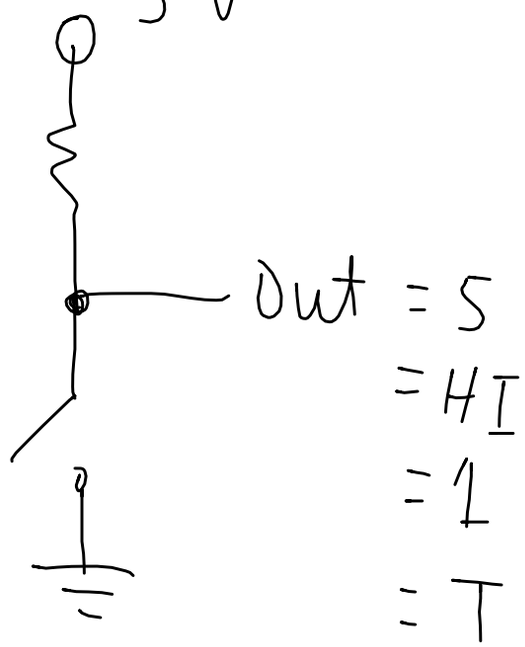
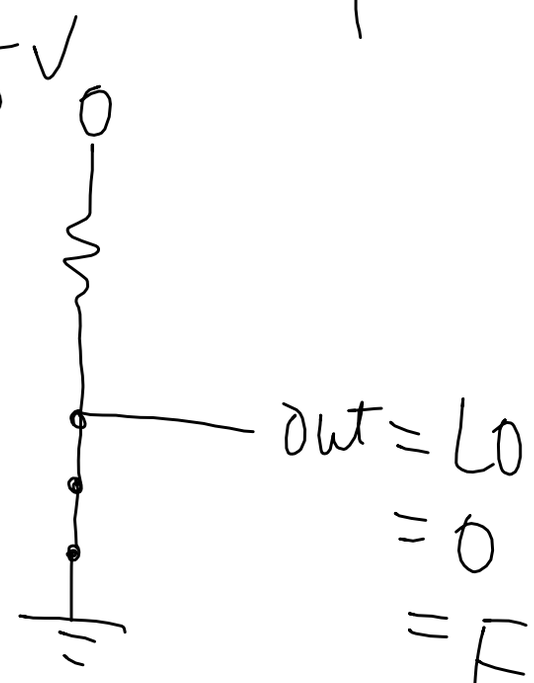
Digital Electronics

Two states: HI = 1 = T

Low = 0 = F





Replace switch w/ transistor + voila

The math ....

- Binary ⇒ 0, +1 ⇒ base 2
- Octal ⇒ 0 → 7 ⇒ base 8
- Hexadecimal ⇒ (0 → 15) → 0-9, A → F ⇒ base 16

$$\begin{array}{r} {\scriptstyle 1\ \ \ } \\ +\ 1 \\ \hline 10 \end{array} \qquad \begin{array}{r} {\scriptstyle 1\ 1\ \ } \\ 101 \\ +\ 011 \\ \hline 1000 \end{array}$$

what about subtraction?

2 ways: Sign bit

0101 = 5

1101 = -5

$$\begin{array}{rr} 0101 & 5 \\ +\ 1101 & -5 \\ \hline 10010 & = 2 \end{array}$$

Bad

Second way:

101 ⇒ 5

010 one's complement

\+ 1

011 two's complement

101

\+ 011

1000

Terminology:

0, 1 ⇒ bits

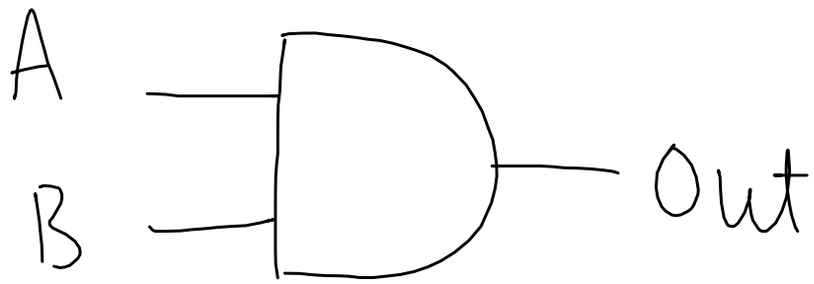
01010101 ⇒ byte

(0101 → nibble)

