**1. Main File:**

LAB 3 GRP 7 SESS 201 REPORT

**LogicalStep\_Lab3\_ top.vhd file:**

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);

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;

architecture Energy\_Monitor of LogicalStep\_Lab3\_top is

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

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

-- COMPONENTS USED

--

-- COMPONENT 1 (SevenSegment)

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 2 (1-Bit Comparator)

component Compx1 port (

A : in std\_logic\_vector; -- The 1 bit data to be compared

B : in std\_logic\_vector; -- The 1 bit data to be compared

AGTB : out std\_logic\_vector; -- 1-bit output showing if A > B

AEQB : out std\_logic\_vector; -- 1-bit output showing if A = B

ALTB : out std\_logic\_vector -- 1-bit output showing if A < B

);

end component;

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

-- COMPONENT 3 (4-Bit Comparator)

component Compx4 port (

A3GTB3 : in std\_logic\_vector;

A3EQB3 : in std\_logic\_vector;

A3LTB3 : in std\_logic\_vector;

A2GTB2 : in std\_logic\_vector;

A2EQB2 : in std\_logic\_vector;

A2LTB2 : in std\_logic\_vector;

A1GTB1 : in std\_logic\_vector;

A1EQB1 : in std\_logic\_vector;

A1LTB1 : in std\_logic\_vector;

A0GTB0 : in std\_logic\_vector;

A0EQB0 : in std\_logic\_vector;

A0LTB0 : in std\_logic\_vector;

AGTB : out std\_logic\_vector; -- 1-bit output showing if A > B

AEQB : out std\_logic\_vector; -- 1-bit output showing if A = B

ALTB : out std\_logic\_vector -- 1-bit output showing if A < B

);

end component;

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

-- COMPONENT 4 (7-Bit Digit Mux)

component segment7\_mux port (

clk : in std\_logic := '0';

DIN2 : in std\_logic\_vector(6 downto 0); -- outputs to the Seven Segment LED board

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;

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

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

-- SIGNALS USED

--

signal A : std\_logic\_vector (3 downto 0);

signal B : std\_logic\_vector (3 downto 0);

signal seg7\_A : std\_logic\_vector(6 downto 0);

signal seg7\_B : std\_logic\_vector(6 downto 0);

signal A0 : std\_logic\_vector (0 downto 0);

signal B0 : std\_logic\_vector (0 downto 0);

signal A1 : std\_logic\_vector (0 downto 0);

signal B1 : std\_logic\_vector (0 downto 0);

signal A2 : std\_logic\_vector (0 downto 0);

signal B2 : std\_logic\_vector (0 downto 0);

signal A3 : std\_logic\_vector (0 downto 0);

signal B3 : std\_logic\_vector (0 downto 0);

signal A3GTB3 : std\_logic\_vector (0 downto 0);

signal A3EQB3 : std\_logic\_vector (0 downto 0);

signal A3LTB3 : std\_logic\_vector (0 downto 0);

signal A2GTB2 : std\_logic\_vector (0 downto 0);

signal A2EQB2 : std\_logic\_vector (0 downto 0);

signal A2LTB2 : std\_logic\_vector (0 downto 0);

signal A1GTB1 : std\_logic\_vector (0 downto 0);

signal A1EQB1 : std\_logic\_vector (0 downto 0);

signal A1LTB1 : std\_logic\_vector (0 downto 0);

signal A0GTB0 : std\_logic\_vector (0 downto 0);

signal A0EQB0 : std\_logic\_vector (0 downto 0);

signal A0LTB0 : std\_logic\_vector (0 downto 0);

signal AGTB : std\_logic\_vector (0 downto 0);

signal AEQB : std\_logic\_vector (0 downto 0);

signal ALTB : std\_logic\_vector (0 downto 0);

signal Fdoor : std\_logic\_vector (0 downto 0);

signal window : std\_logic\_vector (0 downto 0);

signal Bdoor : std\_logic\_vector (0 downto 0);

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

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

begin

A <= sw(7 downto 4); -- DESIRED TEMPERATURE => A

B <= sw(3 downto 0); -- CURRENT TEMPERATURE => B

Fdoor <= pb(2 downto 2); -- Front Door

window <= pb(1 downto 1); -- Windows

Bdoor <= pb(0 downto 0); -- Back Door

A0 <= sw(4 downto 4);

