH

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:*
www.ryerson.ca/senate/current/pol60.pdf.

Name: Xinyu Hadrian Hu, and Duan Wei Zhang

Student Number: 500194233, and 500824903

Date: June 26, 2020

COE 608: Computer Architecture and Design

COE 608: Lab 3, Part 2 Report

Purpose

The purpose of this lab is to generate the components of an 8-bit ALU, with six different operations. The addition and subtraction units must be done in structural form of VHDL. We also have to include the seven-segment display driver into the build for this ALU. This is a simplified version of the 32-bit ALU for testing before implementing the 32-bit ALU.

Design and Implementation

The following: Figure 1, Figure 2, Figure 3, and Figure 4 are the 8-bit ALU, operational inverter, BCD to seven-segment display driver, and the 8-bit register codes needed for the function of the 8-bit ALU. The truth table for the 8-bit ALU is from the waveforms generated in Quartus II. The values of the 8-bit register are placed as HEX values for the input and output of d and Q, respectively, otherwise the table will be unreasonably large and unhelpful to the reader. There is no need to show the 8-bit adder, since the operations of the 8-bit adder is the same as those of the 32-bit and 1-bit adders from the previous lab 3a.

8-bit ALU			
OP Name	**Neg/Tsel**	**ALU-Select**	**Operation Performed**
AND (logical)	0	0 0	Result <= a AND b
OR (logical)	0	0 1	Result <= a OR b
ADD	0	1 0	Result <= a + b
SUB	1	1 0	Result <= a - b
ROL	1	0 0	Result <= a << 1
ROR	1	0 1	Result <= a >> 1

Figure 1: 8-bit ALU Truth Table

Op Inverter	
OP	**not OP**
0	1
1	0

Figure 2: Op Inverter Truth Table

BCD to 7 segment display		
BCD Code	**7seg Code**	**Symbol**
0 0 0 0	0 0 0 0 0 0 1	0
0 0 0 1	1 0 0 1 1 1 1	1
0 0 1 0	0 0 1 0 0 1 0	2
0 0 1 1	0 0 0 0 1 1 0	3
0 1 0 0	1 0 0 1 1 0 0	4
0 1 0 1	0 1 0 0 1 0 0	5
0 1 1 0	0 1 0 0 0 0 0	6
0 1 1 1	0 0 0 1 1 1 1	7
1 0 0 0	0 0 0 0 0 0 0	8
1 0 0 1	0 0 0 0 1 0 0	9
1 0 1 0	0 0 0 1 0 0 0	A
1 0 1 1	1 1 0 0 0 0 0	B
1 1 0 0	0 1 1 0 0 0 1	C
1 1 0 1	1 0 0 0 0 1 0	D
1 1 1 0	0 1 1 0 0 0 0	E
1 1 1 1	0 1 1 1 0 0 0	F

Figure 3: BCD to 7-segment display converter Truth Table

8 bit register				
clk	**clr**	**ld**	**d (hex)**	**Q (hex)**
0	0	1	0 0	0 0
1	0	1	0 1	0 1
0	0	1	0 2	0 1
1	0	1	0 3	0 3
0	0	1	0 4	0 3
1	0	1	0 5	0 5
0	0	1	0 6	0 5
1	0	1	0 7	0 7
0	0	1	0 8	0 7
1	0	1	0 9	0 9
0	0	1	0A	0 9
1	0	1	0B	0 B
0	0	1	0C	0 B
1	0	1	0D	0 D
0	0	1	0E	0 D
1	0	1	0F	0 F
0	0	1	10	0 F
1	0	1	1 1	1 1
0	0	1	1 2	1 1
1	0	1	1 3	1 3
0	0	1	1 4	1 3
1	0	1	1 5	1 5
0	0	1	1 6	1 5
1	0	1	1 7	1 7
0	0	1	1 8	1 7
1	1	1	1 9	0
0	1	1	2A	0
1	1	1	2B	0
0	1	1	2C	0
1	1	1	2D	0
0	1	1	2E	0
1	1	1	2F	0
0	1	1	3 0	0
1	1	1	3 1	0

Figure 4: 8-bit register Truth Table

The final implementation of the 8-bit ALU in block diagram, schematic format is shown in the next page:

See Figure 5 below.

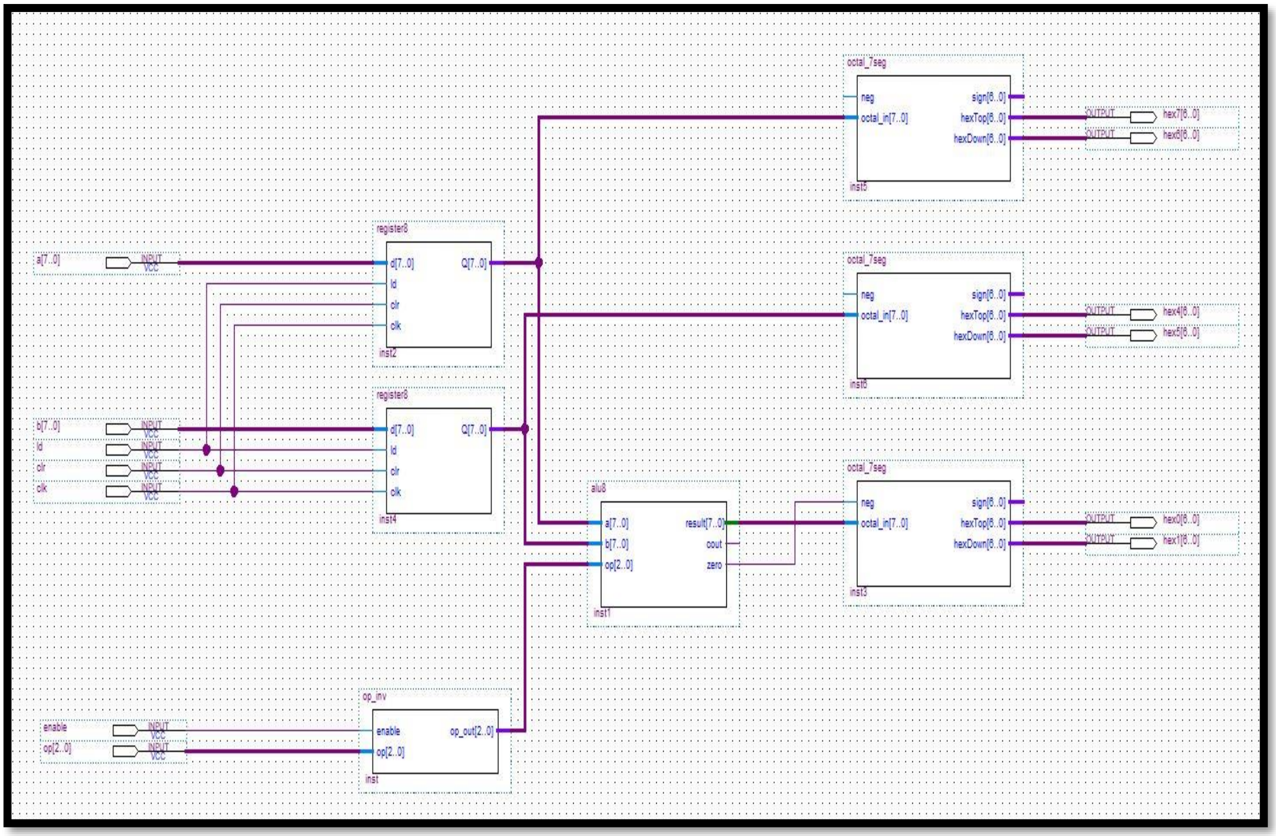


Figure 5: 8-bit ALU Schematic

Observations and Results

Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, and Figure 13 are the waveform results of both functional and timing for the 8-bit ALU.

