CSC258: Computer Organization

Combinational Logic

Anonymous: Quizzes and Fairness

"… A lot of students in earlier sections share the quiz question with students who have the tutorial later in the evening …"

- I've had questions and concerns raised in office hours, too, about the overall fairness of marking.
- Some TAs are more lenient than others, some quizzes are easier than others, …

Anonymous: Quizzes and Fairness

- Because of the variables involved (time, TA, quiz difficulty), fairness across sections is difficult.

- You can help: *please don't explicitly undermine the process.*
  - The quizzes come from the recommended exercises. Don't provide additional guidance.
- I will be monitoring performance across sections and making adjustments as necessary.

Anonymous: Lab Length

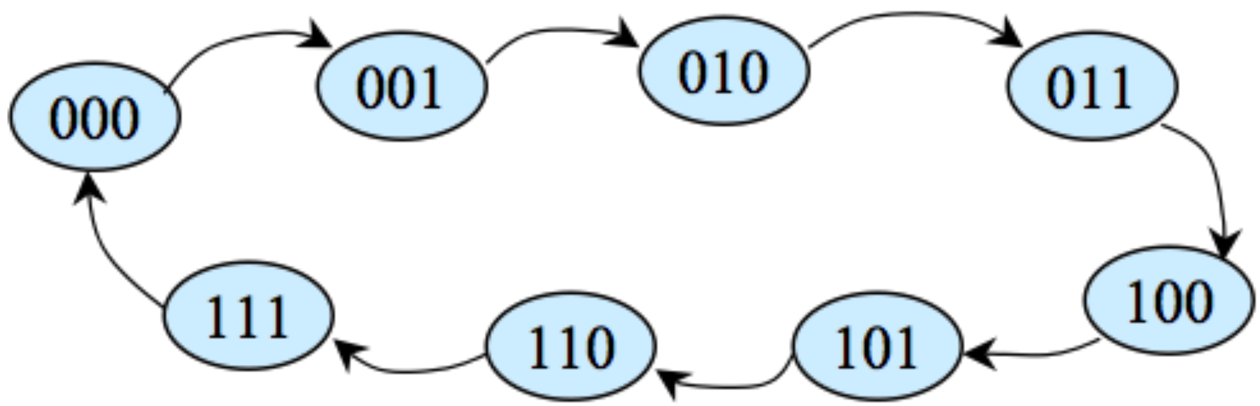
“Labs are a bit too long”

- Labs will get longer for the next couple of weeks, and then they’ll decline in length at the end of the term.
- For the next couple of weeks, you may wish to do some work before your lab.
  - I will try to get labs posted by Saturday each week.

Finite State Machines

What is “State”?

- State is the current “value” of the system
- For example, a counter has “state”



Representing State

- One choice: State tables
- They are truth tables!
  - One input is the current state
  - Another is the new state

State(t)	X	State (t+1)	Q(t)
0	0	0	0
0	1	1	0
1	0	0	0
1	1	1	1

Representing State

- State tables can get very complex
  - Systems may have many, many possible states
- Here's the state table for a 3-bit counter

State(t)	State(t+1)
000	001
001	010
010	011
011	100
100	101
101	110
110	111
111	000

