Lecture 3: State Machine Review

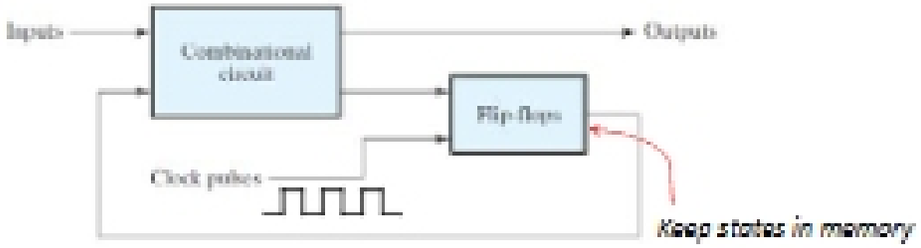
- Sequential Logic
- Control and data interaction
- ASM Chart
- Examples
- Mealy vs Moore

Announcement

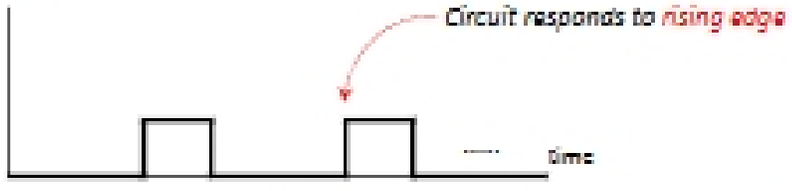
- Project 0 posted on sakai, flowchart and question answers due next Monday
- Please send your group information to lubo@udel.edu

Sequential Logic

Synchronous circuit = clocked sequential circuit



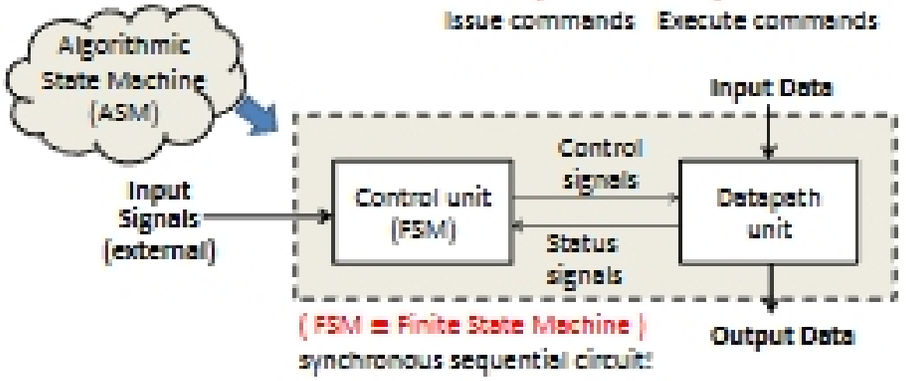
EX: CLK



Control and data interaction

➢ Digital systems use binary info for either control or data processing

Issue commands — Execute commands



(FSM = Finite State Machine)

synchronous sequential circuit:

① Control logic { Sends commands, Receives status } "Master"

② Datapath logic { Executes commands, Reports status } "Slave"

How to Design Your System?

Basic idea:

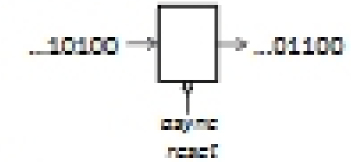
Sequential logic synthesis covered in 202:

- What do you want to circuit to do?
- ↓
- Determine the inputs, outputs, and states that describe circuit behavior
- ↓
- Reduce states, assign states (involve the state diagram)
- ↓
- Determine flip-flop type, derive the flip-flop equations (involve state table, excitation, K-maps…)
- ↓
- Draw the circuit

Main concerns of 222 projects (first two steps)

Serial 2's complementer

Example Serial 2's complementer (Mealy version)

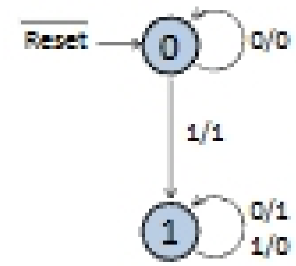


2's complement is formed by :

