Name: Xinyu Hadrian Hu, and Duan Wei Zhang

Student Number: 500194233, and 500824903

Date: June 26, 2020

COE 608: Computer Architecture and Design

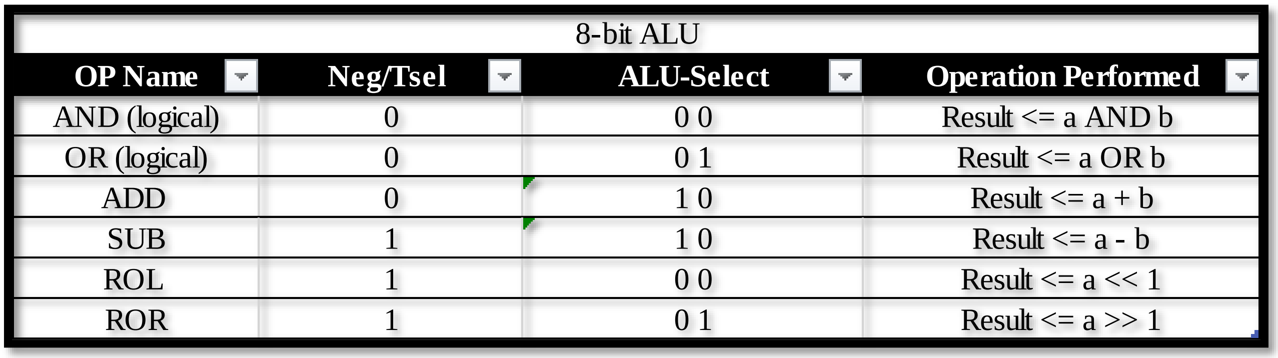
COE 608: Lab 3, Part 2 Report

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

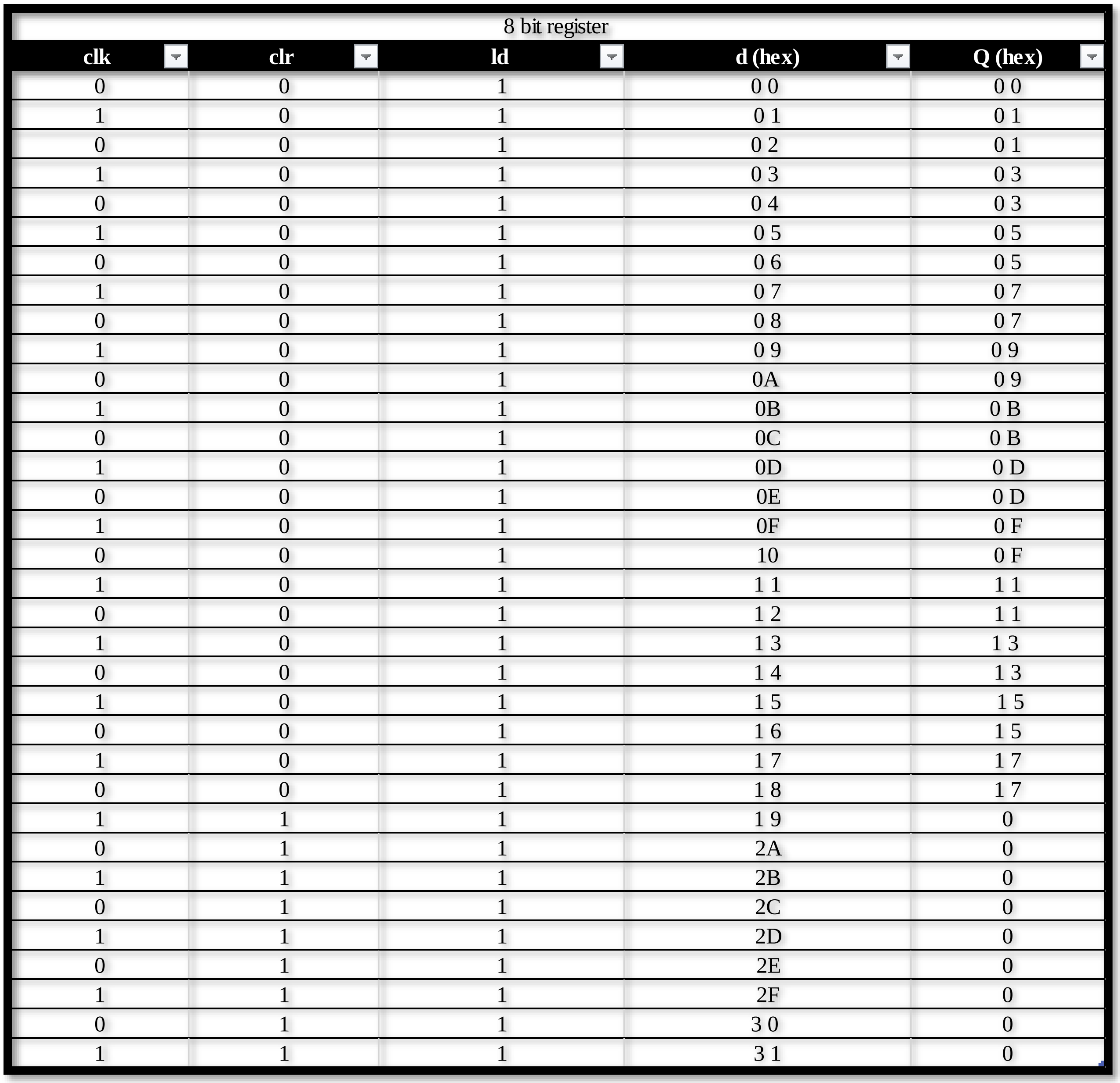
Figure 1: 8-bit ALU Truth Table

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.

