S. B. JAIN INSTITUTE OF TECHNOLOGY,MANAGEMENT & RESEARCH,NAGPUR.

Practical No. 03

Aim: Design and Simulation of Flip Flops and verify it using test bench

Name of Student	**: Aadesh Motghare**
Roll No.	**: 41(ET20065)**
Semester/Year	**: 6th Sem/3rd Year**
Academic Session	**: 2022-23**
Date of Performance	**:**
Date of Submission	**:**

AIM: Design and Simulation of flip flops and verify it using test bench

OBJECTIVE:

- To verify the functionality of all Flips Flops
- To build and test various sequential logic circuits.

SOFTWARE: - Xilinx ISE9.1.

THEORY:-

Flips flops are indispendable building blocks for sequential circuits. Latches are also used occasionally in sequential circuits.The fundamental difference between latches and flip flops is that while latches are level sensitive ,flip flops are edge sensitive. This means that a latch is transparent during the whole time in which clock is ‘1’(positive level sensitive) and opaque when the clock is ‘0’(vice versa for negative level sensitive latch), while a flip flop is transparent only during one of the clock transitions ,either from ‘0’ to ‘1 ’ called positive edge triggered or from ‘1’to ‘0’ called negative edge triggered.

There are 4 types of flip flops

- SR flip flop(SRFF)
- JK flip flop(JKFF)
- D flip flop(DFF)
- T flip flop(TFF)

S R FLIP FLOP

A positive edge triggered SR FF with clock clk is shown in figures below. Fig 1.a shows circuit symbol and fig 1.b shows the circuit diagram.

Flip flop has 2 outputs Q and Q bar. The output Q follows the SR input according to the truth table given below when there is a positive edge at the clock input . The output Qbar is complement of Q.

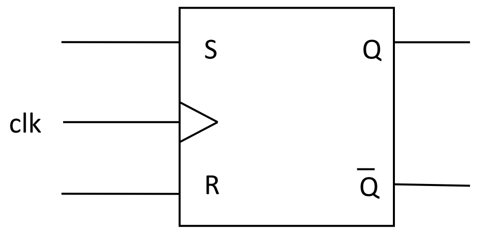


Fig 1 a. S R FLIP FLOP circuit symbol

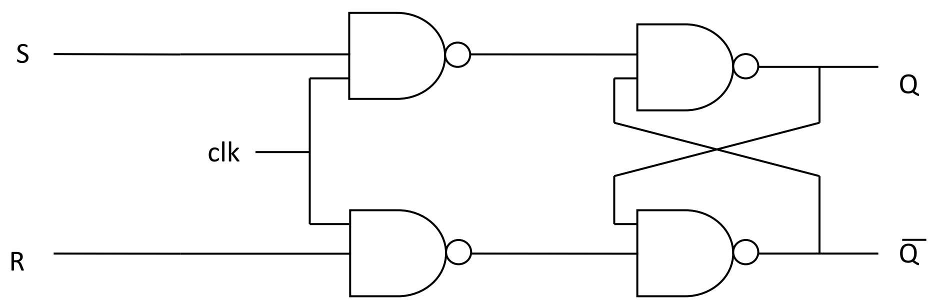


Fig 1 b. S R FLIP FLOP circuit diagram

Truth Table

Clk	**S**	**R**	**Q(n+1)**
-	**X**	**X**	**Qn**
↑	**0**	**0**	**Qn**
↑	**0**	**1**	**0**
↑	**1**	**0**	**1**
↑	**1**	**1**	**Invalid**

Table 1: Truth Table of S R flip flop

J K FLIP FLOP

A positive edge triggered J K flip flop with clock clk is shown below.It has 2 outputs Q and $\overline{Q}$ which vary depending on the combination of J and K as per truth table shown in table 2.

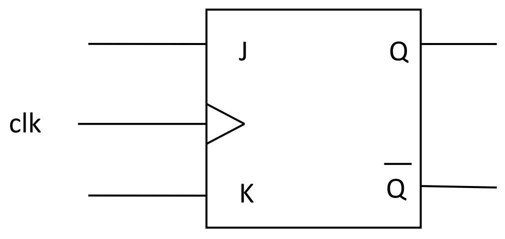


Fig 2.a Circuit symbol for J K flip flop

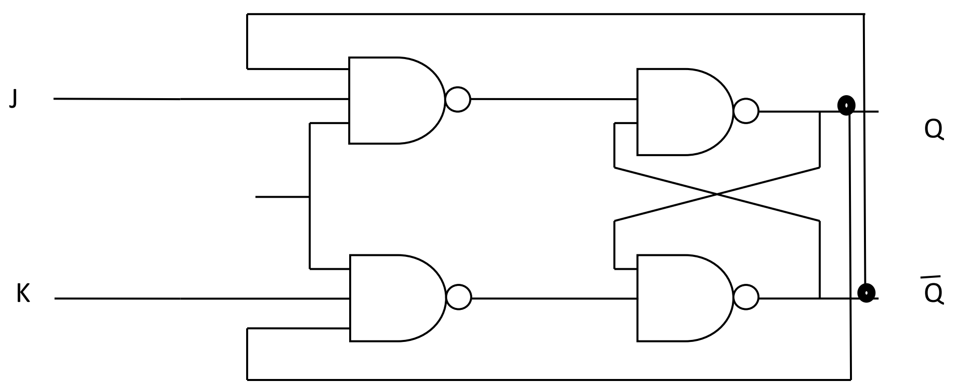


Fig 2.b Circuit diagram for J K flip flop

Truth Table

Clk	J	K	Q(n+1)
-	X	X	Qn
↑	0	0	Qn
↑	0	1	0
↑	1	0	1
↑	1	1	$\overline{Qn}$

Table 2: Truth Table of J K flip flop

D FLIP FLOP

A positive edge triggered D FF with clock clk is shown in Figure 1.

Flip flop has 2 outputs Q and Q bar.The output Q follows the D input when there is a positive edge at the clock input . The output Qbar is not of Q.

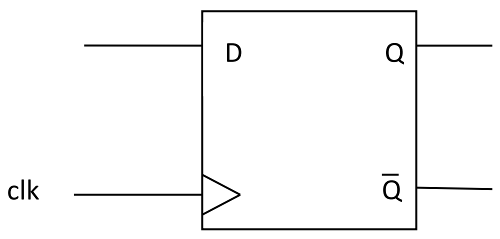


Fig 3.a D FLIP FLOP circuit symbol

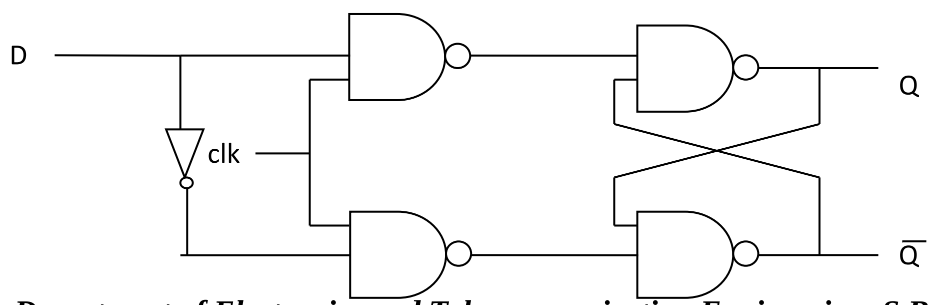


Fig 3.b D FLIP FLOP circuit diagram

Truth Table

Clk	**D**	**Q(n+1)**
-	**x**	**Qn**
↑	**0**	**0**
↑	**1**	**1**

Table 3: Truth Table of D flip flop

T flip flop

A positive edge triggered T FF with clock clk is shown in fig 4 below.

Flip flop has 2 outputs Q and Q bar.The output Q follows the T input as per the truth table 4 when there is a positive edge at the clock input . The output Qbar is complement of Q.

