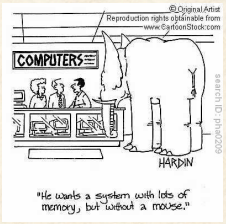
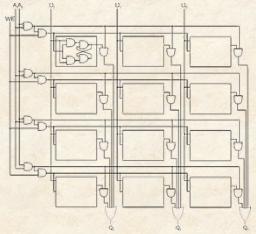
UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 202
Introduction to Computation

Professor Andrea Arpaci-Dusseau
Spring 2010

Lecture 12: How does a computer… remember?





CS 202 Schedule

10/1	F	… Remember? (Hardware perspective)
10/4	M	Artificial Intelligence
10/6	W	Vision
10/8	F	Social Robots
10/11	M	Visualization
10/13	W	Exam 1 Review
10/15	F	Exam 1
10/18	M	Regular Lecture
10/20	W	Regular Lecture
10/22	F	Project 1: Draft due for peer comments on website
10/25	M	Regular Lecture
10/27	W	Project 1 Due: Class demos

Exam 1

Administrative Details

- In-class, Friday 10/15
- Closed notes, closed reading, closed laptops
- Covers lectures and homeworks 1-5
  - Very similar to homework questions

Understanding Code and Algorithms

- You will not write any code yourself
  - How do given Scripts behave? How many times will loop execute? What will be the value of this variable? Code equivalent?
- Draw corresponding decision tree for given code

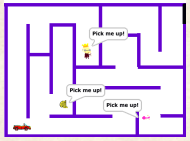
Understanding Logic and Modern Computers

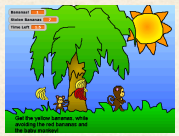
- Binary numbers, truth tables, sum of products, memory

Project 1: Variables!

Points-based Game – Open ended

- Must contain integer variable and boolean variables
- Must have multiple levels of difficulty

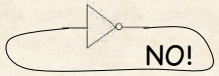




Review: Combinational Circuits

Combinational Circuit

- Always gives same output for given set of inputs
  - ex: adder always generates sum and carry, regardless of previous inputs
- Cycles are not allowed
  - Cannot have feedback from output back to input



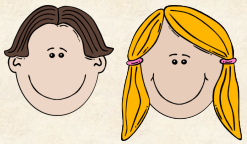
- Useful for many, but not all, aspects of computation
  - Arithmetic Logic Unit (ALU)

Today's Challenge

How can we remember information with just AND, OR, NOT?

Quick Review: Boolean Logic

How to express:
Matt will go to party if and only if Sue goes to party?



Two boolean variables: M and S

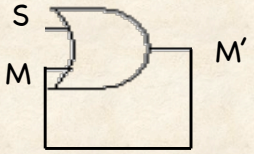
M = S

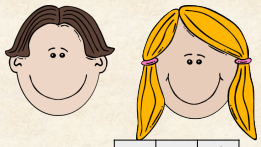
S ——— M

More Complicated Expression

Matt doesn't like changing his mind…

Represent with a circuit:
Matt will go to the party if Sue goes or if he already wanted to go





Scenarios

S	M	M'
0	0	0
0	1	1
1	0	1
1	1	1

Sequential Circuits

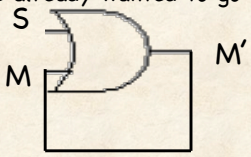
Sequential Circuit (vs. Combinational)

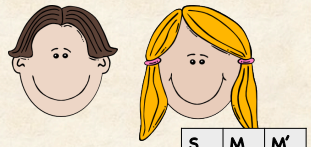
- Stores information: state
- Output depends on state + input
  - Given same input might produce different output, depending on stored information
- *Example:* ticket counter
  - Advances when push button, output depends on previous state
- Cycles are allowed
  - Can have feedback from output to input
- Useful for building memory!

More Complicated Expression

Matt doesn't like changing his mind…

Represent with a circuit:
Matt will go to the party if and only if Sue goes or he already wanted to go



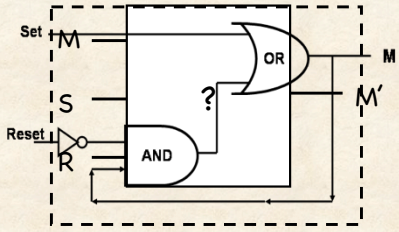


S	M	M'
0	0	0
0	1	1
1	0	1
1	1	1

Problem with this circuit (or Matt)?
Once going, can't change mind! Once M=1, always M=1

How can Matt change his Mind?

Matt will go to the party if and only if Sue goes OR (he already wanted to go AND Rita does not go)



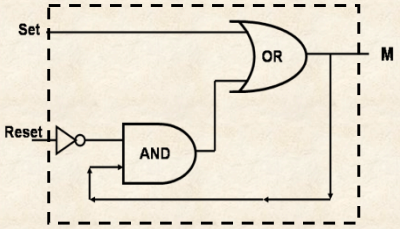
How would you express?

M' = S OR (M AND NOT R)

R, S: "**control**" inputs
What is S doing?
Setting state
What is R doing?
Resetting state (to 0)

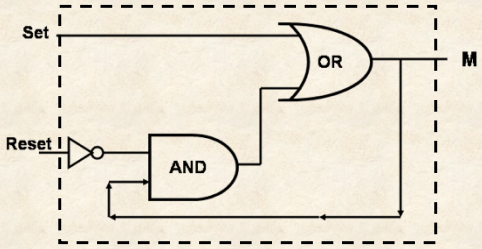
How can Matt change his Mind?

