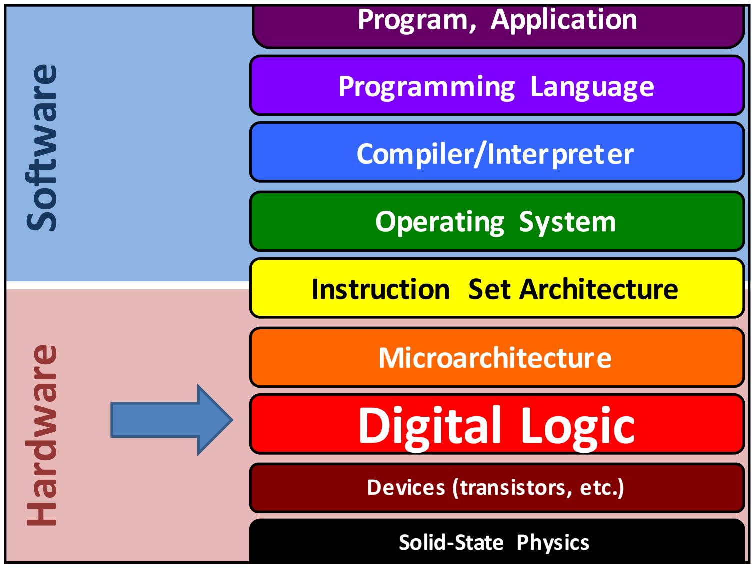
Digital Logic

*Gate*way to computer science



Software

Hardware

Program, Application

Programming Language

Compiler/Interpreter

Operating System

Instruction Set Architecture

Microarchitecture

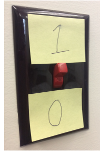
Digital Logic

Devices (transistors, etc.)

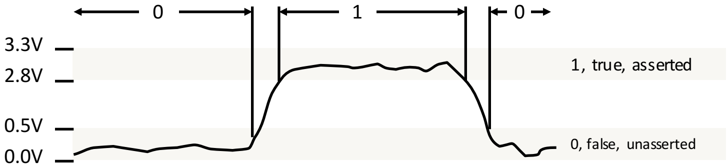
Solid-State Physics

Digital logic values

Abstraction!



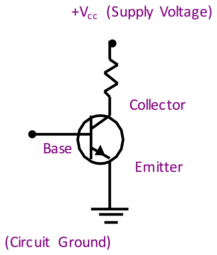
A digital circuit has only two logical values present.



*Exact voltage levels and their interpretation varies with technology.

Transistors

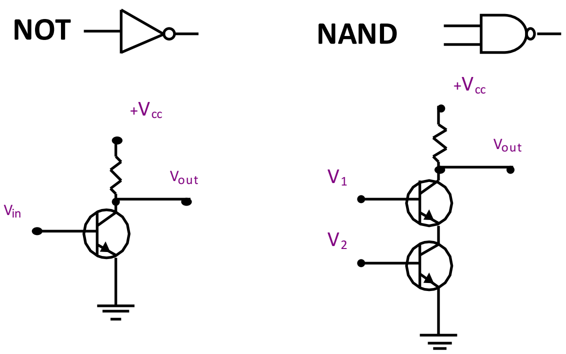
Very fast binary switches.



Logic Gates

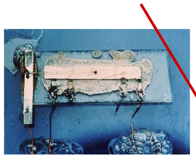
Abstraction!

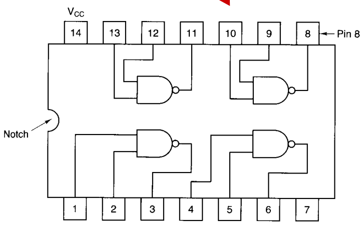
Tiny electronic devices that compute functions on two-valued signals.



Integrated Circuits (invented 1950s)

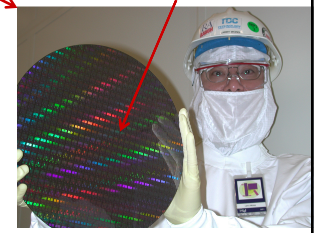
Gates are manufactured in units called integrated circuits.
From SSI (tens) to VLSI (hundreds of thousands to billions)



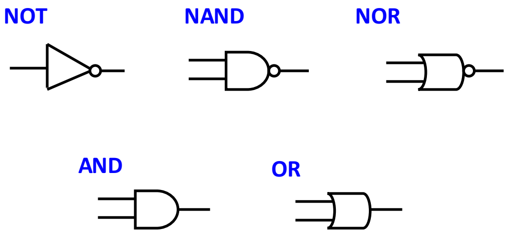


Chip

Wafer

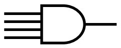


Five basic gates



Define with truth tables.

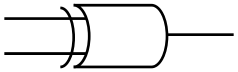
*Gates with more than two inputs are possible:



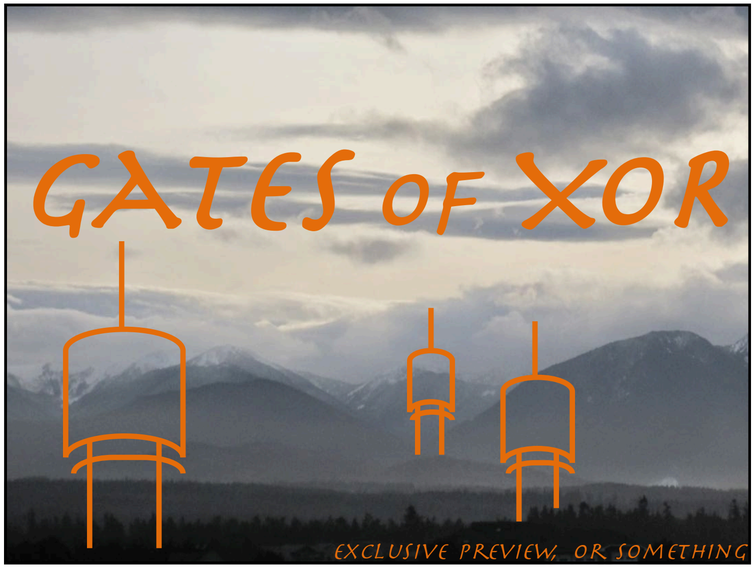
Design new gates from old

exclusive or often used as a one-bit comparator.

XOR

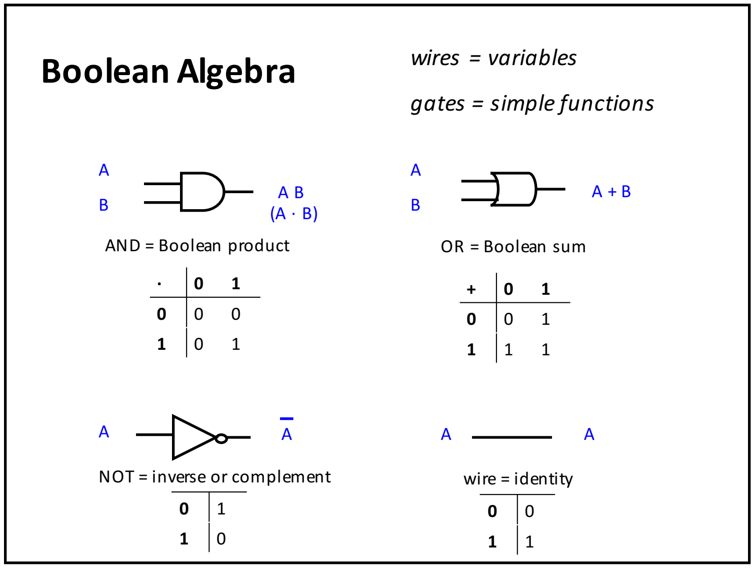


Future video game designers, Halloween costumers extraordinaire, sci-fi/fantasy screenwriters, I have an idea…