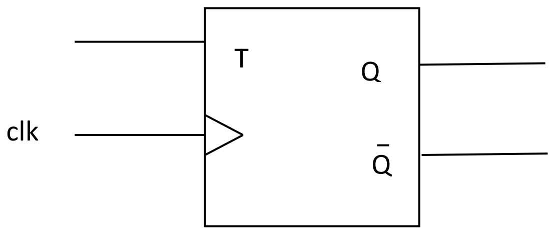


Fig 4. Circuit Symbol for T flip flop

Truth Table

clk	**T**	**Q(n+1)**
--	x	Qn
↑	0	Qn
↑	1	$\bar{Q}$n

Table 4: Truth Table of T flip flop

VHDL Code

STEPS FOR PROGRAM:-

Step 1.Library /Package Declaration : Involves declaration of all libraries and respective packages used in the design.

```
LIBRARY library_name;

USE library_name.package_name.all;
```

Step 2.Entity:

```
    ENTITY entity_name is

            PORT(signal_name(s): mode signal_type;

                    signal_name(s): mode signal_type;
                    …);
    end ENTITY entity_name;
```

Signals of the same mode and signal_type can be grouped on 1 line**.**

MODE describes the direction data is transferred through port

- in – data flows into the port
- out – data flows out of port *only*
- buffer – data flows out of port *as well as read* internally.
- inout – bi-directional data flow into and out of port

SIGNAL_TYPE defines the data type for the signal(s)

- bit – single signals that can have logic values 0 and 1.
- bit_vector – bus signals(vector form of bit) that can have logic values 0 and 1.

- std_logic – part of std_logic_1164 package of IEEE library. Used to represent 2 value logical values i.e 0 and 1 as well as other values such as high impedance ,don't care and others as described below.
- std_logic_vector – bus signals (vector form of std_logic) but IEEE standard for simulation and synthesis note that all vectors must have a range specified example for a 4 bit bus: bit_vector (3 downto 0) or std_logic_vector (3 downto 0).

In order to use std_logic and std_logic_vector we must include the library and package usage declarations in the VHDL model before the entity statement as follows:
LIBRARY ieee;
USE ieee.std_logic_1164.all;

Values for std-logic:

U	un-initialized (undefined logic value)
X	forced unknown logic value
0	Logic low
1	Logic High
Z	high impedance (tri-state)
W	weak unknown
L	weak 0
H	weak 1
-	don't care value (for synthesis minimization)

Step 3: Architecture Declaration

```
architecture architecture name of entity_name
  architecture_declarative_part;
begin
Statements;
 end architecture_name;
```

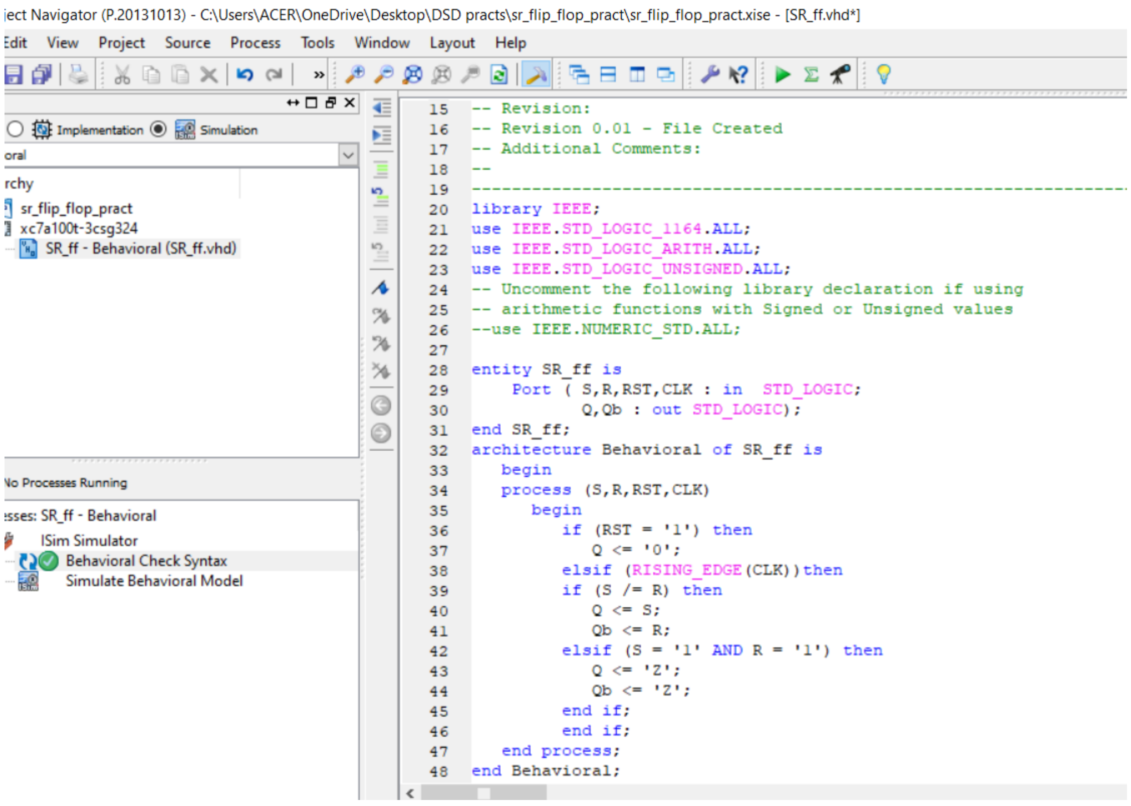
Here we should specify the entity name for which we are writing the architecture body. The architecture statements should be inside the begin and end keyword. Architecture declarative part may contain variables, constants, or component declaration.

Step 4: Simulate the VHDL code and remove the syntax

errors if any.

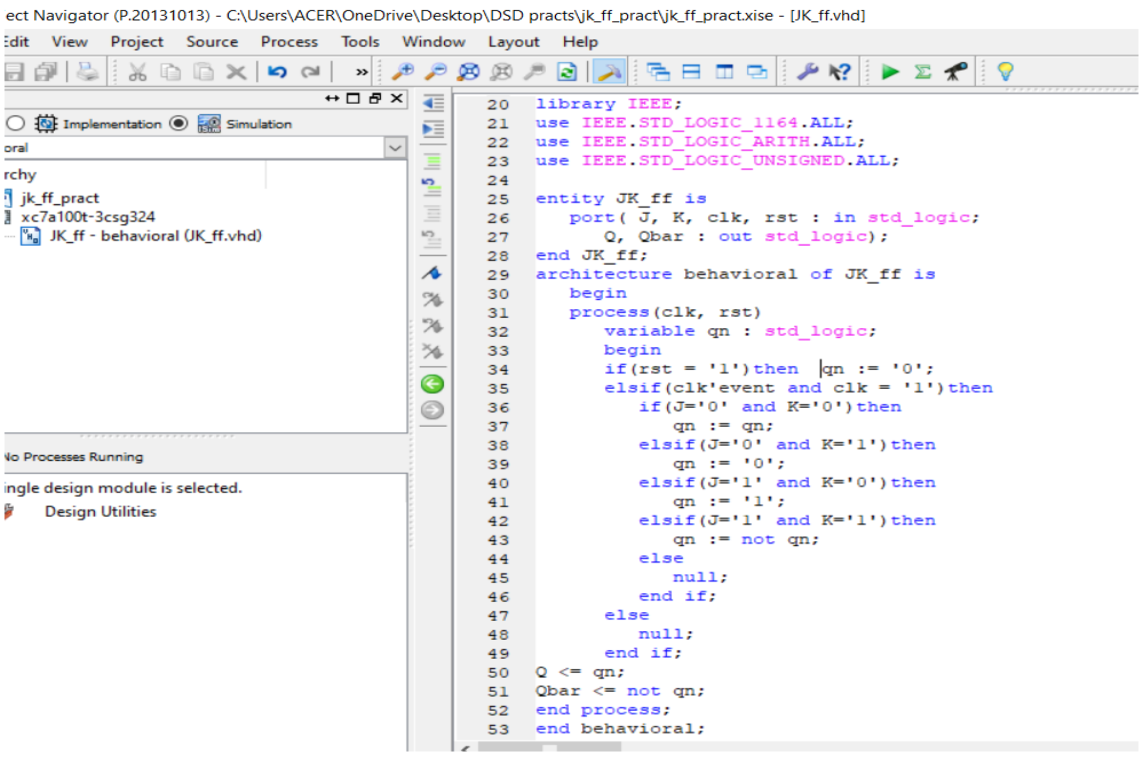
Step 5: Write the testbench and verify the design

