# LAB3 GRP7 SESS202 REPORT

## VHDL SOURCE CODE 1: TOP LEVEL STRUCTURAL

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 fourbitcomparator port(

bitA0, bitA1, bitA2, bitA3, bitB0, bitB1, bitB2, bitB3 : in std\_logic;

AGTB : out std\_logic;

AEQB : out std\_logic;

ALTB : out std\_logic

);

end component;

component SevenSegment port ( -- converts a 4-bit number to 7 bit-number to be use for the 7-segment display

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 segment7\_mux port ( -- the mux to turn the 2 7-segment display on simultaneously

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 EnergyControl is port (

CurrentGreater, CurrentEqual, CurrentSmaller : in std\_logic;

DoorWindowOpenControl : in std\_logic\_vector(2 downto 0); -- our push buttons

-- remember it is active low, so 0 means door open

--Fdoor->pb2 window->pb1 Bdoor ->pb0

--Fdoor-> led6 window->led5, Bdoor -> led4

FurnaceON: out std\_logic; -- 1 if on

SystemAtTemp: out std\_logic;-- 1 if on

ACON: out std\_logic;-- 1 if on

BlowerON: out std\_logic;-- 1 if on

DoorWindowOpenOutput: out std\_logic\_vector(2 downto 0)-- 1 if on

);

end component;

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

signal Current\_Temp : std\_logic\_vector(3 downto 0); -- represents the current Temp

signal Desired\_Temp : std\_logic\_vector(3 downto 0); -- represents the desired temp

signal GT, EQ, LT : std\_logic;

signal sevenseg\_Current\_Temp, sevenseg\_Desired\_Temp : std\_logic\_vector(6 downto 0); -- the signal that goes directly into the sevenseg mux

begin

----- input from our switches for temperature -----------------

Current\_Temp <= sw(7 downto 4); -- left side input

Desired\_Temp <= sw(3 downto 0); -- right side input

---------------- comparing Current\_Temp to Desired\_Temp -----------------------------

INST1: fourbitcomparator port map(Current\_Temp(0),Current\_Temp(1),Current\_Temp(2),Current\_Temp(3),

Desired\_Temp(0),Desired\_Temp(1),Desired\_Temp(2),Desired\_Temp(3),

GT,EQ,LT); -- a bit inefficient in terms of amount if input, but does the job

-- could be cleaner

------------------DISPLAYING THE TEMPS--------------------

INST2: SevenSegment port map(Current\_Temp, sevenseg\_Current\_Temp);

INST3: sevenSegment port map(Desired\_Temp, sevenseg\_Desired\_Temp);

INST4: segment7\_mux port map (clkin\_50 , sevenseg\_Current\_Temp, sevenseg\_Desired\_Temp, seg7\_data, seg7\_char1, seg7\_char2);

------------------END OF DISPLAYING-----------------------

------------------CONTROLLING THE outputs using our Energy Control component------------------------------

INST5: EnergyControl port map(GT,EQ,LT, pb(2 downto 0), leds(0), leds(1), leds(2), leds(3), leds(6 downto 4) );

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

end Energy\_Monitor;

## VHDL SOURCE CODE 2: 1-BIT COMPARATOR

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity SingleBitComparator is port (

bit1 : in std\_logic;

bit2 : in std\_logic;

gt : out std\_logic;

et : out std\_logic;

lt : out std\_logic

);

end SingleBitComparator;

architecture Behavioural of SingleBitComparator is

Begin

gt <= bit1 and not bit2; --bit1 is 1 and bit2 is 0

et <= NOT (bit1 XOR bit2); --when bit1 and bit2 are the same values

lt <= not bit1 and bit2; --bit1 is 0 and bit2 is 1

end Behavioural;

## VHDL SOURCE CODE 3: 4-BITS COMPARATOR

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity FourBitComparator is port (

bitA0, bitA1, bitA2, bitA3, bitB0, bitB1, bitB2, bitB3 : in std\_logic;

AGTB : out std\_logic;

AEQB : out std\_logic;

ALTB : out std\_logic

);

end FourBitComparator;

architecture Behavioural of FOurBitComparator is

component SingleBitComparator port(

bit1 : in std\_logic;

bit2 : in std\_logic;

gt : out std\_logic;

et : out std\_logic;

lt : out std\_logic

);

end component;

signal A3LTB3, A3GTB3, A3EQB3, A2LTB2, A2GTB2, A2EQB2,A1LTB1, A1GTB1, A1EQB1, A0LTB0, A0GTB0, A0EQB0 : std\_logic;

signal ALTB\_signal , AEQB\_signal , AGTB\_signal : std\_logic;

