University of Virginia 21 September 2011
cs1120: Introduction of Computing
Explorations in Language, Logic, and Machines

Class 13: Modeling Computers

Assignments Due

- **Friday, 23 September:** Problem Set 3
- **Monday, 3 October:** Problem Set 4
- **Wednesday, 12 October:** Exam 1 Due (will be take-home, handed out on Friday, 7 October)

Upcoming Help Schedule

Today: 5-6:30pm (Jiamin, Rice 1st)
Thursday: 9:45-11am (Dave, Rice 507); 10-11:30am (Peter, Rice 1st); 1-2:30pm (Joseph, Rice 1st); 4:30-7:30pm (Jonathan/Jiamin, Rice 1st)

Transistors

By the end of the class, the only thing that you think is "magic" about how applications like Google and Facebook work is the physics that make a **transistor** work. We assume John Bardeen, Walter Brattain, and William Shockley got this right (back in 1947) since they got a Nobel Prize for it, and that there are two devices that behave like this:

Normal Control:

output = (**and** *source control*)

Inverted Control: (with source wired to **on**)

output = (**not** *control*)

Each input and output variable is a single bit, represented by a voltage using digital abstraction (voltages below a set threshold represent **0**, and those above a set threshold represent **1**.)

Why is Silicon Valley located around Palo Alto, CA?

"Truth table": a table that defines a *function* by showing a mapping between inputs and outputs.

Input: *A*	Input: *B*	Output: (and *A B*)
0	0	0
0	1	0
1	0	0
1	1	1

Truth table for a three-bit adder:

A	B	C	R_1	R_0
0	**0**	**0**	**0**	**0**
0	**0**	**1**	**0**	**1**
0	**1**	**0**		
0	**1**	**1**		
1	**0**	**0**		
1	**0**	**1**		

R_1 = (**or** (**and3** (not A) B C) (**and3** A (not B) C) (**and3** A B (not C)) (**and3** A B C))

How can we implement **(and3 a b c)** using our transistors?

How can we implement **(or a b)** using our transistors?

What makes a good *model*?

To model a computer, what are the three things we need to model?