lecture 3

Combinational logic 1

- truth tables

- Boolean algebra

- sum of products and product-of-sums

- logic gates

Let A, B be binary variables

("boolean")

$1 \equiv$ true, $0 \equiv$ false

<u>Notation</u>:

$$A \bullet B \equiv A \text{ and } B$$

$$A + B \equiv A \text{ or } B$$

$$\overline{A} \equiv \text{not } A$$

One uses $+, \bullet$ instead of $\vee, \wedge$.

which you may have seen elsewhere.

Truth Tables

Notation:

$A \cdot B \equiv$ A and B

$A + B \equiv$ A or B

$\overline{A} \equiv$ not A

A	B	$A \cdot B$	$A + B$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

A	$\overline{A}$
0	1
1	0

(exclusive or)

NAND NOR XOR

A	B	$\overline{A \cdot B}$	$\overline{A + B}$	$A \oplus B$
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	0	0	0

There are 2^4 = 16 possible boolean functions.

$$f : \{0,1\} \times \{0,1\} \rightarrow \{0,1\}$$

A	B	Y_1	Y_2	Y_3	...	Y_{16}
0	0					
0	1					
1	0					
1	1					

We typically only work with AND, OR, NAND, NOR, XOR.

Laws of Boolean Algebra

identity	$A + 0 = A$	$A \cdot 1 = A$
inverse	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
one and zero	$A + 1 = 1$	$A \cdot 0 = 0$
commutative	$A + B = B + A$	$A \cdot B = B \cdot A$
associative	$(A + B) + C = A + (B + C)$	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
distributive	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$
de Morgan	$(\overline{A + B}) = \overline{A} \cdot \overline{B}$	$\overline{A \cdot B} = \overline{A} + \overline{B}$

Laws of Boolean Algebra

distributive $A + (B \cdot C) = (A + B) \cdot (A + C)$

Note this one behaves differently from integers or reals.

Example

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

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

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

A	B	C	$A \cdot B \cdot C$	$\overline{A \cdot B \cdot C}$	$A \cdot B$	$A \cdot C$	$A \cdot B + A \cdot C$	Y
0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0
0	1	0	0	1	0	0	0	0
0	1	1	0	1	0	0	0	0
1	0	0	0	1	0	0	0	0
1	0	1	0	1	0	1	1	1
1	1	0	0	1	1	0	1	1
1	1	1	1	0	1	1	1	0

Sum of Products

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

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

Q: For 3 variables A, B, C, how many terms can we have in a sum of products representation ?

A: 2^3 = 8 i.e. previous slide

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

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

$$= \left(\overline{A \cdot \overline{B} \cdot C}\right) \cdot \left(\overline{A \cdot B \cdot \overline{C}}\right)$$

$$= \underbrace{\left(\overline{A} + B + \overline{C}\right) \cdot \left(\overline{A} + \overline{B} + C\right)}$$

called a "product of sums"

How to write Y as a "product of sums" ?

First, write its complement $\overline{Y}$ as a sum of products.

Because of time constraints, I decided to skip this example in the lecture.

You should go over it on your own.

A	B	C	Y	$\overline{Y}$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

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

Then write $Y = \overline{\overline{Y}}$ and apply de Morgan's Law.

$$\overline{Y} = \bar{A}\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot\bar{B}\cdot C + \bar{A}\cdot B\cdot\bar{C} + \bar{A}\cdot B\cdot C + A\cdot\bar{B}\cdot\bar{C} + A\cdot B\cdot C$$

$$\overline{\overline{Y}} = \overline{\left(\bar{A}\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot\bar{B}\cdot C + \bar{A}\cdot B\cdot\bar{C} + \bar{A}\cdot B\cdot C + A\cdot\bar{B}\cdot\bar{C} + A\cdot B\cdot C\right)}$$

$$= \overline{\left(\bar{A}\cdot\bar{B}\cdot\bar{C}\right)}\cdot\overline{\left(\bar{A}\cdot\bar{B}\cdot C\right)}\cdot\overline{\left(\bar{A}\cdot B\cdot\bar{C}\right)}\cdot\overline{\left(\bar{A}\cdot B\cdot C\right)}\cdot\overline{\left(A\cdot\bar{B}\cdot\bar{C}\right)}\cdot\overline{\left(A\cdot B\cdot C\right)}$$

$$= (A+B+C)\cdot(A+B+\bar{C})\cdot(A+\bar{B}+C)\cdot(A+\bar{B}+\bar{C})\cdot(\bar{A}+B+C)\cdot(\bar{A}+\bar{B}+\bar{C})$$

