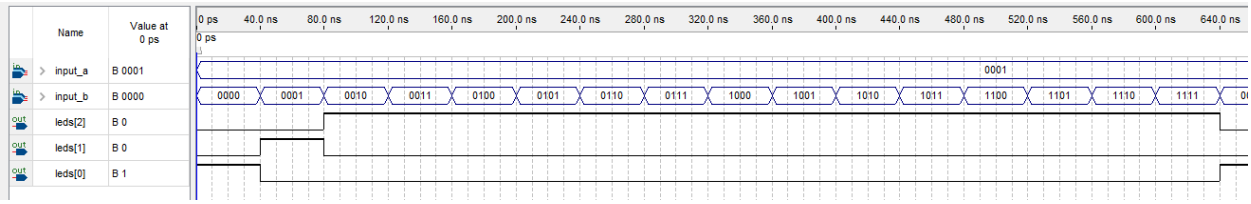
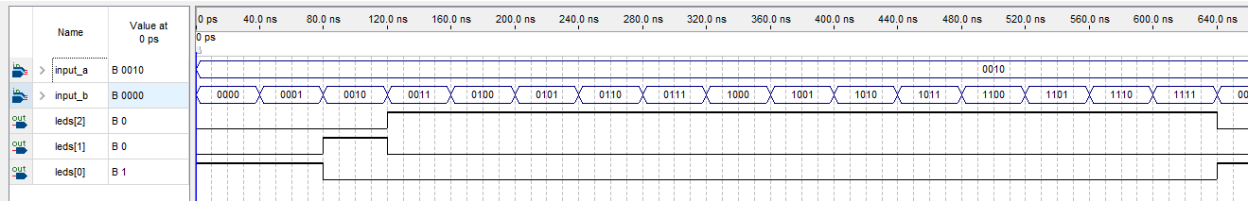
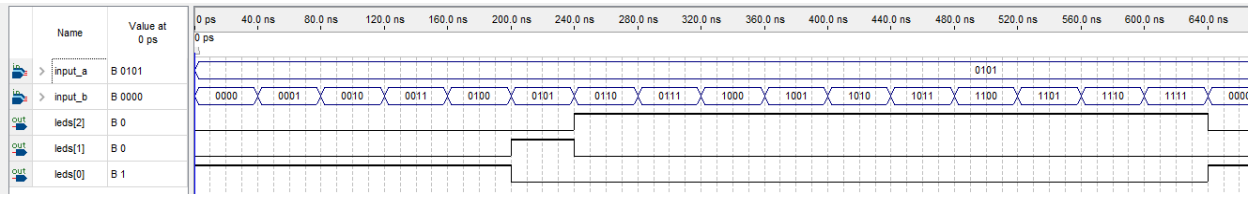
Part A - simulation 1



