

Lab 11 - State Machines

In this lab, you've learned about One Hot and Binary state machines and how to build them.

Rubric

Item	Description	Value
Summary Answers	Your writings about what you learned in this lab.	25%
Question 1	Your answers to the question	25%
Question 2	Your answers to the question	25%
Question 3	Your answers to the question	25%

Lab Summary

We learned how to build Finite State Machines using both one hot and binary encoding. We practiced creating the diagrams and deriving “next-state” logic and designing the outputs. We also saw the differences in complexity between one hot (many flip flops but more simple logic) and binary (fewer flip flops but more complex). It helped to better understand how they work inside digital circuits.

Lab Questions

1 - Compare and contrast One Hot and Binary encodings

One Hot encoding uses one flip flop per state. Its easier to design the logic because transitions are more simple. You just check if the active flip flop is set. Binary uses a smaller number of flip flops and each flip flop represents a state in binary. It saves flip flops but makes the next state logic more complicated because there are multiple bits that have to be checked.

2 - Which method did your team find easier, and why?

We found one hot encoding to be easier to understand because writing the logic and outputs were just more simple. We only had to check one flip at a time instead of managing multiple bits. Binary was definitely harder to wrap our head around.

3 - In what conditions would you have to use one over the other?
Think about resource utilization on the FPGA.

One hot encoding is better when there are small number of state because the FPGA has more flip flops and a simple logic. Binary is better when there are lots of states because it uses less flip flops. Even though the logic becomes more complex it can save resources.

Code Submission

Upload a .zip of all your code or a public repository on GitHub.