CODE: SR flip flop


ject Navigator (P.20131013) - C:\Users\ACER\OneDrive\Desktop\DSD practs\sr_flip_flop_pract\sr_flip_flop_pract.xise - [SR_ff.vhd*]
Edit View Project Source Process Tools Window Layout Help
Implementation
Simulation
oral
rchy
sr_flip_flop_pract
xc7a100t-3csg324
SR_ff - Behavioral (SR_ff.vhd)
No Processes Running
sses: SR_ff - Behavioral
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

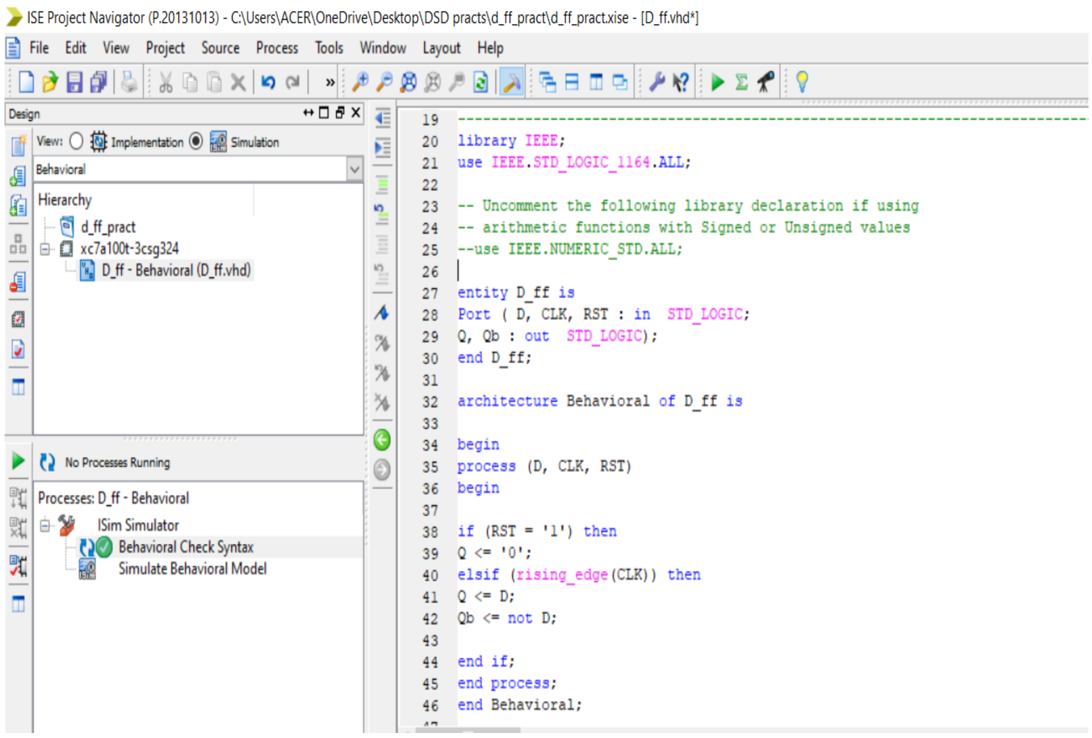
```
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

entity SR_ff is
    Port ( S,R,RST,CLK : in  STD_LOGIC;
           Q,Qb : out STD_LOGIC);
end SR_ff;
architecture Behavioral of SR_ff is
   begin
   process (S,R,RST,CLK)
      begin
         if (RST = '1') then
            Q <= '0';
         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;
end Behavioral;
```

2) JK Flip-Flop


ect Navigator (P.20131013) - C:\Users\ACER\OneDrive\Desktop\DSD practs\jk_ff_pract\jk_ff_pract.xise - [JK_ff.vhd]
Edit View Project Source Process Tools Window Layout Help
Implementation
Simulation
oral
rchy
jk_ff_pract
xc7a100t-3csg324
JK_ff - behavioral (JK_ff.vhd)
No Processes Running
ingle design module is selected.
Design Utilities

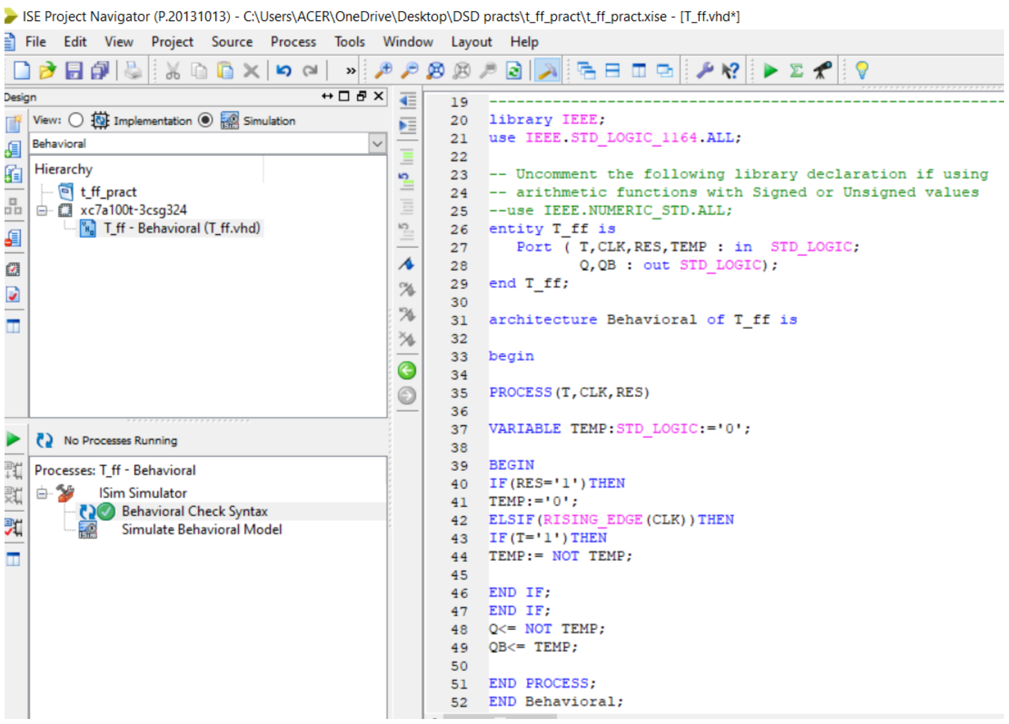
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity JK_ff is
    port( J, K, clk, rst : in std_logic;
          Q, Qbar : out std_logic);
end JK_ff;
architecture behavioral of JK_ff is
    begin
    process(clk, rst)
        variable qn : std_logic;
        begin
        if(rst = '1')then  qn := '0';
        elsif(clk'event and clk = '1')then
            if(J='0' and K='0')then
                qn := qn;
            elsif(J='0' and K='1')then
                qn := '0';
            elsif(J='1' and K='0')then
                qn := '1';
            elsif(J='1' and K='1')then
                qn := not qn;
            else
                null;
            end if;
        else
            null;
        end if;
Q <= qn;
Qbar <= not qn;
end process;
end behavioral;
```

3) D Flip-Flop


ISE Project Navigator (P.20131013) - C:\Users\ACER\OneDrive\Desktop\DSD practs\d_ff_pract\d_ff_pract.xise - [D_ff.vhd*]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Behavioral
Hierarchy
d_ff_pract
xc7a100t-3csg324
D_ff - Behavioral (D_ff.vhd)
No Processes Running
Processes: D_ff - Behavioral
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

