Lecture 20

- Logistics
  - Graded HW back today
  - HW7 Due next Wednesday
  - Lab 9 starts next week (2 week lab)
- Last lecture
  - Intro to Moore and Mealy machines
- Today
  - More Moore and Mealy machines
  - A bigger FSM example: Hungry Robot Ant in Maze

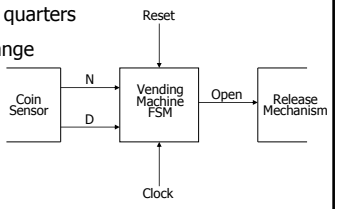
The "WHY" slide

- Bigger FSM example
  - Many things in the world are finite state machines (are we humans?) and most of them are far bigger and more complex than the examples we had so far. It is good to be able to understand and solve for a project-level FSM problem that's a lot more realistic to FSMs that you may design in industry or in grad school.

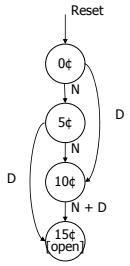
Example: A vending machine

- 15 cents for a cup of coffee
- Doesn't take pennies or quarters
- Doesn't provide any change

Previously we had mix of Moore and Mealy



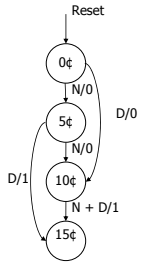
A vending machine: Moore machine



present state	inputs D	N	next state	present output
0¢	0	0	0¢	0
	0	1	5¢	0
	1	0	10¢	0
	1	1	–	–
5¢	0	0	5¢	0
	0	1	10¢	0
	1	0	15¢	0
	1	1	–	–
10¢	0	0	10¢	0
	0	1	15¢	0
	1	0	15¢	0
	1	1	–	–
15¢	–	–	15¢	1

symbolic state table

A vending machine: Mealy machine



present state	inputs D	N	next state	present output
0¢	0	0	0¢	0
	0	1	5¢	0
	1	0	10¢	0
	1	1	–	–
5¢	0	0	5¢	0
	0	1	10¢	0
	1	0	15¢	1
	1	1	–	–
10¢	0	0	10¢	0
	0	1	15¢	1
	1	0	15¢	1
	1	1	–	–
15¢	–	–	15¢	1

symbolic state table

A vending machine: State encoding

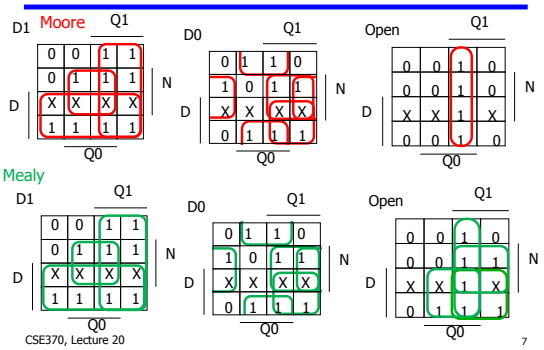
Moore

present state Q1	Q0	inputs D	N	next state D1	D0	present output
0	0	0	0	0	0	0
		0	1	0	1	0
		1	0	1	0	0
		1	1	–	–	–
0	1	0	0	0	1	0
		0	1	1	0	0
		1	0	1	1	0
		1	1	–	–	–
1	0	0	0	1	0	0
		0	1	1	1	0
		1	0	1	1	0
		1	1	–	–	–
1	1	–	–	1	1	1

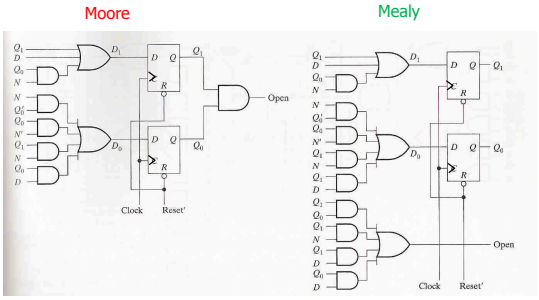
Mealy

present state Q1	Q0	inputs D	N	next state D1	D0	present output
0	0	0	0	0	0	0
		0	1	0	1	0
		1	0	1	0	0
		1	1	–	–	–
0	1	0	0	0	1	0
		0	1	1	0	0
		1	0	1	1	1
		1	1	–	–	–
1	0	0	0	1	0	0
		0	1	1	1	1
		1	0	1	1	1
		1	1	–	–	–
1	1	–	–	1	1	1

A vending machine: Logic minimization

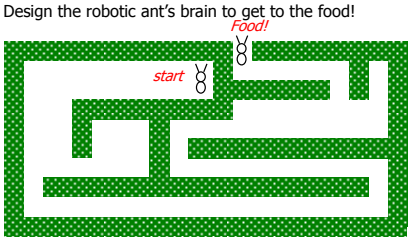


A vending machine: Implementation



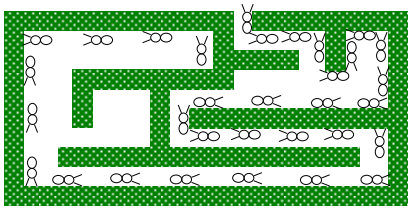
Robotic ant in a maze

- Robot ant, physical maze
  - Maze has no islands
  - Corridors are wider than ant
  - Design the robotic ant's brain to get to the food!



