📘 Day 28 Topic: Verilog codes for Asynchronous Up Counter Using Cadence (Xcelium)

Synchronous Up Counter

A Synchronous Up Counter is a counter in which all flip-flops are triggered by the same clock signal simultaneously.
This means all flip-flops change state together on each clock pulse, making the counting process synchronized and faster compared to an asynchronous counter.

Key Points

- All flip-flops share a common clock input.
- The count increases (000 → 001 → 010 → ... → 111) with each clock pulse.
- The next state of each flip-flop depends on the current states of the previous flip-flops.
- No ripple delay, so operation is fast and reliable.
- Usually built using JK or T flip-flops.

Example: 3-bit Synchronous Up Counter

Clock Pulses	Q2	Q1	Q0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

🔹 1.1) Design Code

syn_up_counter.v

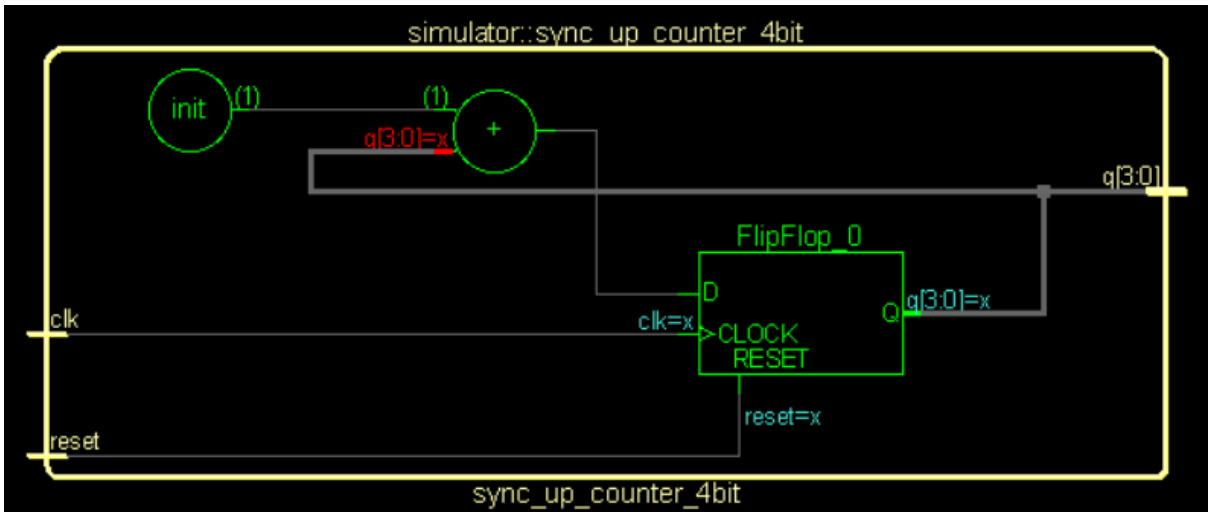
```
// 4-bit Synchronous Up Counter
module sync_up_counter_4bit (
    input clk, reset,
    output reg [3:0] q
);
    always @(posedge clk or posedge reset) begin
        if (reset)
            q <= 4'b0000;
        else
            q <= q + 1'b1;
    end
endmodule
```

- **1.2) Test Bench Code**

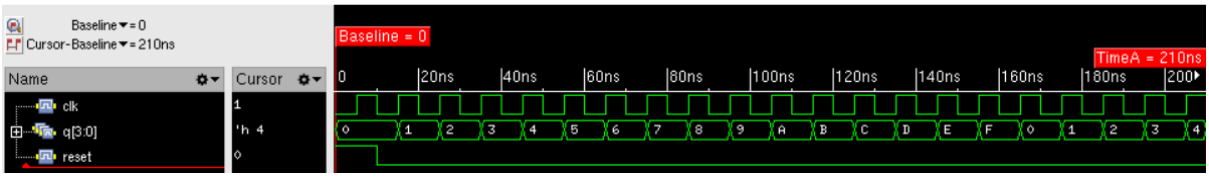
tb.v

```
module tb_sync_up_counter_4bit;
    reg clk, reset;
    wire [3:0] q;

    sync_up_counter_4bit uut (.clk(clk), .reset(reset), .q(q));

    initial begin
        clk = 0;
        forever #5 clk = ~clk;
    end

    initial begin
        reset = 1; #10;
        reset = 0;
        #200 $finish;
    end

    initial begin
        $monitor("Time=%0t | Q=%b", $time, q);
    end
endmodule
```

- **1.3) Schematic**



- **1.4) Wave Forms**



Applications

- **Digital clocks and timers**
- **Frequency dividers**
- **Control circuits in digital systems**
- **Counters in processors and microcontrollers**