```
----------------------------------------------------------------------------------
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;

entity D_ff is
Port ( D, CLK, RST : in  STD_LOGIC;
Q, Qb : out  STD_LOGIC);
end D_ff;

architecture Behavioral of D_ff is

begin
process (D, CLK, RST)
begin

if (RST = '1') then
Q <= '0';
elsif (rising_edge(CLK)) then
Q <= D;
Qb <= not D;

end if;
end process;
end Behavioral;
```

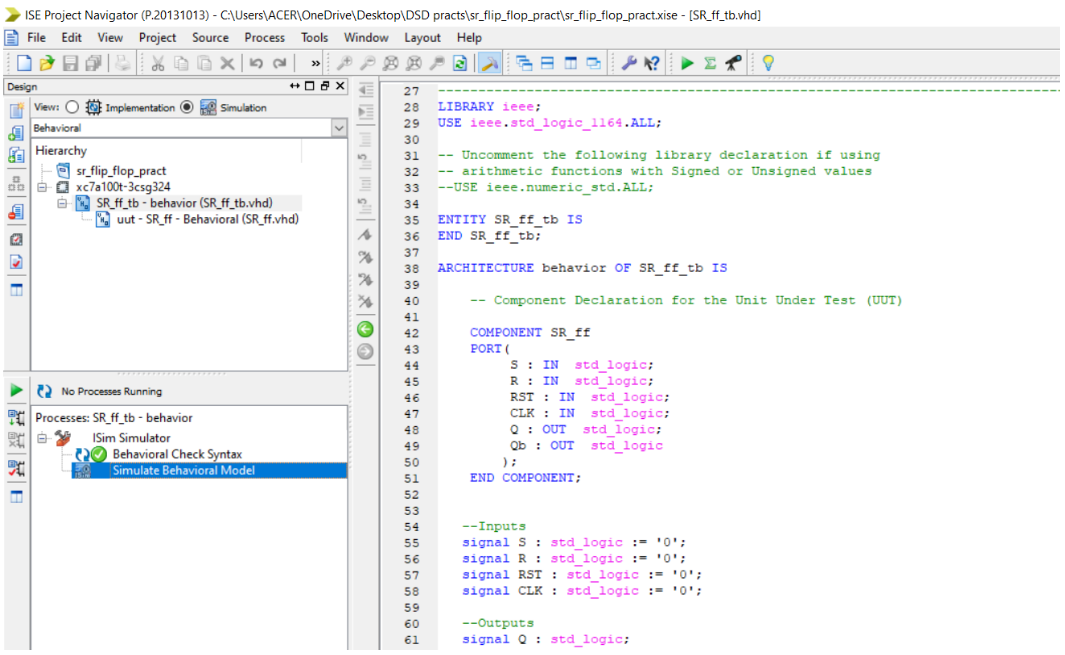
4) T Flip-Flop


ISE Project Navigator (P.20131013) - C:\Users\ACER\OneDrive\Desktop\DSD practs\t_ff_pract\t_ff_pract.xise - [T_ff.vhd*]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Behavioral
Hierarchy
t_ff_pract
xc7a100t-3csg324
T_ff - Behavioral (T_ff.vhd)
No Processes Running
Processes: T_ff - Behavioral
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

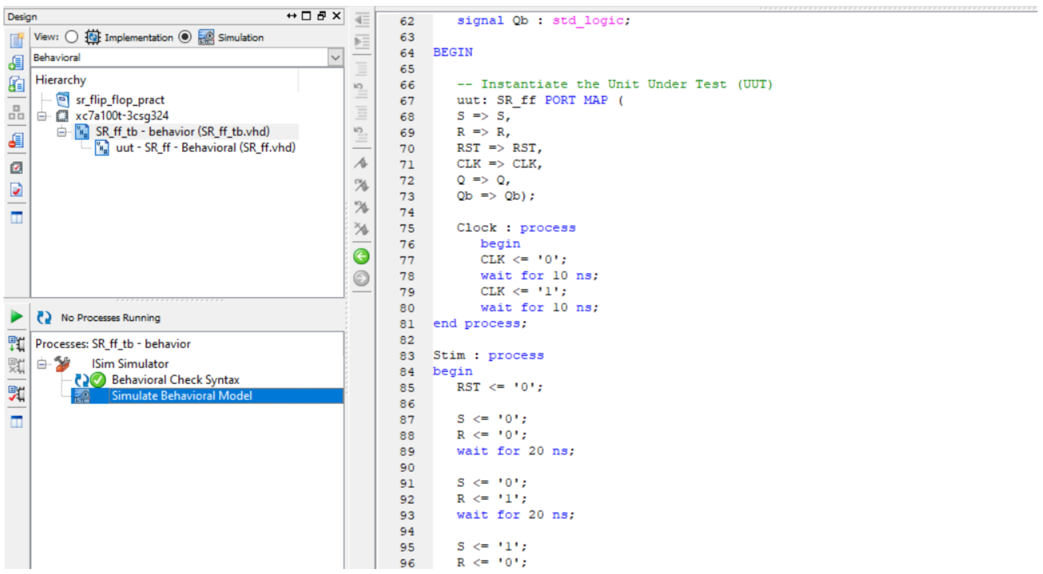
```
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
entity T_ff is
    Port ( T,CLK,RES,TEMP : in  STD_LOGIC;
           Q,QB : out STD_LOGIC);
end T_ff;

architecture Behavioral of T_ff is

begin

PROCESS(T,CLK,RES)

VARIABLE TEMP:STD_LOGIC:='0';

BEGIN
IF(RES='1')THEN
TEMP:='0';
ELSIF(RISING_EDGE(CLK))THEN
IF(T='1')THEN
TEMP:= NOT TEMP;

END IF;
END IF;
Q<= NOT TEMP;
QB<= TEMP;

END PROCESS;
END Behavioral;
```

TESTBENCH:

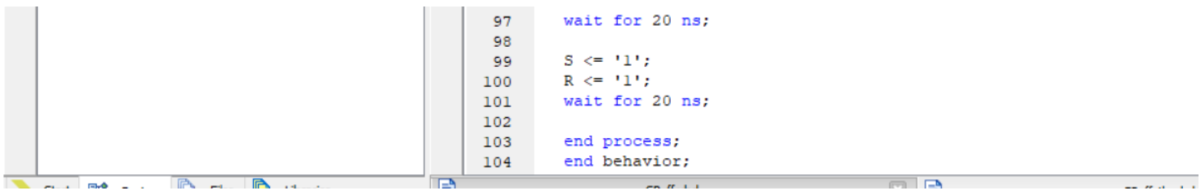
1) SR Flip-Flop


ISE Project Navigator (P.20131013) - C:\Users\ACER\OneDrive\Desktop\DSD practs\sr_flip_flop_pract\sr_flip_flop_pract.xise - [SR_ff_tb.vhd]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Behavioral
Hierarchy
sr_flip_flop_pract
xc7a100t-3csg324
SR_ff_tb - behavior (SR_ff_tb.vhd)
uut - SR_ff - Behavioral (SR_ff.vhd)
No Processes Running
Processes: SR_ff_tb - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

```vhdl
--------------------------------------------------------------------------------
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;

ENTITY SR_ff_tb IS
END SR_ff_tb;

ARCHITECTURE behavior OF SR_ff_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT SR_ff
    PORT(
         S : IN  std_logic;
         R : IN  std_logic;
         RST : IN  std_logic;
         CLK : IN  std_logic;
         Q : OUT  std_logic;
         Qb : OUT  std_logic
        );
    END COMPONENT;


   --Inputs
   signal S : std_logic := '0';
   signal R : std_logic := '0';
   signal RST : std_logic := '0';
   signal CLK : std_logic := '0';

