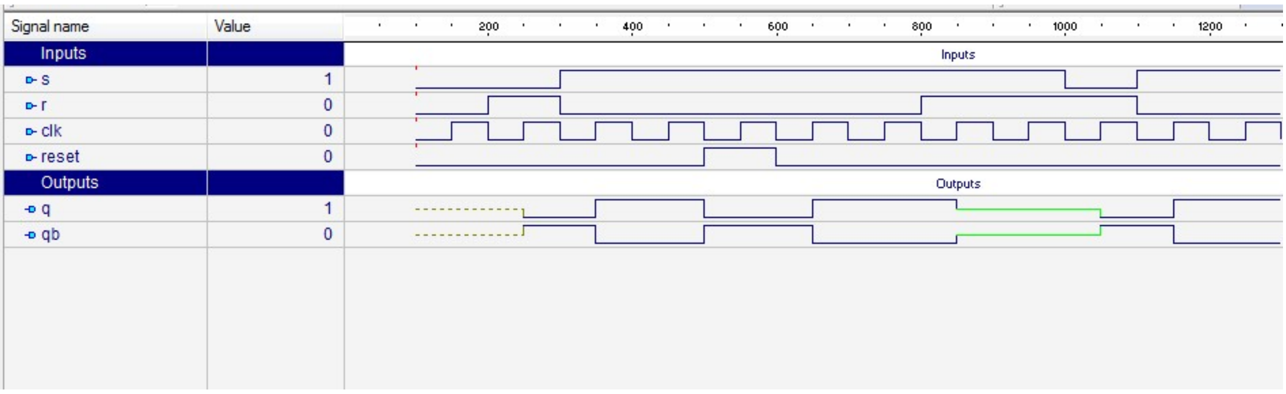
VHDL Code for SR Flip Flop:-

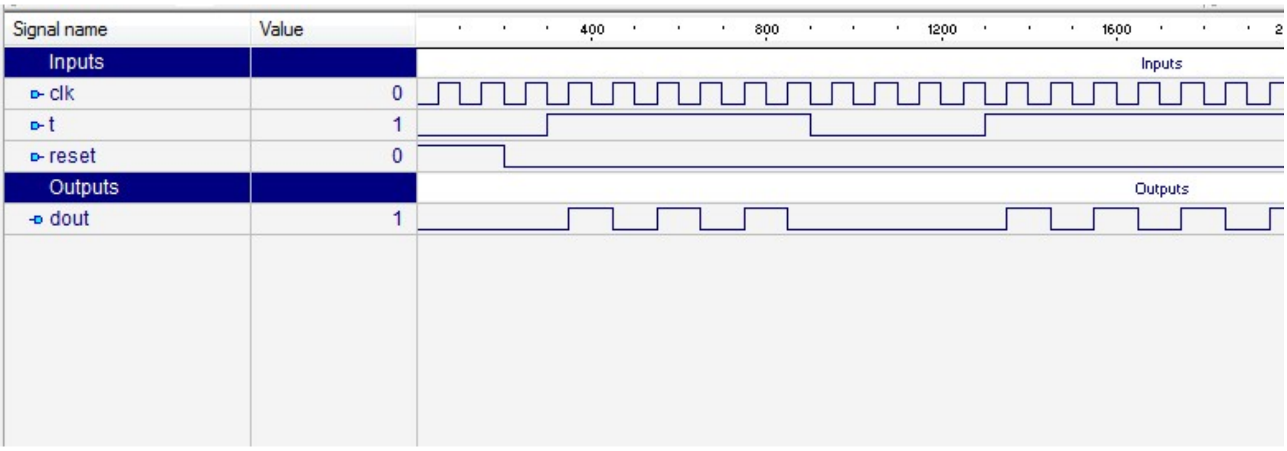
```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity sr_flip_flop is
     port(
         s : in STD_LOGIC;
         r : in STD_LOGIC;
         clk : in STD_LOGIC;
         reset : in STD_LOGIC;
         q : out STD_LOGIC;
         qb : out STD_LOGIC
         );
end sr_flip_flop;

architecture sr_flip_flop_arc of sr_flip_flop is
begin

    srff : process (s,r,clk,reset) is
    begin
        if (reset='1') then
            q <= '0';
            qb <= '1';
        elsif (rising_edge (clk)) then
            if (s/=r) then
                q <= s;
                qb <= r;
            elsif (s='1' and r='1') then
                q <= 'Z';
                qb <= 'Z';
            end if;
        end if;
    end process srff;


end sr_flip_flop_arc;
```