8-bit Register: Functional and Timing Waveforms

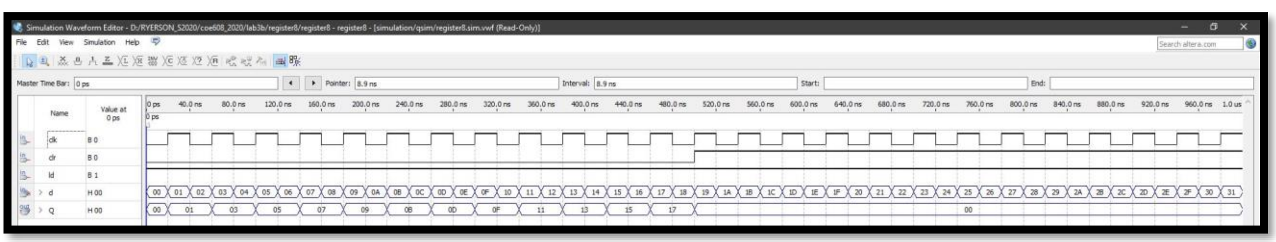


Figure 6: -bit register functional waveform

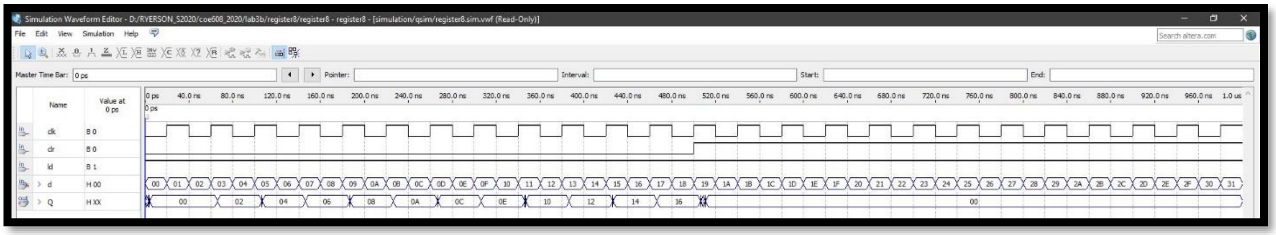


Figure 7: -bit register timing waveform

BCD to 7-segment converter: Functional and Timing Waveforms

Figure 8:: BCD to 7-segment display functional waveform

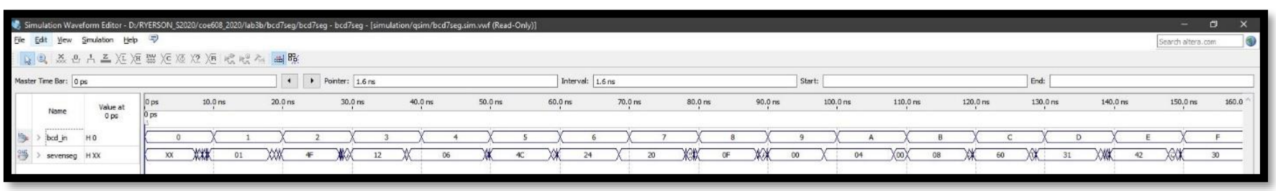


Figure 9: BCD to 7-segment display timing waveform

Operational Inverter: Functional and Timing Waveforms

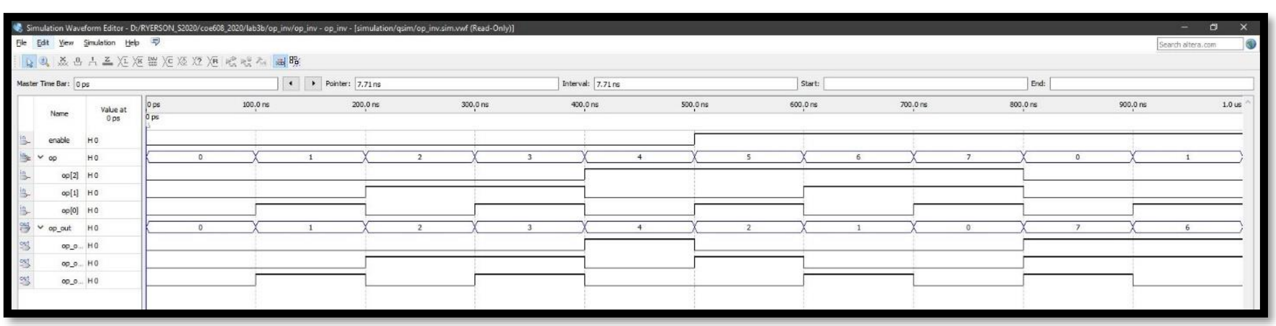


Figure 10: Op-inverter functional waveform

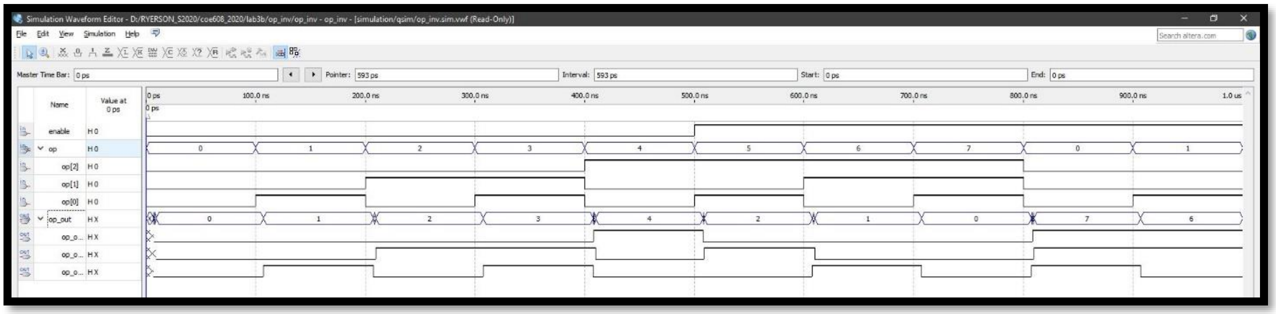


Figure 11: Op-inverter timing waveform

8-bit ALU: Functional and Timing Waveforms

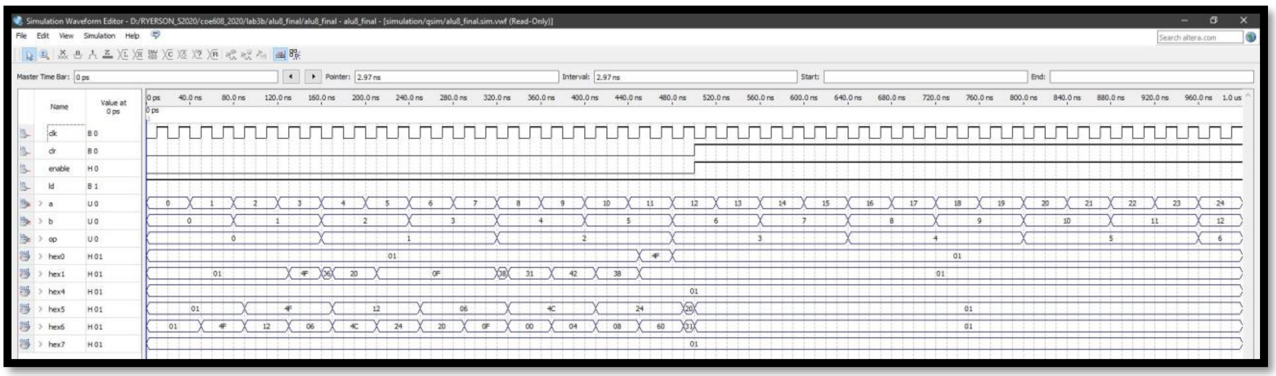


Figure 12: 8-bit ALU functional waveform

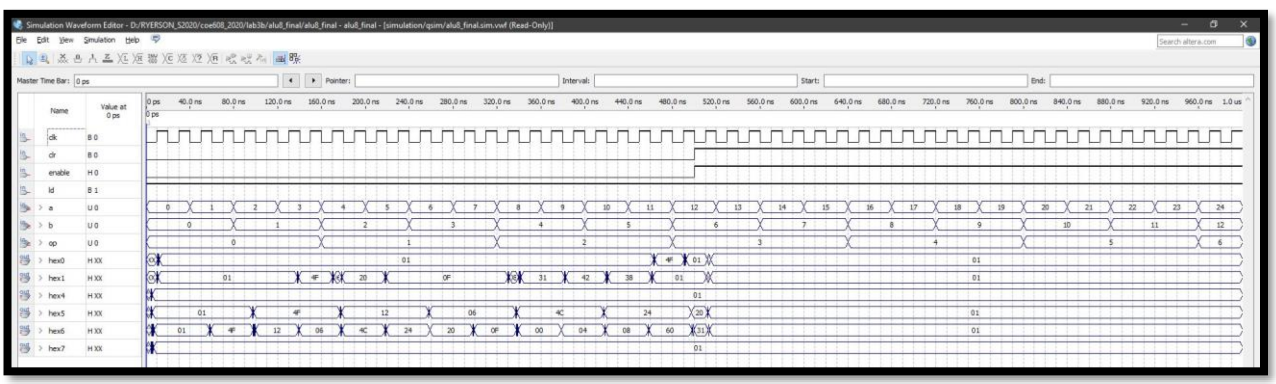
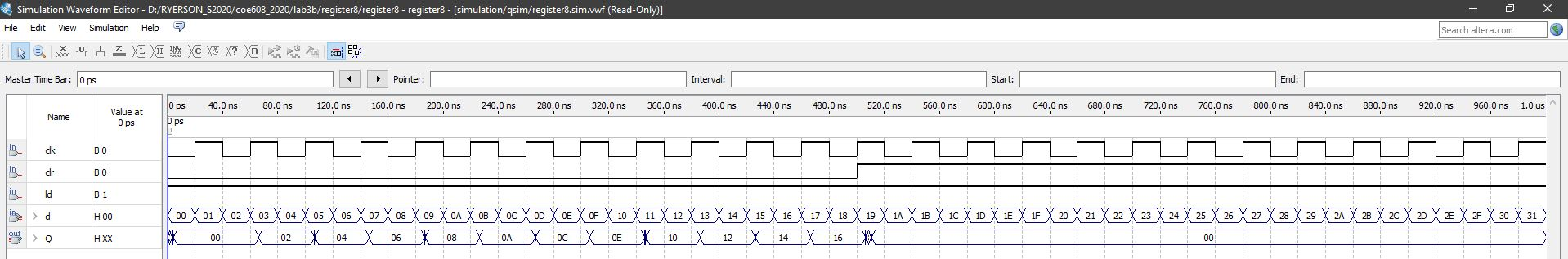


