Counters

Submitted to:

Dr. Gregg Chapman

Srinivastin Subramaniyan

Created by:

Group 18

Ansh Pachauri

Rushi Bhatt

Siwei Fan

ECE 2060

The Ohio State University

Columbus, OH

8 March 2024

Executive Summary

The lab explores the design and simulation of counters, focusing on asynchronous and synchronous types. It begins with simulating an asynchronous binary ripple counter and a synchronous decade counter. Next, it extends the button debounce timer by incorporating a stretch timer to observe its effect on button pressing delays. The project concludes with the creation of a pseudo-random number generator using a multiplexor and a simple up counter. This generator produces random sequences displayed on seven-segment displays, facilitating future color association in a Simon game. Through this lab, the team culminates in a thorough understanding of counter design principles and their practical applications in digital systems.

Introduction

Counters play a pivotal role in digital circuit design, enabling the sequential generation of binary numbers essential for various applications such as timing, control, and data processing (Wakerly, 2017). This lab delves into the fundamental concepts of counters by exploring two primary types: asynchronous and synchronous. Asynchronous counters, exemplified by binary ripple counters, rely on individual flip-flops connected in series, where each flip-flop's output triggers the clock input of the next, resulting in a ripple effect (Lala, 2000). On the other hand, synchronous counters, like the decade counter examined here, employ a common clock signal to synchronize the operation of all flip-flops, ensuring simultaneous state transitions and precise timing (Katz, 2010). Through simulations and practical implementations, this lab aims to elucidate the underlying principles of counter design and their significance in digital system development.

Experimental Methodology

The experimental methodology for this lab encompasses three main parts: simulating counters, modifying the button debounce timer, and creating a pseudo-random number generator.

Firstly, the lab begins by simulating two types of counters: asynchronous and synchronous. This involves setting up simulations for an asynchronous binary ripple counter and a synchronous decade counter using ModelSim software. The asynchronous counter demonstrates the ripple effect, where individual flip-flops are connected in series, with each flip-flop's output triggering the clock input of the next. In contrast, the synchronous counter showcases simultaneous state transitions driven by a common clock signal, ensuring precise timing and coordinated operation of all flip-flops. The schematics for the Asynchronous Binary Ripple counter and the Synchronous Decade counter are constructed and simulated to observe their respective functionalities. It is represented in the diagrams.

Secondly, the button debounce timer is modified to incorporate a stretch timer, adding a delay to button presses. This modification involves adjusting the comparator value of the stretch timer to control the duration of the delay. Initially, the effect of the pulse stretch timer is observed with a

one-second delay, followed by recalculating the comparator value to reduce the delay to 150 milliseconds. The debounce circuit with the Stretch Timer schematic is updated, recompiled, and tested to ensure the desired delay is achieved. It is shown in the result sections.

Lastly, a pseudo-random number generator is designed using a multiplexor and a simple up counter. This generator produces random sequences displayed on seven-segment displays, which are temporarily utilized to visualize the generated codes. The generated sequences will later be associated with specific colors in a Simon game. The Pseudo Random Number Generator Circuit schematic is constructed and implemented to validate its functionality.

Overall, the experimental methodology involves a combination of simulation, modification, and implementation to explore the principles of counters, debounce circuits, and random number generation in digital system design.

