Lab 2 Report

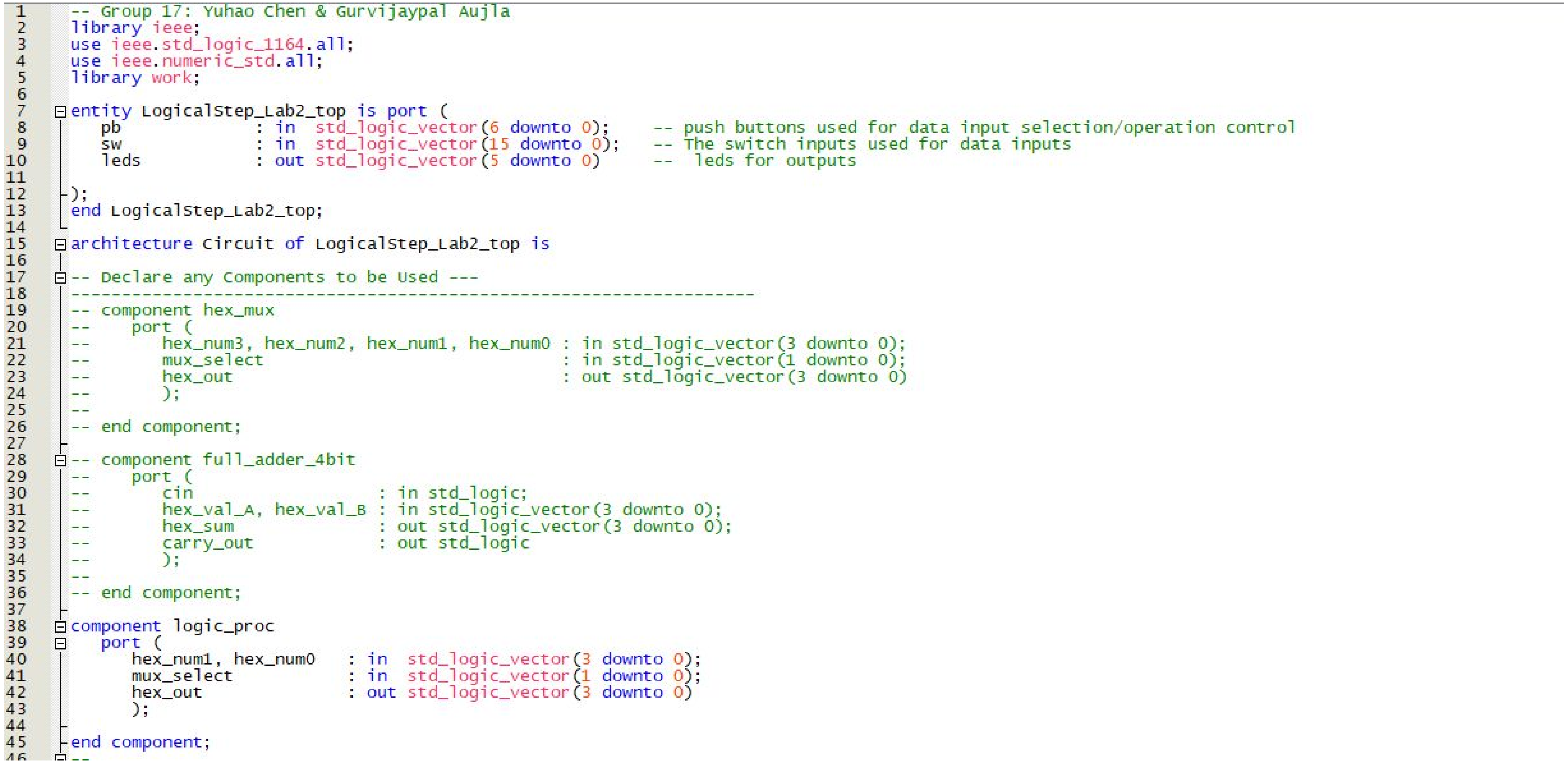
ECE 124

Group 17 Session 204

Gurvijaypal Aujla

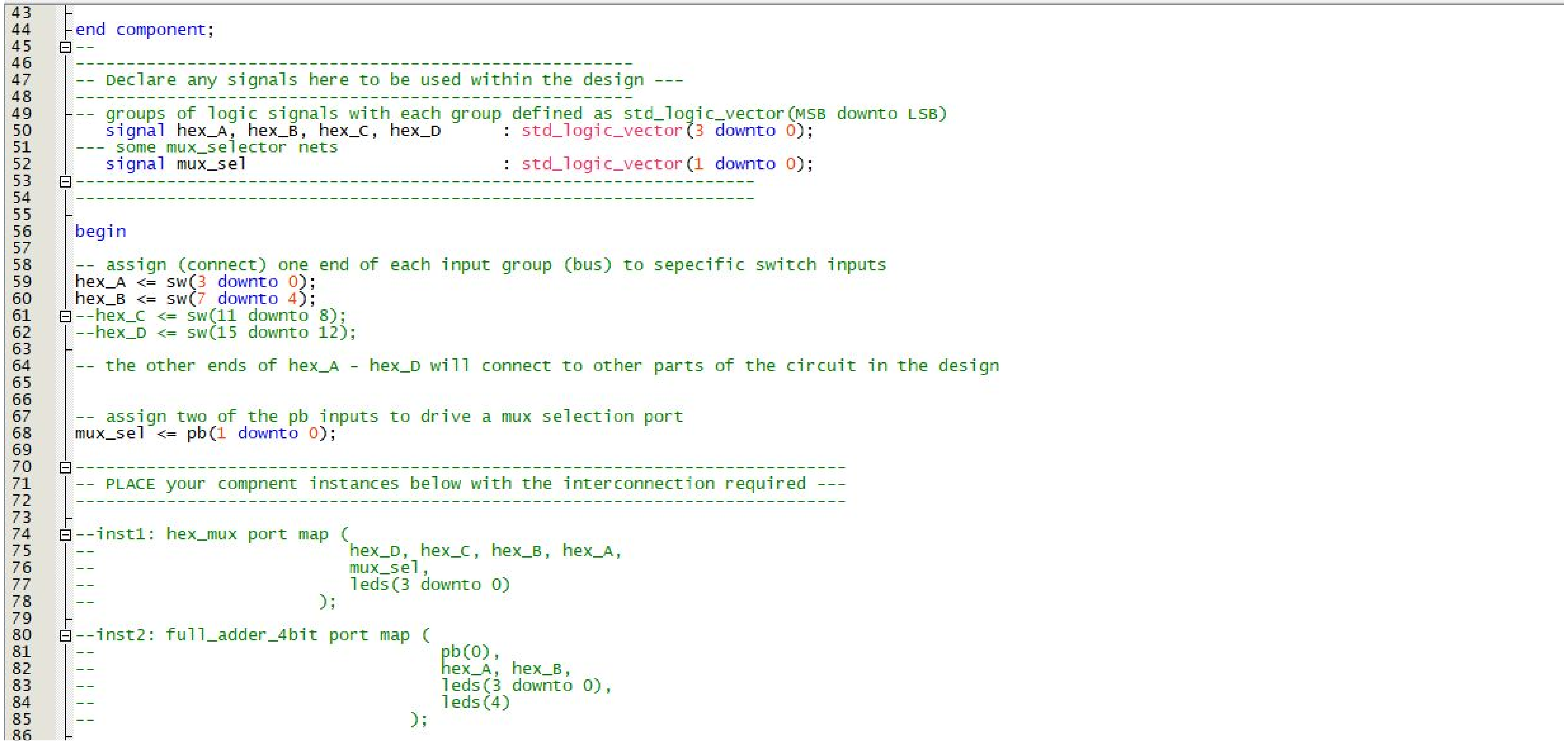
Yuhao Chen

Top-level Design

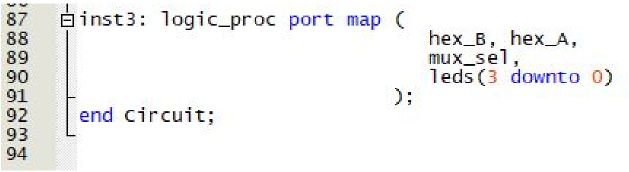


```vhdl
-- Group 17: Yuhao Chen & Gurvijaypal Aujla
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
library work;

entity LogicalStep_Lab2_top is port (
   pb              : in  std_logic_vector(6 downto 0);     -- push buttons used for data input selection/operation control
   sw              : in  std_logic_vector(15 downto 0);    -- The switch inputs used for data inputs
   leds            : out std_logic_vector(5 downto 0)      --  leds for outputs

);
end LogicalStep_Lab2_top;

architecture Circuit of LogicalStep_Lab2_top is

-- Declare any Components to be Used ---
-------------------------------------------------------------------
-- component hex_mux
--    port (
--       hex_num3, hex_num2, hex_num1, hex_num0 : in std_logic_vector(3 downto 0);
--       mux_select                             : in std_logic_vector(1 downto 0);
--       hex_out                                : out std_logic_vector(3 downto 0)
--       );
--
-- end component;

-- component full_adder_4bit
--    port (
--       cin                  : in std_logic;
--       hex_val_A, hex_val_B : in std_logic_vector(3 downto 0);
--       hex_sum              : out std_logic_vector(3 downto 0);
--       carry_out            : out std_logic
--       );
--
-- end component;

component logic_proc
   port (
      hex_num1, hex_num0   : in  std_logic_vector(3 downto 0);
      mux_select           : in  std_logic_vector(1 downto 0);
      hex_out              : out std_logic_vector(3 downto 0)
      );

end component;
--
```

Top-level Design Continued



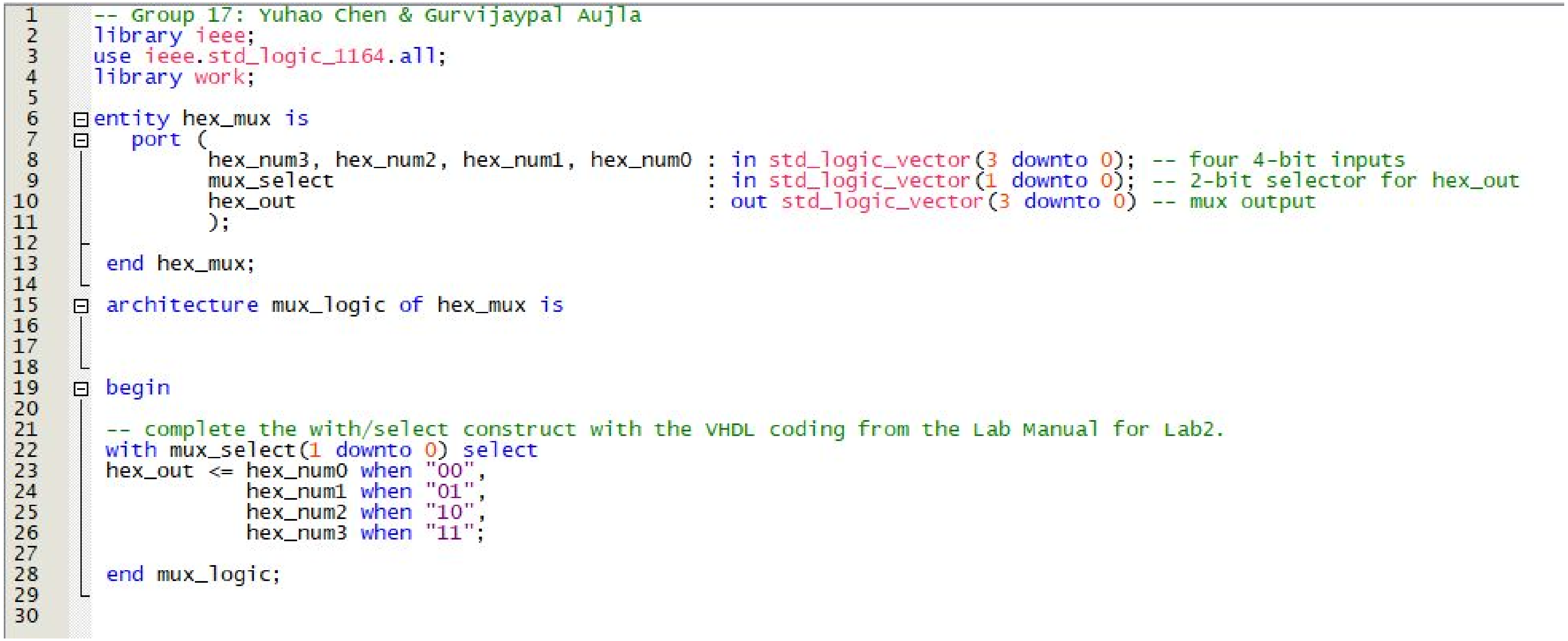
```
end component;
--