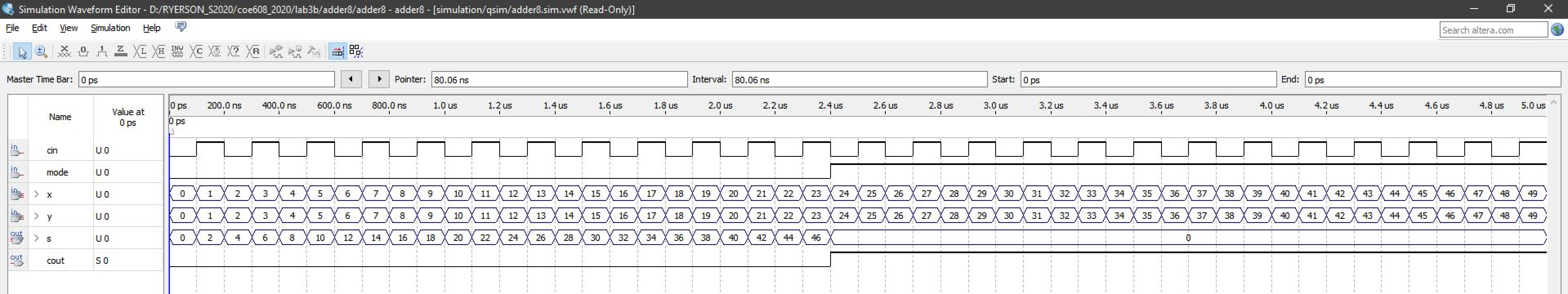
Figure 13: 8-bit ALU timing waveform

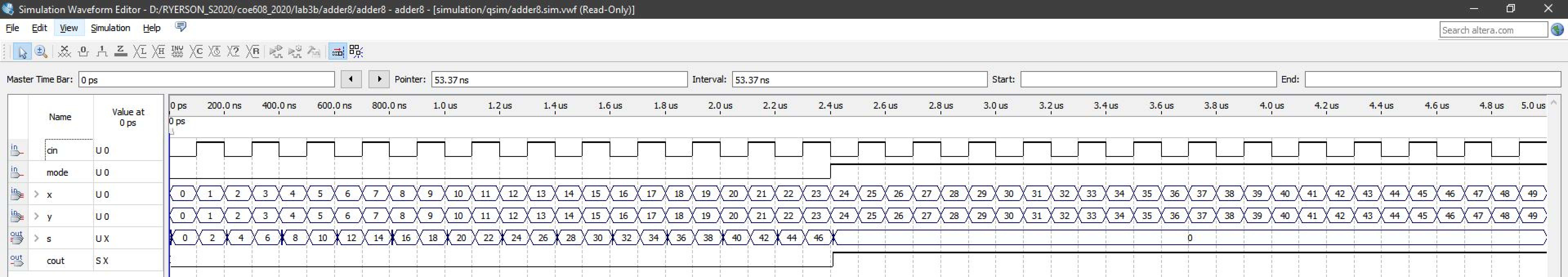
Appendices: VHDL Codes for the ALU, 7-segment displays, 8-bit register, and op inverter

The files are attached as PDF to make this document easier to read.

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;


entity adder8 is
   port(
      cin, mode: in std_logic;
      x, y : in std_logic_vector(31 downto 0);
      s: out std_logic_vector(31 downto 0);
      cout: out std_logic);
end adder8;