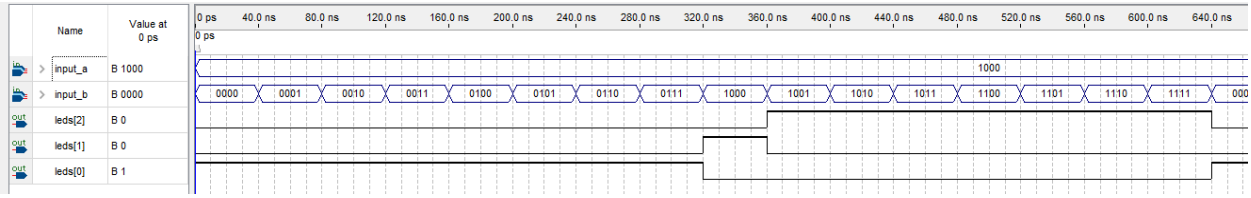
Part A - simulation 2



Part A - simulation 3



Part A - simulation 4



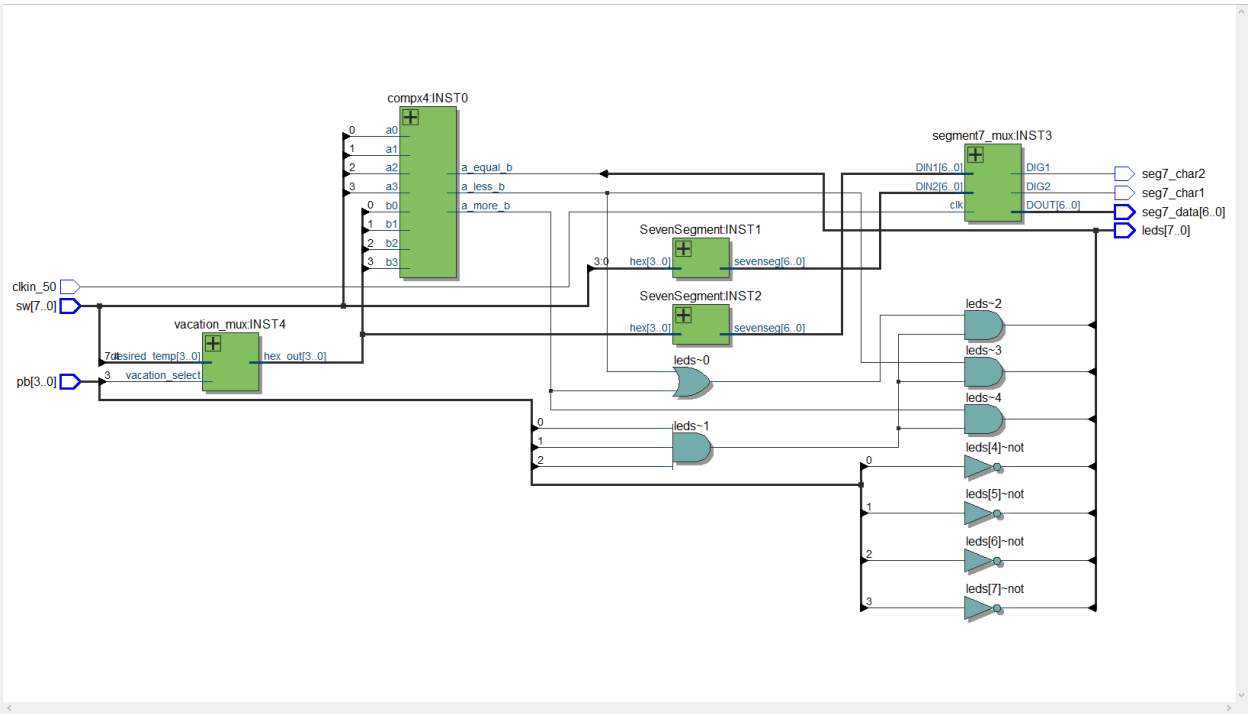
Truth Table for 4-bit Comparator

Comparison Inputs from 1-Bit Comparators												4-Bit Comparator Outputs		
A3<B3	A3=B3	A3>B3	A2<B2	A2=B2	A2>B2	A1<B1	A1=B1	A1>B1	A0<B0	A0=B0	A0>B0	A<B	A=B	A>B
0	0	1	X	X	X	X	X	X	X	X	X	0	0	1
1	0	0	X	X	X	X	X	X	X	X	X	1	0	0
0	1	0	0	0	1	X	X	X	X	X	X	0	0	1
0	1	0	1	0	0	X	X	X	X	X	X	1	0	0
0	1	0	0	0	0	0	0	1	X	X	X	0	0	1
0	1	0	0	0	0	1	0	0	X	X	X	1	0	0
0	1	0	0	0	0	0	0	0	0	0	1	0	0	1
0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Compilation Report

