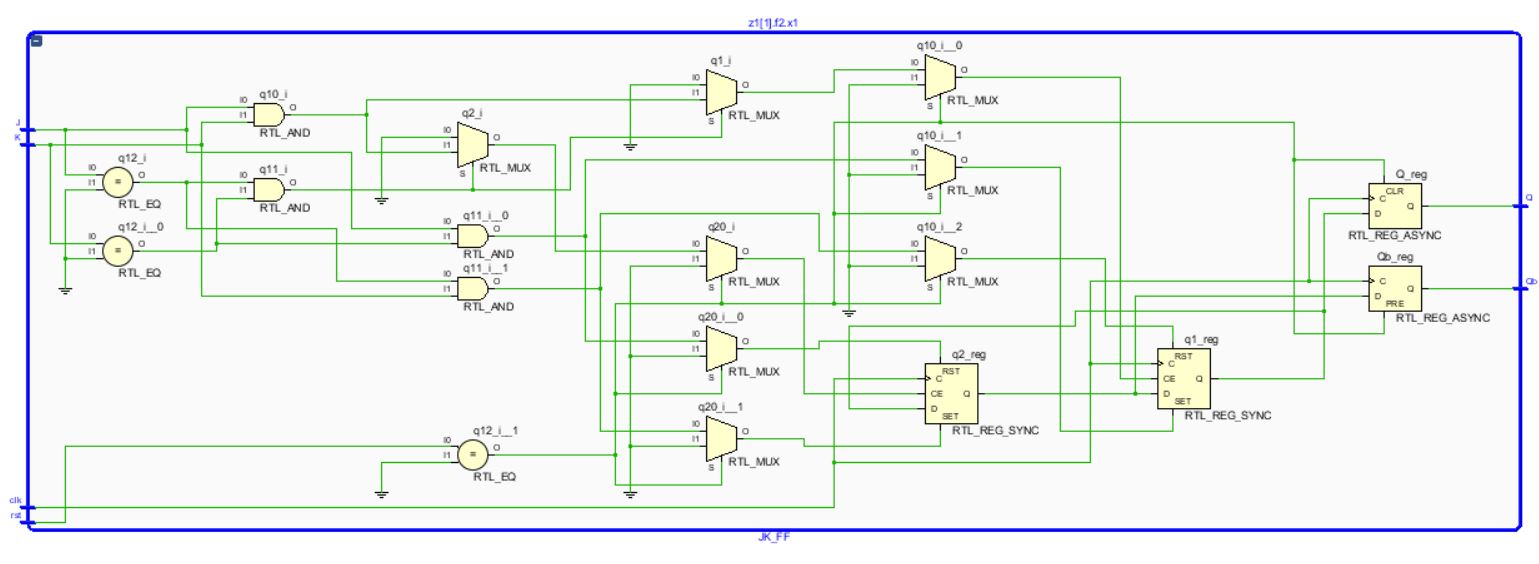
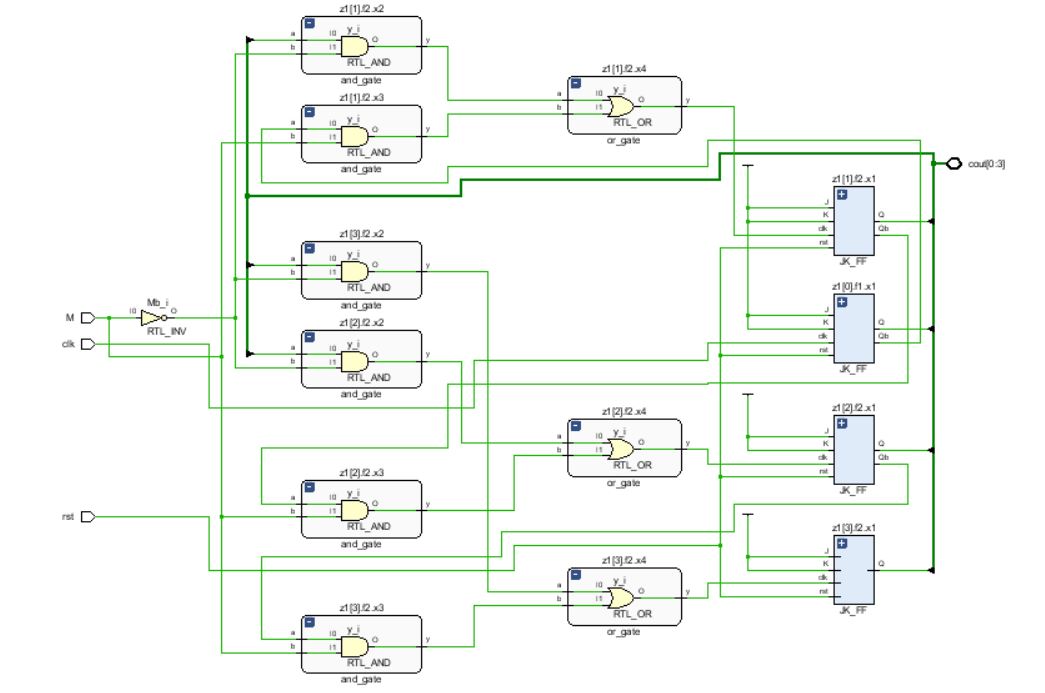
**Practical 6**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | **Aim:** Write a VHDL Code to Implement 4- bit up-down counter using JK flip flop | | | |   **Code:**  library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  entity U\_D\_Count\_4b is  Port ( M : in STD\_LOGIC;  Mb : inout STD\_LOGIC;  clk : in STD\_LOGIC;  rst : in STD\_LOGIC;  cout : out STD\_LOGIC\_VECTOR (0 to 3));    end U\_D\_Count\_4b;  architecture Behavioral of U\_D\_Count\_4b is  component JK\_FF is  Port ( clk : in STD\_LOGIC;  rst : in STD\_LOGIC;  J : in STD\_LOGIC;  K : in STD\_LOGIC;  Q : out STD\_LOGIC;  Qb : out STD\_LOGIC);  end component JK\_FF;  component and\_gate is  Port ( a : in STD\_LOGIC;  b : in STD\_LOGIC;  y : out STD\_LOGIC);  end component and\_gate;  component or\_gate is  Port ( a : in STD\_LOGIC;  b : in STD\_LOGIC;  y : out STD\_LOGIC);  end component or\_gate;  signal cout1:std\_logic\_vector(0 to 3);  signal q1:std\_logic\_vector(0 to 3):="0000";  signal d:std\_logic\_vector(0 to 2);  signal u:std\_logic\_vector(0 to 2);  signal c1:std\_logic\_vector(0 to 2);  --signal Mb:std\_logic;  begin  Mb <= not M;  z1:for i in 0 to 3 generate  f1: if (i=0)generate  x1:JK\_FF port map(clk,rst,'1','1',cout1(i),q1(i));    end generate f1;    f2: if (i>0) generate  x2:and\_gate port map(cout1(i-1),Mb,d(i-1));  x3:and\_gate port map(q1(i-1),M,u(i-1));  x4:or\_gate port map(d(i-1),u(i-1),c1(i-1));  x1:JK\_FF port map(c1(i-1),rst,'1','1',cout1(i),q1(i));  end generate f2;  end generate z1;  cout<=cout1;  end Behavioral; |
|  |