architecture description of adder8 is
   signal c7, c6, c5, c4, c3, c2, c1, c0: std_logic;
      component adder
         port(
            x, y, mode, cin: in std_logic;
            cout, s : out std_logic);
      end component adder;
         begin
            st0: adder port map (x(0), y(0), mode, mode, c1, s(0));
            st1: adder port map (x(1), y(1), mode, c1, c2, s(1));
            st2: adder port map (x(2), y(2), mode, c2, c3, s(2));
            st3: adder port map (x(3), y(3), mode, c3, c4, s(3));
            st4: adder port map (x(4), y(4), mode, c4, c5, s(4));
            st5: adder port map (x(5), y(5), mode, c5, c6, s(5));
            st6: adder port map (x(6), y(6), mode, c6, c7, s(6));
            st7: adder port map (x(7), y(7), mode, c7, cout, s(7));
end description;
```


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/register8/register8 - register8 - [simulation/qsim/register8.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer:
Interval:
Start:
End:
Name
Value at 0 ps
0 ps 40.0 ns 80.0 ns 120.0 ns 160.0 ns 200.0 ns 240.0 ns 280.0 ns 320.0 ns 360.0 ns 400.0 ns 440.0 ns 480.0 ns 520.0 ns 560.0 ns 600.0 ns 640.0 ns 680.0 ns 720.0 ns 760.0 ns 800.0 ns 840.0 ns 880.0 ns 920.0 ns 960.0 ns 1.0 us
0 ps
clk B 0
clr B 0
ld B 1
d H 00
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31
Q H XX
00 02 04 06 08 0A 0C 0E 10 12 14 16 00

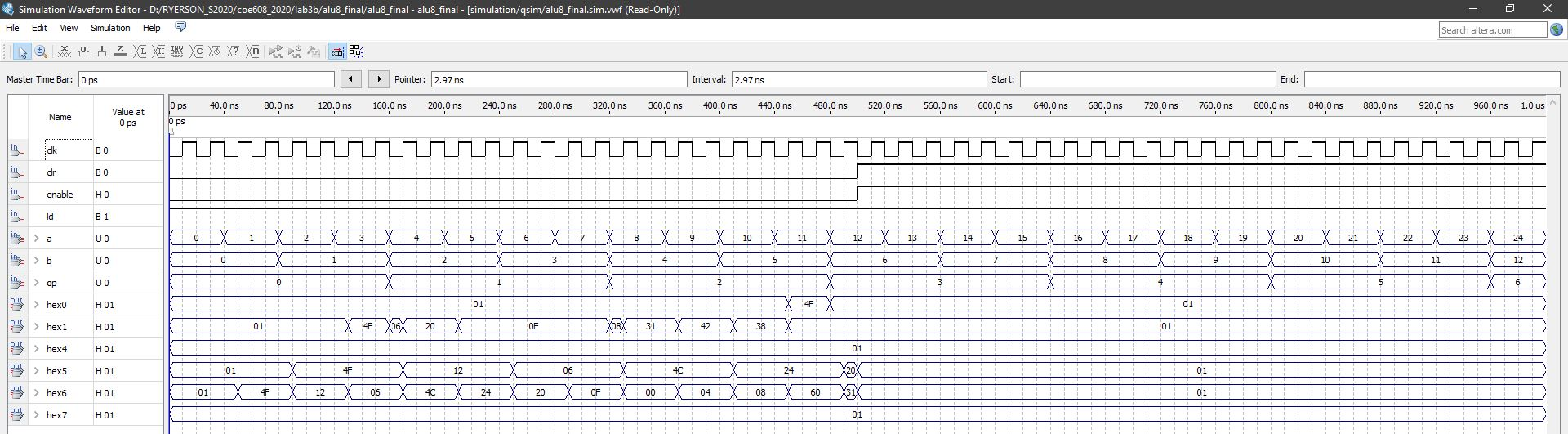

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/adder8/adder8 - adder8 - [simulation/qsim/adder8.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 80.06 ns
Interval: 80.06 ns
Start: 0 ps
End: 0 ps
Name
Value at 0 ps
0 ps 200.0 ns 400.0 ns 600.0 ns 800.0 ns 1.0 us 1.2 us 1.4 us 1.6 us 1.8 us 2.0 us 2.2 us 2.4 us 2.6 us 2.8 us 3.0 us 3.2 us 3.4 us 3.6 us 3.8 us 4.0 us 4.2 us 4.4 us 4.6 us 4.8 us 5.0 us
cin U 0
mode U 0
x U 0
0 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
y U 0
0 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
s U 0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 0
cout S 0

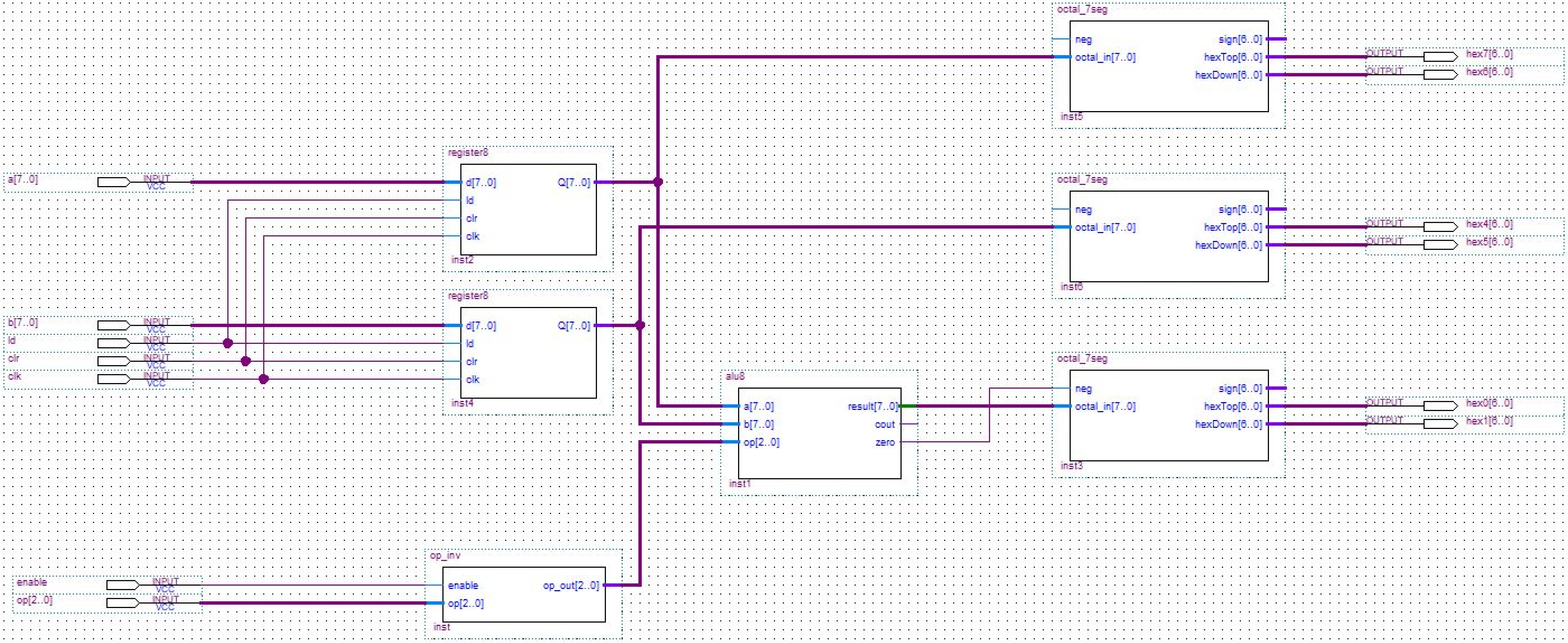

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/adder8/adder8 - adder8 - [simulation/qsim/adder8.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 53.37 ns
Interval: 53.37 ns
Start:
End:
Name
Value at 0 ps
0 ps 200.0 ns 400.0 ns 600.0 ns 800.0 ns 1.0 us 1.2 us 1.4 us 1.6 us 1.8 us 2.0 us 2.2 us 2.4 us 2.6 us 2.8 us 3.0 us 3.2 us 3.4 us 3.6 us 3.8 us 4.0 us 4.2 us 4.4 us 4.6 us 4.8 us 5.0 us
0 ps
cin U 0
mode U 0
x U 0
0 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
y U 0
0 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
s U X
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 0
cout S X

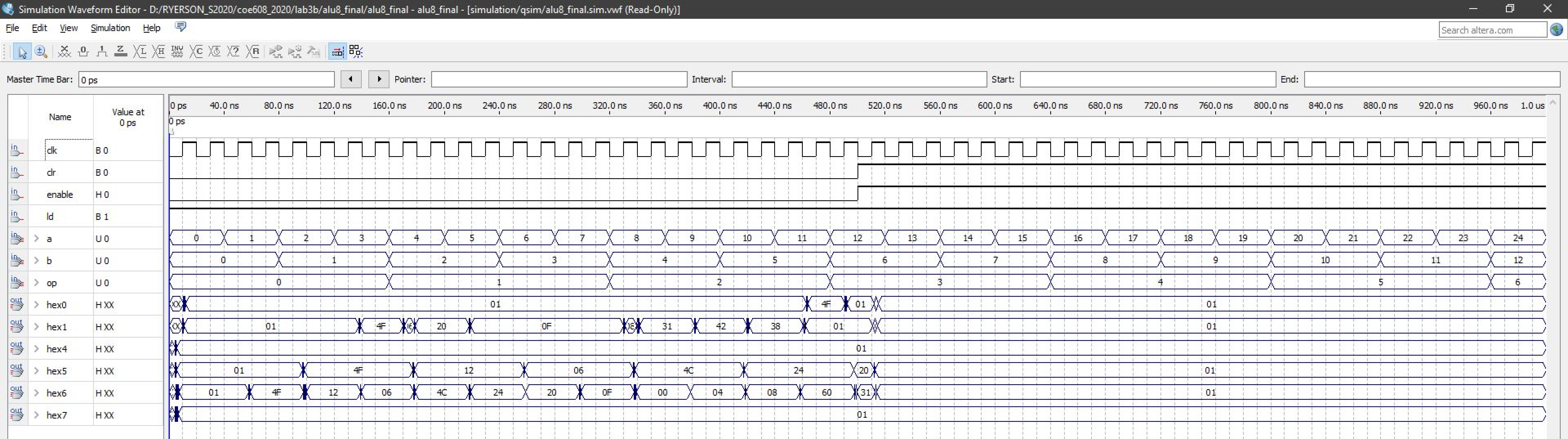
```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use ieee.numeric_std.all;

entity alu8 is
   port(
      a, b: in std_logic_vector(7 downto 0);
      op: in std_logic_vector(2 downto 0);
      result: inout std_logic_vector(7 downto 0);
      cout, zero: out std_logic);
end alu8;

architecture description of alu8 is
   component adder8 is
      port(
         cin, mode: in std_logic;
         x, y: in std_logic_vector(7 downto 0);
         s : out std_logic_vector(7 downto 0);
         cout: out std_logic);
   end component adder8;
      signal addd: std_logic_vector(7 downto 0);
      signal coutplus: std_logic;
      signal shift: std_logic_vector(7 downto 0);
         begin
            muxadd: adder8 port map (op(2), op(2), a, b, addd, coutplus);
               process (op)
                  begin
                     if op = "000" then
                        result <= a and b;
                     elsif op = "001" then
                        result <= a or b;
                     elsif op = "010" then
                        result <= addd;
                        cout <= coutplus;
                     elsif op = "110" then
                        result <= addd;
                        cout <= coutplus;
                     elsif op = "100" then
                        shift(0) <= a(7);
                        shift(7 downto 1) <= a(6 downto 0);
                        result <= shift;
                     elsif op = "101" then
                        shift(7) <= a(0);
                        shift(6 downto 0) <= a(7 downto 1);
                        result <= shift;
                     else
                        result <= (others => '0');
                     end if;

                     if result = "00000000" then
                        zero <= '1';
                     else
                        zero <= '0';
                     end if;

               end process;

end description;
```

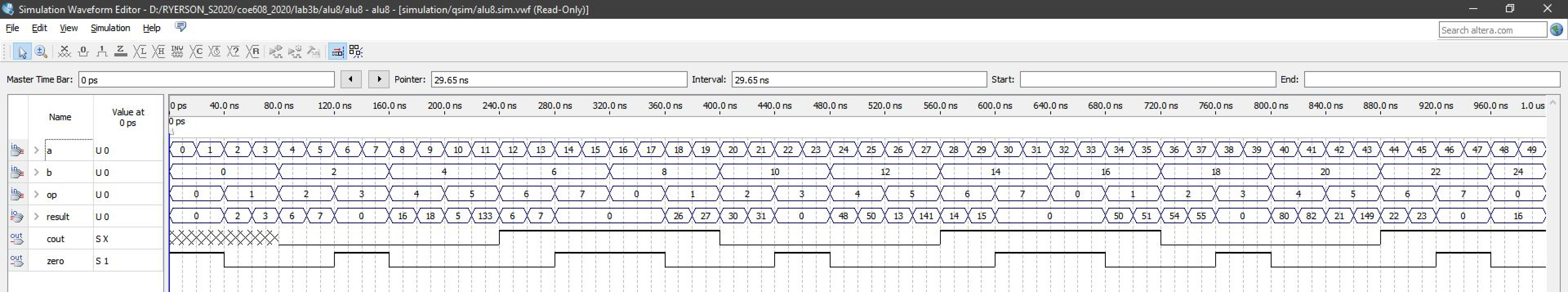
```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY work;

ENTITY alu8_final IS
   PORT
   (
      enable :  IN  STD_LOGIC;
      ld :  IN  STD_LOGIC;
      clr :  IN  STD_LOGIC;
      clk :  IN  STD_LOGIC;
      a :  IN  STD_LOGIC_VECTOR(7 DOWNTO 0);
      b :  IN  STD_LOGIC_VECTOR(7 DOWNTO 0);
      op :  IN  STD_LOGIC_VECTOR(2 DOWNTO 0);
      hex0 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
      hex1 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
      hex4 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
      hex5 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
      hex6 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
      hex7 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0)
   );
END alu8_final;