-------------------------------------------------------
-- Declare any signals here to be used within the design ---
-------------------------------------------------------
-- groups of logic signals with each group defined as std_logic_vector(MSB downto LSB)
   signal hex_A, hex_B, hex_C, hex_D      : std_logic_vector(3 downto 0);
--- some mux_selector nets
   signal mux_sel                         : std_logic_vector(1 downto 0);
-------------------------------------------------------------------
-------------------------------------------------------------------

begin

-- assign (connect) one end of each input group (bus) to sepecific switch inputs
hex_A <= sw(3 downto 0);
hex_B <= sw(7 downto 4);
--hex_C <= sw(11 downto 8);
--hex_D <= sw(15 downto 12);

-- the other ends of hex_A - hex_D will connect to other parts of the circuit in the design


-- assign two of the pb inputs to drive a mux selection port
mux_sel <= pb(1 downto 0);

----------------------------------------------------------------------------
-- PLACE your compnent instances below with the interconnection required ---
----------------------------------------------------------------------------

--inst1: hex_mux port map (
--                         hex_D, hex_C, hex_B, hex_A,
--                         mux_sel,
--                         leds(3 downto 0)
--                      );

--inst2: full_adder_4bit port map (
--                                  pb(0),
--                                  hex_A, hex_B,
--                                  leds(3 downto 0),
--                                  leds(4)
--                               );
```



```
inst3: logic_proc port map (
                            hex_B, hex_A,
                            mux_sel,
                            leds(3 downto 0)
                           );
end Circuit;
```