**RTL DIAGRAM:**

****

****

**Test bench Code :**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity Tb\_U\_D\_Count\_4b is

-- Port ( );

end Tb\_U\_D\_Count\_4b;

architecture Behavioral of Tb\_U\_D\_Count\_4b is

component U\_D\_Count\_4b is

Port ( M : in STD\_LOGIC;

Mb : inout STD\_LOGIC;

clk : in STD\_LOGIC;

rst : in STD\_LOGIC;

cout : out STD\_LOGIC\_VECTOR (0 to 3));

end component U\_D\_Count\_4b;

signal m1,mb1,clk1,rst1:std\_logic;

signal cout1:std\_logic\_vector(0 to 3);

begin

X1:U\_D\_Count\_4b port map(m1,mb1,clk1,rst1,cout1);

process

begin

rst1<='0';

wait for 5ns;

rst1<='1';

wait;

end process;

process

begin

clk1<='0';

wait for 5ns;

clk1<='1';

wait for 5ns;

end process;

process

begin

m1<='1';--upcount

wait for 500ns;

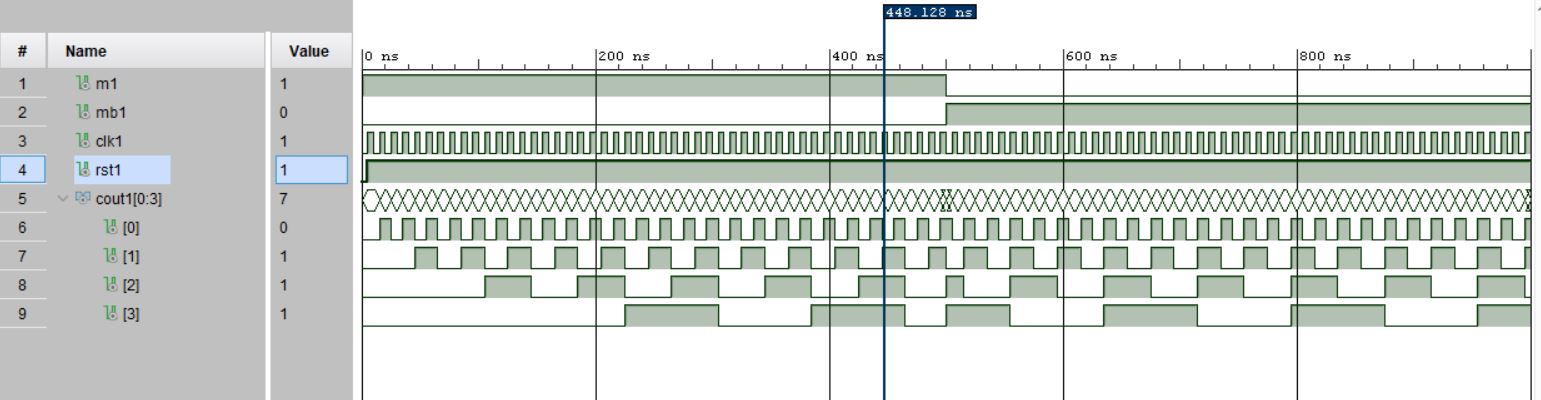
m1<='0';--downcount

wait;

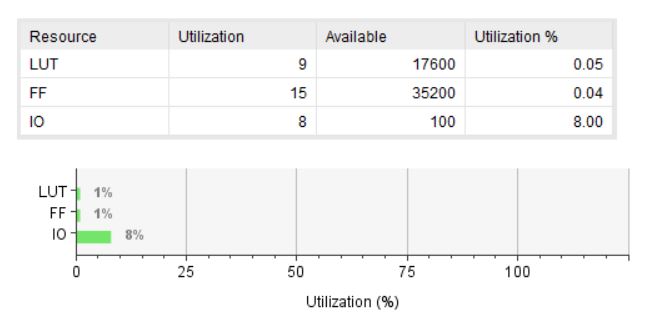
end process;

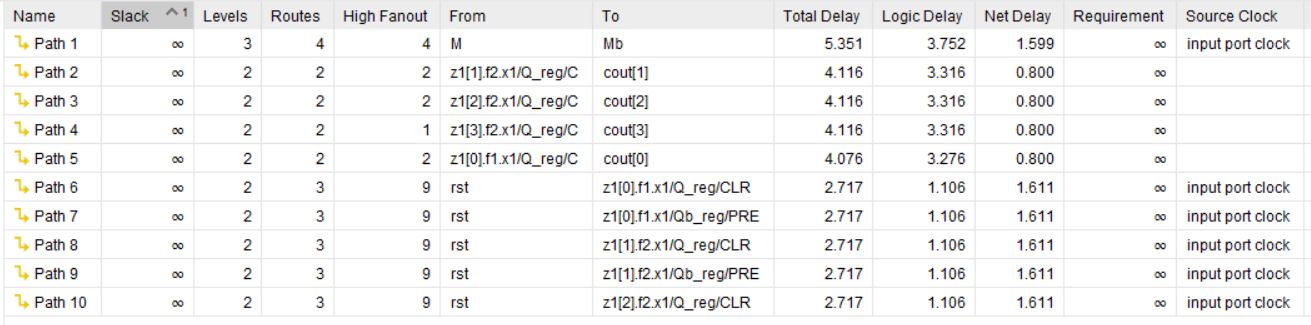
end Behavioral;