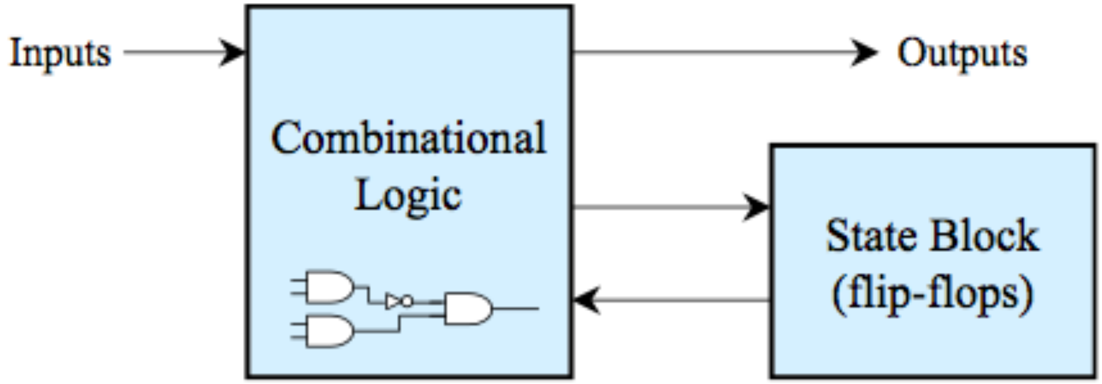
Finite State Machines

- Instead of using tables, we can use graphs
  - Humans are much better at processing data visually, in graph form
  - We know a -huge- amount about graphs
- Our graphs are called “Finite State Machines”
  - Finite: Limited number of states
  - Machine: Theoretical model of a computer
  - Heavily used in theoretical computer science

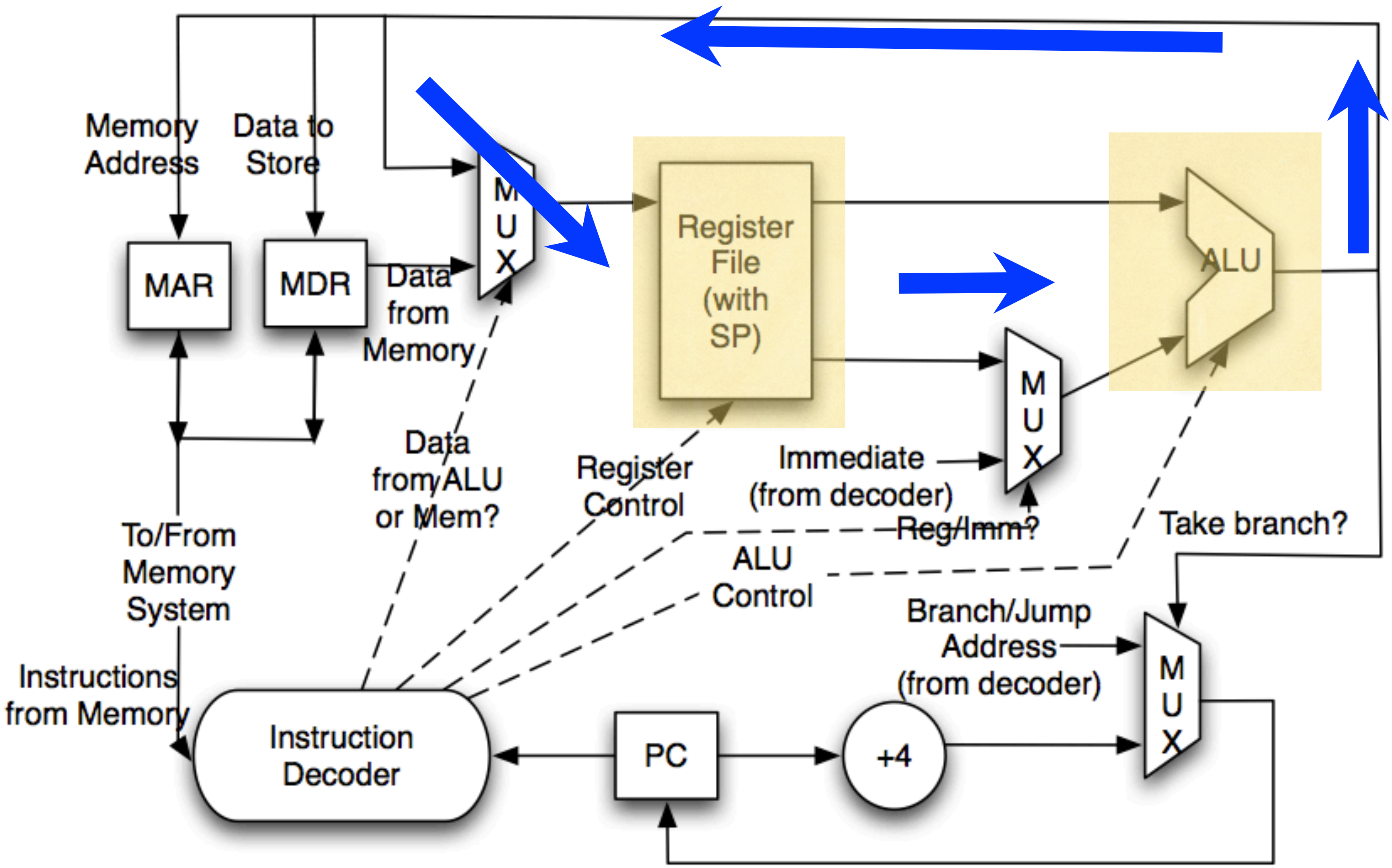
FSM Components

- A set of states and an initial state
- A set of legal inputs and a mapping of transitions between states
  - "Given input x in state A, transition to B"
- A set of actions (outputs) associated with each transition (Mealy machine) or state (Moore machine)

Designing FSMs



- Circuits with state will be designed in two blocks
    - State block: stores the current machine state
    - Combinational block: given inputs and the current state, computes the new state and outputs
- Processors are FSMs:
  an ALU (combinational) + registers (state)


Memory Address
Data to Store
MAR
MDR
Data from Memory
M U X
Register File (with SP)
ALU
M U X
Data from ALU or Mem?
Register Control
Immediate (from decoder)
Reg/Imm?
Take branch?
ALU Control
To/From Memory System
Branch/Jump Address (from decoder)
M U X
Instructions from Memory
Instruction Decoder
PC
+4

Implementing a FSM

1. Draw the state diagram (or table), numbering each state
2. Select the number of flip-flops required to store the state
   - How many bits are required to describe the state?
3. Build a truth table for each output, including the new state
   - Building multiple truth tables can help decrease complexity -- it's like building modules in software
   - Don't care conditions will help a huge amount
4. Implement the combinational logic from the truth tables (Alternately, design from high-level components.)

The GoldenEye Pen

Pen Grenade (GoldenEye)

Gadget: Pen Grenade
Movie: GoldenEye
Owner: James Bond
Status: Destroyed

An exploding pen that could be used to threaten the bank manager next time you "sign on the dotted line" but not in this case, the pen actually contributed heavily on the success of the mission.

Use

After fuel had been spilled thought Trevelyan's HQ in Cuba, Boris fiddled nervously with the pen twirling it around in his fingers and clicking it. The pen in detonated in four seconds after the pen is clicked three times, Bond kept a count on this. When it was about to explode he threw it from Boris' hands into the puddle of fuel and ducked for cover.

Specs

A ordinary look aluminum Parker pen, with an extraordinary feature, the pen was actually a Clas-4 grenade. The fuse was activated by three clicks of the pen, to detonate in 4 seconds, however it could be disarmed by clicking three times within that four seconds.

From: http://www.mi6-hq.com/sections/q-branch/pengrenade.php3?t=mi6&s=ge

Your Mission

... whether or not you choose to accept it:

- Design the control circuitry for the pen grenade.
- Your circuit should take, as input, a clock signal and a “click” signal.
  - The click signal is True iff the button on the pen has been depressed during the clock cycle.
  - The clock is fast enough that a human can only click the pen once in a cycle.
- Your circuit should output a signal “armed” that is true iff the grenade is in “countdown mode” (to explode after 4 seconds).
  - Don’t worry about the “BOOM!” state. That’s a different state machine.

Begin by drawing the state diagram that describes the desired behaviour.

A Variation

... whether or not you choose to accept it:

- Design the control circuitry for the pen grenade.
- Your circuit should take, as input, a clock signal and a “click” signal.
  - The click signal is True iff the button on the pen has been depressed during the clock cycle.
  - The clock is fast enough that a human can only click the pen once in a cycle.
  - **There are 100 clock signals per second.**
- **Your circuit should output a signal “boom” that is true iff the grenade should explode.**

What part of your circuit is affected by this modification?

Tips

- Make assumptions to reduce the truth table
  - Assume only one input will arrive in a cycle.
  - Ignore an output on the incoming arc (or set up your machine as a Moore machine).
- Fix your assumptions in the hardware!
  - Example: Add logic to change the state to the trap state if more than one input arrives at a time.
- Implement extra functionality, like *reset*, using appropriate hardware.