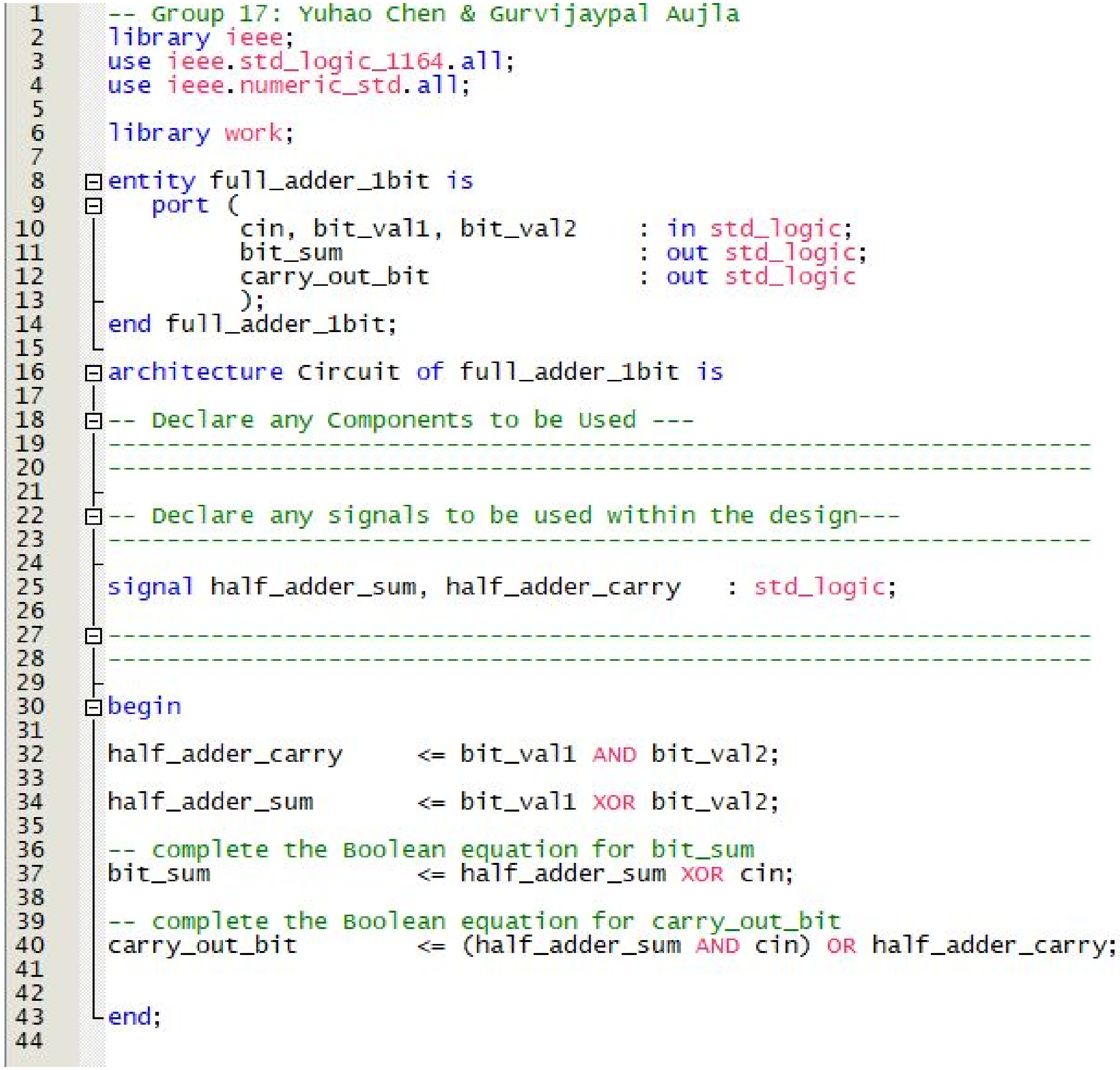
t

4 to 1 Multiplexer Design



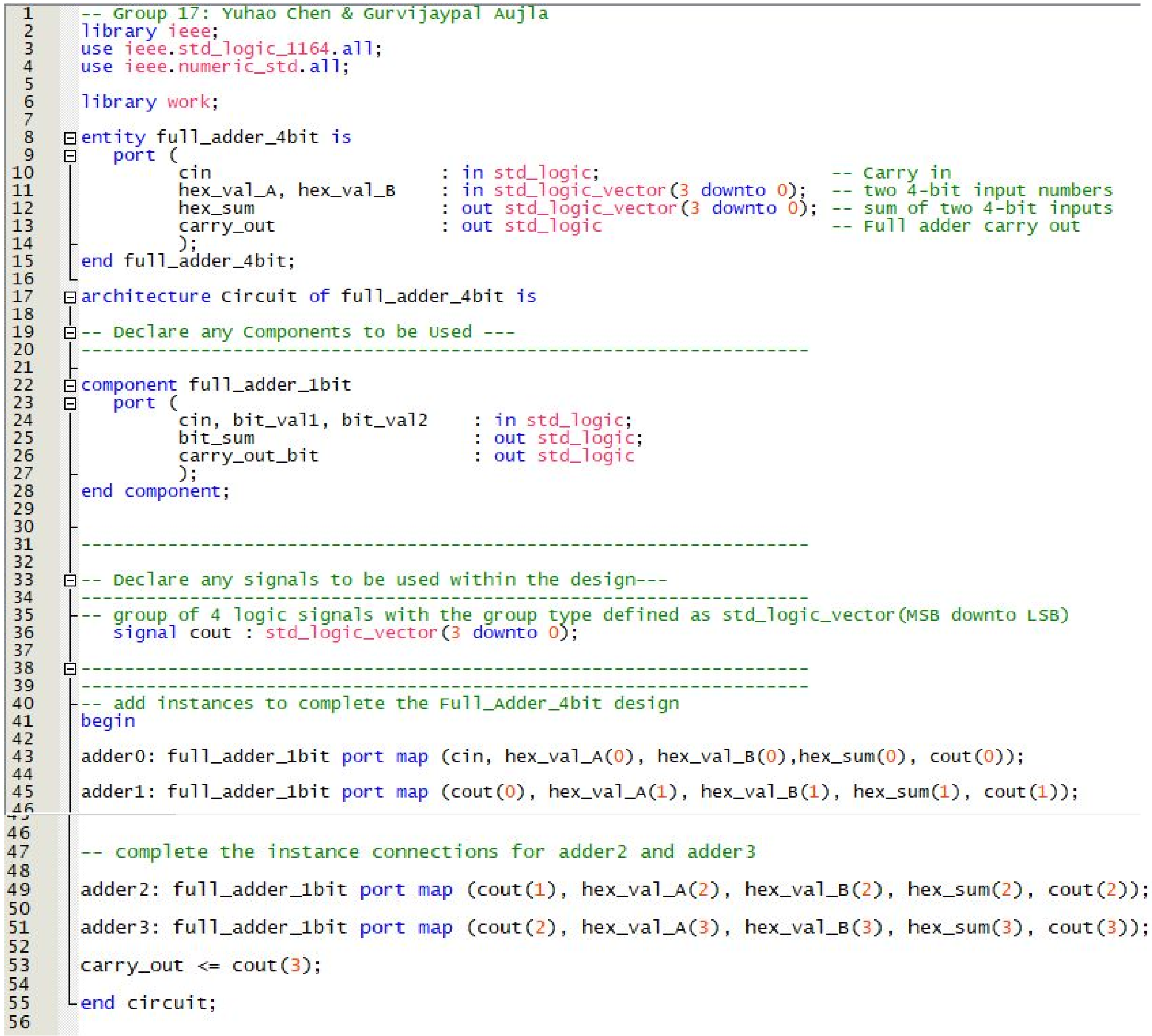
```
-- Group 17: Yuhao Chen & Gurvijaypal Aujla
library ieee;
use ieee.std_logic_1164.all;
library work;

entity hex_mux is
   port (
         hex_num3, hex_num2, hex_num1, hex_num0 : in std_logic_vector(3 downto 0); -- four 4-bit inputs
         mux_select                             : in std_logic_vector(1 downto 0); -- 2-bit selector for hex_out
         hex_out                                : out std_logic_vector(3 downto 0) -- mux output
         );

 end hex_mux;

 architecture mux_logic of hex_mux is



 begin

 -- complete the with/select construct with the VHDL coding from the Lab Manual for Lab2.
 with mux_select(1 downto 0) select
 hex_out <= hex_num0 when "00",
            hex_num1 when "01",
            hex_num2 when "10",
            hex_num3 when "11";

 end mux_logic;
```

1-bit Full Adder Design



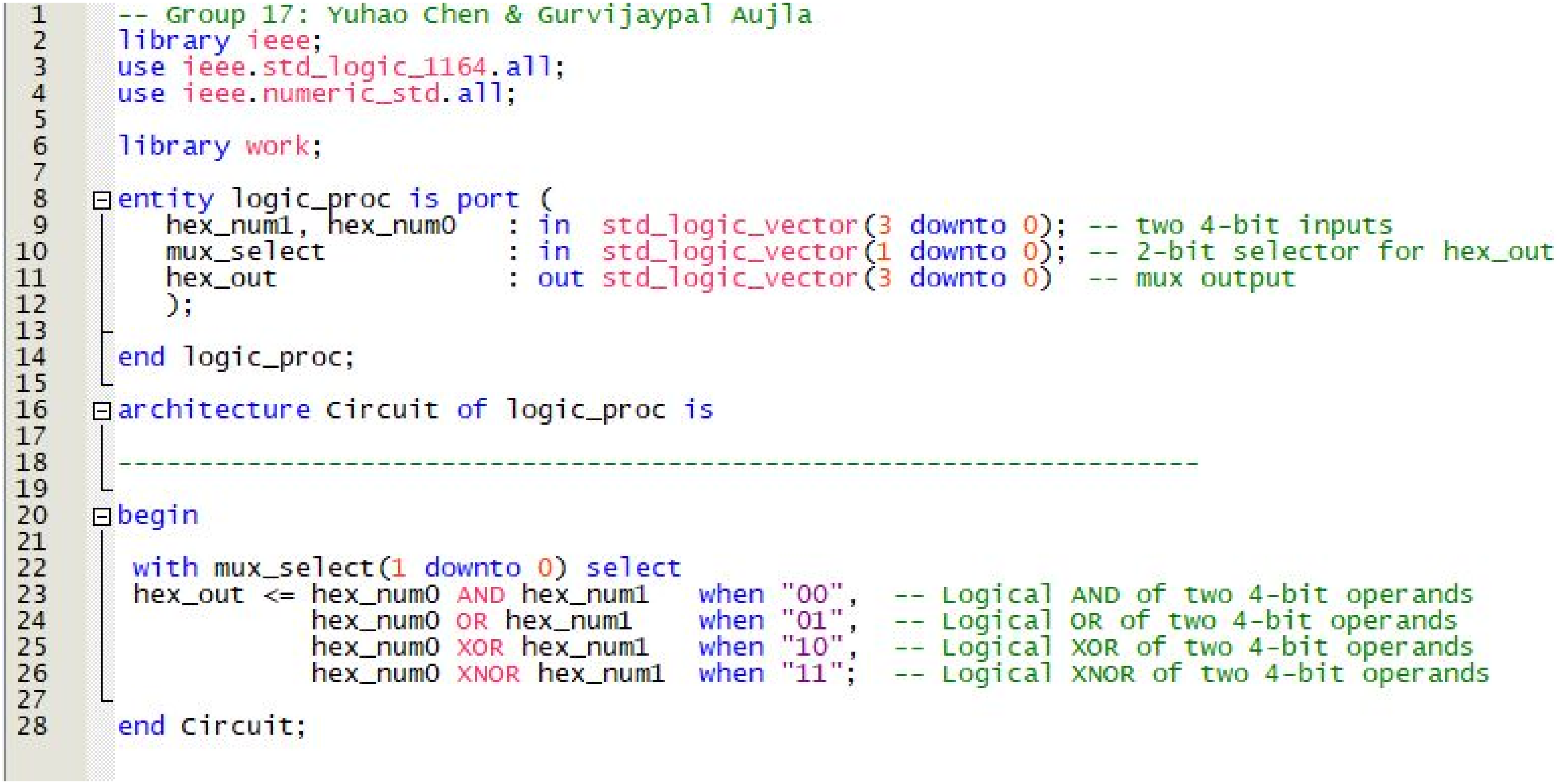
```vhdl
-- Group 17: Yuhao Chen & Gurvijaypal Aujla
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

library work;

entity full_adder_1bit is
   port (
         cin, bit_val1, bit_val2    : in std_logic;
         bit_sum                    : out std_logic;
         carry_out_bit              : out std_logic
         );
end full_adder_1bit;

architecture Circuit of full_adder_1bit is

-- Declare any Components to be Used ---
-------------------------------------------------------------------
-------------------------------------------------------------------

-- Declare any signals to be used within the design---
-------------------------------------------------------------------

signal half_adder_sum, half_adder_carry   : std_logic;

-------------------------------------------------------------------
-------------------------------------------------------------------

begin

half_adder_carry     <= bit_val1 AND bit_val2;

half_adder_sum       <= bit_val1 XOR bit_val2;

-- complete the Boolean equation for bit_sum
bit_sum              <= half_adder_sum XOR cin;

-- complete the Boolean equation for carry_out_bit
carry_out_bit        <= (half_adder_sum AND cin) OR half_adder_carry;


end;
```

