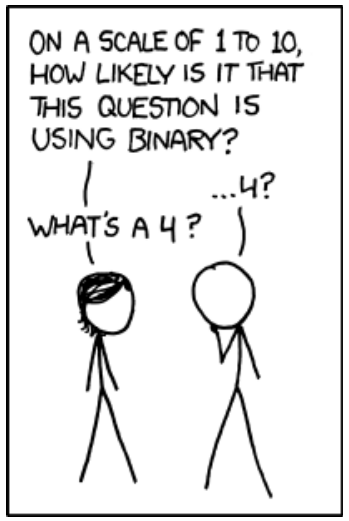
Weather: Temperatures around 1100000°_{2} F
Sports: CS Professor runs 100_2 meter dash in under 3_{10} seconds!

CS 101010 Today

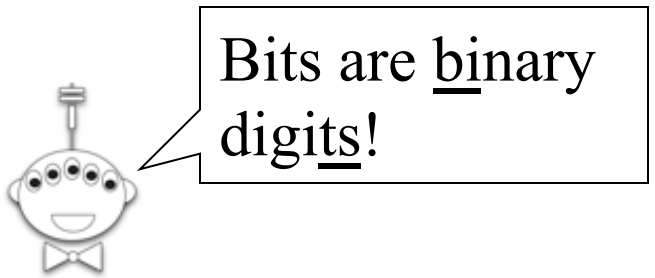
California to adopt binary as official number system

(Claremont, AP): Professors at Harvey Mudd College have successfully lobbied the state legislature to go "binary": dropping base 10 in favor of base 2. "There are a lot efficiencies and cost savings to be gained by this move," said one CS prof who has been working for his entire career to see this change. "For example, keyboards will have only two numbers on their numeric key pads, resulting in a substantial savings in plastic which will, in turn, contribute to a cleaner environment." State legislators were evidently most swayed by the argument that California would collect substantially more revenue in the new system, allowing the state to balance its budget immediately and even show a substantial surplus. "For example, right now a typical traffic ticket is $50. After converting to base 2, it will be $110010. That's a lot of revenue for the state! A person who paid $3000 in state taxes each year will now pay $101110111000!" However, some skeptics noted that this enthusiasm is misguided. "It looks like a silver bullet, but notice that there will be substantial costs in converting all highway signs to binary." The state Highway Patrol noted that the speed limit will now increase to 1000001 miles/hour which may have other undesirable consequences. "We're upgrading our police vehicles to reach these new higher speeds," said one official.

Binary Review



- What is 24 in binary?
- What is 110111_2 in decimal?
- What's the largest number we represent with 3 bits? 7 bits? n bits?



"Quiz"

There are 10 kinds of "people" in the universe: those who use binary, and those who do not!

Name:

Convert these two binary numbers to decimal:

```
32 16 8 4 2 1
110011   10001000
```

Convert these two decimal numbers to binary:

28 **101**

Add these two binary numbers:

```
   101101
+     1110
--------
```

WITHOUT converting to decimal !

Multiply these binary numbers:

```
   101101
*     1110
--------
```

```
  1
  529
+ 742
 1271
```

Hint: **Remember this algorithm? It's the same…**

ASCII Codes

American Standard Code for Information Interchange

0	nul	1	soh	2	stx	3	etx	4	eot	5	enq	6	ack	7	bel
8	bs	9	ht	10	nl	11	vt	12	np	13	cr	14	so	15	si
16	dle	17	dc1	18	dc2	19	dc3	20	dc4	21	nak	22	syn	23	etb
24	can	25	em	26	sub	27	esc	28	fs	29	gs	30	rs	31	us
32	sp	33	!	34	"	35	#	36	$	37	%	38	&	39	'
40	(	41	)	42	*	43	+	44	,	45	-	46	.	47	/
48	0	49	1	50	2	51	3	52	4	53	5	54	6	55	7
56	8	57	9	58	:	59	;	60	<	61	=	62	>	63	?
64	@	65	A	66	B	67	C	68	D	69	E	70	F	71	G
72	H	73	I	74	J	75	K	76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S	84	T	85	U	86	V	87	W
88	X	89	Y	90	Z	91	[	92	\	93	]	94	^	95	_
96	`	97	a	98	b	99	c	100	d	101	e	102	f	103	g
104	h	105	i	106	j	107	k	108	l	109	m	110	n	111	o
112	p	113	q	114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124	|	125	}	126	~	127	del

Why are there six symbols sitting between the upper-case and lower-case letters?

Shifting

- What happens to binary numbers when we move bits to the right or left?

101000 ← 10100 ← 1010 → 101 → 10

Negative Numbers?

- With 3 bits (binary digits), we can represent the numbers 0..7

- But what about negative numbers?
  - Sign/Magnitude (used for “floating point”)
  - 2’s complement (used for integers)

Modular Arithmetic

- If we are doing addition “mod 8”, what happens to numbers when we add
  1 ?
  7 ?
  6 ?

Signed vs. Unsigned Numbers

- Unsigned interpretation:
  n bits let us represent the numbers
  $0, 1, 2, \ldots, 2^n-1$

- Signed (2's complement) interpretation:
  n bits let us represent the numbers
  $0, 1, 2, \ldots, 2^{n-1}-1, \ -2^{n-1}, -2^{n-1}+1, \ldots, -2, -1$

- Best of all: addition and subtraction don't care which interpretation you're using!

2's Complement: Finding the negative

- To find the negative of a number in 2's complement:
  - Subtract from 2^n
  - Or, invert all the bits and add 1.

- What is -3 as an 8-bit signed number?

Octal and Hexadecimal

- Bases that are powers of 2 are easy to convert to/from binary, because you can work digit-by-digit!