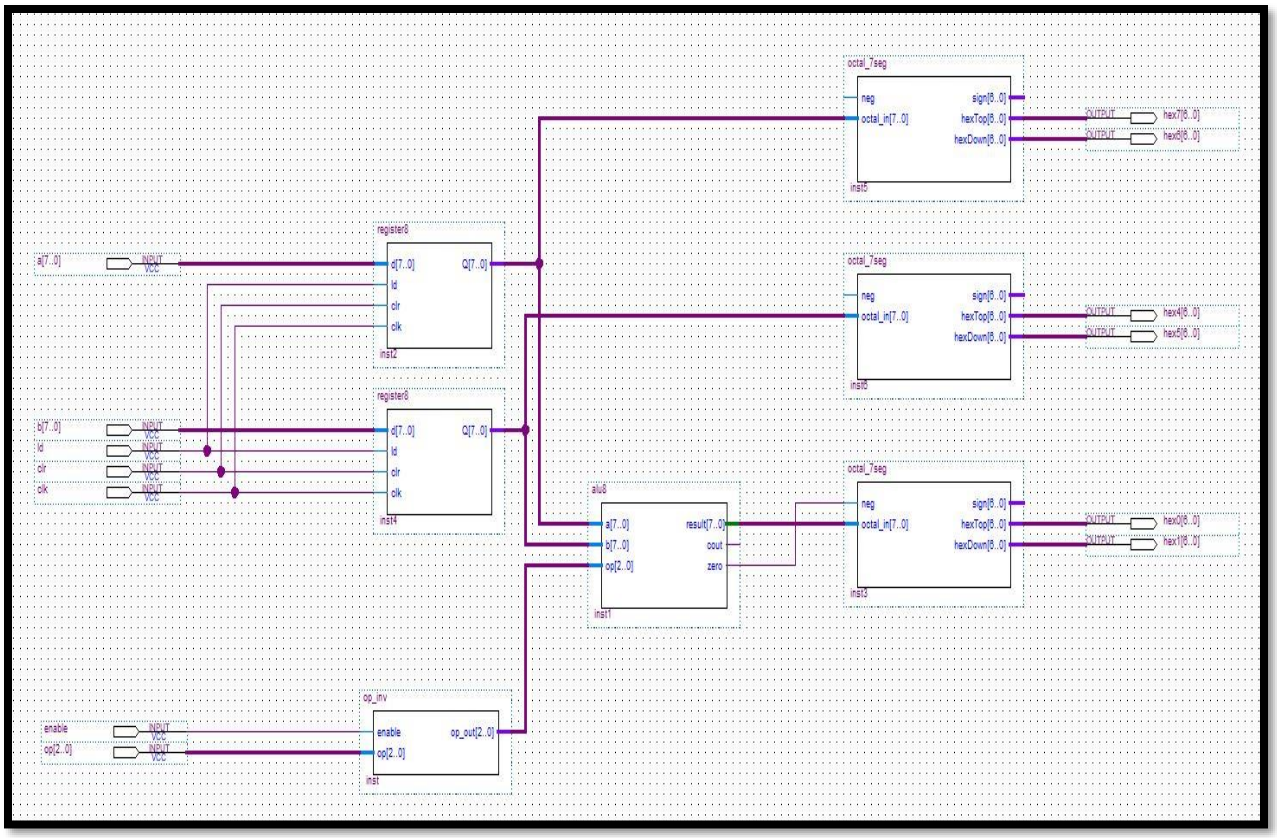


Figure 5: 8-bit ALU Schematic

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

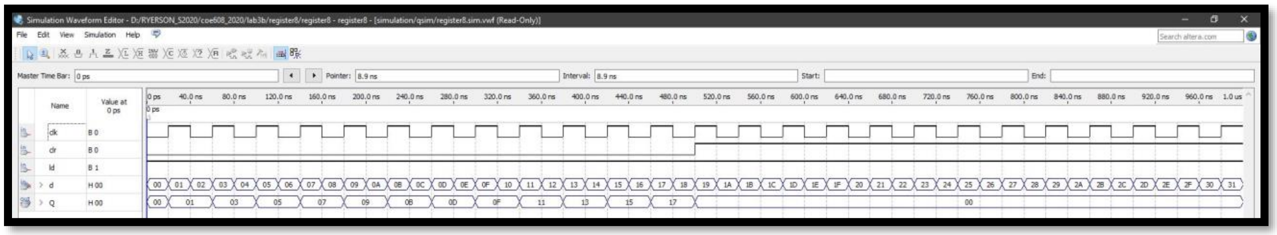
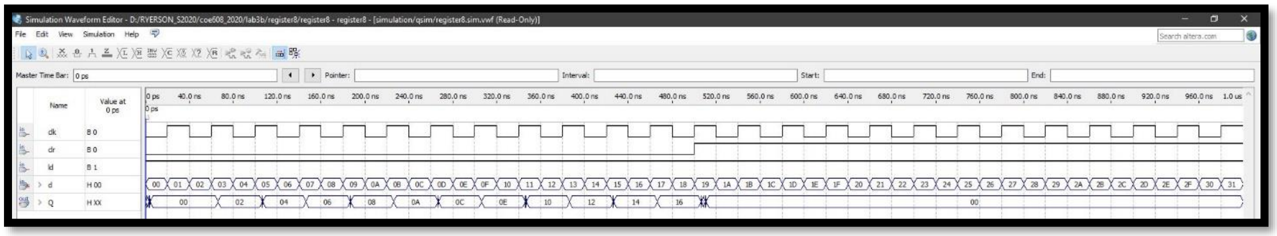

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/register8/register8 - register8 - [simulation/qsim/register8.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 8.9 ns
Interval: 8.9 ns
Start:
End:
Name
Value at 0 ps
clk B 0
clr B 0
ld B 1
d H 00
Q H 00

Figure 6: -bit register functional waveform

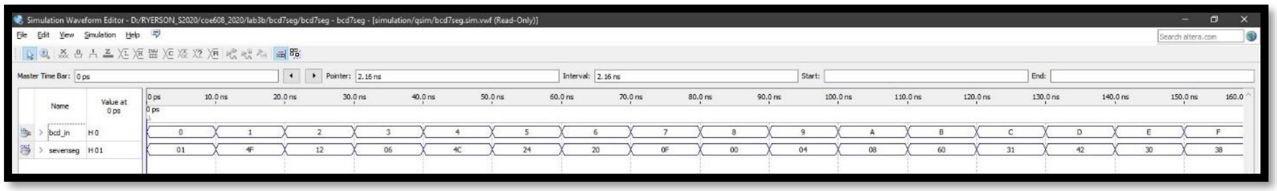

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/register8/register8 - register8 - [simulation/qsim/register8.sim.vwf (Read-Only)]

Master Time Bar: 0 ps

```
Name  Value at 0 ps
clk   B 0
clr   B 0
ld    B 1
d     H 00
Q     H XX
```

Figure 7: -bit register timing waveform

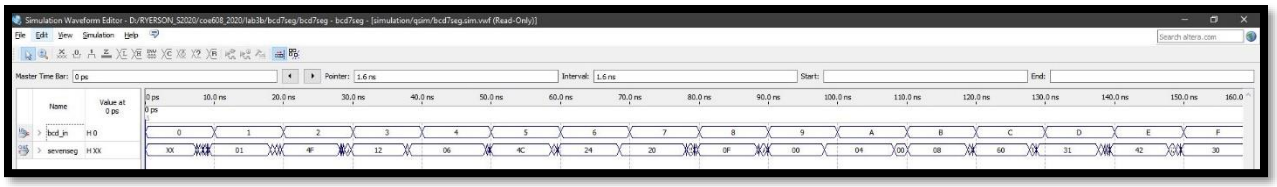
BCD to 7-segment converter: Functional and Timing Waveforms


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/bcd7seg/bcd7seg - bcd7seg - [simulation/qsim/bcd7seg.sim.vwf (Read-Only)]

Master Time Bar: 0 ps Pointer: 2.16 ns Interval: 2.16 ns

```
Name      Value at 0 ps
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
```

