lecture 3

Combinational logic 1

Truth Tables

Let A, B be binary variables ("boolean")

1 ≡ true, 0 ≡ false

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

A	B	NAND $\overline{A \cdot B}$	NOR $\overline{A + B}$	(exclusive or) XOR $A \oplus B$
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	0	0	0

There are $2^4 = 16$ boolean functions

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

We typically only work with the 5 shown above.

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

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

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	

Example

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

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

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

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

For 3 variables A, B, C, we can have up to $2^3 = 8$ terms in the sum of products representation

$$Y = f(A, B, C, D, E)$$

A	B	C	D	E	Y
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	1	0	0
		etc			1
					1
					1
					0
					1
					⋮
1	1	1	1	1	0

2^5 rows

5 columns

$$y = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{E} + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot E + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D \cdot \overline{E} + \text{etc}$$

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

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

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

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

called a "product of sums"

How to write Z as a product of sums?

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

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

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

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

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

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

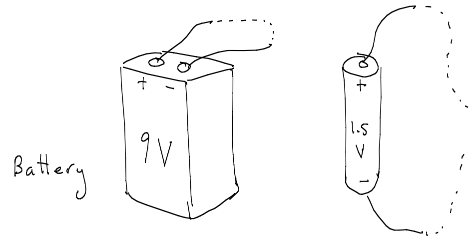
A	B	C	Y
0	X	X	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

X means "don't care"

(allows us to simplify truth table)

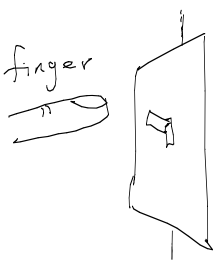
What are 0's, 1's in computer?

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

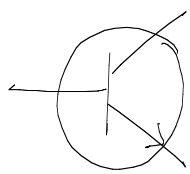


- in a computer, wires can have two voltages: HIGH ("1") or LOW ("0")

Switch (current ON or OFF)

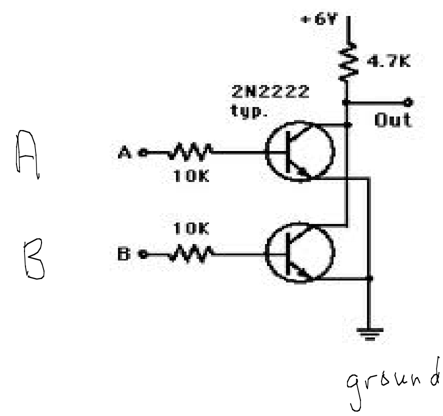


light switch on wall (mechanical control)



transistor in computer (controlled by current)

Using transistors and resistors one can build circuits that compute logical operations,



$\overline{A+B}$

(NOR)

Moore's Law

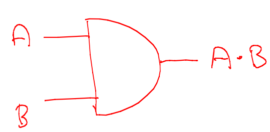
(Gordon Moore - founder of Intel)

"number of transisters per mm^2 doubles every two years" (1965)

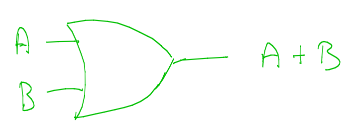
Still holds true today!

Logic "Gates"

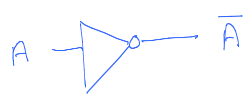
AND



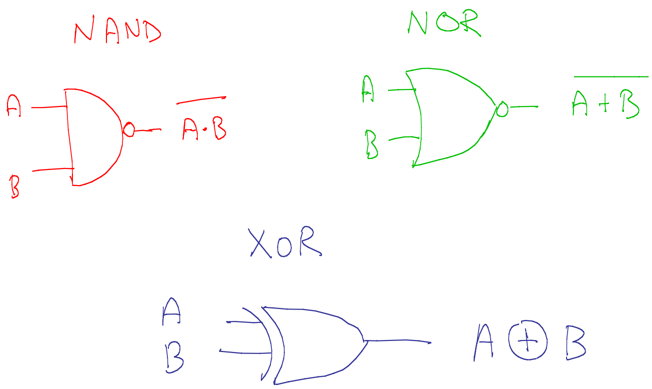
OR



NOT

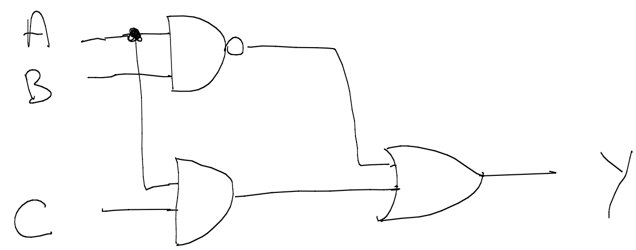


Logic Gates

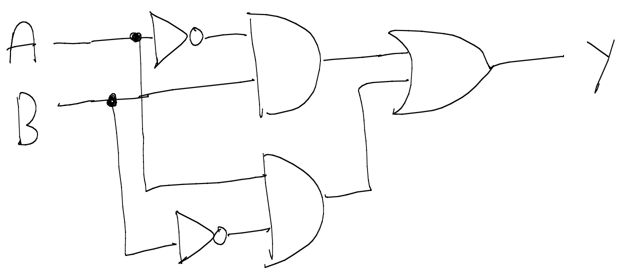


Logic circuit

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

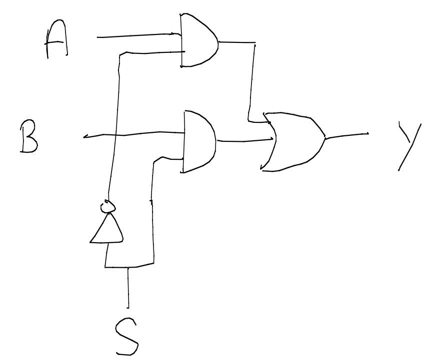


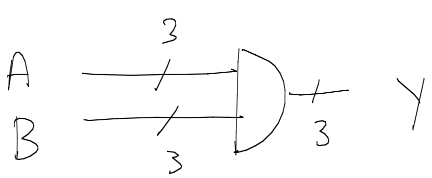
$$Y = \overline{A} \cdot B + A \cdot \overline{B} = A \oplus B$$