A1 <= sw(5 downto 5);

A2 <= sw(6 downto 6);

A3 <= sw(7 downto 7);

B0 <= sw(0 downto 0);

B1 <= sw(1 downto 1);

B2 <= sw(2 downto 2);

B3 <= sw(3 downto 3);

leds (0 downto 0) <= ALTB AND Fdoor AND Bdoor AND window;

leds (1 downto 1) <= AEQB;

leds (2 downto 2) <= AGTB AND Fdoor AND Bdoor AND window;

leds (3 downto 3) <= (AGTB OR ALTB) AND Fdoor AND Bdoor AND window;

leds (4 downto 4) <= NOT(Bdoor);

leds (5 downto 5) <= NOT(window);

leds (6 downto 6) <= NOT(Fdoor);

INST1: SevenSegment port map(A, seg7\_A);

INST2: SevenSegment port map(B, seg7\_B);

INST3: segment7\_mux port map(clkin\_50, seg7\_A, seg7\_B, seg7\_data, seg7\_char1, seg7\_char2);

INST4: Compx1 port map(A0, B0, A0GTB0, A0EQB0, A0LTB0);

INST5: Compx1 port map(A1, B1, A1GTB1, A1EQB1, A1LTB1);

INST6: Compx1 port map(A2, B2, A2GTB2, A2EQB2, A2LTB2);

INST7: Compx1 port map(A3, B3, A3GTB3, A3EQB3, A3LTB3);

INST8: Compx4 port map(A3GTB3, A3EQB3, A3LTB3, A2GTB2, A2EQB2, A2LTB2, A1GTB1, A1EQB1, A1LTB1, A0GTB0, A0EQB0, A0LTB0, AGTB, AEQB, ALTB);

end Energy\_Monitor;

**2. Sub Files:**

**Compx1.vhd file :**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity Compx1 is port (

A : in std\_logic\_vector;

B : in std\_logic\_vector;

AGTB : out std\_logic\_vector;

AEQB : out std\_logic\_vector;

ALTB : out std\_logic\_vector

);

end Compx1;

architecture Comp1 of Compx1 is

begin

AGTB <= A AND NOT(B);

AEQB <= A XNOR B; --(NOT(A) AND NOT(B)) OR (A AND B)

ALTB <= NOT(A) AND B;

end Comp1;

**Compx4.vhd file :**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity Compx4 is port (

A3GTB3 : in std\_logic\_vector;

A3EQB3 : in std\_logic\_vector;

A3LTB3 : in std\_logic\_vector;

A2GTB2 : in std\_logic\_vector;

A2EQB2 : in std\_logic\_vector;

A2LTB2 : in std\_logic\_vector;

A1GTB1 : in std\_logic\_vector;

A1EQB1 : in std\_logic\_vector;

A1LTB1 : in std\_logic\_vector;

A0GTB0 : in std\_logic\_vector;

A0EQB0 : in std\_logic\_vector;

A0LTB0 : in std\_logic\_vector;

AGTB : out std\_logic\_vector;

AEQB : out std\_logic\_vector;

ALTB : out std\_logic\_vector

);

end Compx4;

architecture Comp4 of Compx4 is

begin

AGTB <= A3GTB3 OR (A3EQB3 AND (A2GTB2 OR (A2EQB2 AND (A1GTB1 OR (A1EQB1 AND A0GTB0)))));

