Module 01: Fundamentals of Computing

Intro to Computer Science 1 - C++
Professor Scott Frees

What do computers do?

Computers don't actually *do* much…

- Add/Subtract/Multiply/Divide numbers
- Remember numbers

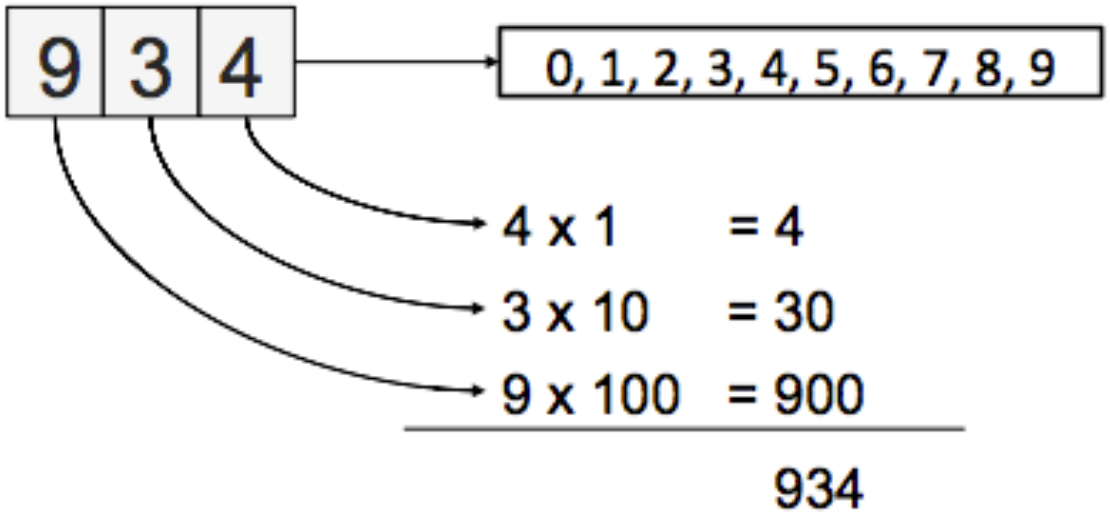
Everything else is just a combination of these….

billions of them… each second….

Numbers

We think of numbers with a base of 10

- Each digit can be one of 10 different values
- Why?



Computers don't have fingers...

Computers are built of electronic circuitry

At the lowest level, they consist of transistors, small circuits that can control if electricity flows through them or not: **on and off.**

Instead of 10 states, a computer works with 2 states per digit: 0 and 1.... otherwise known as binary numbers.

Binary Numbers

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10

4-Bits: 0 - 15
8-Bits: 0 - 255
16-Bits: 0 - 65,535
32-Bits: 0 - 4,294,967,295
64-Bits: 0 - $1.84467441 \times 10^{19}$

Less compact, but just as expressive!

Binary numbers - uses

A single "bit" is either **1** or **0**.

