EXPERIMENT No.: - 10

Simulation of VHDL Models.

TITLE:

Implementation of SR flip-flop in VHDL using Behavioural Modelling.

OBJECTIVE:

To implement the Behavioural model of SR flip-flop in ISE Simulator.

THEORY:

SR flip flop is one of the fundamental sequential circuit possible. This simple flip-flop is basically a one-bit memory storage device that has two inputs, one which will ‘SET’ the device (i.e., the output is 1), and is labelled S and other which will reset the device (i.e., the output is 0), labelled R. Thus, the name SR stands for “SET-RESET”.

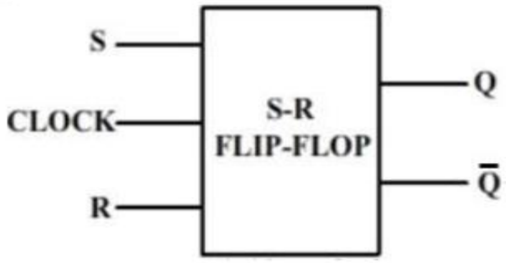
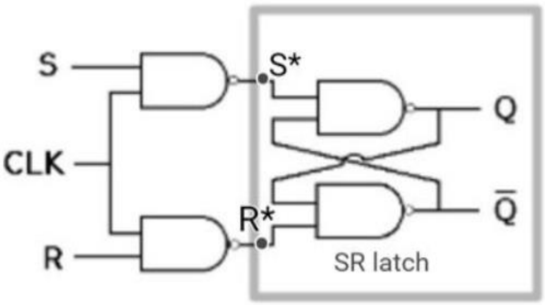


Fig.1 Symbol for SR flip flop



Since the SR flip-flop can be constructed by using clock transition, therefore we will provide a clock to SR flip-flop.

Thus, we can say

S* = (S . CLOCK)’ = S’ . CLOCK’

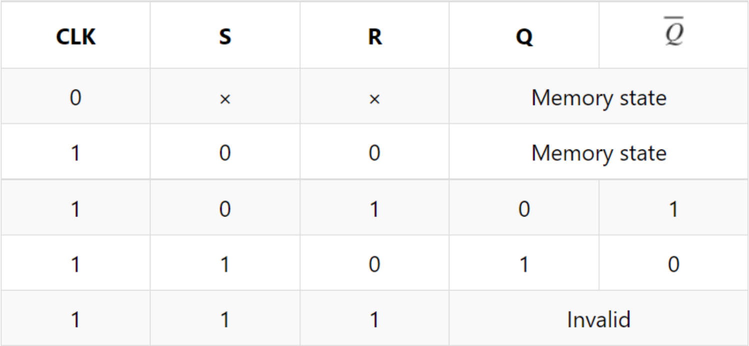
R* = (R . CLOCK)’ = R’ . CLOCK’

CASE 1: - Now if CLOCK is 0 then S*=1 and R*=1 and here S and R will be treated as don’t care conditions, then we get Q and Q’ in a memory state that is holding previous values.

CASE 2: - If CLOCK is 1 then S*=S’ and R*=R’, now there will be 4 more cases depending upon the values of S and R

a.) S=0 and R=0 then S* and R* both becomes 1 and we get outputs Q and Q’ holding memory state.
b.) S=0 and R=1 then S*=1 and R*=0 then we get Q=0 and Q’=1, we get both outputs as complement of each other.
c.) S=1 and R=0 then S*=0 and R*=1, then we get Q=1 and Q’=0.
d.) S=1 and R=1 then S*=0 and R*=0 then we get Q and Q’ in the invalid state, i.e., not used condition

The truth table of SR flip-flop is given below.



CLK	S	R	Q	$\overline{Q}$
0	×	×	Memory state	
1	0	0	Memory state	
1	0	1	0	1
1	1	0	1	0
1	1	1	Invalid	

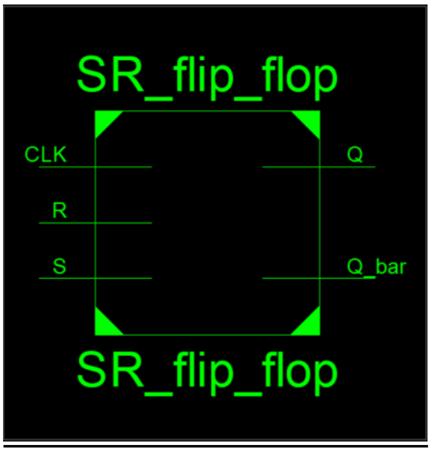
SOURCE CODE: -

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity SR_flip_flop is
   Port ( S : in  STD_LOGIC;
          R : in  STD_LOGIC;
          CLK : in  STD_LOGIC;
                              Q_bar : out  STD_LOGIC;
          Q : out  STD_LOGIC);
end SR_flip_flop;