ARCHITECTURE bdf_type OF alu8_final IS

COMPONENT op_inv
   PORT(enable : IN STD_LOGIC;
        op : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
        op_out : OUT STD_LOGIC_VECTOR(2 DOWNTO 0)
   );
END COMPONENT;

COMPONENT alu8
   PORT(a : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
        b : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
        op : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
        result : INOUT STD_LOGIC_VECTOR(7 DOWNTO 0);
        cout : OUT STD_LOGIC;
        zero : OUT STD_LOGIC
   );
END COMPONENT;

COMPONENT register8
   PORT(ld : IN STD_LOGIC;
        clr : IN STD_LOGIC;
        clk : IN STD_LOGIC;
        d : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
        Q : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
   );
END COMPONENT;

COMPONENT octal_7seg
   PORT(neg : IN STD_LOGIC;
        octal_in : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
        hexDown : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);
        hexTop : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);
        sign : OUT STD_LOGIC_VECTOR(6 DOWNTO 0)
   );
END COMPONENT;

SIGNAL   SYNTHESIZED_WIRE_7 :  STD_LOGIC_VECTOR(7 DOWNTO 0);
```

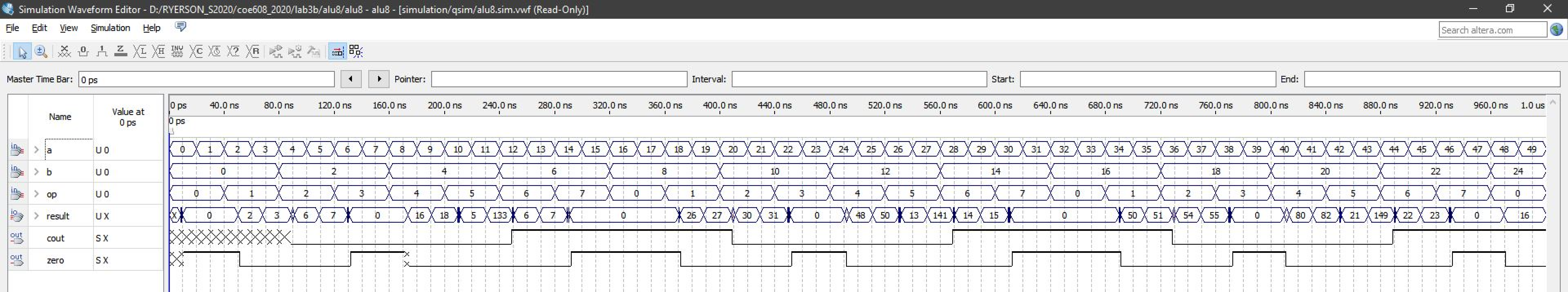
```
SIGNAL	SYNTHESIZED_WIRE_8 :  STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL	op_inv1 :  STD_LOGIC_VECTOR(2 DOWNTO 0);
SIGNAL	zero1 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_4 :  STD_LOGIC_VECTOR(7 DOWNTO 0);


BEGIN



b2v_inst : op_inv
PORT MAP(enable => enable,
		 op => op,
		 op_out => op_inv1);


b2v_inst1 : alu8
PORT MAP(a => A_2_ALU,
		 b => B_2_ALU,
		 op => op_inv1,
		 result => result1,
		 zero => zero1);


b2v_inst2 : register8
PORT MAP(ld => ld,
		 clr => clr,
		 clk => clk,
		 d => a,
		 Q => A_2_ALU);


b2v_inst3 : octal_7seg
PORT MAP(neg => zero1,
		 octal_in => result1,
		 hexDown => hex1,
		 hexTop => hex0);


b2v_inst4 : register8
PORT MAP(ld => ld,
		 clr => clr,
		 clk => clk,
		 d => b,
		 Q => B_2_ALU);


b2v_inst5 : octal_7seg
PORT MAP(octal_in => A_2_ALU,
		 hexDown => hex6,
		 hexTop => hex7);


b2v_inst6 : octal_7seg
PORT MAP(octal_in => B_2_ALU,
		 hexDown => hex5,
		 hexTop => hex4);


END bdf_type;
```


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/alu8_final/alu8_final - alu8_final - [simulation/qsim/alu8_final.sim.vwf (Read-Only)]
File Edit View Simulation Help
Master Time Bar: 0 ps
Pointer: 2.97 ns
Interval: 2.97 ns
Start:
End:
Name
Value at 0 ps
clk B 0
clr B 0
enable H 0
ld B 1
a U 0
b U 0
op U 0
hex0 H 01
hex1 H 01
hex4 H 01
hex5 H 01
hex6 H 01
hex7 H 01
0 ps 40.0 ns 80.0 ns 120.0 ns 160.0 ns 200.0 ns 240.0 ns 280.0 ns 320.0 ns 360.0 ns 400.0 ns 440.0 ns 480.0 ns 520.0 ns 560.0 ns 600.0 ns 640.0 ns 680.0 ns 720.0 ns 760.0 ns 800.0 ns 840.0 ns 880.0 ns 920.0 ns 960.0 ns 1.0 us
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
0 1 2 3 4 5 6 7 8 9 10 11 12
0 1 2 3 4 5 6
01 4F 01
01 4F 06 20 0F 08 31 42 38 01
01
01 4F 12 06 4C 24 20 01
01 4F 12 06 4C 24 20 0F 00 04 08 60 31 01
01

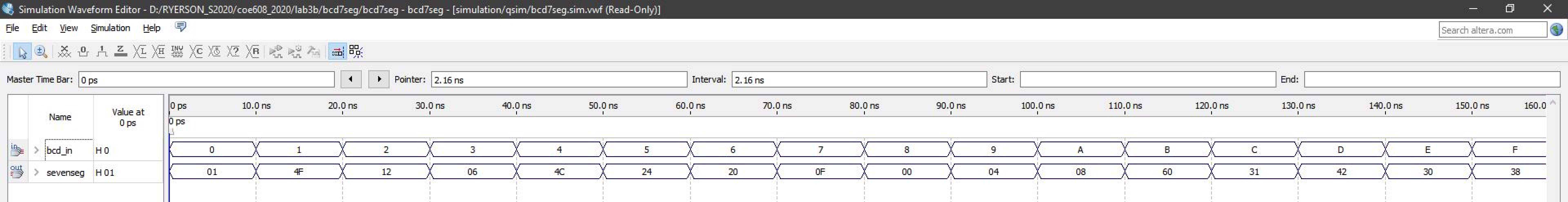

octal_7seg
neg
octal_in[7..0]
sign[6..0]
hexTop[6..0]
hexDown[6..0]
inst5
OUTPUT
hex7[6..0]
OUTPUT
hex6[6..0]
register8
a[7..0]
INPUT
VCC
d[7..0]
Q[7..0]
ld
clr
clk
inst2
octal_7seg
neg
octal_in[7..0]
sign[6..0]
hexTop[6..0]
hexDown[6..0]
inst6
OUTPUT
hex4[6..0]
OUTPUT
hex5[6..0]
register8
b[7..0]
ld
clr
clk
d[7..0]
Q[7..0]
ld
clr
clk
inst4
alu8
a[7..0]
b[7..0]
op[2..0]
result[7..0]
cout
zero
inst1
octal_7seg
neg
octal_in[7..0]
sign[6..0]
hexTop[6..0]
hexDown[6..0]
inst3
OUTPUT
hex0[6..0]
OUTPUT
hex1[6..0]
op_inv
enable
op[2..0]
INPUT
VCC
enable
op[2..0]
op_out[2..0]
inst

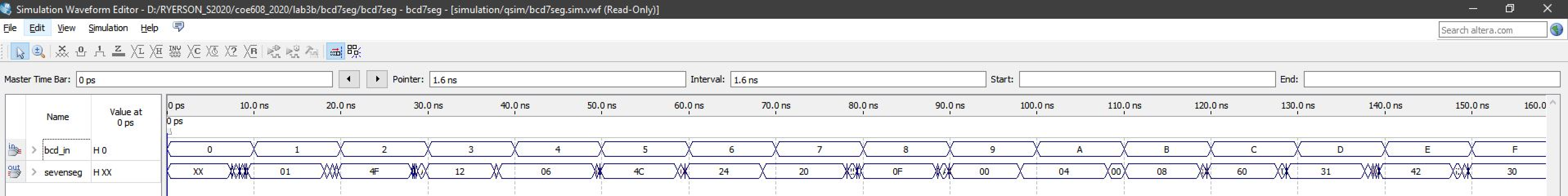

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/alu8_final/alu8_final - alu8_final - [simulation/qsim/alu8_final.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer:
Interval:
Start:
End:
Name
Value at 0 ps
0 ps 40.0 ns 80.0 ns 120.0 ns 160.0 ns 200.0 ns 240.0 ns 280.0 ns 320.0 ns 360.0 ns 400.0 ns 440.0 ns 480.0 ns 520.0 ns 560.0 ns 600.0 ns 640.0 ns 680.0 ns 720.0 ns 760.0 ns 800.0 ns 840.0 ns 880.0 ns 920.0 ns 960.0 ns 1.0 us
clk B 0
clr B 0
enable H 0
ld B 1
a U 0 : 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
b U 0 : 0 1 2 3 4 5 6 7 8 9 10 11 12
op U 0 : 0 1 2 3 4 5 6
hex0 H XX : XX 01 4F 01 01
hex1 H XX : XX 01 4F 20 0F 31 42 38 01 01
hex4 H XX : 01
hex5 H XX : 01 4F 12 06 4C 24 20 01
hex6 H XX : 01 4F 12 06 4C 24 20 0F 00 04 08 60 31 01
hex7 H XX : 01


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/alu8/alu8 - alu8 - [simulation/qsim/alu8.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 29.65 ns
Interval: 29.65 ns
Start:
End:
Name
Value at 0 ps
0 ps 40.0 ns 80.0 ns 120.0 ns 160.0 ns 200.0 ns 240.0 ns 280.0 ns 320.0 ns 360.0 ns 400.0 ns 440.0 ns 480.0 ns 520.0 ns 560.0 ns 600.0 ns 640.0 ns 680.0 ns 720.0 ns 760.0 ns 800.0 ns 840.0 ns 880.0 ns 920.0 ns 960.0 ns 1.0 us
a U 0 0 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
b U 0 0 2 4 6 8 10 12 14 16 18 20 22 24
op U 0 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0
result U 0 0 2 3 6 7 0 16 18 5 133 6 7 0 26 27 30 31 0 48 50 13 141 14 15 0 50 51 54 55 0 80 82 21 149 22 23 0 16
cout S X
zero S 1


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/alu8/alu8 - alu8 - [simulation/qsim/alu8.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer:
Interval:
Start:
End:
Name
Value at 0 ps
0 ps 40.0 ns 80.0 ns 120.0 ns 160.0 ns 200.0 ns 240.0 ns 280.0 ns 320.0 ns 360.0 ns 400.0 ns 440.0 ns 480.0 ns 520.0 ns 560.0 ns 600.0 ns 640.0 ns 680.0 ns 720.0 ns 760.0 ns 800.0 ns 840.0 ns 880.0 ns 920.0 ns 960.0 ns 1.0 us
a U 0
0 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
b U 0
0 2 4 6 8 10 12 14 16 18 20 22 24
op U 0
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0
result U X
X 0 2 3 6 7 0 16 18 5 133 6 7 0 26 27 30 31 0 48 50 13 141 14 15 0 50 51 54 55 0 80 82 21 149 22 23 0 16
cout S X
zero S X

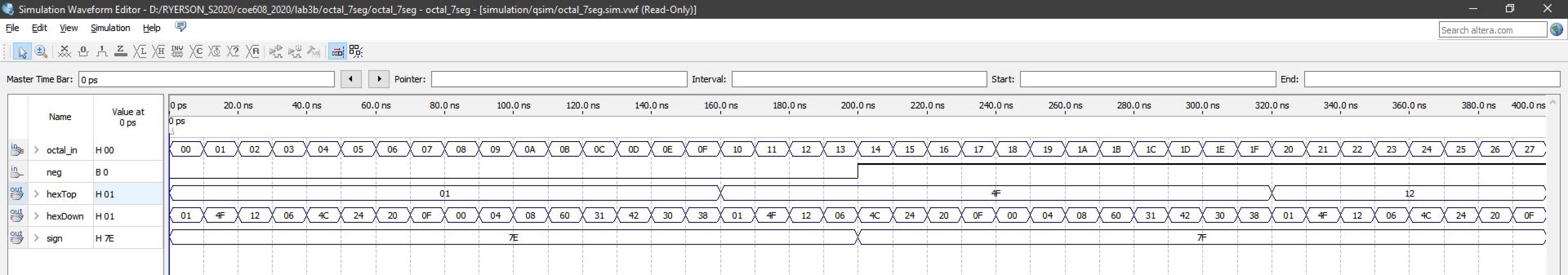
```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use ieee.numeric_std.all;
use ieee.numeric_bit.all;

