Faculty of Engineering & Technology
Electrical & Computer Engineering Department

Advanced Digital Systems Design ENCS3310

Project Report

Prime/Fibonacci Up/down Counter

Prepared by:

Ghazi Haj Qassem 1210778

Instructor: Abdellatif Abu-Issa

Section: 1

Date: Jul 4th, 2023

Table of Contents

Table of Figures

Introduction and Background

The up-down counter to be designed counts two sequences: prime numbers sequence and Fibonacci sequence. Thus, It is crucial to the design process to understand the mathematics behind the prime numbers and the Fibonacci sequence.

An integer p is said to be a prime number if and only if p is divisible only by itself and the integer 1.

The Fibonacci sequence $\{f_k\}^n_{k=0}$ is defined as: $f_n = f_{n-1} + f_{n-2}$, where $f_1 = 1$ and $f_0 = 0$.

Hence, the sequences for which the counter is to be designed are:

Term Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Prime terms	2	3	5	7	11	13	17	19	23	29	31
Fibonacci Terms	0	1	1	2	3	5	8	13	21	34	55

Design Philosophy

To fulfill the design at hand, from the table of sequences presented in the preceding section, it is useful to break the counter into two stages: the term-index counter and a combinational-logic circuit that maps the term index to its corresponding prime or Fibonacci term, as shown below.

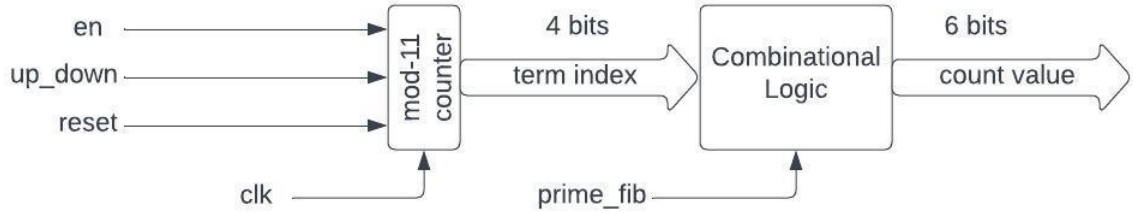

Figure 1: Counter Diagram

First Stage

The idea behind the mod-11 counter is simple. It counts up or down based on the control the syncronous input up_down, it is enabled to count based on a syncrhonous enable input, and resets asynchronously due to the reset input. It is designed using T flip flops with the help of state table, as demonstrated below.

Note that the reset input is active low, and the flip flops are triggered at the positive edge.