Moving them:

1. Serial - one bit after another
2. Parallel - all n bits at once

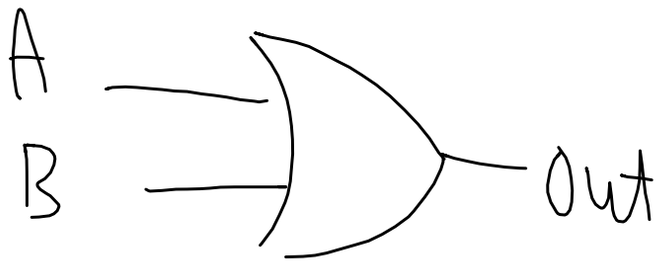
Logic Gates



A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

AND

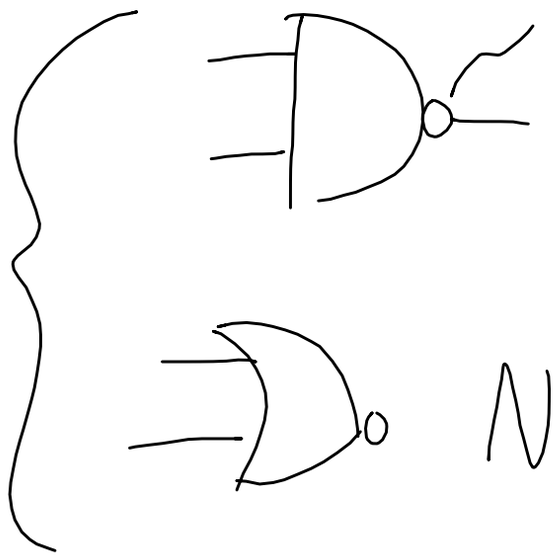
$A \cdot B$



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

OR

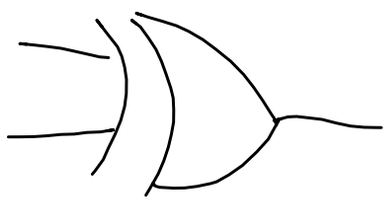
$A + B$



NAND = NOT AND $\overline{A \cdot B}$

NOR

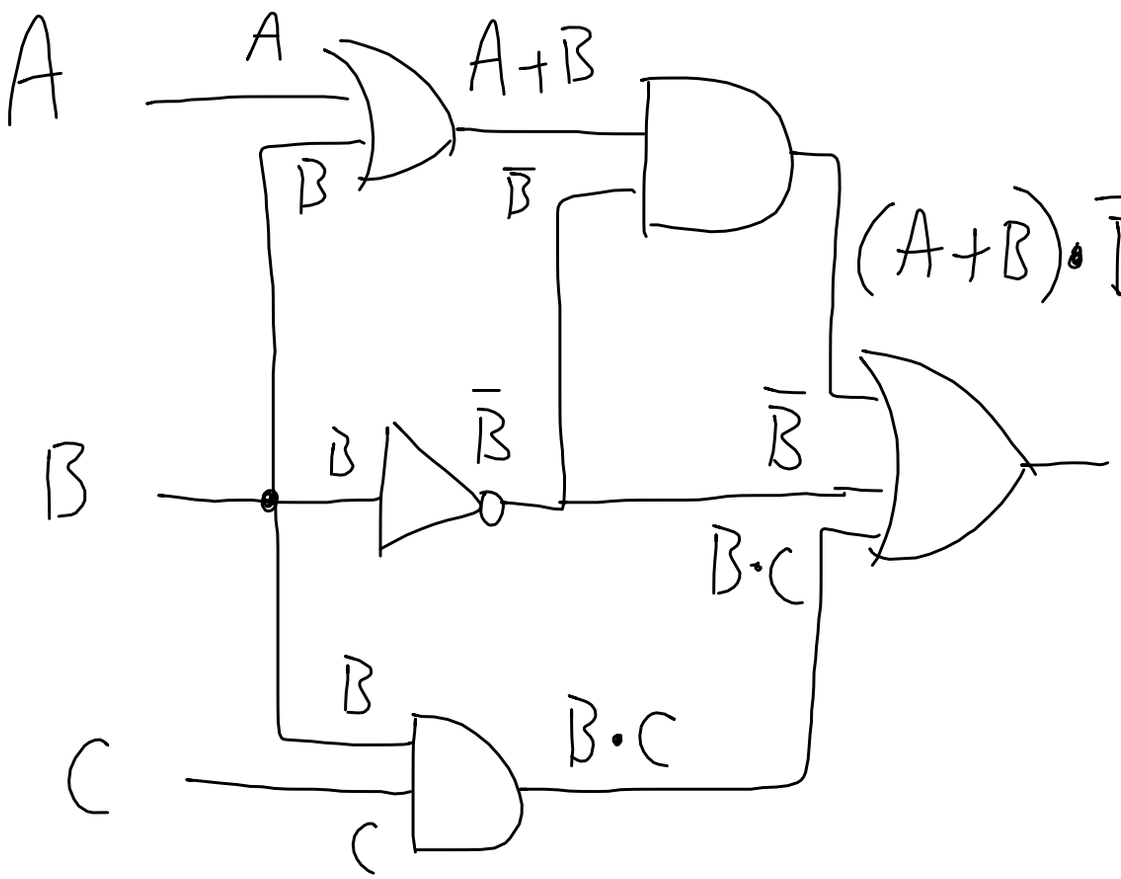
Can make any 2-input gate from combination of NANDs, NORs, & inverters