entity bcd7seg is
   port(
      bcd_in: in std_logic_vector(3 downto 0);
      sevenseg: out std_logic_vector(6 downto 0));
end bcd7seg;

architecture behavior of bcd7seg is
   begin
      process(bcd_in)
         begin
            case bcd_in is
               when "0000" => sevenseg <= "0000001"; --0
               when "0001" => sevenseg <= "1001111"; --1
               when "0010" => sevenseg <= "0010010"; --2
               when "0011" => sevenseg <= "0000110"; --3
               when "0100" => sevenseg <= "1001100"; --4
               when "0101" => sevenseg <= "0100100"; --5
               when "0110" => sevenseg <= "0100000"; --6
               when "0111" => sevenseg <= "0001111"; --7
               when "1000" => sevenseg <= "0000000"; --8
               when "1001" => sevenseg <= "0000100"; --9
               when "1010" => sevenseg <= "0001000"; --A
               when "1011" => sevenseg <= "1100000"; --b
               when "1100" => sevenseg <= "0110001"; --C
               when "1101" => sevenseg <= "1000010"; --d
               when "1110" => sevenseg <= "0110000"; --E
               when "1111" => sevenseg <= "0111000"; --F
            end case;
      end process;
end behavior;
```

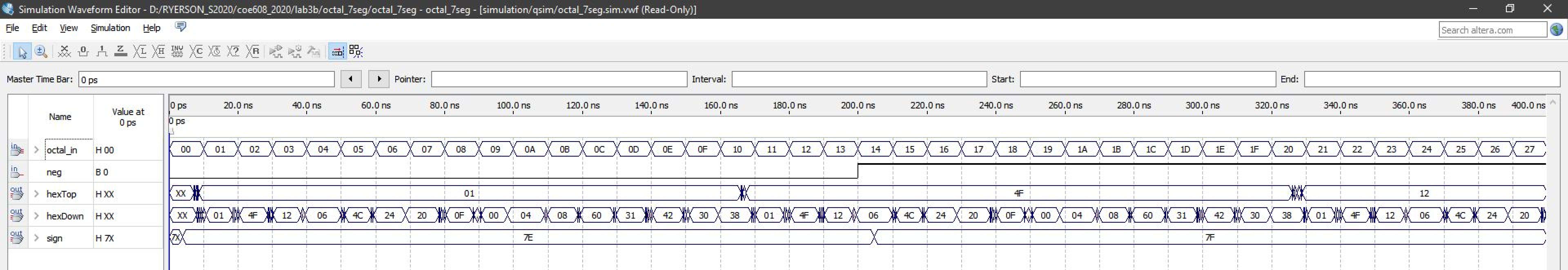

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/bcd7seg/bcd7seg - bcd7seg - [simulation/qsim/bcd7seg.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 2.16 ns
Interval: 2.16 ns
Start:
End:
Name
Value at 0 ps
0 ps
10.0 ns
20.0 ns
30.0 ns
40.0 ns
50.0 ns
60.0 ns
70.0 ns
80.0 ns
90.0 ns
100.0 ns
110.0 ns
120.0 ns
130.0 ns
140.0 ns
150.0 ns
160.0
bcd_in
H 0
0 1 2 3 4 5 6 7 8 9 A B C D E F
sevenseg
H 01
01 4F 12 06 4C 24 20 0F 00 04 08 60 31 42 30 38


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/bcd7seg/bcd7seg - bcd7seg - [simulation/qsim/bcd7seg.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 1.6 ns
Interval: 1.6 ns
Start:
End:
Name
Value at 0 ps
0 ps
10.0 ns
20.0 ns
30.0 ns
40.0 ns
50.0 ns
60.0 ns
70.0 ns
80.0 ns
90.0 ns
100.0 ns
110.0 ns
120.0 ns
130.0 ns
140.0 ns
150.0 ns
160.0
bcd_in
H 0
0 1 2 3 4 5 6 7 8 9 A B C D E F
sevenseg
H XX
XX 01 4F 12 06 4C 24 20 0F 00 04 00 08 60 31 42 30

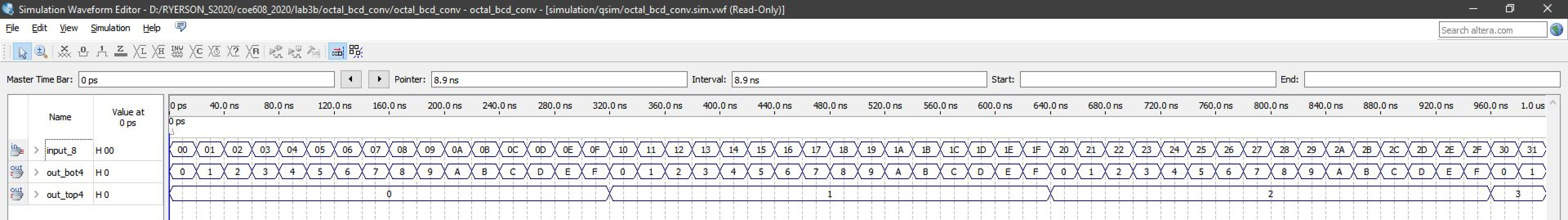
```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use ieee.numeric_bit.all;
use ieee.numeric_std.all;
use ieee.std_logic_misc.all;

