Experiment 6

Aim: Write Verilog codes to create a Positive edge trigger JK Flip flop and verify it using a testbench.

Theory:

A JK flip-flop is a sequential logic device with two inputs, J (set) and K (reset), and two outputs, Q (output) and Q' (complement of Q). The flip-flop operates on the rising edge or falling edge of a clock signal (positive-edge or negative-edge triggered). It can be used to store one bit of information. The primary operations of a JK flip-flop are as follows:

When J = 0 and K = 0, the flip-flop remains in its current state (no change).

When J = 0 and K = 1, the flip-flop resets (Q becomes 0).

When J = 1 and K = 0, the flip-flop sets (Q becomes 1).

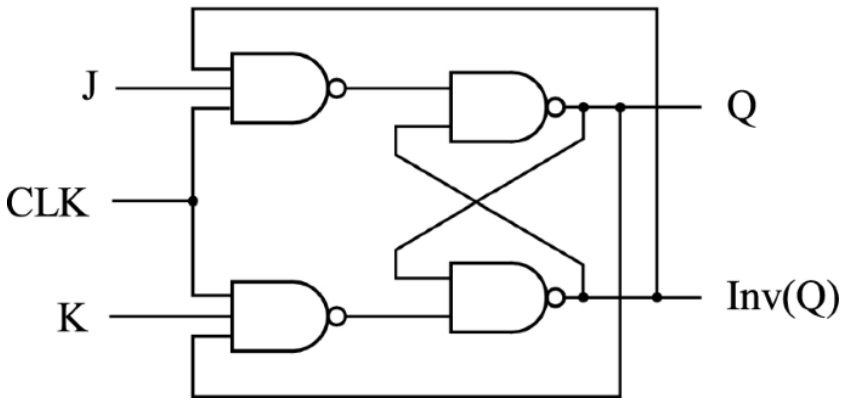
When J = 1 and K = 1, the flip-flop toggles its state (Q' becomes the complement of Q).

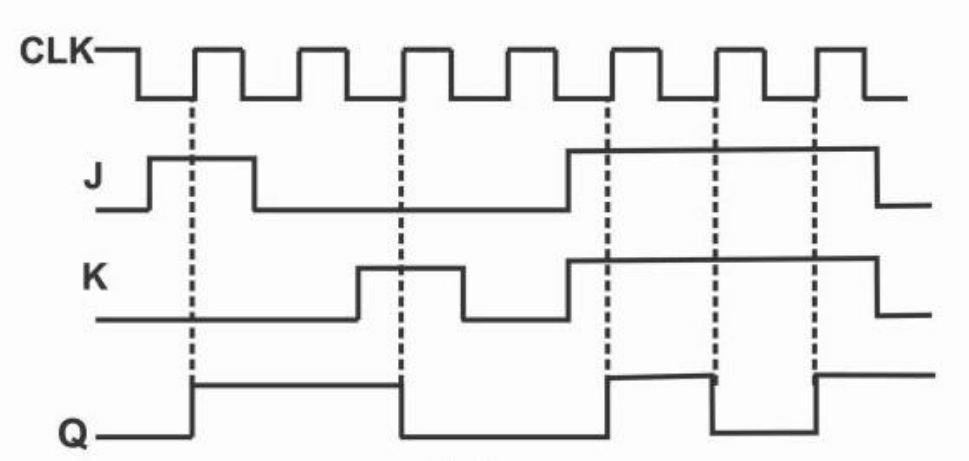
JK Flip-Flop Truth Table:

Here is the truth table for a positive-edge triggered JK flip-flop:

J	K	CLK	Q(n)	Q(n+1)
0	0	↑	Q(n)	Q(n)
0	1	↑	Q(n)	0
1	0	↑	Q(n)	1
1	1	↑	Q(n)	~Q(n)

In this truth table, Q(n) represents the current state of the flip-flop, Q(n+1) represents the next state, and CLK represents the clock signal.





