7

EXPERIMENT-7
REALIZATION OF FLIP FLOPS

1. AIM**:** To Design, simulate and implement Flip Flops in Behavioral Flow Modeling
2. **SOFTWARE USED:** Xilinx Vivado 2022.2

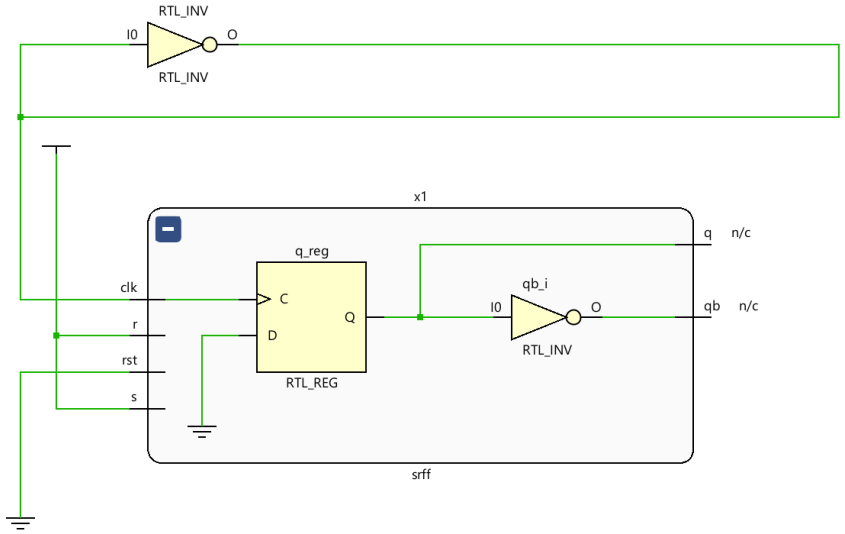
3. **PROCEDURE:**
    - Open the Xilinx Vivado project navigator.
    - Open the Design source and go to add / create sources
    - Create new file, give appropriate name save it.
    - Open the file in the editor and write the Verilog code.
    - Open the Design source and go to add / create sources to create the test bench
    - Open the editor and write the Verilog code for test bench.
    - After completion of Verilog code set your test bench as top module in simulation settings and Run Simulation.
    - To view RTL schematic, select RTL Analysis in which select open elaborate design, select Schematic.

Flip Flops

1.1 AIM: SR Flip Flop

1.2 SOFTWARE USED: Xilinx Vivado 2022.2

1.3 RTL Schematic:



1.4 TRUTH TABLE:

INPUT			OUTPUT
Clk	**S**	**R**	Q_{n+1}
0	**X**	**X**	Q_n
1	**0**	**0**	Q_n
1	**0**	**1**	**0**
1	**1**	**0**	**1**
1	**1**	**1**	**Invalid**

1.5 VERILOG CODE :

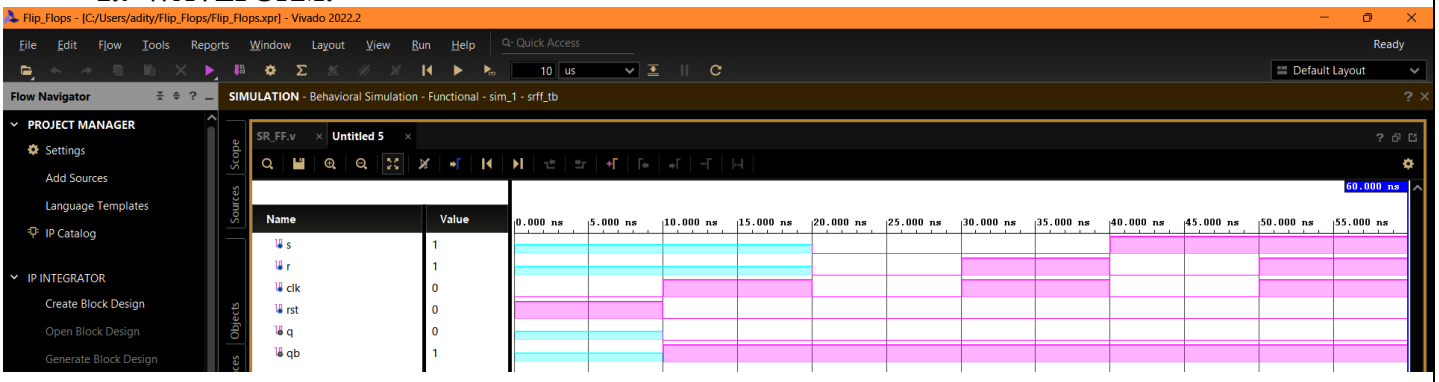
```
module srff(q,qb,s,r,clk,rst);
output q,qb;
 input s,r,clk,rst;
 reg q;
 assign qb=~q;
 always@(posedge clk)
 begin
 if(clk)
 q=1'b0;
 else
 case({s,r})
 2'b00: q<=q;
 2'b01: q<=1'b0;
 2'b10: q<=1'b1;
 2'b11: q<=1'bx;
 endcase
 end
 endmodule
```

