Exercises with Finite State Machines

CS 64: Computer Organization and Design Logic
Lecture #17

Ziad Matni
Dept. of Computer Science, UCSB

FINAL IS COMING!

- **Wednesday, 6/13** in this classroom
- **Starts at 12:00 PM **SHARP****
  - Please start arriving 10-15 minutes before class
- **I may ask you to change seats**
- Please bring your UCSB IDs with you

- **Closed book**: no calculators, no phones, no computers
- ***Only*** the MIPS Reference Card is allowed
- **You will write your answers on the exam sheet itself.**

What's On the Final Exam?

- Everything

What's On the Final Exam?

- Binary Arithmetic
- MIPS Assembly Language
  - Programming
  - Instructions
  - MIPS Calling Convention
- Digital Logic Design
  - Simplification using Logic Arithmetic, TTs, & Kmaps
  - Combinatorial design, incl. muxes, ALUs
  - Sequential design, incl. latches, DFF
  - Finite State Machine design

To Review

- ALL the class lectures and demos
- Your notes
- MIPS Reference Card
- ALL the class labs
  - If you get any Ethics/Impact of CS questions, they'd be short/M.C. questions…

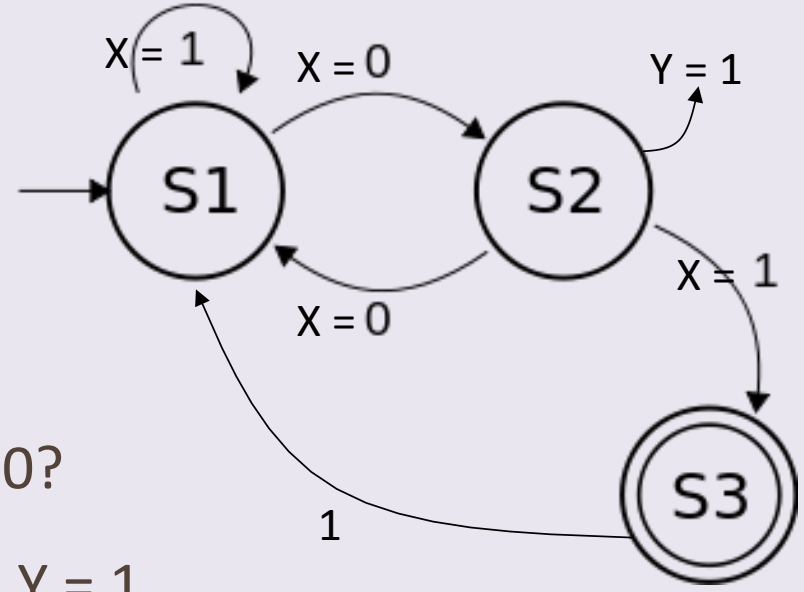
Lecture Outline

- FSM Exercises

Any Questions From Last Lecture?

FSM Exercise 1

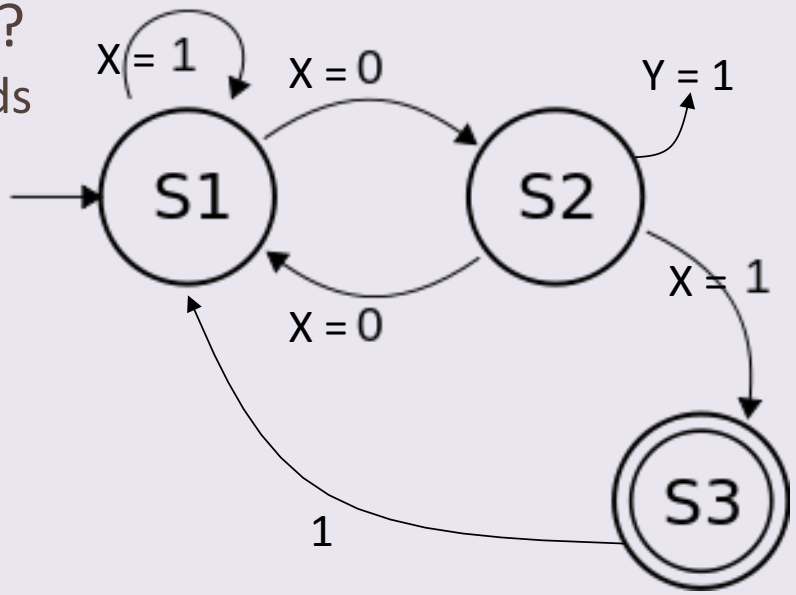
- Given a FSM described with the following state diagram where:
  - The initial state is S1
  - There is only 1 input, X
  - There is only 1 output, Y, and it is initialized to 0