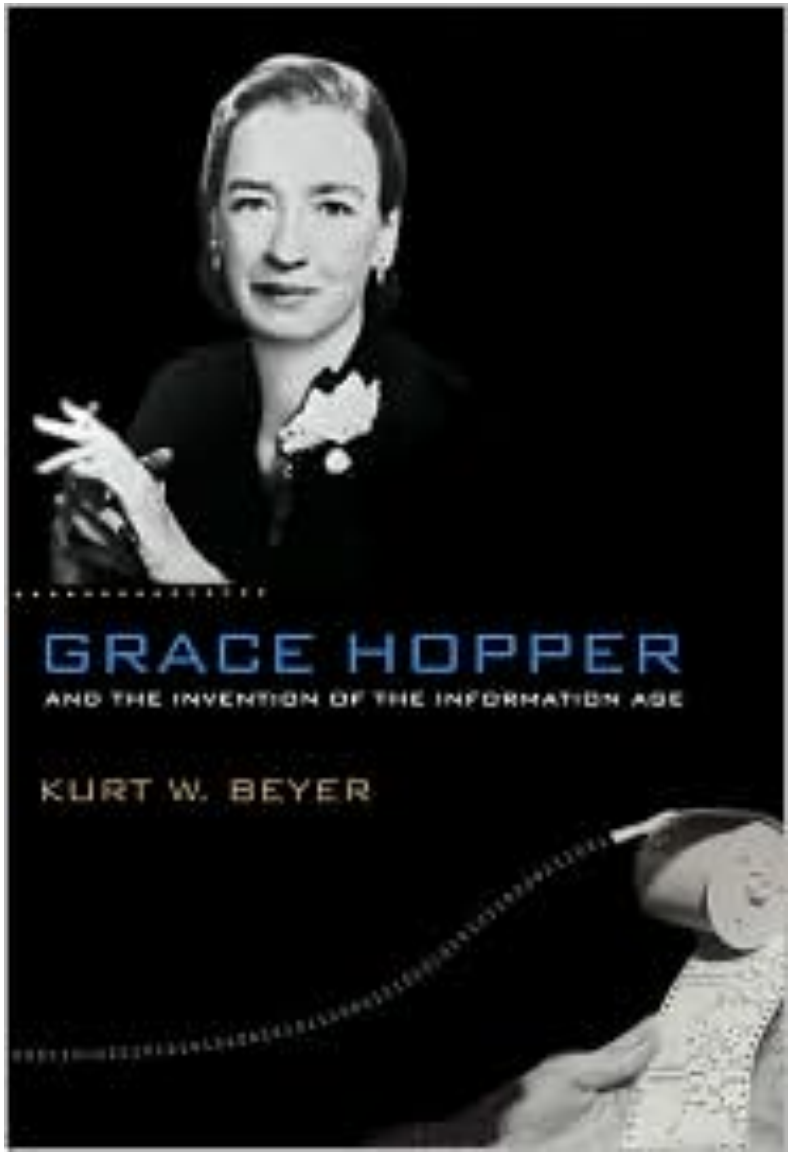
Convert 0755 from octal to binary

Convert 0xDEADBEEF from hexadecimal to binary

0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Rear Admiral Grace Murray Hopper



- "the third programmer of the first modern computer."
- Inventor of the first compiler
- Driving force behind Cobol

  "They told me computers could only do arithmetic."

Of course, coding took a toll on people's ability to think correctly while away from the computer. The Mark I had to be coded with octal digits, so every month Hopper's "checkbook would be unbalanced because she would slip into octal addition and subtraction when balancing her accounts."

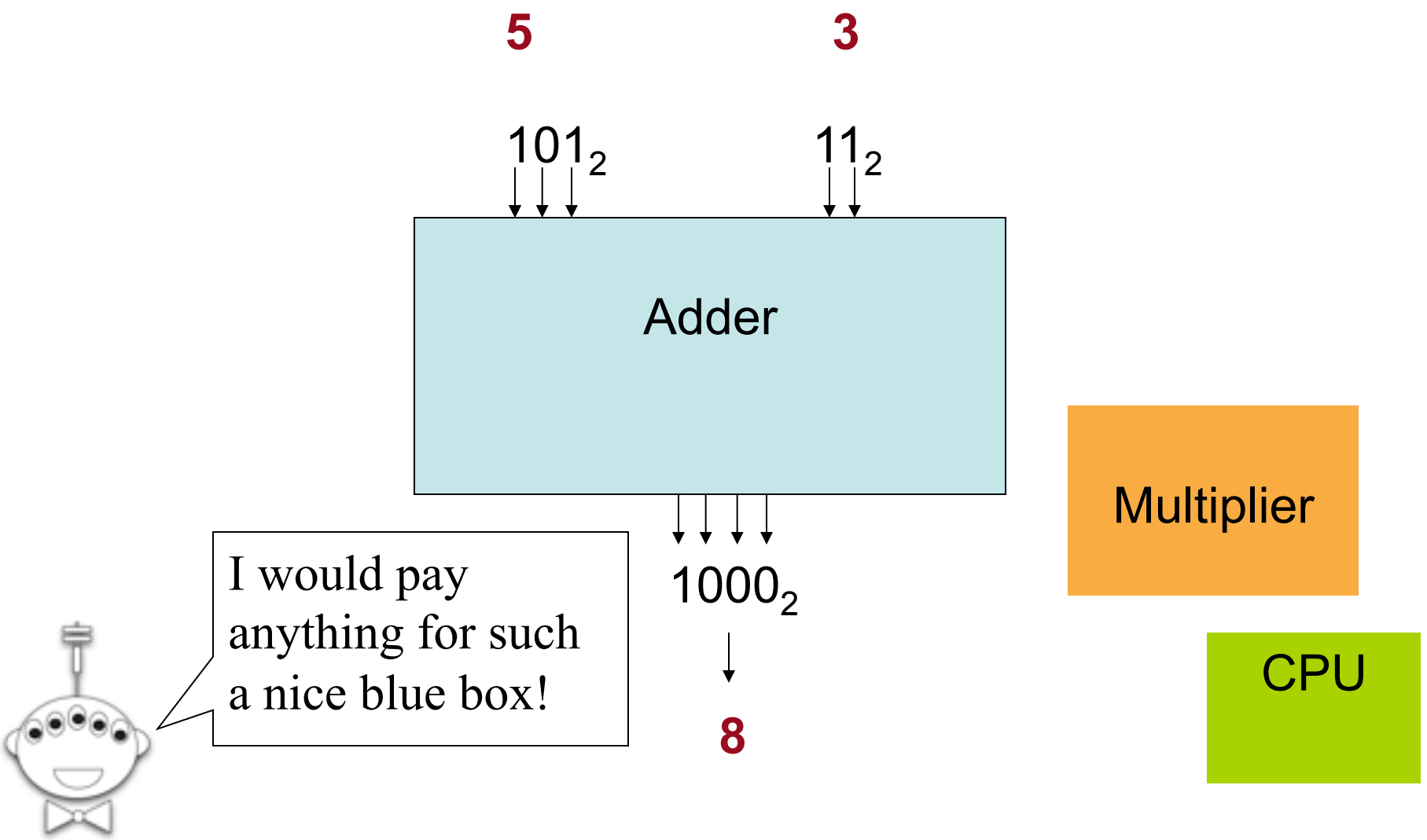
A Recent Application of Octal

```
knuth: www/assignments>  ls -l a*.html
-rw-r--r-- 1 stone infosys 33131 Sep  3 18:21 a1.html
-rw-r--r-- 1 stone infosys 27267 Sep 10 16:11 a2.html
-rw-r--r-- 1 stone infosys 24093 Sep 13 12:38 a3.html
-rw-r--r-- 1 stone infosys 16207 Sep 22 14:46 a4.html
-rw------- 1 stone infosys 35406 Sep 28 15:02 a5.html

knuth: www/assignments% chmod 644 a5.html

knuth: www/assignments% ls -l a*.html
-rw-r--r-- 1 stone infosys 33131 Sep  3 18:21 a1.html
-rw-r--r-- 1 stone infosys 27267 Sep 10 16:11 a2.html
-rw-r--r-- 1 stone infosys 24093 Sep 13 12:38 a3.html
-rw-r--r-- 1 stone infosys 16207 Sep 22 14:46 a4.html
-rw-r--r-- 1 stone infosys 35406 Sep 28 15:02 a5.html
```

