CPLD MAX3000A

D Flip-Flop and Shift Reg

4th April, 2023

VHDL Programming and CPLD Application the VHDL codes.

Components used:

Quartus software,CPLD Board and USB connector

Summary:

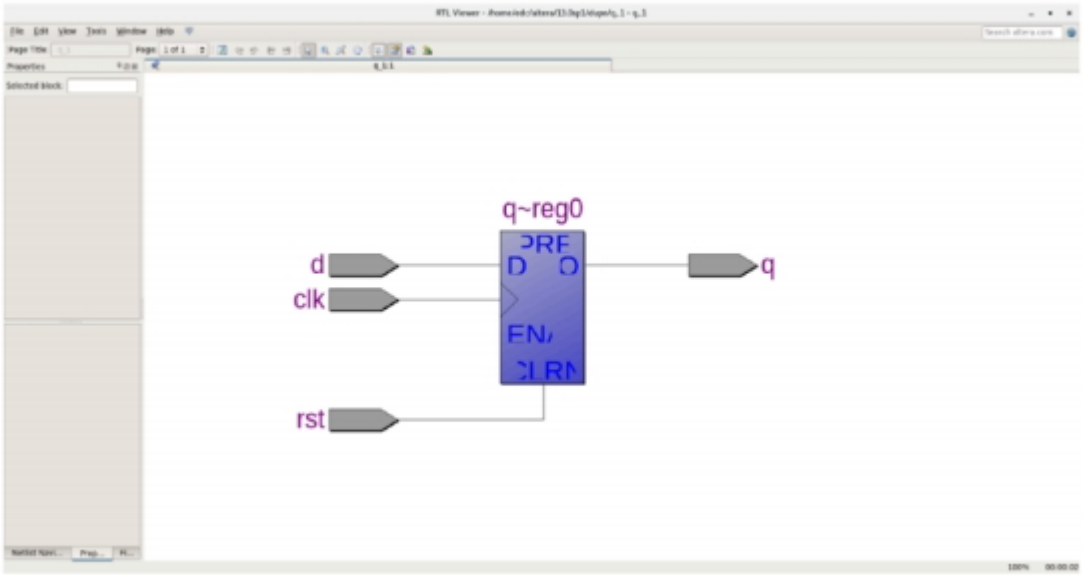
Using Quartus application to write the VHDL code for the given problems and dump the code the CPLD(blue board).

1. DFF

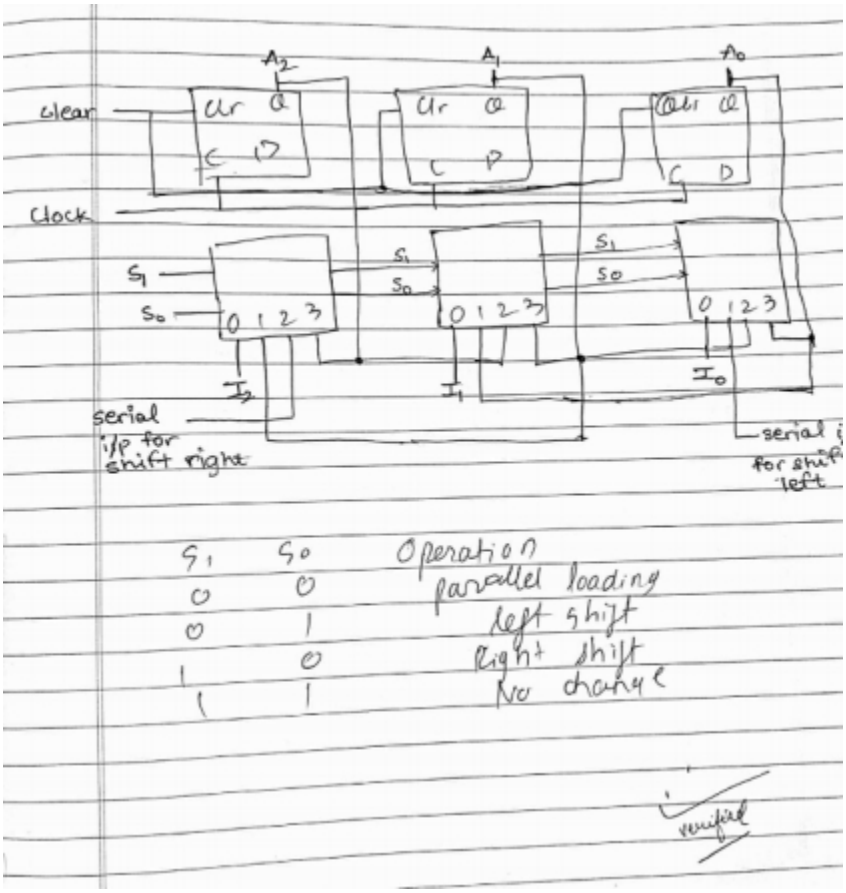
Code:

D Flip-Flop in Behavioural form:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity DFF_b is
port(d,clk,rst:in std_logic;
q:out std_logic
);
end DFF_b;