Sometimes we have expressions where various combinations of input variables give the same output. In the example below, if A is false then any combination of B and C will give the same output (namely true).

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

A	B	C	$A \cdot \overline{B} \cdot C$	$\overline{A}$	Y
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	1	0	1
1	1	0	0	0	0
1	1	1	0	0	0

Don't Care

We can simplify the truth table in such situations.

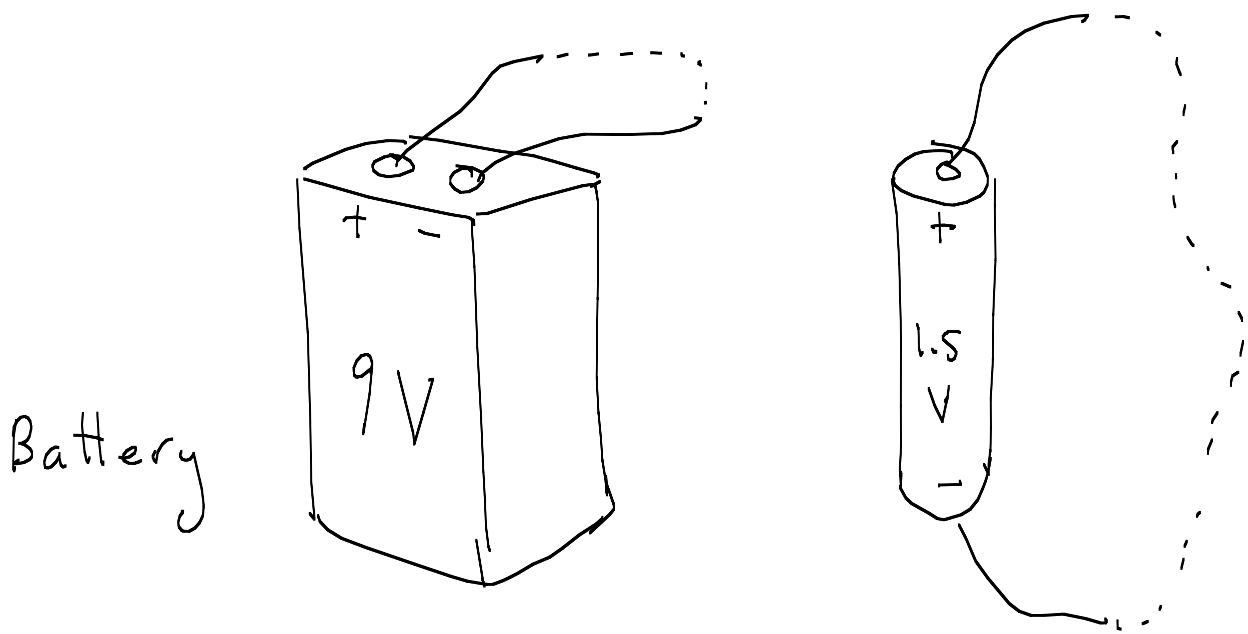
$$Y = A \cdot \overline{B} \cdot C + \overline{A}$$

A	B	C	Y
0	X	X	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

X means we "don't care" what values are there.

What are the 0's and 1's in a computer?

A wire can have a voltage difference between two terminals, which drives current.