**SIMULATION WAVEFORM :**

****

**SYNTHESIS SUMMARY:**

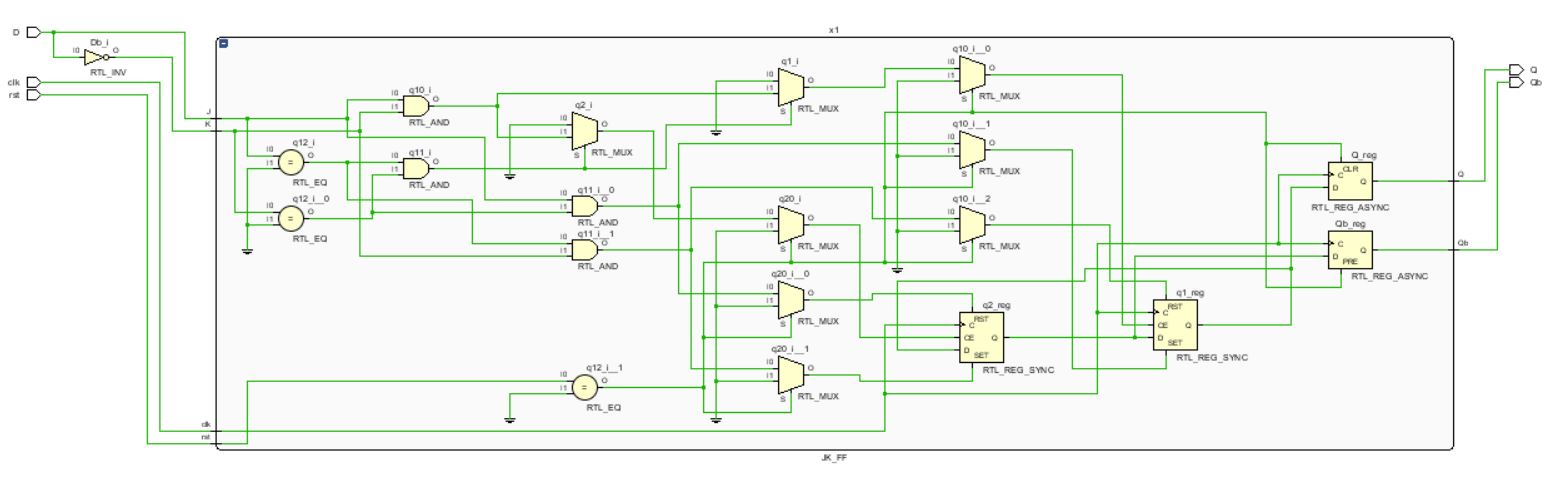
****

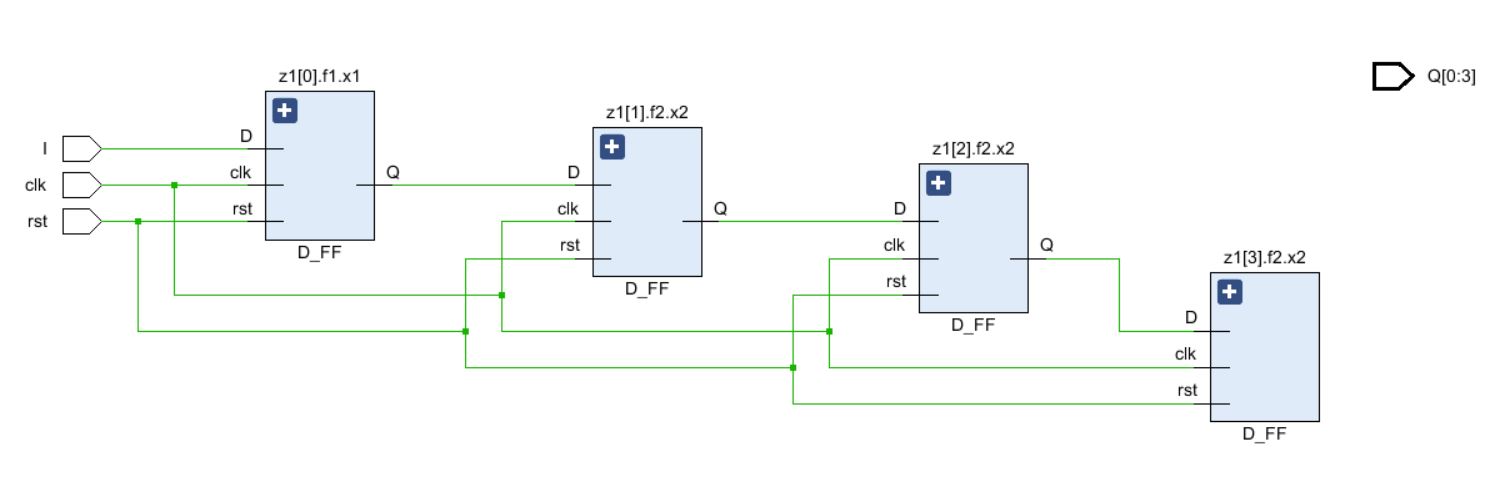


Maximum Combinational Delay: 5.351nSec

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | **Aim:** Write a VHDL Code to implement 4-bit serial in-parallel out shift register | | | |   **Code:**  library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  entity Sin\_Pout\_4b is  Port ( clk : in STD\_LOGIC;  rst : in STD\_LOGIC;  I : in STD\_LOGIC;  Q : out STD\_LOGIC\_VECTOR (0 to 3));  end Sin\_Pout\_4b;  architecture Behavioral of Sin\_Pout\_4b is  component D\_FF is  Port ( clk : in STD\_LOGIC;  rst : in STD\_LOGIC;  D : in STD\_LOGIC;  Q : out STD\_LOGIC;  Qb : out STD\_LOGIC);  end component D\_FF;  signal u1:std\_logic\_vector(0 to 3);  signal u2:std\_logic\_vector(0 to 3);  begin  z1: for k in 0 to 3 generate  f1:if(k=0) generate    x1:D\_FF port map(clk,rst,I,u1(k),u2(k));  end generate f1;  f2:if(k>0) generate    x2:D\_FF port map(clk,rst,u1(k-1),u1(k),u2(k));  end generate f2;  end generate z1;  Q<=u1;  end Behavioral; |
