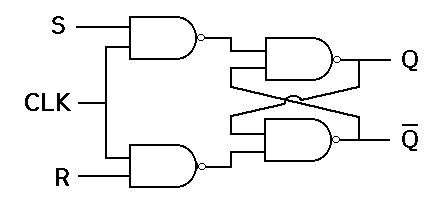
**EXPERIMENT – 6**

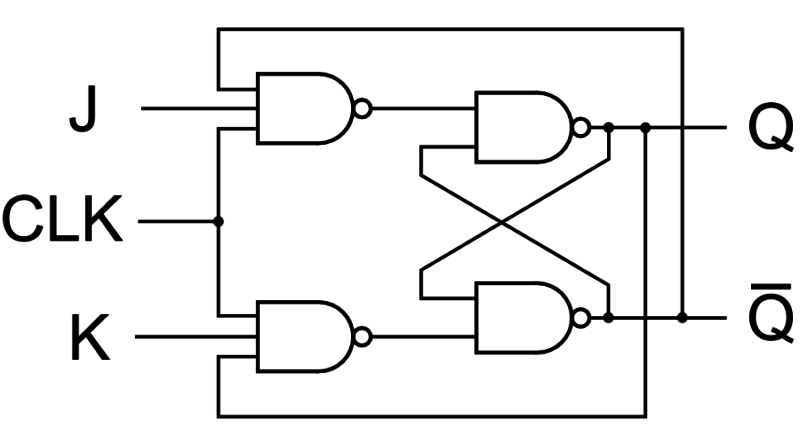
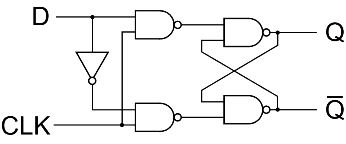
**AIM -** WRITE VHDL CODE FOR FLIP-FLOP AND SIMULATE IT USING MODELSIM.

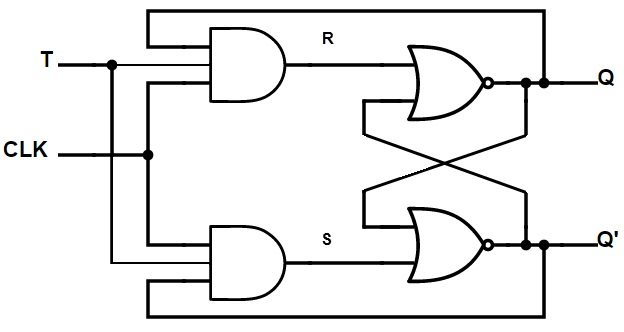
**THEORY** - A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems. Flip-flops and latches are used as data storage elements. It is the basic storage element in sequential logic. But first, let’s clarify the difference between a latch and a flip-flop.

There are 4 types of flip-flop:

1. SR FLIP-FLOP
2. JK FLIP-FLOP
3. T FLIP-FLOP
4. D FLIP-FLOP







**VHDL CODE :**

**D FLIPFLOP**

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_arith.all;

ENTITY D\_FLIPFLOP IS

PORT( D,CLOCK: in std\_logic; Q: out std\_logic);

END ENTITY D\_FLIPFLOP;

--

ARCHITECTURE D\_FLIPFLOP\_BEHAV OF D\_FLIPFLOP IS

BEGIN

process(CLOCK)

begin

if(rising\_edge(CLOCK)) then

Q <= D;

end if;

end process;

END ARCHITECTURE D\_FLIPFLOP\_BEHAV;

**JK FLIPFLOP**

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_arith.all;

ENTITY JK\_FLIPFLOP IS

port( J,K,CLOCK: in std\_logic; Q,QBAR: out std\_logic);

END ENTITY JK\_FLIPFLOP;

--

ARCHITECTURE JK\_FLIPFLOP\_BEHAV OF JK\_FLIPFLOP IS

BEGIN

process(CLOCK)

begin

if (rising\_edge(CLOCK)) then

if(J/=K) then

Q <= J;

QBAR <= not J;

elsif(J='1' and K='1') then

Q <= not J;

QBAR <= J;

end if;

end if;

end process;

END ARCHITECTURE JK\_FLIPFLOP\_BEHAV;

**SR FLIPFLOP**

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_arith.all;

ENTITY SR\_FLIPFLOP IS

port(S,R,CLOCK: in std\_logic; Q,QBAR: out std\_logic);

END ENTITY SR\_FLIPFLOP;

ARCHITECTURE SR\_FLIPFLOP\_BEHAV OF SR\_FLIPFLOP IS

BEGIN

process(CLOCK)

begin

if(rising\_edge(CLOCK)) then

if(S/=R) then

Q<=S;