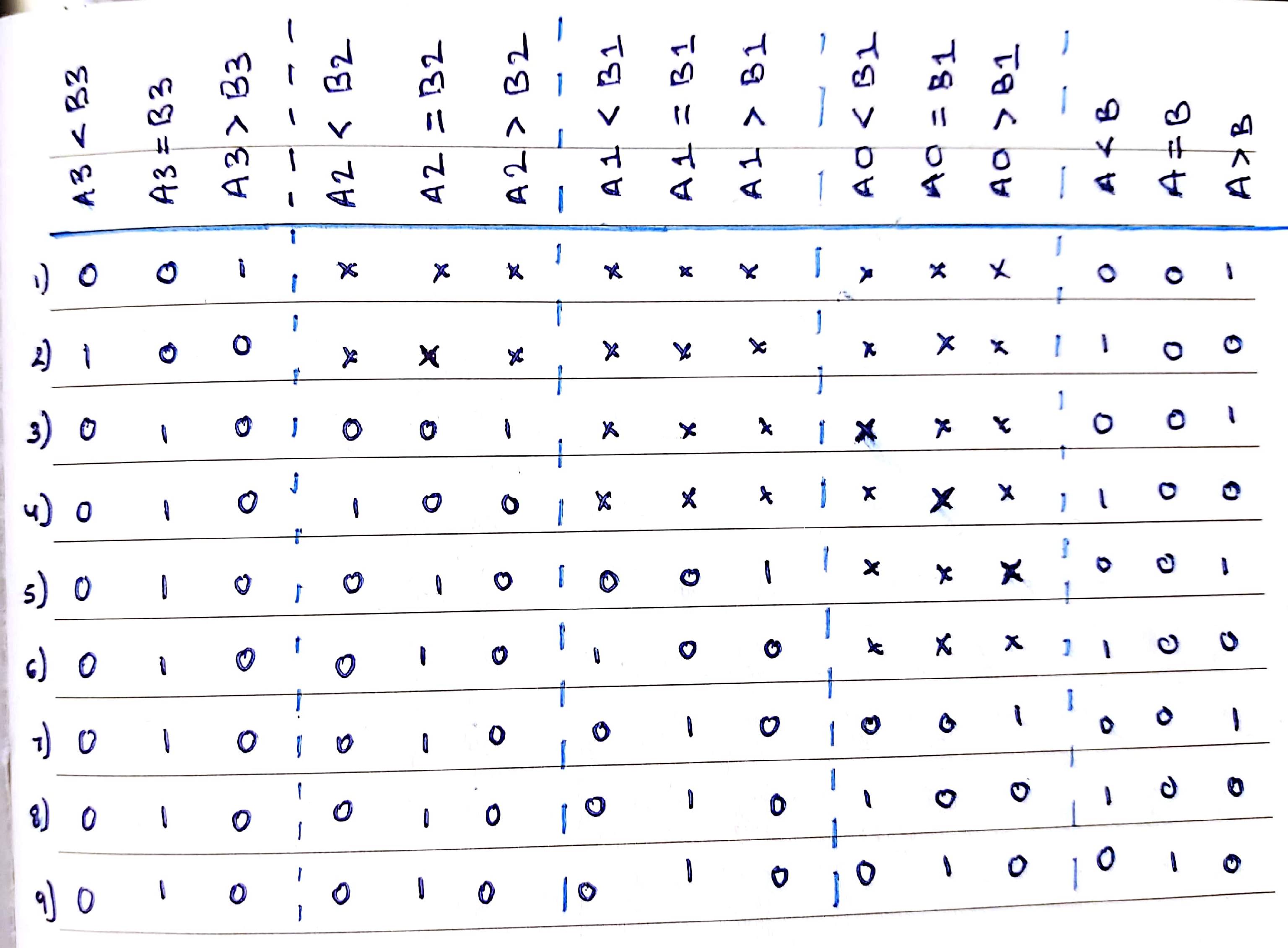
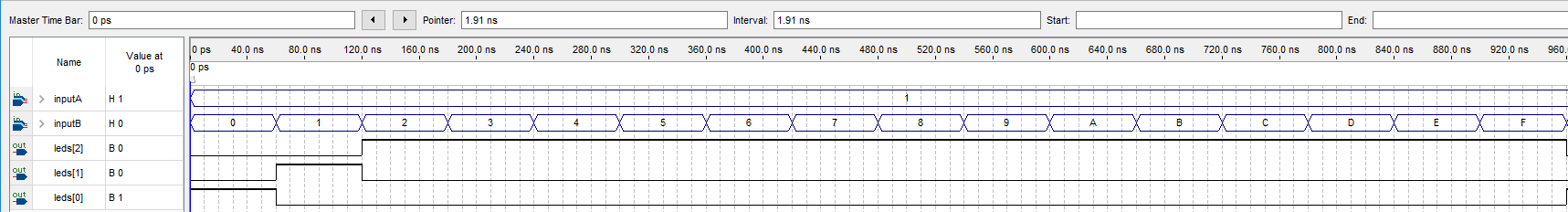
AEQB <= A3EQB3 AND A2EQB2 AND A1EQB1 AND A0EQB0;

