

EXPERIMENT 5

- AIM : Design and simulate JK flip flop and D flip flop using Verilog behavioural modelling. Designing of JK flip flop using D flip flop and 2x1 multiplexer

- COMPONENT / SOFTWARE USED :

Component / Software	Specification
• ICs	7476, 7474, 74157, 7404
• Bread board, power supply, LED's, Resistors, Switches, Connecting wires	As per requirement
• Software (s) used	Vivado 2016.1

- THEORY :

Flip flop: Flip flop is the basic one bit digital memory circuit. It can store either 0 or 1. Flip flops are the basic building blocks of most sequential circuits. Flip flops are the fundamental components of shift registers and counters. A flip flop has two outputs, Q and \bar{Q} which are always in opposite states. If Q is 1, then \bar{Q} is 0, and the flip flop is said to be set or on. If Q is 0, then \bar{Q} is 1, and the flip flop is said to be reset, off or cleared. It can maintain a binary state indefinitely (as long as power is delivered to the circuit) until directed by an input signal to switch states. There are several types of flip flop, and the inputs vary with each type.

Flip flops trigger only during a signal transition (from 0 to 1 or from 1 to 0) and is disabled during the rest of the clock cycle pulse duration. Positive edge means positive transition i.e. signal transition from 0 to 1. Negative edge means negative transition, i.e. signal transition from 1 to 0.

Flip flops have asynchronous inputs that are used to force the flip flop into a particular state independent of the clock. The input that sets the flip flop to 1 is called a Preset. The input that clears the flip flop is 0 is called Clear. When power is turned on in a digital system, the state of the flip flop is unknown. The direct inputs are useful for bringing all flip flops in the system to a known starting state prior to the clocked operation.

JK Flip Flop:

The JK Flip flop is very versatile and also most widely used. The outputs are complement to each other. A clock input is included in edge triggered flip flops.

Step	Inputs		Ps	Ns	Inputs				Output	Comment
	J	K	$Q(t)$	$Q(t+1)$	CLK	J	K	Q		
0	0	0	0	0	1	X	X	$Q(t)$	No change	
0	0	1	1	1	0	0	0	$Q(t)$	No change	
0	1	1	0	0	0	0	1	1	Set	
1	0	0	1	1	0	1	0	0	Reset	
1	0	1	1	1	0	1	1	$\bar{Q}(t)$	Toggle	
1	1	0	1	1	JK Flip Flop Function table					
1	1	1	1	0						

Characteristic table of JK FF

Teacher's Signature : _____

D Flip Flop

In D flip flop, the output is same as input i.e. when $D=0$, the flip flop is reset and when $D=1$, the flip flop is set.

P.S	Input	NS		Inputs		Output	Comments
$Q(t)$	D	$Q(t+1)$		CLK	D	Q	
0	0	0		0	X	$Q(t)$	No change
0	1	1		1	0	0	Reset
1	0	0		1	1	1	Set
1	1	1					

• Conversion of D flip flop to JK flip flop

To convert one type of flip flop to another type, a combination circuit is designed using function table, next state equation and the excitation table such that if the inputs of the required flip flop are fed as inputs at the combinational circuit and the output of the combinational circuit is connected to the inputs of the actual flip flop, then the output of the actual flip flop is the output of the required flip flop.

P.S	N.S	Flip Flop inputs		
$Q(t)$	$Q(t+1)$	J	K	D
0	0	0	X	0
0	1	1	X	1
1	0	X	1	0
1	1	X	0	1

Excitation table

Flip flop 1/Ps	PS	NS	Required 1/Ps	
J + K	$Q(t)$	$Q(t+1)$	D	
0 0	0	0	0	
0 0	1	1	1	
0 1	0	0	0	
0 1	1	0	0	
1 0	0	1	1	
1 0	1	1	1	
1 1	0	1	1	
1 1	1	0	0	

Conversion table D flip flop to JK FF

From the above conversion table and the KMap simplification of the Boolean expressions for D input is: $D = J\bar{Q} + \bar{K}Q$. The D flip flop can be converted into a JK flip flop by giving the value of D in the D flip flop. Figure 5.6 shows the logic diagram for D flip flop converted to form the JK flip flop using required logic gates to implement the combinational logic. Boolean function can be implemented using multiplexer. Thus the combinational logic is ~~also~~ replaced by 2x1 Mux where select input is assigned the output of D flip flop $S_0 = Q$, and the inputs are assigned $I_0 = J$, $I_1 = \bar{K}$.

