Paul Kafka

9/25/13

Emb. Sys. Des.

Homework 3

74163 VHDL

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

entity hw3 is
    port
    (
        LOAD       : in std_logic;
        DATA_A     : in std_logic;
        DATA_B     : in std_logic;
        clk        : in std_logic;
        DATA_C     : in std_logic;
        DATA_D     : in std_logic;
        CLR        : in std_logic;
        ENP        : in std_logic;
        ENT        : in std_logic;

        Q_A       : buffer std_logic;
        Q_B       : buffer std_logic;
        Q_C       : buffer std_logic;
        Q_D       : buffer std_logic;

        RC0       : out std_logic
    );

end entity;

architecture rtl of hw3 is
signal a : std_logic_vector (5 downto 0);
signal b : std_logic_vector (5 downto 0);
signal c : std_logic_vector (5 downto 0);
signal d : std_logic_vector (5 downto 0);

signal temp: std_logic_vector (3 downto 0);
signal first : std_logic;

begin
-- Combinational Logic
    first <= ENP AND ENT;

    --For Q_A
    a(0) <= (NOT LOAD) OR (NOT CLR);
    a(1) <= a(0) NAND a(3);
    a(2) <= first OR a(0);
    a(3) <= (DATA_A NAND CLR) NAND a(0);
    a(4) <= a(1) AND a(2);
    a(5) <= a(2) AND a(3);

    --For Q_B
    b(0) <= Q_A AND first;
    b(1) <= a(0) NAND b(3);
    b(2) <= b(0) OR a(0);
    b(3) <= (DATA_B NAND CLR) NAND a(0);
    b(4) <= b(1) AND b(2);
    b(5) <= b(2) AND b(3);

    --For Q_C
    c(0) <= Q_B AND Q_A AND first;
    c(1) <= a(0) NAND c(3);
    c(2) <= c(0) OR a(0);
    c(3) <= (DATA_C NAND CLR) NAND a(0);
    c(4) <= c(1) AND c(2);
    c(5) <= c(2) AND c(3);

    --For Q_D
    d(0) <= Q_C AND Q_B AND Q_A AND first;
    d(1) <= a(0) NAND d(3);
    d(2) <= a(0) OR d(0);
    d(3) <= (DATA_D NAND CLR) NAND a(0);
    d(4) <= d(1) AND d(2);
    d(5) <= d(2) AND d(3);


--JK Flip Flops
process(a,b,c,d, clk)


    begin
      --a(4) is J and a(5) is K
        if (clk 'EVENT AND clk = '1') then
            if (a(4) ='0' and a(5)='0') then
               temp(0) <= temp(0);
            elsif (a(4)='0' and a(5)='1') then
               temp(0) <= '0';
            elsif (a(4)='1' and a(5)='0') then
               temp(0) <= '1';
            elsif (a(4)='1' and a(5)='1') then
               temp(0) <= not (temp(0));
            end if;
         end if;

        if (clk 'EVENT AND clk = '1') then
            if (b(4) ='0' and b(5)='0') then
               temp(1) <= temp(1);
            elsif (b(4)='0' and b(5)='1') then
               temp(1) <= '0';
            elsif (b(4)='1' and b(5)='0') then
               temp(1) <= '1';
            elsif (b(4)='1' and b(5)='1') then
               temp(1) <= not (temp(1));
            end if;
         end if;

         if (clk 'EVENT AND clk = '1') then
            if (c(4) ='0' and c(5)='0') then
               temp(2) <= temp(2);
            elsif (c(4)='0' and c(5)='1') then
               temp(2) <= '0';
            elsif (c(4)='1' and c(5)='0') then
               temp(2) <= '1';
            elsif (c(4)='1' and c(5)='1') then
               temp(2) <= not (temp(2));
            end if;
         end if;

        if (clk 'EVENT AND clk = '1') then
            if (d(4) ='0' and d(5)='0') then
               temp(3) <= temp(3);
            elsif (d(4)='0' and d(5)='1') then
               temp(3) <= '0';
            elsif (d(4)='1' and d(5)='0') then
               temp(3) <= '1';
            elsif (d(4)='1' and d(5)='1') then
               temp(3) <= not (temp(3));
            end if;
         end if;

    end process;

--Outputs
Q_A <= temp(0);
Q_B <= temp(1);
Q_C <= temp(2);
Q_D <= temp(3);
RC0 <=  Q_D AND Q_C and Q_B AND Q_A AND ENT;

end rtl;
```

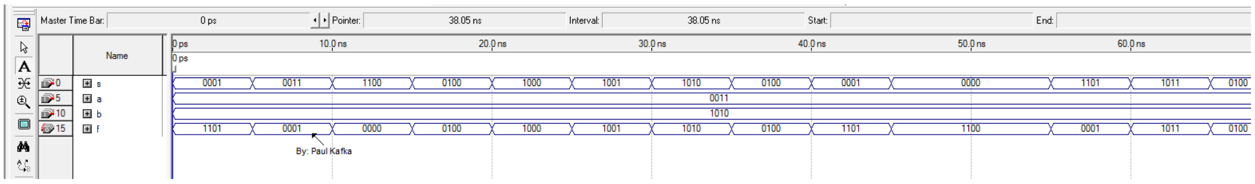
Simulation

N/A

74LS181 VHDL

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

entity hw3b is
    port
    (
        a      : in std_logic;
        b      : in std_logic;
        s      : in std_logic_vector (3 downto 0);

        f       : out std_logic
    );

end entity;

architecture logic of hw3b is
begin

process(a,b,s)

    begin
        case s is
            when "0000" =>
                f <= NOT a;
            when "0001" =>
                f <= NOT (a and b);
            when "0010" =>
                f <= NOT a or b;
            when "0011" =>
                f <= '1';
            when "0100" =>
                f <= NOT (a or b);
            when "0101" =>
                f <= NOT b;
            when "0110" =>
                f <= NOT (a xor b);
            when "0111" =>
                f <= a or NOT b;
            when "1000" =>
                f <= not a and b;
            when "1001" =>
                f <= a xor b;
            when "1010" =>
                f <= b;
            when "1011" =>
                f <= a or b;
            when "1100" =>
                f <= '0';
            when "1101" =>
                f <= a and not b;
            when "1110" =>
                f <= a and b;
            when "1111" =>
                f <= a;
        end case;
    end process;

end logic;
```

Simulation



Master Time Bar: 0 ps

Pointer: 38.05 ns

Interval: 38.05 ns

Start:

End:

Name
0 ps
10.0 ns
20.0 ns
30.0 ns
40.0 ns
50.0 ns
60.0 ns
0 s: 0001, 0011, 1100, 0100, 1000, 1001, 1010, 0100, 0001, 0000, 1101, 1011, 0100
5 a: 0011
10 b: 1010
15 f: 1101, 0001, 0000, 0100, 1000, 1001, 1010, 0100, 1101, 1100, 0001, 1011, 0100
By: Paul Kafka