On or **Off**

True and **False**

Boolean algebra works with true/false

NOT	A	OUT
	0	1
	1	0

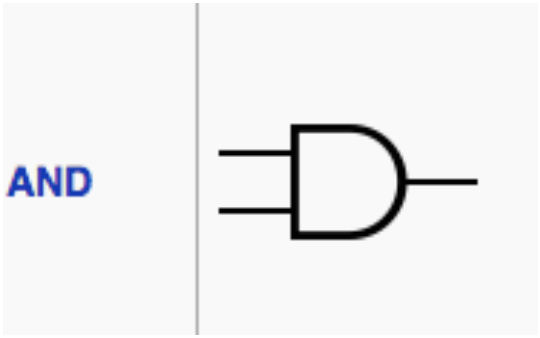
AND	A	B	OUT
	0	0	0
	0	1	0
	1	0	0
	1	1	1

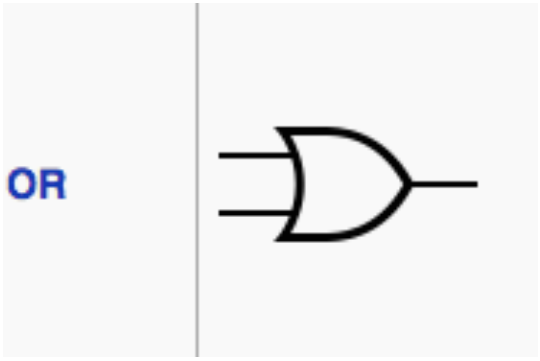
XOR	A	B	OUT
	0	0	0
	0	1	1
	1	0	1
	1	1	0

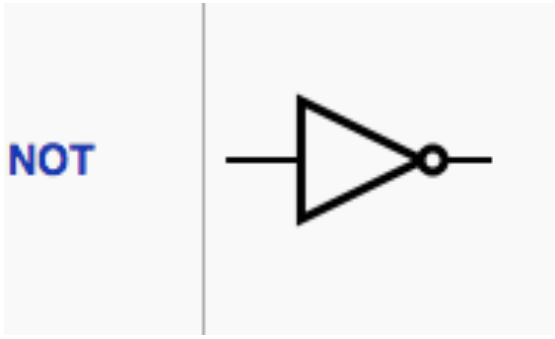
OR	A	B	OUT
	0	0	0
	0	1	1
	1	0	1
	1	1	1

Logic Gates

