Sequential Logic Design: Part 1: 4-bit asynchronous up counter

Author Adil R.

May 2025

1 Introduction

A 4-bit asynchronous up counter is a sequential logic circuit that counts from 0000 to 1111 (0 to 15 in decimal) and then rolls over to 0000. Unlike synchronous counters, in asynchronous counters (also called ripple counters), the flip-flops are not clocked simultaneously. The clock signal ripples through the flip-flops, with each subsequent flip-flop being triggered by the output of the previous one.

2 Design Description

The 4-bit asynchronous up counter is constructed using four T flip-flops connected in series. The key characteristics are:

- Each flip-flop divides the frequency of the previous stage by 2
- The output of each flip-flop serves as the clock input for the next flip-flop
- All flip-flops are negative-edge triggered
- The counter exhibits a ripple effect as the clock propagates through the chain

2.1 Truth Table

Q3	Q2	Q1	Q0	Decimal
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Table 1: 4-bit asynchronous up counter output sequence

3 Verilog Implementation

3.1 T Flip-Flop Module

```
1
2  module t_ff(
3
4      input       clk,
5      input       rst,
6      output reg q
7
8  );
9
10     always @(negedge clk or posedge rst) begin
11         if (rst) begin
12             q <= 1'b0;
13         end
14         else begin
15             q <= ~q;
16         end
17     end
18
19 endmodule
```

3.2 Top-Level Counter Module

```
1
2
3  module async_counter_4bit(
4
5      input        clk,
6      input        rst,
7      output [3:0] q
8
9  );
10     // Instantiate four T flip-flops
11     t_ff ff0 (
12         .clk(clk),
13         .rst(rst),
14         .q(q[0])
15     );
16
17     t_ff ff1 (
18         .clk(q[0]),
19         .rst(rst),
20         .q(q[1])
21     );
22
23     t_ff ff2 (
24         .clk(q[1]),
25         .rst(rst),
26         .q(q[2])
27     );
28
29     t_ff ff3 (
30         .clk(q[2]),
31         .rst(rst),
32         .q(q[3])
33     );
34
35
36 endmodule
```

4 Test Bench

```
`timescale 1ns/1ps

module tb_async_counter_4bit;
    // Inputs
    reg clk;
    reg rst;

    // Outputs
    wire [3:0] q;

    // Instantiate Unit Under Test
    async_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;  // Active high reset
        #10 rst = 1'b0;
        #160 $finish;
    end

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

5 Conclusion

- The 4-bit asynchronous up counter was successfully designed and implemented in Verilog HDL.
- The simulation results confirm that the counter correctly sequences through all 16 states (0000 to 1111) before rolling over.

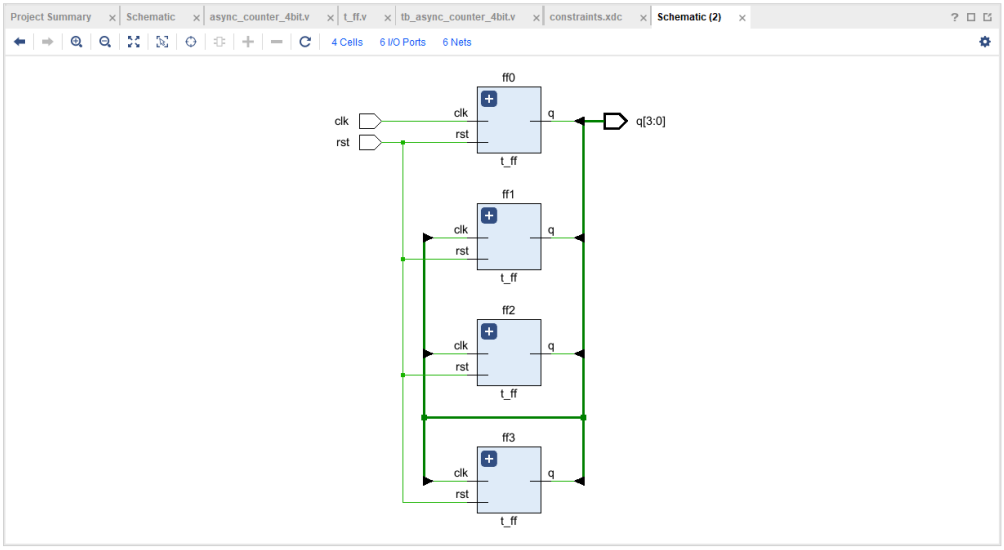

Figure 1: Schematic diagram of the 4-bit asynchronous up counter

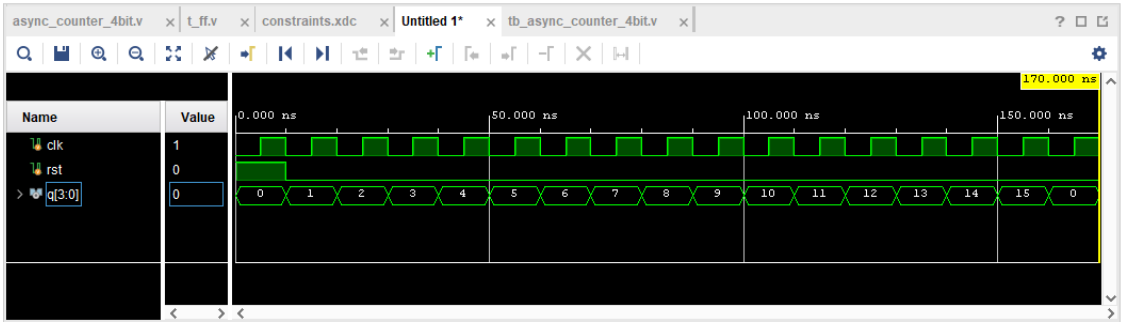

Figure 2: Simulation waveform showing the counter operation