   --Outputs
   signal Q : std_logic;
```

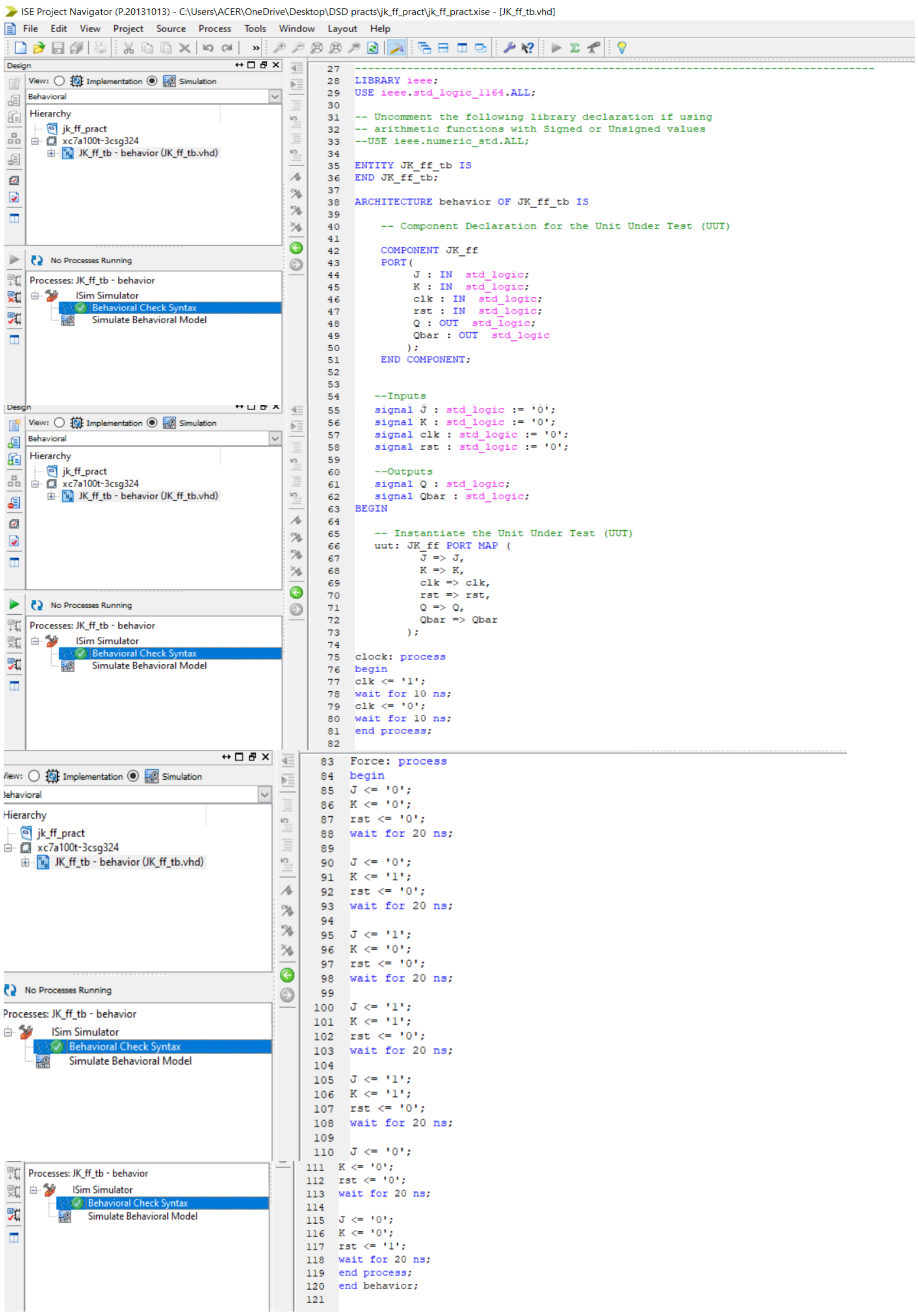

Design
View: Implementation Simulation
Behavioral
Hierarchy
sr_flip_flop_pract
xc7a100t-3csg324
SR_ff_tb - behavior (SR_ff_tb.vhd)
uut - SR_ff - Behavioral (SR_ff.vhd)
No Processes Running
Processes: SR_ff_tb - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

```vhdl
   signal Qb : std_logic;

BEGIN

   -- Instantiate the Unit Under Test (UUT)
   uut: SR_ff PORT MAP (
   S => S,
   R => R,
   RST => RST,
   CLK => CLK,
   Q => Q,
   Qb => Qb);

   Clock : process
      begin
      CLK <= '0';
      wait for 10 ns;
      CLK <= '1';
      wait for 10 ns;
end process;

Stim : process
begin
   RST <= '0';

   S <= '0';
   R <= '0';
   wait for 20 ns;

   S <= '0';
   R <= '1';
   wait for 20 ns;

   S <= '1';
   R <= '0';
```



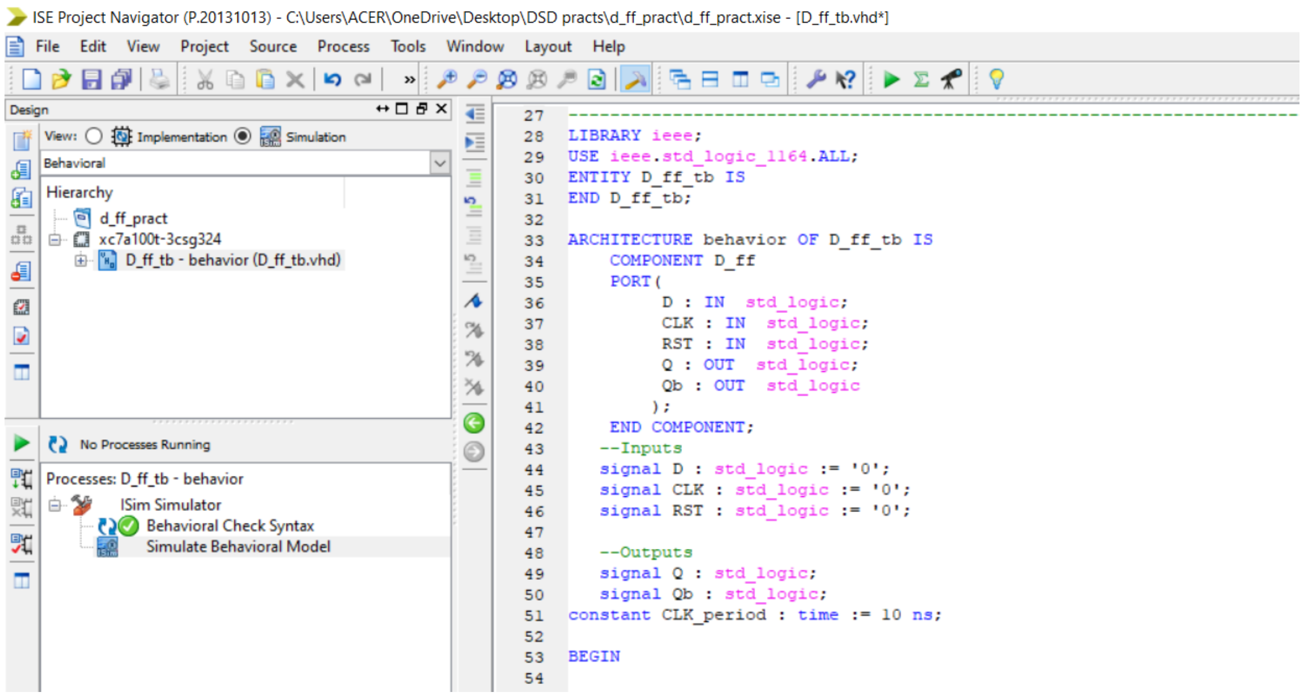
```vhdl
   wait for 20 ns;

   S <= '1';
   R <= '1';
   wait for 20 ns;