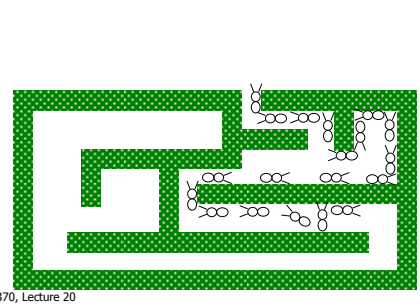
Robot ant specifics

- Sensors: L and R antennae, 1 if touching wall
- Actuators: F - forward step, TL/TR - turn left/right
- Goal: find way out of maze to get to food.
- Strategy: keep the wall on the right



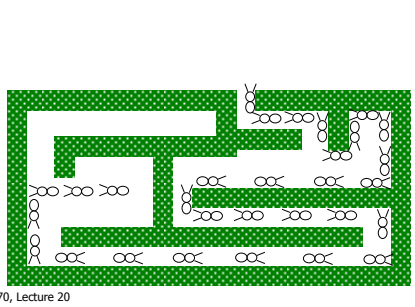
Example: ant brain (special case 1)

- Left (L) Antenna touching the wall



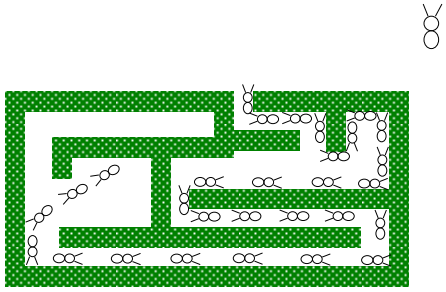
Example: ant brain (special case 2)

- Ant Lost

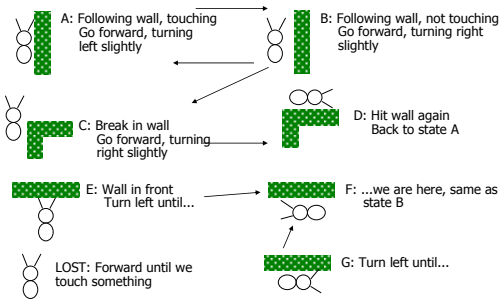


Example: ant brain (special case 2)

- Ant Lost (another example)



Robot Ant behavior



Notes & strategy

- Notes
  - Maze has no islands
  - Corridors are wider than ant
  - Don't worry about startup
  - Assume a Moore machine
  - Assume D flip-flops
- Strategy
  - Keep the wall on the right

Design the ant-brain FSM

1. State diagram
2. State-transition table
3. State minimization
4. State encoding
5. Next-state logic minimization
6. Implement the design

Notations

- Sensors on L and R antennae
  - Sensor = "1" if touching wall; "0" if not touching wall
    - L'R' ≡ no wall
    - L'R ≡ wall on right
    - LR' ≡ wall on left
    - LR ≡ wall in front
- Movement
  - F ≡ forward one step
  - TL ≡ turn left slightly
  - TR ≡ turn right slightly

1. State Diagram

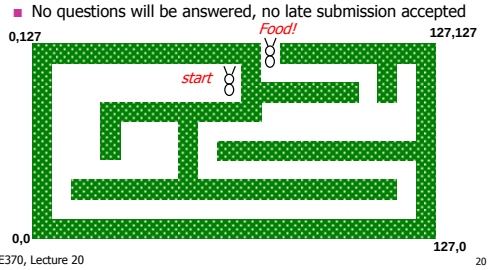
State Minimization, Encoding, and Circuit

- Next class

Extra credit (worth 10pts equivalent in a midterm)

Design the robotic ant's brain with virtual maze representation

- Due last day in class, Friday, March 13; printouts only
- Graded on clarity and completeness of explanation
- No questions will be answered, no late submission accepted



The maze

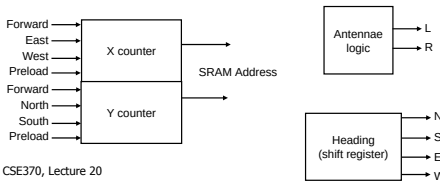
- Virtual maze
  - 128 × 128 grid
    - Stored in memory
    - 16384 8-bit words
  - *YX* is maze addresses
    - *X* is the ant's horizontal position (7 bits)
    - *Y* is the ant's vertical position (7 bits)
  - Each memory location says
    - 00000001 ≡ No wall
    - 00000010 ≡ North wall
    - 00000100 ≡ West wall
    - 00001000 ≡ South wall
    - 00010000 ≡ East wall
    - 00100000 ≡ Exit

Can have multiple walls
Example: 00001100
⇒ Walls on South and East

Design of different components

Predesigned:

Submit the designs for:



Recommendations

- Memory controller
  - Move horizontally: Increment or decrement *X*
  - Move vertically: Increment or decrement *Y*
- Shift register for heading
  - N: 0001
  - W: 0010
  - S: 0100
  - E: 1000
  - Rotate right when ant turns right
  - Rotate left when ant turns left
- Combinational logic for antennae logic