entity octal_7seg is
   port(
      neg : in std_logic;
      octal_in : in std_logic_vector(7 downto 0);
      sign, hexTop, hexDown : out std_logic_vector(6 downto 0));
end octal_7seg;

architecture behavior of octal_7seg is
   signal bcdTop, bcdDown : std_logic_vector(3 downto 0);
      begin
         process(octal_in)
            begin
               bcdTop <= octal_in(7 downto 4);
               bcdDown <= octal_in(3 downto 0);
                  case bcdTop is
                     when "0000" => hexTop <= "0000001"; --0
                     when "0001" => hexTop <= "1001111"; --1
                     when "0010" => hexTop <= "0010010"; --2
                     when "0011" => hexTop <= "0000110"; --3
                     when "0100" => hexTop <= "1001100"; --4
                     when "0101" => hexTop <= "0100100"; --5
                     when "0110" => hexTop <= "0100000"; --6
                     when "0111" => hexTop <= "0001111"; --7
                     when "1000" => hexTop <= "0000000"; --8
                     when "1001" => hexTop <= "0000100"; --9
                     when "1010" => hexTop <= "0001000"; --A
                     when "1011" => hexTop <= "1100000"; --b
                     when "1100" => hexTop <= "0110001"; --C
                     when "1101" => hexTop <= "1000010"; --d
                     when "1110" => hexTop <= "0110000"; --E
                     when "1111" => hexTop <= "0111000"; --F
                     when others => hexTop <= (others => '0');
                  end case;
                  case bcdDown is
                     when "0000" => hexDown <= "0000001"; --0
                     when "0001" => hexDown <= "1001111"; --1
                     when "0010" => hexDown <= "0010010"; --2
                     when "0011" => hexDown <= "0000110"; --3
                     when "0100" => hexDown <= "1001100"; --4
                     when "0101" => hexDown <= "0100100"; --5
                     when "0110" => hexDown <= "0100000"; --6
                     when "0111" => hexDown <= "0001111"; --7
                     when "1000" => hexDown <= "0000000"; --8
                     when "1001" => hexDown <= "0000100"; --9
                     when "1010" => hexDown <= "0001000"; --A
                     when "1011" => hexDown <= "1100000"; --b
                     when "1100" => hexDown <= "0110001"; --C
                     when "1101" => hexDown <= "1000010"; --d
                     when "1110" => hexDown <= "0110000"; --E
                     when "1111" => hexDown <= "0111000"; --F
                     when others => hexDown <= (others => '0');
                  end case;
         if neg = '0' then
            sign <= "1111110";
```

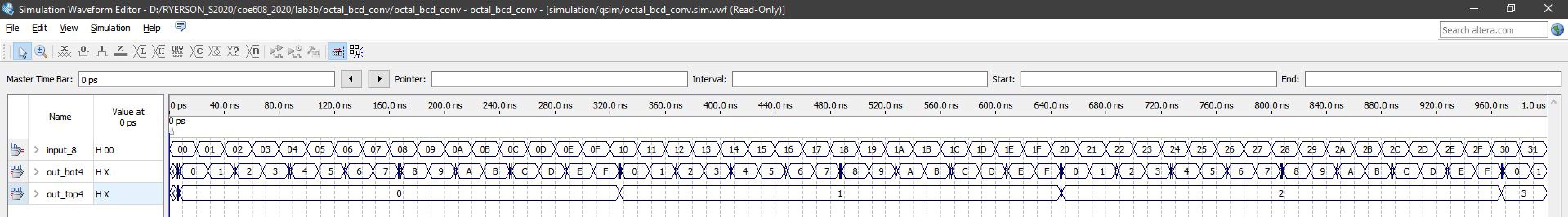
```
          elsif neg = '1' then
             sign <= "1111111";
          end if;
       end process;
end behavior;

```


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/octal_7seg/octal_7seg - octal_7seg - [simulation/qsim/octal_7seg.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer:
Interval:
Start:
End:
Name
Value at 0 ps
0 ps 20.0 ns 40.0 ns 60.0 ns 80.0 ns 100.0 ns 120.0 ns 140.0 ns 160.0 ns 180.0 ns 200.0 ns 220.0 ns 240.0 ns 260.0 ns 280.0 ns 300.0 ns 320.0 ns 340.0 ns 360.0 ns 380.0 ns 400.0 ns
octal_in H 00
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27
neg B 0
hexTop H 01
01 4F 12
hexDown H 01
01 4F 12 06 4C 24 20 0F 00 04 08 60 31 42 30 38 01 4F 12 06 4C 24 20 0F 00 04 08 60 31 42 30 38 01 4F 12 06 4C 24 20 0F
sign H 7E
7E 7F

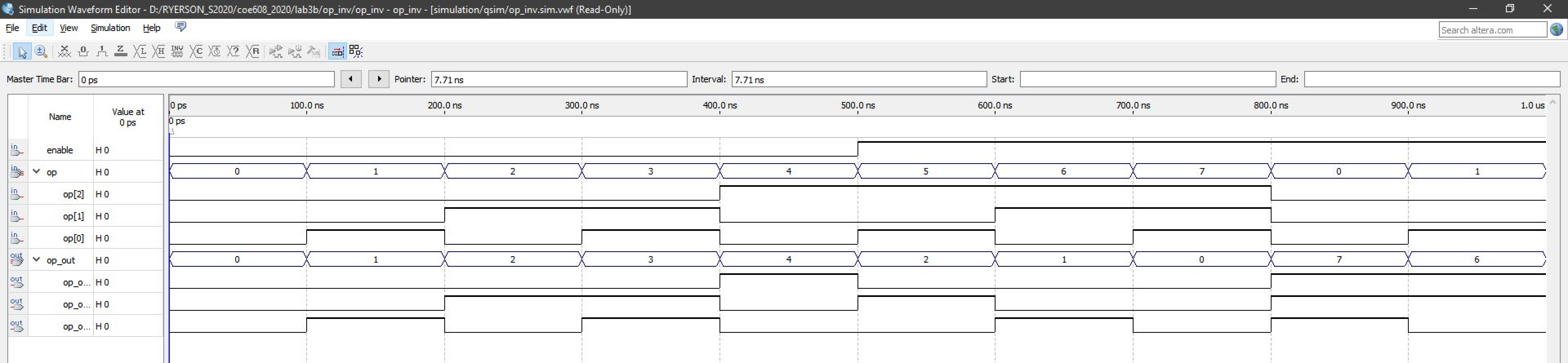

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/octal_7seg/octal_7seg - octal_7seg - [simulation/qsim/octal_7seg.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer:
Interval:
Start:
End:
Name
Value at 0 ps
0 ps 20.0 ns 40.0 ns 60.0 ns 80.0 ns 100.0 ns 120.0 ns 140.0 ns 160.0 ns 180.0 ns 200.0 ns 220.0 ns 240.0 ns 260.0 ns 280.0 ns 300.0 ns 320.0 ns 340.0 ns 360.0 ns 380.0 ns 400.0 ns
0 ps
octal_in H 00
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27
neg B 0
hexTop H XX
XX 01 4F 12
hexDown H XX
XX 01 4F 12 06 4C 24 20 0F 00 04 08 60 31 42 30 38 01 4F 12 06 4C 24 20 0F 00 04 08 60 31 42 30 38 01 4F 12 06 4C 24 20
sign H 7X
7X 7E 7F

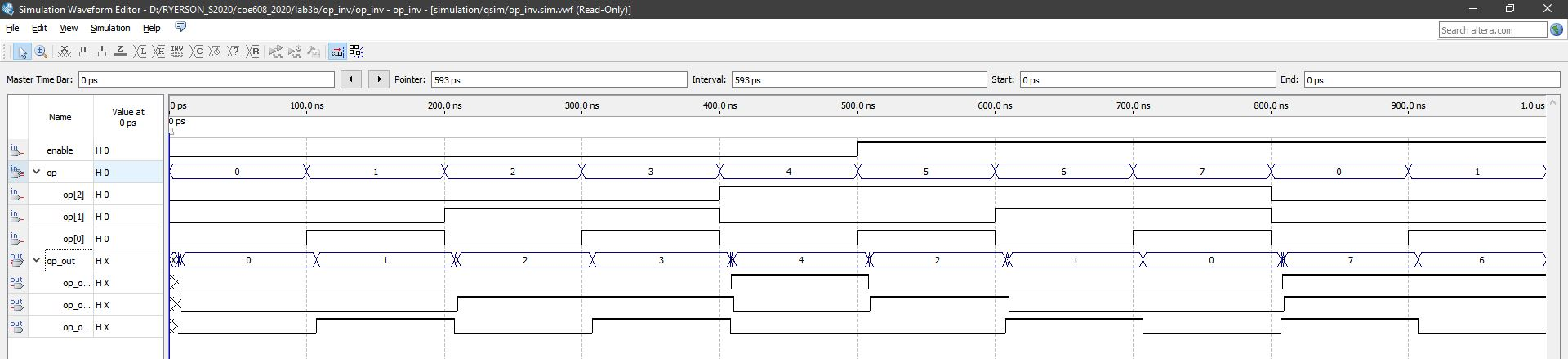
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity octal_bcd_conv is
   port(
      input_8  : in std_logic_vector(7 downto 0);
      out_top4: out std_logic_vector(3 downto 0);
      out_bot4 : out std_logic_vector(3 downto 0));
end octal_bcd_conv;

architecture behavior of octal_bcd_conv is
   begin
      process(input_8)
         begin
            out_top4 <= input_8(7 downto 4);
            out_bot4 <= input_8(3 downto 0);
      end process;
end behavior;
```