   end process;
   end behavior;
```

2) JK Flip-Flop

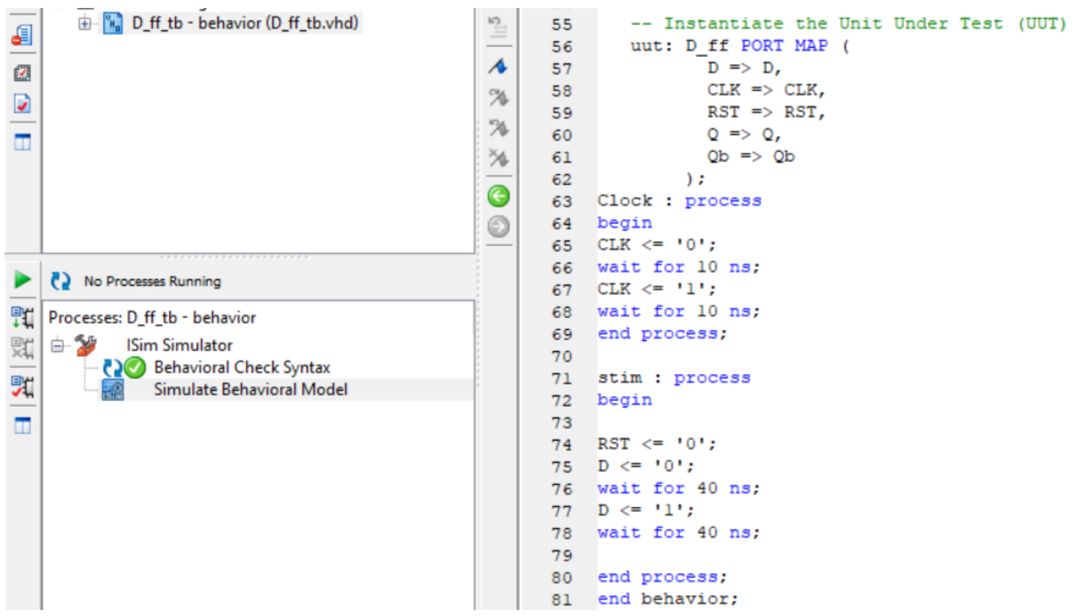

ISE Project Navigator (P.20131013) - C:\Users\ACER\OneDrive\Desktop\DSD practs\jk_ff_pract\jk_ff_pract.xise - [JK_ff_tb.vhd]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Behavioral
Hierarchy
jk_ff_pract
xc7a100t-3csg324
JK_ff_tb - behavior (JK_ff_tb.vhd)
No Processes Running
Processes: JK_ff_tb - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

```
--------------------------------------------------------------------------------
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;

ENTITY JK_ff_tb IS
END JK_ff_tb;

ARCHITECTURE behavior OF JK_ff_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT JK_ff
    PORT(
         J : IN  std_logic;
         K : IN  std_logic;
         clk : IN  std_logic;
         rst : IN  std_logic;
         Q : OUT  std_logic;
         Qbar : OUT  std_logic
        );
    END COMPONENT;


   --Inputs
   signal J : std_logic := '0';
   signal K : std_logic := '0';
   signal clk : std_logic := '0';
   signal rst : std_logic := '0';

   --Outputs
   signal Q : std_logic;
   signal Qbar : std_logic;
BEGIN

   -- Instantiate the Unit Under Test (UUT)
   uut: JK_ff PORT MAP (
          J => J,
          K => K,
          clk => clk,
          rst => rst,
          Q => Q,
          Qbar => Qbar
        );

clock: process
begin
clk <= '1';
wait for 10 ns;
clk <= '0';
wait for 10 ns;
end process;

Force: process
begin
J <= '0';
K <= '0';
rst <= '0';
wait for 20 ns;

J <= '0';
K <= '1';
rst <= '0';
wait for 20 ns;

J <= '1';
K <= '0';
rst <= '0';
wait for 20 ns;

J <= '1';
K <= '1';
rst <= '0';
wait for 20 ns;

J <= '1';
K <= '1';
rst <= '0';
wait for 20 ns;

J <= '0';
K <= '0';
rst <= '0';
wait for 20 ns;

J <= '0';
K <= '0';
rst <= '1';
wait for 20 ns;
end process;
end behavior;

```

3) D Flip-Flop

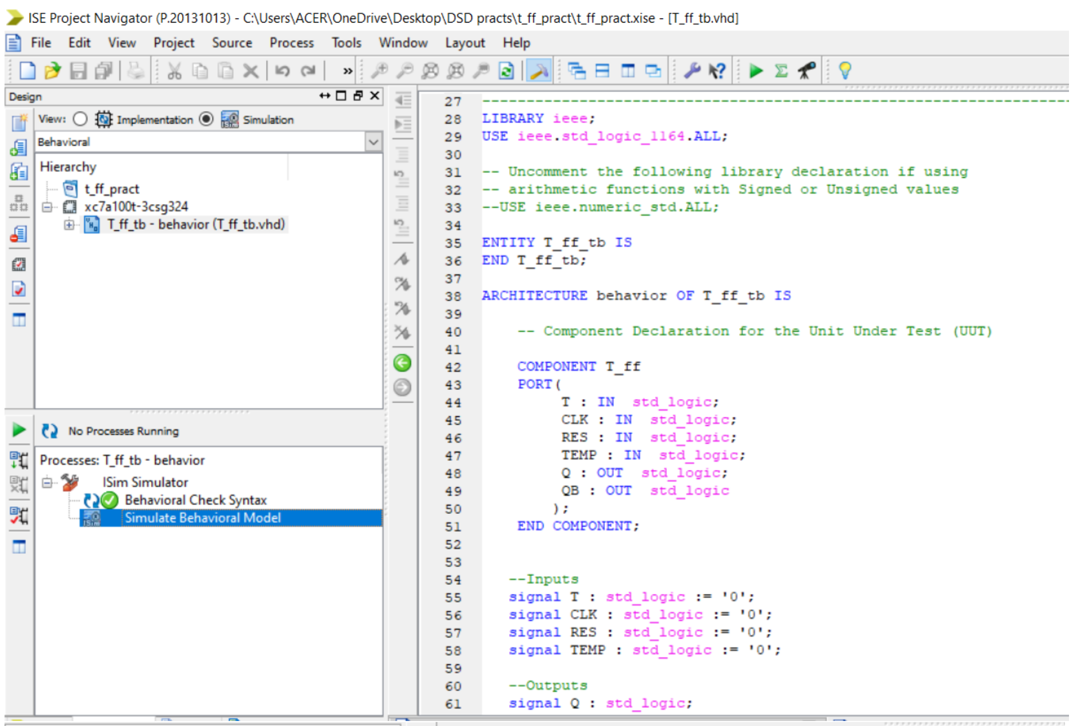

ISE Project Navigator (P.20131013) - C:\Users\ACER\OneDrive\Desktop\DSD practs\d_ff_pract\d_ff_pract.xise - [D_ff_tb.vhd*]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Behavioral
Hierarchy
d_ff_pract
xc7a100t-3csg324
D_ff_tb - behavior (D_ff_tb.vhd)
No Processes Running
Processes: D_ff_tb - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

```
--------------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY D_ff_tb IS
END D_ff_tb;

ARCHITECTURE behavior OF D_ff_tb IS
    COMPONENT D_ff
    PORT(
         D : IN  std_logic;
         CLK : IN  std_logic;
         RST : IN  std_logic;
         Q : OUT  std_logic;
         Qb : OUT  std_logic
        );
    END COMPONENT;
   --Inputs
   signal D : std_logic := '0';
   signal CLK : std_logic := '0';
   signal RST : std_logic := '0';

   --Outputs
   signal Q : std_logic;
   signal Qb : std_logic;
constant CLK_period : time := 10 ns;

