-------------------------------------**4 Bit Ring Counter (LEFT SHIFT) USING IF ELSE**---------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity ringcounter is

 Port ( data : in STD\_LOGIC\_VECTOR (03 downto 0);

 output : out STD\_LOGIC\_VECTOR (03 downto 0);

 clk, clr, l : in STD\_LOGIC);

end ringcounter;

architecture Behavioral of ringcounter is

signal temp: std\_logic\_vector(3 downto 0);

begin

process(clk, clr, l)

 begin

 If (clr = '1') then temp <= "0000";

 ElsIf (l = '1') then temp <= data;

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

 temp <= temp(2 downto 0)&temp(3);

 End If;

End Process;

output <= temp;

end Behavioral;

-------------------------------------**4 Bit Ring Counter (LEFT SHIFT) USING IF ELSE**---------------------------------------------



-------------------------------------**4 Bit Twisted Ring Counter USING IF ELSE**---------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity twistedringcounter is

 Port ( data : in STD\_LOGIC\_VECTOR (03 downto 0);

 output : out STD\_LOGIC\_VECTOR (03 downto 0);

 clk, clr, l: in STD\_LOGIC);

end twistedringcounter;

architecture Behavioral of twistedringcounter is

signal temp: std\_logic\_vector(3 downto 0);

begin

process(clk, clr)

 begin

 If (clr = '1') then temp <= "0000";

 ElsIf (l = '1') then temp <= data;

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

 temp <= temp(2 downto 0)&not(temp(3));

 End If;

End Process;

output <= temp;

end Behavioral;

-------------------------------------**4 Bit Twisted Ring Counter USING IF ELSE**---------------------------------------------



-------------------------------------**8 Bit Bidirectional Shift Register USING IF ELSE**---------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity bireg is

 Port ( din : in STD\_LOGIC;

 sin : in STD\_LOGIC;

 m : in STD\_LOGIC;

 clk : in STD\_LOGIC;

 clr : in STD\_LOGIC;

 sl : out STD\_LOGIC;

 sr : out STD\_LOGIC);

end bireg;

architecture Behavioral of bireg is

signal temp: std\_logic\_vector(7 downto 0);

begin

process (clk,clr)

 begin

 If(clr = '0') then temp<="00000000";

 ElsIf(clk'event and clk = '0') then

 If(m = '1') then temp <= din & temp(6 downto 0);

 Else temp <= temp(7 downto 1) & sin;

 End If; End If;

End Process;

 sr <= temp(7);

 sl <= temp(0);

end Behavioral;

-------------------------------------**8 Bit Bidirectional Shift Register USING IF ELSE**---------------------------------------------



-------------------------------------**DECODER 3:8 USING CASE**---------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating

---- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity DEOCDER38CASE1 is

 Port ( D : in STD\_LOGIC\_VECTOR (02 downto 0);

 Y : out STD\_LOGIC\_VECTOR (07 downto 0));

end DEOCDER38CASE1;

architecture Behavioral of DEOCDER38CASE1 is

begin

PROCESS(D)

 BEGIN

 CASE D IS

 WHEN "000" => Y <= "00000001";

 WHEN "001" => Y <= "00000010";

 WHEN "010" => Y <= "00000100";

 WHEN "011" => Y <= "00001000";

 WHEN "100" => Y <= "00010000";

 WHEN "101" => Y <= "00100000";

 WHEN "110" => Y <= "01000000";

 WHEN "111" => Y <= "10000000";

 WHEN OTHERS => Y <= "ZZZZZZZZ";

 END CASE;

END PROCESS;

end Behavioral;

-------------------------------------**DECODER 3:8 USING CASE**---------------------------------------------



-------------------------------------**4 BIT ADDER USING STRUCTURAL MODEL**---------------------------------------------

-- PROGRAM FOR 1 BIT ADDER

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity ADDER1BIT is

 Port(A : in std\_logic;

 B : in std\_logic;

 CO :out std\_logic;

 CIN : IN STD\_LOGIC;

 Y : out std\_logic);

end ADDER1BIT;

ARCHITECTURE DATAFLOW OF ADDER1BIT IS

BEGIN

Y <= (A XOR B XOR CIN);

CO <= ((A AND B) OR (A AND CIN) OR (B AND CIN));

END DATAFLOW;

-- PROGRAM FOR 4BIT ADDER

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity ADDER4STRUCT is

 Port ( A : in STD\_LOGIC\_VECTOR (03 downto 0);

 B : in STD\_LOGIC\_VECTOR (03 downto 0);

 CO : out STD\_LOGIC;

 CIN : in STD\_LOGIC;

 Y : out STD\_LOGIC\_VECTOR (03 downto 0));

end ADDER4STRUCT;

architecture STRUCTURAL of ADDER4STRUCT is

COMPONENT ADDER1BIT IS

PORT(A : in std\_logic;

 B : in std\_logic;

 CO : out std\_logic;

 CIN : IN STD\_LOGIC;

 Y : out std\_logic);

end COMPONENT;

SIGNAL TEMPO : STD\_LOGIC\_VECTOR(3 DOWNTO 0);

SIGNAL TEMP : STD\_LOGIC\_VECTOR(2 DOWNTO 0);

BEGIN

U1 : ADDER1BIT PORT MAP (A(0),B(0),TEMP(0),CIN,TEMPO(0));

U2 : ADDER1BIT PORT MAP (A(1),B(1),TEMP(1),TEMP(0),TEMPO(1));

U3 : ADDER1BIT PORT MAP (A(2),B(2),TEMP(2),TEMP(1),TEMPO(2));

U4 : ADDER1BIT PORT MAP (A(3),B(3),CO,TEMP(2),TEMPO(3));

Y <= TEMPO;

end STRUCTURAL;

-------------------------------------**4 BIT ADDER USING STRUCTURAL MODEL**---------------------------------------------