Computing Digitally



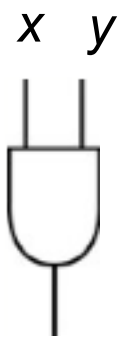
Digital Logic Gates

x	NOT x
0	1
1	0

Also written $\overline{x}$

x	y	x AND y
0	0	0
0	1	0
1	0	0
1	1	1

Also written xy



x AND y

x	y	x OR y
0	0	0
0	1	1
1	0	1
1	1	1

Also written $x+y$

x OR y

From Description to Circuit!

Words

f is a function of TWO binary variables s.t. the output is *1* if and only if exactly one of the two inputs is *1*

Table

x	*y*	*f(x,y)*
0	*0*	*0*
0	*1*	*1*
1	*0*	*1*
1	*1*	*0*

Formula

Circuit

From Description to Circuit!

Abstract → Concrete

Words

f is a function of TWO binary variables s.t. the output is *1* if and only if exactly one of the two inputs is *1*

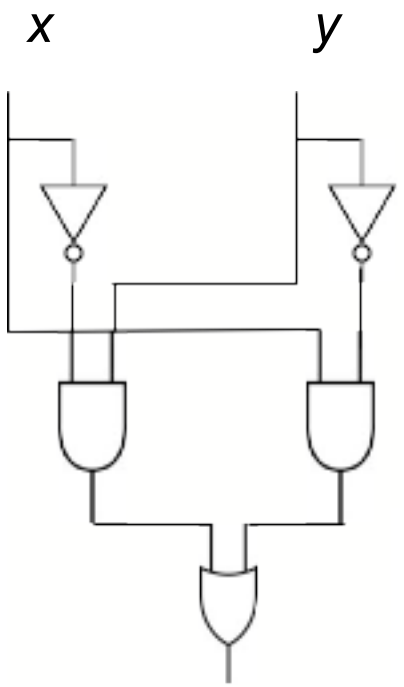
Table

x	*y*	*f(x,y)*
0	*0*	*0*
0	*1*	*1*
1	*0*	*1*
1	*1*	*0*

Formula

$\bar{x}y + x\bar{y}$

Circuit



That's nothin' but a two-bit adder!

Actually, it's a "half adder"

Try building circuits for these…

x	*y*	output
0	0	1
0	1	0
1	0	0
1	1	0

x	*y*	output
0	0	1
0	1	0
1	0	0
1	1	1

x	*y*	*z*	output
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

YIKES!

Try This One (WORKSHEET)…

Consider this function…

Words	Truth Table	Formula
A function of THREE binary inputs *x,y,z* where the output is 1 iff the number of 1's is odd		Circuit

This is called an "odd" "parity" circuit.

Just cuz it's odd doesn't mean you should parity it!

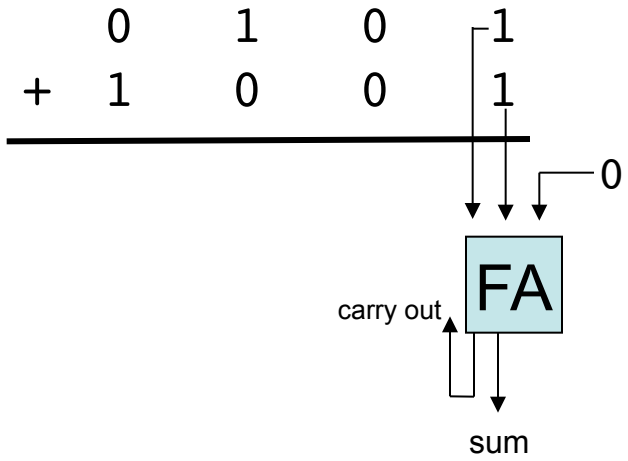
A Circuit for Adding!

Base 2 Addition

```
    0   1   0   1
+   1   0   0   1
-------------------
```

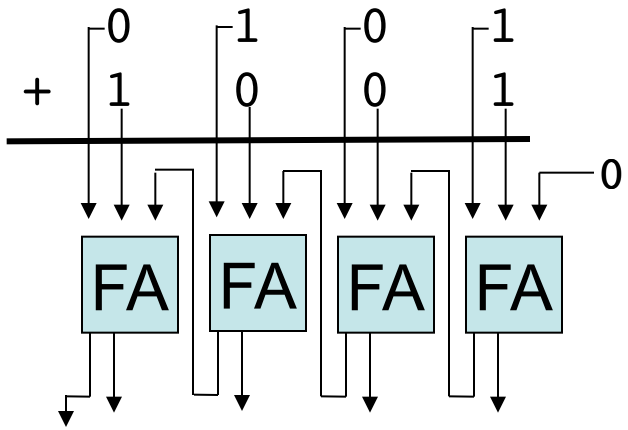
A Circuit for Adding!