4-bit Full Adder Design



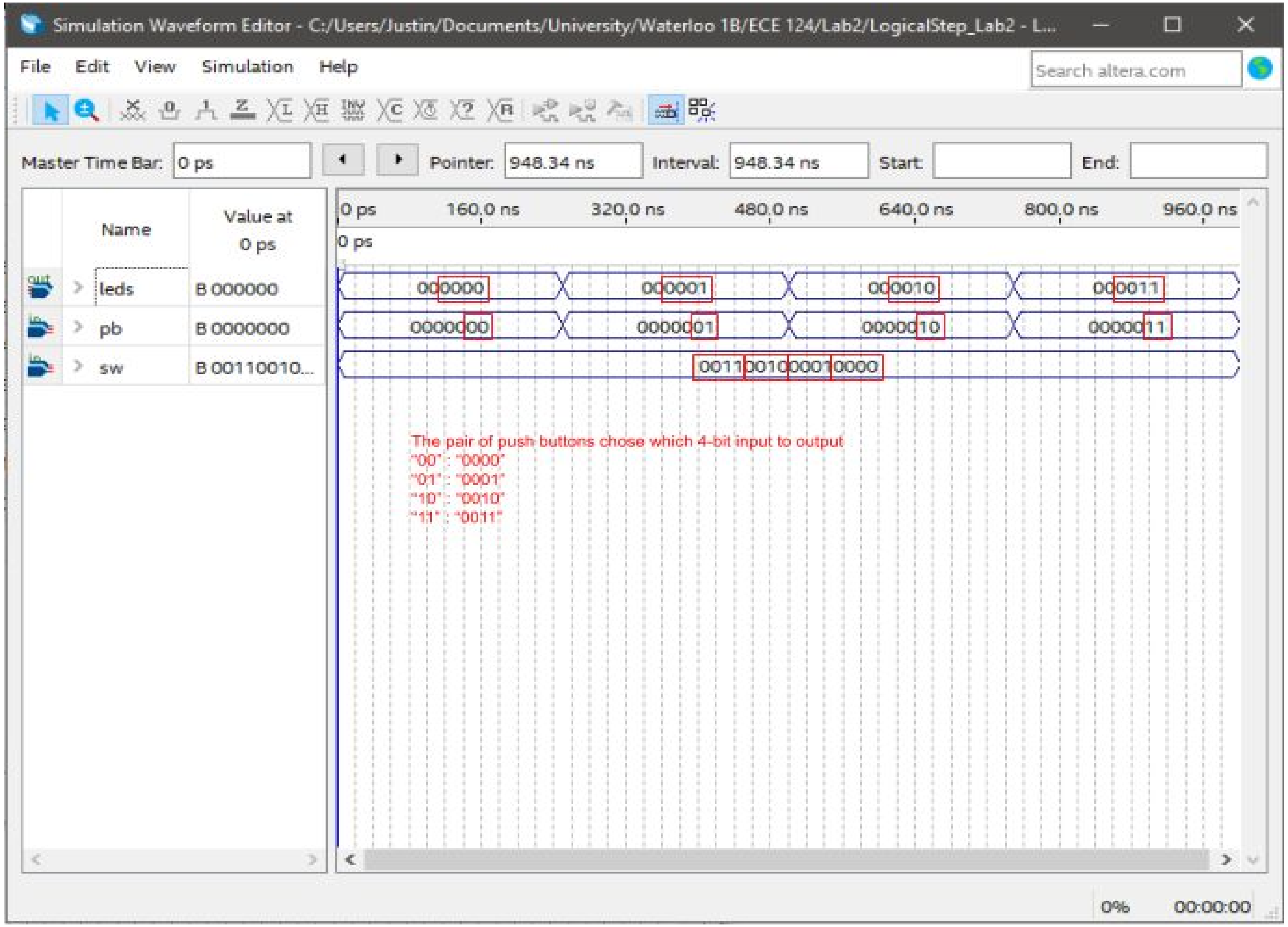
```
-- Group 17: Yuhao Chen & Gurvijaypal Aujla
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

library work;

entity full_adder_4bit is
   port (
         cin                      : in std_logic;                       -- Carry in
         hex_val_A, hex_val_B     : in std_logic_vector(3 downto 0);    -- two 4-bit input numbers
         hex_sum                  : out std_logic_vector(3 downto 0);   -- sum of two 4-bit inputs
         carry_out                : out std_logic                       -- Full adder carry out
         );
end full_adder_4bit;

architecture circuit of full_adder_4bit is

-- Declare any Components to be Used ---
-------------------------------------------------------------------

component full_adder_1bit
   port (
         cin, bit_val1, bit_val2     : in std_logic;
         bit_sum                     : out std_logic;
         carry_out_bit               : out std_logic
         );
end component;


-------------------------------------------------------------------

-- Declare any signals to be used within the design---
-------------------------------------------------------------------
-- group of 4 logic signals with the group type defined as std_logic_vector(MSB downto LSB)
   signal cout : std_logic_vector(3 downto 0);

-------------------------------------------------------------------
-------------------------------------------------------------------
-- add instances to complete the Full_Adder_4bit design
begin

adder0: full_adder_1bit port map (cin, hex_val_A(0), hex_val_B(0),hex_sum(0), cout(0));

adder1: full_adder_1bit port map (cout(0), hex_val_A(1), hex_val_B(1), hex_sum(1), cout(1));


-- complete the instance connections for adder2 and adder3

adder2: full_adder_1bit port map (cout(1), hex_val_A(2), hex_val_B(2), hex_sum(2), cout(2));

adder3: full_adder_1bit port map (cout(2), hex_val_A(3), hex_val_B(3), hex_sum(3), cout(3));

carry_out <= cout(3);

end circuit;
```