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/octal_bcd_conv/octal_bcd_conv - octal_bcd_conv - [simulation/qsim/octal_bcd_conv.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 8.9 ns
Interval: 8.9 ns
Start:
End:
Name
Value at 0 ps
0 ps 40.0 ns 80.0 ns 120.0 ns 160.0 ns 200.0 ns 240.0 ns 280.0 ns 320.0 ns 360.0 ns 400.0 ns 440.0 ns 480.0 ns 520.0 ns 560.0 ns 600.0 ns 640.0 ns 680.0 ns 720.0 ns 760.0 ns 800.0 ns 840.0 ns 880.0 ns 920.0 ns 960.0 ns 1.0 us
0 ps
input_8 H 00
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31
out_bot4 H 0
0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1
out_top4 H 0
0 1 2 3


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/octal_bcd_conv/octal_bcd_conv - octal_bcd_conv - [simulation/qsim/octal_bcd_conv.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer:
Interval:
Start:
End:
Name
Value at 0 ps
0 ps
40.0 ns 80.0 ns 120.0 ns 160.0 ns 200.0 ns 240.0 ns 280.0 ns 320.0 ns 360.0 ns 400.0 ns 440.0 ns 480.0 ns 520.0 ns 560.0 ns 600.0 ns 640.0 ns 680.0 ns 720.0 ns 760.0 ns 800.0 ns 840.0 ns 880.0 ns 920.0 ns 960.0 ns 1.0 us
input_8 H 00
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31
out_bot4 H X
0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1
out_top4 H X
0 1 2 3

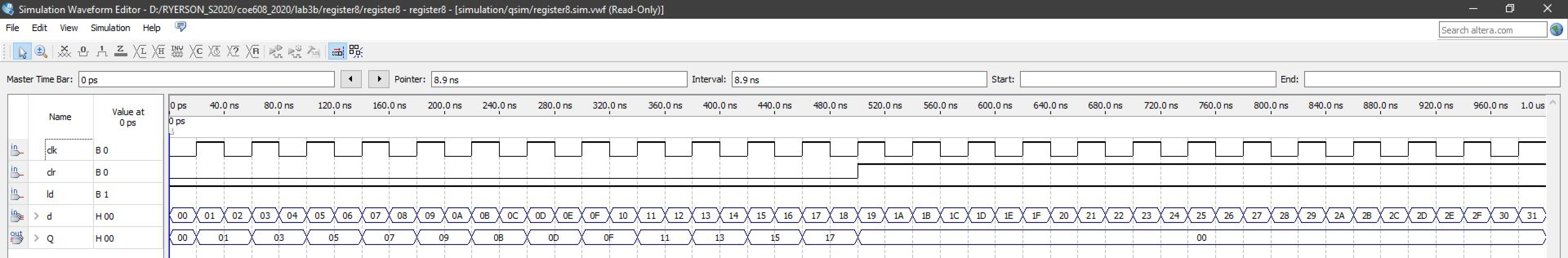
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity op_inv is
   port(
      enable: in std_logic;
      op: in std_logic_vector(2 downto 0);
      op_out : out std_logic_vector (2 downto 0));
end op_inv;

architecture behavior of op_inv is
   begin
      process(enable, op)
         begin
            if enable = '0' then
               op_out <= op;
            elsif enable = '1' then
               op_out <= not op;
            else
               op_out <= "---";
            end if;
      end process;
end behavior;
```

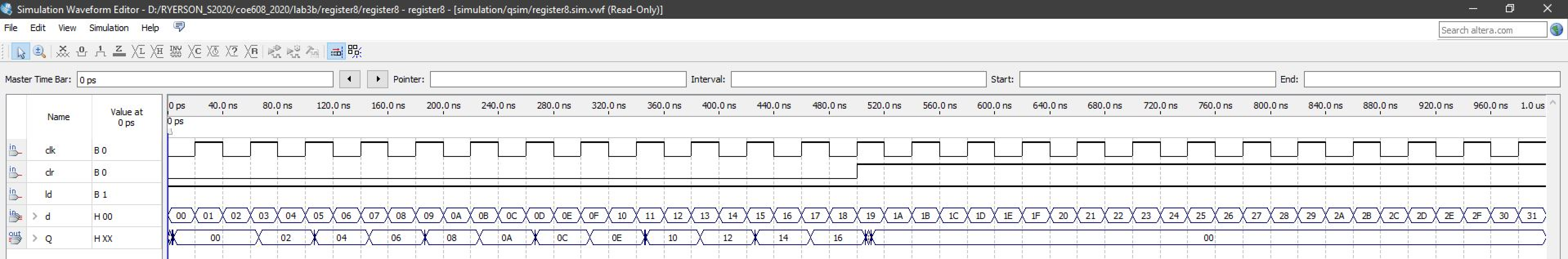

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/op_inv/op_inv - op_inv - [simulation/qsim/op_inv.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 7.71 ns
Interval: 7.71 ns
Start:
End:
Name
Value at 0 ps
0 ps
100.0 ns
200.0 ns
300.0 ns
400.0 ns
500.0 ns
600.0 ns
700.0 ns
800.0 ns
900.0 ns
1.0 us
enable H 0
op H 0
0 1 2 3 4 5 6 7 0 1
op[2] H 0
op[1] H 0
op[0] H 0
op_out H 0
0 1 2 3 4 2 1 0 7 6
op_o... H 0
op_o... H 0
op_o... H 0



Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/op_inv/op_inv - op_inv - [simulation/qsim/op_inv.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 593 ps
Interval: 593 ps
Start: 0 ps
End: 0 ps
Name
Value at 0 ps
0 ps
100.0 ns
200.0 ns
300.0 ns
400.0 ns
500.0 ns
600.0 ns
700.0 ns
800.0 ns
900.0 ns
1.0 us
enable H 0
op H 0
0 1 2 3 4 5 6 7 0 1
op[2] H 0
op[1] H 0
op[0] H 0
op_out H X
0 1 2 3 4 2 1 0 7 6
op_o... H X
op_o... H X
op_o... H X

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity register8 is
   port (
   d: in std_logic_vector(7 downto 0);
   ld,clr,clk: in std_logic;
   Q: out std_logic_vector(7 downto 0));
end register8;

architecture description of register8 is
   begin
      process (ld,clr,clk)
         begin
            if ld = '0' or clr = '1' then
               Q <= (others => '0');
            elsif rising_edge(clk) then
               Q <= d;
            end if;
      end process;
end description;
```


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/register8/register8 - register8 - [simulation/qsim/register8.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 8.9 ns
Interval: 8.9 ns
Start:
End:
Name
Value at 0 ps
0 ps 40.0 ns 80.0 ns 120.0 ns 160.0 ns 200.0 ns 240.0 ns 280.0 ns 320.0 ns 360.0 ns 400.0 ns 440.0 ns 480.0 ns 520.0 ns 560.0 ns 600.0 ns 640.0 ns 680.0 ns 720.0 ns 760.0 ns 800.0 ns 840.0 ns 880.0 ns 920.0 ns 960.0 ns 1.0 us
clk B 0
clr B 0
ld B 1
d H 00
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31
Q H 00
00 01 03 05 07 09 0B 0D 0F 11 13 15 17 00


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/register8/register8 - register8 - [simulation/qsim/register8.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer:
Interval:
Start:
End:
Name
Value at 0 ps
0 ps 40.0 ns 80.0 ns 120.0 ns 160.0 ns 200.0 ns 240.0 ns 280.0 ns 320.0 ns 360.0 ns 400.0 ns 440.0 ns 480.0 ns 520.0 ns 560.0 ns 600.0 ns 640.0 ns 680.0 ns 720.0 ns 760.0 ns 800.0 ns 840.0 ns 880.0 ns 920.0 ns 960.0 ns 1.0 us
0 ps
clk B 0
clr B 0
ld B 1
d H 00
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31
Q H XX
00 02 04 06 08 0A 0C 0E 10 12 14 16 00