Flow Status	Successful - Sun Jun 18 19:01:32 2017
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Standard Edition
Revision Name	LogicalStep_Lab3_top
Top-level Entity Name	LogicalStep_Lab3_top
Family	MAX 10
Device	10M08SAE144C8G
Timing Models	Final
Total logic elements	49 / 8,064 (< 1 %)
Total combinational functions	49 / 8,064 (< 1 %)
Dedicated logic registers	11 / 8,064 (< 1 %)
Total registers	11
Total pins	30 / 101 (30 %)
Total virtual pins	0
Total memory bits	0 / 387,072 (0 %)
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 1 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 1 (0 %)

RTL Viewer



Compx1

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

entity compx1 is
port (
    bit_A, bit_B                    : in std_logic; --one-bit input (A, B)
    A_less_B, A_equal_B, A_more_B   : out std_logic --one-bit ouptut
);

end entity compx1;

architecture compx1 of compx1 is

begin

-- for the multiplexing of two hex input busses
A_less_B <= NOT bit_A AND bit_B; --when A is 0 and B is 1
A_equal_B <= (bit_A AND bit_B) OR (NOT(bit_A) AND NOT(bit_B)); --when A is 1 and B is 1 or when A is 0 and B is 0
A_more_B <= bit_A AND NOT bit_B; --when A is 1 and B is 0

end compx1;
```

LogicalStep_Lab3_top

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

entity LogicalStep_Lab3_top is port (
    clkin_50    : in  std_logic;
    pb          : in  std_logic_vector(3 downto 0); -- pb(2) fdoor; pb(1) window; pb(0) bdoor
    sw          : in  std_logic_vector(7 downto 0); -- The switch inputs
    leds        : out std_logic_vector(7 downto 0); -- for displaying the switch content
    seg7_data   : out std_logic_vector(6 downto 0); -- 7-bit outputs to a 7-segment
    seg7_char1  : out std_logic;                    -- seg7 digi selectors
    seg7_char2  : out std_logic                     -- seg7 digi selectors

);
end LogicalStep_Lab3_top;
```

```vhdl
architecture Energy_Monitor of LogicalStep_Lab3_top is
-- Components Used
---------------------------------------------------------------------
  component compx4 port (
    a0, b0, a1, b1, a2, b2, a3, b3   : in std_logic; --a is the current temp, b is the desired temp
    a_more_b, a_equal_b, a_less_b    : out std_logic --one-bit output based on 4 compx1
);
    end component;

    component segment7_mux port (
     clk        :  in std_logic := '0';
     DIN2       :  in std_logic_vector(6 downto 0);
     DIN1       :  in std_logic_vector(6 downto 0);
     DOUT       :  out std_logic_vector(6 downto 0);
     DIG2       :  out std_logic;
     DIG1       :  out std_logic
     );
    end component;

  component SevenSegment port (
    hex         :  in  std_logic_vector(3 downto 0);   -- The 4 bit data to be displayed
    sevenseg    :  out std_logic_vector(6 downto 0)    -- 7-bit outputs to a 7-segment
);
    end component;

  component vacation_mux port (
    desired_temp        : in std_logic_vector(3 downto 0);
    vacation_select     : in std_logic;
    hex_out             : out std_logic_vector(3 downto 0)
);
    end component;
---------------------------------------------------------------------

-- Create any signals, or temporary variables to be used
signal temp_led0     : std_logic; -- a more b i.e. current more desired; truth value for led0 (furnace on, below temp)
signal temp_led1     : std_logic; -- truth value for led1 (at temp)
signal temp_led2     : std_logic; -- a more b i.e. current more desired; truth value for led2 (ac on, below temp)
signal seg7_A        : std_logic_vector(6 downto 0); -- left display
signal seg7_B        : std_logic_vector(6 downto 0); -- right display
signal desired       : std_logic_vector(3 downto 0); -- output of vacation mux

-- Here the circuit begins

begin
leds(3) <= (temp_led0 OR temp_led2) AND (pb(2) AND pb(1) AND pb(0));  --blower status
leds(2) <= temp_led0 AND (pb(2) AND pb(1) AND pb(0));                  -- furnace status
leds(1) <= temp_led1;                                                 --desired temp status
leds(0) <= temp_led2 AND (pb(2) AND pb(1) AND pb(0));                 -- ac status

leds(4) <= NOT(pb(0)); -- flip switches for led values
leds(5) <= NOT(pb(1));
leds(6) <= NOT(pb(2));
leds(7) <= NOT(pb(3));

INST0: compx4 port map(sw(0), desired(0), sw(1), desired(1), sw(2), desired(2), sw(3), desired(3), temp_led2, temp_led1, temp_led0);
INST1: SevenSegment port map(sw(3 downto 0), seg7_A);
INST2: SevenSegment port map(desired, seg7_B);
INST3: segment7_mux port map(clkin_50, seg7_A, seg7_B, seg7_data, seg7_char1, seg7_char2);
INST4: vacation_mux port map(sw(7 downto 4), pb(3), desired); -- mux to select between '0100' and switches

end Energy_Monitor;
```