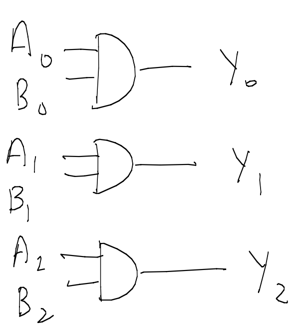


Multiplexor (selector)

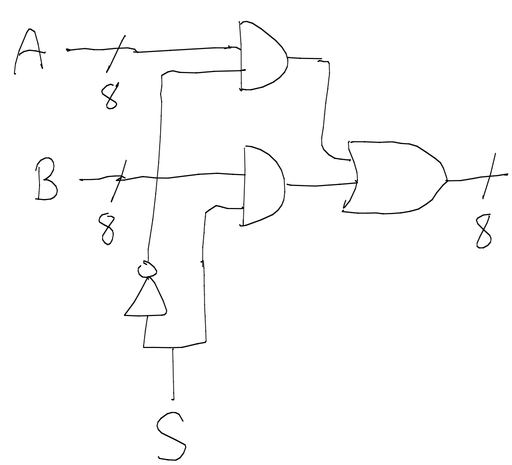
$$Y = \overline{S} \cdot A + S \cdot B$$







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

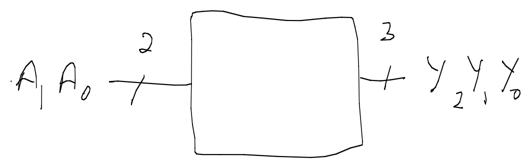


Read-Only memory

A_1	A_0	Y_2	Y_1	Y_0
0	0	0	1	1
0	1	0	0	1
1	0	0	0	0
1	1	1	0	0

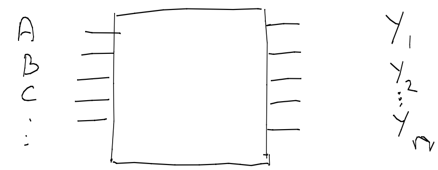
address   data

The order of variables matters.



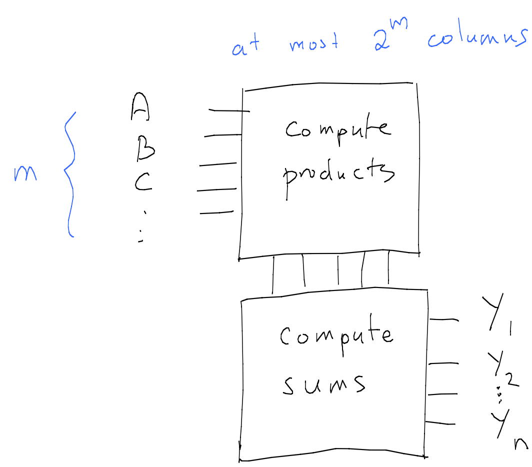
Programmable Logic Array (PLA)

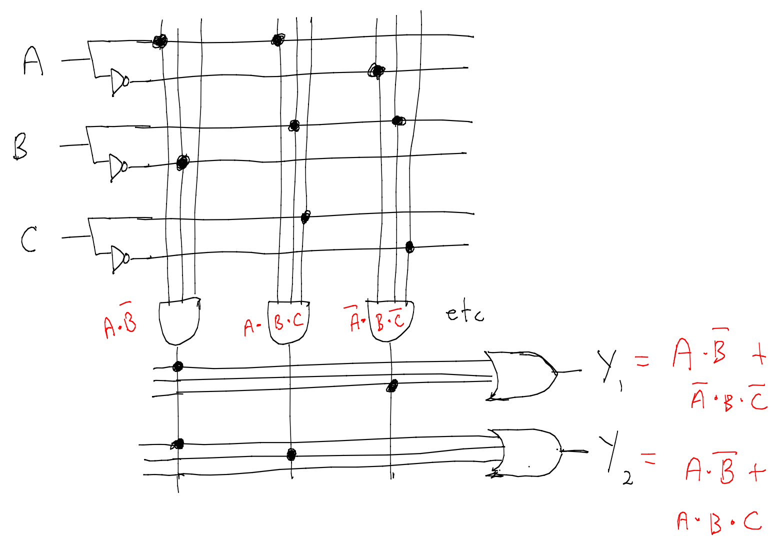
$$Y_1 = f_1(A, B, C, \ldots)$$
$$\vdots$$
$$Y_n = f_n(A, B, C, \ldots)$$



Suppose you need some circuit that computes these functions.

Write each function as a sum of products.





- start with fully connected circuit (say 8 input variables and 12 output variables) and then "cut the wires" to produce particular functions $Y_1 \ldots Y_n$

- cheaply mass produced (Texas Instruments, National Semiconductor, ...)