Boolean operations can be implemented *in hardware circuits (20-30 transistors)*





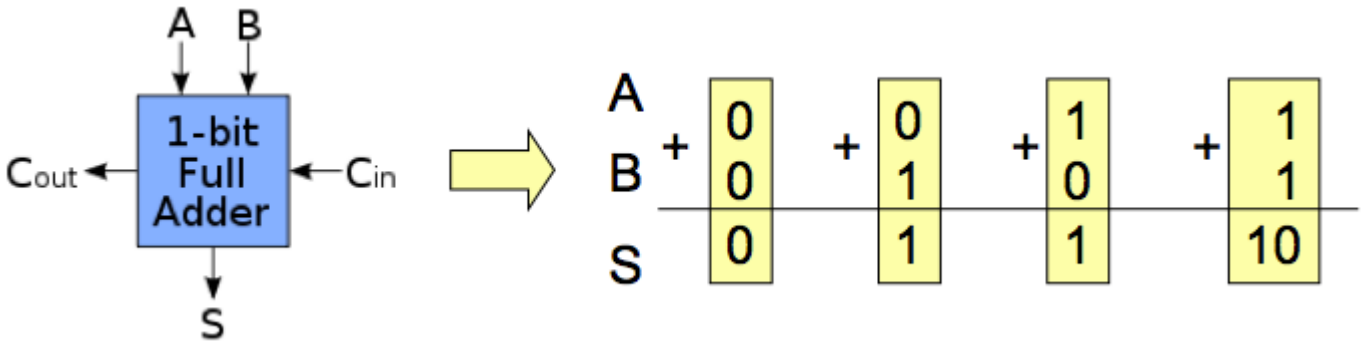


Arithmetic

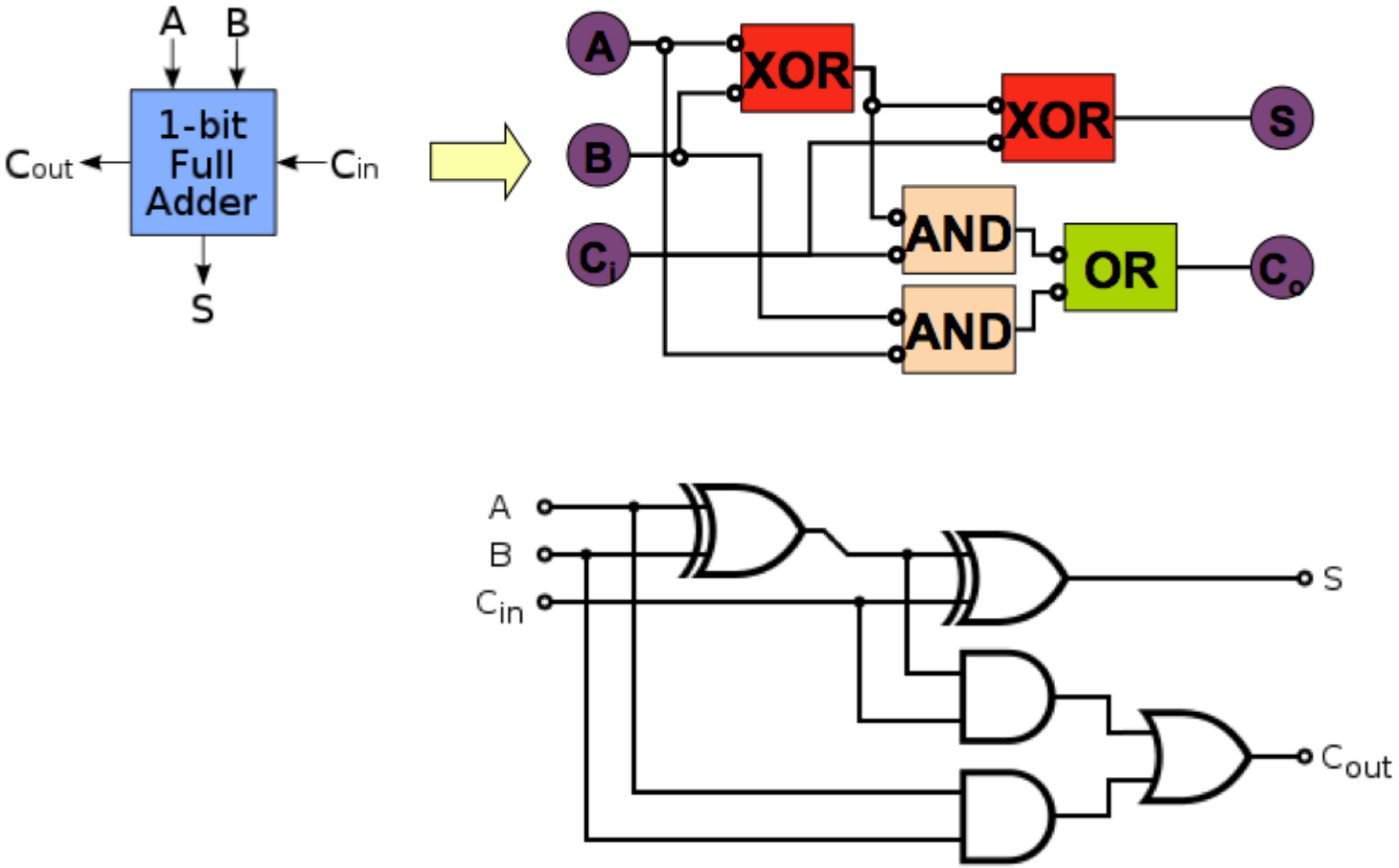
Gates are "boxes" containing transistors

They *implement* boolean logic

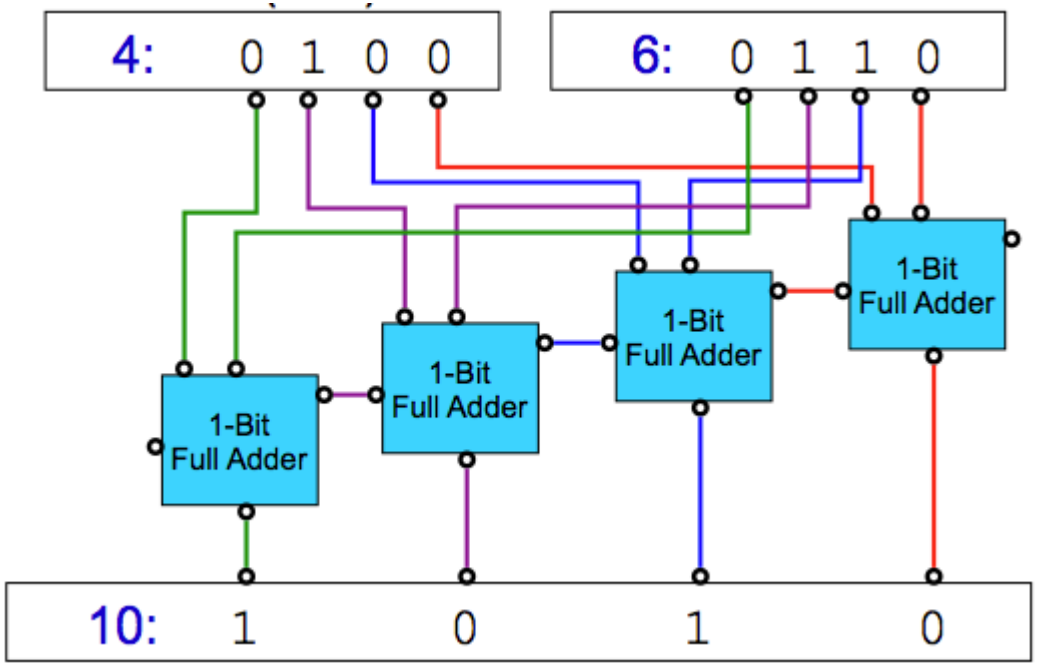
Arithmetic units are "boxes" containing gates