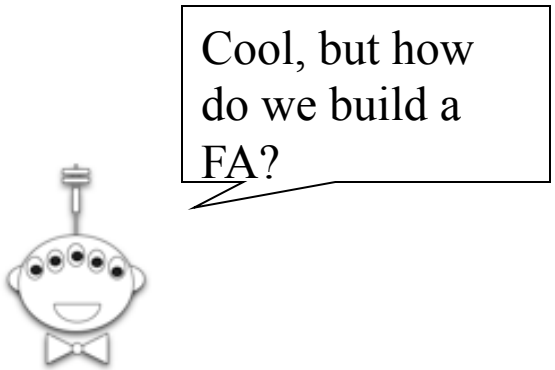
Base 2 Addition



A Circuit for Adding!

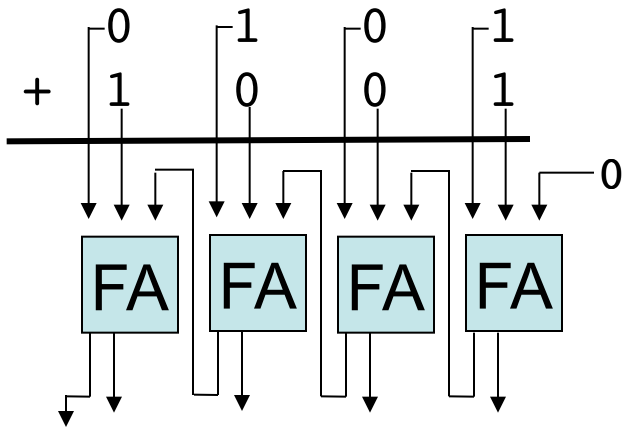
Base 2 Addition

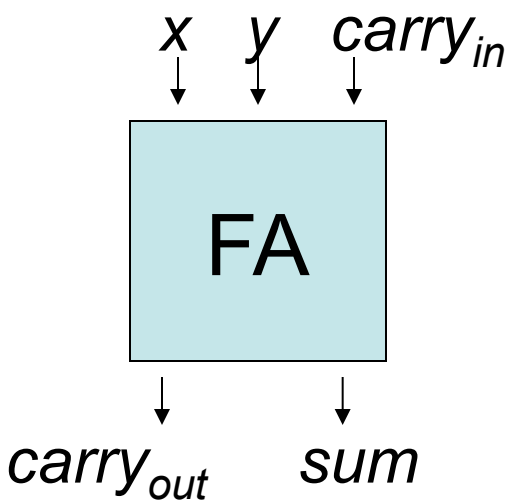




A Circuit for Adding!

Base 2 Addition

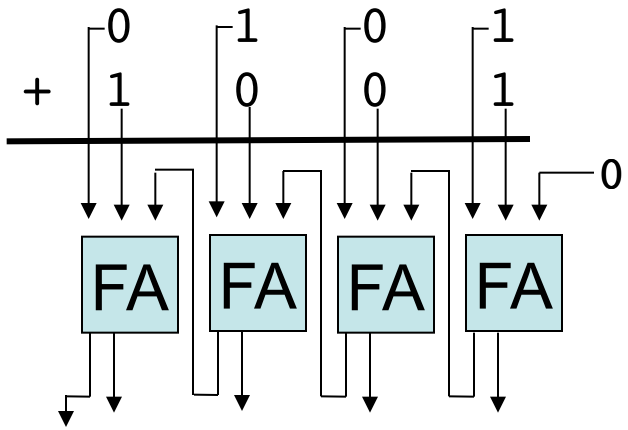


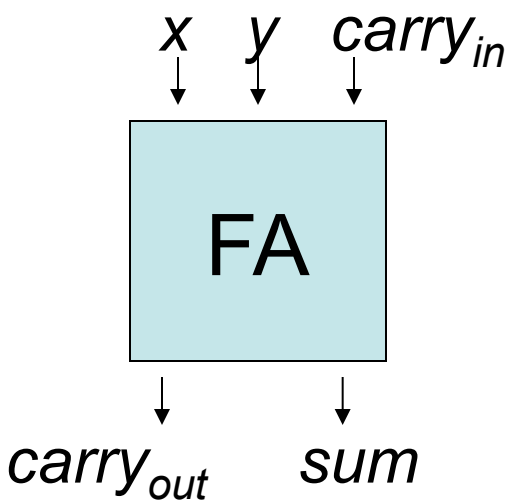


x	y	$carry_{in}$	sum	$carry_{out}$
0	0	0	0	0
0	0	1	1	0

A Circuit for Adding!

Base 2 Addition



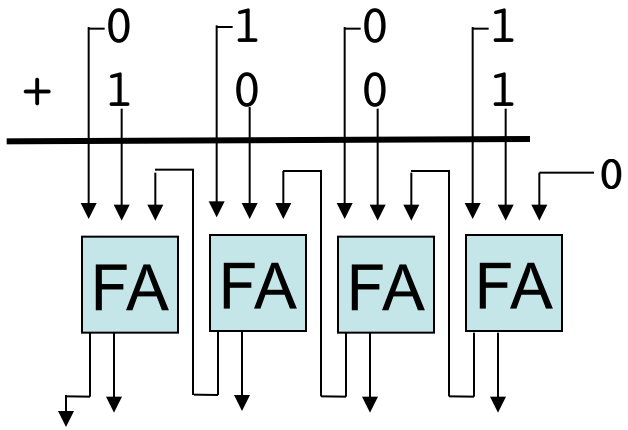


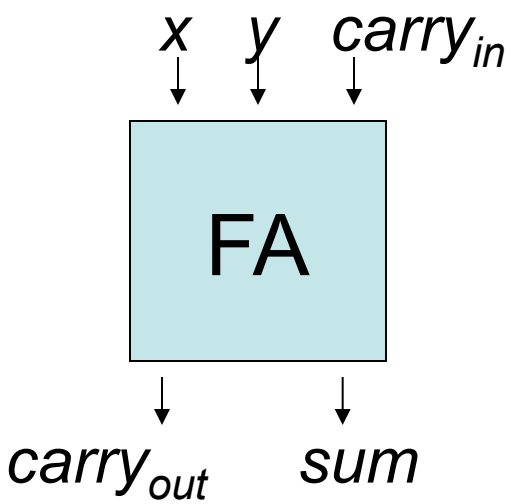
x	y	$carry_{in}$	sum	$carry_{out}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Now what!??

