Practical 6

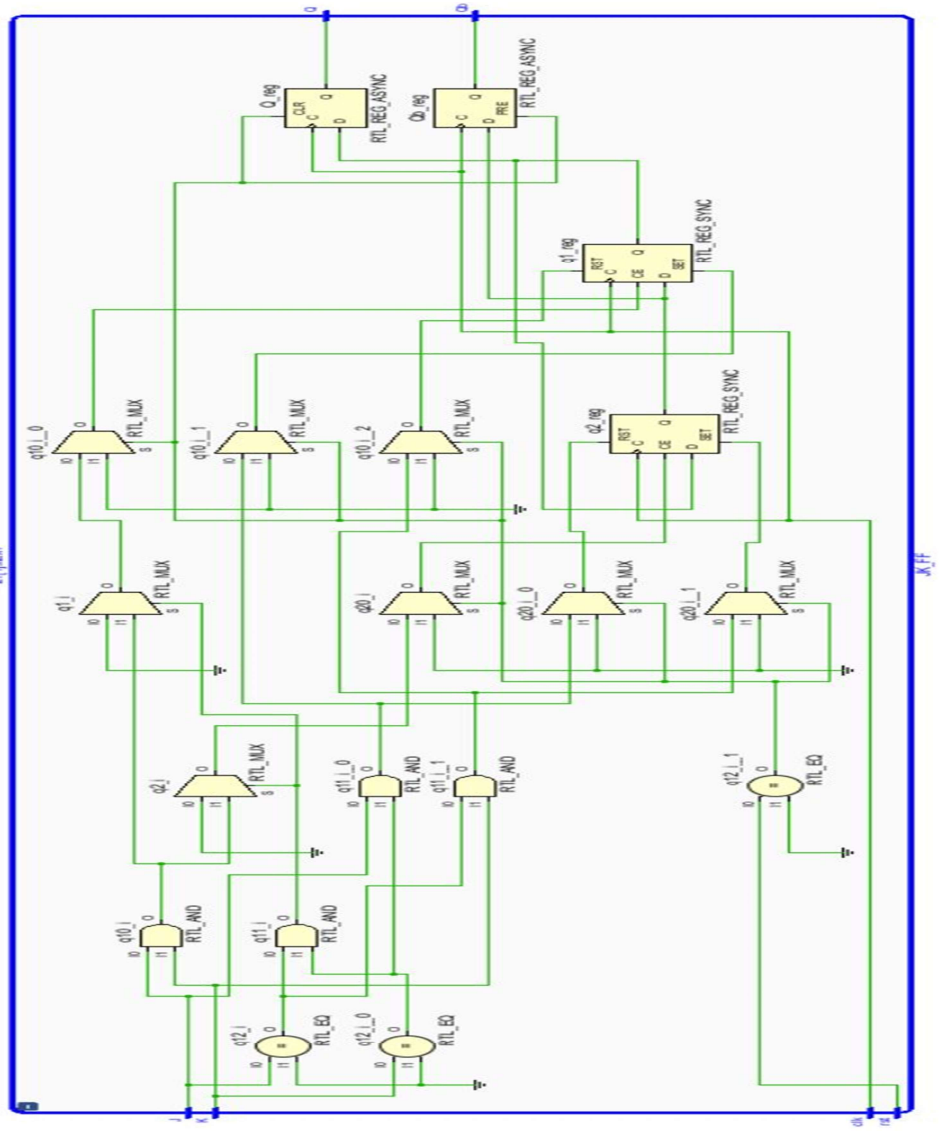
Aim: Write a VHDL Code to Implement 4- bit up-down counter using JK flip flop

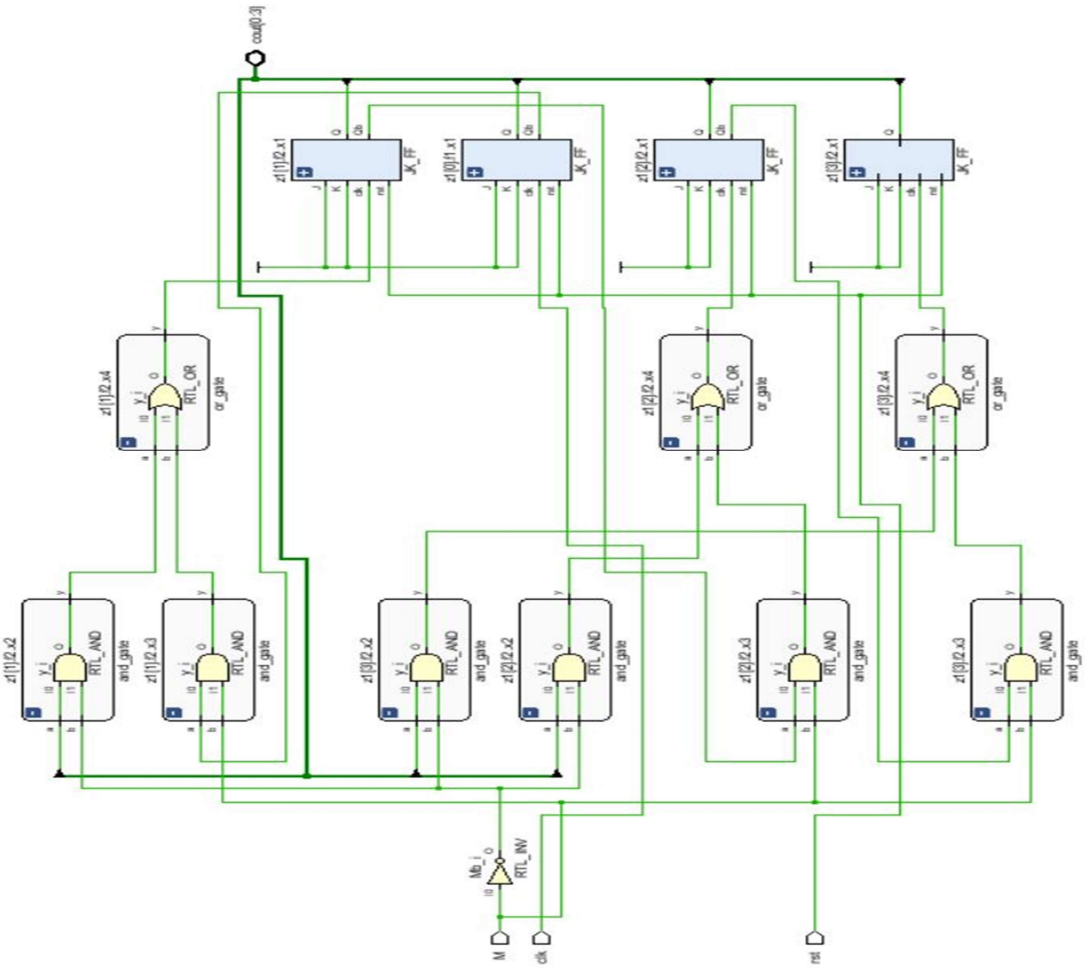
Code:

```
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:

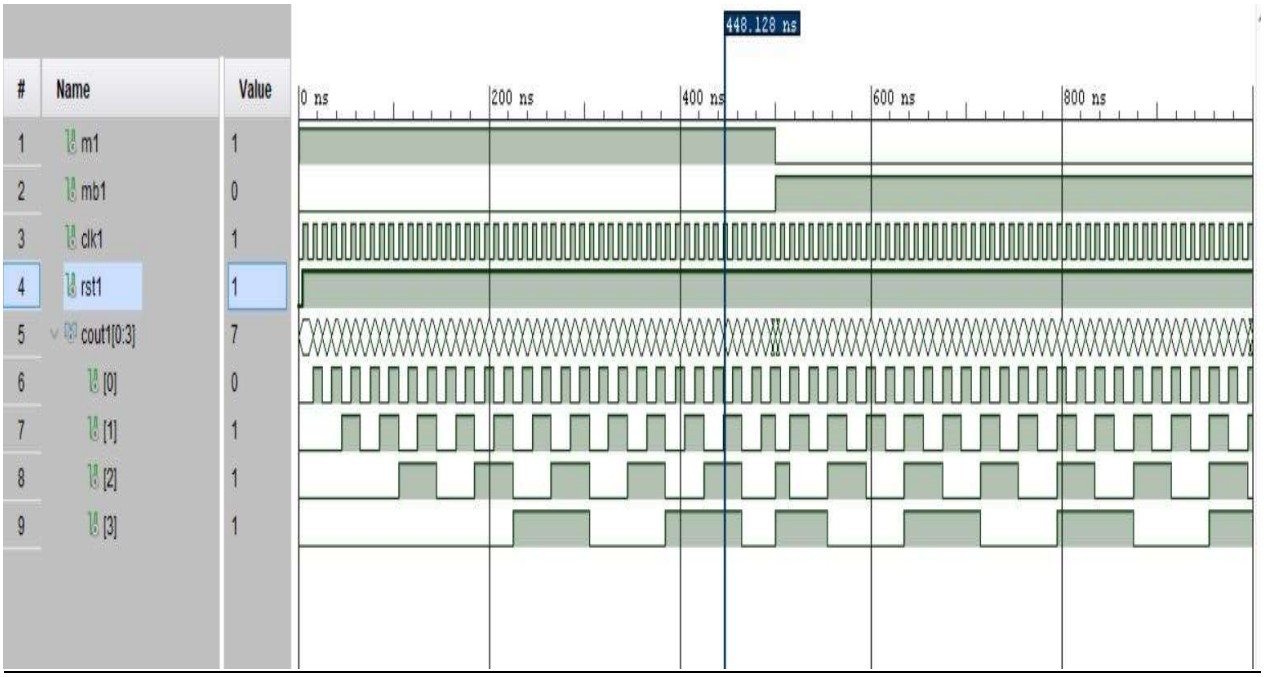




Test bench Code :

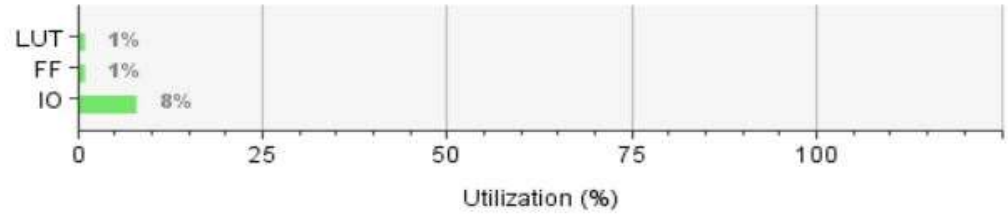
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

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:

Resource	Utilization	Available	Utilization %
LUT	9	17600	0.05
FF	15	35200	0.04
IO	8	100	8.00



Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	∞	3	4	4	M	Mb	5.351	3.752	1.599	∞	input port clock
