Course Name: Embedded Systems Design

Course Number and Section: 14:332:493:02

Homework #3

Submitted by: Angie Flores - af770

Date Submitted: 02/20/2020

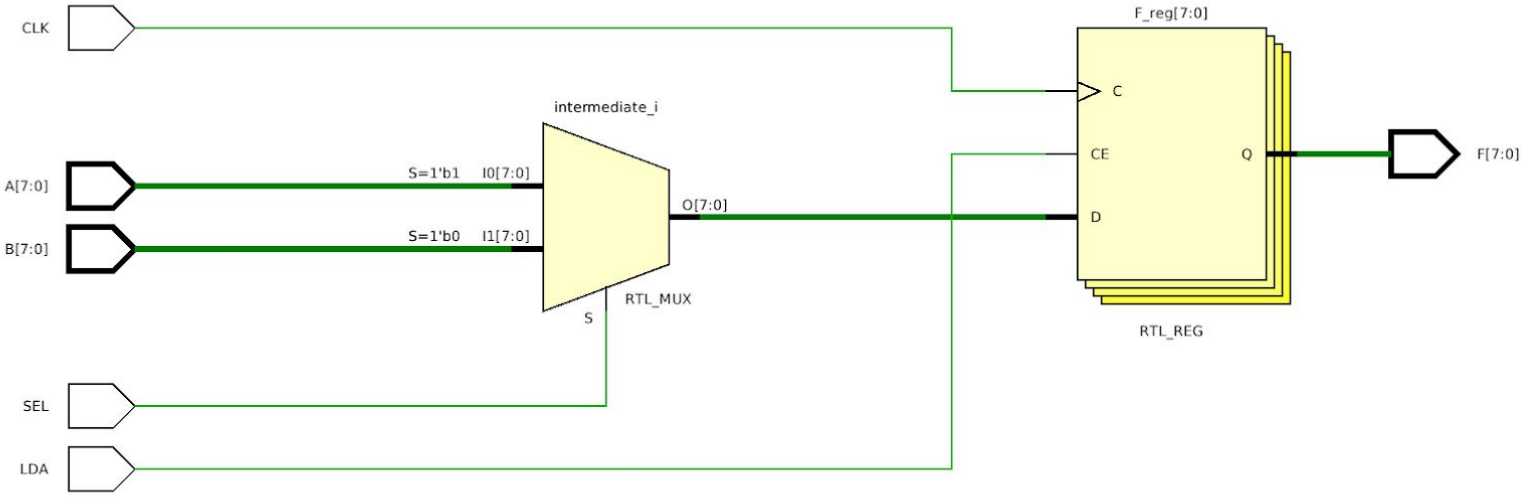
CHAPTER 7

- **EXERCISE 1**

VHDL Code:

```
1  library ieee;
2      use ieee.std_logic_1164.all;
3      use ieee.numeric_std.all;
4
5  entity entidad is
6      port(
7
8          A,B  : in std_logic_vector(7 downto 0);
9          SEL, CLK, LDA  : in std_logic;
10     F: out std_logic_vector(7 downto 0)
11
12
13     );
14 end entidad;
15
16 architecture behavior of entidad is
17
18     signal intermediate : std_logic_vector(7 downto 0);
19
20 begin
21     r:process(CLK)
22     begin
23         if (rising_edge(CLK)) then
24             if (LDA = '1') then
25             F <= intermediate;
26             end if;
27         end if;
28     end process;
29 with SEL select
30 intermediate<=A when '1',
31           B when '0',
32           (others => '0') when others;
33
34 end behavior;
35
```

RTL in Vivado:

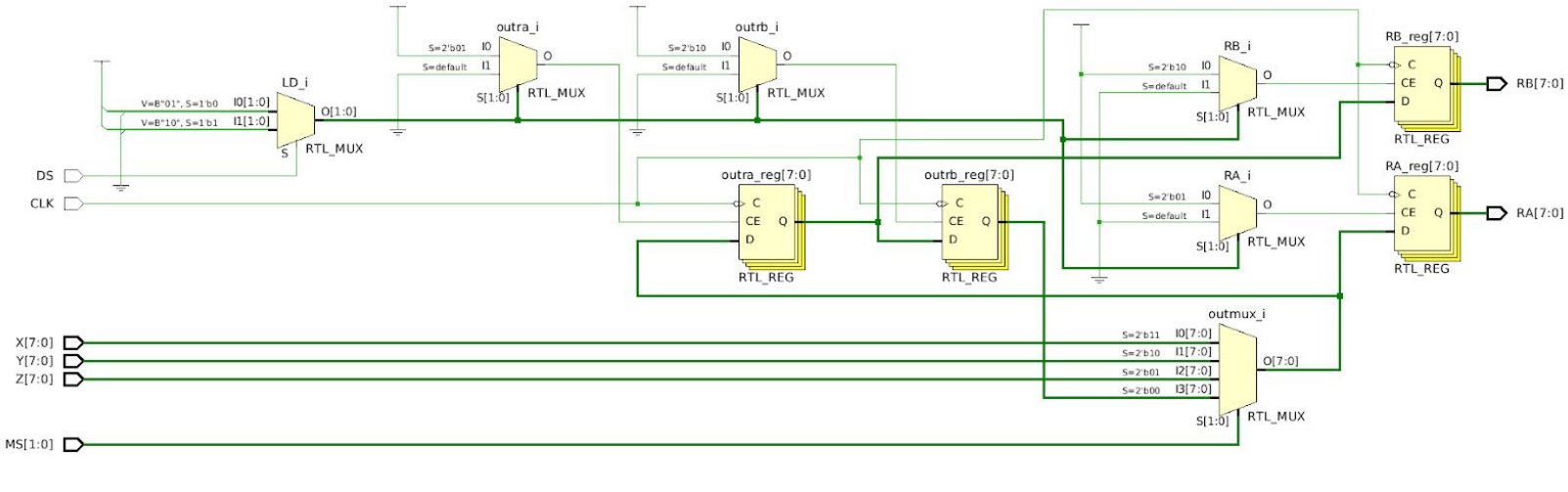


- **EXERCISE 2**

VHDL Code:

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

entity entidad is
    port(

        X,Y,Z  : in std_logic_vector(7 downto 0);
    MS: in std_logic_vector(1 downto 0);
        CLK, DS  : in std_logic;
    RA, RB: out std_logic_vector(7 downto 0)


    );
end entidad;

architecture behavior of entidad is

    signal outmux, outra, outrb : std_logic_vector(7 downto 0);
    signal LD: std_logic_vector(1 downto 0);

begin
    registera: process(CLK)
    begin
        if (falling_edge(CLK)) then
            if (LD="01") then
            outra <= outmux;
            RA <= outmux;
            end if;
        end if;
    end process;

    registerb: process(CLK)
    begin
        if (falling_edge(CLK)) then
            if (LD="10") then
            outrb <= outra;
            RB <= outra;
            end if;
        end if;
    end process;

dec:process(DS)
begin
    case DS is
      when '0' => LD <= "01";
      when '1' => LD <= "10";
      when others => LD <= "00";
    end case;
end process;

with MS select
outmux <= X when "11",
      Y when "10",
      Z when "01",
      outrb when "00",
      "00000000" when others;
end behavior;
```

RTL in Vivado:



- **EXERCISE 3**

VHDL Code:

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

entity entidad is
    port(

        X , Y : in std_logic_vector(7 downto 0);
    LDA,LDB,S1,S0,CLK: in std_logic;
    RB: out std_logic_vector(7 downto 0)


    );
end entidad;

architecture behavior of entidad is

    signal outmux1, outmux2, outra, outrb : std_logic_vector(7 downto 0);

begin
    registera: process(CLK)
    begin
        if (rising_edge(CLK)) then
            if (LDA='1') then
            outra <= outmux1;
            end if;
        end if;
    end process;

    registerb: process(CLK)
    begin
        if (rising_edge(CLK)) then
            if (LDB='1') then
            outrb <= outmux2;
            RB <= outmux2;
            end if;
        end if;
    end process;

with S1 select
outmux1 <= X when '1',
      outrb when '0',
      "00000000" when others;


with S0 select
outmux2 <= outra when '1',
       Y when '0',
      "00000000" when others;

end behavior;
```

