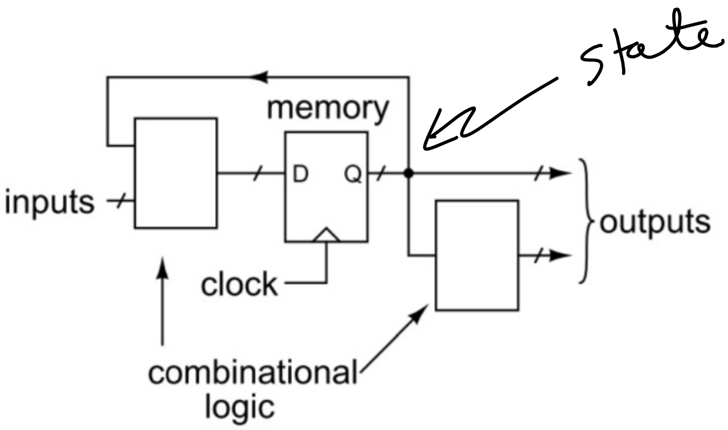
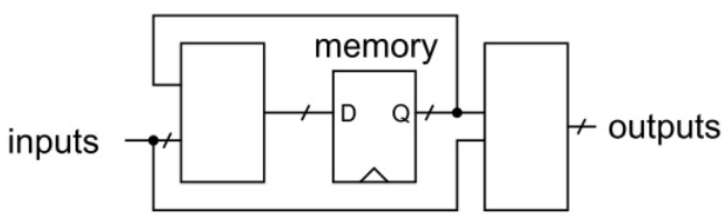
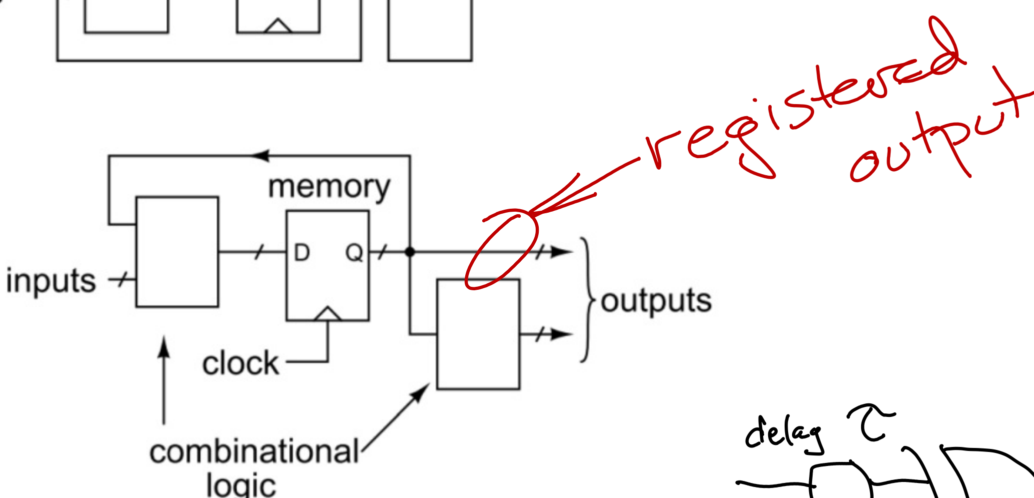
State Machines

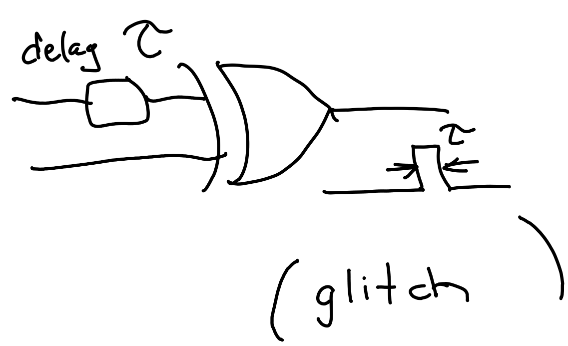
Exercise 1: Which signals in the above diagrams indicate the current state?