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

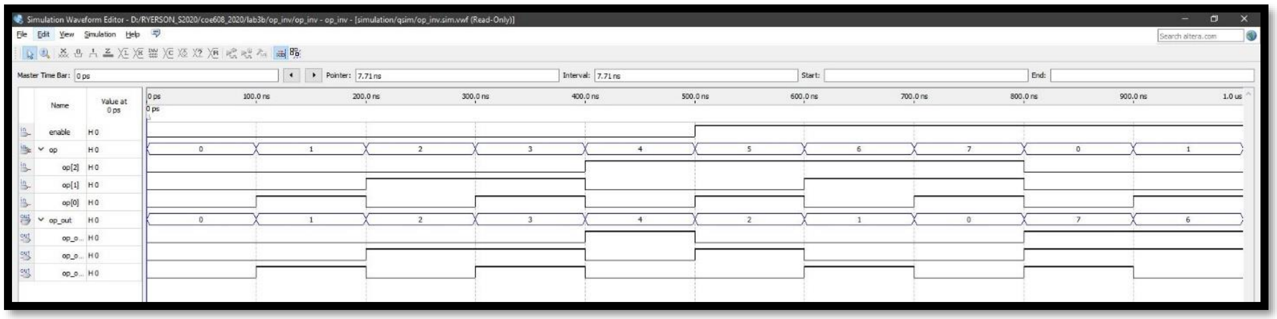

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/bcd7seg/bcd7seg - bcd7seg - [simulation/qsim/bcd7seg.sim.vwf (Read-Only)]

Master Time Bar: 0 ps Pointer: 1.6 ns Interval: 1.6 ns

```
Name      Value at 0 ps
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
```

Figure 9: BCD to 7-segment display timing waveform

Operational Inverter: Functional and Timing Waveforms


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/op_inv/op_inv - op_inv - [simulation/qsim/op_inv.sim.vwf (Read-Only)]

Master Time Bar: 0 ps Pointer: 7.71 ns Interval: 7.71 ns

```
Name      Value at 0 ps
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
```

Figure 10: Op-inverter functional waveform

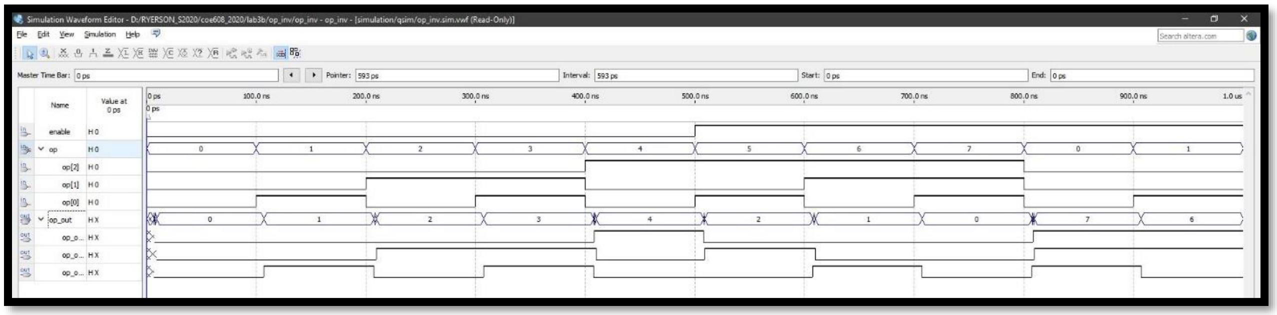


Figure 11: Op-inverter timing waveform

8-bit ALU: Functional and Timing Waveforms

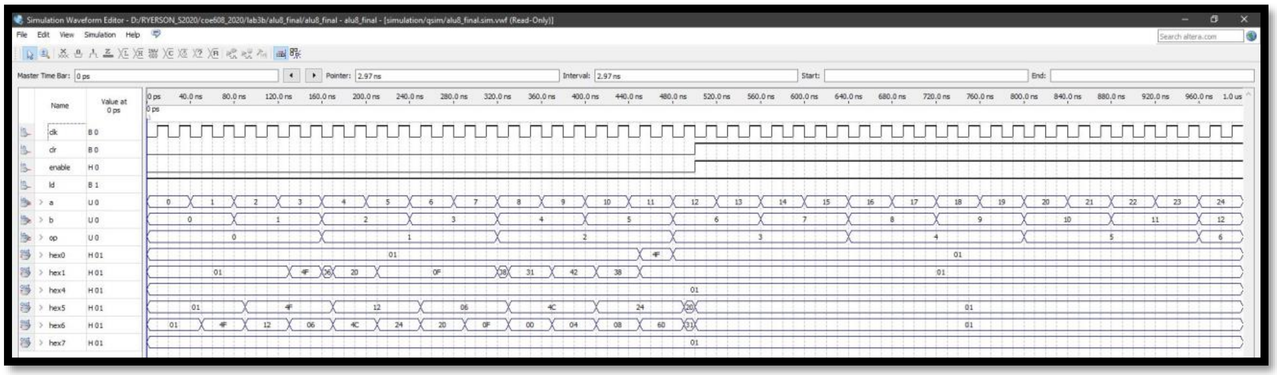


Figure 12: 8-bit ALU functional waveform

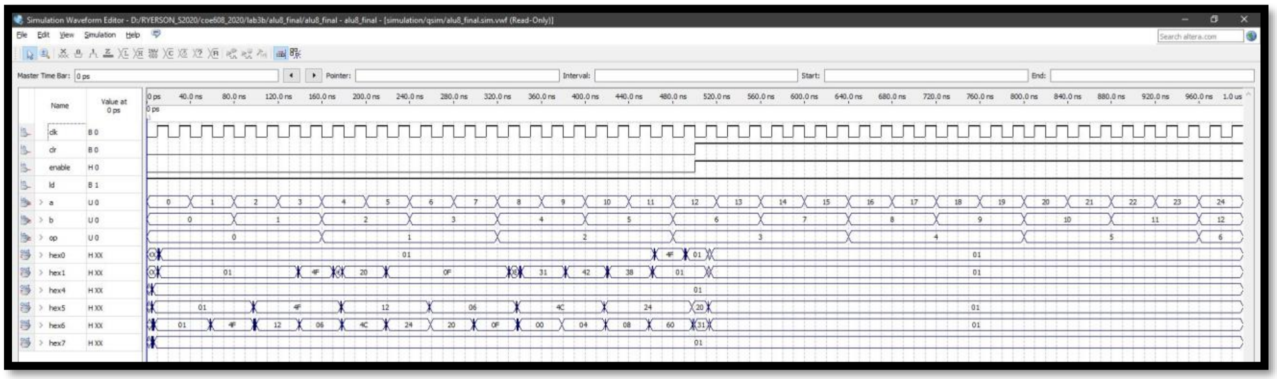
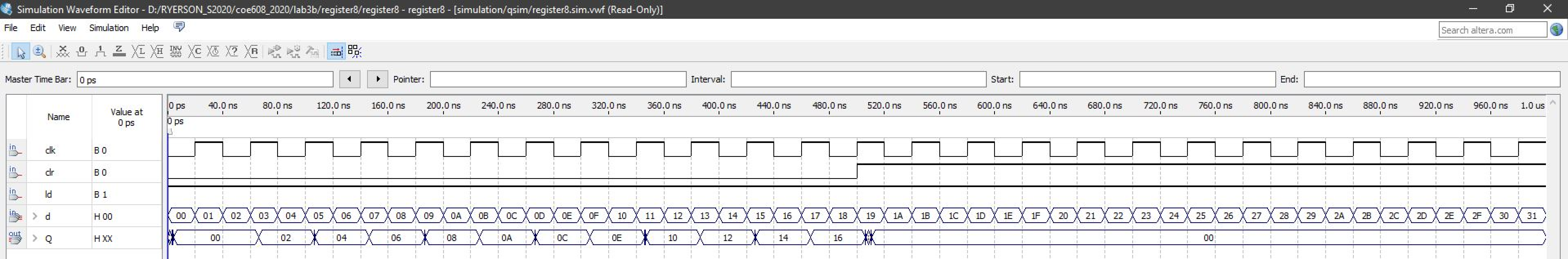


