Lab 5

Pragaash Mohan

October 27, 2020

Part 1

Top level entity

```
--Importing library
--Creating Top-level entity

library ieee;
use ieee.std_logic_1164.all;



entity task_1 is
port(
                KEY, SW           : in std_logic_vector(1 downto 0);
                LEDR              : out std_logic_vector(1 downto 0);
                HEX0, HEX1        : out std_logic_vector(0 to 6)
        );

end task_1;

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


architecture behav of task_1 is


--Importing components

                --T flip-flop
                component t_ff port
                (

                        clk           : in std_logic;
```

```
                t                                       : in std_logic;
                o_put           : out std_logic;
                clr                             : in std_logic
        );

        end component;

        -- Hexadecimal display
        component Hex7 port
        (

                c    : in std_logic_vector(3 downto 0);
                disp : out std_logic_vector(0 to 6)

        );

        end component;

    signal clk, clr         : std_logic;
    signal t, o_put    : std_logic_vector(7 downto 0);


        begin

                clk <= not KEY(0);
                clr <= not SW(0);
                LEDR <= SW;

                --states

                t(0) <= SW(1); --when Enable is high
                t(1) <= SW(1) and o_put(0);
                t(2) <= t(1) and o_put(1);
                t(3) <= t(2) and o_put(2);
                t(4) <= t(3) and o_put(3);
                t(5) <= t(4) and o_put(4);
                t(6) <= t(5) and o_put(5);
                t(7) <= t(6) and o_put(6);
-----------------------------------------------------

                --Instansiating flip-flop eight times:

                s0 : t_ff port map (clk, t(0), o_put(0), clr);
                s1 : t_ff port map (clk, t(1), o_put(1), clr);
```

```
                    s2 : t_ff port map (clk, t(2), o_put(2), clr);
                    s3 : t_ff port map (clk, t(3), o_put(3), clr);
                    s4 : t_ff port map (clk, t(4), o_put(4), clr);
                    s5 : t_ff port map (clk, t(5), o_put(5), clr);
                    s6 : t_ff port map (clk, t(6), o_put(6), clr);
                    s7 : t_ff port map (clk, t(7), o_put(7), clr);

                    --Printing to display
                    disp_1 : Hex7 port map ( o_put(7 downto 4), HEX1 );
                    disp_0 : Hex7 port map ( o_put(3 downto 0), HEX0 );

end behav;
```

T Flip-flop

```
--Creating T flip-flop

library ieee;
use ieee.std_logic_1164.all;

entity t_ff is
        port (
                        clk, clr, t      : in std_logic;
                        o_put                           : out std_logic
                        );

        end t_ff;

architecture behav of t_ff is
                signal temp : std_logic;

begin
                process (clk, clr, t, temp)
                begin

                        if rising_edge(clk) then
                                if t = '1' then
                                        temp <= not temp;
                                end if;

                        end if;


                        --if clear is low, then value set to '0'
                        if (clr <= '0') then
                                temp <= '0';

                        end if;
                        o_put <= temp;

                end process;



end behav;
```

Hex7

```
--Creating hexadecimal display

library ieee;
use ieee.std_logic_1164.all;

entity Hex7 is
        port
        (
                c               : in std_logic_vector(3 downto 0);
                disp        : out std_logic_vector(0 to 6)
        );
end Hex7;


architecture behav of Hex7 is
begin

        process (c)
        begin
                case c is
                        when "0000" => disp <= "0000001"; -- 0
                        when "0001" => disp <= "1001111"; -- 1
                        when "0010" => disp <= "0010010"; -- 2
                        when "0011" => disp <= "0000110"; -- 3
                        when "0100" => disp <= "1001100"; -- 4
                        when "0101" => disp <= "0100100"; -- 5
                        when "0110" => disp <= "0100000"; -- 6
                        when "0111" => disp <= "0001111"; -- 7
                        when "1000" => disp <= "0000000"; -- 8
                        when "1001" => disp <= "0000100"; -- 9
                        when "1010" => disp <= "0001000"; -- A
                        when "1011" => disp <= "1100000"; -- B
                        when "1100" => disp <= "0110001"; -- C
                        when "1101" => disp <= "1000010"; -- D
                        when "1110" => disp <= "0110000"; -- E
                        when "1111" => disp <= "0111000"; -- F
                        when others => disp <= "1111111";
                end case;
        end process;

end behav;
```