Begin

INST1: SingleBitComparator port map(bitA0, bitB0, A0GTB0,A0EQB0, A0LTB0);

INST2: SingleBitComparator port map(bitA1, bitB1, A1GTB1,A1EQB1, A1LTB1);

INST3: SingleBitComparator port map(bitA2, bitB2, A2GTB2,A2EQB2, A2LTB2);

INST4: SingleBitComparator port map(bitA3, bitB3, A3GTB3,A3EQB3, A3LTB3);

-- equal if all 4 bits compare to equal

AEQB\_signal <= A3EQB3 and A2EQB2 and A1EQB1 and A0EQB0;

-- greater if all bits compared early are equal and this bit comparing is bigger

AGTB\_signal <= (A3GTB3) or (A3EQB3 and A2GTB2) or (A3EQB3 and A2EQB2 AND A1GTB1) OR (A3EQB3 AND A2EQB2 AND A1EQB1 AND A0GTB0);

--else....

ALTB <= not(AEQB\_signal OR AGTB\_signal);

-- just setting the other signal to the real output

AEQB <= AEQB\_signal;

AGTB <= AGTB\_signal;

end Behavioural;

## VHDL SOURCE CODE 4: ENERGY CONTROL

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity EnergyControl is port (

CurrentGreater, CurrentEqual, CurrentSmaller : in std\_logic;

DoorWindowOpenControl : in std\_logic\_vector(2 downto 0); -- our push buttons

-- remember it is active low, so 0 means door open

FurnaceON: out std\_logic; -- 1 if on

SystemAtTemp: out std\_logic;-- 1 if on

ACON: out std\_logic;-- 1 if on

BlowerON: out std\_logic;-- 1 if on

DoorWindowOpenOutput: out std\_logic\_vector(2 downto 0)-- 1 if on

);

end EnergyControl;

architecture Behaviour of EnergyControl is

signal isolatedSystem: std\_logic; --- 1 if it is isolated

begin

-- hooking up the PB to LEDS to show if anything is open

DoorWindowOpenOutput(0) <= not DoorWindowOpenControl(0);

DoorWindowOpenOutput(1) <= not DoorWindowOpenControl(1);

DoorWindowOpenOutput(2) <= not DoorWindowOpenControl(2);

-- check if everything is shut

isolatedSystem <= DoorWindowOpenControl(0) AND DoorWindowOpenControl(1) AND DoorWindowOpenControl(2);

-- check if blower should be on

BlowerON <= (CurrentGreater or CurrentSmaller) AND (isolatedSystem);

-- check if AC should be on

ACON <= (CurrentGreater)AND (isolatedSystem);

-- check if Furnace should be on

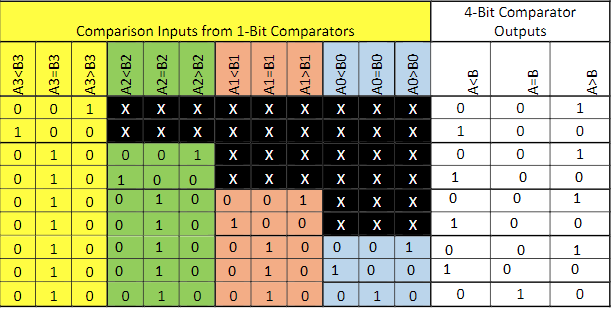
FurnaceON <= (CurrentSmaller) AND (isolatedSystem);