A Circuit for Adding!

Base 2 Addition



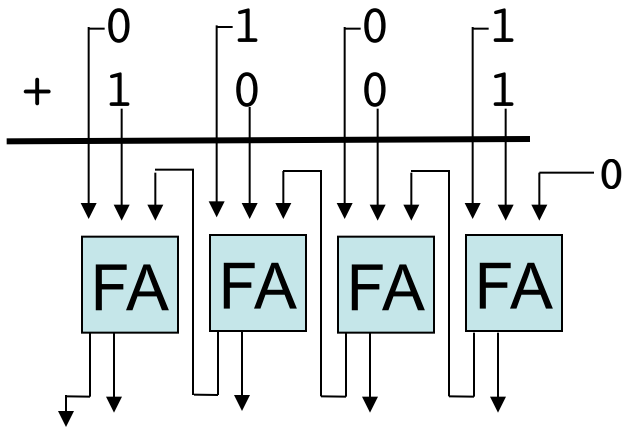


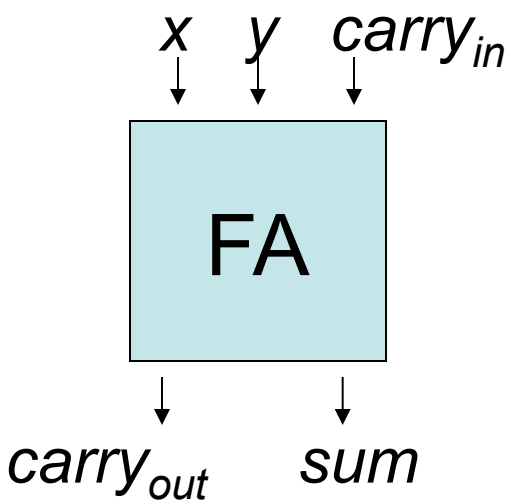
x	y	$carry_{in}$	sum	$carry_{out}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

This looks vaguely familiar…

A Circuit for Adding!

Base 2 Addition

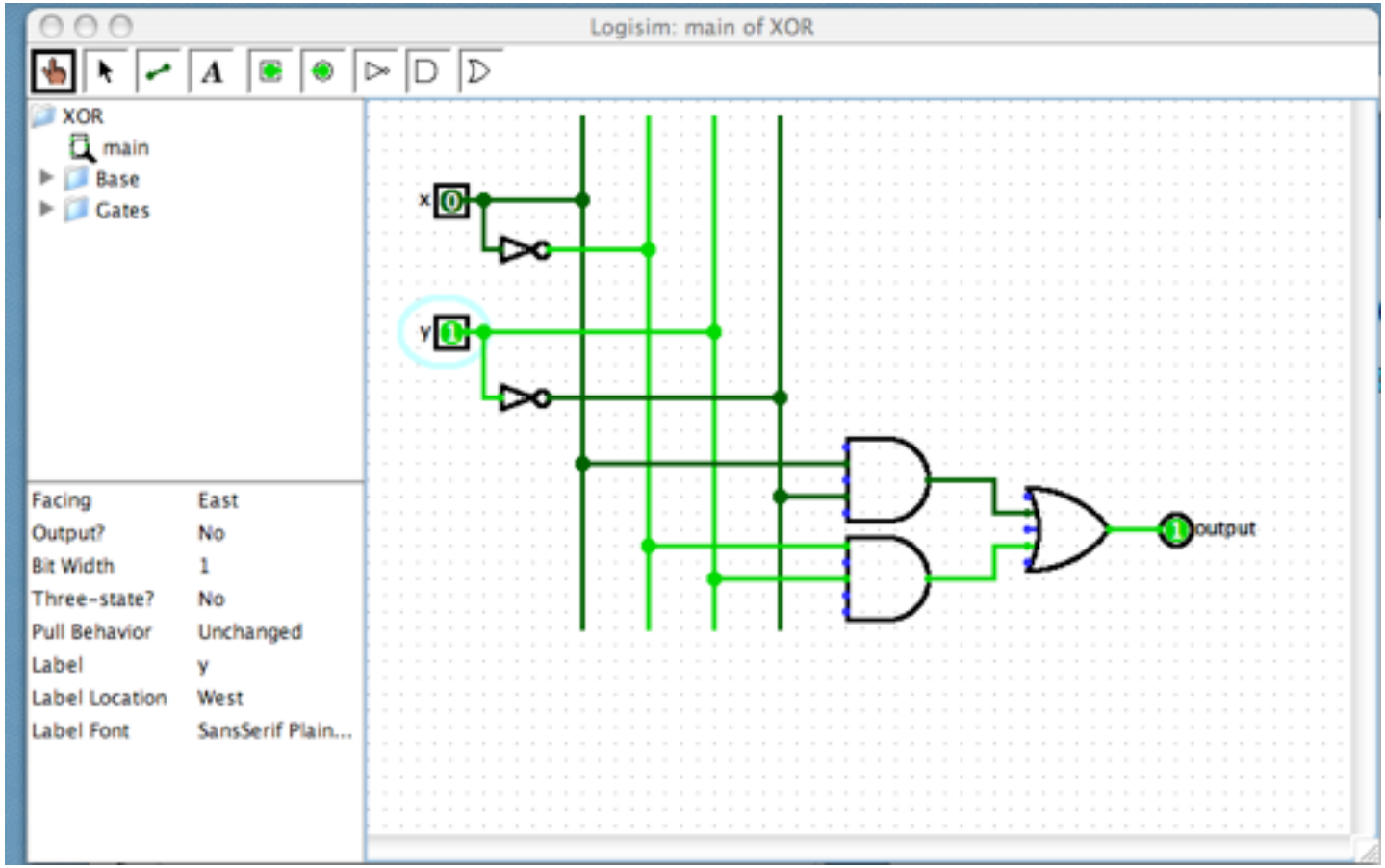




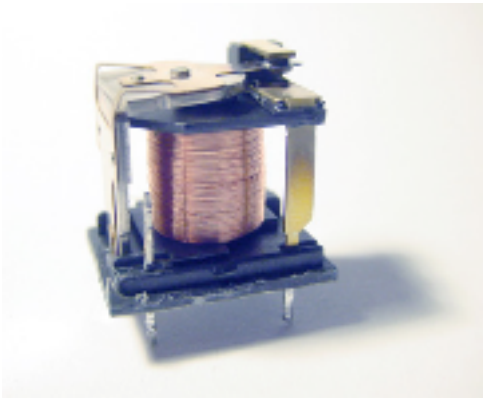
x	y	$carry_{in}$	sum	$carry_{out}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