Part 4

Top level entity

```
--Top level

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

entity part_4 is
        port
        (
                CLOCK_50                                : in std_logic;
                HEX0, HEX1, HEX2, HEX3          : out std_logic_vector(0 to 6)

        );
end part_4;

architecture behav of part_4 is

        component mux2b4_1 port
        (
                s                                       : in std_logic_vector(1 downto 0);
                u,v,w,x                 : in std_logic_vector(0 to 6);

                m                                       : out std_logic_vector(0 to 6)
        );
        end component;

        component t_ff

        generic ( n : integer := 8 );
        port
        (
                clk                     : in std_logic;
                t                               : in std_logic;
                clr                     : in std_logic;
                o_put                   : inout std_logic_vector(n-1 downto 0)
        );

        end component;

        signal q_clk : std_logic_vector(25 downto 0);
```

```
signal clr_clk : std_logic;
signal q_disp : std_logic_vector(1 downto 0);
signal clr_disp : std_logic;


begin

clk_counter : process (CLOCK_50, q_clk)

begin
--Checking if event has occured on signal
        if CLOCK_50'event and CLOCK_50 = '0' then
                if q_clk >= 50000000 then
                        clr_clk <= '0';
                else
                        clr_clk <= '1';
                end if;
        end if;
end process;

dig_counter : process (clr_clk, q_disp)

begin
--Checking if event has occured on signal

        if clr_clk'event and clr_clk = '0' then
                if q_disp >= 3 then
                        clr_disp <= '0';
                else
                        clr_disp <= '1';
                end if;
        end if;
end process;

c26bit : t_ff
        generic map ( 26 )
        port map ( CLOCK_50, '1', clr_clk, q_clk );
c2bit : t_ff
        generic map ( 2 )
        port map ( clr_clk, '1', clr_disp, q_disp );
```

```
        --Setting up rotation value
        MUX0 : mux2b4_1 port map
        (
                q_disp,
                "0000001", -- 0
                "1000010", -- d
                "0110000", -- E
                "1001111", -- 1
                HEX0
        );

        MUX1 : mux2b4_1 port map
        (
                q_disp,
                "1001111", -- 1
                "0000001", -- 0
                "1000010", -- d
                "0110000", -- E
                HEX1
        );

        MUX2 : mux2b4_1 port map
        (
                q_disp,
                "0110000", -- E
                "1001111", -- 1
                "0000001", -- 0
                "1000010", -- d
                HEX2
        );

        MUX3 : mux2b4_1 port map
        (
                q_disp,
                "1000010", -- d
                "0110000", -- E
                "1001111", -- 1
                "0000001", -- 0
                HEX3
        );

end behav;
```

Modified counter

```
--modified bit counter

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

entity t_ff is
        generic ( n : integer := 8 );

        port(
                clk, clr, t                 : in std_logic;
                o_put                                   : inout std_logic_vector(n-1 do

                );
        end t_ff;


architecture behav of t_ff is

        begin

                process (clk, clr, t)

                begin
                        if rising_edge(clk) then
                                if t= '1' then
                                        o_put <= o_put +1;
                                end if;
                        end if;
                                if (clr <= '0') then
                                        o_put <= (others => '0');

                                end if;

                end process;

end behav;
```

MUX

```
--2bit 4-1 MUX
library ieee;
use ieee.std_logic_1164.all;

entity  mux2b4_1 is
        port
        (
                s                               : in std_logic_vector(1 downto 0);
                u,v,w,x                 : in std_logic_vector(0 to 6);

                m                               : out std_logic_vector(0 to 6)
        );

end mux2b4_1;

architecture behav of mux2b4_1 is
begin

        process (s, u, v, w, x)
        begin
                case s is
                        when "00" => m <= u;
                        when "01" => m <= v;
                        when "10" => m <= w;
                        when "11" => m <= x;
                        when others => m <= (others => 'Z');

                end case;
        end process;
end behav;
```