Section 2-1 and 2-2 Basic Binary Operations

Thursday, January 21, 2021 4:39 PM

Our math has to all be done with 1's and 0's. An Irish guy name Boole invented a way using logic.

e.g., Boolean Logic.

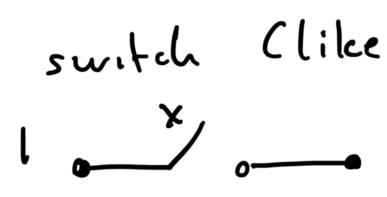
We'll use variables like X, Y etc to represent inputs and outputs (of our logic circuits)

Each variable can take on values of "1" or "0"

Basic Operations

AND, OR, and NOT

Consider a switch (like a light switch)



we'll say $X=0 \rightarrow$ switch is open

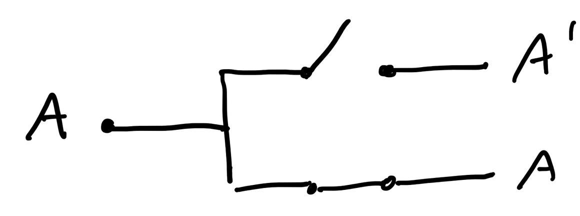
$X=1 \rightarrow$ switch is closed

If switch is closed, the electricity get through to the light (not shown) and the light is on

If switch is open, light is off

A switch can be "normally open" - think doorbell or normally closed (sensor on burglar alarm)

Some switches are arranged with two outputs, such that if one is open the other is closed



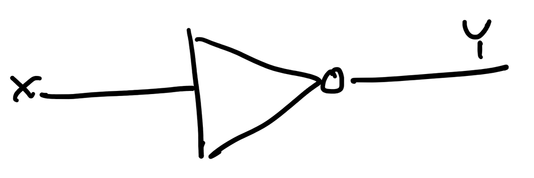
A' is the "complement" of A

or A' = NOT A

We use a "NOT" gate or inverter

"gate" is a circuit that does some logic

NOT gate:

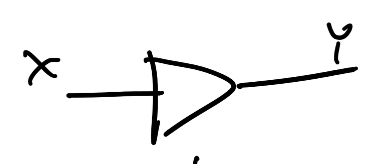


$Y = x'$ or $Y =$ NOT x

Truth table

X	Y
0	1
1	0

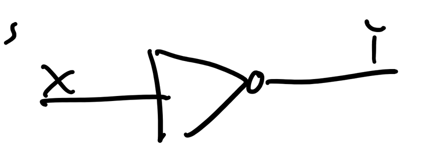
Aside: this is the circuit symbol for a buffer



$Y = x$

Doesn't perform any logic - used to amplify current or something.

But this

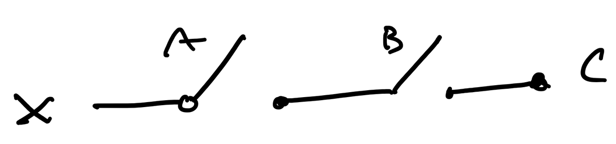


is a NOT gate

the bubble means "NOT"

That's the simplest logic gate.

Suppose you had



input is x → let it be 1

output is C

There are four possible conditions: A is open, B open

A open, B closed

A closed, B open

A closed, B closed

A=0 means switch is open

Truth Table

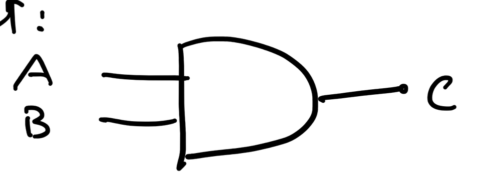
A	B	C
0	0	
0	1	
1	0	
1	1	

∴ C is only "true" (equal to 1) if both A and B are 1

$C = A \cdot B$

↑ "AND" - both have to be true to get a true output

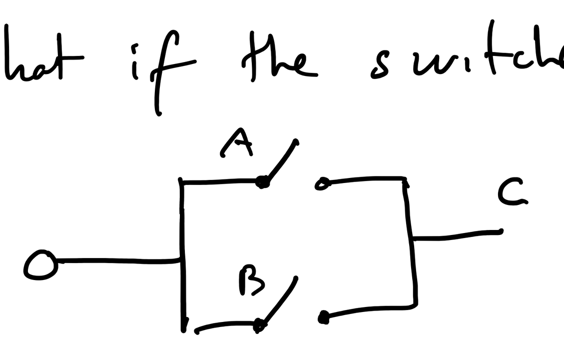
Logic gate for that:



The dot mean AND but we usually don't draw it

Now, what if the switches are in parallel:

input = 1



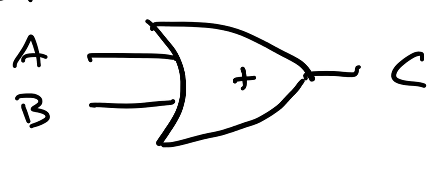
Now, if either switch is closed, C=1

A	B	C

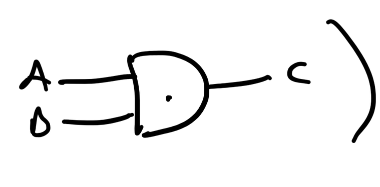
This function is called "OR"

if A OR B is closed (or both) the output is 1

OR gate:



(compare to AND



)

are "+" and "·" the same as "add" or "multiply"?

OR

0	0	0
0	1	
1	0	
1	1	

0+0=0 ✓

AND

0	0	
0	1	
		0
		1

In lab, one of the early labs has you use a "NOR" gate → this is "NOT OR" or complement of "OR"

X	Y	OR	NOR
0	0	0	
0	1	1	
1	0	1	
1	1	1	

What should the NOR gate look like? Think OR + complement

There is also a NAND gate. NOT AND

X	Y	X AND Y	NAND

These variables X, Y, A, B, C etc are Boolean variables

NOT, OR, AND, etc are Boolean operators

We'll do more with them in the next section.