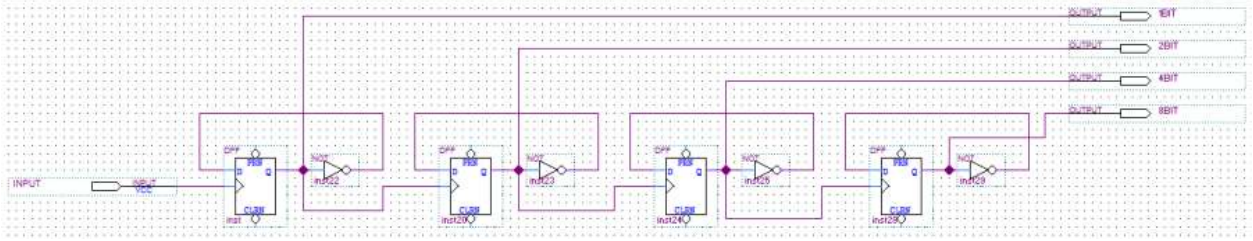

Figure 1 - Asynchronous Binary Ripple Schematic

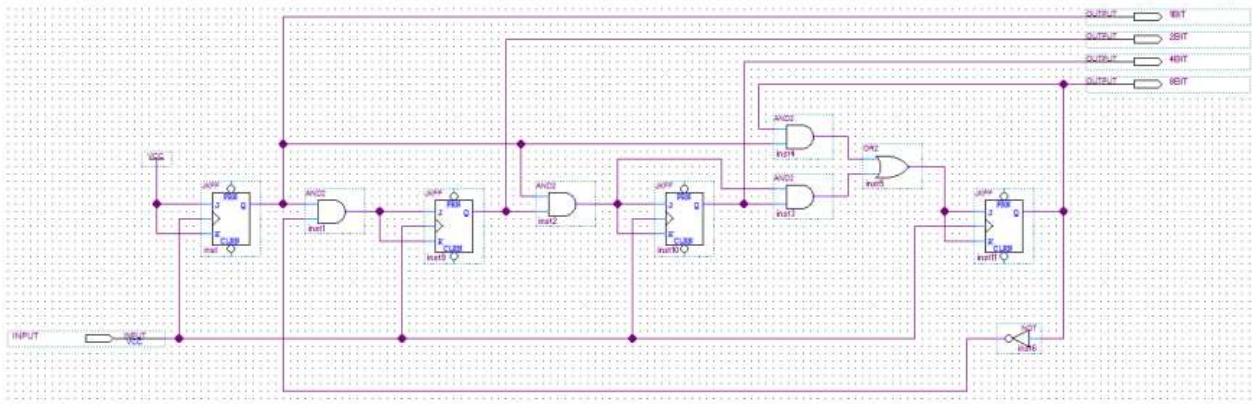

Figure 2 - Synchronous Decade Schematic

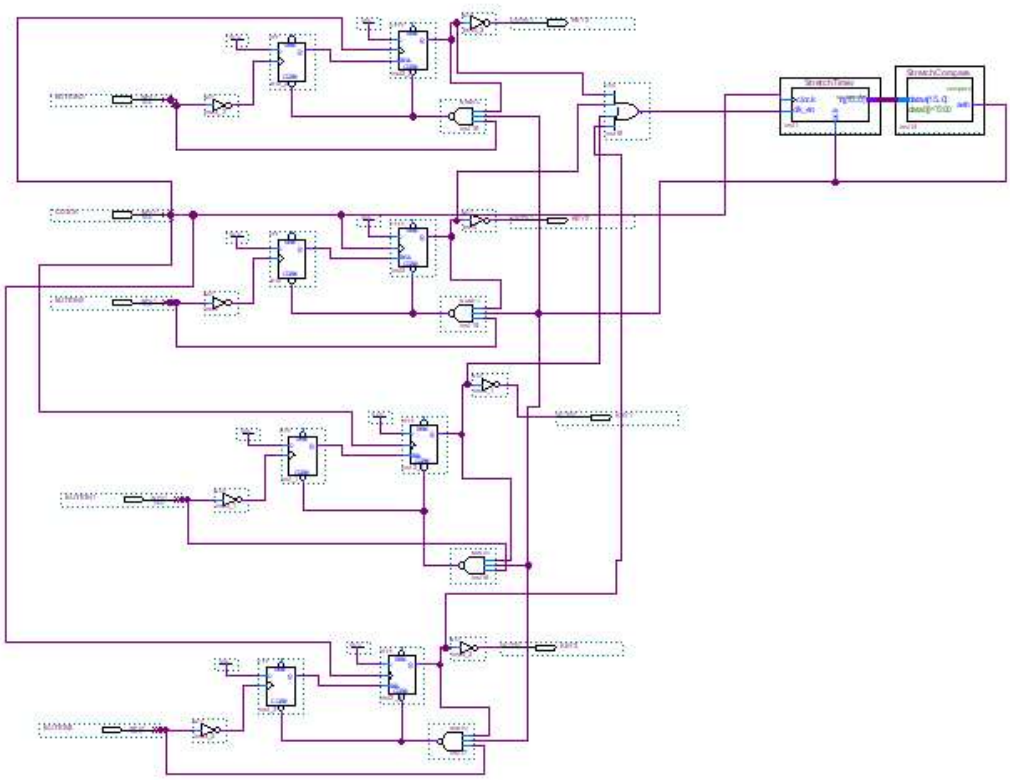

Figure 3 - Button Debounce circuit with Stretch Timer schematic

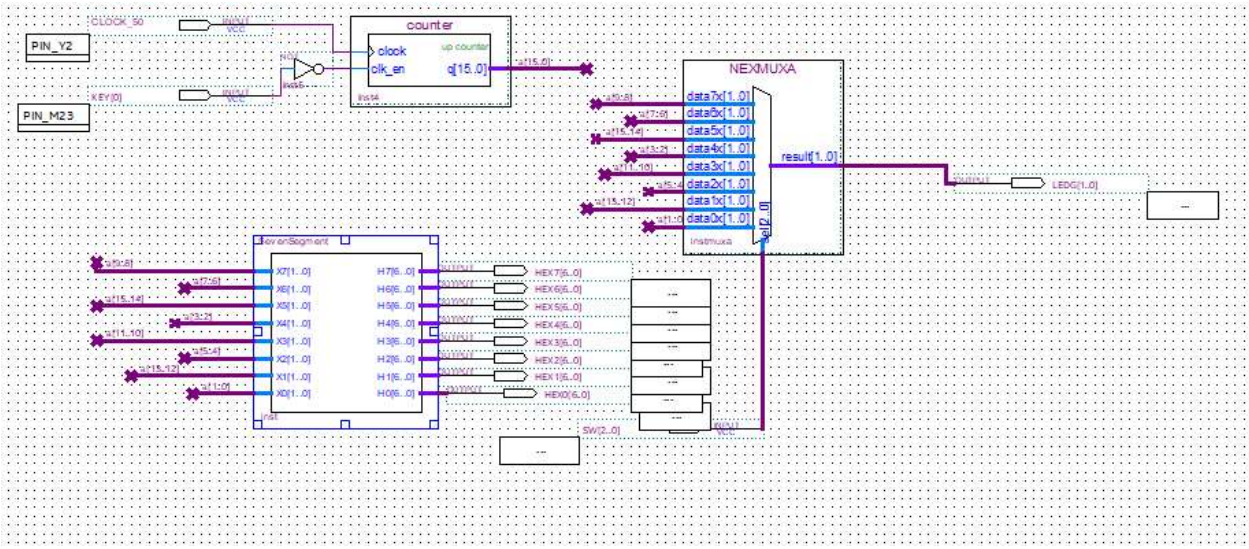

Figure 4 - Pseudo Random Number Generator Circuit

Results

The following diagrams represent the results for the simulation for the lab. As the graphs show, the simulations of the asynchronous binary ripple counter and the synchronous decade counter yielded valuable insights into their respective behaviors. In the asynchronous binary ripple counter simulation, the expected ripple effect was observed, where each flip-flop's output triggered the clock input of the next flip-flop in series, resulting in a sequential counting sequence. This demonstrated the inherent asynchronous nature of the counter, with each stage's transition dependent on the preceding stage's output. Conversely, the synchronous decade counter simulation exhibited synchronized state transitions, driven by a common clock signal. This ensured simultaneous updates across all flip-flops, leading to precise and coordinated counting operations. Additionally, the implementation of the Seven Segment VHDL code facilitated the display of generated sequences on the seven-segment displays, providing a visual representation of the pseudo-random numbers generated by the circuit. Overall, the results highlight the distinct characteristics of asynchronous and synchronous counters, emphasizing their role in digital system design and operation.

