**ASSIGNMENT 2**

**Name-Shahrukh Padaniya**

**Main code:-**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity MainCode is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : in STD\_LOGIC;

D : in STD\_LOGIC;

AA : inout STD\_LOGIC;

BB : inout STD\_LOGIC;

CC : inout STD\_LOGIC;

DD : inout STD\_LOGIC;

CLR : in STD\_LOGIC;

CLK : in STD\_LOGIC;

T : in STD\_LOGIC);

end MainCode;

architecture Behavioral of MainCode is

component counter is

port(clk : in STD\_LOGIC;

clr: in STD\_LOGIC;

Enable: in STD\_LOGIC;

QA: out STD\_LOGIC;

QB: out STD\_LOGIC;

QC: out STD\_LOGIC;

QD: out STD\_LOGIC);

end component counter;

component comparator4bit is

Port ( A0 : in STD\_LOGIC;

A1 : in STD\_LOGIC;

A2 : in STD\_LOGIC;

A3 : in STD\_LOGIC;

B0 : in STD\_LOGIC;

B1 : in STD\_LOGIC;

B2 : in STD\_LOGIC;

B3 : in STD\_LOGIC;

GR : out STD\_LOGIC;

EQ : out STD\_LOGIC;

LS : out STD\_LOGIC);

end component comparator4bit;

signal q3,q2,q1,q0,reset,gr,ls:std\_logic;

begin

g1:counter port map(clk=>clk,clr=>(reset or clr),enable=>t,qa=>q0,qb=>q1,qc=>q2,qd=>q3);

AA<=q0;

BB<=q1;

CC<=q2;

DD<=q3;

g2: comparator4bit port map(B0=>AA,B1=>BB,B2=>CC,B3=>DD,A0=>A,A1=>B,A2=>C,A3=>D,EQ=>reset,gr=>gr,ls=>ls);

end Behavioral;

**Code for Counter: -**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity counter is

Port ( clk : in STD\_LOGIC;

clr: in STD\_LOGIC;

Enable: in STD\_LOGIC;

QA: out STD\_LOGIC;

QB: out STD\_LOGIC;

QC: out STD\_LOGIC;

QD: out STD\_LOGIC);

end counter;

architecture Behavioral of counter is

signal q1,q2,q3,q4:STD\_LOGIC;

component TFF is

port(input:in STD\_LOGIC;clk: in STD\_LOGIC;clr: in STD\_LOGIC;x: out STD\_LOGIC);

end component TFF;

begin

g1:TFF port map (input=>Enable,clk=>clk,clr=>clr,x=>q1);

QA<=q1;

g2:TFF port map (input=>q1,clk=>clk,clr=>clr,x=>q2);

QB<=q2;

g3:TFF port map (input=>(q2 and q1),clk=>clk,clr=>clr,x=>q3);

QC<=q3;

g4:TFF port map (input=>(q3 and q2 and q1),clk=>clk,clr=>clr,x=>q4);

QD<=q4;

end Behavioral;

**Code for T flip flop: -**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity TFF is

Port ( input : in STD\_LOGIC;

clk : in STD\_LOGIC;

clr : in STD\_LOGIC;

x : out STD\_LOGIC;

xbar:out STD\_LOGIC);

end TFF;

architecture Behavioral of TFF is

signal n: STD\_LOGIC;

begin

process(clk,clr)

begin

if clr = '1' then

n<='0';

elsif (clk='1' and clk'event) then

if input='0' then

n<=n;

elsif input = '1' then

n<=not(n);

end if;

end if;

end process;

x<=n;

xbar<=not(n);

end Behavioral;

Code for Comparator 4 bit: -

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity comparator4bit is

Port ( A0 : in STD\_LOGIC;

A1 : in STD\_LOGIC;

A2 : in STD\_LOGIC;

A3 : in STD\_LOGIC;

B0 : in STD\_LOGIC;

B1 : in STD\_LOGIC;

B2 : in STD\_LOGIC;

B3 : in STD\_LOGIC;

GR : out STD\_LOGIC;

EQ : out STD\_LOGIC;

LS : out STD\_LOGIC);

end comparator4bit;

architecture Behavioral of comparator4bit is

component OneBitComparator is

port(A:in std\_logic;B:in std\_logic;GREATER:inout std\_logic;EQUAL:inout std\_logic;LESS:inout std\_logic);

end component OneBitComparator;

signal ls1,ls2,ls3,ls4,eq1,eq2,eq3,eq4,gr1,gr2,gr3,gr4: std\_logic;

begin

g1:OneBitComparator port map(A=>A0,B=>B0,GREATER=>gr1,EQUAL=>eq1,LESS=>ls1);

g2:OneBitComparator port map(A=>A1,B=>B1,GREATER=>gr2,EQUAL=>eq2,LESS=>ls2);

g3:OneBitComparator port map(A=>A2,B=>B2,GREATER=>gr3,EQUAL=>eq3,LESS=>ls3);

g4:OneBitComparator port map(A=>A3,B=>B3,GREATER=>gr4,EQUAL=>eq4,LESS=>ls4);s

EQ<= eq1 and eq2 and eq3 and eq4;

LS<= ls4 or (eq4 and ls3) or (eq4 and eq3 and ls2) or (eq4 and eq3 and eq1 and ls1);

GR<= gr4 or (eq4 and gr3) or (eq4 and eq3 and gr2) or (eq4 and eq3 and eq1 and gr1);

end Behavioral;

**Code for 1 bit comparator: -**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

LESS : out STD\_LOGIC;

EQUAL : out STD\_LOGIC;

GREATER : out STD\_LOGIC);

end OneBitComparator;

architecture Behavioral of OneBitComparator is

begin

process(a,b)

begin

LESS<=(not A)and B;

EQUAL<=A xnor B;

GREATER<=A and (not B);

end process;

end Behavioral;

**Output:-**