BEGIN

```

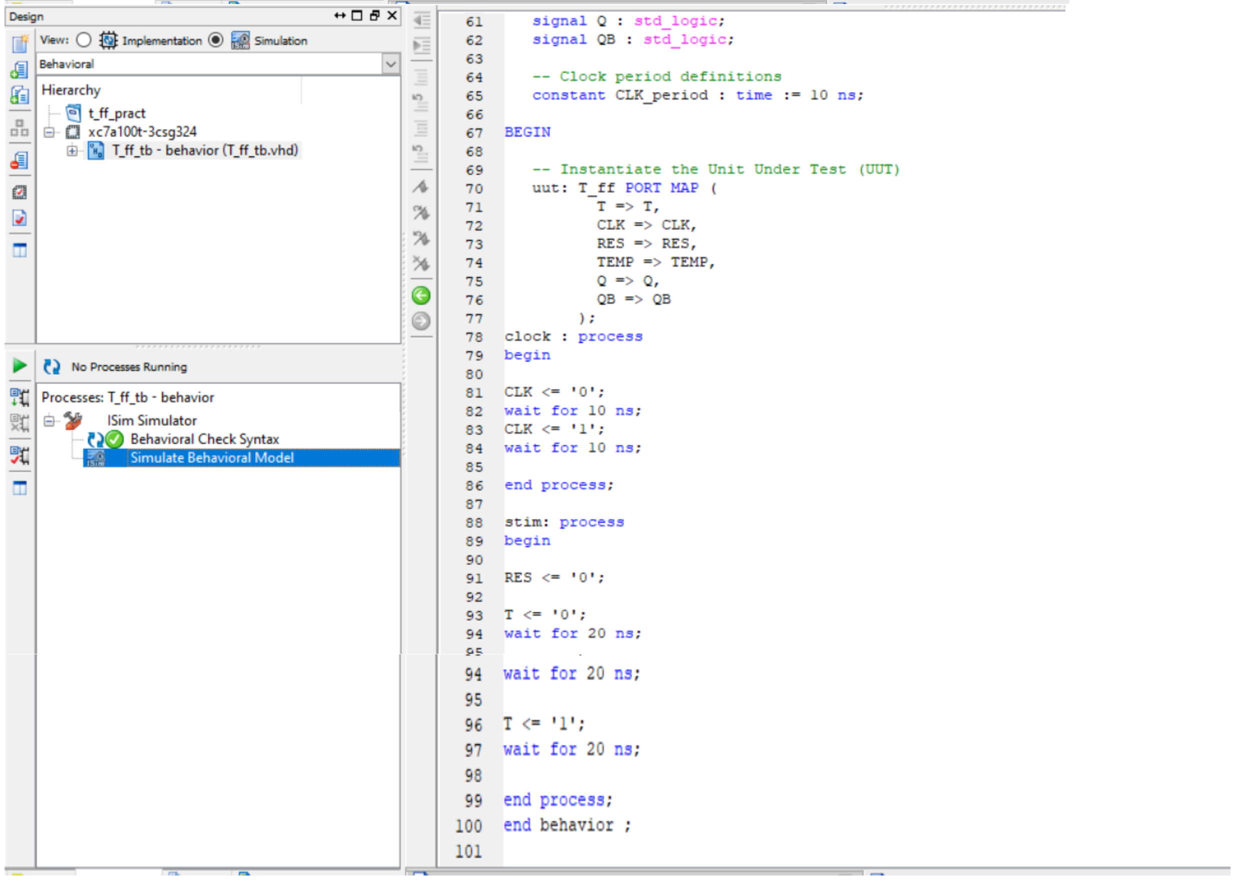

D_ff_tb - behavior (D_ff_tb.vhd)
No Processes Running
Processes: D_ff_tb - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

```
   -- Instantiate the Unit Under Test (UUT)
   uut: D_ff PORT MAP (
          D => D,
          CLK => CLK,
          RST => RST,
          Q => Q,
          Qb => Qb
        );
Clock : process
begin
CLK <= '0';
wait for 10 ns;
CLK <= '1';
wait for 10 ns;
end process;

stim : process
begin

RST <= '0';
D <= '0';
wait for 40 ns;
D <= '1';
wait for 40 ns;

end process;
end behavior;
```

4) T Flip-Flop


ISE Project Navigator (P.20131013) - C:\Users\ACER\OneDrive\Desktop\DSD practs\t_ff_pract\t_ff_pract.xise - [T_ff_tb.vhd]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Behavioral
Hierarchy
t_ff_pract
xc7a100t-3csg324
T_ff_tb - behavior (T_ff_tb.vhd)
No Processes Running
Processes: T_ff_tb - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

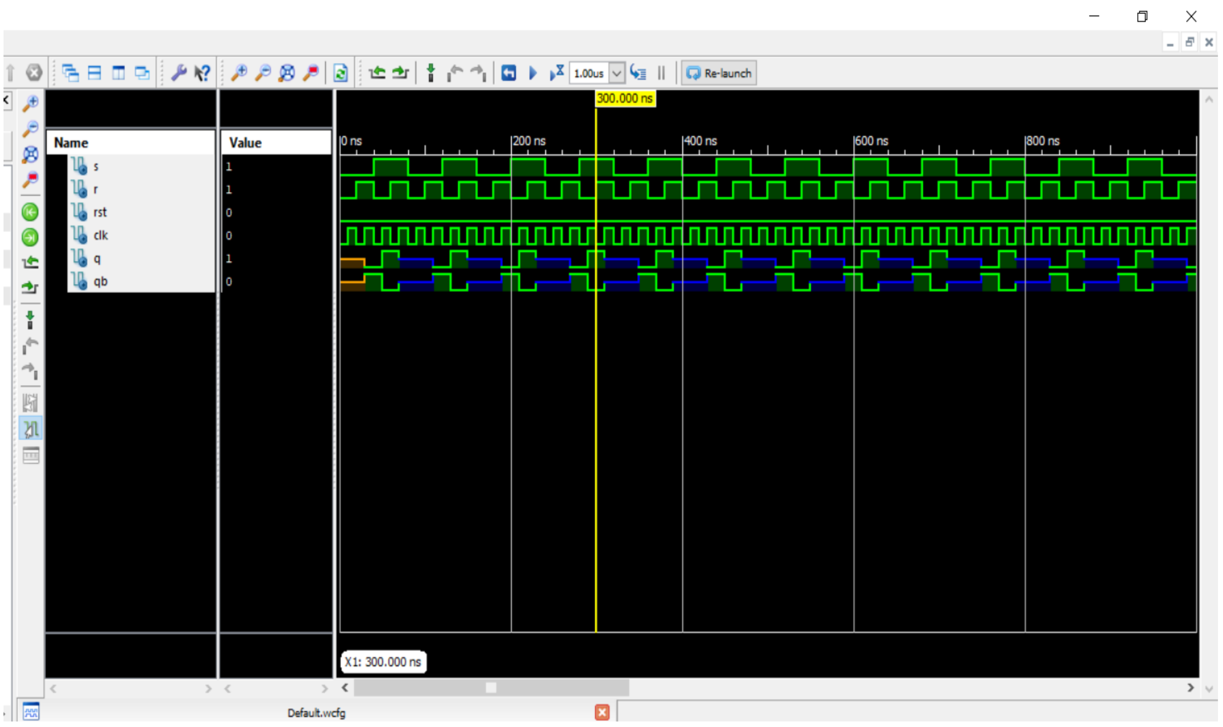
```
----------------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

ENTITY T_ff_tb IS
END T_ff_tb;

ARCHITECTURE behavior OF T_ff_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT T_ff
    PORT(
         T : IN  std_logic;
         CLK : IN  std_logic;
         RES : IN  std_logic;
         TEMP : IN  std_logic;
         Q : OUT  std_logic;
         QB : OUT  std_logic
        );
    END COMPONENT;


   --Inputs
   signal T : std_logic := '0';
   signal CLK : std_logic := '0';
   signal RES : std_logic := '0';
   signal TEMP : std_logic := '0';

   --Outputs
   signal Q : std_logic;
```


Design
View: Implementation Simulation
Behavioral
Hierarchy
t_ff_pract
xc7a100t-3csg324
T_ff_tb - behavior (T_ff_tb.vhd)
No Processes Running
Processes: T_ff_tb - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

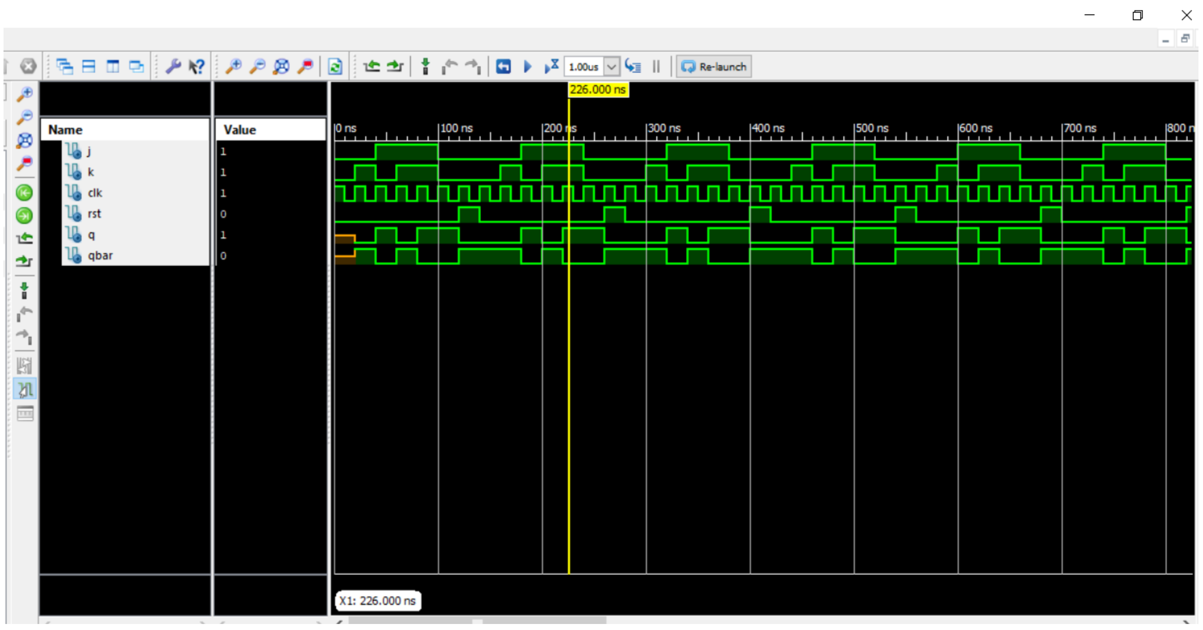
```
   signal Q : std_logic;
   signal QB : std_logic;