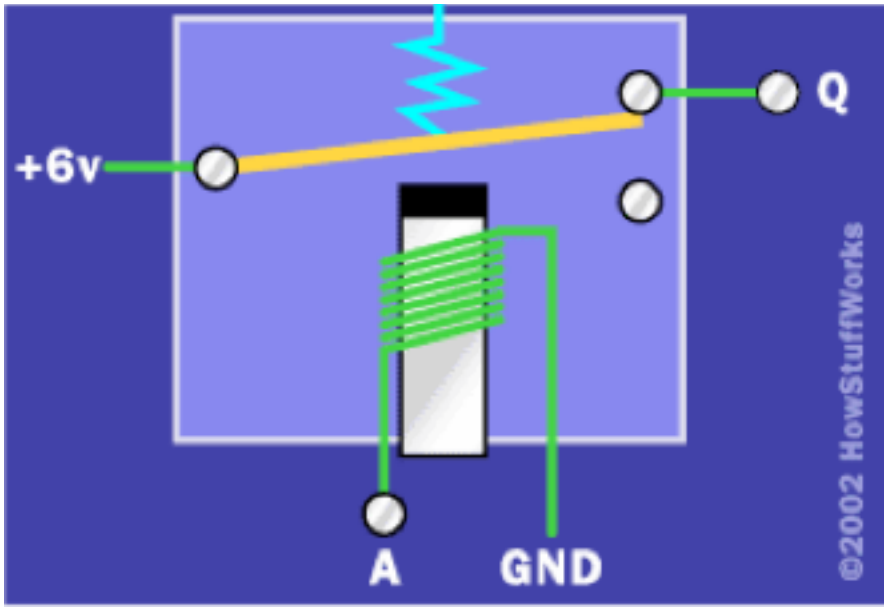
Ah! I know how to do that!

Logisim: A Digital Logic Simulator

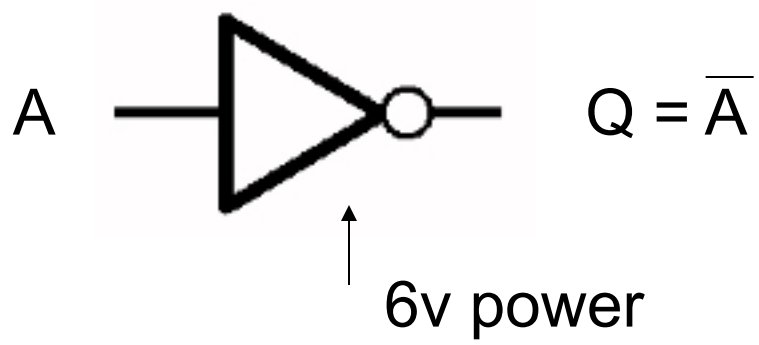


Implementing Gates with Relays



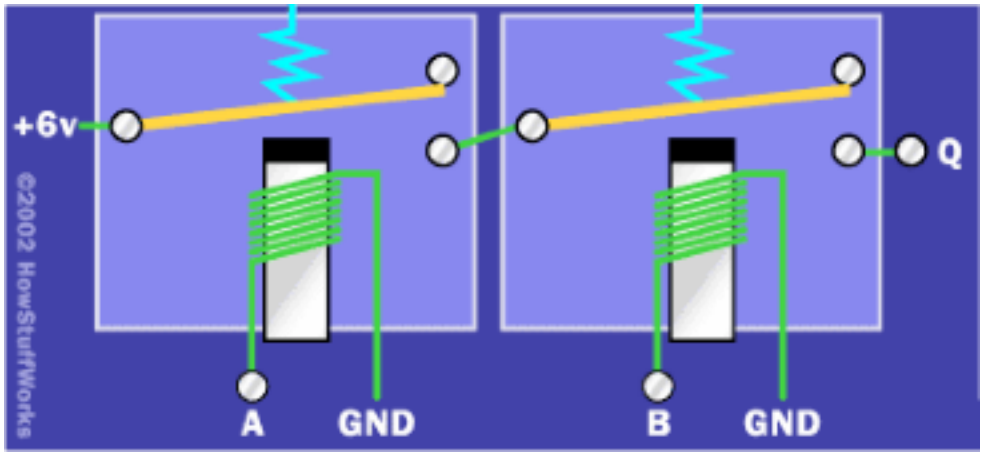


NOT Gate



Figures courtesy of HowStuffWorks.com

And AND...