Code:

Behavioural

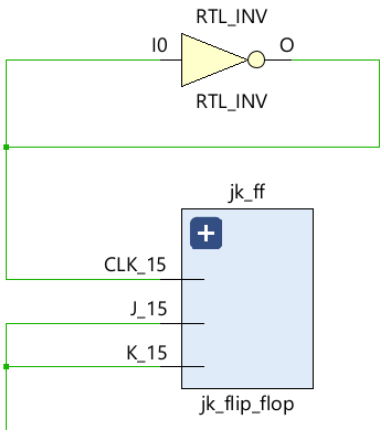
```
module jk_flip_flop (
  input CLK,     // Clock input
  input J,       // J input
  input K,       // K input
  output Q,      // Q output
  output Q_bar   // Q' (Q bar) output
);

  reg Q, Q_bar; // Flip-flop outputs

  always @(posedge CLK) begin
    if (J & ~K)
      Q <= 1'b1;
    else if (~J & K)
      Q <= 1'b0;
  end

  always @(posedge CLK) begin
    if (J & ~K)
      Q_bar <= 1'b0;
    else if (~J & K)
      Q_bar <= 1'b1;
  end

endmodule
```

Testbench:

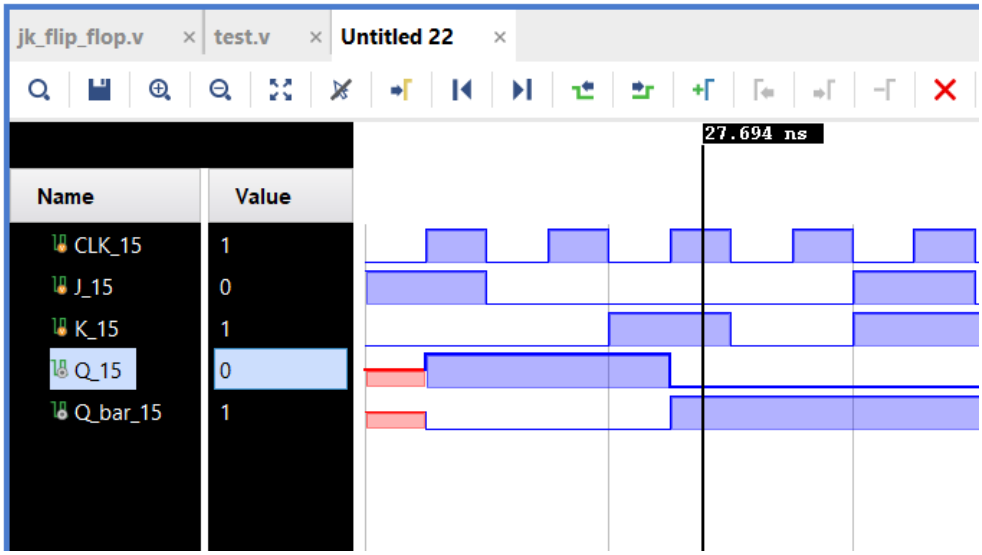
```
module testbench;

 reg CLK, J, K;
 wire Q, Q_bar;

 // Instantiate the JK flip-flop
 jk_flip_flop jk_ff (
   .CLK(CLK),
   .J(J),
   .K(K),
   .Q(Q),
   .Q_bar(Q_bar)
 );

 // Clock generation
 always begin
   #5 CLK = ~CLK;
 end

 // Stimulus generation
 initial begin
   CLK = 0;
   J = 0;
   K = 0;

   // Test case 1: Set Q
   J = 1;
   K = 0;
   #10;
   J = 0;
   #10;

   // Test case 2: Reset Q
   J = 0;
   K = 1;
   #10;
   K = 0;
   #10;

   // Test case 3: Toggle Q
   J = 1;
   K = 1;
   #10;
   J = 0;
   #10;
```

```
    K = 0;
    #10;

    $finish;
  end

  // Display results
  always @(posedge CLK) begin
    $display("Time=%0t J=%b K=%b Q=%b Q_bar=%b", $time, J, K, Q,
Q_bar);
  end

endmodule
```