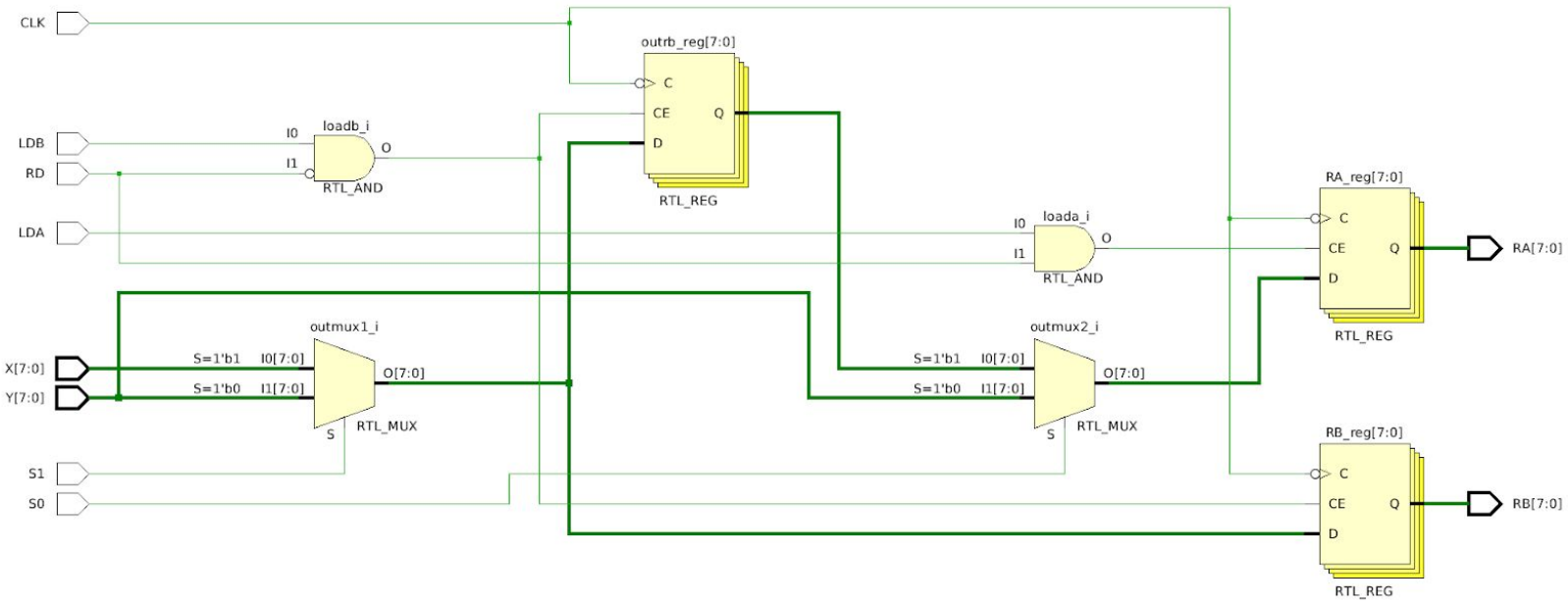
RTL in Vivado:

- **EXERCISE 4**

VHDL Code:

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

entity entidad is
    port(

        X , Y : in std_logic_vector(7 downto 0);
    RD, LDA,LDB,S1,S0,CLK: in std_logic;
    RA, RB: out std_logic_vector(7 downto 0)


    );
end entidad;

architecture behavior of entidad is

    signal outmux1, outmux2, outrb : std_logic_vector(7 downto 0);
    signal loada,loadb : std_logic;

begin
    registera: process(CLK)
    begin
        if (falling_edge(CLK)) then
            if (loada='1') then
            RA <= outmux2;
            end if;
        end if;
    end process;

    registerb: process(CLK)
    begin
        if (falling_edge(CLK)) then
            if (loadb='1') then
            outrb <= outmux1;
            RB <= outmux1;
            end if;
        end if;
    end process;

loadb <= LDB AND NOT(RD);
loada <= LDA AND RD;

with S0 select
outmux2 <= outrb when '1',
       Y when '0',
      "00000000" when others;


with S1 select
outmux1 <= X when '1',
       Y when '0',
      "00000000" when others;

