

Day 18 Topic: Verilog codes for SR 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.
-

SR Flip-Flop Description

An SR flip-flop (Set-Reset flip-flop) is one of the basic edge-triggered sequential circuits.

It is the clocked version of the SR latch. While the latch is level-sensitive, the SR flip-flop is edge-sensitive (output changes only on the rising or falling edge of the clock).

It is used to store one bit of information.

Inputs and Outputs

- Inputs:
 - S (Set input)
 - R (Reset input)
 - Clock (CLK) – edge-triggered (rising ↑ or falling ↓ depending on design)
 - Outputs:
 - Q (main output)
 - Q' (complement of Q)
-

Working

The SR flip-flop responds to S and R inputs only at the clock edge.

1. S = 0, R = 0 → No change (Q holds its previous state)
2. S = 0, R = 1 → Reset (Q = 0, Q' = 1)
3. S = 1, R = 0 → Set (Q = 1, Q' = 0)

4. $S = 1, R = 1 \rightarrow$ Invalid/Forbidden state (both outputs become uncertain \rightarrow not allowed)
-

Truth Table (Edge-Triggered)

CLK Edge S R 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)

$\uparrow / \downarrow \quad 1\ 1$ Invalid

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

Block Diagram

- Implemented using two cross-coupled NAND or NOR gates.
 - A clock input is added to make it edge-triggered.
- ◆ 1.1) Design Code

```
sr_ff.v
// SR Flip-Flop (Edge Triggered)
module sr_ff (
    input wire S, R, CLK,
    output reg Q,
    output wire Qbar
);
    assign Qbar = ~Q;

    always @ (posedge CLK) begin
        if (S && !R)
            Q <= 1;           // Set
        else if (!S && R)
            Q <= 0;           // Reset
        else if (!S && !R)
            Q <= Q;           // Hold
        else
            Q <= 1'bX;       // Invalid
    end
endmodule
```

◆ 1.2) Test Bench Code

```
tb.v

module tb_sr_ff;
    reg S, R, CLK;
    wire Q, Qbar;

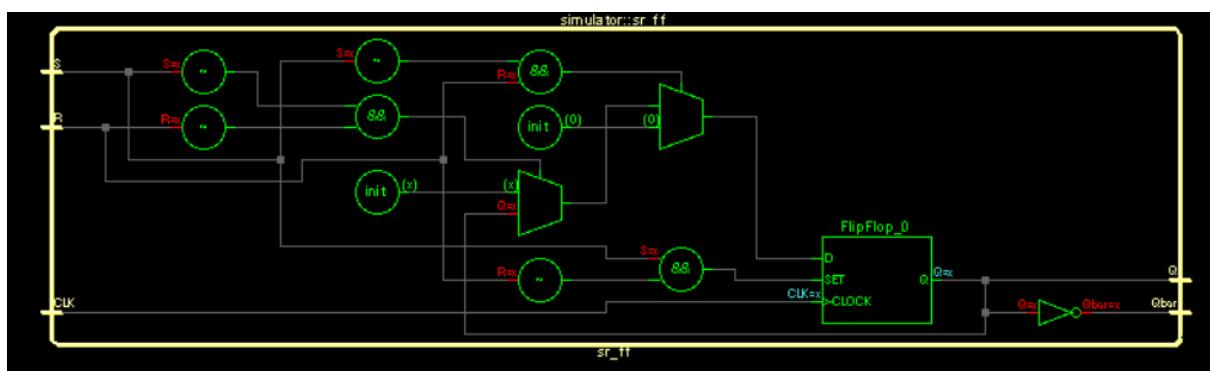
    sr_ff uut (.S(S), .R(R), .CLK(CLK), .Q(Q), .Qbar(Qbar));

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

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

        S=0; R=0; #10;
        S=1; R=0; #10; // Set
        S=0; R=0; #10; // Hold
        S=0; R=1; #10; // Reset
        S=1; R=1; #10; // Invalid
        $finish;
    end
endmodule
```

◆ 1.3) Schematic



◆ 1.4) Wave Forms



Applications

- Basic Storage: Stores one bit of data.
- Control Circuits: Used in control units where set/reset operations are needed.
- Registers & Memory: Used as a building block for larger memory devices.
- Simple State Machines: Provides set and reset functionality in sequential systems.