Compx4

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity compx4 is
5   port (
6       a0, b0, a1, b1, a2, b2, a3, b3   : in std_logic; --one-bit input (A, B)
7       a_more_b, a_equal_b, a_less_b    : out std_logic --one-bit output based on 4 compx1
8   );
9
10  end entity compx4;
11
12  architecture compx4 of compx4 is
13
14  -- Components Used --
15    component compx1 port (
16      bit_A, bit_B                     : in std_logic; --one-bit input (A, B)
17      A_less_B, A_equal_B, A_more_B    : out std_logic --one-bit ouptut
18    );
19      end component;
20
21  -- Signals Created --
22  signal a0_more_b0 : std_logic;
23  signal a1_more_b1 : std_logic;
24  signal a2_more_b2 : std_logic;
25  signal a3_more_b3 : std_logic;
26  signal a0_equal_b0 : std_logic;
27  signal a1_equal_b1 : std_logic;
28  signal a2_equal_b2 : std_logic;
29  signal a3_equal_b3 : std_logic;
30  signal a0_less_b0 : std_logic;
31  signal a1_less_b1 : std_logic;
32  signal a2_less_b2 : std_logic;
33  signal a3_less_b3 : std_logic;
34
35
36  -- Circuit Begins --
37  begin
38
39  INST0: compx1 port map(a0, b0, a0_more_b0, a0_equal_b0, a0_less_b0);
40  INST1: compx1 port map(a1, b1, a1_more_b1, a1_equal_b1, a1_less_b1);
41  INST2: compx1 port map(a2, b2, a2_more_b2, a2_equal_b2, a2_less_b2);
42  INST3: compx1 port map(a3, b3, a3_more_b3, a3_equal_b3, a3_less_b3);
43
44
45  a_equal_b <= a0_equal_b0 AND a1_equal_b1 AND a2_equal_b2 AND a3_equal_b3;
46  a_more_b <= (a3_more_b3) OR
47              (a3_equal_b3 AND a2_more_b2) OR
48              (a3_equal_b3 AND a2_equal_b2 AND a1_more_b1) OR
49              (a3_equal_b3 AND a2_equal_b2 AND a1_equal_b1 AND a0_more_b0);
50  a_less_b <= (a3_less_b3) OR
51              (a3_equal_b3 AND a2_less_b2) OR
52              (a3_equal_b3 AND a2_equal_b2 AND a1_less_b1) OR
53              (a3_equal_b3 AND a2_equal_b2 AND a1_equal_b1 AND a0_less_b0);
54  end compx4;
```

vacation_mux

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity vacation_mux is
5   port (
6      desired_temp         : in std_logic_vector(3 downto 0);
7      vacation_select      : in std_logic;
8      hex_out              : out std_logic_vector(3 downto 0)
9   );
10
11  end entity vacation_mux;
12
13  architecture mux_logic of vacation_mux is
14
15  begin
16
17  with vacation_select select
18  hex_out <= "0100" when '0', --when pb3 is not pressed
19             desired_temp when '1'; --when pb3 is pressed
20  end mux_logic;
```