XOR

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

There are theorems that help simplify the logic



$(A+B)\cdot\overline{B} + \overline{B} + B\cdot C$

Reduces to

$\overline{B} + C$

One handy theorem:

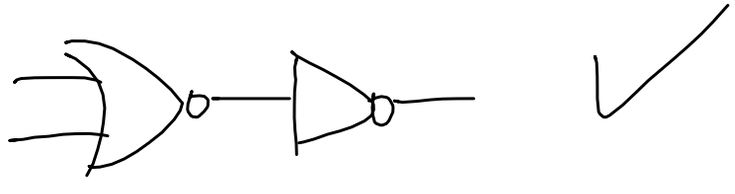
De Morgan's theorem

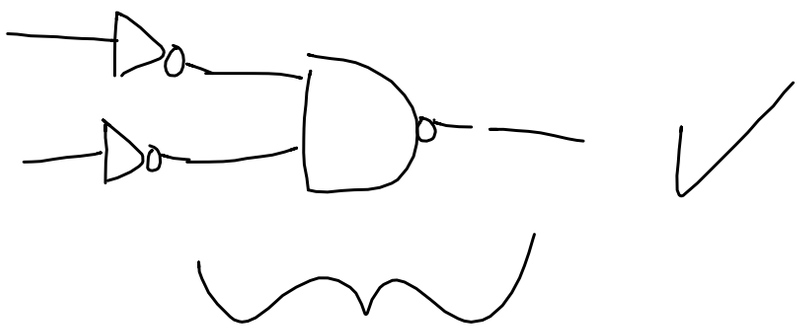
$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