GATES OF XOR

EXCLUSIVE PREVIEW, OR SOMETHING



Boolean Algebra

wires = variables

gates = simple functions

AND = Boolean product

·	0	1
0	0	0
1	0	1

OR = Boolean sum

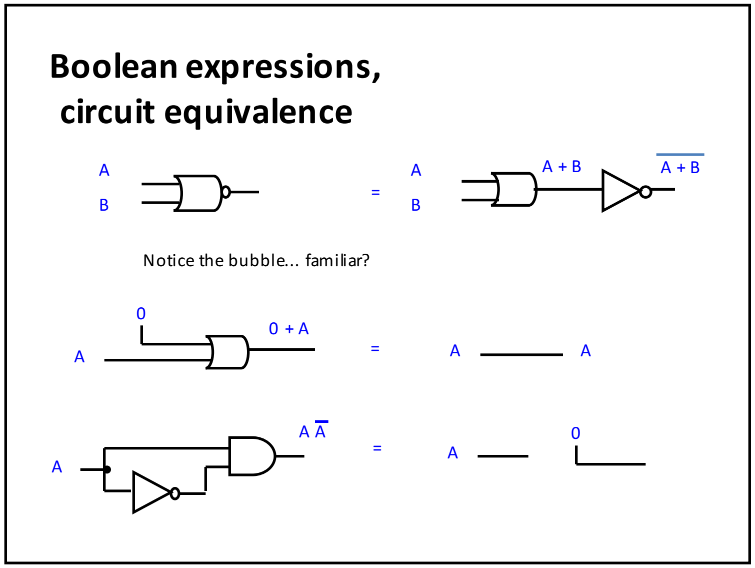
+	0	1
0	0	1
1	1	1

NOT = inverse or complement

0	1
1	0

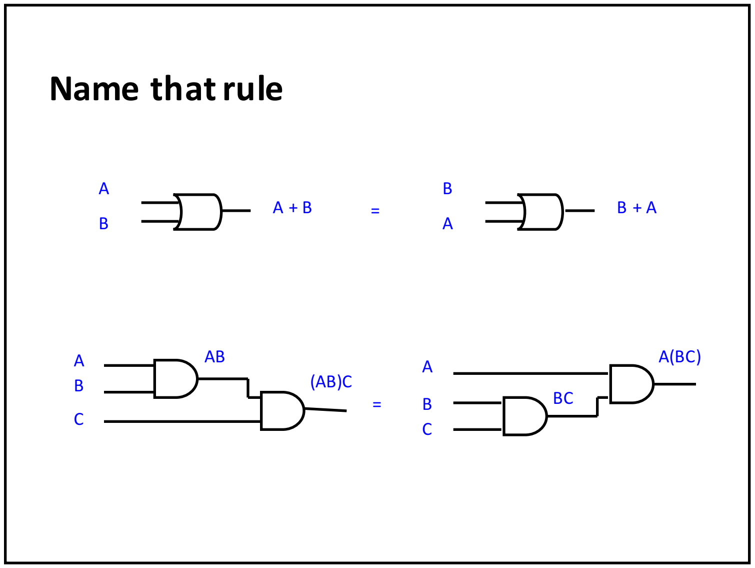
wire = identity

0	0
1	1



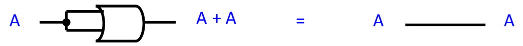
Boolean expressions, circuit equivalence

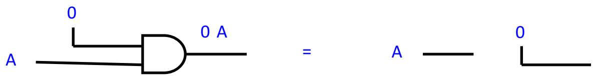
Notice the bubble... familiar?

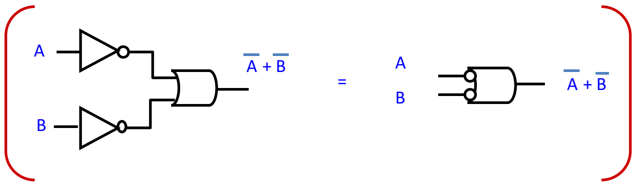


Name that rule

Rules, rules, rules

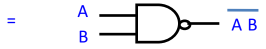


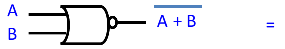




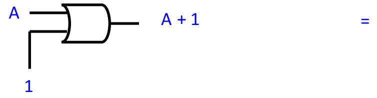
DeMorgan's Law

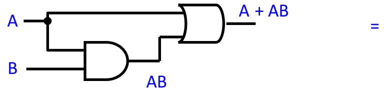
(**double bubble**, toil and trouble, in Randy's words…)





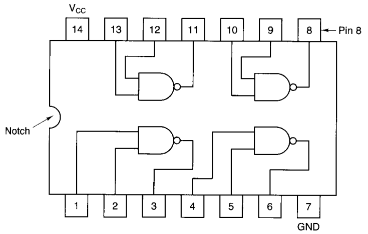
Take care





A universal gate*

Let's prove it! Build AND, OR, NOR, NOT using only NAND.

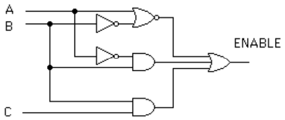


*Are there others?

Circuit simplification

*All other things equal, smaller circuits are cheaper, faster, cooler, and easier to design.

Can we find a simpler circuit that performs the same function?*



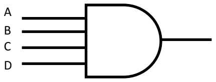
Start with an equivalent Boolean expression

F(A, B, C) =

Check with a truth table... with 3 inputs?

Code detectors

A four input AND gate recognizes exactly one input code.



Design a code detector that recognizes ABCD = 1001.

Design a code detector that recognizes either ABCD = 1001 or ABCD = 1111.

Voting machines

A **majority circuit** outputs 1 whenever a majority of its inputs equal 1.
Design a simple majority circuit for three inputs.

A	B	C	Majority
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Sum of products

A **sum of products** representation is

the logical sum (OR)
of products (AND)
of inputs or their complements (NOT).

Inputs			Outputs		
A	B	C	D	E	F
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

Try building for E: Think of summing/ORing code detectors.