Logic Processor Design



```vhdl
-- Group 17: Yuhao Chen & Gurvijaypal Aujla
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

library work;

entity logic_proc is port (
    hex_num1, hex_num0    : in  std_logic_vector(3 downto 0); -- two 4-bit inputs
    mux_select            : in  std_logic_vector(1 downto 0); -- 2-bit selector for hex_out
    hex_out               : out std_logic_vector(3 downto 0)  -- mux output
    );

end logic_proc;

architecture Circuit of logic_proc is

-------------------------------------------------------------------

begin

 with mux_select(1 downto 0) select
 hex_out <= hex_num0 AND hex_num1   when "00",  -- Logical AND of two 4-bit operands
            hex_num0 OR hex_num1    when "01",  -- Logical OR of two 4-bit operands
            hex_num0 XOR hex_num1   when "10",  -- Logical XOR of two 4-bit operands
            hex_num0 XNOR hex_num1  when "11";  -- Logical XNOR of two 4-bit operands

end Circuit;
```

4 4-bit to 1 4-bit Multiplexer (MUX) Waveform



Simulation Waveform Editor - C:/Users/Justin/Documents/University/Waterloo 1B/ECE 124/Lab2/LogicalStep_Lab2 - L...

Master Time Bar: 0 ps

Pointer: 948.34 ns

Interval: 948.34 ns

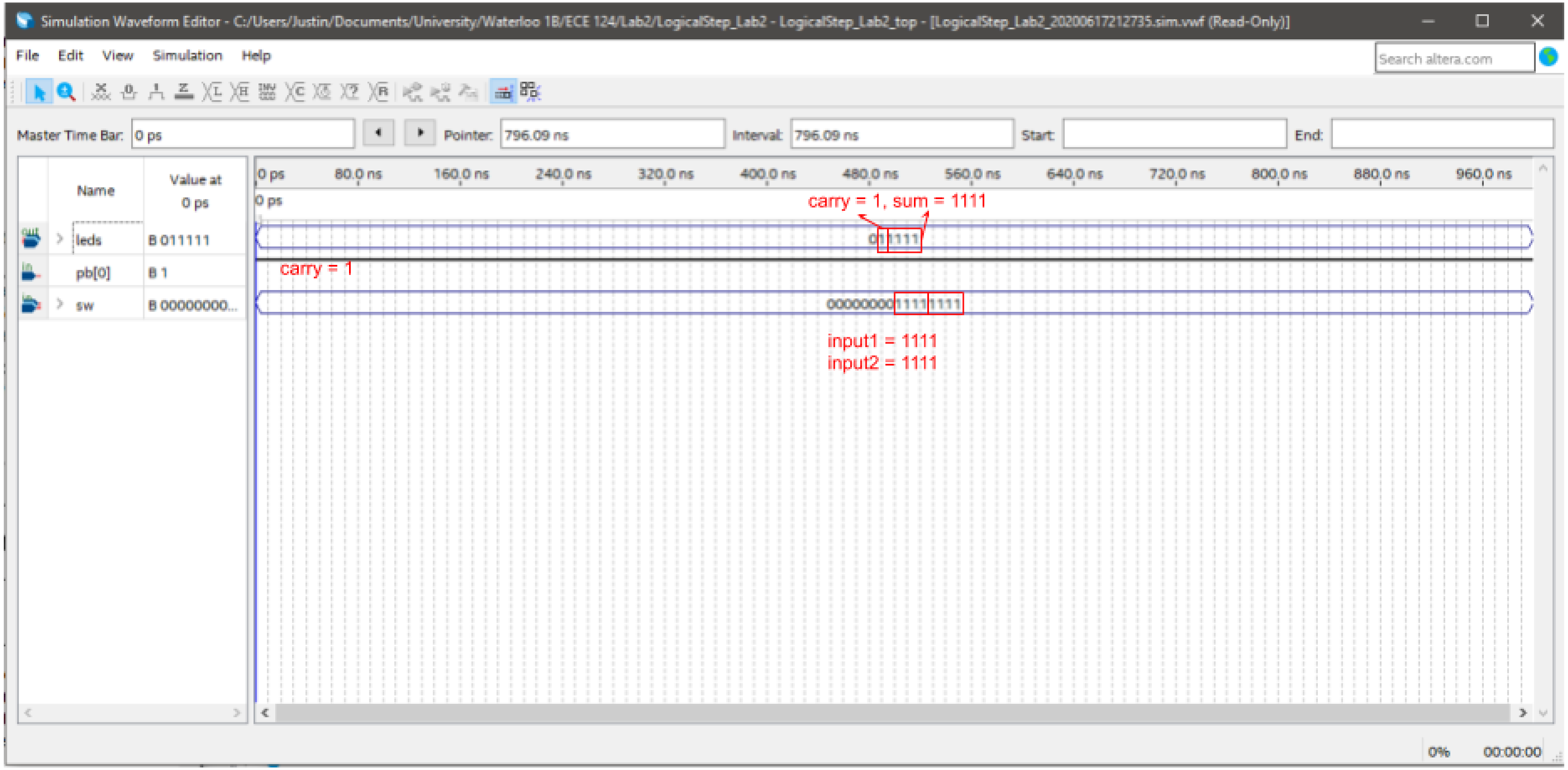
Start:

End:

Name	Value at 0 ps
leds	B 000000
pb	B 0000000
sw	B 00110010...