architecture Behavioral of DFF_b is
begin
  process(rst,clk)
  begin
    if(rst= '1') then
      q<='0';
      elsif(clk'event and clk= '1') then
      q<=d;
    end if;
  end process;
end Behavioral
```



2. Design of Shift Reg

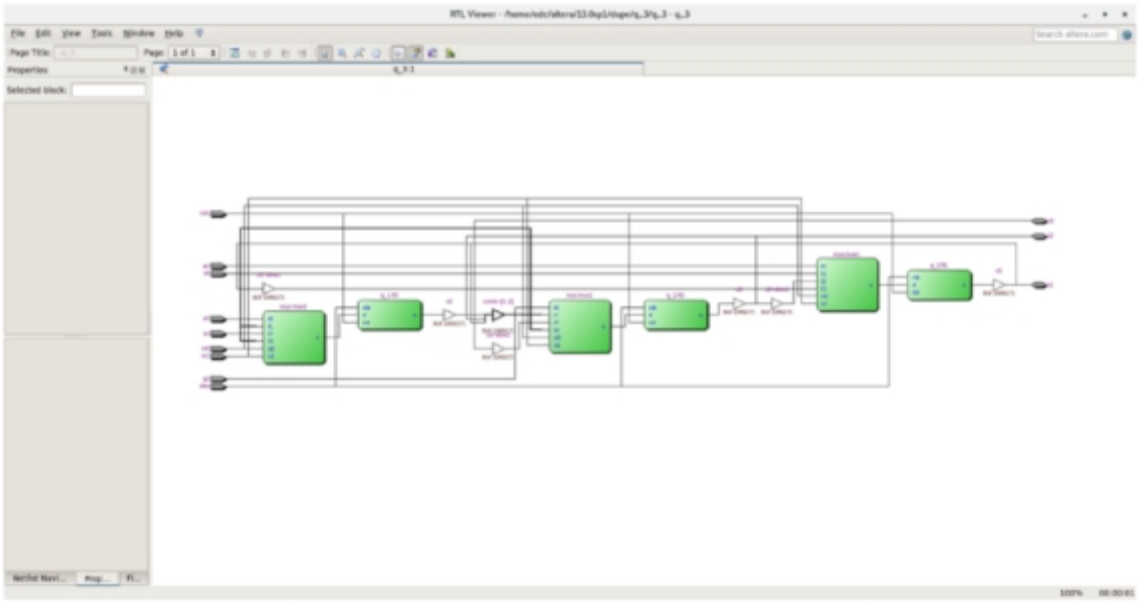


3. Shift Reg

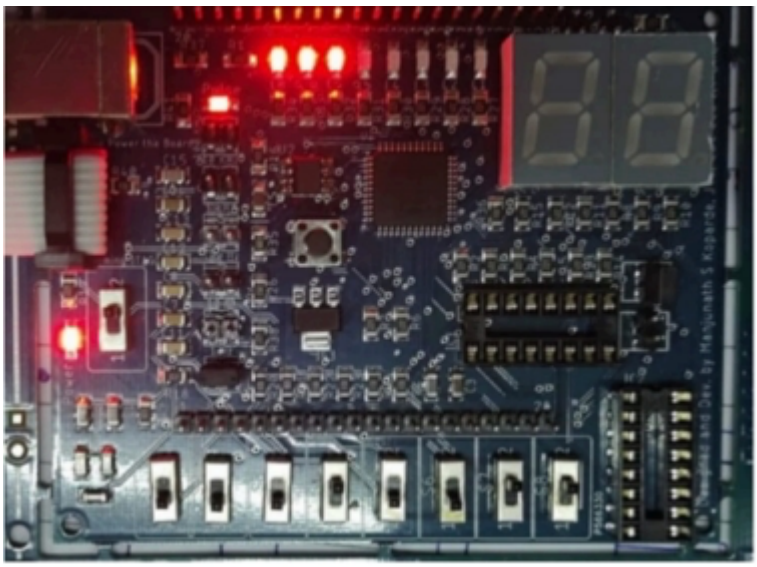
```
--MUX component:
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity mux is
port(i0,i1, i2, i3, s0, s1: in std_logic;
y: out std_logic);
end mux;

architecture behavior of mux is
  Begin
  c1: process (s0,s1)
    Begin
      if(s0= '0') then
        if(s1= '0') then
          y<=i0;
          else
          y<=i1;
        end if;
        else
        if(s1= '0') then
          y<=i2;
        else
          y<=i3;
        end if;
      end if;
  end process c1;
end behavior;

-- D Flip-Flop component:
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
entity dff_shiftreg is
port(d,clk,rst:in std_logic;
q:out std_logic
);
end dff_shiftreg;

architecture Behavioral of dff_shiftreg is
  begin
  process(rst,clk)
  begin
    if(rst= '1') then
    --IF RESET IS 1 THEN SET OUTPUT TO 0
    q<='0';
    elsif(clk'event and clk= '1') then
    q<=d;
    end if;
  end process;
end Behavioral;

-- Universal Shift Register:
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity shift-reg is
port(a1, a2, a3, sli, sri, m0, m1, clks, rsts: in std_logic;
o1, o2, o3: inout std_logic);
end shift-reg;

architecture structural of usr is
  component dff_shiftreg is
  port(d,clk,rst:in std_logic;
  q:out std_logic);
  end component;
```

```
  component mux is
  port(i0, i1, i2, i3, s0, s1: in std_logic;
  y: out std_logic);
  end component;

  signal d1,d2,d3:std_logic;
  begin
    mux1: mux port map(i0=>a1, i1=>sli, i2=>o2, i3=>o1, s0=>m0,
s1=>m1, y=>d1);
    mux2: mux port map(i0=>a2, i1=>o1, i2=>o3, i3=>o2, s0=>m0,
s1=>m1, y=>d2);
    mux3: mux port map(i0=>a3, i1=>o2, i2=>sri, i3=>o3, s0=>m0,
s1=>m1, y=>d3);
    ff1: dff_shiftreg port map(d=>d1, clk=>clks, rst=>rsts, q=>o1);
    ff2: dff_shiftreg port map(d=>d2, clk=>clks, rst=>rsts, q=>o2);
    ff3: dff_shiftreg port map(d=>d3, clk=>clks, rst=>rsts, q=>o3);
end structural;
```



Circuit Snapshots:



Conclusion:

Writing VHDL codes and dumping those codes in CPLD boards for simulation of circuits for easier implementation of a combination of logic gates.

Thank You.