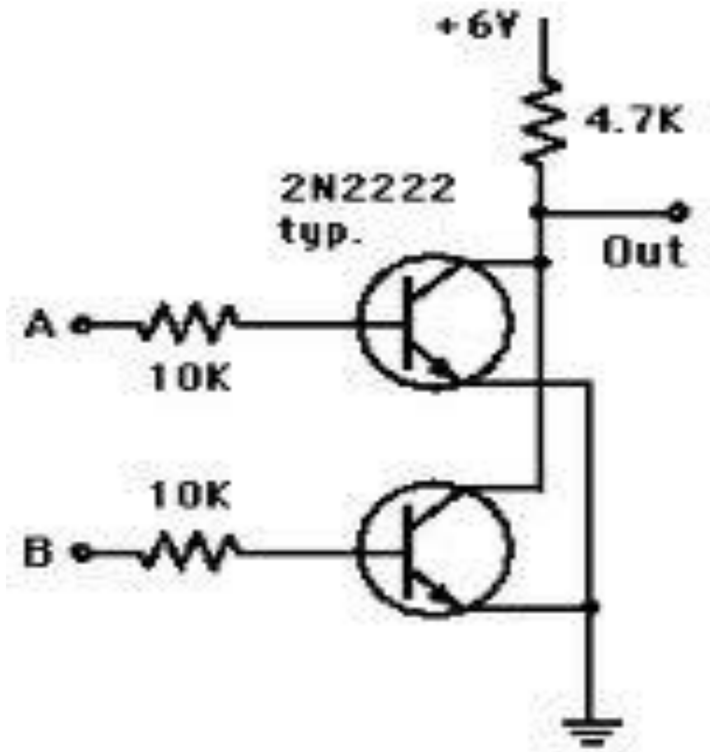


In a computer, wires can have two voltages:
high (1, current ON) or low (0, current ~OFF)

Using circult elements called "transistors" and "resistors", one can built circuits called "gates" that compute logical operations.

A

B



$\overline{A + B}$

(NOR)

For each of the OR, AND, NAND, XOR gates, you would have a different circuit.

Moore's Law (Gordon Moore was founder of Intel)

The number of transisters per mm^2 approximately doubles every two years. (1965)

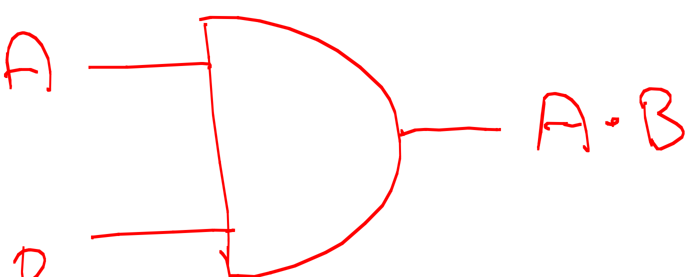
It is an observation, not a physical law.

It still holds true today, although people think that this cannot continue, because of limits on the size of atom and laws of quantum physics.

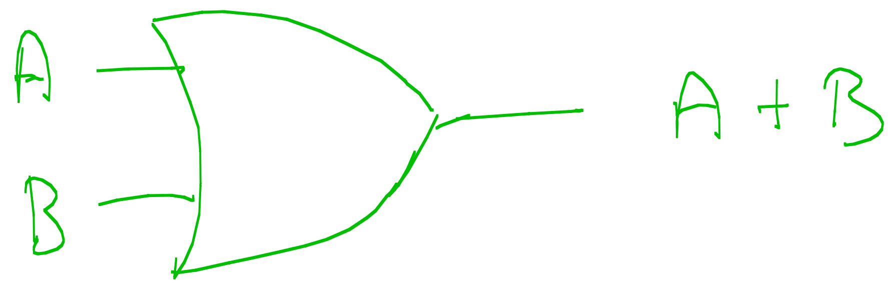
http://phys.org/news/2015-07-law-years.html