|  |

**RTL DIAGRAM:**

****

****

**Test bench Code :**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity Tb\_Sin\_Pout\_4b is

-- Port ( );

end Tb\_Sin\_Pout\_4b;

architecture Behavioral of Tb\_Sin\_Pout\_4b is

component Sin\_Pout\_4b is

Port ( clk : in STD\_LOGIC;

rst : in STD\_LOGIC;

I : in STD\_LOGIC;

Q : inout STD\_LOGIC\_VECTOR (0 to 3));

end component Sin\_Pout\_4b;

signal k,clk1,rst1:std\_logic;

signal Q1:std\_logic\_vector(0 to 3):="0000";

begin

X1:Sin\_Pout\_4b port map(clk1,rst1,k,Q1);

process

begin

rst1<='0';

wait for 5ns;

rst1<='1';

wait;

end process;

process

begin

clk1<='0';

wait for 5ns;

clk1<='1';

wait for 5ns;

end process;

process

begin

k<='1';

wait for 10ns;

k<='0';

wait for 10ns;

k<='1';

wait for 10ns;

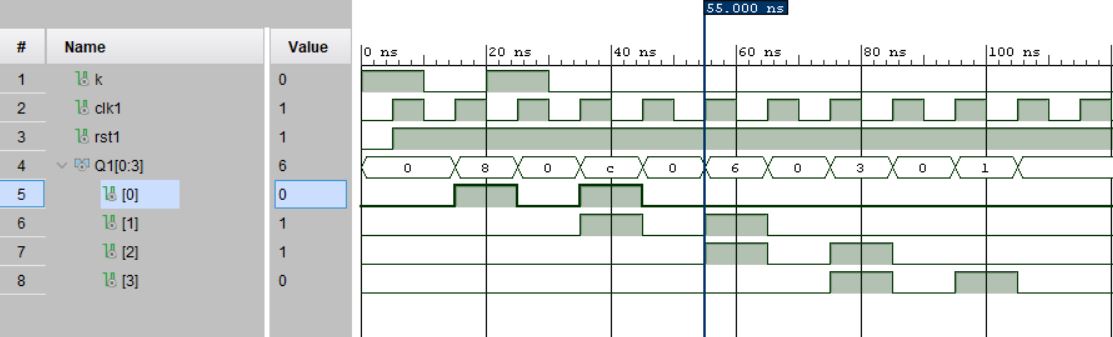
k<='0';

wait;

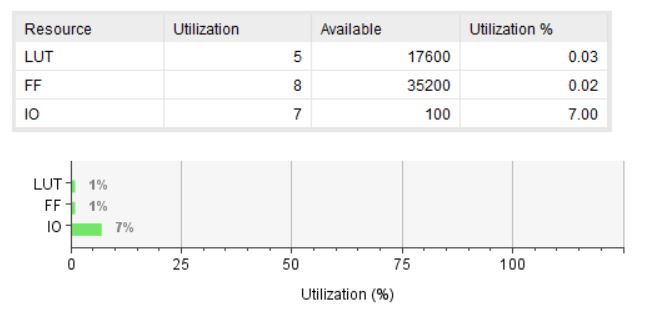
end process;