Matt will go to the party if Sue goes OR if the following holds: (he already wanted to go AND Rita does not go)



S	R	M	M'
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

R-S Flip-Flop (Caution: Simplified !!)



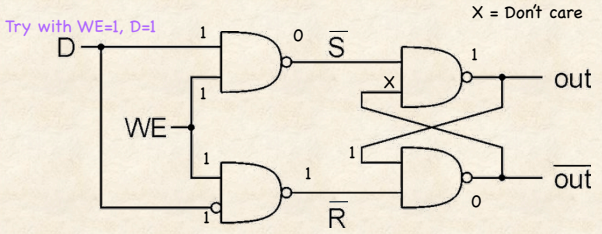
If Set = 1 (and Reset = 0), M = 1
If Reset = 1 (and Set = 0), M = 0
If Set = 0, Reset = 0, M keeps old value!

Not best if both Set and Reset = 1 (who wins?)

Convenient 1-Bit Memory

Two inputs: D (data) and WE (write enable)

- when WE = 1, latch is set to value of D
- when WE = 0, latch holds previous value (ignores D)

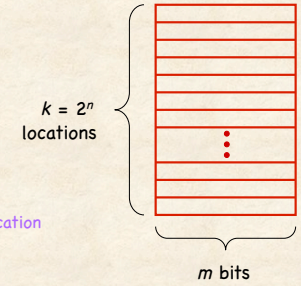


Random Access Memory (RAM)

Memory: Remembers lots of bits, not just 1 bit

- Logical $k \times m$ array of stored bits

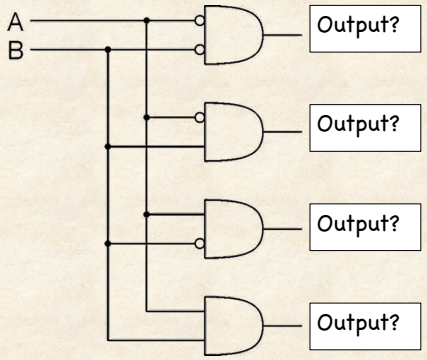
Address Space:
number of locations
(usually a power of 2)



Addressability:
number of bits per location
(e.g., 32 bits)

Mystery Circuit

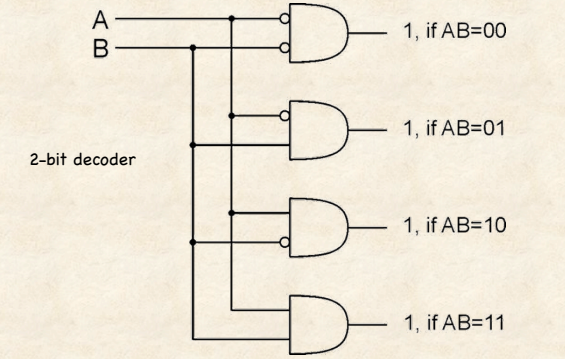
What does this combinational circuit do?



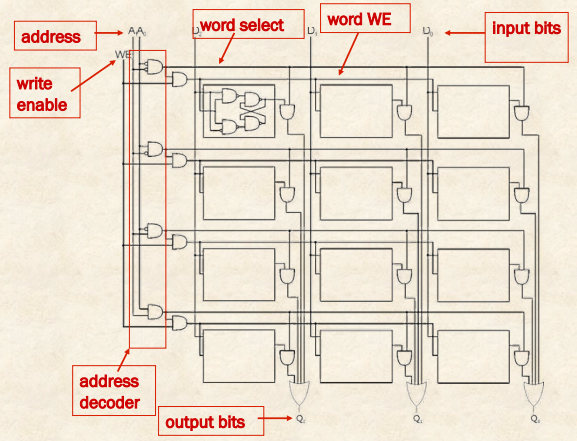
Decoder Circuit

n inputs, 2^n outputs

- exactly one output is 1 for each input pattern



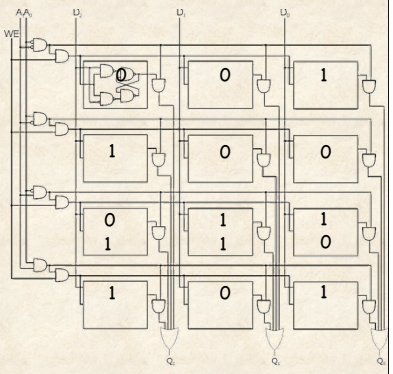
2^2 x 3 bit Random Access Memory



Example RAM Operation

What happens when:

WE	A1	A0	D2	D1	D0	Q2	Q1	Q0
0	0	0	1	1	1			
0	0	1	1	1	1			
1	1	0	1	1	0			
0	1	0	1	1	1			



Today's Summary

Sequential circuits (vs. combinational)

- Can remember values using feedback loops in circuits!
- Implement Random Access Memory (RAM)

Reading

- Chapter 5.1 – 5.2.1 (Pages 151-183)

Announcements

- Homework 3: Graded by noon today…
- Homework 4: Due today
- Homework 5: Available this afternoon
- Project 1: Start thinking about soon
- Exam 1: Two weeks from today…