(1) Until first 1 occurs, output = input
(2) When first 1 occurs, output = input'

} Two STATES

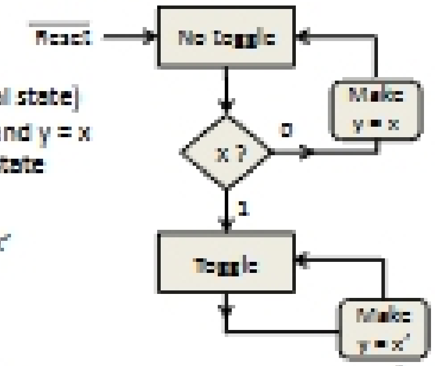
State Diagram



Algorithm

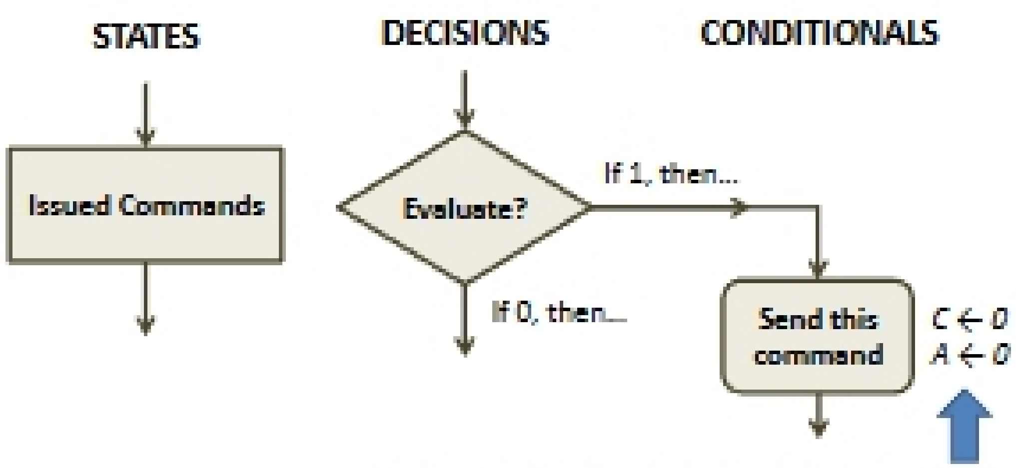
- No toggle state (initial state)
  If x = 0, stay in state and y = x
  If x = 1, go to toggle state
- Toggle state
  Stay in state and y = x'

Flow chart



ASM Chart

- ASM chart is basically a flow chart with details of :

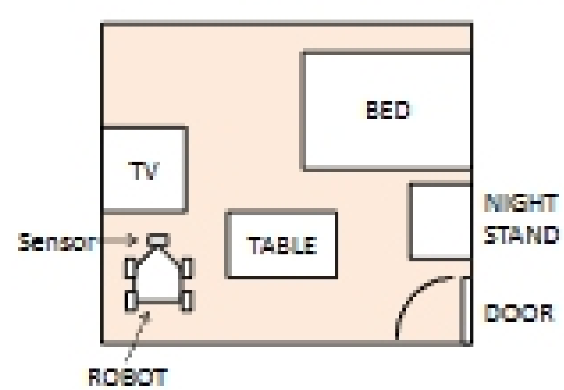


The datapath actions are written on the side

Robot Controller

Another example Robot controller

Design problem: Design a finite state controller for a robot so that it overcomes obstacles.



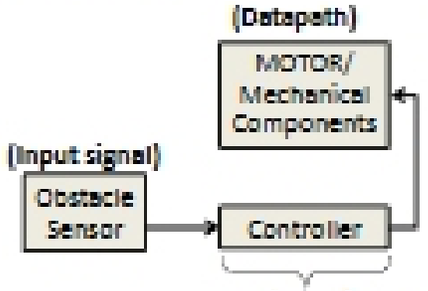
Requirements:

i) Sensor output = 1 if there is an obstacle
ii) Control lines: L = 1 (Left), R = 1 (Right)
iii) When an obstacle is detected, the robot should turn right until no obstacle is detected.
The next time an obstacle is detected, turn left until no obstacle is detected.
And so on….