Exercise 2: Which outputs are registered? Which outputs could change whenever the input changes?







Exercise 3: Why?

> One approach is to begin by listing all the required *combinations* of the outputs. For a Moore state machine that has only registered outputs each of these will correspond to a state.

for Moore SM output is only function of state thus there must be at least one state corresponding to each output.

Exercise 4: If we used 8-bits of state information, how many states could be represented? What if we used 8 bits of state but used a "one-hot" encoding?

$2^8 = 256$ states

8 states w/ one-hot encoding

Exercise 5: The link below describes a game. List the top-level game states. Decompose each of these into multiple states. Repeat.

reset

intro — {

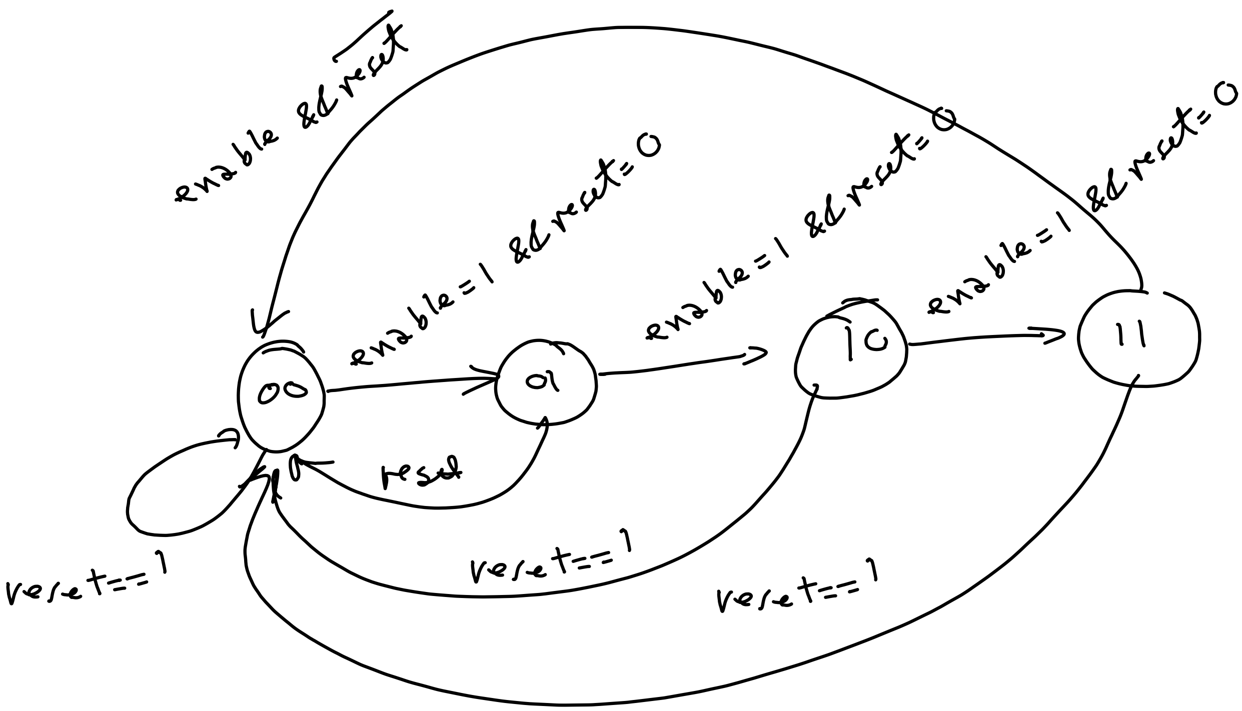
play

record — {

Exercise 6: What happens if both reset and enable are asserted?

reset will set count = 2'b00

Exercise 7: Draw the state transition diagram.



Exercise 8: Rewrite the state transition table and the module using *n* and *n+1*.

	reset		
n	0	1	n+1

Exercise 9: Write the state transition table for each state machine.