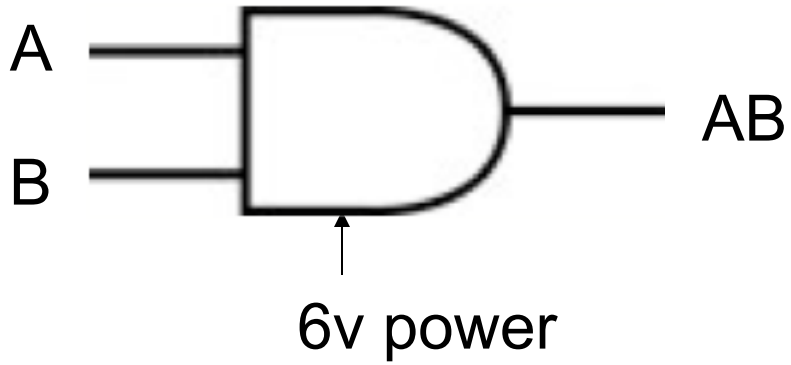
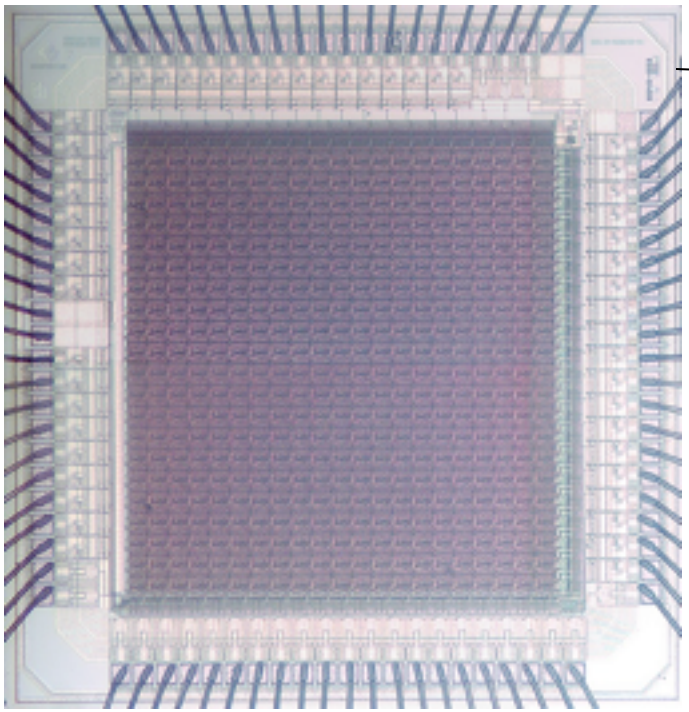