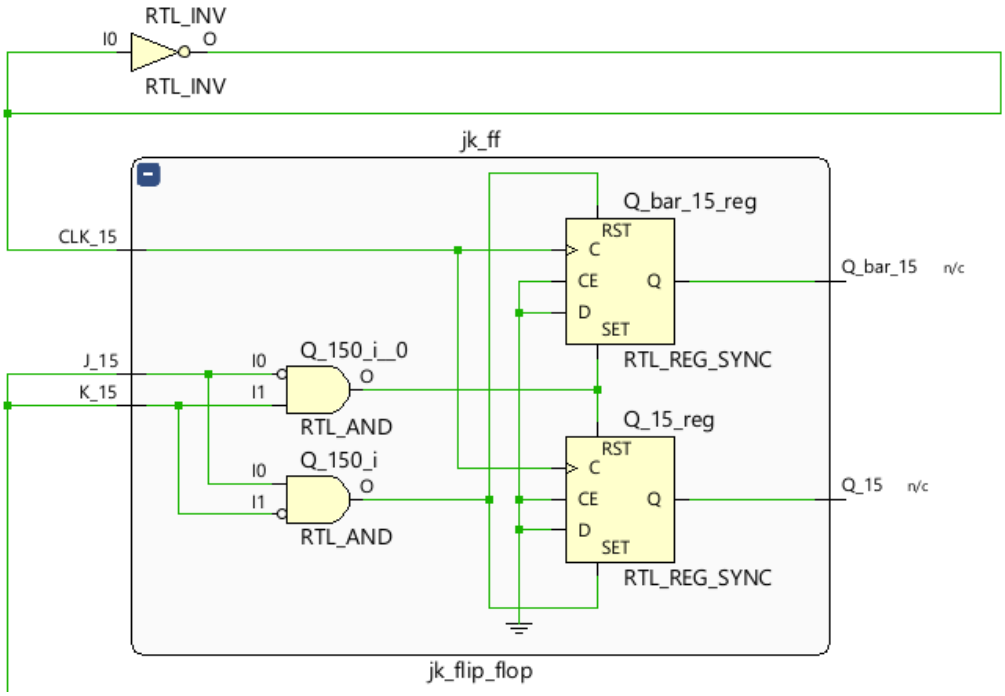
RTL Schematic:



Function Verification:



Post synthesis schematic:



Post power report:

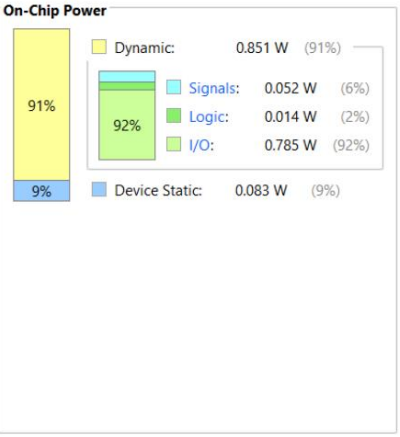
Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**0.934 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**
Junction Temperature:	**26.8°C**
Thermal Margin:	58.2°C (30.7 W)
Ambient Temperature:	25.0 °C
Effective θJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power



Post synthesis timing summary:

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	∞	3	2	2	J_15	Q_bar15	5.836	3.399	2.437	∞	input port clock
Path 2	∞	3	2	2	K15	Q15	5.700	3.315	2.384	∞	input port clock

Post utilization/ area summary:

Hierarchy	Name
Summary	testbench
Slice Logic	

Conclusion:

- In this experiment, we written Verilog code for Positive edge trigger JK Flip flop and we have also written code for its testbench and verified its outputs using a testbench.