📘 **Day 14 Topic: Verilog codes for JK Latch Using Cadence (Xcelium)**

🔹 What is a Latch?

A latch is the simplest form of a memory element in digital electronics.

- It can store 1 bit of data.
- It is level-sensitive ➔ works as long as the Enable (or Clock) input is active.
- Unlike flip-flops (which are edge-triggered), latches are transparent when enabled.

🔹 D Latch

The D latch is also called a Data latch or Transparent latch.

👉 It is designed to overcome the invalid state problem of the SR latch.

How?

- In SR latch, when S=1 and R=1, it was invalid.
- The D latch removes this problem by having only one input (D).
- Internally, D latch ensures that R = NOT(D) and S = D.

So you only give one input (D), and the latch stores it safely.

🔹 Inputs and Outputs

- Inputs:
  - D (Data input)
  - Enable (or Clock/Control input, often called G or EN)
- Outputs:
  - Q (main output)
  - Q̅ (complement output)

🔹 Working (Truth Table)

When Enable = 1 (latch is open/transparent):

Enable	D	Q(next)	Description
0	X	Q(prev)	Hold (no change, latch is closed)
1	0	0	Reset (Q = 0)
1	1	1	Set (Q = 1)

👉 When Enable = 1 → Q follows D ("transparent mode").
👉 When Enable = 0 → Q holds its last value ("latched mode").

🔹 Symbol of D Latch

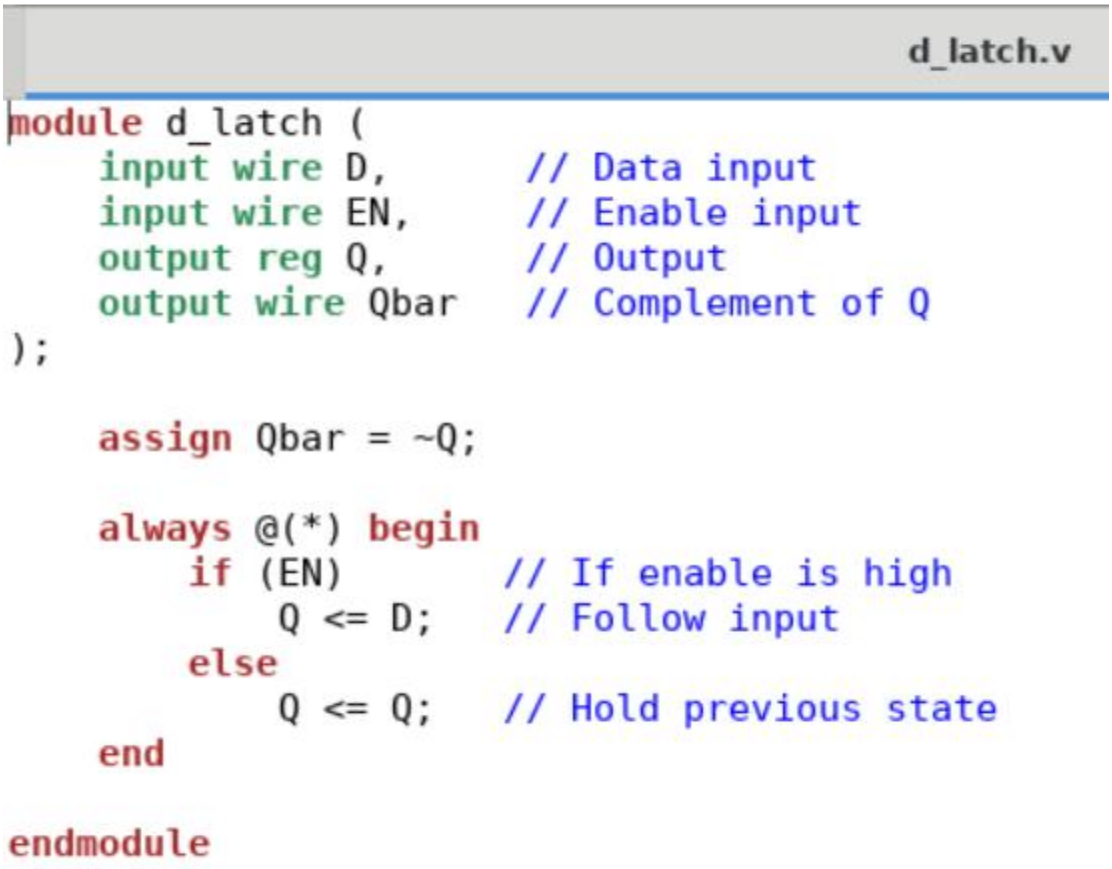
- Inputs: D, Enable
- Outputs: Q, Q̅
- Often drawn as a block with D and EN on the left, Q/Q̅ on the right.

🔹 Why is it called "Transparent Latch"?

Because:

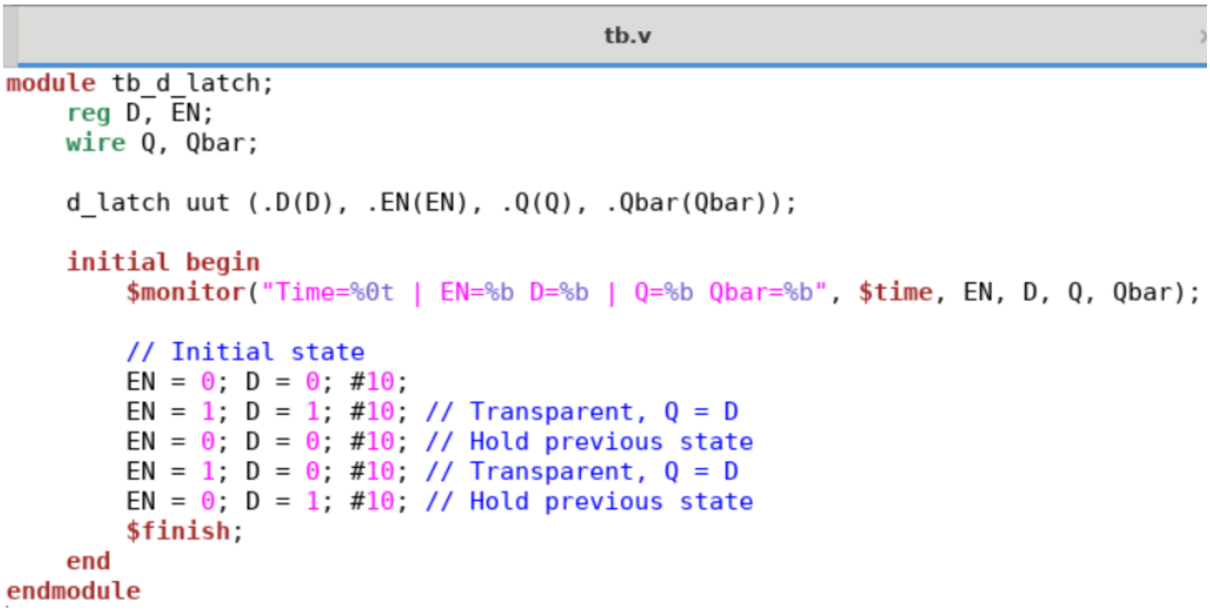
- When Enable = 1, the output Q is directly equal to input D → it looks "transparent".
- When Enable = 0, the latch becomes opaque and holds the last value.

🔹 **1.1) Design Code**



```verilog
module d_latch (
    input wire D,       // Data input
    input wire EN,      // Enable input
    output reg Q,       // Output
    output wire Qbar    // Complement of Q
);

    assign Qbar = ~Q;

    always @(*) begin
        if (EN)       // If enable is high
            Q <= D;   // Follow input
        else
            Q <= Q;   // Hold previous state
    end

endmodule
```

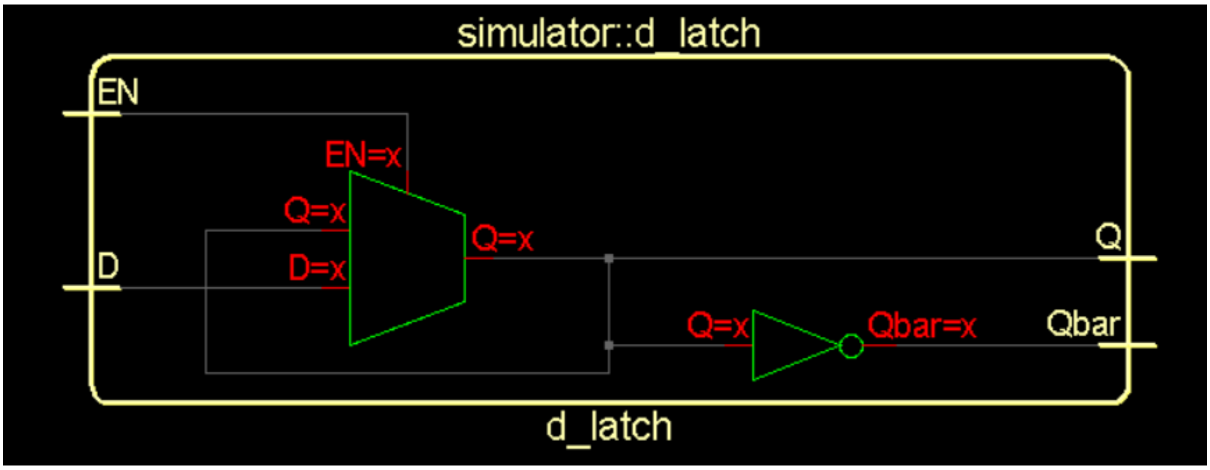
- **1.2) Test Bench Code**



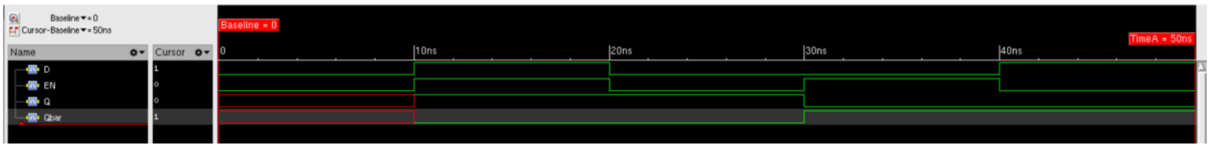
tb.v

```
module tb_d_latch;
    reg D, EN;
    wire Q, Qbar;

    d_latch uut (.D(D), .EN(EN), .Q(Q), .Qbar(Qbar));

    initial begin
        $monitor("Time=%0t | EN=%b D=%b | Q=%b Qbar=%b", $time, EN, D, Q, Qbar);

        // Initial state
        EN = 0; D = 0; #10;
        EN = 1; D = 1; #10; // Transparent, Q = D
        EN = 0; D = 0; #10; // Hold previous state
        EN = 1; D = 0; #10; // Transparent, Q = D
        EN = 0; D = 1; #10; // Hold previous state
        $finish;
    end
endmodule
```

1.3) Schematic



1.4) Wave Forms



Applications

- Temporary data storage (1-bit memory).
- Level-sensitive registers.
- Used inside flip-flop construction (e.g., a D flip-flop is built using 2 D latches).
- Delay elements in sequential circuits.