Path 2	∞	2	2	2	z1[1].f2.x1/Q_reg/C	cout[1]	4.116	3.316	0.800	∞	
Path 3	∞	2	2	2	z1[2].f2.x1/Q_reg/C	cout[2]	4.116	3.316	0.800	∞	
Path 4	∞	2	2	1	z1[3].f2.x1/Q_reg/C	cout[3]	4.116	3.316	0.800	∞	
Path 5	∞	2	2	2	z1[0].f1.x1/Q_reg/C	cout[0]	4.076	3.276	0.800	∞	
Path 6	∞	2	3	9	rst	z1[0].f1.x1/Q_reg/CLR	2.717	1.106	1.611	∞	input port clock
Path 7	∞	2	3	9	rst	z1[0].f1.x1/Qb_reg/PRE	2.717	1.106	1.611	∞	input port clock
Path 8	∞	2	3	9	rst	z1[1].f2.x1/Q_reg/CLR	2.717	1.106	1.611	∞	input port clock
Path 9	∞	2	3	9	rst	z1[1].f2.x1/Qb_reg/PRE	2.717	1.106	1.611	∞	input port clock
Path 10	∞	2	3	9	rst	z1[2].f2.x1/Q_reg/CLR	2.717	1.106	1.611	∞	input port clock

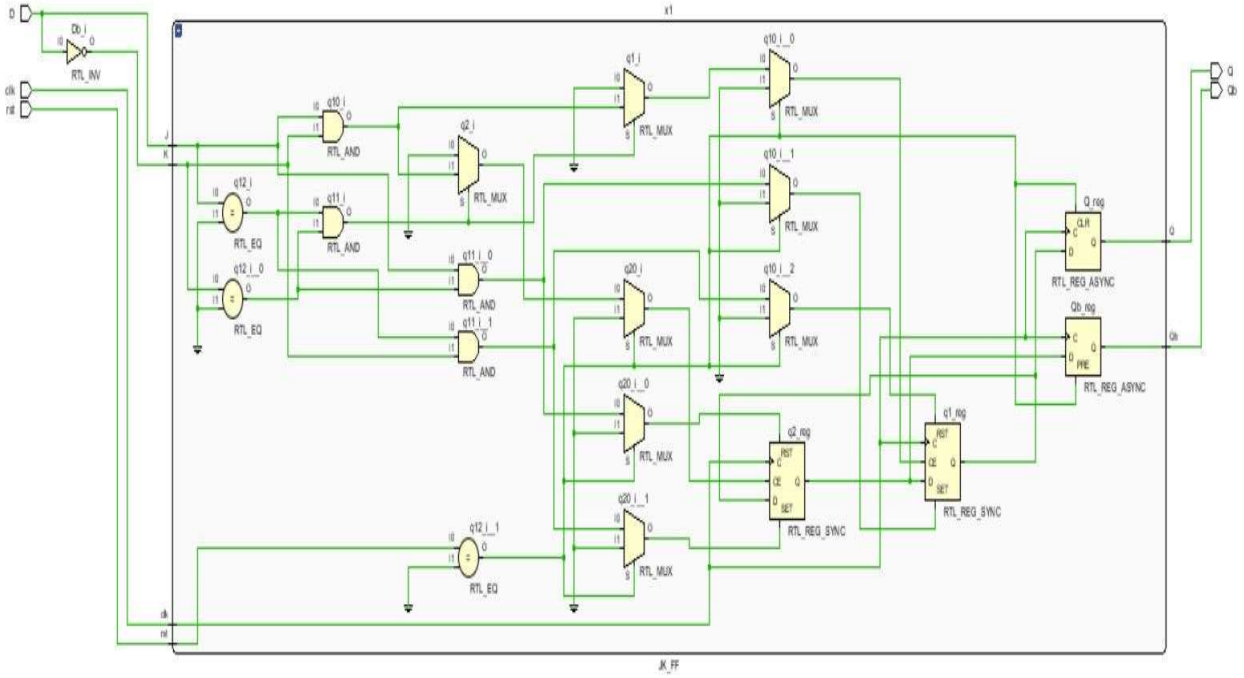
Maximum Combinational Delay: 5.351nSec

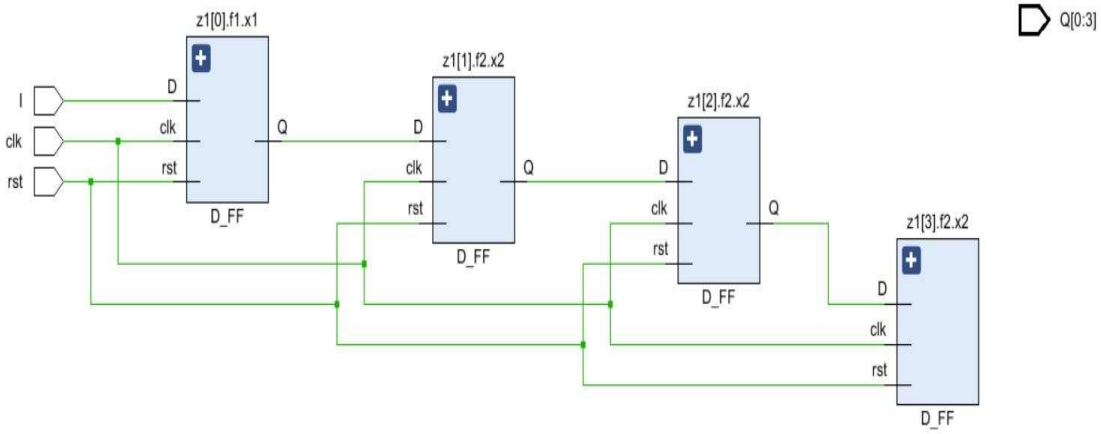
Aim: Write a VHDL Code to implement 4-bit serial in-parallel out shift register

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:

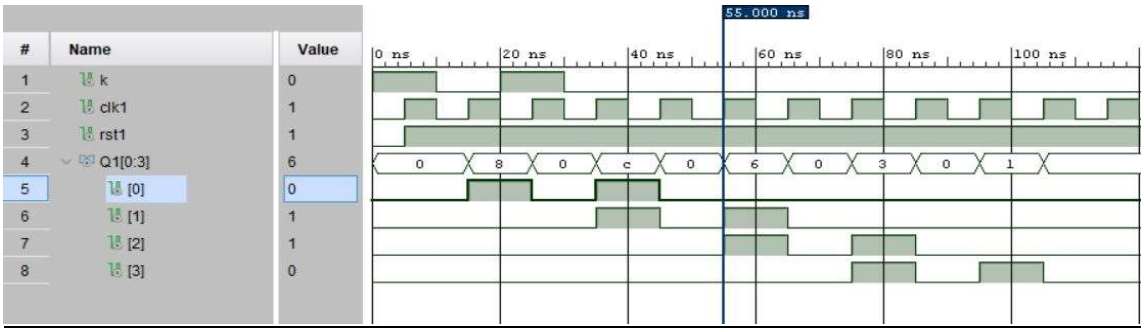

x1
D
clk
rst
Db_i
RTL_INV
J
K
q10_i
RTL_AND
q12_i
RTL_EQ
q11_i
RTL_AND
q2_i
RTL_MUX
q12_i_0
RTL_EQ
q11_i_0
RTL_AND
q11_i_1
RTL_AND
q1_i
RTL_MUX
q20_i
RTL_MUX
q20_i_0
RTL_MUX
q20_i_1
RTL_MUX
q12_i_1
RTL_EQ
q10_i_0