end behavior;
```

RTL in Vivado:

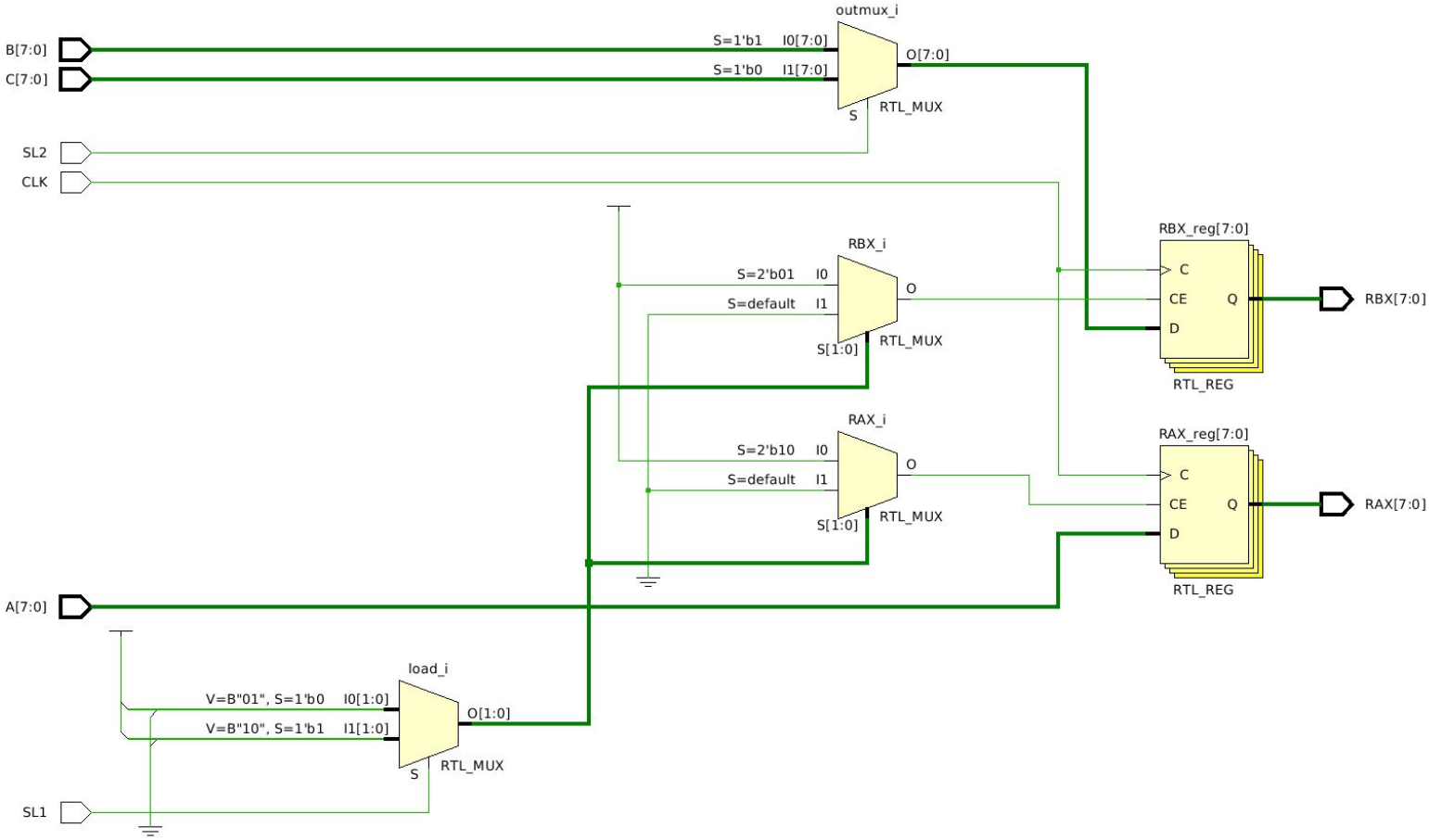


- **EXERCISE 5**

VHDL Code:

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

entity entidad is
    port(

        A, B,C: in std_logic_vector(7 downto 0);
    SL1, SL2, CLK: in std_logic;
       RAX,RBX: out std_logic_vector(7 downto 0)

    );
end entidad;

architecture behavior of entidad is

    signal outmux:std_logic_vector(7 downto 0);
    signal load:std_logic_vector(1 downto 0);

begin
    registera: process(CLK)
    begin
        if (rising_edge(CLK)) then
            if (load="10") then
            RAX <= A;
            end if;
        end if;
    end process;

    registerb: process(CLK)
    begin
        if (rising_edge(CLK)) then
            if (load="01") then
            RBX <= outmux;
            end if;
        end if;
    end process;

dec:process(SL1)
begin
    case SL1 is
      when '0' => load <= "01";
      when '1' => load <= "10";
      when others => load <= "00";
    end case;
end process;

with SL2 select
outmux <= B when '1',
      C when '0',
      "00000000" when others;

end behavior;
```

RTL in Vivado:



- **EXERCISE 6**

VHDL Code:

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

entity entidad is
    port(

        A, B,C: in std_logic_vector(7 downto 0);
    SEL1, SEL2, CLK: in std_logic;
    RAP, RBP: out std_logic_vector(7 downto 0)

    );
end entidad;

architecture behavior of entidad is

    signal outmux:std_logic_vector(7 downto 0);
    signal load:std_logic_vector(1 downto 0);

begin
    registera: process(CLK)
    begin
        if (rising_edge(CLK)) then
            if (load="10") then
            RAP <= outmux;
            end if;
        end if;
    end process;

    registerb: process(CLK)
    begin
        if (rising_edge(CLK)) then
            if (load="01") then
            RBP <= C;
            end if;
        end if;
    end process;

dec:process(SEL2)
begin
    case SEL2 is
      when '0' => load <= "01";
      when '1' => load <= "10";
      when others => load <= "00";
    end case;
end process;

with SEL1 select
outmux <= A when '1',
      B when '0',
      "00000000" when others;

end behavior;
```

RTL in Vivado: