Lecture 19

◆ Logistics
  - HW7 due now
  - A few days off before HW8 kicks in
  - Midterm review session tomorrow 4:15 EEB125
  - Midterm 2 in class (45min long, starts at 10:35am)

◆ Last lecture
  - Moore and Mealy machines

◆ Today
  - A bigger example: Hungry Robot Ant in Maze

---

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!

![Robotic ant in a maze diagram]
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

A: Following wall, touching
   Go forward, turning left slightly

B: Following wall, not touching
   Go forward, turning right slightly

C: Break in wall
   Go forward, turning right slightly

D: Hit wall again
   Back to state A

E: Wall in front
   Turn left until...

F: ...we are here, same as state B

G: Turn left until...

LOST: Forward until we touch something

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

Robot Ant behavior

A: Following wall, touching
   Go forward, turning
   left slightly

B: Following wall, not touching
   Go forward, turning
   right slightly

C: Break in wall
   Go forward, turning
   right slightly

D: Hit wall again
   Back to state A

E: Wall in front
   Turn left until...

F: ...we are here, same as
   state B

G: Turn left until...

LOST: Forward until we
       touch something

CSE370, Lecture 19
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**
2. State Transition Table

- Using symbolic states and outputs

<table>
<thead>
<tr>
<th>state</th>
<th>L</th>
<th>R</th>
<th>next state</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOST</td>
<td>0</td>
<td>0</td>
<td>LOST</td>
<td>F</td>
</tr>
<tr>
<td>LOST</td>
<td>-1</td>
<td>1</td>
<td>E/G</td>
<td>F</td>
</tr>
<tr>
<td>LOST</td>
<td>1</td>
<td>-1</td>
<td>E/G</td>
<td>F</td>
</tr>
<tr>
<td>E/G</td>
<td>0</td>
<td>0</td>
<td>B</td>
<td>TL</td>
</tr>
<tr>
<td>E/G</td>
<td>0</td>
<td>1</td>
<td>E/G</td>
<td>TL</td>
</tr>
<tr>
<td>E/G</td>
<td>1</td>
<td>0</td>
<td>E/G</td>
<td>TL</td>
</tr>
<tr>
<td>B</td>
<td>-1</td>
<td>0</td>
<td>C</td>
<td>TR, F</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>-1</td>
<td>A</td>
<td>TR, F</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>A</td>
<td>TL, F</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>E/G</td>
<td>TL, F</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>-1</td>
<td>C</td>
<td>TR, F</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>0</td>
<td>A</td>
<td>TR, F</td>
</tr>
</tbody>
</table>

3. State minimization

- Any equivalent states?
Sure! Now you can represent states with 2 bits

4. State encoding

<table>
<thead>
<tr>
<th>state L R</th>
<th>next state</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOST 0 0</td>
<td>LOST</td>
<td>F</td>
</tr>
<tr>
<td>LOST - 1</td>
<td>E/G</td>
<td>F</td>
</tr>
<tr>
<td>LOST 1 -</td>
<td>E/G</td>
<td>F</td>
</tr>
<tr>
<td>E/G 0 0</td>
<td>B/C</td>
<td>TL</td>
</tr>
<tr>
<td>E/G 0 1</td>
<td>E/G</td>
<td>TL</td>
</tr>
<tr>
<td>E/G 1 -</td>
<td>E/G</td>
<td>TL</td>
</tr>
<tr>
<td>A 0 0</td>
<td>B</td>
<td>TL, F</td>
</tr>
<tr>
<td>A - 1</td>
<td>A</td>
<td>TL, F</td>
</tr>
<tr>
<td>A 1 -</td>
<td>E/G</td>
<td>TL, F</td>
</tr>
<tr>
<td>B/C - 0</td>
<td>B/C</td>
<td>TR, F</td>
</tr>
<tr>
<td>B/C - 1</td>
<td>A</td>
<td>TR, F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>state L R</th>
<th>next state</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 1 0 0</td>
</tr>
<tr>
<td>0 0 - 1</td>
<td>0 1 1 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 1 -</td>
<td>0 1 0 0 1</td>
<td></td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>1 1 0 0 1</td>
<td></td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>1 1 0 1 0</td>
<td></td>
</tr>
<tr>
<td>0 1 1 -</td>
<td>0 1 0 1 0</td>
<td></td>
</tr>
<tr>
<td>1 0 0</td>
<td>1 1 0 1 0</td>
<td></td>
</tr>
<tr>
<td>1 0 1 -</td>
<td>1 1 0 1 0</td>
<td></td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>1 1 1 1 0</td>
<td></td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>0 1 0 1 0</td>
<td></td>
</tr>
<tr>
<td>1 1 1 -</td>
<td>0 1 0 1 0</td>
<td></td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>1 1 1 1 0</td>
<td></td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>1 1 1 1 0</td>
<td></td>
</tr>
</tbody>
</table>

CSE370, Lecture 19
5. Next state logic minimization

<table>
<thead>
<tr>
<th>state inputs</th>
<th>next state outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>X, Y, L, R</td>
<td>X', Y', F, TR, TL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inputs</th>
<th>F TR TL</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>00</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>01</td>
<td>1 1 0 1</td>
</tr>
<tr>
<td>01</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>10</td>
<td>1 1 1 0</td>
</tr>
<tr>
<td>10</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>11</td>
<td>1 1 1 0</td>
</tr>
<tr>
<td>11</td>
<td>1 0 1 0</td>
</tr>
</tbody>
</table>

6. Circuit Implementation

- Outputs are a function of the current state only - Moore machine
Extra credit  
(worth 15pts equivalent in a midterm)

Design the robotic ant’s brain with virtual maze representation
- Due last day in class, Friday, June 6; printouts only
- Graded on clarity and completeness of explanation
- No questions will be answered

![Virtual maze diagram]

The maze

- Virtual maze
  - 128 x 128 grid
  - Stored in memory
  - 16384 8-bit words
  - XY 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
    - 00010000 = South wall
    - 00100000 = East wall
    - 00100000 = Exit

Can have multiple walls
Example: 00001100
⇒ Walls on South and East
Design of different components

Predesigned:

- **Ant-Brain FSM**
  - Forward
  - Turn right
  - Turn left

- **SRAM**
  - Maze
  - Data

Submit the designs for:

- **X counter**
  - Forward
  - East
  - West
  - Preload
  - SRAM Address

- **Y counter**
  - North
  - South
  - Preload

- **Antennae logic**
  - L
  - R

- **Heading (shift register)**
  - North
  - South
  - East
  - West

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**