Figure 13: 8-bit ALU timing waveform

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.

```
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;
```

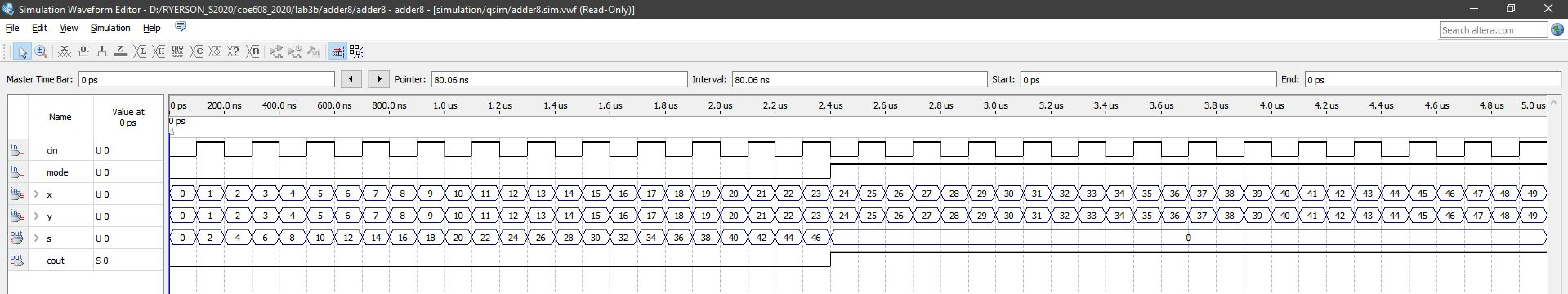


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/register8/register8 - register8 - [simulation/qsim/register8.sim.vwf (Read-Only)]

Master Time Bar: 0 ps   Pointer: Interval: Start: End:

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

Name	Value at 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)]

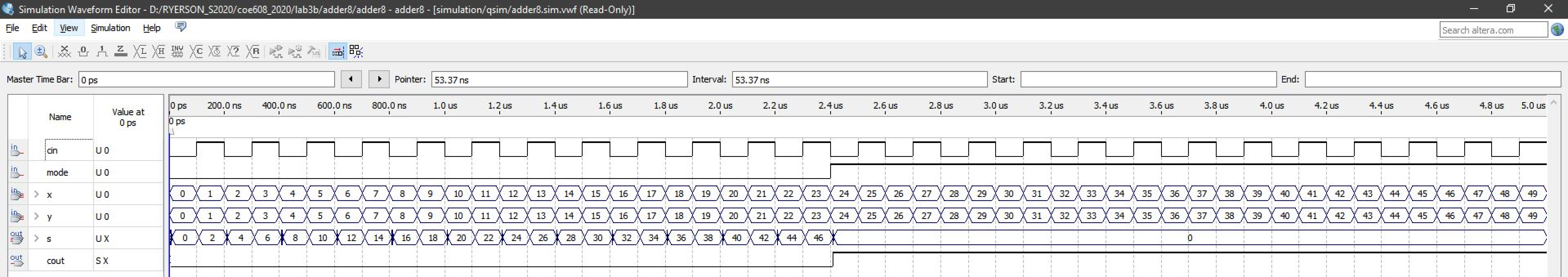
Master Time Bar: 0 ps
Pointer: 80.06 ns
Interval: 80.06 ns
Start: 0 ps
End: 0 ps

Name	Value at 0 ps
cin	U 0
mode	U 0
x	U 0
y	U 0
s	U 0
cout	S 0

x: 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: 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: 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



File
Edit
View
Simulation
Help
Search altera.com
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
x: 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: 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: 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

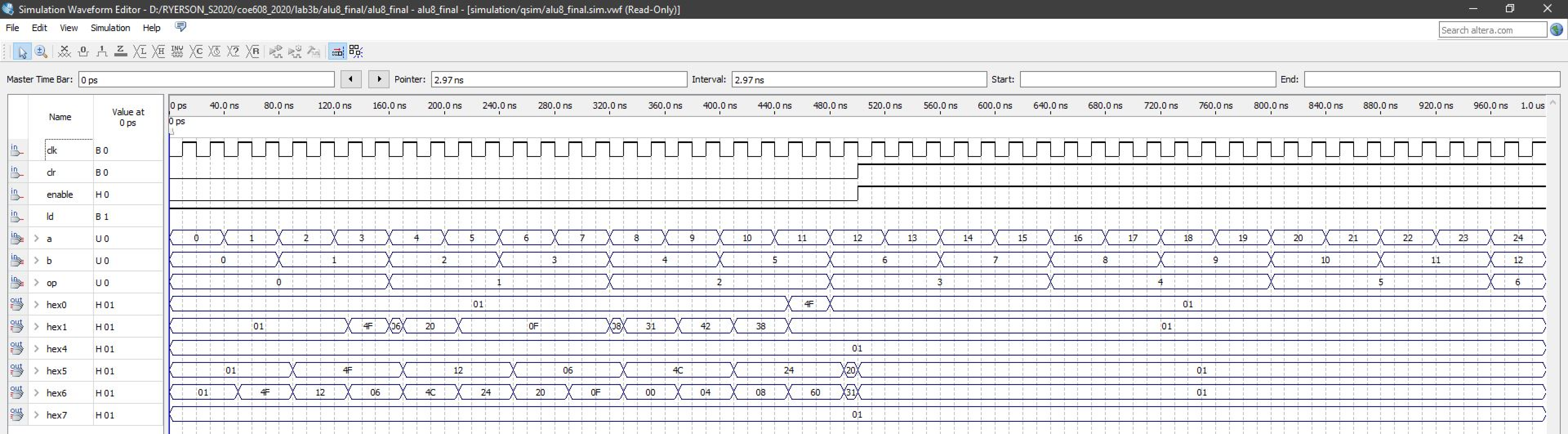
Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/adder8/adder8 - adder8 - [simulation/qsim/adder8.sim.vwf (Read-Only)]

Master Time Bar: 0 ps
Pointer: 53.37 ns
Interval: 53.37 ns
Start:
End:

Name	Value at 0 ps
cin	U 0
mode	U 0
x	U 0
y	U 0
s	U X
cout	S X

```
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;


```

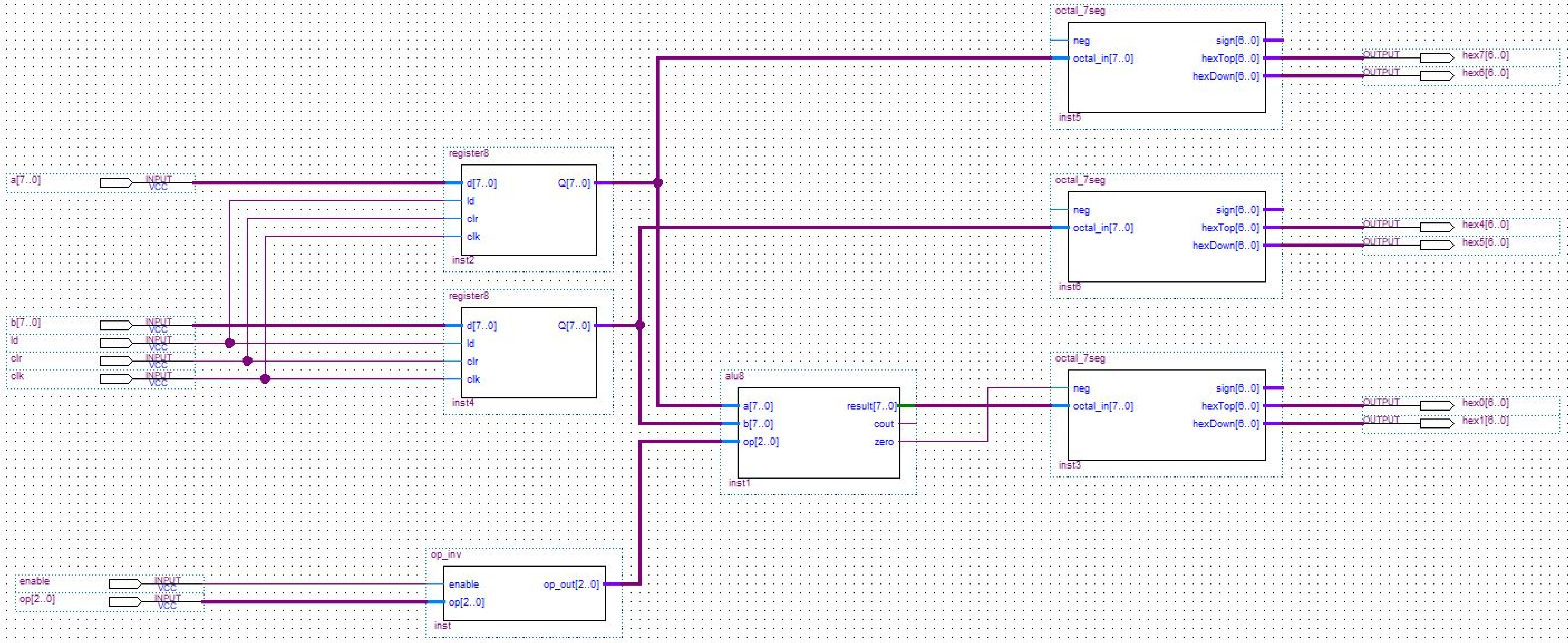
```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;

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;
```

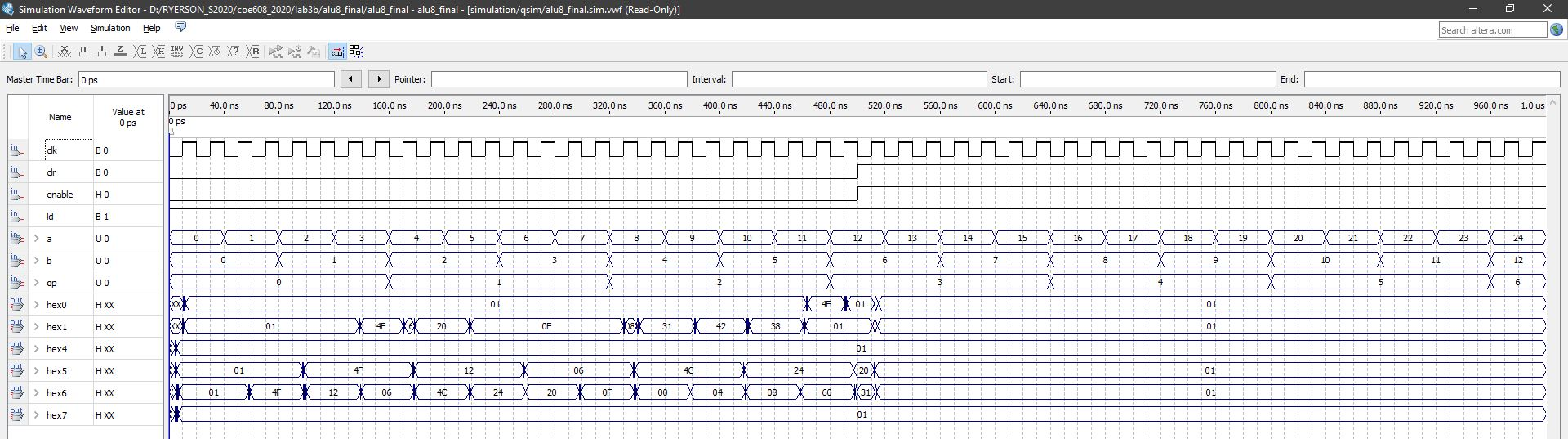


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/alu8_final/alu8_final - alu8_final - [simulation/qsim/alu8_final.sim.vwf (Read-Only)]

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

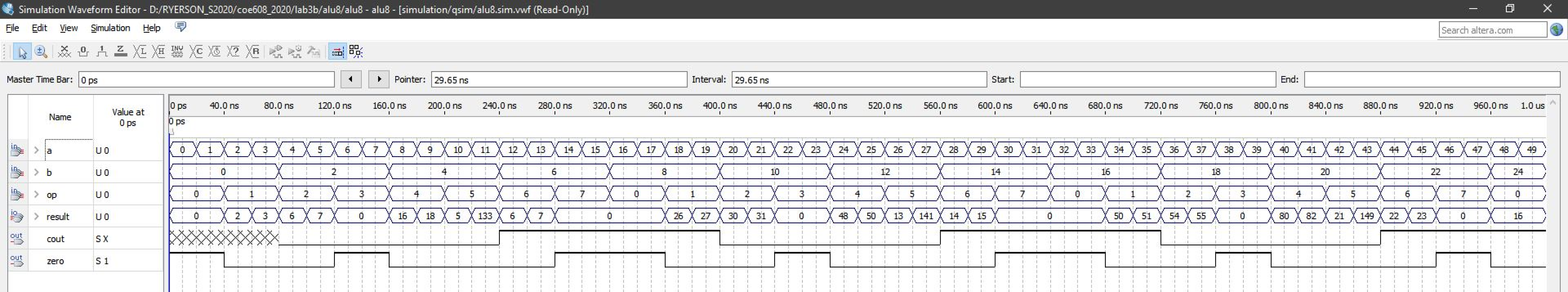

a[7..0]
b[7..0]
ld
clr
clk
INPUT VCC
register8
d[7..0]
ld
clr
clk
Q[7..0]
inst2
inst4
alu8
a[7..0]
b[7..0]
op[2..0]
result[7..0]
cout
zero
inst1
op_inv
enable
op[2..0]
op_out[2..0]
inst
octal_7seg
neg
octal_in[7..0]
sign[6..0]
hexTop[6..0]
hexDown[6..0]
inst5
inst6
inst3
OUTPUT
hex7[6..0]
hex6[6..0]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



Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/alu8_final/alu8_final - alu8_final - [simulation/qsim/alu8_final.sim.vwf (Read-Only)]

Master Time Bar: 0 ps
Pointer:
Interval:
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 XX
hex1	H XX
hex4	H XX
hex5	H XX
hex6	H XX
hex7	H XX


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/alu8/alu8 - alu8 - [simulation/qsim/alu8.sim.vwf (Read-Only)]

File Edit View Simulation Help

Master Time Bar: 0 ps

Pointer: 29.65 ns

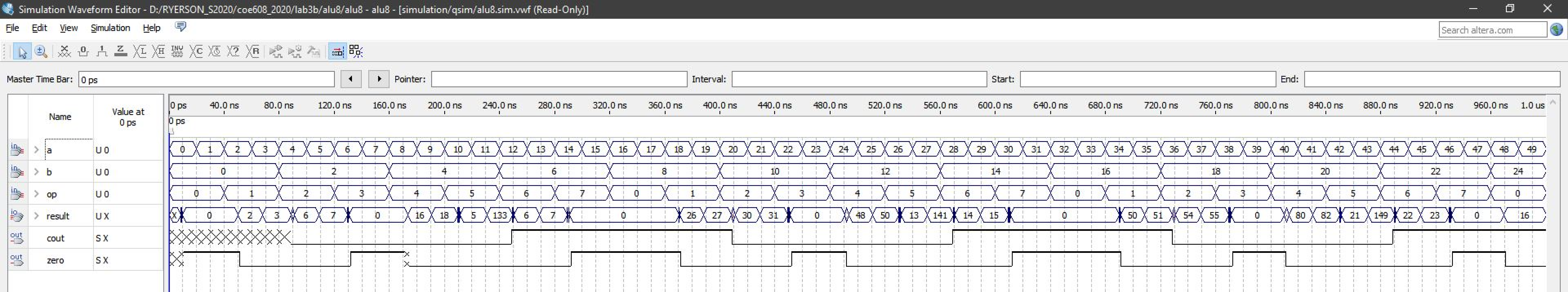
Interval: 29.65 ns

Start:

End:

Name	Value at 0 ps
a	U 0
b	U 0
op	U 0
result	U 0
cout	S X
zero	S 1

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: 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: 0 2 4 6 8 10 12 14 16 18 20 22 24
op: 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: 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

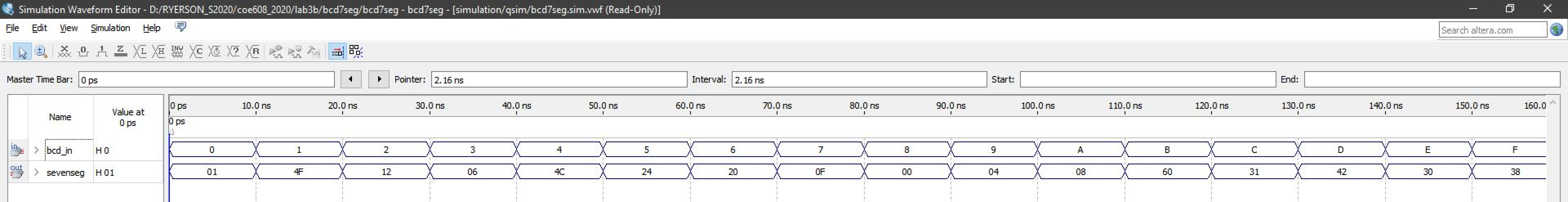

Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/alu8/alu8 - alu8 - [simulation/qsim/alu8.sim.vwf (Read-Only)]

File Edit View Simulation Help

Master Time Bar: 0 ps   Pointer: Interval: Start: End:

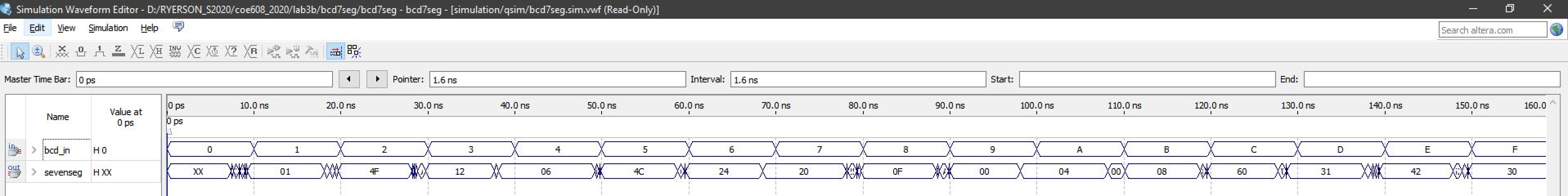
```
Name     Value at 0 ps   0 ps  40.0 ns  80.0 ns  120.0 ns  160.0 ns  200.0 ns  240.0 ns  280.0 ns  320.0 ns  360.0 ns  400.0 ns  440.0 ns  480.0 ns  520.0 ns  560.0 ns  600.0 ns  640.0 ns  680.0 ns  720.0 ns  760.0 ns  800.0 ns  840.0 ns  880.0 ns  920.0 ns  960.0 ns  1.0 us
a        U 0    0 1 2 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)]

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
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

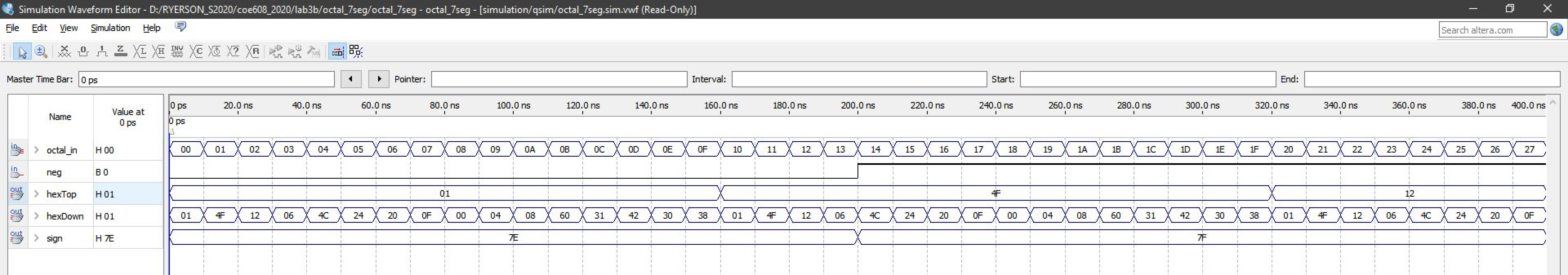


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/bcd7seg/bcd7seg - bcd7seg - [simulation/qsim/bcd7seg.sim.vwf (Read-Only)]

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
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;

```

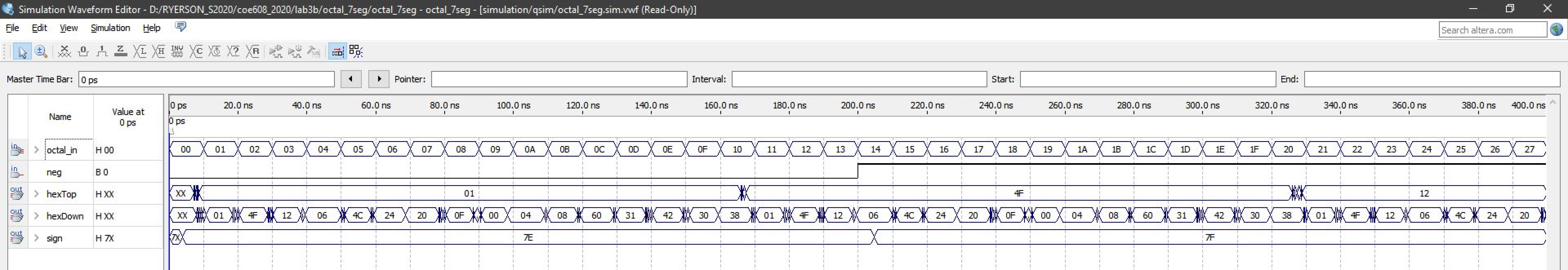


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/octal_7seg/octal_7seg - octal_7seg - [simulation/qsim/octal_7seg.sim.vwf (Read-Only)]

Master Time Bar: 0 ps   Pointer: Interval: Start: End:

Name	Value at 0 ps
octal_in	H 00
neg	B 0
hexTop	H 01
hexDown	H 01
sign	H 7E

```
          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  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
hexTop    01 | 4F | 12
hexDown   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      7E | 7F
```



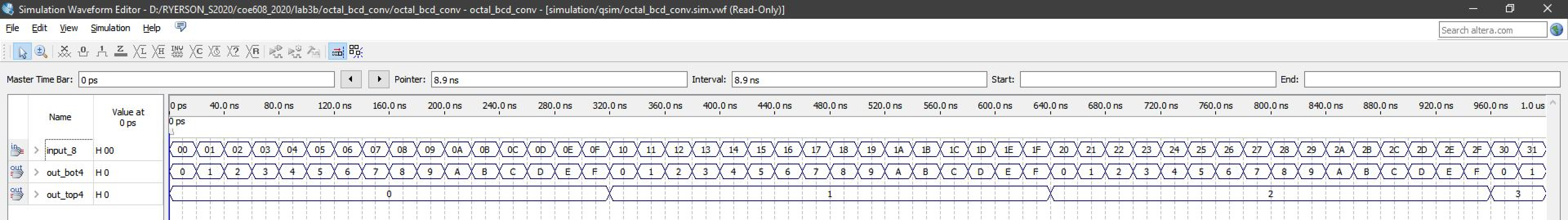
Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/octal_7seg/octal_7seg - octal_7seg - [simulation/qsim/octal_7seg.sim.vwf (Read-Only)]

Master Time Bar: 0 ps Pointer: Interval: Start: End:

Name	Value at 0 ps
octal_in	H 00
neg	B 0
hexTop	H XX
hexDown	H XX
sign	H 7X

```
octal_in: 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
hexTop:   XX 01 4F 12
hexDown:  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:     7X 7E 7F
```

```vhdl
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:

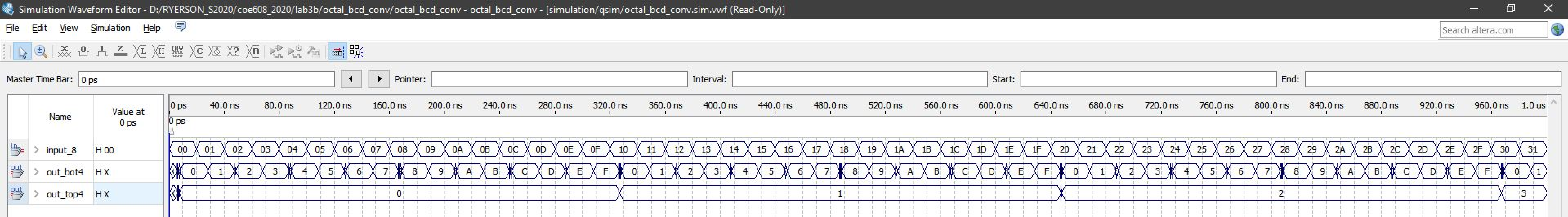
Name	Value at 0 ps
input_8	H 00
out_bot4	H 0
out_top4	H 0

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: 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: 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: 0 1 2 3

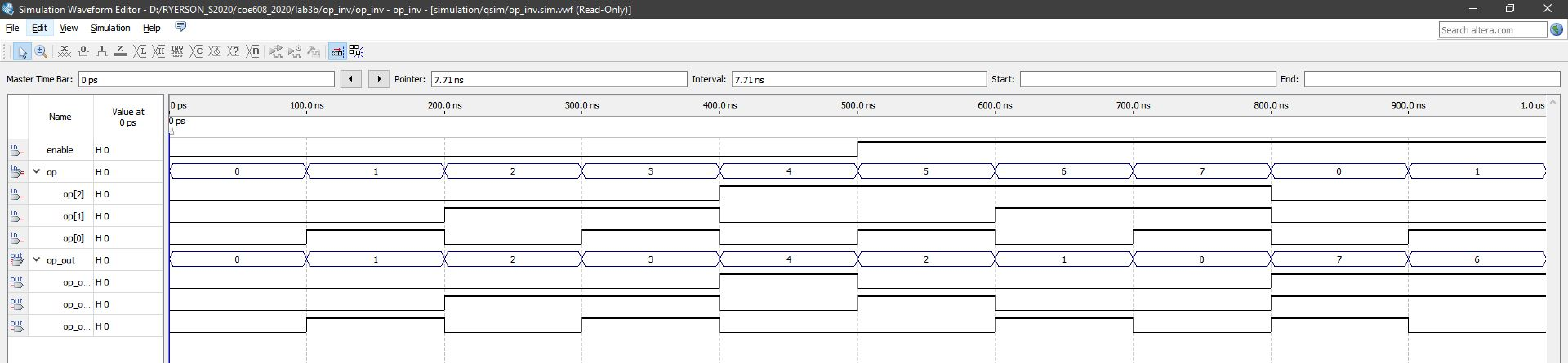


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/octal_bcd_conv/octal_bcd_conv - octal_bcd_conv - [simulation/qsim/octal_bcd_conv.sim.vwf (Read-Only)]

Master Time Bar: 0 ps   Pointer: Interval: Start: End:

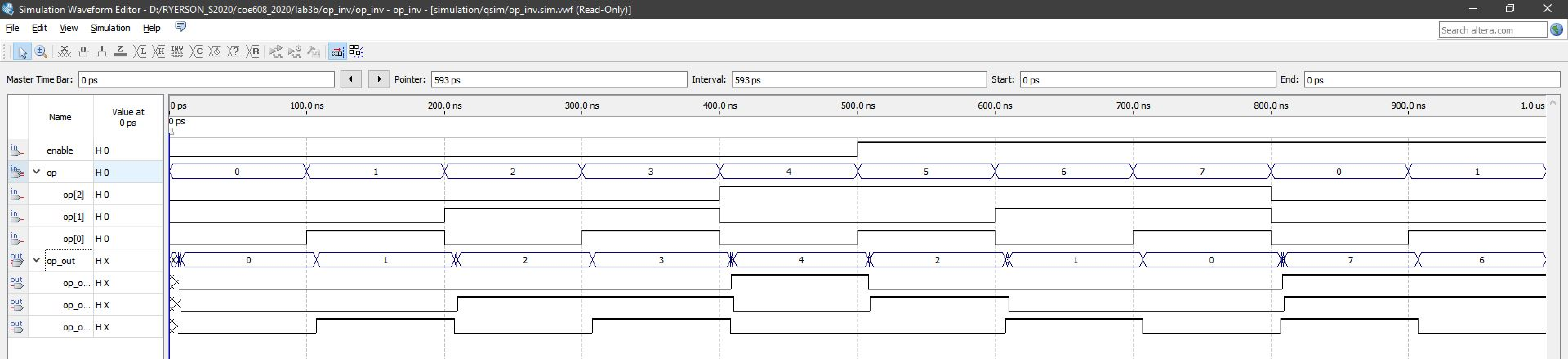
```
            0 ps 40.0 ns 80.0 ns 120.0 ns 160.0 ns 200.0 ns 240.0 ns 280.0 ns 320.0 ns 360.0 ns 400.0 ns 440.0 ns 480.0 ns 520.0 ns 560.0 ns 600.0 ns 640.0 ns 680.0 ns 720.0 ns 760.0 ns 800.0 ns 840.0 ns 880.0 ns 920.0 ns 960.0 ns 1.0 us
input_8   H 00  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 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
```

