

# 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

Summarize your learnings from the lab here.

In this lab we learned how to create two different types of state machines, and we explored their actions on the board. We learned how to implement the state machines in one hot and binary format, and we also learned how to derive the logic from a binary state table by using a k-map.

## Lab Questions

### 1 - Compare and contrast One Hot and Binary encodings

The One Hot encoding makes current state logic decisions much easier, as we don't need to look at the value of multiple D-FFs. However, it requires many more D-FFs as the complexity of state machines grows. The downsides of binary encoding are that there are often invalid representations. However, it is more efficient to represent complex state machines with binary encoding.

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

Our team found the one hot method easier. This is because this method did not require making a k-map but rather the logic could be derived from just looking at the table and connecting the values. Once we were able to do that, we were successfully able to assign them.

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

The complexity and number of states of the state machine, and the on-chip resources with how effective they are at handling high state numbers. I would also consider how much state storage the machine would use in order to determine which one should be used over the other.

## Code Submission

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