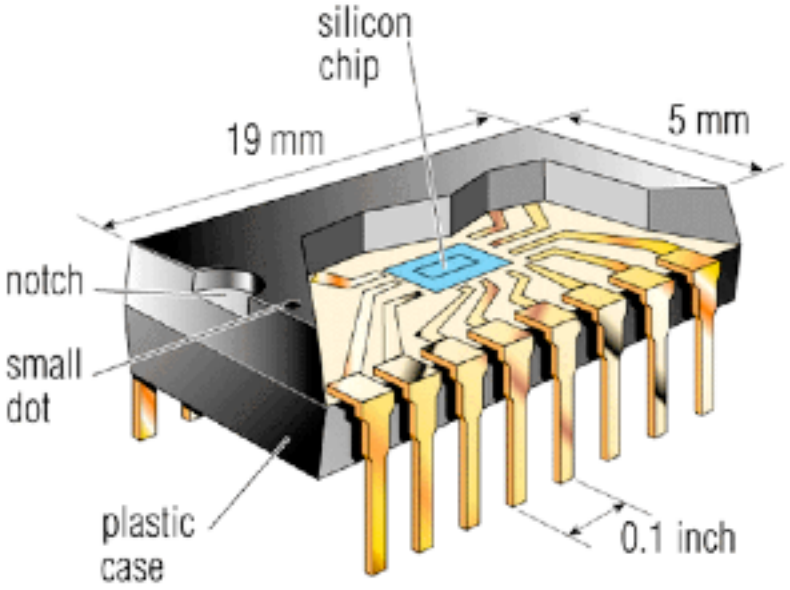
Figure courtesy of HowStuffWorks.com

Integrated Circuits



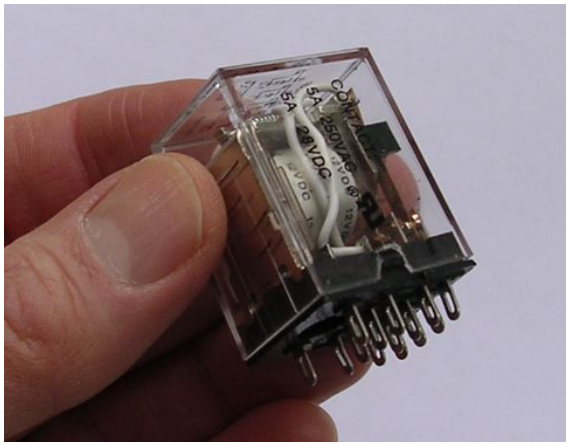
3mm

http://personalpages.manchester.ac.uk/staff/p.dudek/projects/scamp



http://www.helicon.co.uk

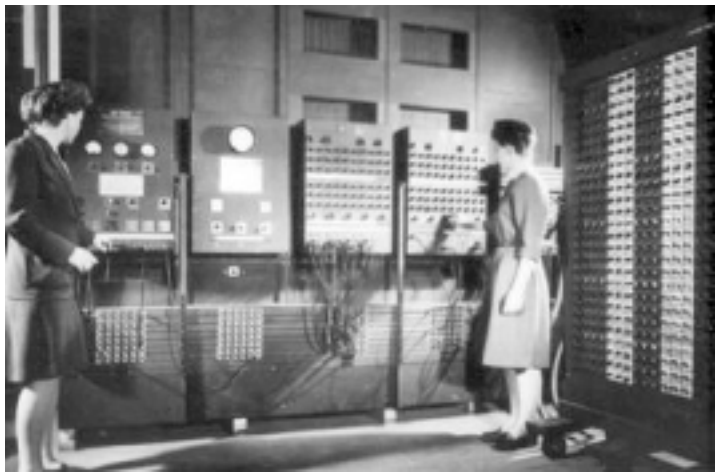
A real early relay computer



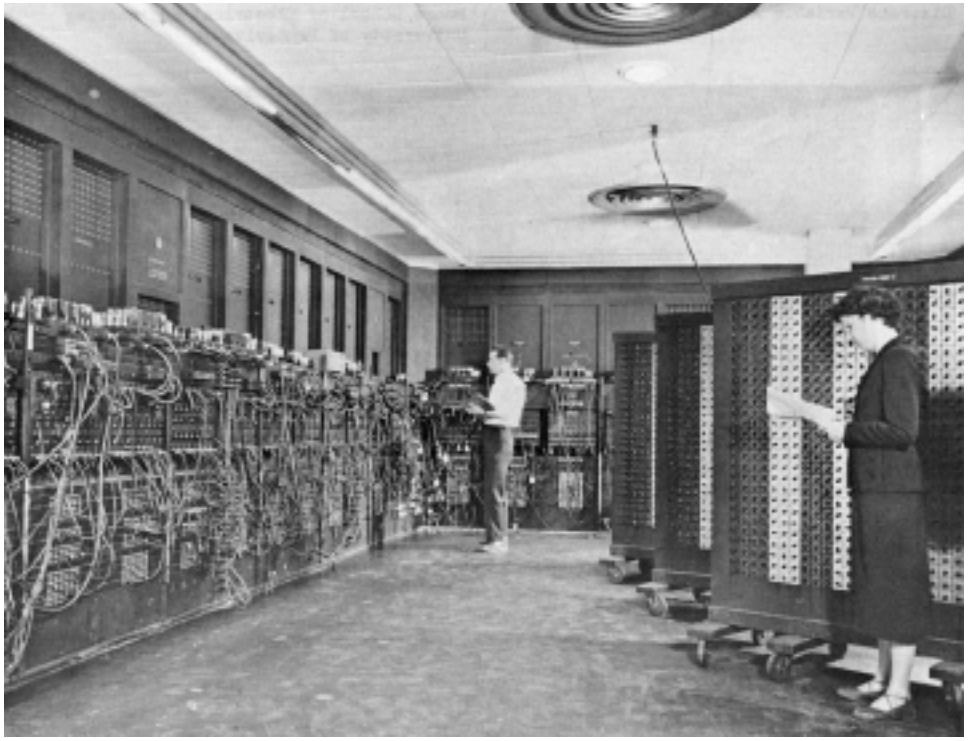
REALY
EARLY
RELAY!
(LEARY?)

See the anagram solver at http://www.ssynth.co.uk/~gay/anagram.html

Relays to Vacuum Tubes, …



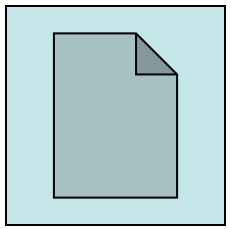
A vacuum tube



Eniac, 1946

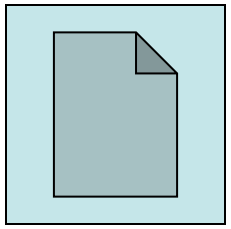
Images courtesy of en.wikipedia.org

Digital Logic with Water and Legos!



Hydraulic Logic Gates!

http://www.cs.princeton.edu/introcs/lectures/fluid-computer.swf



Lego Logic Gates!

http://goldfish.ikaruga.co.uk/logic.html

AND, OR, NOT is a "Universal" Set!

x	y	z	Output
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
0	1	0	0
1	1	1	1

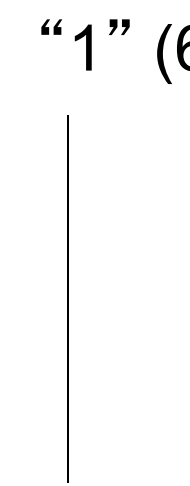
$\bar{x}\,\bar{y}\,\bar{z}$

The Minterm Expansion Principle Revisited!

x_1	x_2	x_3 ...	x_{100}	output
0	0	0	0	1
0	0	0	1	1
■	■	■		
1	1	1	1	1

The Minterm Expansion Principle Revisited!

x_1	x_2	x_3 ...	x_{100}	output
0	0	0	0	1
0	0	0	1	1
■	■	■		
1	1	1	1	1



Bad News! You're Stranded on a Desert Island…

Good News! There's no homework on this island….

Bad News! There are no computers on this island…

Good news! A Radio Shack!

So we can build our own computer from scratch…

Better news! Starbucks!

… while sipping mint mocha frappacinnos!

Bad news! They're out of OR gates

AND

NOT

Can we implement *every possible* boolean function with AND and NOT gates alone???

Are AND and NOT a Universal Set?

De Morgan's Laws:

$$\overline{x\,y} = \bar{x} + \bar{y}$$

$$\overline{x+y} = \bar{x}\,\bar{y}$$

Augustus De Morgan: 1806-1871

Is there a single type of gate that forms a Universal Set?

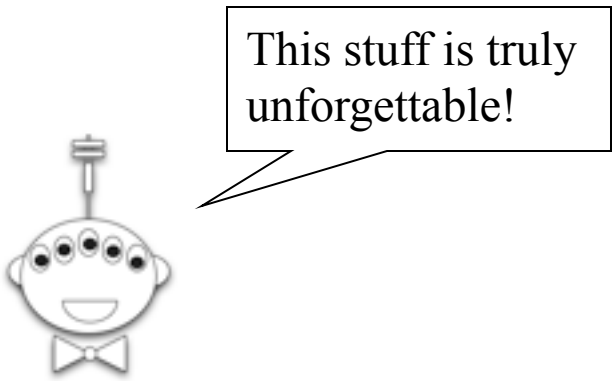
De Morgan's Laws:

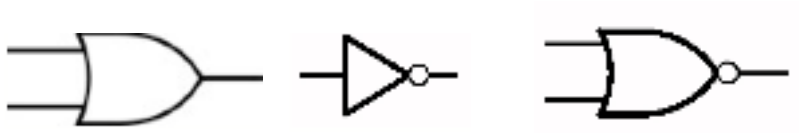
$$\overline{x\,y} = \bar{x} + \bar{y}$$

$$\overline{x+y} = \bar{x}\,\bar{y}$$

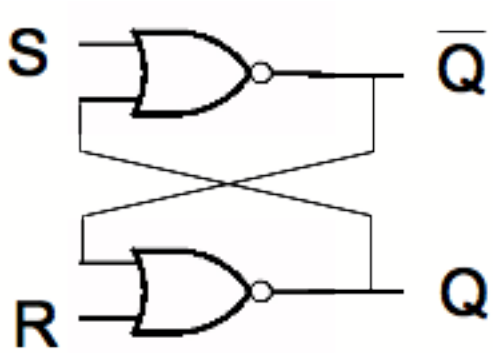
Augustus De Morgan: 1806-1871

A 1-bit Memory





OR + NOT = NOR



From S-R Latches to D-Latches