QBAR<=R;

elsif (S='1' and R='1') then

Q<='Z';

QBAR<='Z';

end if;

end if;

end process;

END ARCHITECTURE SR\_FLIPFLOP\_BEHAV;

**T FLIPFLOP**

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_arith.all;

ENTITY T\_FLIPFLOP IS

port( T,CLOCK: in std\_logic; Q: out std\_logic);

END ENTITY T\_FLIPFLOP;

ARCHITECTURE T\_FLIPFLOP\_BEHAV OF T\_FLIPFLOP IS

BEGIN

process(CLOCK)

begin

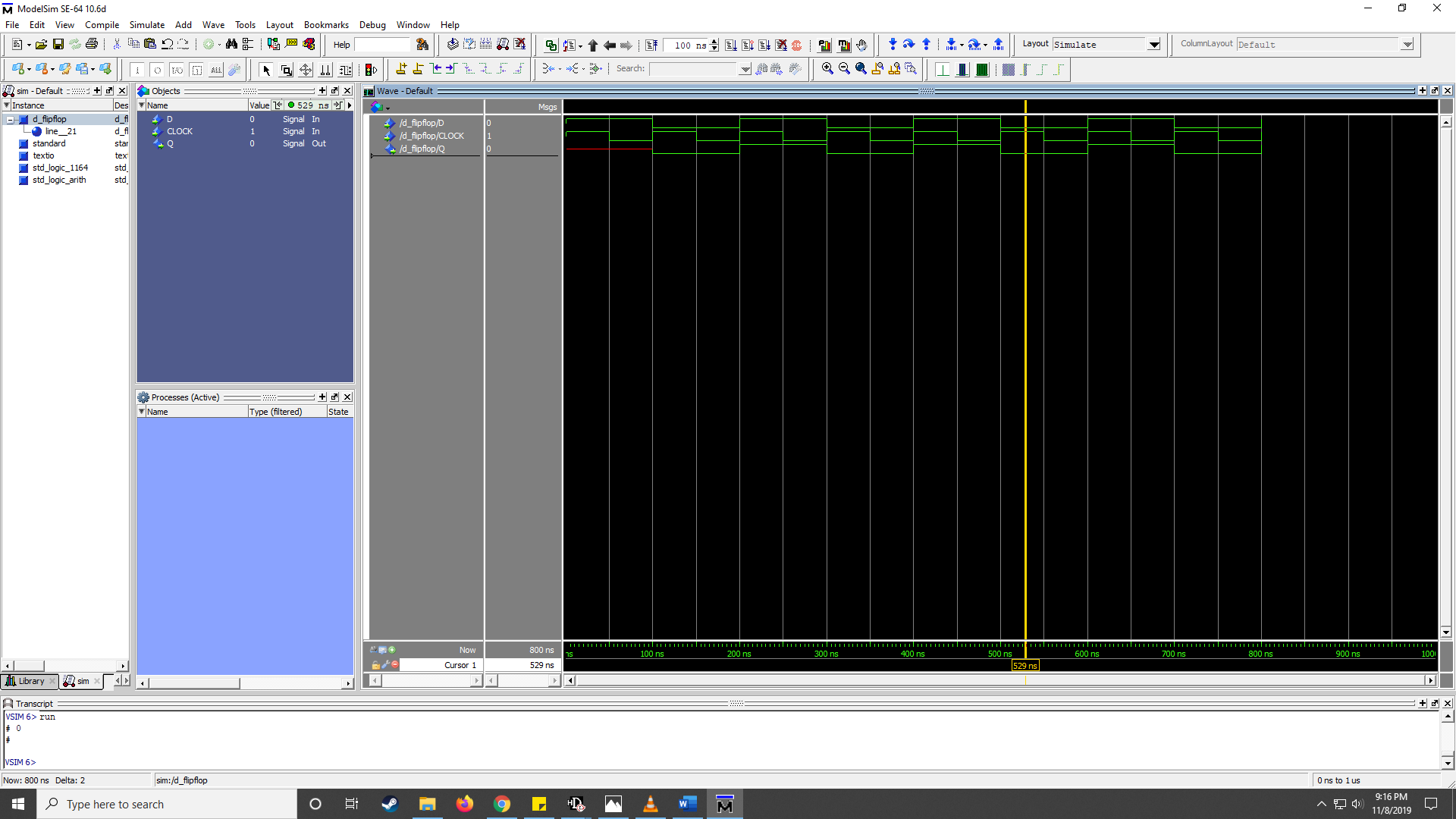
if(rising\_edge(CLOCK)) then

Q<=not T;

end if;

end process;

END ARCHITECTURE T\_FLIPFLOP\_BEHAV;

**RESULT:**

