

Day 19 Topic: Verilog codes for JK FlipFlop Using Cadence (Xcelium)

◆ What is a Flip-Flop?

- A flip-flop is a small memory element in digital electronics.
 - It can store 1 bit of data: either 0 (LOW) or 1 (HIGH).
 - It is used in registers, counters, and memory circuits.
-

JK Flip-Flop Description

A JK flip-flop is a type of edge-triggered sequential circuit.

It is the clocked version of the JK latch. While a JK latch is level-sensitive (output changes as long as Enable = 1), a JK flip-flop is edge-sensitive (output changes only at the rising or falling edge of the clock pulse).

This makes it more reliable in synchronous digital systems.

Inputs and Outputs

- Inputs:
 - J (Set input)
 - K (Reset input)
 - Clock (CLK) → edge-triggered (\uparrow or \downarrow depending on design)
 - Optional Preset & Clear (asynchronous inputs in practical ICs)
- Outputs:
 - Q (main output)
 - Q' (complement output)

Working

The JK flip-flop works similarly to the JK latch, but it updates the output only at the clock edge.

- J = 0, K = 0 → No change (Q holds previous state)
- J = 0, K = 1 → Reset (Q = 0, Q' = 1)

- $J = 1, K = 0 \rightarrow \text{Set } (Q = 1, Q' = 0)$
 - $J = 1, K = 1 \rightarrow \text{Toggle } (Q \text{ switches to opposite state})$
-

Truth Table (Edge-Triggered)

CLK Edge J K Next State (Q^{n+1})

$\uparrow / \downarrow \quad 0\ 0\ Q^n$ (no change)

$\uparrow / \downarrow \quad 0\ 1\ 0$ (reset)

$\uparrow / \downarrow \quad 1\ 0\ 1$ (set)

$\uparrow / \downarrow \quad 1\ 1\ \bar{Q}^n$ (toggle)

(\uparrow means rising-edge triggered, \downarrow means falling-edge triggered depending on design)

Block Diagram

- The JK flip-flop is implemented using:
 - Master-Slave arrangement of JK latches (to avoid race conditions).
 - Or directly as an edge-triggered circuit.
- ♦ 1.1) Design Code

```
jk_ff.v
// JK Flip-Flop
module jk_ff (
    input wire J, K, CLK,
    output reg Q,
    output wire Qbar
);
    assign Qbar = ~Q;

    always @(posedge CLK) begin
        case ({J,K})
            2'b00: Q <= Q;           // Hold
            2'b01: Q <= 0;          // Reset
            2'b10: Q <= 1;          // Set
            2'b11: Q <= ~Q;         // Toggle
        endcase
    end
endmodule
```

◆ 1.2) Test Bench Code

```
tb.v
module tb_jk_ff;
reg J, K, CLK;
wire Q, Qbar;

// Instantiate JK FF
jk_ff uut (.J(J), .K(K), .CLK(CLK), .Q(Q), .Qbar(Qbar));

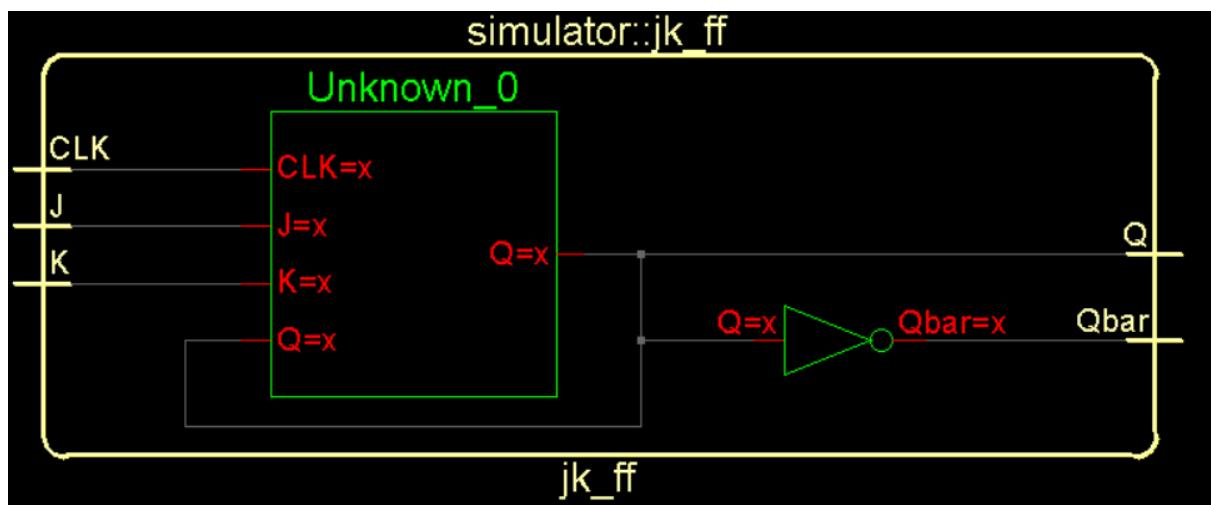
// Clock generation: 10ns period
initial begin
    CLK = 0;
    forever #5 CLK = ~CLK;
end

initial begin
    $monitor("T=%0t | CLK=%b J=%b K=%b | Q=%b Qbar=%b", $time, CLK, J, K, Q, Qbar);

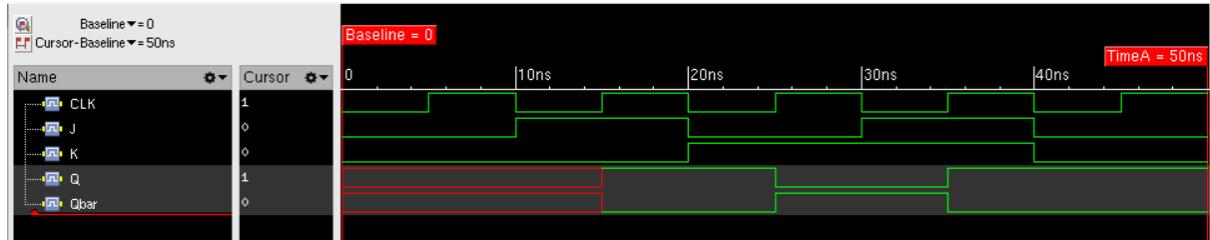
    // Initialize
    J=0; K=0; #10; // Hold
    J=1; K=0; #10; // Set
    J=0; K=1; #10; // Reset
    J=1; K=1; #10; // Toggle
    J=0; K=0; #10; // Hold again

    $finish;
end
endmodule
```

◆ 1.3) Schematic



◆ 1.4) Wave Forms



Applications

- Counters: Since it toggles, JK flip-flop is widely used in binary counters (asynchronous & synchronous).
- Shift Registers: Used to design sequential data storage.
- Frequency Division: A chain of JK flip-flops divides frequency by powers of 2.
- Finite State Machines: Useful in sequential control circuits.