--check if current temperature is equal

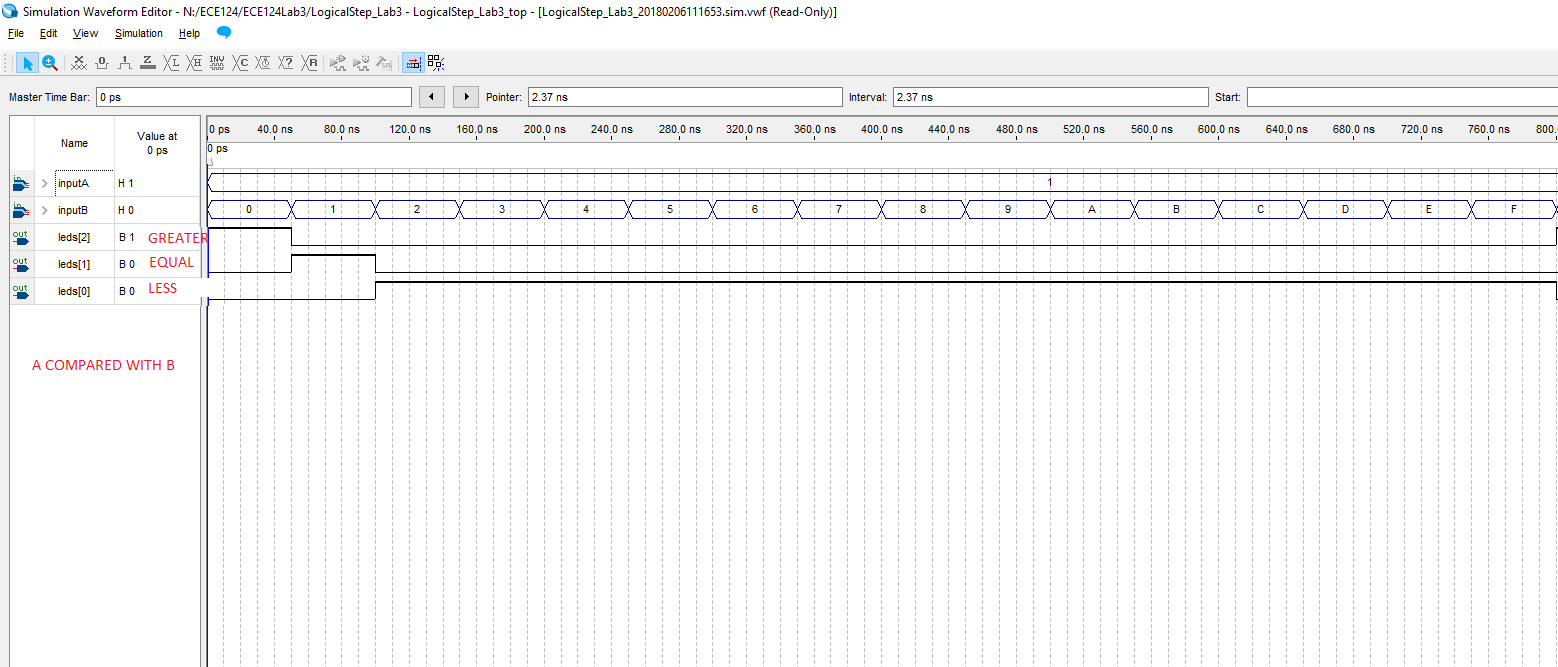
SystemAtTemp <= (CurrentEqual);

end Behaviour;