- What state do you end up in if X takes on the sequential values 0110?
- Which of these inputs will result in Y = 1 at the end of their sequences?
  - **A. 0110**
  - **B. 1111101**
  - **C. 0101010**

FSM Exercise 1b

- How many bits do we need to represent all the states in this FSM?
  - Using regular, non-alternative methods
- Write the T.T. for this FSM
- Write the next-state functions for this FSM
- Design the digital logic circuit to implement this FSM, showing all inputs and all outputs



FSM Exercise 2

- Design a FSM that takes in 2 single-bit binary inputs, A and B and always is reset to (begins in) an initial state.

- The machine will move from the initial state only if **A&&B** is true. Once it does that, however, it will go through N states sequentially, once for every time **A||B** is true.

- On the last state, it simply goes to the initial state again and repeats.

A. Draw the state diagram for N = 3.
B. Using the "one-hot method", how many bits do we need to represent all the states in this FSM?
C. Write the next-state functions for this FSM using the approach in B.
D. Design the dig. logic circuit to implement the FSM using your results so far

FSM Exercise 3

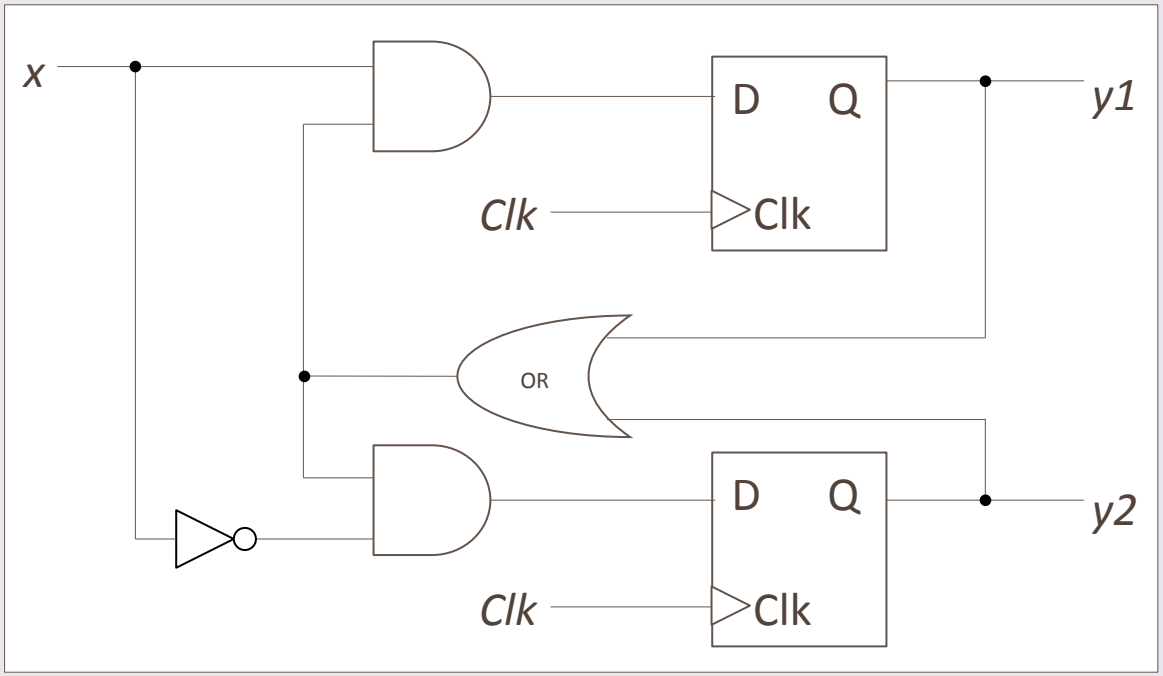
- Consider the FSM circuit, shown here using the "one-hot method"

A. Identify the non-clock inputs and outputs

B. Write the next state equations

C. Write the T.T. for this FSM

D. Draw the state diagram for this FSM



Your To Dos

- Lab #9 due Friday
- Bring your questions re: Final on Thursday

</LECTURE>