LAB 10

Laboratory Report for CS 2420

Brent Johnson

Computer Science

Texas State University

Department of Computer Science

Bj1107@txstate.edu

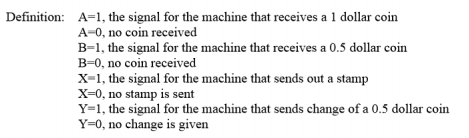
*Abstract* –  This lab was an opportunity for students to implement what they have learned about Mealy and Moore machines. We were told to implement a simple stamp vending machine that accepts dollars and 50 cent pieces. The machine issues change when too much money is inserted.

# Introduction

The theme of today’s lab was independence. There was little assistance provided by the lab instructor. Students were asked to create a finite state machine that represents the vending machine on their own. They also had their own choice between a Mealy or a Moore machine. Independence is important to this stage in our learning because it allows us to accurately gauge how much we actually understand. Also, because most of us hope to become paid professionals, independence is important. No company is going to pay someone to do something and then walk then through it step by step.

# Experimental Method

 To begin this lab, we were given the following instruction: “Design a sequential logic circuit for an automatic ticket vending machine. The machine can only accept a .5 dollar coin or a 1 dollar coin. When the machine receives 2 dollar, it sends out a stamp. If the machine already received 1.5 dollars, and the 1 dollar is given, then it sends out a stamp as well as a change of .5 dollars.” We were then given a chart to tell which input corresponds to which action. The chart is below.



Next I decided to use a Mealy machine implementation. This means that current state along with current input will be considered when deciding next state. First, I drew a diagram that outlines the logical function of the machine.

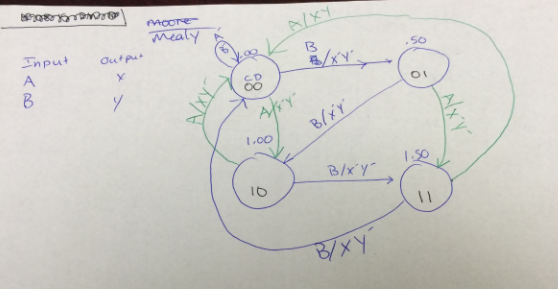
After I drew the outline, I used an excitation table to summarize the logical functions Cnext and Dnext. This allowed me to create Kmaps for Cnext, Dnext, X and Y. These Kmaps allowed me to reduce the equations to their simplest forms.

Next I used these optimized equations to implement my design in DSCH using positive edge triggered D flip flops.

# Results

My first decision was between a Mealy or a Moore machine. I decided to use a Mealy machine because Mealy machine consider current input along with current state to decide next state. This is important because there are two different type of input in this implementation. This means that input should be considered when deciding next state. Although the stamp vending machine could also be implemented using a Moore machine, I believe this would have involved more variable because input in not considered. This is my defense of my reasoning for choosing a Mealy machine over a Moore machine for this implementation.

The first step was the draw a diagram for this machine. The diagram I came up with is as follows:



Then I used the diagram to come up with an excitation table. The excitation table is as follows:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | Cnext | Dnext | X | Y |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

Next I used the excitation table to create Kmaps for the equations Cnext, Dnext, X, and Y. The Kmaps and equations are as follows:

Cnext = AC’ + BC’D + A’B’C + A’CD’

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB\CD | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | 1 | 1 |
| 01 | 0 | 1 | 0 | 1 |
| 11 | X | X | X | X |
| 10 | 1 | 1 | 0 | 0 |

Dnext = BD’ + A’B’D + AC’D

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB\CD | 00 | 01 | 11 | 10 |
| 00 | 0 | 1 | 1 | 0 |
| 01 | 1 | 0 | 0 | 1 |
| 11 | X | X | X | X |
| 10 | 0 | 1 | 0 | 0 |

