📘 **Day 27 Topic: Verilog codes for Asynchronous Up Counter Using Cadence (Xcelium)**

Asynchronous Up Counter

An Asynchronous Up Counter is a type of counter in which the flip-flops do not receive the clock signal simultaneously.
The output of one flip-flop acts as the clock input for the next flip-flop, making it asynchronous (not all triggered together).

It is called an "Up Counter" because the count increases (000 → 001 → 010 → ... → 111) with each clock pulse.

Key Points

- Also known as a Ripple Counter because the clock pulse "ripples" through the flip-flops.
- First flip-flop gets the external clock.
- Each subsequent flip-flop is triggered by the output of the previous one.
- Count sequence goes upward (in binary).
- Simple in design but slow due to propagation delay.

Example: 3-bit Asynchronous 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**

asyn_up_counter.v

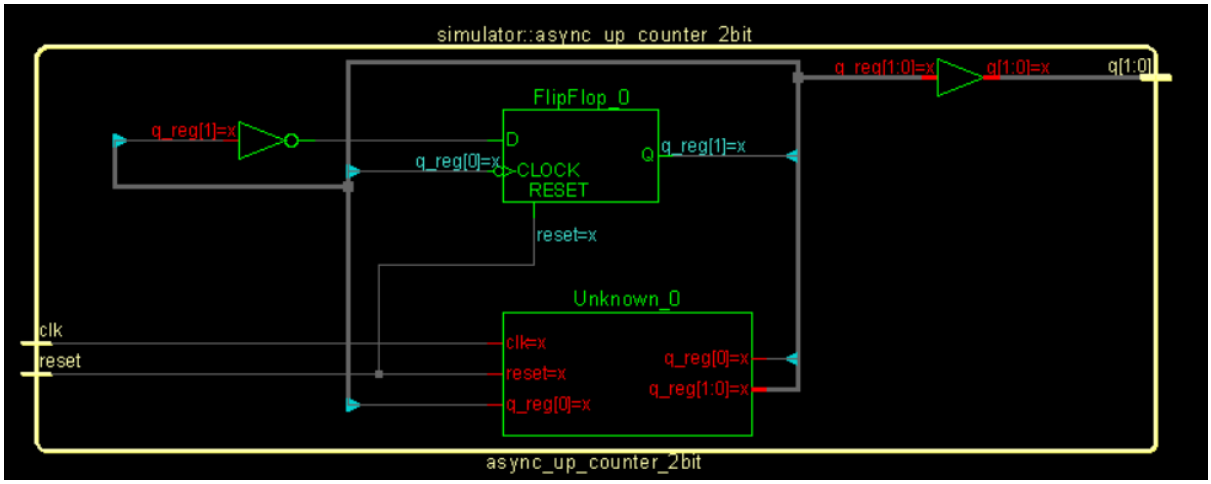
```
// 2-bit Asynchronous Up Counter
module async_up_counter_2bit (
    input clk, reset,
    output [1:0] q
);
    reg [1:0] q_reg;

    always @(posedge clk or posedge reset)
    begin
        if (reset)
            q_reg <= 2'b00;
        else
            q_reg[0] <= ~q_reg[0]; // Toggle FF0
    end

    always @(negedge q_reg[0] or posedge reset)
    begin
        if (reset)
            q_reg[1] <= 1'b0;
        else
            q_reg[1] <= ~q_reg[1]; // Toggle FF1 on nege
    end

    assign q = q_reg;
endmodule
```

- **1.2) Test Bench Code**

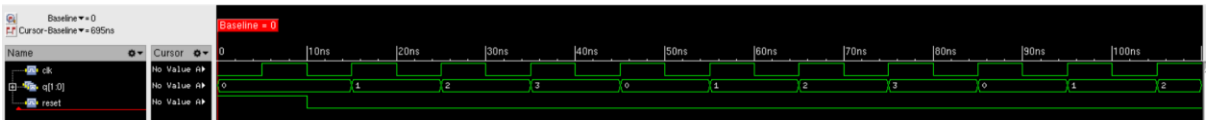
tb.v

```
module tb_async_up_counter_2bit;
    reg clk, reset;
    wire [1:0] q;

    async_up_counter_2bit uut (.clk(clk), .reset(reset), .q(q));

    initial begin
        clk = 0;
        forever #5 clk = ~clk; // Clock with 10ns period
    end

    initial begin
        reset = 1; #10;
        reset = 0;
        #100 $finish;
    end

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

- **1.3) Schematic**



- **1.4) Wave Forms**



Applications

- **Digital clocks**
- **Frequency dividers**
- **Simple event counters**
- **Timers and control circuits**