ALTB <= A3LTB3 OR (A3EQB3 AND (A2LTB2 OR (A2EQB2 AND (A1LTB1 OR (A1EQB1 AND A0LTB0)))));

end Comp4;

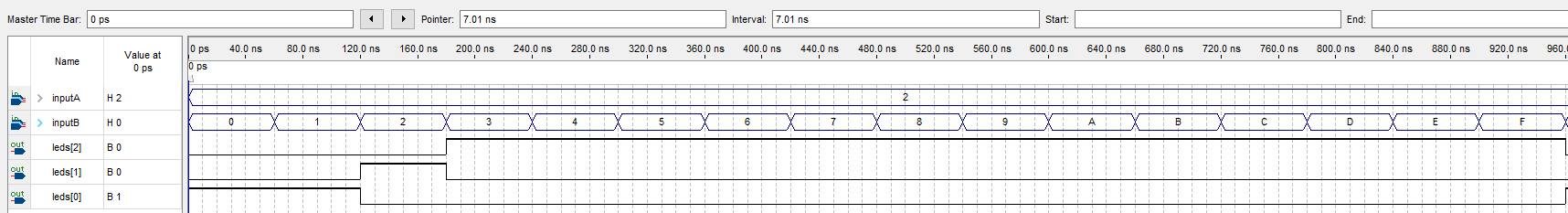
Simulation 1: Input\_A = hex 1

**Simulations of Comparator:**

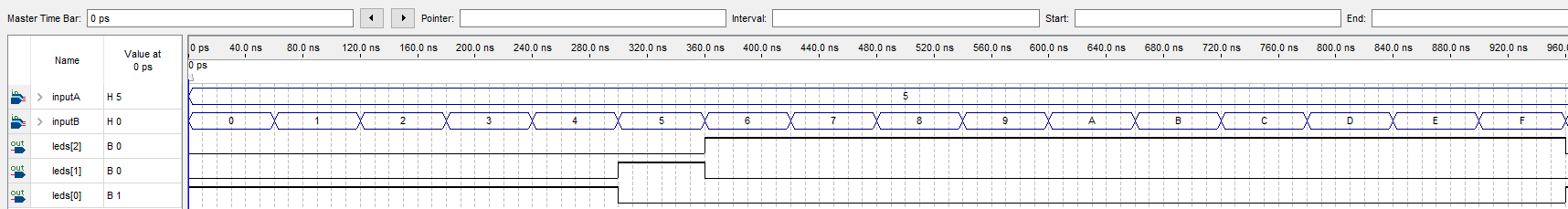


**Truth Table for 4-Bit Comparator**

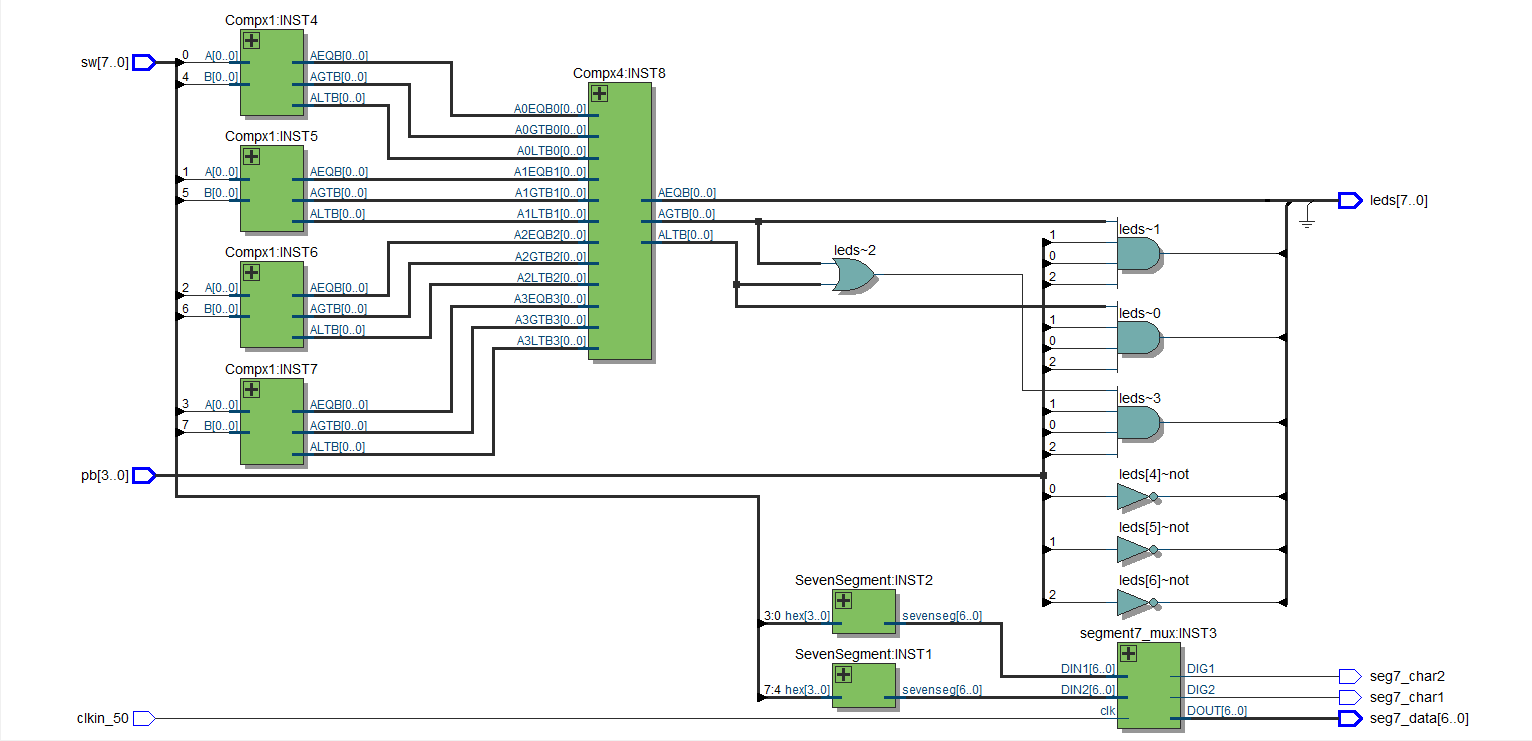