PROCEDURE

- For software simulation
 - Create a model with required number of variables and mentions its input/output.

- (b) Write the description of given Boolean function using operators or by using the built in primitive gates.
- (c) Synthesize to make RTL Schematic
- (d) Create another module referred as test bench to verify the functionality and to obtain the waveforms of input and output.
- (e) Follows the steps required to stimulate the design and compare the desired output with the corresponding truth table.
- (f) Take the screenshots of the RTL schematic and simulated waveforms.

• For hardware implementation

- (a) Turn off the power of the trainer kit before constructing any circuit
- (b) Connect power supply (+5V DC) pin and ground pin to the respective pins of the trainer kit.
- (c) Place the ICs properly on the bread board in the trainer kit.
- (d) Connect Vcc and GND pins of each chip to the power supply and ground bus strips on the bread board.
- (e) Connect the input and output pins of chips to the input switcher and output LEDs respectively in the trainer kit
- (f) Check the connections before you turn on the power
- (g) Apply various combinations of inputs according to truth tables and observe outputs

■ Design Code for JK Flip Flop

```

module JK_FF (output q,qb, input j,k,clk);
reg q = 0;
always @ (posedge clk)
case ({j,k})
  2'b00 : q <= q;
  2'b01 : q <= 0;
  2'b10 : q <= 1;
  2'b11 : q <= ~q;
endcase
assign qb = q;
endmodule

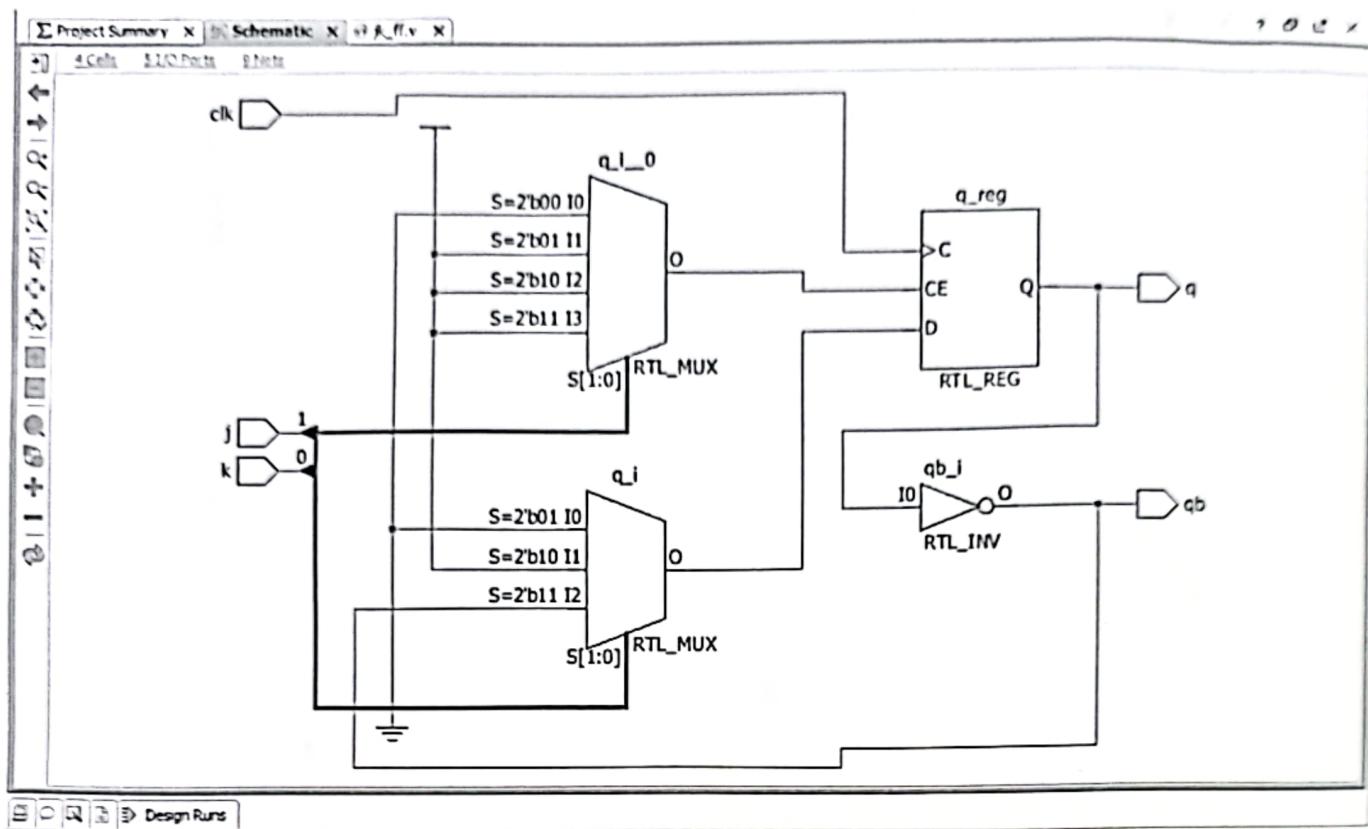
```