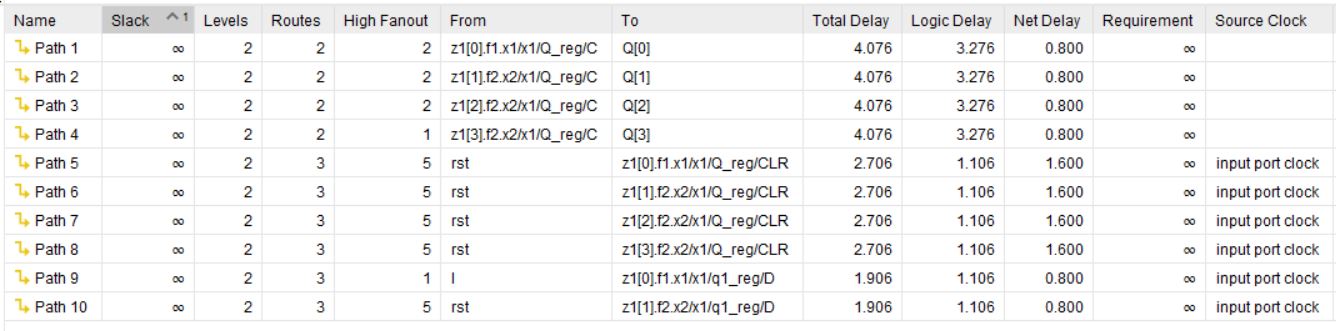
end Behavioral;

**SIMULATION WAVEFORM :**

****

**SYNTHESIS SUMMARY:**

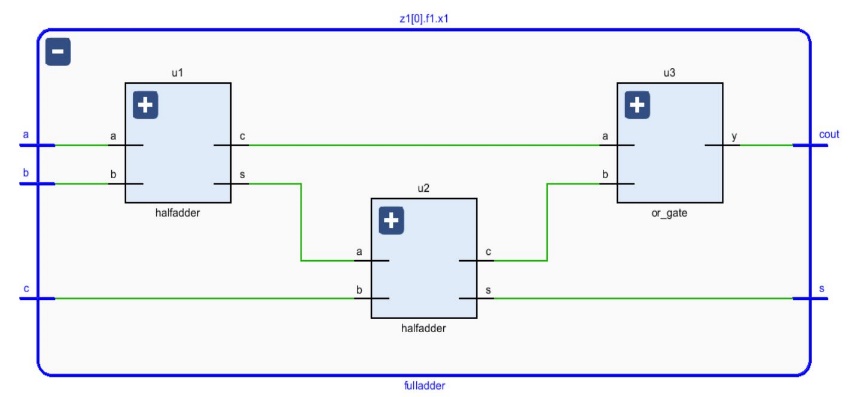
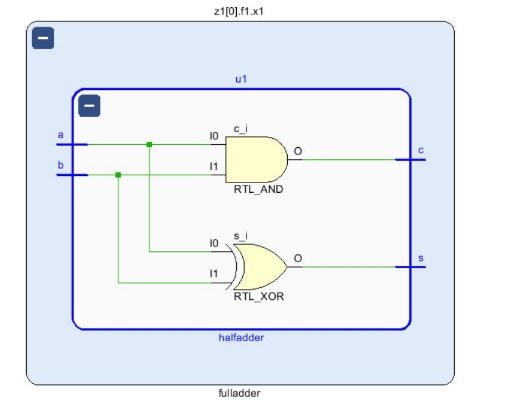
****