```vhdl
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)]

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
op	H 0	0	1	2	3	4	5	6	7	0	1
op_out	H 0	0	1	2	3	4	2	1	0	7	6

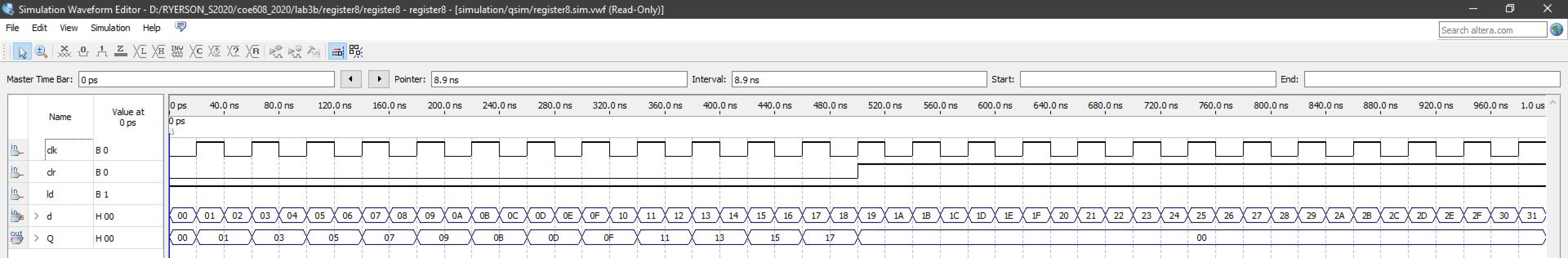


Simulation Waveform Editor - D:/RYERSON_S2020/coe608_2020/lab3b/op_inv/op_inv - op_inv - [simulation/qsim/op_inv.sim.vwf (Read-Only)]

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
enable	H 0	0	0	0	0	0	1	1	1	1	1
op	H 0	0	1	2	3	4	5	6	7	0	1
op[2]	H 0	0	0	0	0	1	1	1	1	0	0
op[1]	H 0	0	0	1	1	0	0	1	1	0	0
op[0]	H 0	0	1	0	1	0	1	0	1	0	1
op_out	H X	0	1	2	3	4	2	1	0	7	6
op_o...	H X	0	0	0	0	1	0	0	0	1	1
op_o...	H X	0	0	1	1	0	1	0	0	1	1
op_o...	H X	0	1	0	1	0	0	1	0	1	0

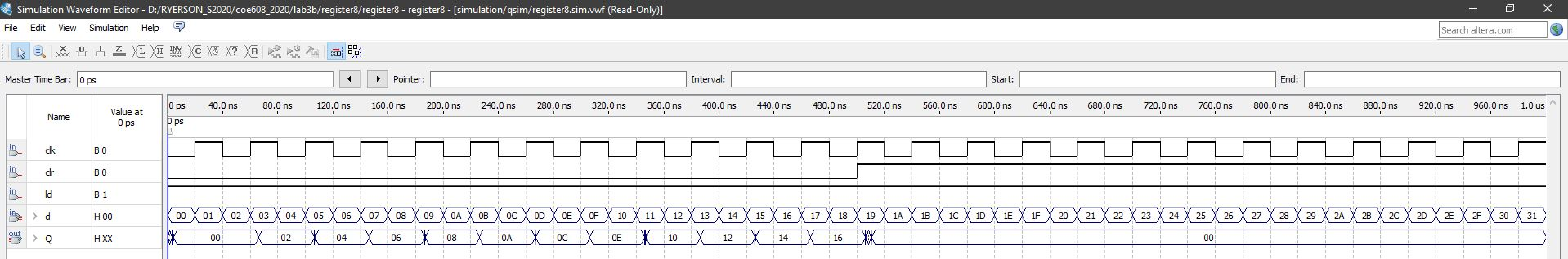
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity register8 is
   port (
   d: in std_logic_vector(7 downto 0);
   ld,clr,clk: in std_logic;
   Q: out std_logic_vector(7 downto 0));
end register8;

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)]

Master Time Bar: 0 ps
Pointer: 8.9 ns
Interval: 8.9 ns
Start:
End:

Name	Value at 0 ps
clk	B 0
clr	B 0
ld	B 1
d	H 00
Q	H 00

```
d: 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: 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)]

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

Name	Value at 0 ps
clk	B 0
clr	B 0
ld	B 1
d	H 00
Q	H XX