The pair of push buttons chose which 4-bit input to output
"00" : "0000"
"01" : "0001"
"10" : "0010"
"11" : "0011"

Full Adder Examples (High range, Mid range, Low Range)



Simulation Waveform Editor - C:/Users/Justin/Documents/University/Waterloo 1B/ECE 124/Lab2/LogicalStep_Lab2 - LogicalStep_Lab2_top - [LogicalStep_Lab2_20200617212735.sim.vwf (Read-Only)]

Master Time Bar: 0 ps

Pointer: 796.09 ns

Interval: 796.09 ns

Start:

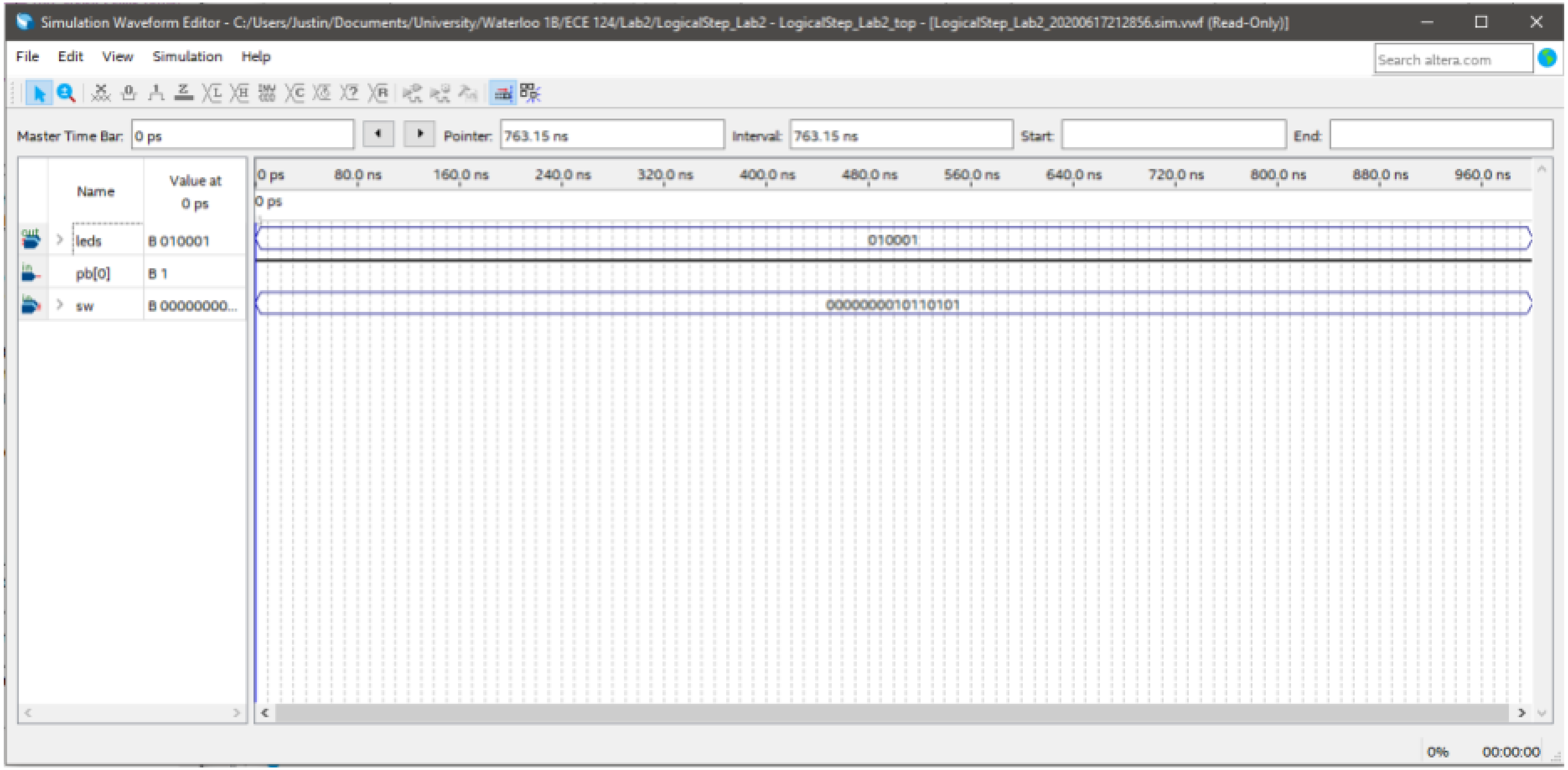
End:

Name	Value at 0 ps
leds	B 011111
pb[0]	B 1
sw	B 00000000...

carry = 1, sum = 1111

carry = 1

input1 = 1111
input2 = 1111



Simulation Waveform Editor - C:/Users/Justin/Documents/University/Waterloo 1B/ECE 124/Lab2/LogicalStep_Lab2 - LogicalStep_Lab2_top - [LogicalStep_Lab2_20200617212856.sim.vwf (Read-Only)]