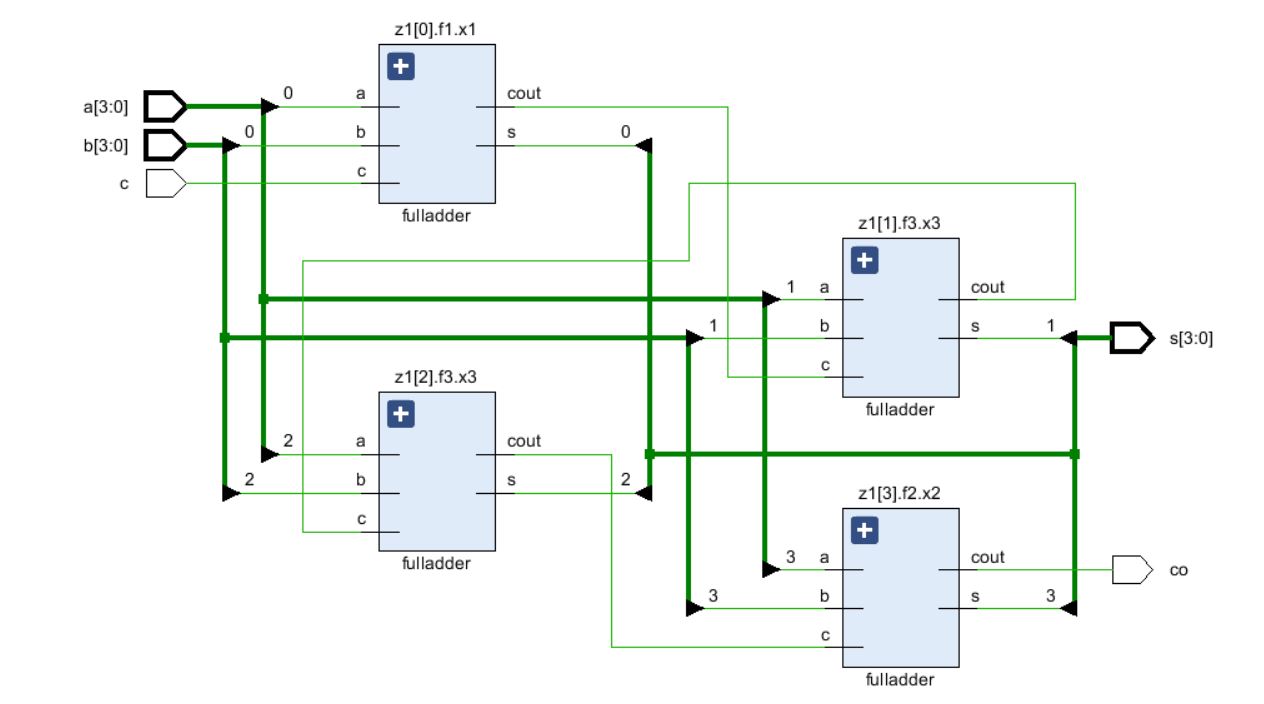


Maximum Combinational Delay: 4.076nSec

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | **Aim:** Write a VHDL Code to implement 4-bit ripple adder using generate statement | | | | |   **Code:**  library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  entity rpl\_adder\_4b is  Port ( a : in STD\_LOGIC\_VECTOR (3 downto 0);  b : in STD\_LOGIC\_VECTOR (3 downto 0);  c : in STD\_LOGIC;  s : out STD\_LOGIC\_VECTOR (3 downto 0);  co : out STD\_LOGIC);  end rpl\_adder\_4b;  architecture Behavioral of rpl\_adder\_4b is  component fulladder is  Port ( a : in STD\_LOGIC;  b : in STD\_LOGIC;  c : in STD\_LOGIC;  s : out STD\_LOGIC;  cout : out STD\_LOGIC);  end component fulladder;  signal c1:std\_logic\_vector(0 to 2);  begin  z1: for i in 0 to 3 generate  f1:if(i=0) generate  x1:fulladder port map(a(i),b(i),c,s(i),c1(0));  end generate f1;    f2:if(i=3) generate  x2:fulladder port map(a(i),b(i),c1(i-1),s(i),co);  end generate f2;    f3:if(0<i and i<3) generate  x3:fulladder port map(a(i),b(i),c1(i-1),s(i),c1(i));  end generate f3;    end generate z1;  end Behavioral; |
|  |

**RTL DIAGRAM:**

****

****

**Test bench Code :**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity Tb\_rpl\_adder\_4b is

-- Port ( );

end Tb\_rpl\_adder\_4b;

architecture Behavioral of Tb\_rpl\_adder\_4b is

component rpl\_adder\_4b is

Port ( a : in STD\_LOGIC\_VECTOR (3 downto 0);

b : in STD\_LOGIC\_VECTOR (3 downto 0);

c : in STD\_LOGIC;

s : out STD\_LOGIC\_VECTOR (3 downto 0);

co : out STD\_LOGIC);

end component rpl\_adder\_4b;

signal a1 : STD\_LOGIC\_VECTOR (3 downto 0);

signal b1 : STD\_LOGIC\_VECTOR (3 downto 0);

signal c1 : STD\_LOGIC;

signal s1 : STD\_LOGIC\_VECTOR (3 downto 0);

signal c01 : STD\_LOGIC;

begin

X1:rpl\_adder\_4b port map(a1,b1,c1,s1,c01);

process

begin

c1<='0';

a1<="0000";

b1<="0000";

wait for 10ns;

a1<="1111";

b1<="0000";

wait for 10ns;

a1<="1111";

b1<="1111";

wait for 10ns;

a1<="1111";

b1<="1010";

wait for 10ns;

c1<='1';

a1<="0000";

b1<="0000";

wait for 10ns;

a1<="1111";

b1<="0000";

wait for 10ns;

a1<="1111";

b1<="1111";

wait for 10ns;

a1<="1111";