■ Test bench code for JK flip flop

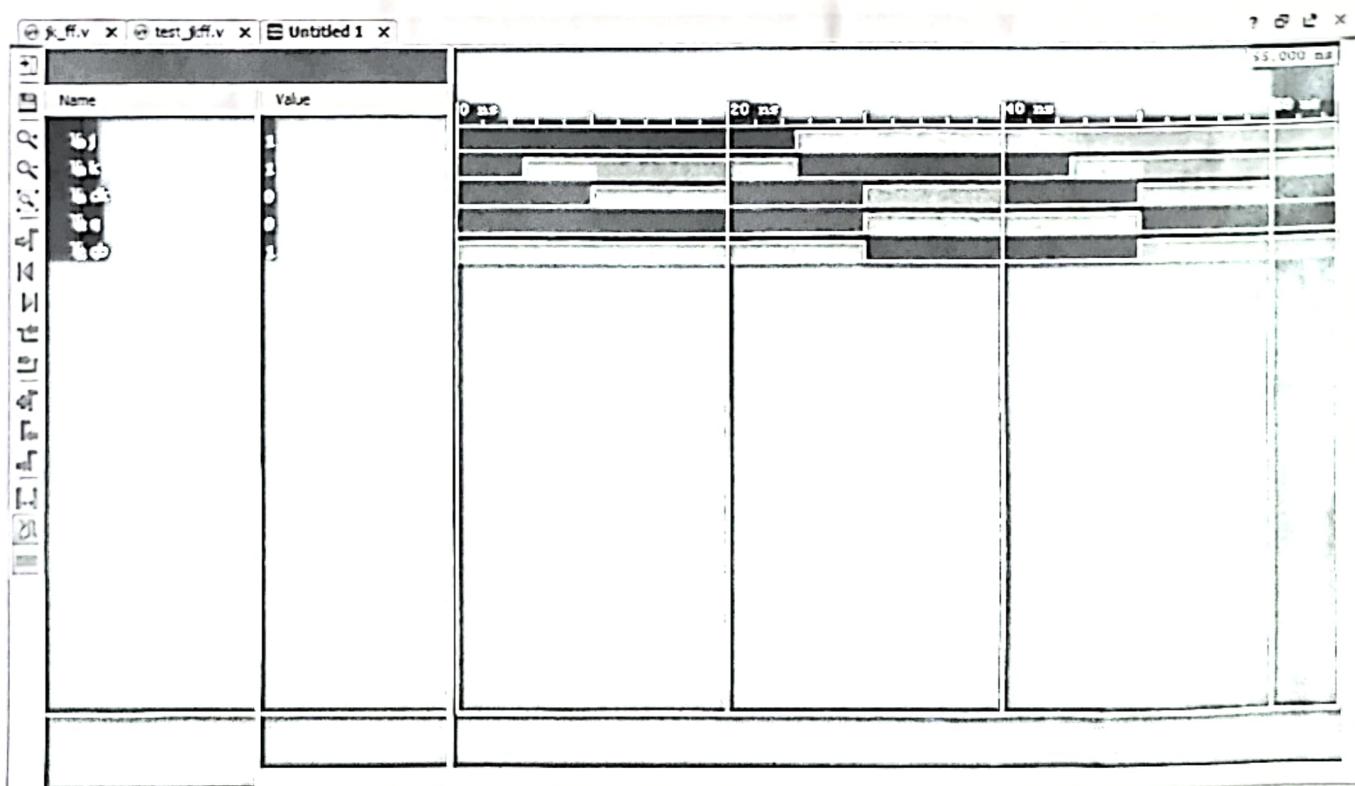
```

module test_jkff();
reg j,k,clk=0;
always #10 clk=~clk;
JK_FF DUT (q,qb,j,k,clk);
initial begin
  j<=0; k<=0; #5
  j<=0; k<=1; #20
  j<=1; k<=0; #20
  j<=1; k<=1; #20;
  $finish;
end
endmodule

