# COMP 122/L Lecture 27

Kyle Dewey

#### Outline

• Finite state machines

#### Finite State Machines

#### Finite State Machines

Basic idea: computation is done via traversal of **states**, where the states are known ahead of time.

#### Finite State Machines

# Basic idea: computation is done via traversal of **states**, where the states are known ahead of time.



## Example: Lock and Key



# Example: Counting Change



# Significance

- Can encode many problems using finite state machines (FSMs)
- FSMs can be implemented with sequential circuits
- Internals of processors can be encoded with FSMs

# Significance

- Can encode many problems using finite state machines (FSMs)
- FSMs can be implemented with sequential circuits
- Internals of processors can be encoded with FSMs



Step I: Encode each state in binary

Step I: Encode each state in binary



Step I: Encode each state in binary



Step 2: Make truth table mapping current state to next state

Step 2: Make truth table mapping current state to next state



Step 2: Make truth table mapping current state to next state





				S1		For N1	
S1	S0	N1	NO	S0	0	1	
0	0	0	1	0	0	0	
0	1	1	0	1	1	Х	
1	0	0	0				
1	1	Х	Х				



Step 4: Build sequential circuit

Step 4: Build sequential circuit



N0 = !S1!S0N1 = S0

Step 4: Build sequential circuit



# FSMs with External Inputs

Same process, but with more inputs in the truth table

# FSMs with External Inputs

Same process, but with more inputs in the truth table



# FSMs with External Inputs

Same process, but with more inputs in the truth table



KI	KR	ΚT	S1	S0	N1	NO
0	0	0	0	0	0	0
••	• •	• •	• •	• •	• •	• •
1	0	0	0	0	0	1
••	• •	• •	• •	• •	• •	• •

#### FSMs with Outputs

Additional outputs in truth table. Output on the corresponding state.

# FSMs with Outputs

Additional outputs in truth table. Output on the corresponding state.



# FSMs with Outputs

Additional outputs in truth table. Output on the corresponding state.