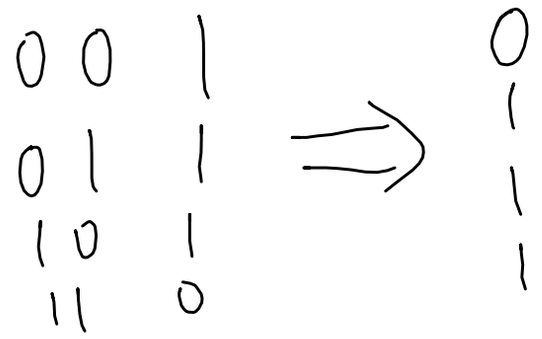
+

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

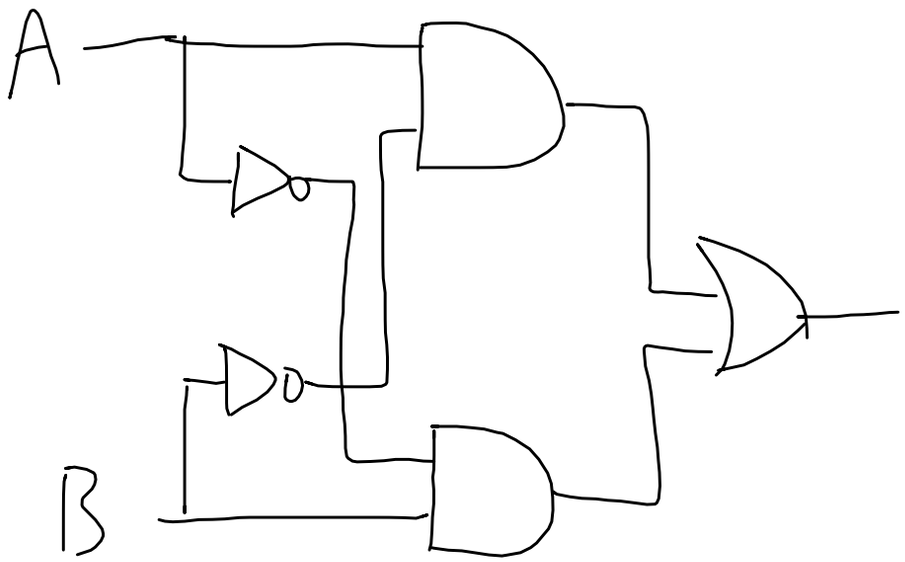
Try this: using only NANDS, NORs & inverters,
make an OR gate





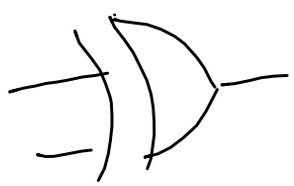


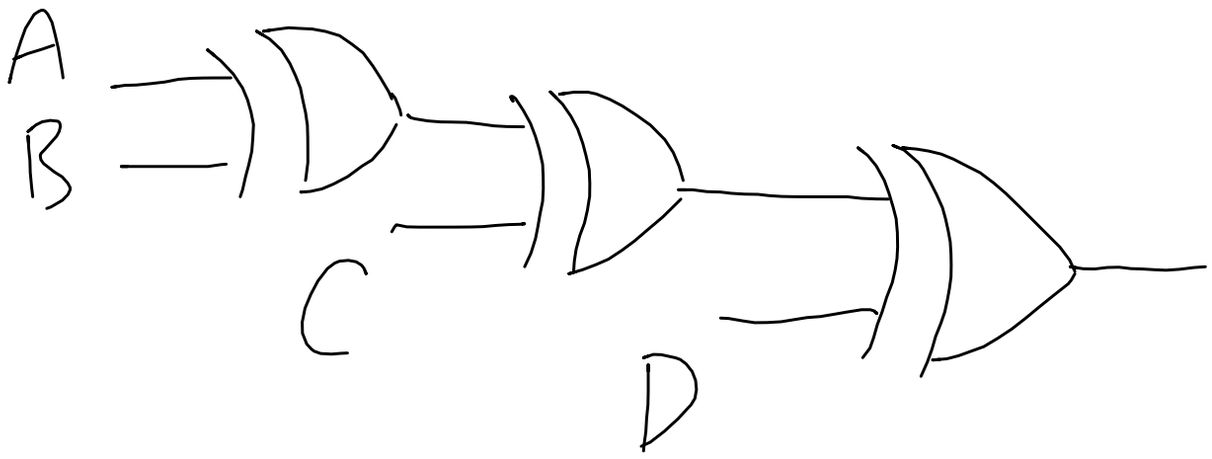
Sometimes you see the result best in a truth table