architecture Behavioral of SR_flip_flop is
```

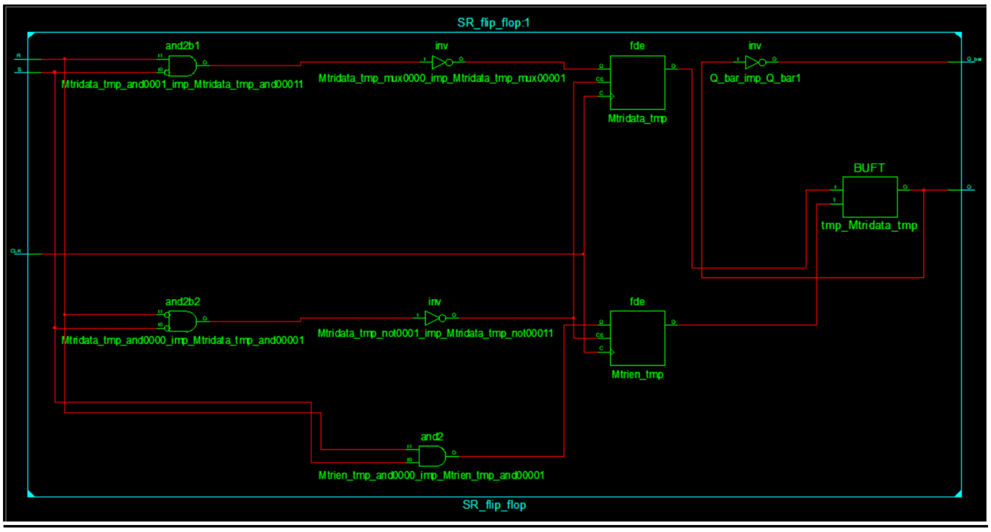
```
begin
    PROCESS(CLK)
        variable tmp: std_logic;
        begin
            if(CLK='1' and CLK' event) then
                if(S='0' and R='0')then
                    tmp:=tmp;
                    elsif(S='1' and R='1')then
                    tmp:='Z';
                    elsif(S='0' and R='1')then
                    tmp:='0';
                    else
                    tmp:='1';
                end if;
            end if;
            Q <= tmp;
            Q_bar <= not tmp;
        end PROCESS;

end Behavioral;
```

OUTPUT: -

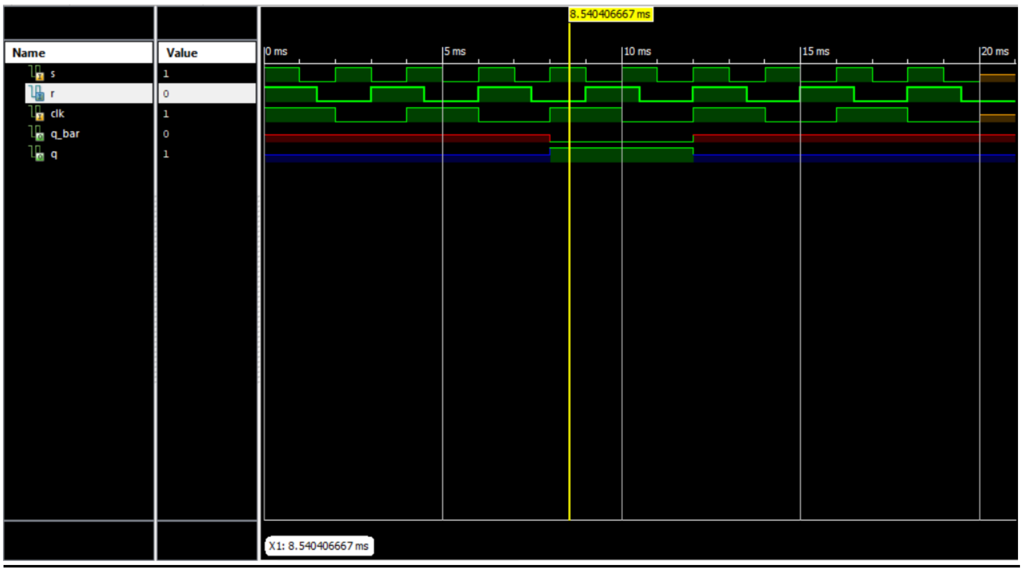
ENTITY DIAGRAM: -



DETAILED ENTITY DIAGRAM: -


SR_flip_flop:1
and2b1
Mtridata_tmp_and0001_imp_Mtridata_tmp_and00011
inv
Mtridata_tmp_mux0000_imp_Mtridata_tmp_mux00001
fde
Mtridata_tmp
inv
Q_bar_imp_Q_bar1
BUFT
tmp_Mtridata_tmp
and2b2
Mtridata_tmp_and0000_imp_Mtridata_tmp_and00001
inv
Mtridata_tmp_not0001_imp_Mtridata_tmp_not00011
fde
Mtrien_tmp
and2
Mtrien_tmp_and0000_imp_Mtrien_tmp_and00001
SR_flip_flop

SIMULATION: -



Name	Value
s	1
r	0
clk	1
q_bar	0
q	1

RESULT:

Thus, we have implemented the Behavioural Model of SR flip flop in VHDL, and the Entity Diagram and RTL are shown above. Also, the flip flop’s truth table has been verified using the simulation of the waveform in ISE Simulator.

CONCLUSION:

In this experiment, we learnt about how to implement the Behavioural model of SR flip flop in VHDL and also verified the truth table of this flip flop using the waveform generated by ISE Simulator.