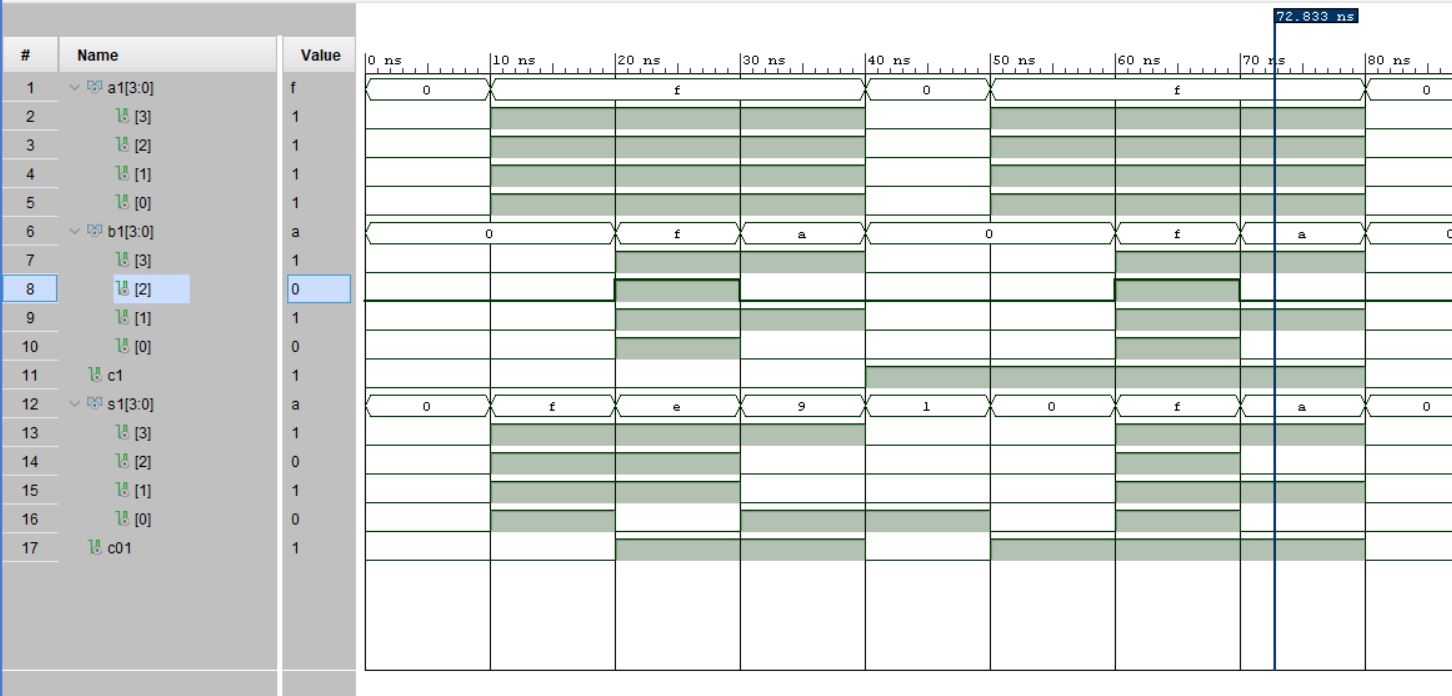
b1<="1010";

wait for 10ns;

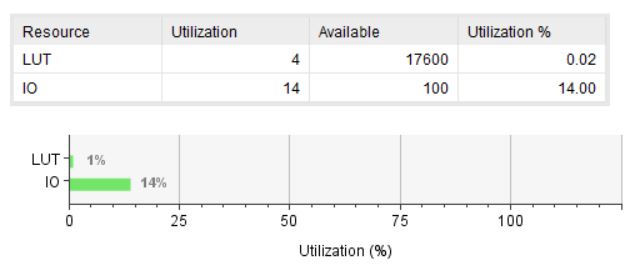
end process;

end Behavioral;

**SIMULATION WAVEFORM :**

****

**SYNTHESIS SUMMARY:**

****



Maximum Combinational Delay: 5.942nSec