Signal name	Value
Inputs	
s	1
r	0
clk	0
reset	0
Outputs	
q	1
qb	0

VHDL Code for D Flip Flop:-

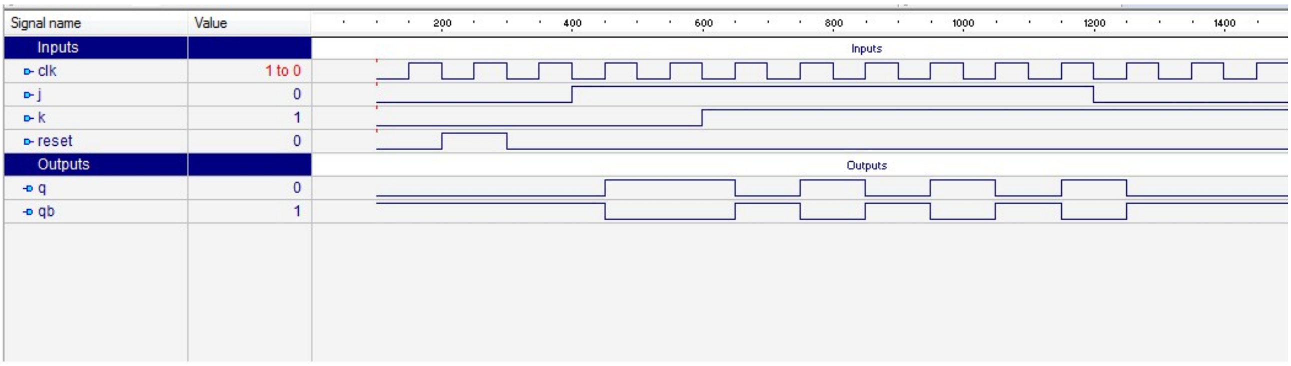
```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity tff_using_dff is
     port(
         clk : in STD_LOGIC;
         t : in STD_LOGIC;
         reset : in STD_LOGIC;
         dout : out STD_LOGIC
         );
end tff_using_dff;

architecture tff_using_dff_arc of tff_using_dff is

component d_flip_flop is
     port(
         clk : in STD_LOGIC;
         din : in STD_LOGIC;
         reset : in STD_LOGIC;
         dout : out STD_LOGIC
         );
end component d_flip_flop;

signal ip : std_logic;
signal op : std_logic;

begin

    ip <= op xor t ;
    u0 : d_flip_flop port map (clk => clk,
                        din => ip,
                        reset => reset,
                        dout => op);

    dout <= op;


end tff_using_dff_arc;
```



Signal name	Value
Inputs	
clk	0
t	1
reset	0
Outputs	
dout	1

VHDL Code for JK Flip Flop:-

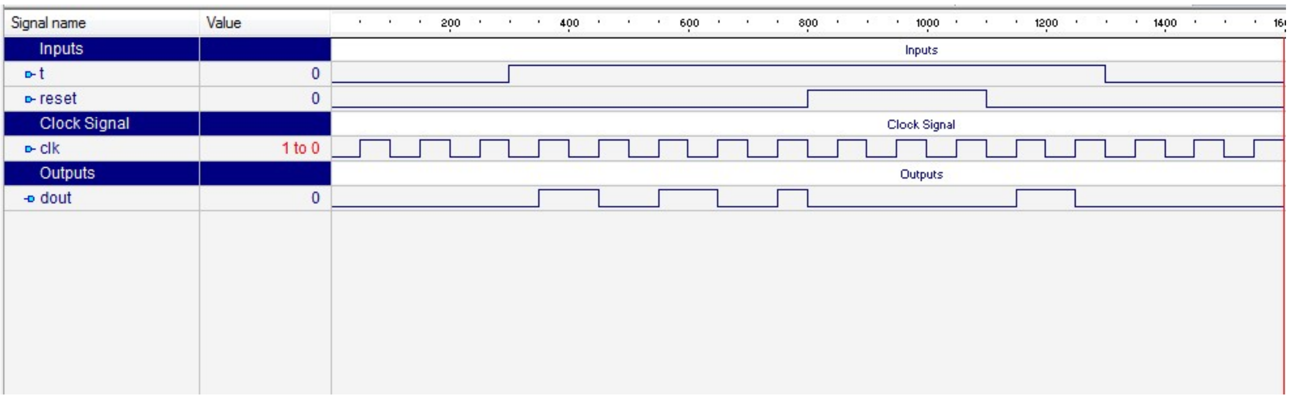
```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity jk_flip_flop is
     port(
         j : in STD_LOGIC;
         k : in STD_LOGIC;
         clk : in STD_LOGIC;
         reset : in STD_LOGIC;
         q : out STD_LOGIC;
         qb : out STD_LOGIC
         );
end jk_flip_flop;

architecture jk_flip_flop_arc of jk_flip_flop is
begin

    jkff : process (j,k,clk,reset) is
    variable m : std_logic := '0';
    begin
        if (reset='1') then
            m := '0';
        elsif (rising_edge (clk)) then
            if (j/=k) then
                m := j;
            elsif (j='1' and k='1') then
                m := not m;
            end if;
        end if;
        q <= m;
        qb <= not m;
    end process jkff;


end jk_flip_flop_arc;
```



Signal name	Value
Inputs	
clk	1 to 0
j	0
k	1
reset	0
Outputs	
q	0
qb	1

VHDL Code for T Flip Flop:-

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity Toggle_flip_flop is
     port(
         t : in STD_LOGIC;
         clk : in STD_LOGIC;
         reset : in STD_LOGIC;
         dout : out STD_LOGIC
         );
end Toggle_flip_flop;

architecture toggle_flip_flop_arc of Toggle_flip_flop is
begin

    tff : process (t,clk,reset) is
    variable m : std_logic := '0';
    begin
        if (reset='1') then
            m := '0';
        elsif (rising_edge (clk)) then
            if (t='1') then
                m := not m;
            end if;
        end if;
        dout <= m;
    end process tff;


end toggle_flip_flop_arc
```


Signal name
Value
200
400
600
800
1000
1200
1400
Inputs
Inputs
t
0
reset
0
Clock Signal
Clock Signal
clk
1 to 0
Outputs
Outputs
dout
0