   -- Clock period definitions
   constant CLK_period : time := 10 ns;

BEGIN

   -- Instantiate the Unit Under Test (UUT)
   uut: T_ff PORT MAP (
          T => T,
          CLK => CLK,
          RES => RES,
          TEMP => TEMP,
          Q => Q,
          QB => QB
        );
clock : process
begin

CLK <= '0';
wait for 10 ns;
CLK <= '1';
wait for 10 ns;

end process;

stim: process
begin

RES <= '0';

T <= '0';
wait for 20 ns;
wait for 20 ns;

T <= '1';
wait for 20 ns;

end process;
end behavior ;
```

SIMULATION WAVEFORM OF FLIP FLOPS

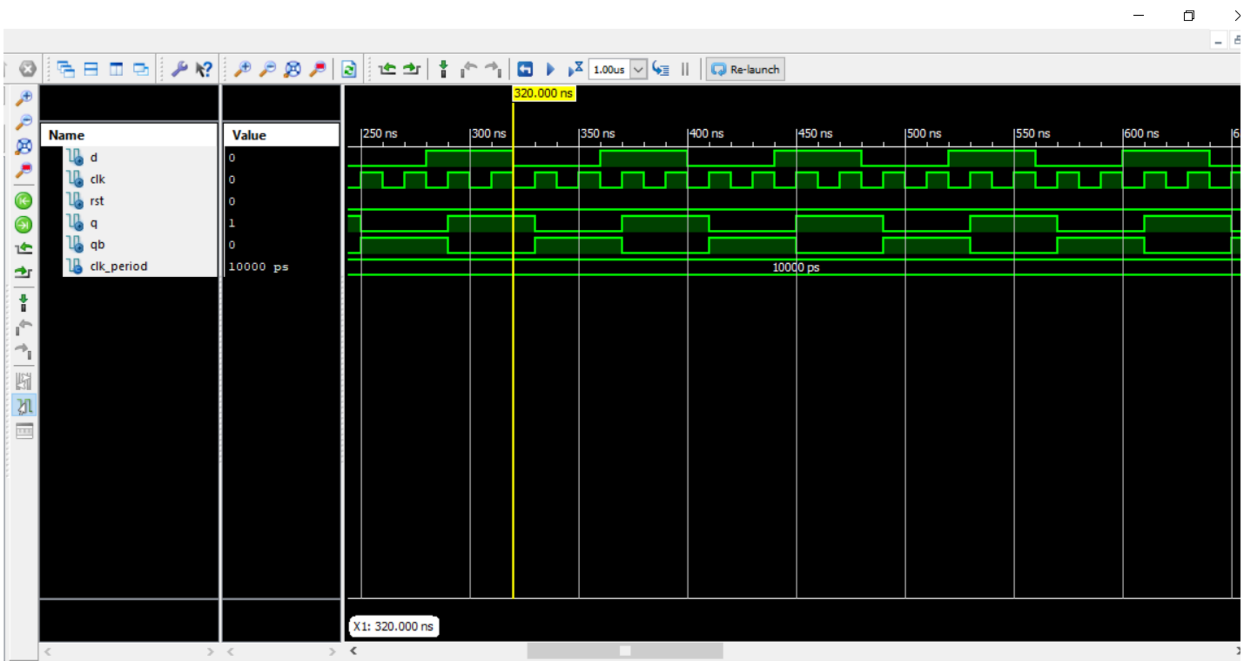
1) SR Flip-Flop



2) JK Flip-Flop

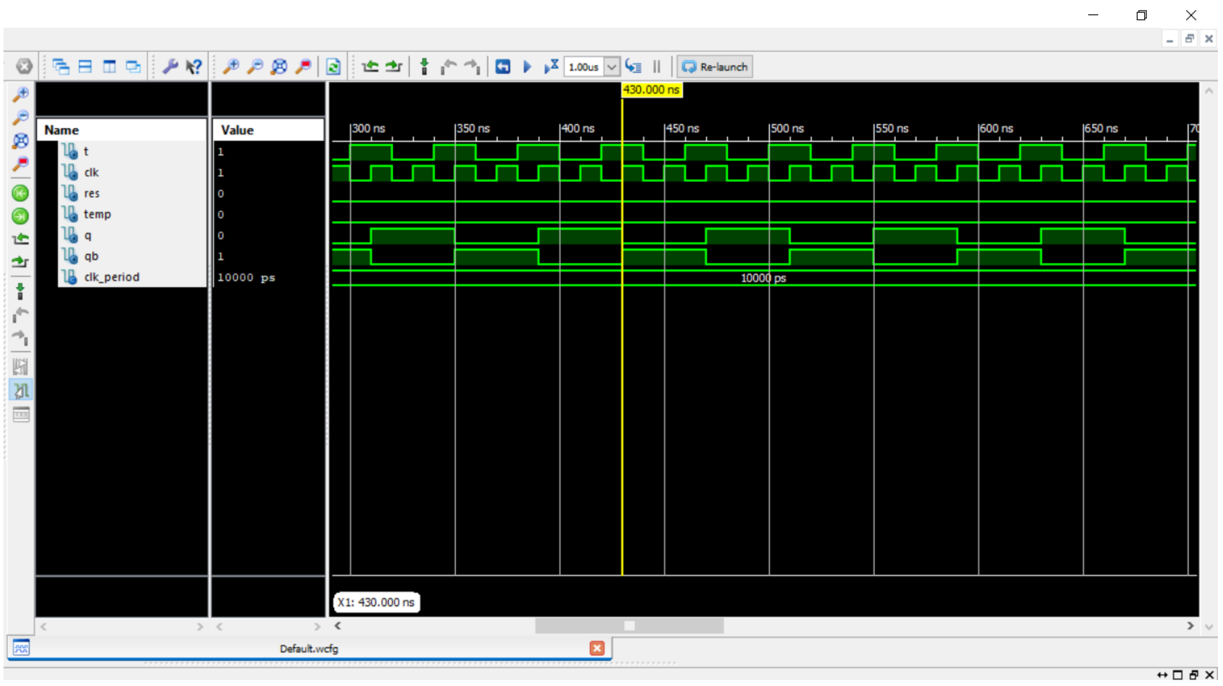


3) D Flip-Flop



Name	Value
d	0
clk	0
rst	0
q	1
qb	0
clk_period	10000 ps

4) T Flip-Flop



Name	Value
t	1
clk	1
res	0
temp	0
q	0
qb	1
clk_period	10000 ps

RESULT:-

The VHDL code for all the flip flops are executed and desired output is obtained.

CONCLUSION:

Here, I successfully Design SR flip flop(SRFF), JK flip flop(JKFF), D flip flop(DFF), T flip flop(TFF) and test bench for the same.

DISCUSSION & VIVA VOCE

1) Design T Flip flop using S R flip flop.
2) Design 4 bit shift register using flip flop.
3) Design up counter using flip flops.

REFERENCE:

- VHDL Primer–J Bhasker –Pearson Education
- NPTEL Video Lecture link https://www.youtube.com/watch?v=2ecMG_OciLo.