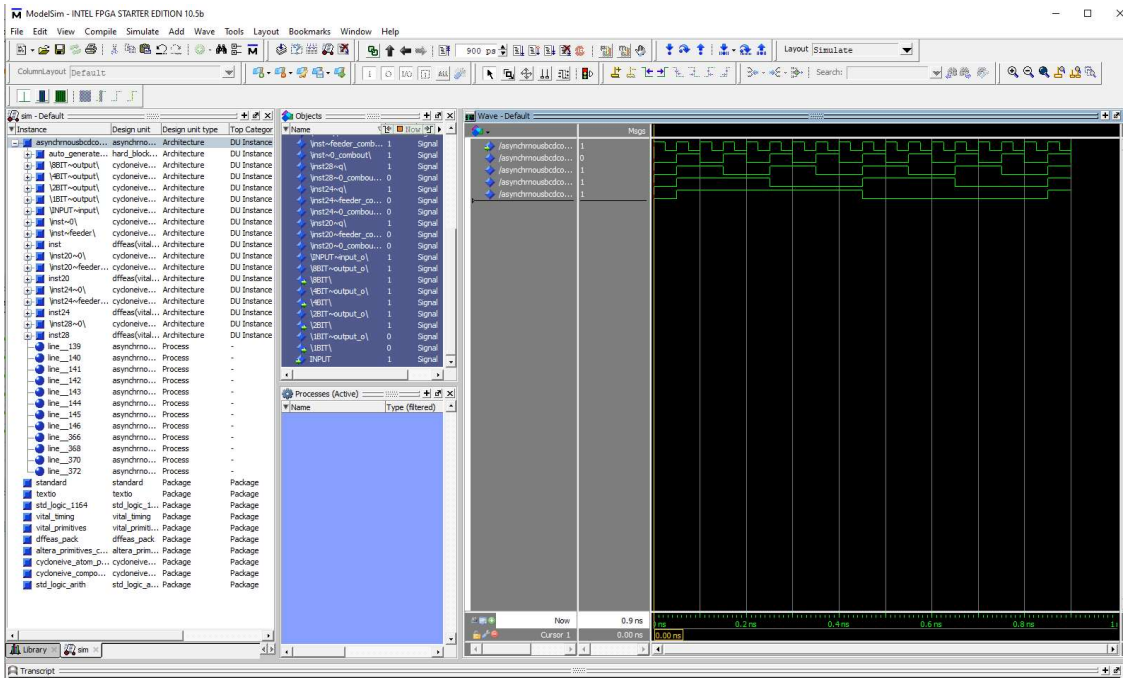

Figure 5 - Asynchronus Binary Ripple Counter Simulation

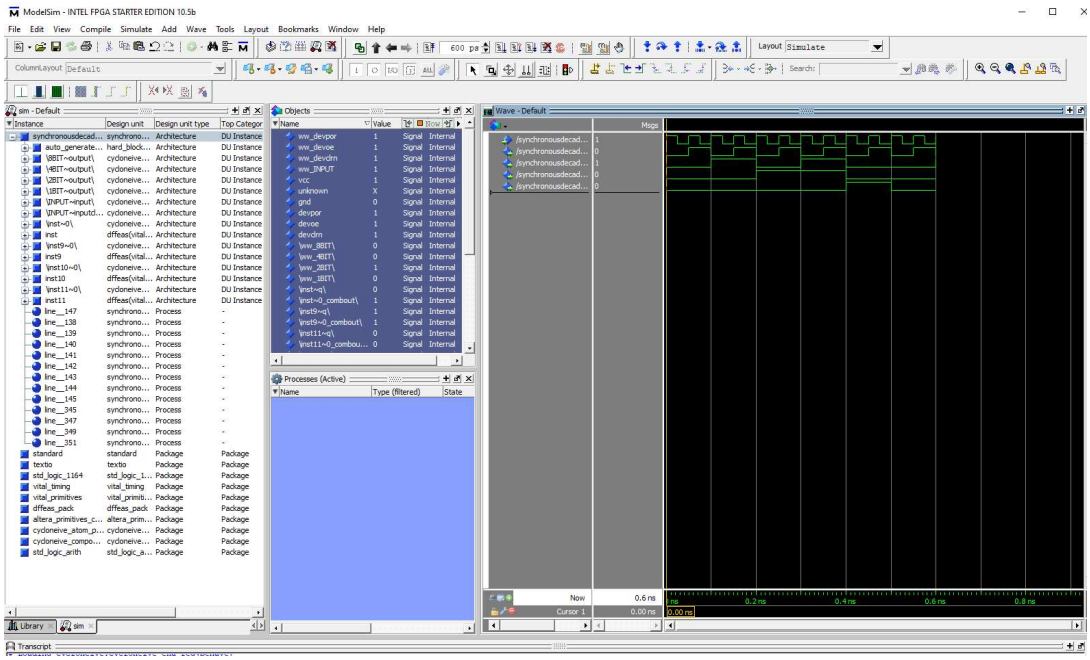

Figure 6 - Synchronous Decade Counter Simulation

```
36
37          H0 <= zero when X0 = "00" else
38                one when X0 = "01" else
39                two when X0 = "10" else
40                three when X0 = "11" else
41                none;
42          H1 <= zero when X1 = "00" else
43                one when X1 = "01" else
44                two when X1 = "10" else
45                three when X1 = "11" else
46                none;
47          H2 <= zero when X2 = "00" else
48                one when X2 = "01" else
49                two when X2 = "10" else
50                three when X2 = "11" else
51                none;
52          H3 <= zero when X3 = "00" else
53                one when X3 = "01" else
54                two when X3 = "10" else
55                three when X3 = "11" else
56                none;
57          H4 <= zero when X4 = "00" else
58                one when X4 = "01" else
59                two when X4 = "10" else
60                three when X4 = "11" else
61                none;
62          H5 <= zero when X5 = "00" else
63                one when X5 = "01" else
64                two when X5 = "10" else
65                three when X5 = "11" else
66                none;
67          H6 <= zero when X6 = "00" else
68                one when X6 = "01" else
69                two when X6 = "10" else
70                three when X6 = "11" else
71                none;
72          H7 <= zero when X7 = "00" else
73                one when X7 = "01" else
74                two when X7 = "10" else
75                three when X7 = "11" else
76                none;
77
78
79    END Behavioral;
```

Figure 7 - Seven Segment VHDL code

Discussion