```



SCHEMATIC REPRESENTATION OF JK FLIP FLOP



WAVEFORM OF JK FLIP FLOP

■ Design code for D flip flop

```

module D_FF (input d,clk,rst, output reg q,qb);
    always @ (posedge clk)
        begin
            if (rst == 1'b1)
                begin
                    q <= 1'b0; qb <= 1'b1;
                end
            else
                begin
                    q <= d; qb <= nd;
                end
        end
endmodule

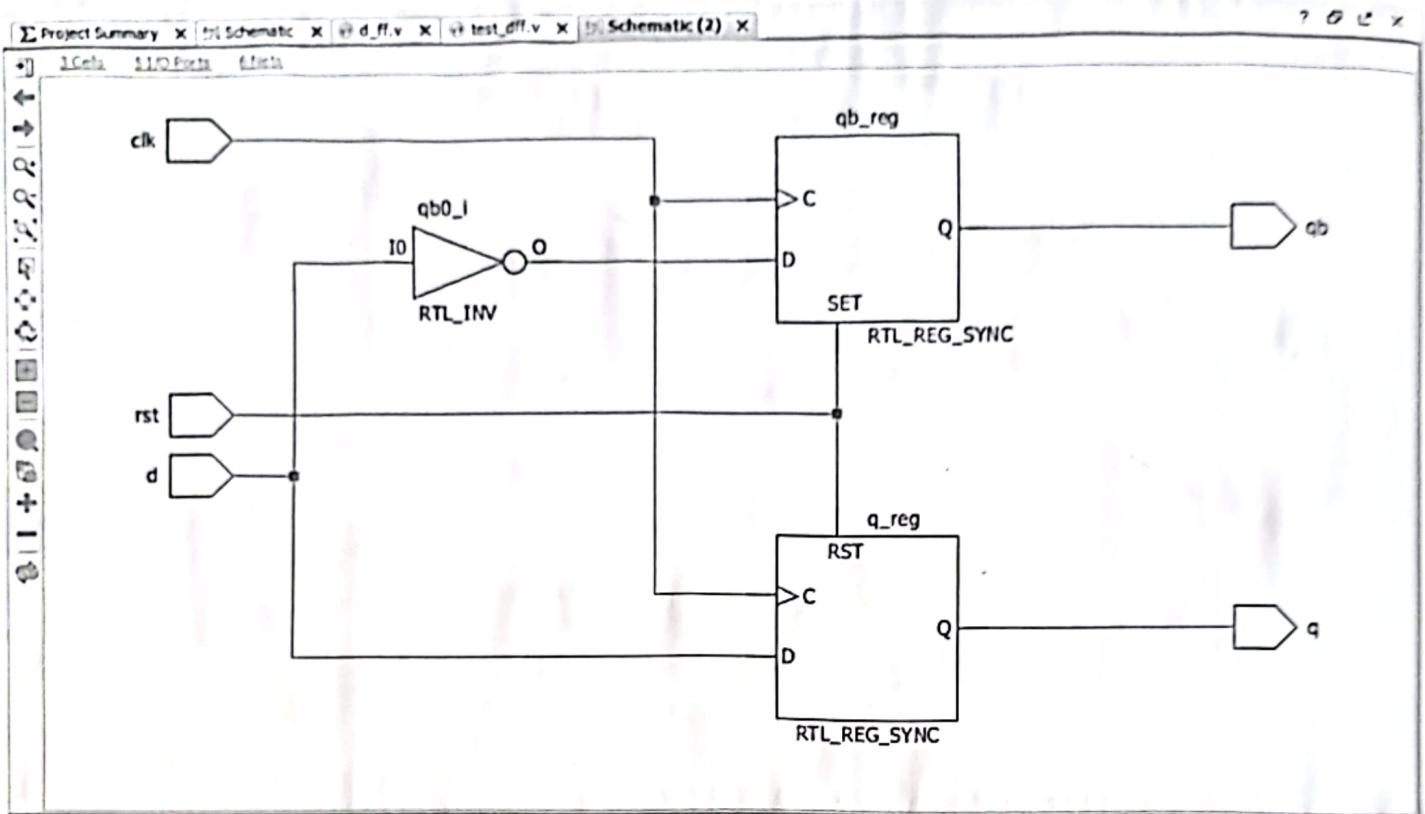
```

■ Test bench for D flip flop

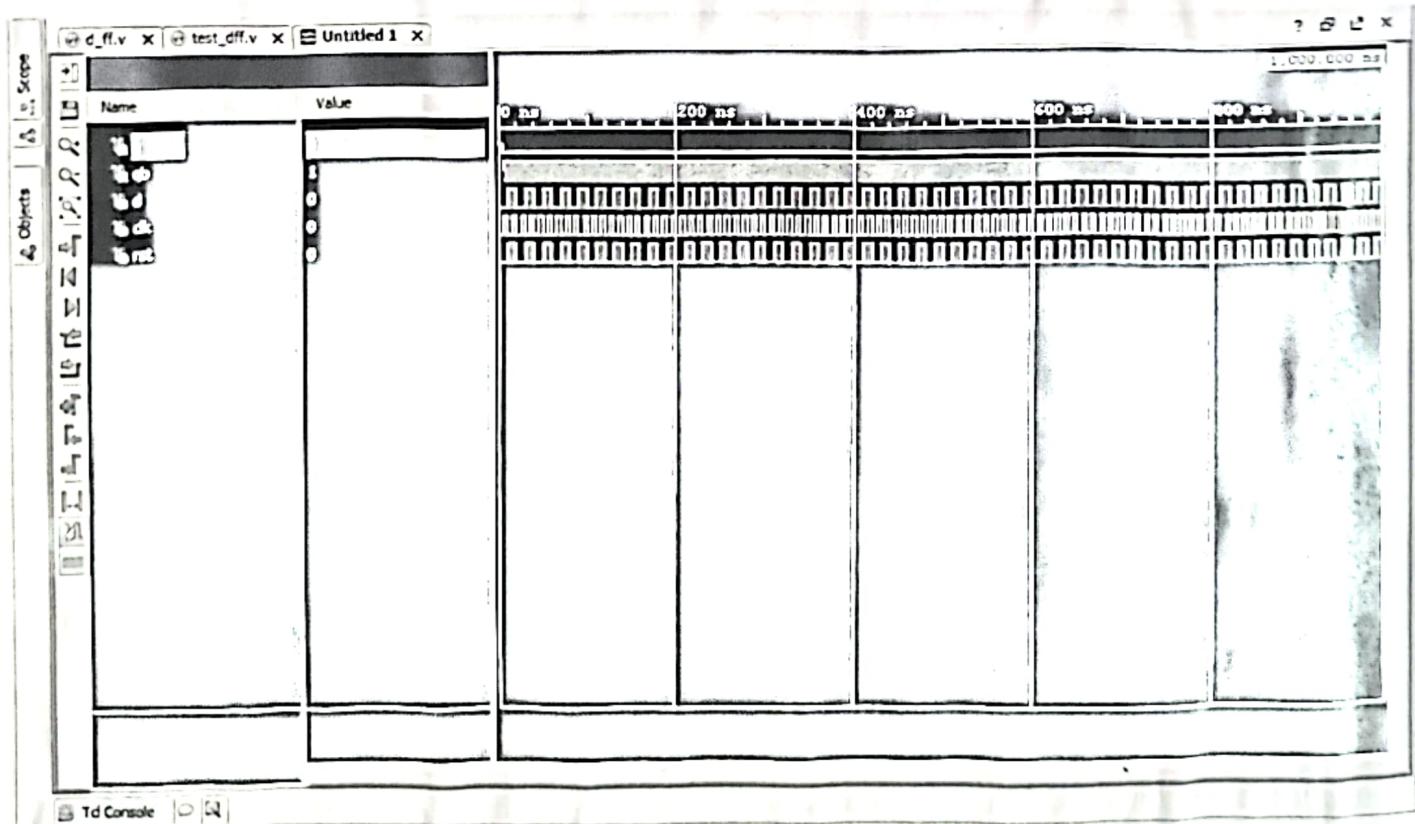
```

module test_DFF();
    wire q,qb;
    reg d,clk,rst;
    D_FF DUT (d,clk,q,qb);
    initial begin
        clk = 1'b0;
        rst = 1'b0;
        d = 1'b0;
    end
    always #5 clk = ~clk;
    always #10 d = nd;
    always #10 rst = nrst;
endmodule

```



SCHEMATIC REPRESENTATION OF D FLIP FLOP



WAVEFORM OF D FLIP FLOP

■ Design code for D flip flop to JK flip flop

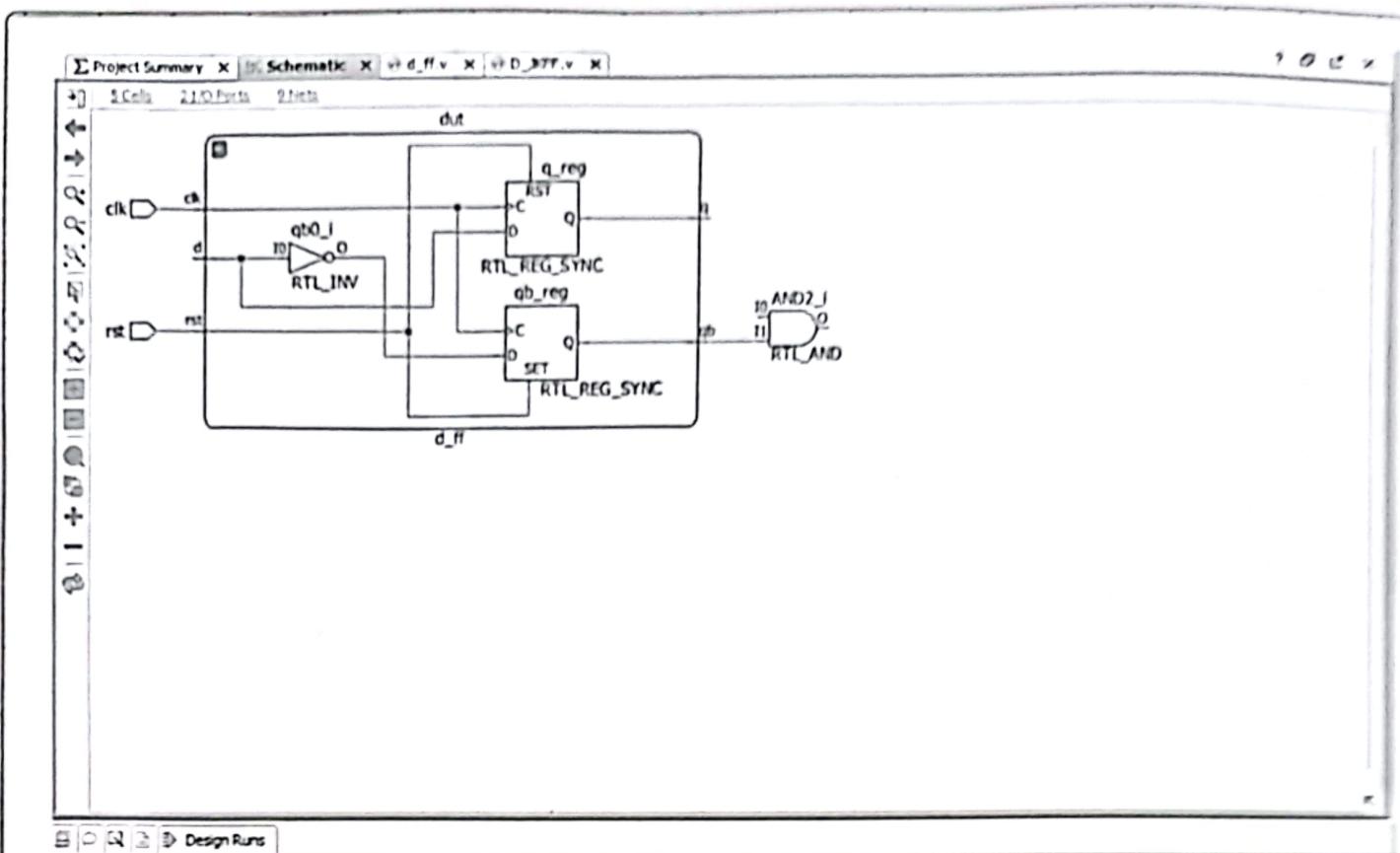
```
module D_JKFF (output q, input j,k,clk,out);
    wire w1,w2,w3;
    D_FF dut (d,clk,out,q,qb);
    not NOT (w1,n);
    and AND1 (w2,w1,q);
    and AND2 (w3,j,qb);
    or OR (d,w2,w3);
endmodule
```

■ Test bench code for D flip flop to JK flip flop

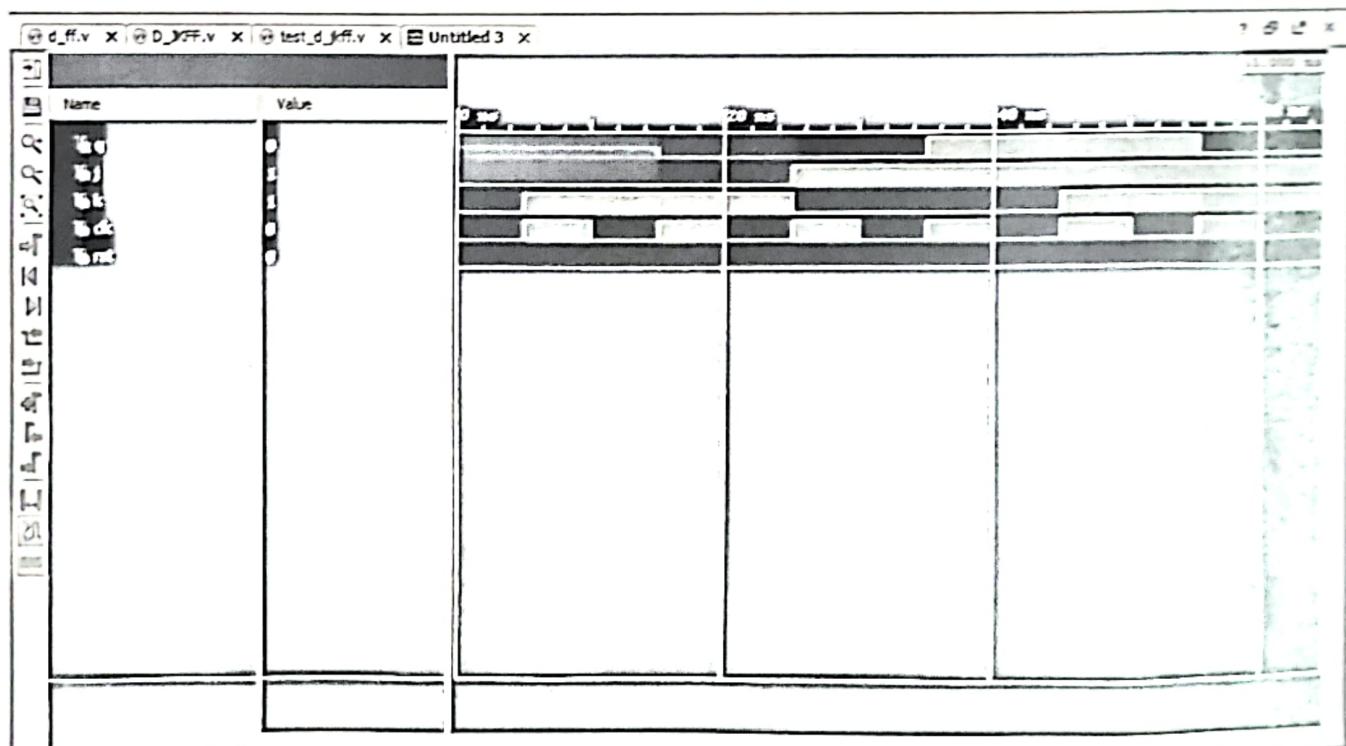
```
module test_d_jkff();
    wire q;
    reg j,k,clk=0,out=0;
    always #5 clk=~clk;
    D_JKFF DUT(q,j,k,clk,out);
    initial begin
        j<=0; k<=0; #15;
        j<=0; k<=0; #20;
        j<=1; k<=1; #20;
        $finish;
    end
endmodule
```

■ DESIGN PROBLEM

Design and Simulation of D Flip flop using JK flip flop
in Verilog HDL. Hardware implementation of D Flip flop
using JK flip flop.



SCHEMATIC REPRESENTATION OF D FLIP FLOP TO JK FLIP FLOP



WAVEFORM OF DFLIP FLOP TO JK FLIP FLOP

flip flop I/P	P_S	N_S	Required I/Ps	
J	$Q(t)$	$\bar{Q}(t+1)$	J	K
0	0	0	0	x
1	0	1	1	x
0	1	0	x	1
1	1	1	x	0

conversion table of JK flip flop to D flip flop

From the above conversion table, K Map simplification of the boolean expression for J and K inputs are: $J = D \cdot \bar{x} = \bar{D}$.

The JK flip can be converted into a D flip flop by giving the value of J and K in the JK flip flop. Figure 5.8 shows the logic diagram for D flip flop converted from JK flip flop.

■ CONCLUSION

Design and simulate JK flip flop and D flip flop using Verilog behavioural modelling has been done and implemented successfully.