RTL_MUX
q10_i_1
RTL_MUX
q10_i_2
RTL_MUX
q2_reg
RTL_REG_SYNC
q1_reg
RTL_REG_SYNC
Q_reg
RTL_REG_ASYNC
Qb_reg
RTL_REG_ASYNC
Q
Qb
JK_FF


Q[0:3]
z1[0].f1.x1
z1[1].f2.x2
z1[2].f2.x2
z1[3].f2.x2
I
clk
rst
D
clk
rst
Q
D_FF

Test bench Code :

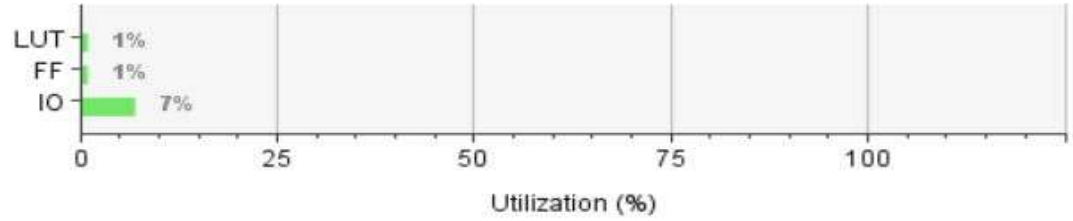
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

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:

Resource	Utilization	Available	Utilization %
LUT	5	17600	0.03
FF	8	35200	0.02
IO	7	100	7.00



Name	Slack ^1	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	∞	2	2	2	z1[0].f1.x1/x1/Q_reg/C	Q[0]	4.076	3.276	0.800	∞	