Robot Controller

➢ **STEP 1:**

Good place to start is the block diagram for the robot operation



Contains variables that
- ➢ keep the states and
- ➢ use sensor info to switch between states

From requirements, the robot requires 4 states:

- (A) Straight [previous turn = left]
- (B) Turn right
- (C) Straight [previous turn = right]
- (D) Turn left

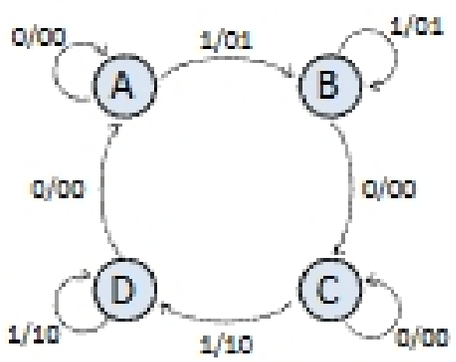
- Input is S = 0 or 1 ← Obstacle detected
- Outputs are L = 0 or 1 ← Turn left

  R = 0 or 1 ← Turn right

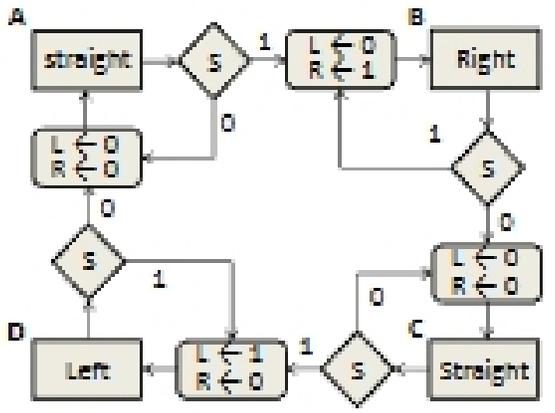
Robot Controller

➢ **STEP 2:**

Now that we have our states, construct the state diagram



Algorithmic State Machine

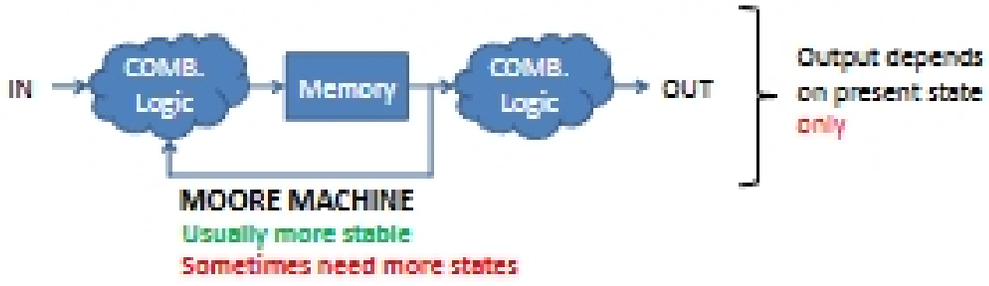


Is this Mealy or Moore?

Notation: L ← 0 means "0" appears at the output of "L"

Mealy vs. Moore

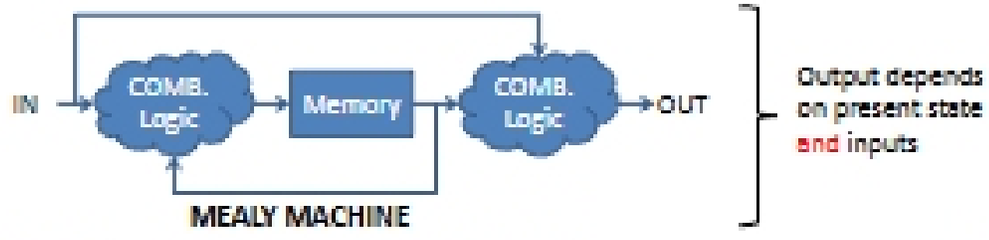
Mealy & Moore: 2 different types of sequential systems



Output depends on present state only

Usually more stable

Sometimes need more states



Output depends on present state and inputs