1.6 TEST BENCH :

```
module srff_tb();
reg s,r,clk,rst;
wire q,qb;
srff x1(q,qb,s,r,clk,rst);
initial clk=0;
always#10clk=~clk;
initial
begin
rst=1'b1;
#10 rst=1'b0;

#10 {s,r}=2'b00;
#10 {s,r}=2'b01;
#10 {s,r}=2'b10;
#10 {s,r}=2'b11;
#10$finish;
end
endmodule
```

1.7 WAVEFORM:



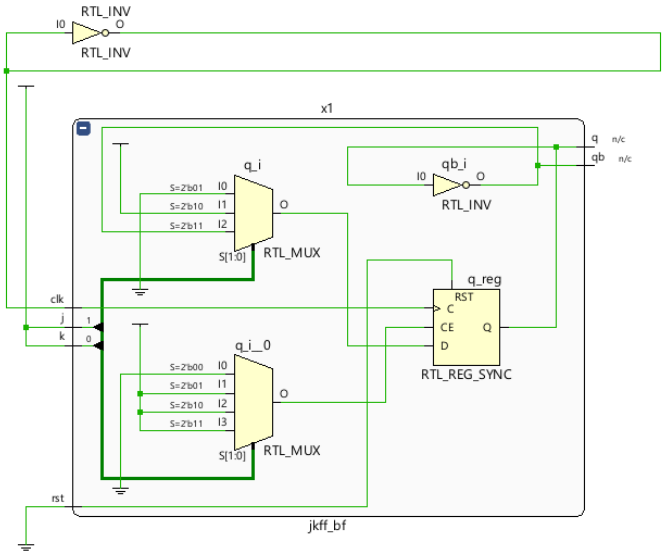
1.8 RESULT:

SR Flip Flop is implemented.

1.1 AIM: JK Flip Flop

1.2 SOFTWARE USED: Xilinx Vivado 2022.2

1.3 RTL Schematic:



1.4 TRUTH TABLE:

INPUT			OUTPUT
Clk	**S**	**R**	Q_{n+1}
0	**X**	**X**	Q_n
1	**0**	**0**	Q_n
1	**0**	**1**	**0**
1	**1**	**0**	**1**
1	**1**	**1**	$\overline{Q_n}$

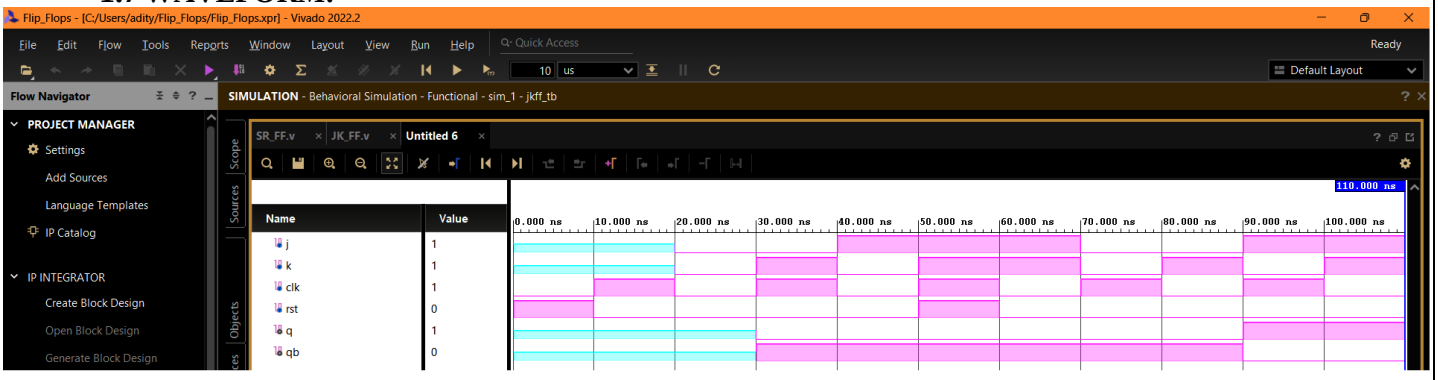
1.5 VERILOG CODE :

```
module srff(q,qb,s,r,clk,rst);
output q,qb;
 input s,r,clk,rst;
 reg q;
 assign qb=~q;
 always@(posedge clk)
 begin
 if(clk)
 q=1'b0;
 else
 case({s,r})
 2'b00: q<=q;
 2'b01: q<=1'b0;
 2'b10: q<=1'b1;
 2'b11: q<=1'bx;
 endcase
 end
 endmodule
```