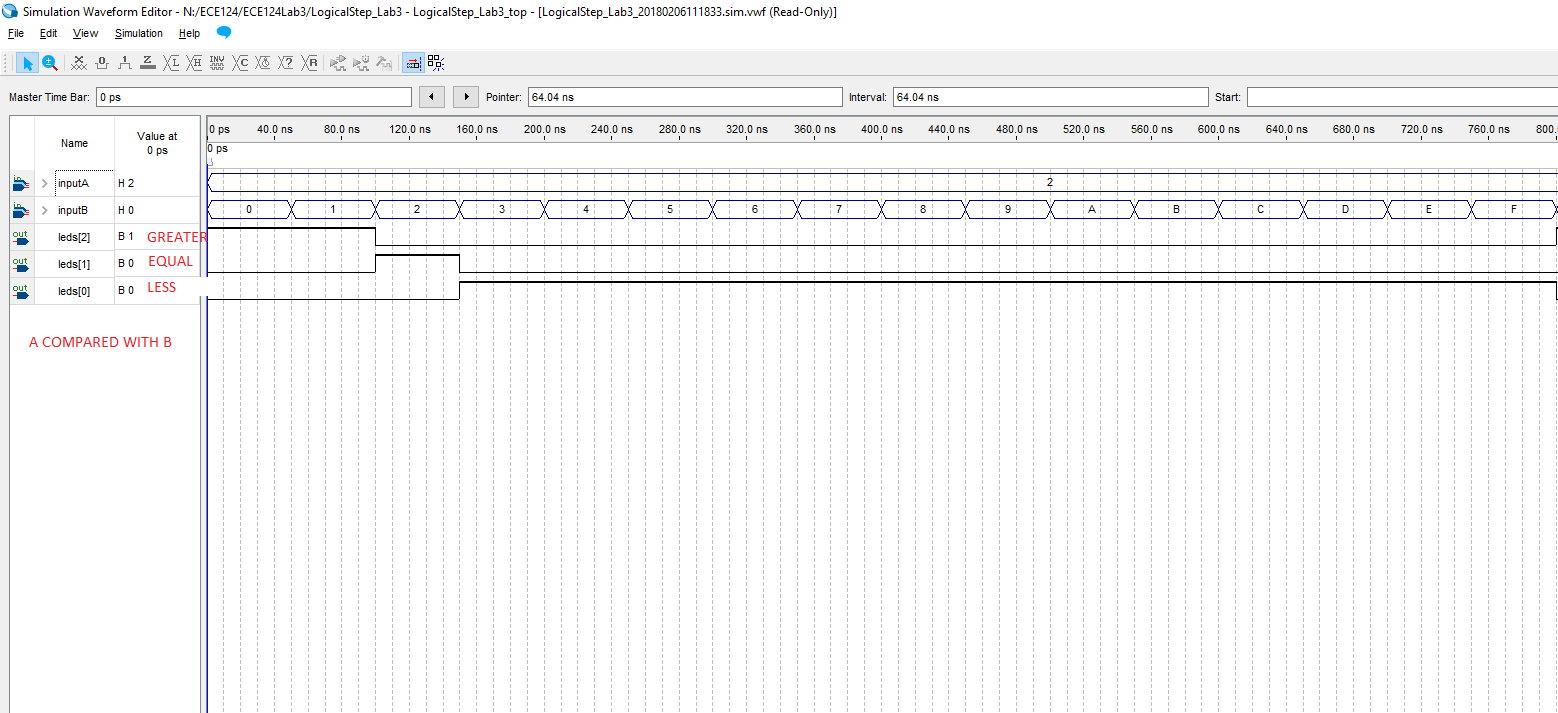
## TABLE1: COMPARATOR TRUTH TABLE



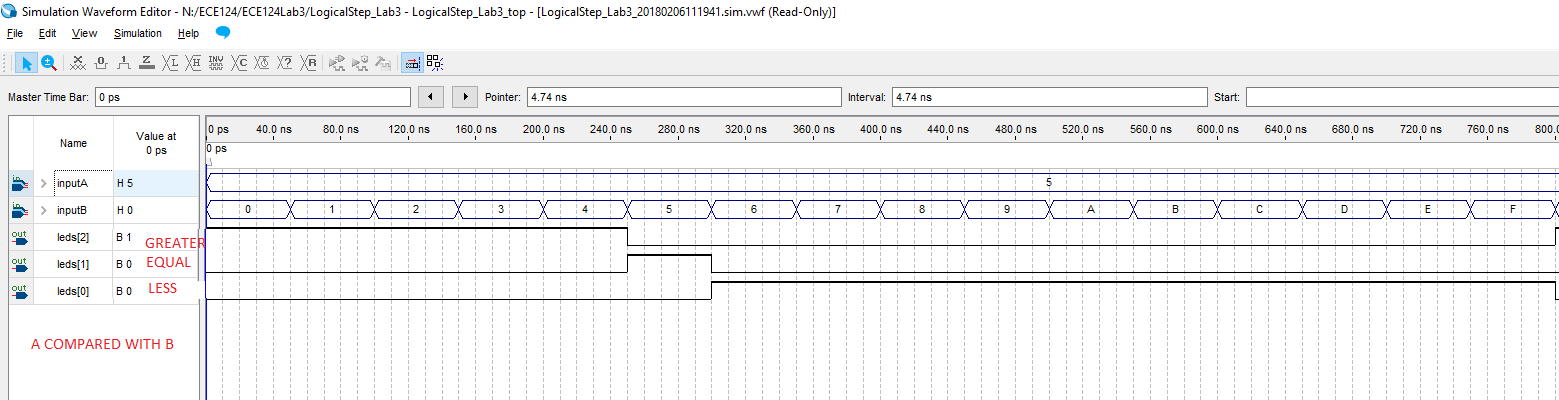
## GRAPH1: SIMULATION1 where HEXA = 1



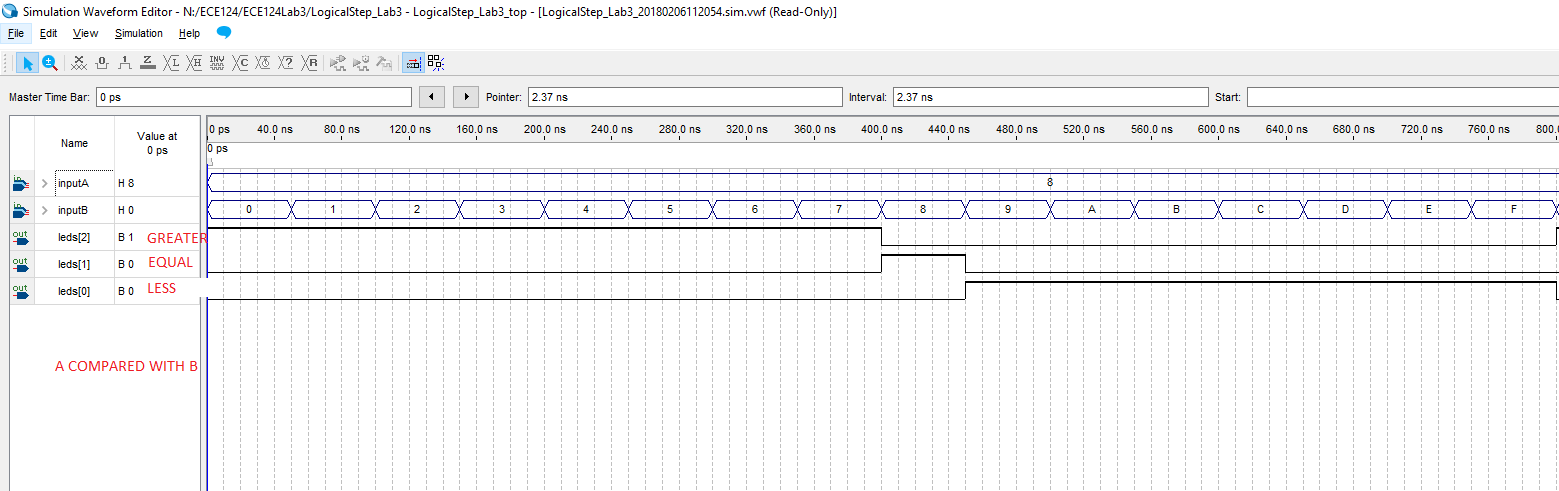
