```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);
   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 are Compx1, Compx4, dwl, vacation_mode_mux, SevenSegment and segment7_mux which are defined below
------------------------------------------------------------------

component Compx4 port (
      inp_A          : in std_logic_vector(3 downto 0); -- input current temp A sw(3..0)
      inp_B          : in std_logic_vector(3 downto 0); -- input desired temp B sw(7..4)
      AGTB           : out std_logic;
      AEQB           : out std_logic;
      ALTB           : out std_logic
      );
end component;


component vacation_mode_mux port (
      pb_button         : in std_logic;
      desired_temp      : in std_logic_vector(3 downto 0);
      fixed_setting     : out std_logic_vector(3 downto 0)
      );
end component;


component dwl port (
         AGTB           : in std_logic;
         AEQB           : in std_logic;
         ALTB           : in std_logic;
         door           : in std_logic;
         window         : in std_logic;
         leds_out       : out std_logic_vector(5 downto 0)
      );
end component;


component SevenSegment is 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 segment7_mux is 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;

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



--

   signal seg7_A        :  std_logic_vector(6 downto 0); --7 bit signal to be displayed on left 7 segment
   signal seg7_B         :  std_logic_vector(6 downto 0); --7 bit signal to be displayed on right 7 segment
   signal inp_A          : std_logic_vector(3 downto 0); -- signal for current temp A sw(3..0)
   signal inp_B          : std_logic_vector(3 downto 0); -- signal for desired temp  sw(7..4)
   signal AGTB           : std_logic; -- signals for storing the outputs from the 4-bit comparator (A>B)
   signal AEQB           : std_logic; -- (A=B)
   signal ALTB           : std_logic; -- (A<B)

   signal TEST_PASS      : std_logic; -- signal for Test_pass
   signal vac_mode       : std_logic; -- signal for Vacation Mode from pb(3)
   signal mc_test        : std_logic; -- signal for MC_TESTMODE from pb(2)
   signal window         : std_logic; -- signal for Window from pb(1)
   signal door           : std_logic; -- signal for door from pb(0)
   signal temp_signal    : std_logic_vector(3 downto 0); -- signal which will store the output from vacation_mode_mux of either desired temp or fixed setting

-- The circuit begins here

begin

Testbench1:
PROCESS (sw, AGTB, AEQB, ALTB, pb(2)) is

variable EQ_PASS, GE_PASS, LE_PASS  : std_logic := '0';

begin
      IF ((sw(3 downto 0) = sw(7 downto 4)) AND (AEQB = '1')) THEN
      EQ_PASS := '1';
      GE_PASS := '0';
      LE_PASS := '0';

      ELSIF ((sw(3 downto 0) > sw(7 downto 4)) AND (AGTB = '1')) THEN
      GE_PASS := '1';
      EQ_PASS := '0';
      LE_PASS := '0';

      ELSIF ((sw(3 downto 0) < sw(7 downto 4)) AND (ALTB = '1')) THEN
      LE_PASS := '1';
      GE_PASS := '0';
      EQ_PASS := '0';

      ELSE
      EQ_PASS := '0';
      GE_PASS := '0';
      LE_PASS := '0';

      END IF;

      TEST_PASS <= NOT(pb(2)) AND ( EQ_PASS OR GE_PASS OR LE_PASS);
      leds(6) <= TEST_PASS;
end process;

   inp_A <= sw(3 downto 0);    -- getting the inputs for current temp from the switches
   inp_B <= sw(7 downto 4);    -- getting the inputs for desired temp from the switches

-- getting inputs from the pbs
   vac_mode <= pb(3);
   mc_test <= pb(2);
   window <= pb(1);
   door <= pb(0);

   leds(7) <= NOT(pb(3)); -- when pb(3) is pressed which indicate vacation mode, then led(7) should light up

 -- Instantiating for the components declared above
   INST1: Compx4 port map(inp_A, temp_signal, AGTB, AEQB, ALTB);
   INST2: vacation_mode_mux port map(pb(3), inp_B, temp_signal);
   INST3: SevenSegment port map(temp_signal, seg7_A);
   INST4: SevenSegment port map(inp_A, seg7_B);
   INST5: segment7_mux port map(clkin_50, seg7_A, seg7_B, seg7_data, seg7_char1, seg7_char2);
   INST6: dwl port map(AGTB, AEQB, ALTB, door, window, leds(5 downto 0));

end Energy_Monitor;
```