1.6 TEST BENCH:

```
module jkff_tb();
reg j,k,clk,rst;
wire q,qb;
jkff_bf x1(q,qb,j,k,clk,rst);
initial clk=0;
always #10 clk=~clk;
initial
begin
rst=1'b1;
#10 rst=1'b0;
#10 {j,k}=2'b00;
#10 {j,k}=2'b01;
#10 {j,k}=2'b10;
#10 {j,k}=2'b11;
rst=1'b1;
#10 rst=1'b0;
#10 {j,k}=2'b00;
#10 {j,k}=2'b01;
#10 {j,k}=2'b10;
#10 {j,k}=2'b11;
#10 $finish;
end
endmodule
```

1.7 WAVEFORM:



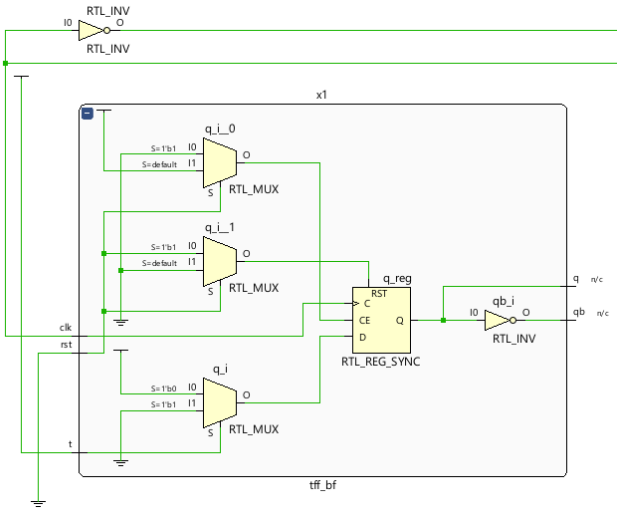
1.8 RESULT:

JK Flip Flop is implemented.

1.1 AIM: T Flip Flop

1.2 SOFTWARE USED: Xilinx Vivado 2022.2

1.3 RTL Schematic:



1.4 TRUTH TABLE:

INPUT		OUTPUT
Clk	**S**	Q_{n+1}
0	**X**	Q_n
1	**0**	Q_n
1	**1**	$\overline{Q_n}$

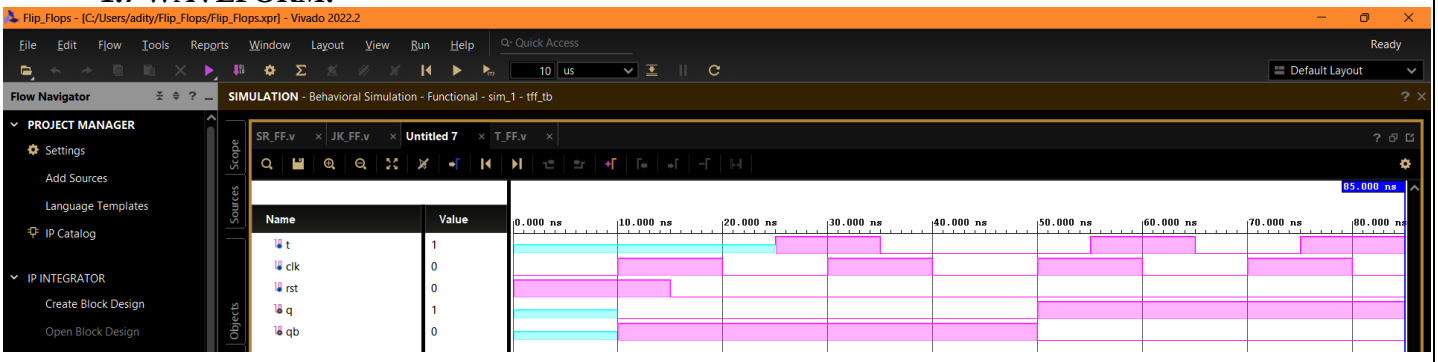
1.5 VERILOG CODE :

```
module tff_bf(q,qb,t,clk,rst);
output q,qb;
input t,clk,rst;
reg q;
assign qb=~q;
always@(posedge clk)
begin
if(rst)
q=1'b0;
else
case({t})
1'b0: q<=1'b1;
1'b1: q<=1'b0;
endcase
end
endmodule
```

