Sequential Logic Design: Part 2: 4-bit synchronous up counter

Author Adil R.

May 2025

1 Introduction

A 4-bit synchronous up counter is a sequential logic circuit that counts from 0000 to 1111 (0 to 15 in decimal) with all flip-flops changing state simultaneously on the same clock edge. Unlike asynchronous counters, synchronous counters use a common clock signal for all flip-flops and additional combinational logic to determine the next state.

2 Design Description

The 4-bit synchronous up counter implemented using T flip-flops with clock gating.

- All flip-flops share the same clock signal
- State transitions occur simultaneously
- No ripple effect between stages
- Higher maximum operating frequency than asynchronous designs
- Requires additional combinational logic

3 Verilog Implementation

3.1 T Flip-Flop Module (Same as Asynchronous Counter)

```
module t_ff(

    input       clk,
    input       rst,
    output reg q

);
    always @(negedge clk or posedge rst) begin
        if (rst) begin
            q <= 1'b0;
        end
        else begin
            q <= ~q;
        end
    end

endmodule
```

3.2 Synchronous Counter Using T Flip-Flops

```
module sync_counter_4bit(

    input       clk,
    input       rst,
    output [3:0] q

);


    // Toggle enable logic
    assign t1 = q[0];                // Toggle FF1 when Q0 is 1
    assign t2 = q[0] & q[1];         // Toggle FF2 when Q0 and Q1 are 1
    assign t3 = q[0] & q[1] & q[2];  // Toggle FF3 when Q0, Q1, and Q2 are 1

    // Instantiate four T flip-flops with common clock
    t_ff ff0 (
        .clk(clk),
        .rst(rst),
        .q(q[0])
    );

    t_ff ff1 (
        .clk(clk & t1),
        .rst(rst),
        .q(q[1])
    );

    t_ff ff2 (
        .clk(clk & t2),
        .rst(rst),
        .q(q[2])
    );

    t_ff ff3 (
        .clk(clk & t3),
        .rst(rst),
        .q(q[3])
    );

endmodule
```

4 Test Bench

```
`timescale 1ns/1ps

module tb_sync_counter_4bit;

    // Inputs
    reg clk;
    reg rst;

    // Outputs
    wire [3:0] q;

    // Instantiate Unit Under Test
    sync_counter_4bit uut (
        .clk(clk),
        .rst(rst),
        .q(q)
    );

    // Clock generation
```

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

    // Reset control
    initial begin
        rst = 1'b1;
        #10 rst = 1'b0;
        #160 $finish;
    end

    // Monitoring
    initial begin
        $monitor("Time = %0t ns, Q = %4b (%0d)",
                 $time, q, q);
    end

endmodule
```

5 Comparison with Asynchronous Counter

Feature	Synchronous	Asynchronous
Clocking	Single clock for all FFs	Ripple clock between FFs
Timing	Simultaneous transitions	Sequential transitions
Max Frequency	Higher	Lower
Logic Complexity	More complex	Simpler
Power Consumption	Higher	Lower

Table 1: Comparison between synchronous and asynchronous counters

6 Conclusion

- The 4-bit synchronous up counter was successfully implemented using both T flip-flop.
- The simulation results confirm that the counter correctly sequences through all 16 states (0000 to 1111).

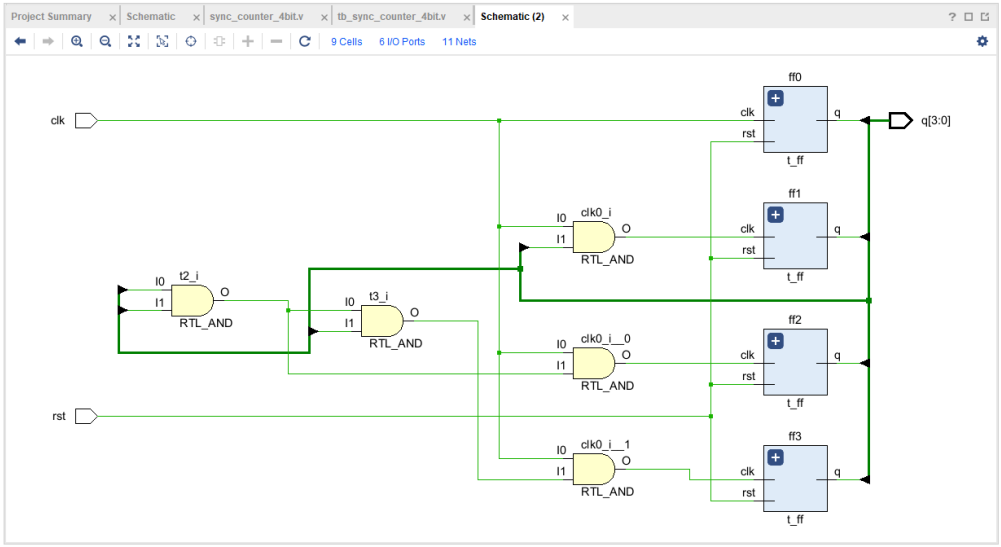

Figure 1: Schematic diagram of the 4-bit synchronous up counter

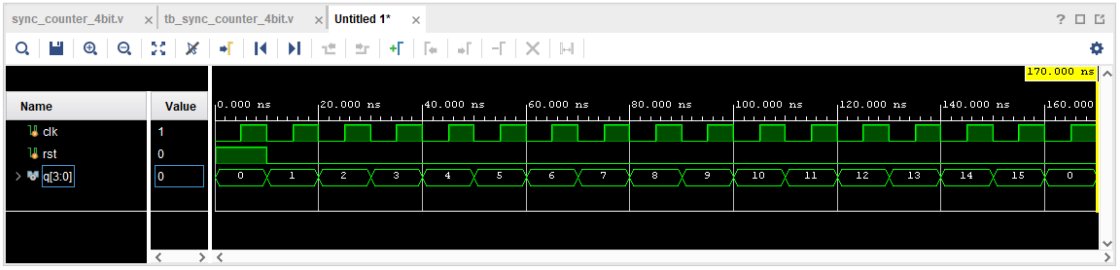

Figure 2: Simulation waveform showing the synchronous counter operation