Logic Gates

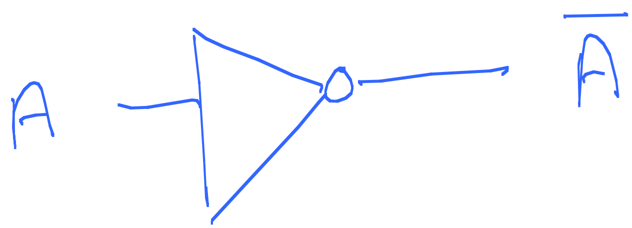
AND



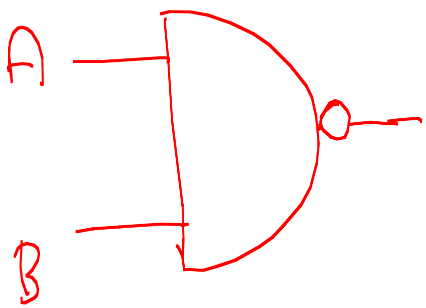
OR



NOT

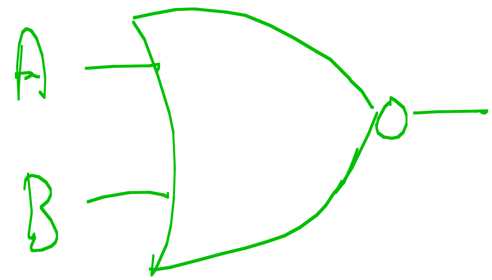


NAND



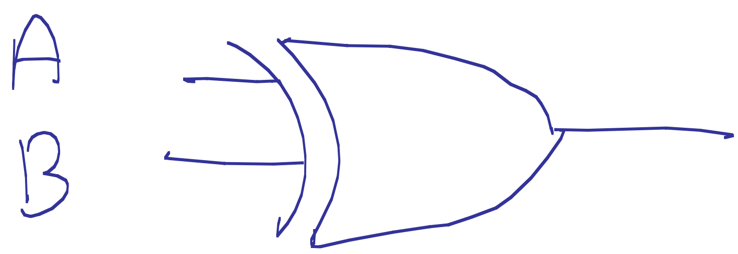