## GRAPH2: SIMULATION2 where HEXA = 2



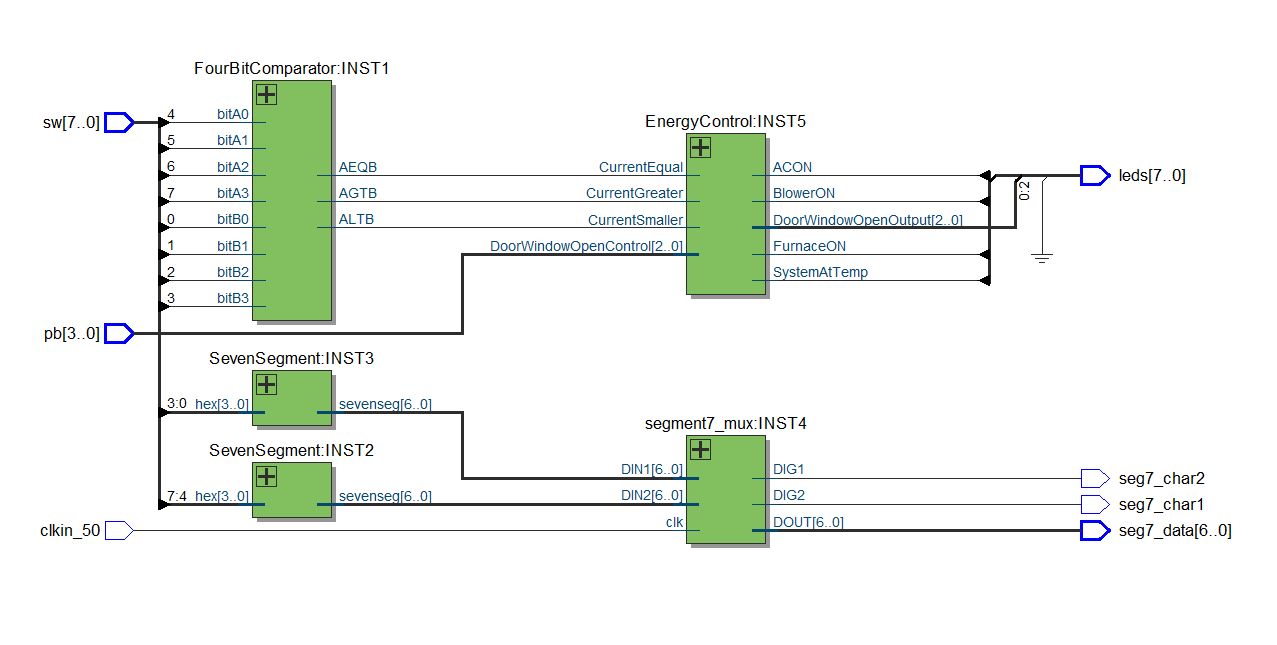
## GRAPH3: SIMULATION3 where HEXA = 5



## GRAPH4: SIMULATION4 where HEXA = 8



## GRAPH5: RTL OF TOP LEVEL DESGIN



## GRAPH6: TOTAL DESIGN LOGIC ELEMETNS