In VHDL code, the constants "four," "five," and "six" would be represented by std_logic_vector values corresponding to the binary representations of the numbers 4, 5, and 6, respectively, considering the seven-segment display outputs as low true. Specifically, "four" would be '0100', "five" would be '0101', and "six" would be '0110', where each bit signifies the activation or deactivation of specific segments to display the corresponding digit on the seven-segment display. Synchronous circuits operate with all elements synchronized to a common clock signal, facilitating precise timing and coordinated behavior across the circuit, while asynchronous circuits transition between states based on individual component states, potentially leading to unpredictable timing and asynchronous behavior. To achieve a 360 ms delay in the Button Debounce circuit's Stretch timer, given an AUD_DACLRCK clock frequency of 48 kHz, the constant value in the Stretch timer would need to be set to 17280 to correspond to the desired delay, calculated by converting the delay to clock cycles based on the clock frequency.

Conclusion

In conclusion, this lab provided a comprehensive exploration of counters, debounce circuits, and pseudo-random number generators in digital system design. Through simulations and practical implementations, the distinct characteristics of asynchronous and synchronous counters were elucidated, showcasing their respective ripple effect and synchronized state transitions. The modification of the button debounce circuit to include a stretch timer demonstrated the importance of precise timing in digital systems, while the creation of a pseudo-random number generator highlighted the practical applications of sequential logic circuits. The discussion questions further enhanced the team’s understanding for different counters and their role in digital system development. Overall, this lab not only enhanced theoretical knowledge but also fostered practical skills in digital circuit design and simulation.

References

Wakerly, J. F. (2017). Digital Design: Principles and Practices (5th ed.). Pearson.
Lala, P. K. (2000). Practical Digital Logic Design and Testing. Prentice Hall.
Katz, R. H. (2010). Contemporary Logic Design (2nd ed.). Pearson.