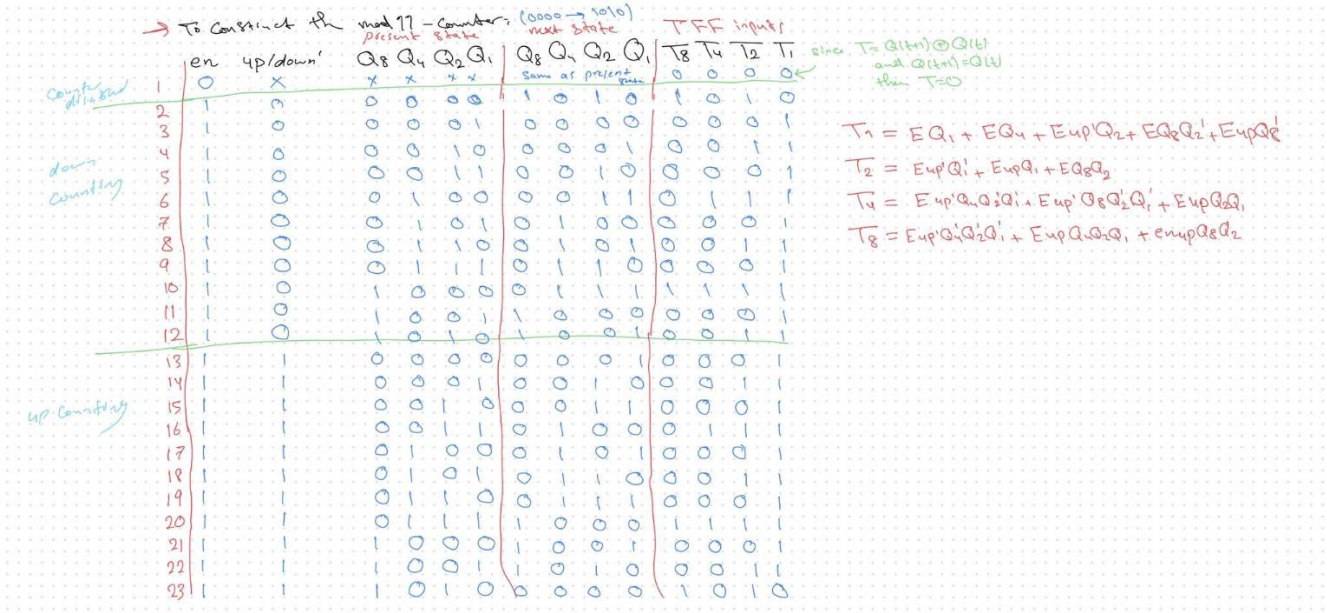

Figure 2: building the first stage of the counter

Second Stage

This is all about mapping the output from the first stage (term index) to the count value, i.e. prime term or Fibonacci term. This is done using a truth table.

