Lab 3

Pragaash Mohan

October 9, 2020

Part 1

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;


ENTITY part_1 IS
        PORT ( Clk, R, S : IN STD_LOGIC;
        Q : OUT STD_LOGIC);
END part_1;


ARCHITECTURE Structural OF part_1 IS
        SIGNAL R_g, S_g, Qa, Qb : STD_LOGIC ;
        ATTRIBUTE KEEP : BOOLEAN;
        ATTRIBUTE KEEP OF R_g, S_g, Qa, Qb : SIGNAL IS TRUE;


BEGIN
        R_g <= R AND Clk;
        S_g <= S AND Clk;
        Qa <= NOT (R_g OR Qb);
        Qb <= NOT (S_g OR Qa);
        Q <= Qa;

END Structural;
```

Part 2

Main VHDL

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

entity part_2 is
        port
        (
                SW                          : in std_logic_vector(1 downto 0);
                LEDR                  : out std_logic_vector(1 downto 0)
        );
end part_2;

architecture Behavioral of part_2 is
        component part2_1 port
        (
                clk,d        : in std_logic;
                q                : out std_logic
        );

        end component;

        signal clk, d, q : std_logic;

begin

        clk <= SW(1);
        d <= SW(0);
        LEDR(0) <= q;
        LEDR(1) <= clk;


        part2_2 : part2_1 port map
        ( clk, d, q );

end Behavioral;
```

Component

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

entity part2_1 is
        port
        (
                clk,d        : in std_logic;
                q                : out std_logic
        );
end part2_1;

architecture Structural of part2_1 is
        signal r, s, r_g, s_g, q_a, q_b : std_logic;

        attribute keep : boolean;
        attribute keep of r, r_g, s_g, q_a, q_b : signal is true;

begin
        s <= d;
        r <= not d;
        s_g <= not(s and clk);
        r_g <= not(r and clk);
        q_a <= not(s_g and q_b);
        q_b <= not(r_g and q_a);

        q <= q_a;

end Structural;
```

Part 3

Main VHDL

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

entity part_3 is
        port
        (
                SW                          : in std_logic_vector(1 downto 0);
                LEDR                : out std_logic_vector(1 downto 0)
        );
end part_3;

architecture Behavioral of part_3 is
        component part2_1 port
        (
                clk, d : in std_logic;
                q                   : out std_logic
        );
        end component;

        signal clk, d, qm, q : std_logic;

begin

        clk <= SW(1);
        d <= SW(0);
        LEDR(0) <= q;
        LEDR(1) <= clk;

        master : part2_1 port map
        ( not clk, d, qm );

        slave : part2_1 port map
        ( clk, qm, q );
end Behavioral;
```

Part 4

Main VHDL

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

entity part_4 is
        port
        (
                clk, d      : in std_logic;
                qa, qb, qc        : out std_logic
        );
end part_4;

architecture Behavioral of part_4 is
        component dLatch port
        (
                clk, d        : in std_logic;
                q                   : out std_logic
        );
        end component;

        component p_ff port
        (
                clk, d        : in std_logic;
                q                   : out std_logic
        );
        end component;

        component n_ff port
        (
                clk, d         : in std_logic;
                q                   : out std_logic
        );
        end component;

begin

        part4_1 : dLatch port map
        ( clk, d, qa );

        part4_2 : p_ff port map
        ( clk, d, qb );
```

```
        part4_3 : n_ff port map
        ( clk, d, qc );

end Behavioral;
```

Components

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

entity dLatch is
        port
        (
                clk, d        : in std_logic;
                q                   : out std_logic
        );
end dLatch;

architecture Behavioral of dLatch is

begin

        process (clk, d)
        begin
                if clk = '1' then
                        q <= d;
                end if;
        end process;

end Behavioral;

library ieee;
use ieee.std_logic_1164.all;

entity p_ff is
        port
        (
                clk, d : in std_logic;
                q                 : out std_logic
        );
end p_ff;

architecture Behavioral of p_ff is

begin
        process (clk, d)
        begin
                if clk'event and clk = '1' then
                        q <= d;
```

```
                end if;
        end process;

end Behavioral;

library ieee;
use ieee.std_logic_1164.all;

entity n_ff is
        port
        (
                clk, d          : in std_logic;
                q                       : out std_logic
        );
end n_ff;

architecture Behavioral of n_ff is

begin
        process (clk, d)
        begin
                if clk'event and clk = '0' then
                        q <= d;
                end if;
        end process;

end Behavioral;
```