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

It's an XOR





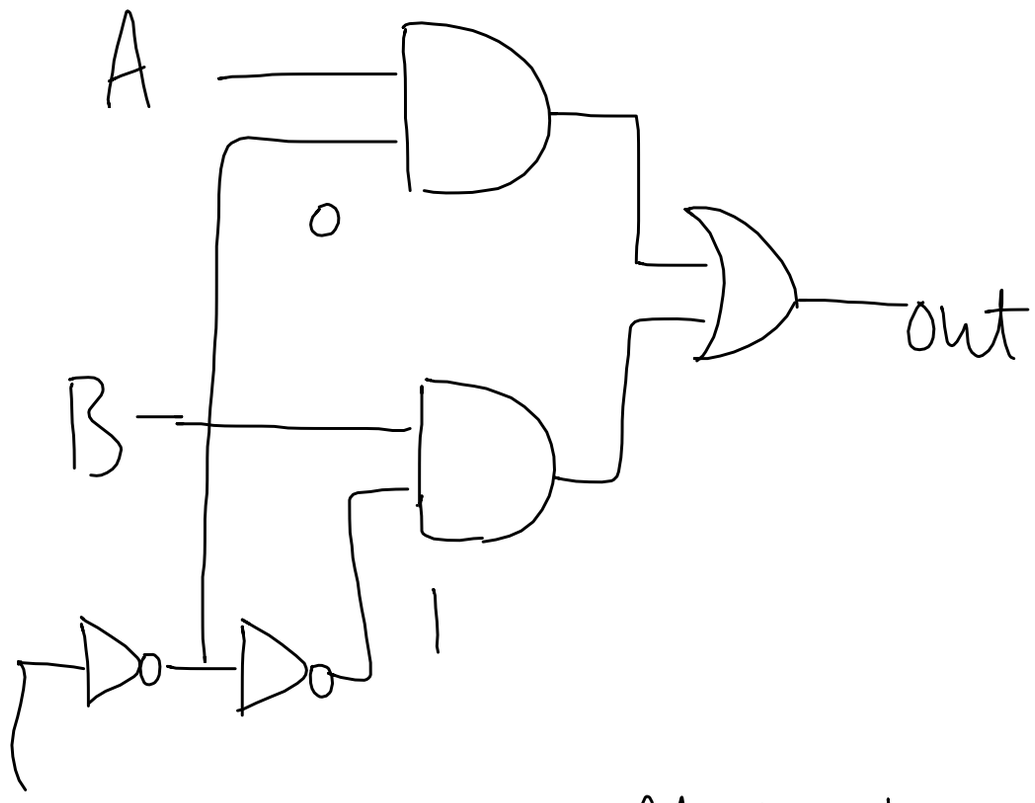
Parity

0 on even # of 1's $\Rightarrow$ 0

odd # " " $\Rightarrow$ 1

A	B	C	D	Out
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1

More moving bits around....



If seiect = 0,

out = A

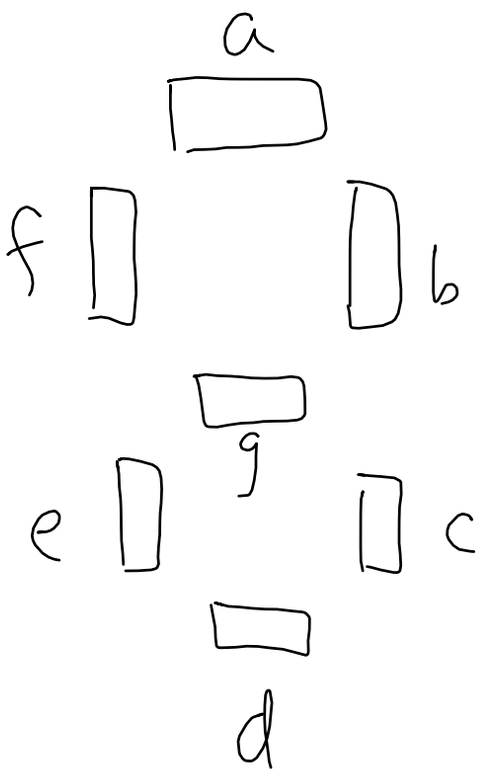
If select = 1,

out = B

Multplexer

(+ decoders)

Example ⇒ 7-segment LED



Write a "2"

Turn on a, b, d, e, g

10 possible "numbers"

Needs 4 bits of info

0010 ⇒ "2"