1.6 TEST BENCH:

```
module tff_tb();
reg t,clk,rst;
wire q,qb;
tff_bf x1(q,qb,t,clk,rst);
initial clk=0;
always #10 clk=~clk;
initial
begin
rst=1'b1;
#15 rst=1'b0;
#10 {t}=1'b1;
#10 {t}=1'b0;
#10 {t}=1'b0;
#10 {t}=1'b1;
#10 {t}=1'b0;
#10 {t}=1'b1;
#10 $finish;
end
endmodule
```

1.7 WAVEFORM:



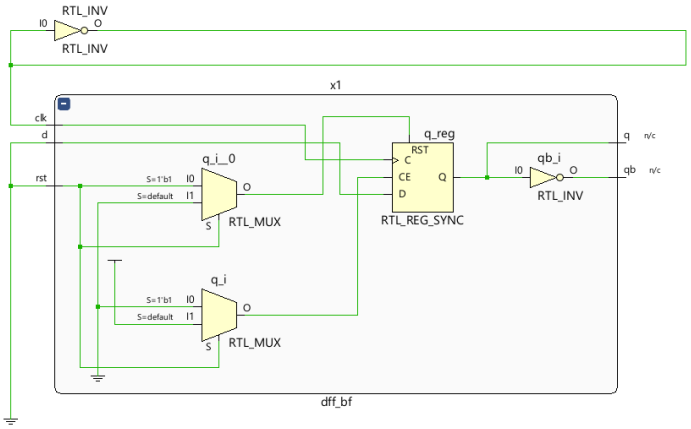
1.8 RESULT:

T Flip Flop is implemented.

1.1 AIM: D Flip Flop

1.2 SOFTWARE USED: Xilinx Vivado 2022.2

1.3 RTL Schematic:



1.4 TRUTH TABLE:

INPUT		OUTPUT
Clk	**S**	**Q_{n+1}**
0	**X**	**Q_n**
1	**0**	**0**
1	**1**	**1**

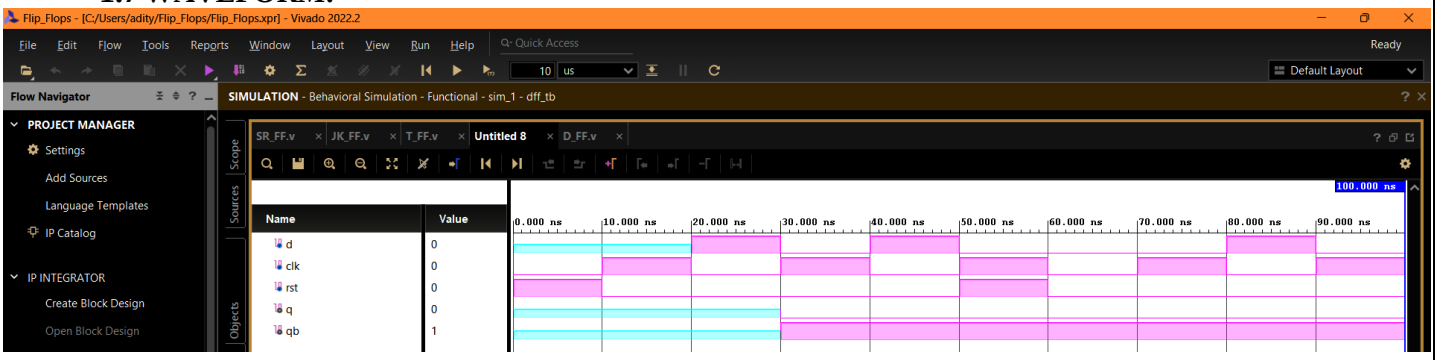
1.5 VERILOG CODE :

```
module dff_bf(q,qb,d,clk,rst);
output q,qb;
input d,clk,rst;
reg q;
assign qb=~q;
always@(posedge clk)
begin
if(rst)
q=1'b0;
else
case({d})
1'b0: q<=1'b0;
1'b1: q<=1'b1;
endcase
end
endmodule
```

1.6 TEST BENCH:

```
module dff_tb();
reg d,clk,rst;
wire q,qb;
dff_bf x1(q,qb,d,clk,rst);
initial clk=0;
always #10 clk=~clk;
initial
begin
rst=1'b1;
#10 rst=1'b0;
#10 {d}=1'b1;
#10 {d}=1'b0;
#10 {d}=1'b1;
#10 {d}=1'b0;
rst=1'b1;
#10 rst=1'b0;
#10 {d}=1'b0;
#10 {d}=1'b1;
#10 {d}=1'b0;
#10 $finish;
end
endmodule
```

1.7 WAVEFORM:



1.8 RESULT:

D Flip Flop is implemented.