Arithmetic - Addition (1 bit)



Arithmetic - Addition - 4-bit



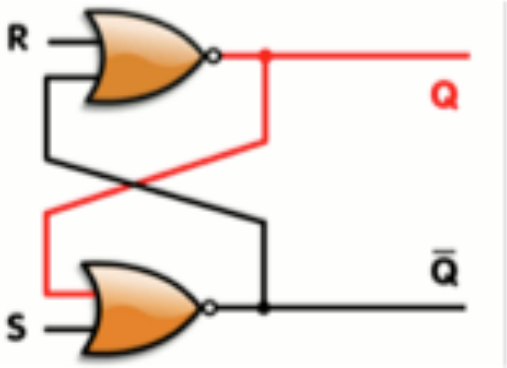
Timing: The left most-bits need to wait until the carry from the right bits is through the circuit

On a 64-bit machine, its the same design - just more!

Remembering Numbers

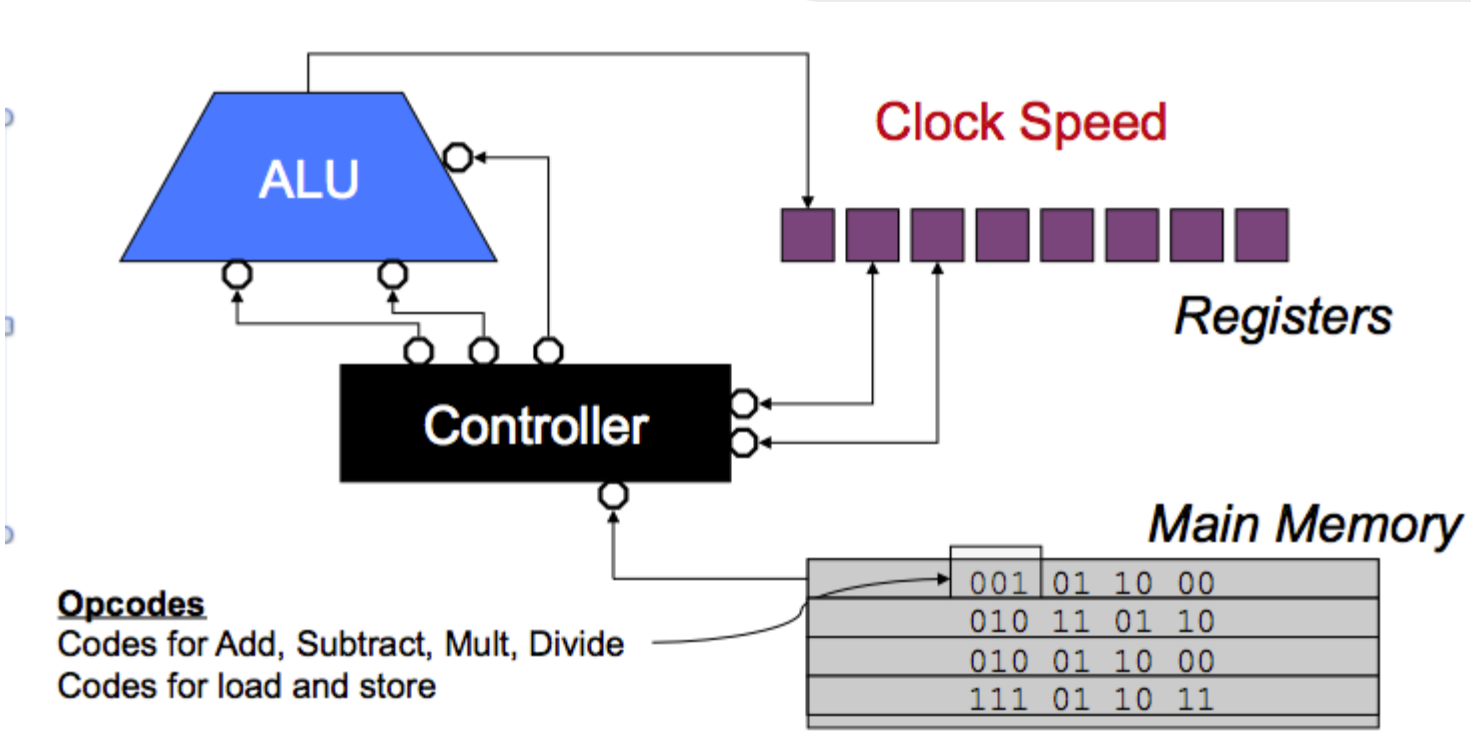
Inputs and outputs must be "held" in memory so they can be used in operations