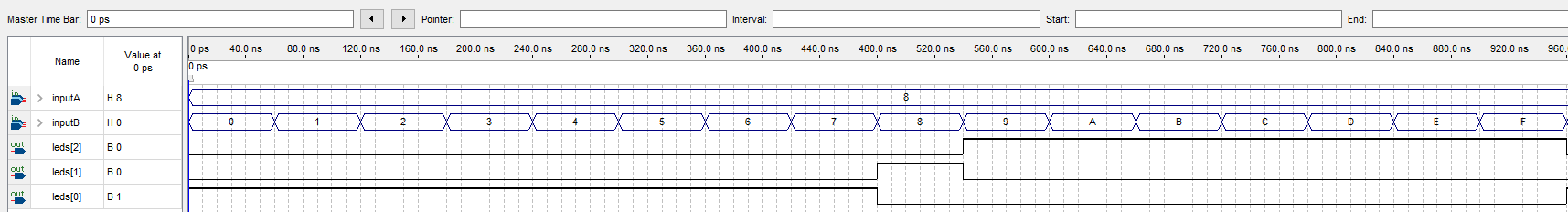
Simulation 2: Input\_A = hex 2



Simulation 3: Input\_A = hex 5

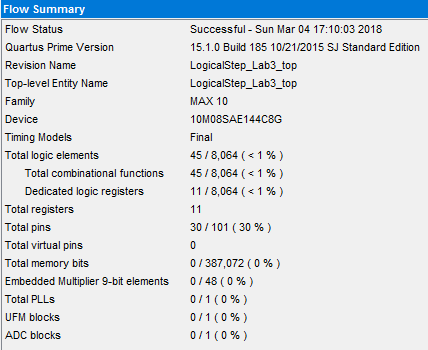


Simulation 4: Input\_A = hex 8



**RTL View of the Logic Design:**

**Total Design Logic Elements Used:**



Total Design Logic Elements Used = 45