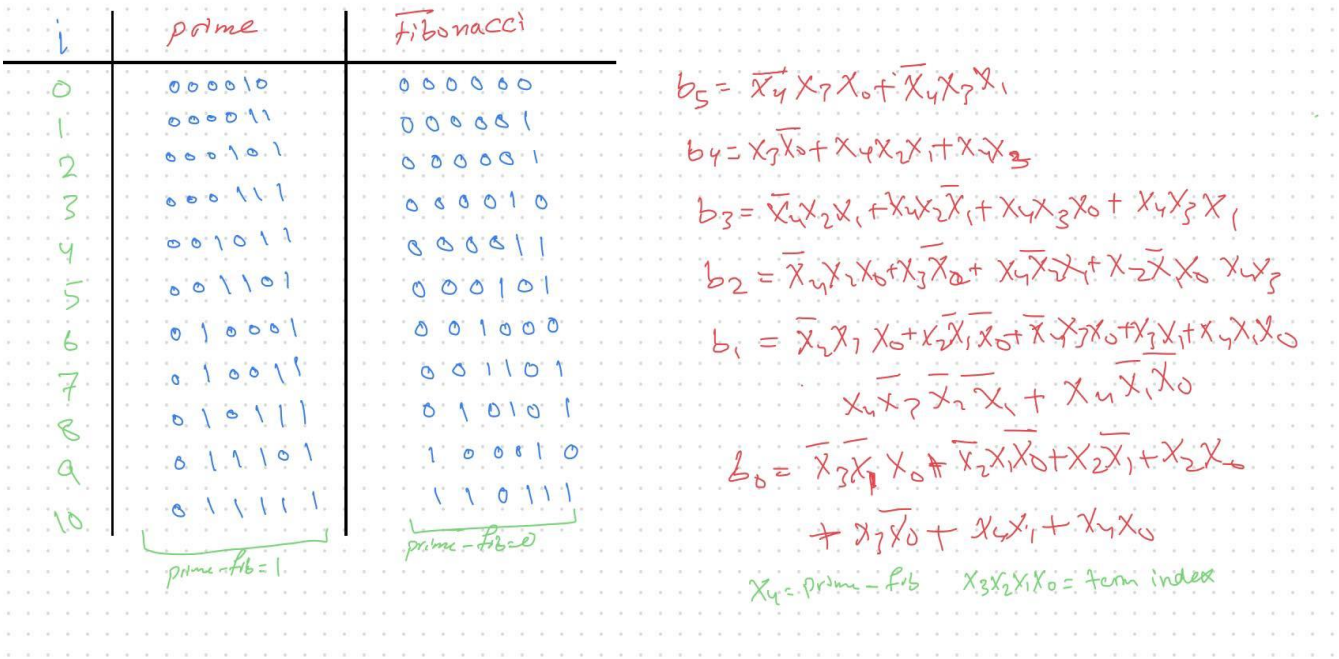

Figure 3: building the second stage of the counter

The inputs (x4, x3, x2, x1, x0) represent (prime_fib, output bits from previous stage), and the outputs (b5, b4, b3, b2, b1, b0) represent the corresponding prime or Fibonacci term bits.

It is important to note that in this design method, the mode control (prime_fib) is asynchronous. However, it can be made synchronous by applying it to be the input to a new flip flop, and the output of the flip flop is the input to the circuit. This ensures that the mode input will change only at the clock edge.

Results

The results below were obtained using a full testbench. It consists of a test generator to generate the expected output based on random test vectors, and a result analyzer that compares the actual output from the design under test to the expected result and prints any errors if the error flag is 1. The error flag is set to 1 when the actual result does not match the expected result.

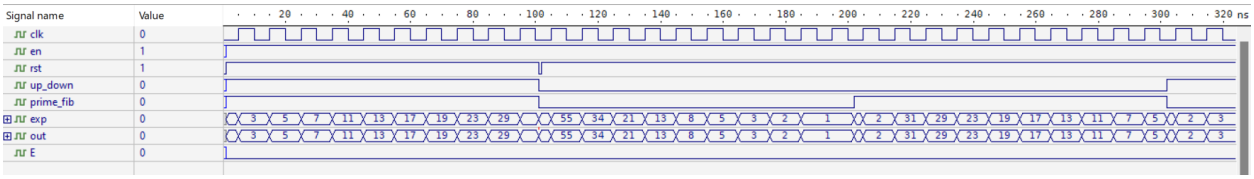

Figure 4: Simulation Results (1)

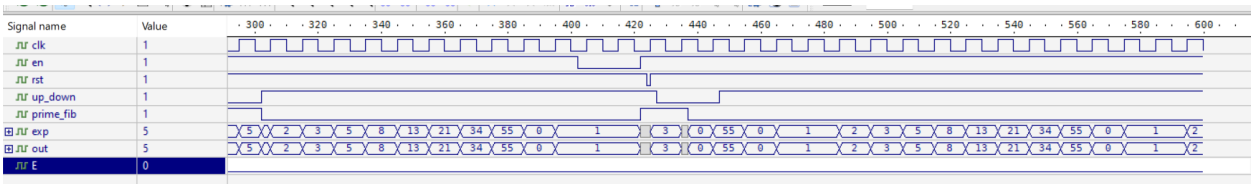

Figure 5: Simulation Results (2)

As observed from the simulation results, where the simulation was run for 600ns, the error flag is always zero. Consequently, no errors are displayed, and the actual result matches the expected result for these random test vectors, which indicates that most likely the design under test is working properly.

However, there are perhaps some corner cases not covered in the random test vectors generated that may produce unexpected results. Therefore, it is useful to run more than a testbench on the same design to ensure its correctness.

Conclusion

In conclusion, the design method was based on the idea of cascaded systems, where a stage produces an output on which the next stage depends. This is beneficial when it comes to the portability of the circuit. For example, the first stage can be replaced to change the number of terms to be counted, whereas the second stage can be replaced to count using different sequences. However, there is a drawback to this approach, where the propagation delay increases. Thus, it should be ensured that the propagation delay of the second stage is less than the period of a clock cycle, or otherwise, the results will be incorrect. As for the results, the circuit passed the testbench that was run, but to ensure its correctness, running multiple testbenches can be useful.