Special electronic components can be toggled from "on" and "off" - called *flip-flops*



The circuit requires power though… so the machine needs to stay on!

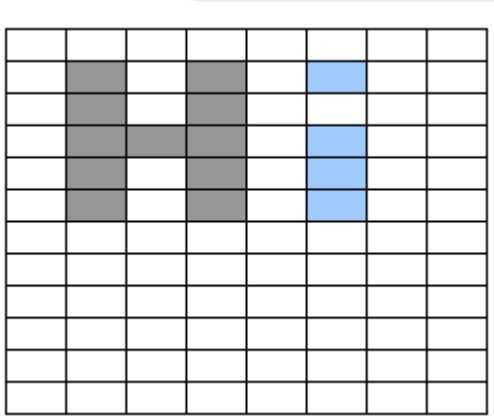
The Central Processing Unit



Clock speed is governed by the time it takes for electricity to flow through the entire systems

Output

A computer screen is divided into a grid



Each grid point is a *pixel*

To output anything (shapes, text, etc.)

- Determine where you want to color the screen (pixel number)
- Encode color to a number
- Store the number in a special place in memory (frame buffer)

Encoding Color

- Colors are specified by a series of bits
- The variation in color is controlled by how many bits you use to represent color

	0: Black
	1: White

24 bits gives you enough to represent all the colors the human eye can detect...

	000: Black
	001: Blue
	010: Green
	011: Cyan
	100: Red
	101: Magenta
	110: Yellow
	111: White

Text - input and output

- The keyboard simply maps keys to numbers
- The computer maps numbers back to characters to draw (pixels!)

PRINTABLE CHARACTERS								
DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
32	0x20	<SPACE>	64	0x40	@	96	0x60	`
33	0x21	!	65	0x41	A	97	0x61	a
34	0x22	"	66	0x42	B	98	0x62	b
35	0x23	#	67	0x43	C	99	0x63	c
36	0x24	$	68	0x44	D	100	0x64	d
37	0x25	%	69	0x45	E	101	0x65	e
38	0x26	&	70	0x46	F	102	0x66	f
39	0x27	'	71	0x47	G	103	0x67	g
40	0x28	(	72	0x48	H	104	0x68	h
41	0x29	)	73	0x49	I	105	0x69	i
42	0x2A	*	74	0x4A	J	106	0x6A	j
43	0x2B	+	75	0x4B	K	107	0x6B	k
44	0x2C	,	76	0x4C	L	108	0x6C	l
45	0x2D	-	77	0x4D	M	109	0x6D	m
46	0x2E	.	78	0x4E	N	110	0x6E	n

Binary Programs

In the end - *instructions* are just a series of bits - where each number corresponds to an operation

- Add/Subtract, Multiply/Divide
- Load and Store number from memory location

Our job as programmers is to tell the computer what to do - using only these basic operations!

Thankfully, we don't need to program in binary, or these simple terms - we have higher level languages that we can use.

Computers are **not** smart

If you remember nothing else from this lecture - remember this!

Computer do **exactly** what you tell them to do

Never more...

Never less...

So don't get angry :)