Master Time Bar: 0 ps

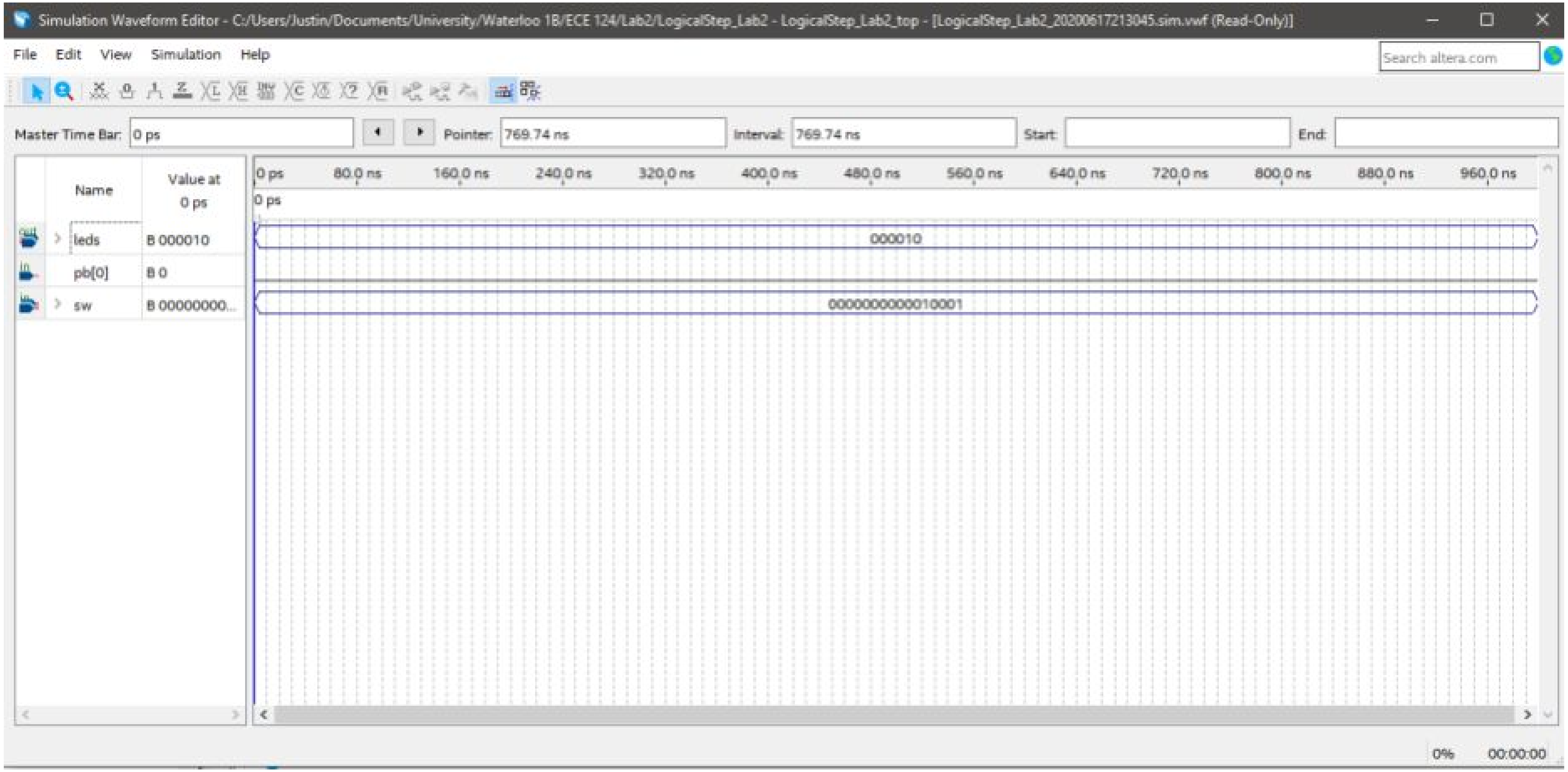
Pointer: 763.15 ns

Interval: 763.15 ns

Start:

End:

Name	Value at 0 ps
leds	B 010001
pb[0]	B 1
sw	B 00000000...



Simulation Waveform Editor - C:/Users/Justin/Documents/University/Waterloo 1B/ECE 124/Lab2/LogicalStep_Lab2 - LogicalStep_Lab2_top - [LogicalStep_Lab2_20200617213045.sim.vwf (Read-Only)]

Master Time Bar: 0 ps

Pointer: 769.74 ns

Interval: 769.74 ns

Start:

End:

Name	Value at 0 ps
leds	B 000010
pb[0]	B 0
sw	B 00000000...

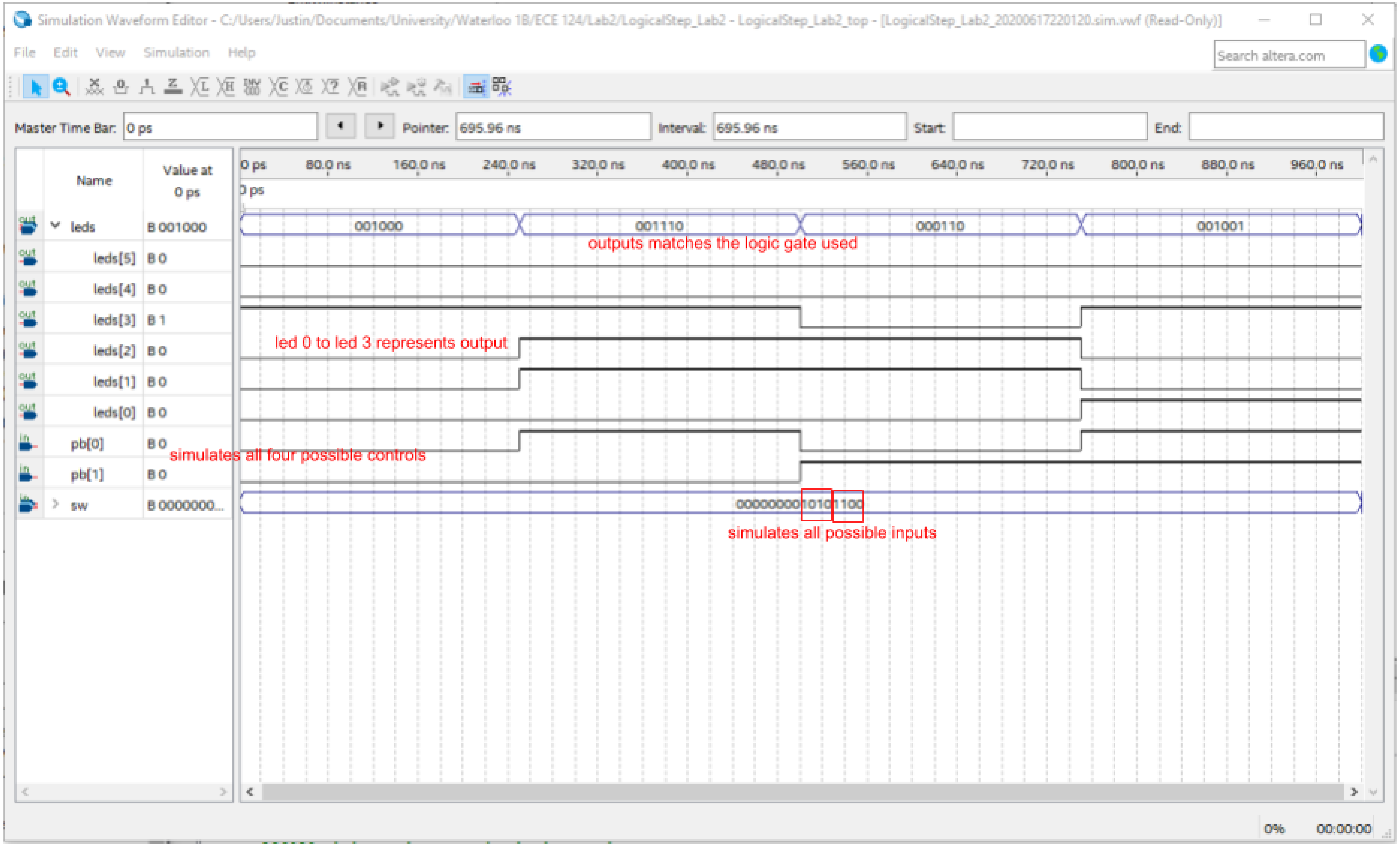
000010

0000000000010001

0%

00:00:00

Logical Processor Waveform



Simulation Waveform Editor - C:/Users/Justin/Documents/University/Waterloo 1B/ECE 124/Lab2/LogicalStep_Lab2 - LogicalStep_Lab2_top - [LogicalStep_Lab2_20200617220120.sim.vwf (Read-Only)]

Master Time Bar: 0 ps

Pointer: 695.96 ns

Interval: 695.96 ns

Start:

End:

Name	Value at 0 ps
leds	B 001000
leds[5]	B 0
leds[4]	B 0
leds[3]	B 1
leds[2]	B 0
leds[1]	B 0
leds[0]	B 0
pb[0]	B 0
pb[1]	B 0
sw	B 0000000...

outputs matches the logic gate used

led 0 to led 3 represents output

simulates all four possible controls

simulates all possible inputs