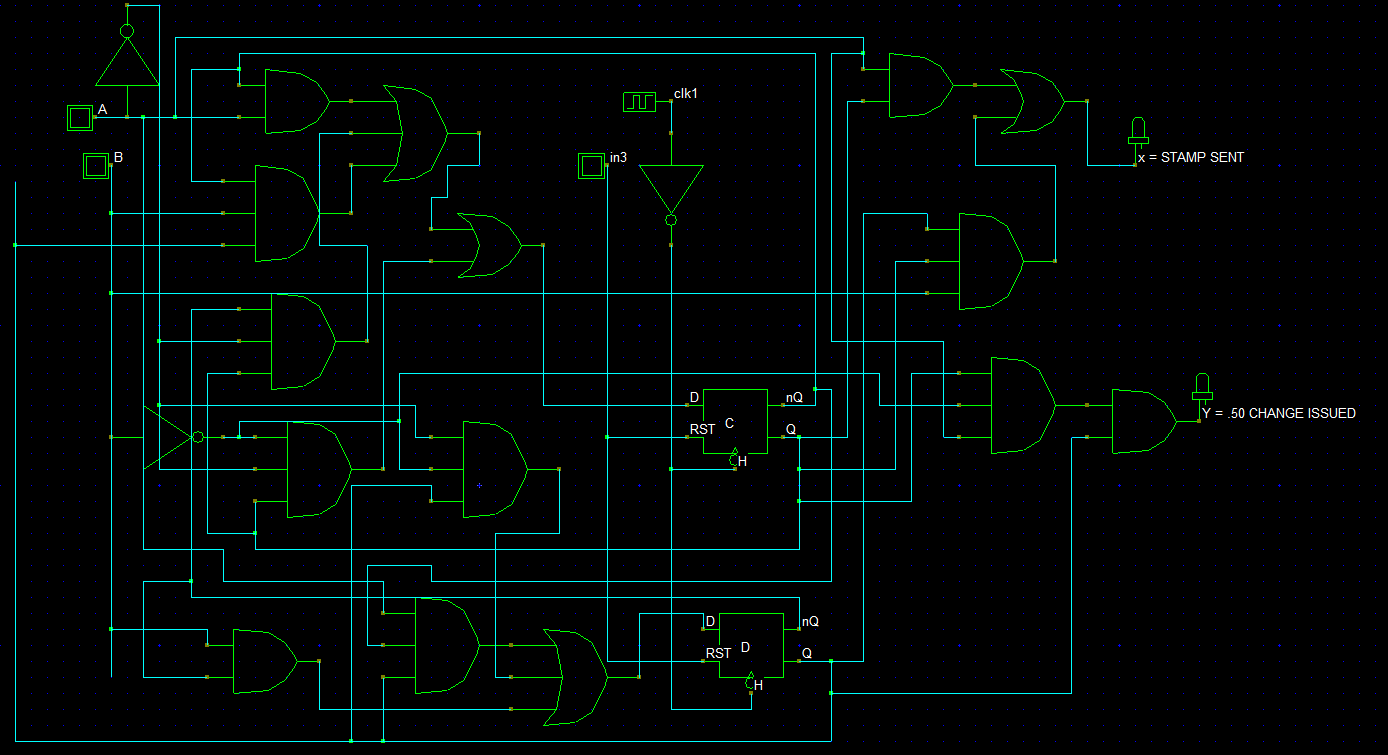
X = AC + BCD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB\CD | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 0 |
| 11 | X | X | X | X |
| 10 | 0 | 0 | 1 | 1 |

I did not make a Kmap for Y because there is only one combination that results in an output of Y. That makes the equations for Y as follows:

Y = AB’CD

Finally, I implemented my Mealy machine using DSCH. I used two positive edge triggered D flip flops for Cnext and Dnext and two simple buttons for input A and B. The output X and Y were tied to led lights to display when they are active. Here is a picture of the schematic I came up with. The schematic itself has also been submitted alongside this report.



# Conclusion

If there is one main problem I noticed with my Mealy implementation, it is the clock. Because the clock is always ticking, it is difficult to troubleshoot my schematic and verify that it is working correctly. If I would have gone with a Moore machine, there would have been more states, meaning more flip flops. However, when troubleshooting I would not have to try to correspond my input to the clock cycle. This lab was an excellent demonstration of how much we (as students) still do not know about digital logic. Good thing there is one more lab to go!