LAB 9

Laboratory Report for CS 2420

Brent Johnson

Computer Science

Texas State University

Department of Computer Science

Bj1107@txstate.edu

*Abstract* – This lab was an introduction to Moore machines. A Moore machine is a finite state machine whose output is determines solely by current state and current input. In other words, previous state is not a factor in determining next state (as it is with a Mealy machine). I learned about transition tables, Kmaps, diagrams, and schematics for Moore machines.

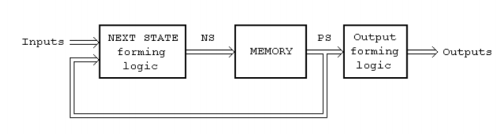
# Introduction

The theme of today’s lab was Moore machines. Moore machines are important because of their importance in clocked sequential systems. Because most modern machines are clocked sequential systems (state only changes with clock signal input), Moore machine logic is very prevalent in today’s digital landscape.

In order to gain understanding surrounding Moore machines, today’s lab focused on the design and implementation of two different Moore machines. This design and implementation includes forming the diagram, transition table, Kmaps, and schematics for each machine.

# Experimental Method

The first objective of the lab was to gain a high-level understanding of the underlying logic at work in a Moore machine. This diagram is a logical representation of what happens in a Moore machine:



Building on this understanding, I began by drawing a diagram of a finite state machine that defines the following machine: A serial device that takes in a binary number and outputs the same number with ever third 1 converted to a 0.

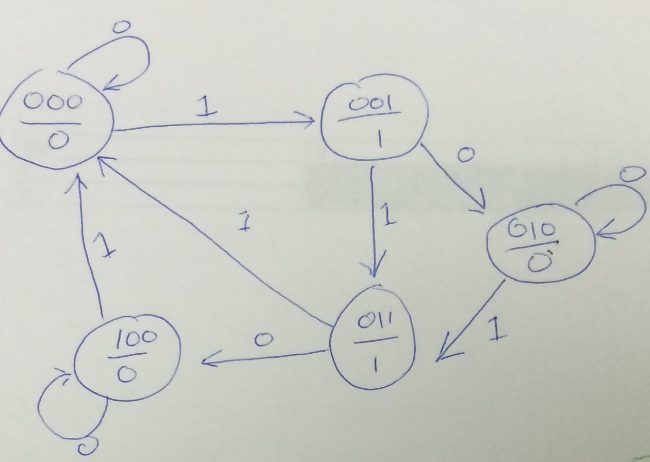
After I drew the diagram, I got to work on the transition table for this device. Next I build Kmaps to reduce the equations for A(next), B(next), and the output. I implemented these equations in DSCH and build the schematics for the machine.

I then repeated this process of drawing a diagram, building a transition table, making Kmaps, and implementing schematics using DSCH for a different machine. This machine counts 3,4,8,9,3,4,8,9 if the input is a 0 and the reversed if the input is a 1.

Finally, I briefly outlined the differences between a Mealy machine and a Moore machine by comparing labs 8 and 9.

# Results

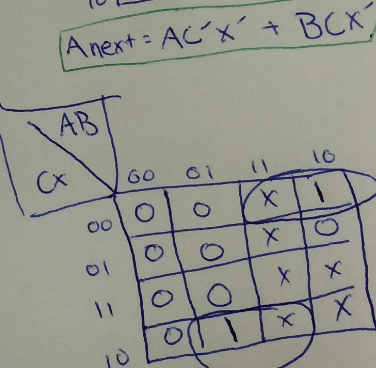
For the first Moore machine, I began by drawing a diagram that outline the machine’s functionality. Here is a copy of my diagram:

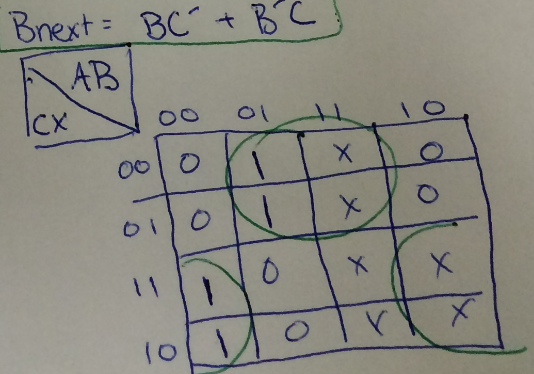


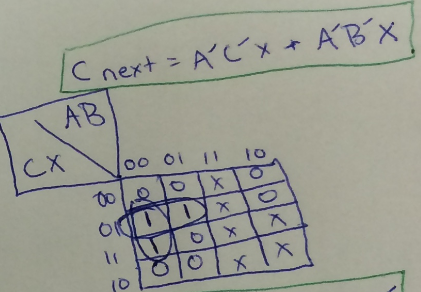
Next I build a transition table based on the functionality outlined in the diagram. Here is a copy of the transition table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| INPUT | | | | OUTPUT | | | |
| A | B | C | X | Anext | Bnext | Cnext | Output |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | X | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X | X |
| 1 | 1 | 1 | 1 | X | X | X | X |

Based on this transition table, I made Kmaps to produce optimal equations for Anext, Bnext, Cnext, and the output. The Kmaps are as follows:







These Kmaps produce the following equations:

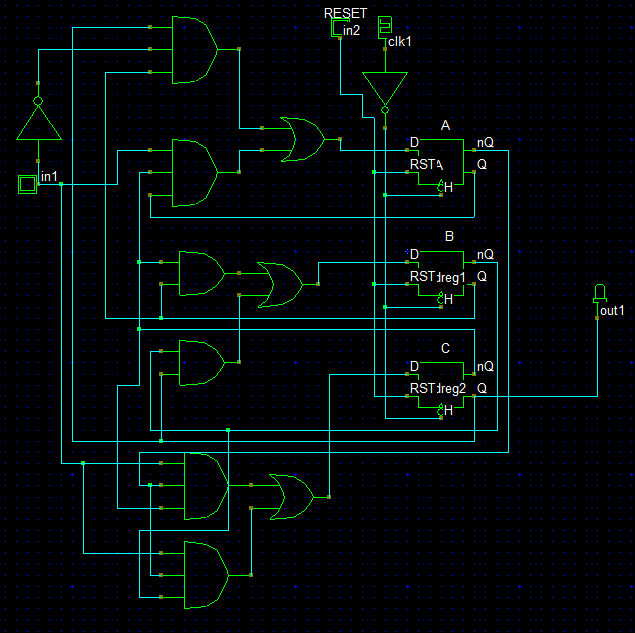
A(next) = AC’X’ + BCX’

B(next) = BC’ + B’C

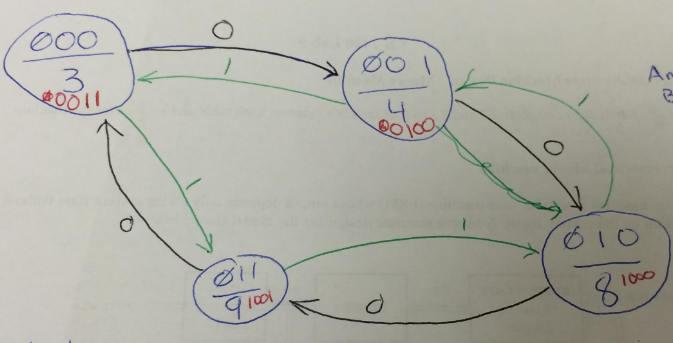
C(next) = A’C’X + A’B’X

Output = C

Using these equations I build a working schematic is DSCH. Here is a picture of my schematic:



After confirming that my schematic was working correctly, I began the process on the other Moore machine. Here is a copy of the diagram I came up with:



Here is the transition table:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | X | A(NEXT) | B(NEXT) | O3 | O2 | O1 | O0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |

Here are the Kmaps:

A(NEXT) = A’B’X + A’BX’ + ABX +

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

B(NEXT) = B’

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

Because the numbers will be displayed using a hex-display, I also made Kmaps for each input on the hex-display.

O0 = A’B’ + AB

|  |  |  |
| --- | --- | --- |
| B\A | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

O1 = A’B’

|  |  |  |
| --- | --- | --- |
| B\A | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |

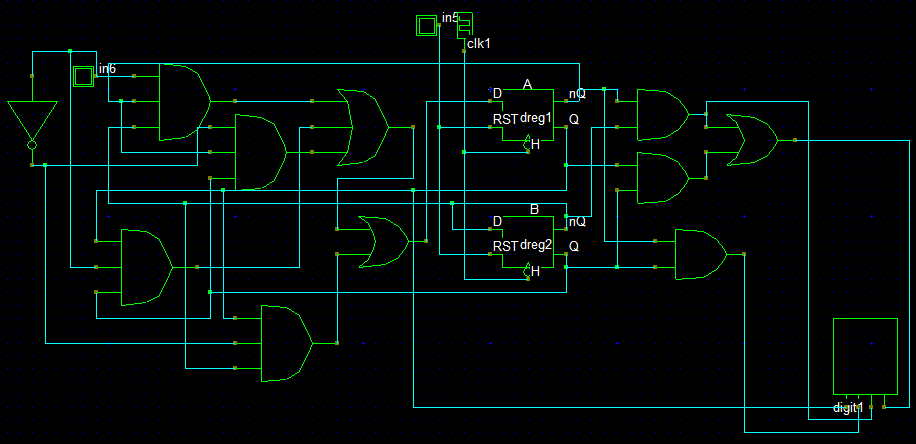
O2 = A’B

|  |  |  |
| --- | --- | --- |
| B\A | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 1 | 0 |

O3 = A

|  |  |  |
| --- | --- | --- |
| B\A | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 0 | 1 |

Using these equations derived from the Kmaps, I implemented this machine using DSCH. Here is a picture of my schematic:



For the final portion of the lab, I analyzed the differences and similarities between Mealy and Moore machines using labs 8 and 9 as references. Conceptually, the difference between a Mealy and Moore machine is whether or not previous state plays a role in determining next state. If previous state in considered, then it is a Mealy machine implementation. If previous state is not considered, it is a Moore machine. Labs 8 and 9 demonstrate that many applications can be implemented using either a Mealy or a Moore machine with little difference in functionality. Moore machines play a more widespread role in today’s computers because of their interconnected nature. When things get connected (either two computers or two parts within the same computer), synchronicity because very important. Because Mealy machines do not wait for the clock signal to change states, synchronicity is very difficult to achieve. Combined with the fact that Moore machines just seemed easier to implement to me personally (because output is easier to compute), this makes me prefer a Moore implementation over a Mealy implementation.

# Conclusion

In this lab I learned a lot about Moore machines. I learned to build diagrams, transition tables, Kmaps, and schematics for Moore machine implementations. Everything seemed pretty straight-forward in this lab. This is probably because I had done (or attempted to do) something very similar in the previous lab. Because of their importance in the modern digital landscape, Moore machines are not likely to be going anywhere any time soon.