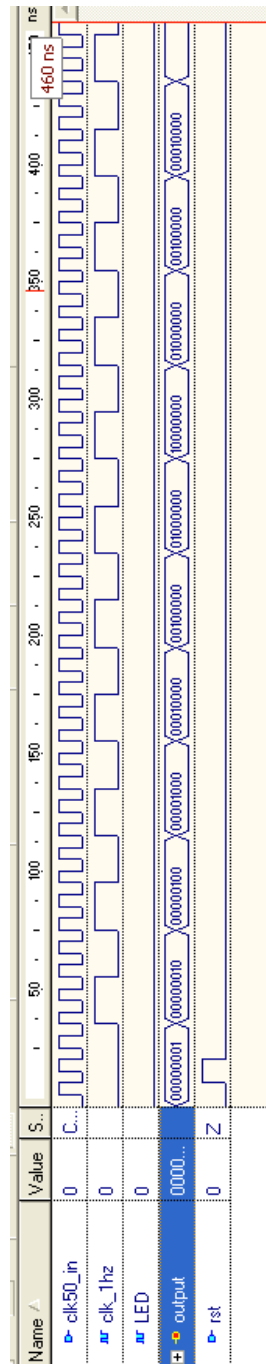


Question #1



```

--sw(7) is the reset button
--a binary combination of sw(0)and sw(1) will control the LED speed
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity main is
    Port ( clk50_in : in bit;
           sw       : in bit_vector(7 downto 0);
           output   : out bit_vector(7 downto 0));
end main;

architecture Behavioral of main is
    signal speed : integer := 25000000;
    signal clk_1hz : bit := '0';
    signal cyclon_eyes : bit_vector(7 downto 0) := "00000001";
begin

    process (sw(0),sw(1))          --setting up the speed
    begin
        if (sw(1)='0' and sw(0)='0') then speed<=25000000; end if;
        if (sw(1)='0' and sw(0)='1') then speed<=25000000; end if;
        if (sw(1)='1' and sw(0)='0') then speed<=10000000; end if;
        if (sw(1)='1' and sw(0)='1') then speed<=5000000; end if;

    end process;

    process (clk50_in,sw(7))
    variable counter : integer;
    begin
        if (clk50_in'event and clk50_in='1') then
            counter := counter + 1;
            if (counter>=speed) then
                counter := 0;
                clk_1hz <= not (clk_1hz);
            end if;
        end if;
        if (sw(7)='1') then counter := 0; end if;
    end process;

    process (clk_1hz,sw(7))
    variable loop_direction : integer;
    begin
        if (clk_1hz'event and clk_1hz='1') then
            if (cyclon_eyes="00000001") then
                loop_direction:=1;
            elsif (cyclon_eyes="10000000") then
                loop_direction:=0;
            end if;

            if (loop_direction=1) then
                cyclon_eyes <= cyclon_eyes sll 1;
            elsif (loop_direction=0) then
                cyclon_eyes <= cyclon_eyes srl 1;
            end if;
        end if;

        if (sw(7)='1') then cyclon_eyes <= "00000001"; end if;

    end process;
    output<=cyclon_eyes;
end Behavioral;

```

Question #2

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity main is
    Port ( clk50_in : in bit;
           sw : in bit_vector(7 downto 0);
           output : out std_logic_vector(7 downto 0));
end main;

architecture Behavioral of main is
    signal speed : integer := 500000000;
    signal slow_clock : bit := '0';
    signal output_binary : std_logic_vector(7 downto 0) := "00000000";
begin

    process (clk50_in,sw(7))
        variable counter : integer;
        begin
            if (clk50_in'event and clk50_in='1') then
                counter := counter + 1;
                if (counter>=speed) then
                    counter := 0;
                    slow_clock <= not (slow_clock);
                end if;
            end if;
            if (sw(7)='1') then counter := 0; end if;
        end process;

        process (slow_clock,sw(7))
        begin
            if (slow_clock'event and slow_clock='1') then
                output_binary <= output_binary + 1;
            end if;

            if (sw(7)='1') then output_binary <= "00000000"; end if;

        end process;

        output<=output_binary;
    end Behavioral;
```

Note: For both questions I used the following pin assignments:

```
NET "clk50_in" LOC = "t9" ;
NET "output<0>" LOC = "k12" ;
NET "output<1>" LOC = "p14" ;
NET "output<2>" LOC = "l12" ;
NET "output<3>" LOC = "n14" ;
NET "output<4>" LOC = "p13" ;
NET "output<5>" LOC = "n12" ;
NET "output<6>" LOC = "p12" ;
NET "output<7>" LOC = "p11" ;
NET "sw<0>" LOC = "f12" ;
NET "sw<1>" LOC = "g12" ;
NET "sw<2>" LOC = "h14" ;
NET "sw<3>" LOC = "h13" ;
NET "sw<4>" LOC = "j14" ;
NET "sw<5>" LOC = "j13" ;
NET "sw<6>" LOC = "k14" ;
NET "sw<7>" LOC = "k13" ;
```