Path 2	∞	2	2	2	z1[1].f2.x2/x1/Q_reg/C	Q[1]	4.076	3.276	0.800	∞	
Path 3	∞	2	2	2	z1[2].f2.x2/x1/Q_reg/C	Q[2]	4.076	3.276	0.800	∞	
Path 4	∞	2	2	1	z1[3].f2.x2/x1/Q_reg/C	Q[3]	4.076	3.276	0.800	∞	
Path 5	∞	2	3	5	rst	z1[0].f1.x1/x1/Q_reg/CLR	2.706	1.106	1.600	∞	input port clock
Path 6	∞	2	3	5	rst	z1[1].f2.x2/x1/Q_reg/CLR	2.706	1.106	1.600	∞	input port clock
Path 7	∞	2	3	5	rst	z1[2].f2.x2/x1/Q_reg/CLR	2.706	1.106	1.600	∞	input port clock
Path 8	∞	2	3	5	rst	z1[3].f2.x2/x1/Q_reg/CLR	2.706	1.106	1.600	∞	input port clock
Path 9	∞	2	3	1	I	z1[0].f1.x1/x1/q1_reg/D	1.906	1.106	0.800	∞	input port clock
Path 10	∞	2	3	5	rst	z1[1].f2.x2/x1/q1_reg/D	1.906	1.106	0.800	∞	input port clock

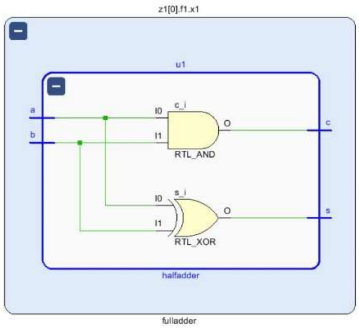
Maximum Combinational Delay: 4.076nSec

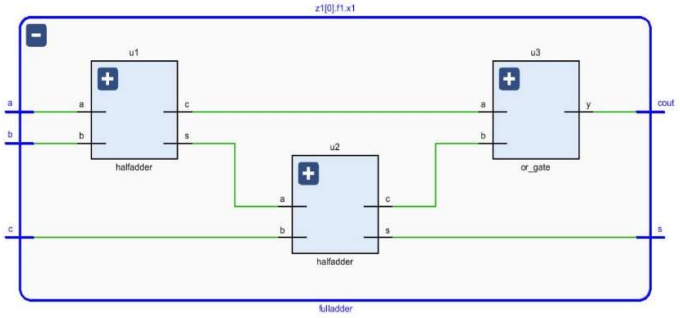
Aim: Write a VHDL Code to implement 4-bit ripple adder using generate statement

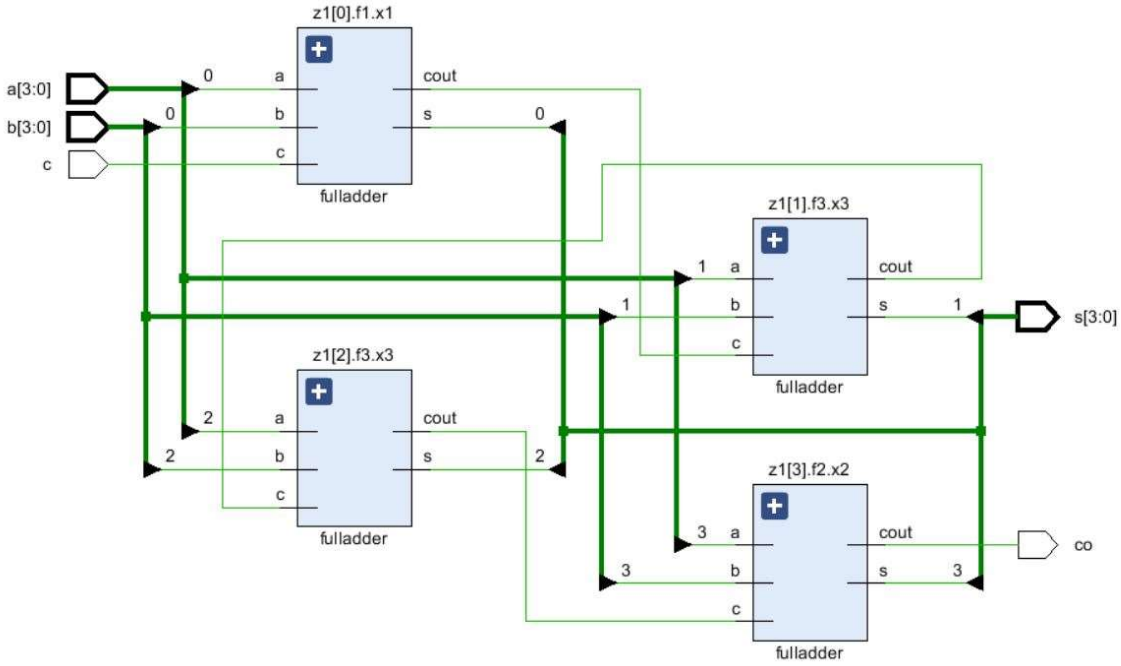
Code:

```
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 :

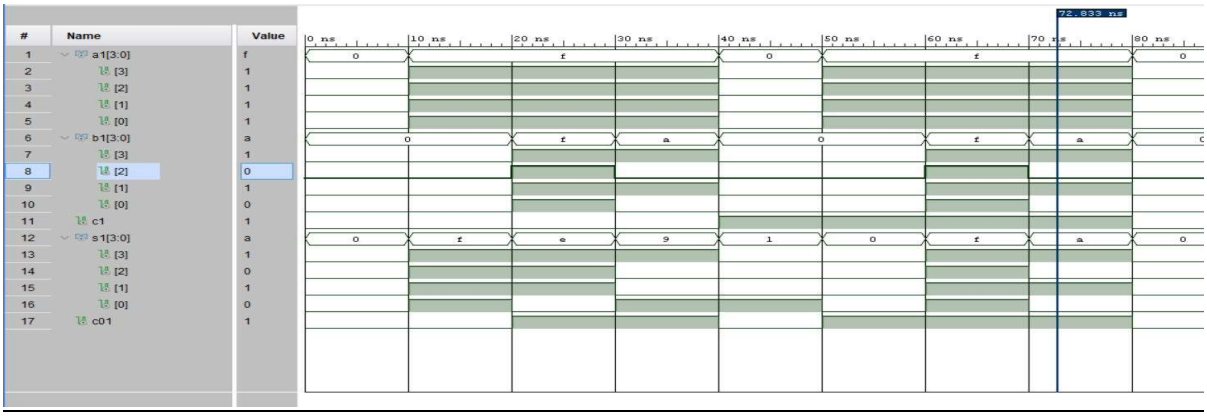
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

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";
```

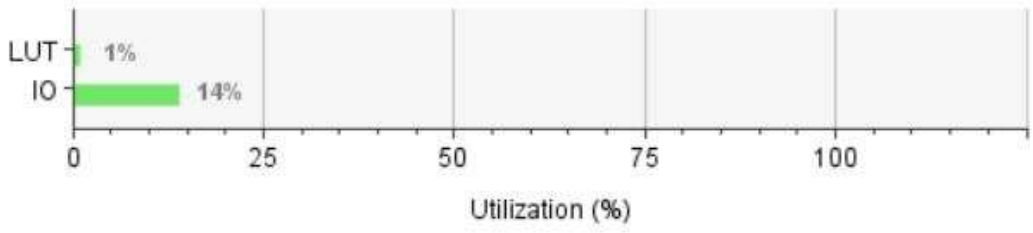
```
b1<="1111";
wait for 10ns;

a1<="1111";
b1<="1010";
wait for 10ns;
end process;

end Behavioral;
```

SIMULATION WAVEFORM :



SYNTHESIS SUMMARY:

Resource	Utilization	Available	Utilization %
LUT	4	17600	0.02
IO	14	100	14.00



Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	∞	4	5	3	a[1]	co	5.942	3.876	2.066	∞	input port clock
Path 2	∞	4	5	3	a[1]	s[2]	5.942	3.876	2.066	∞	input port clock
Path 3	∞	4	5	3	a[1]	s[3]	5.936	3.870	2.066	∞	input port clock
Path 4	∞	3	4	3	b[0]	s[1]	5.379	3.780	1.599	∞	input port clock
Path 5	∞	3	4	3	b[0]	s[0]	5.351	3.752	1.599	∞	input port clock

Maximum Combinational Delay: 5.942nSec