$\overline{A \cdot B}$

NOR



$\overline{A + B}$

XOR

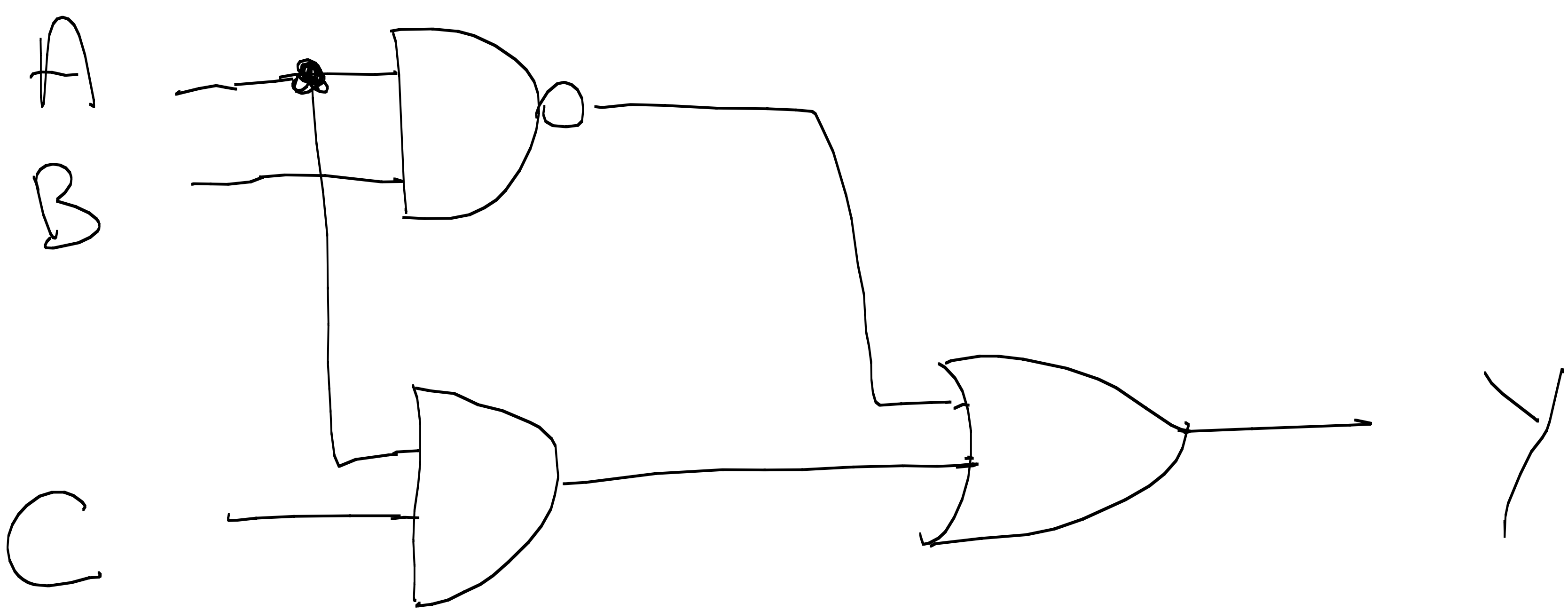


$A \oplus B$

Logic Circuit

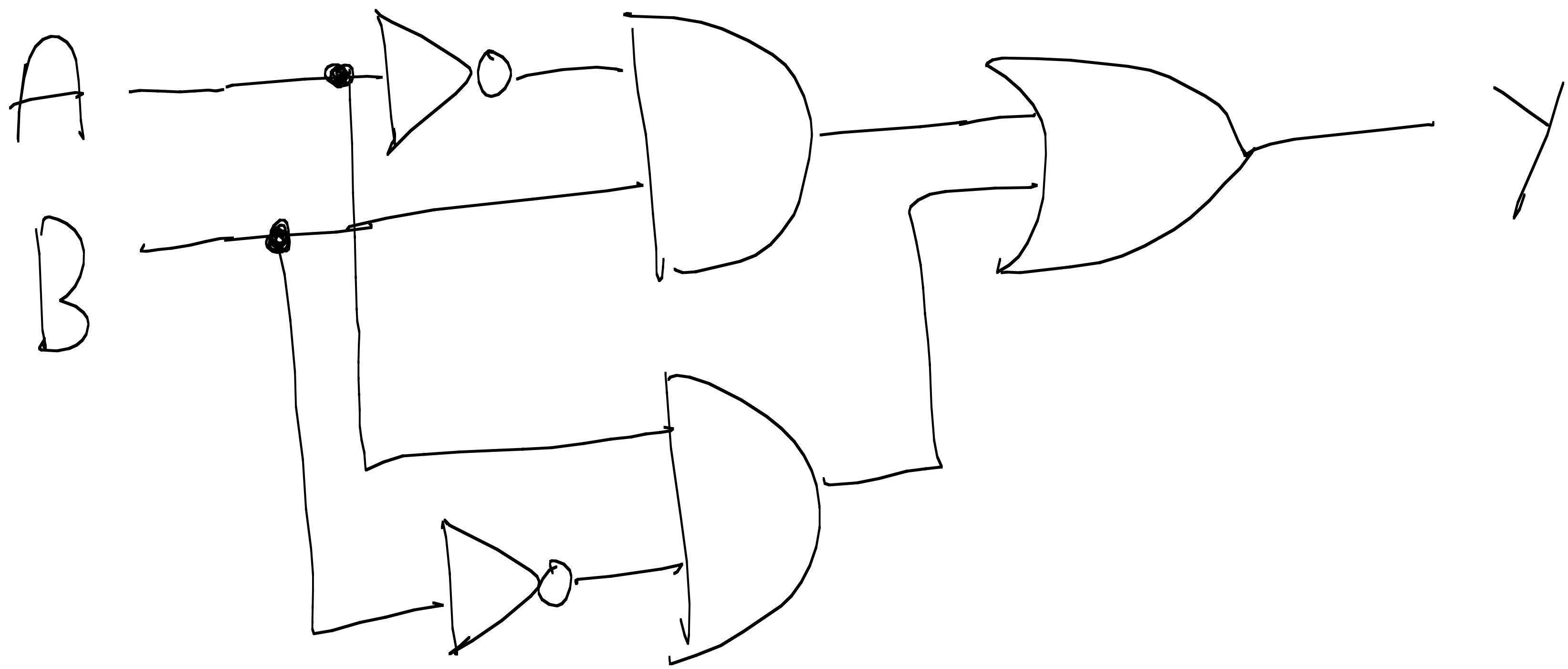
Example:

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



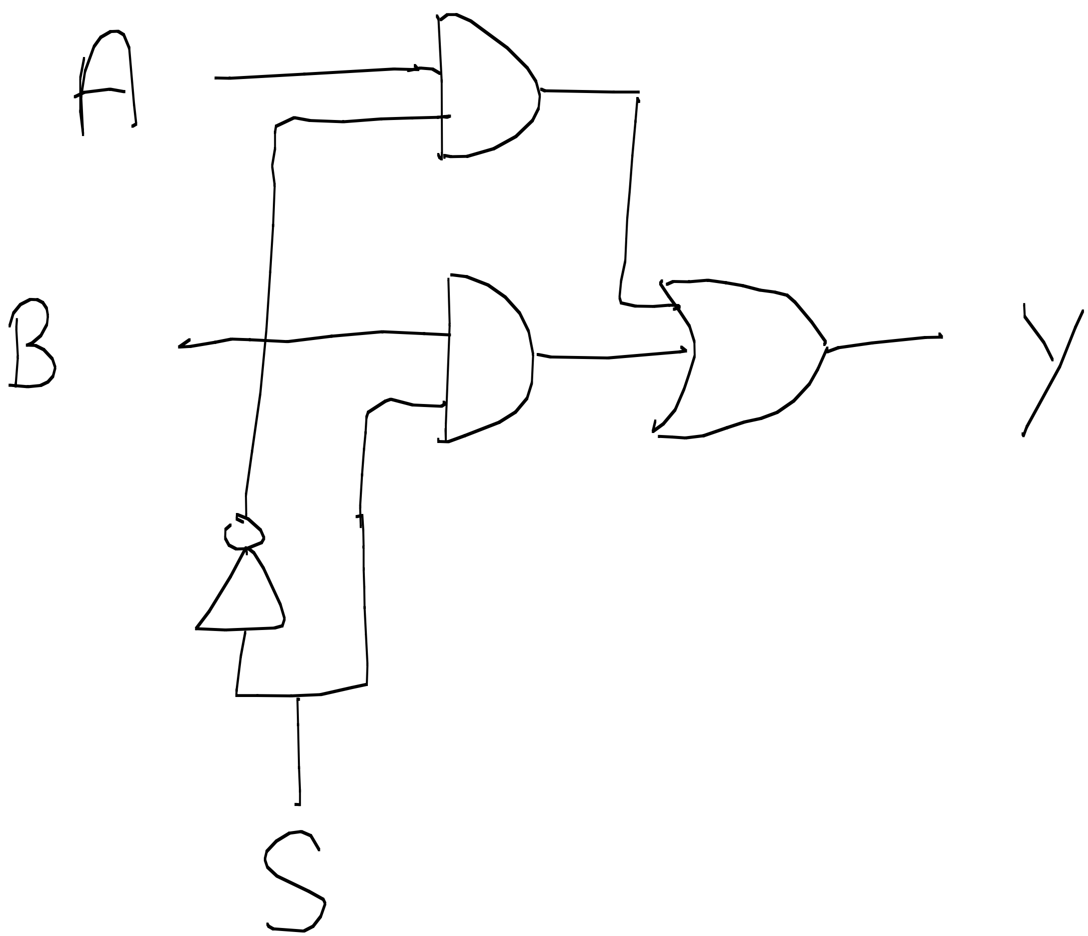
Example: XOR without using an XOR gate

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



Multiplexor (selector)

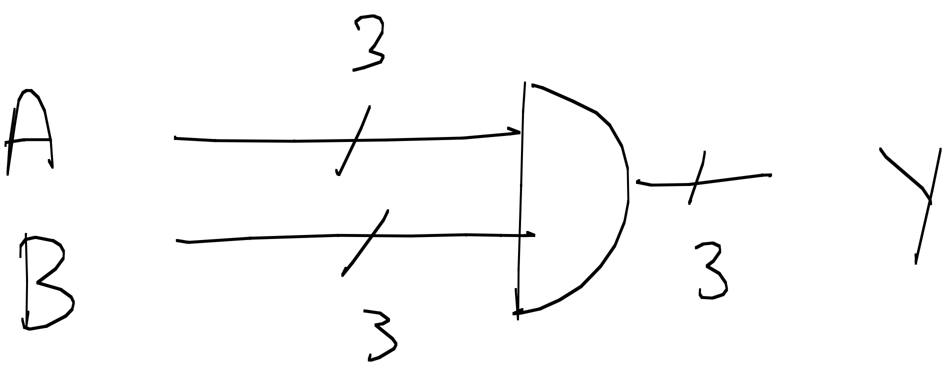
$$Y = \overline{S} \cdot A + S \cdot B$$

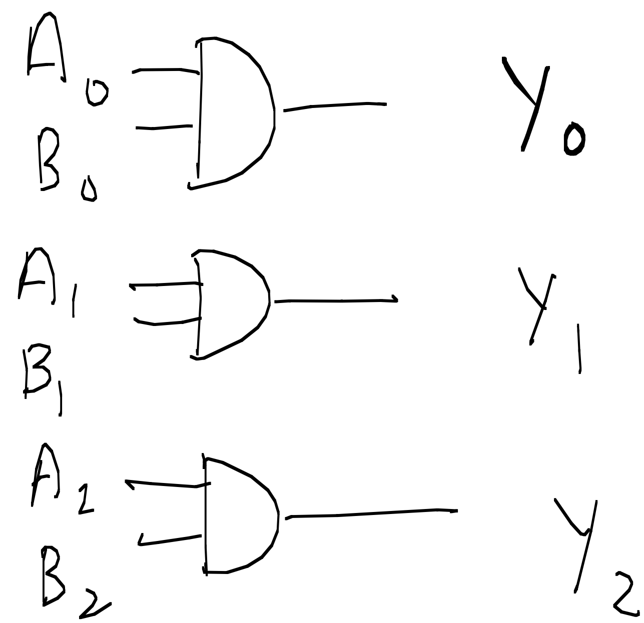


if S
  Y = B
else
  Y = A

Notation

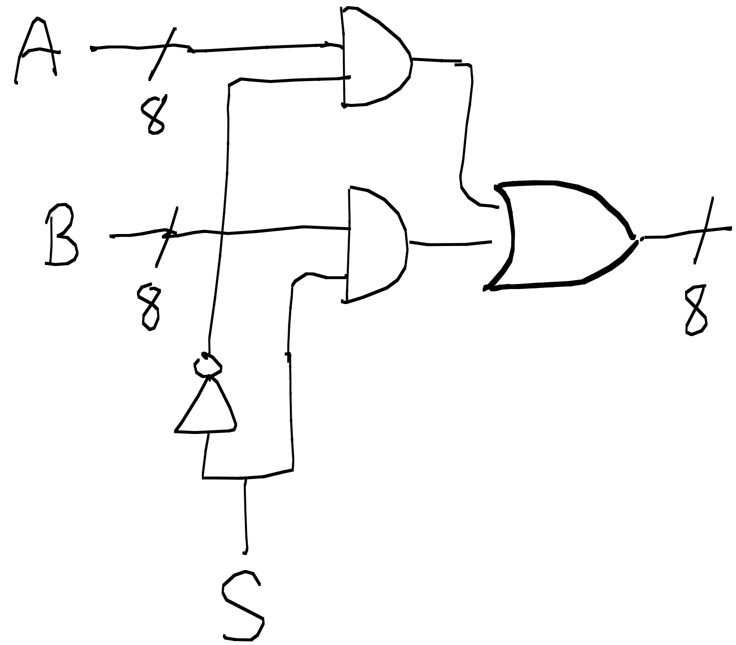
Suppose A and B are each 3 bits ($A_2 A_1 A_0$, $B_2 B_1 B_0$)





Suppose A and B are each 8 bits ($A_7 A_6 \ldots A_0$, $B_7 B_6 \ldots B_0$)
We can define an 8 bit multiplexor (selector).

Notation:



In fact we would build this from 8 separate one-bit multiplexors.

Note that the selector S is a single bit. We are selecting either all the A bits or all the B bits.