Circuits

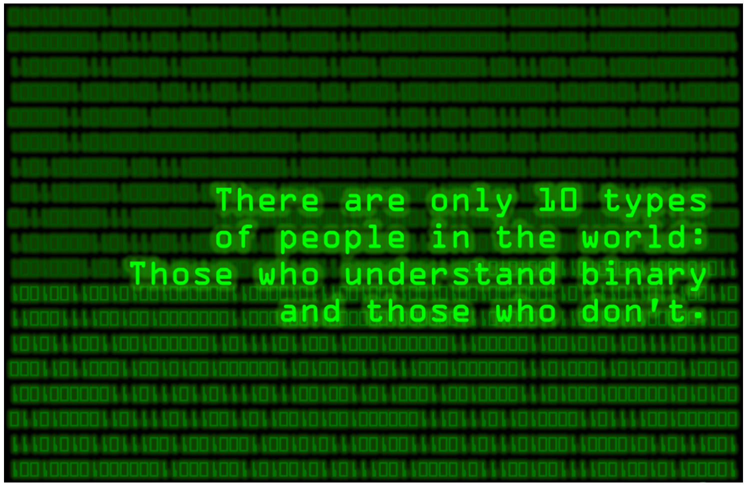
The Language of Computers

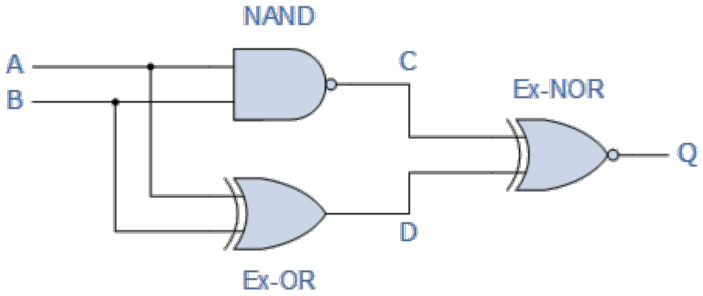
Binary

- 1's and 0's
- Binary Digit = Bit

Boolean Logic

- George Boole
- Logic for binary variables
- True/False or High/Low, 1/0, On/Off
- 1 bit does not a computer make!

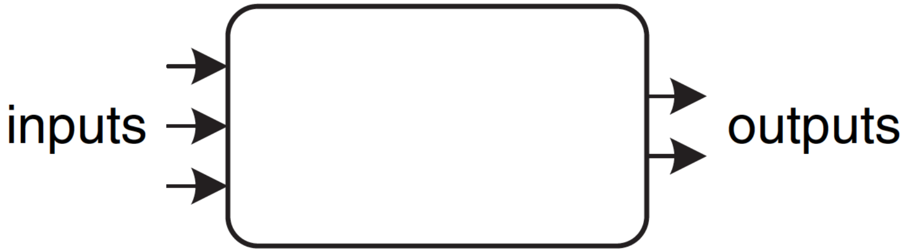




Circuit

Black-box representation

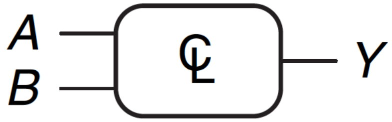
- One or more inputs
- One or more outputs
- Performs some operation



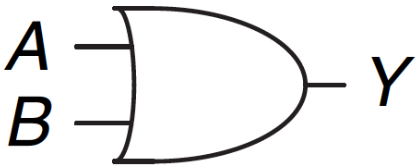
Combinational Circuit

Outputs depend only on the current value of the inputs

How could this not be the case?

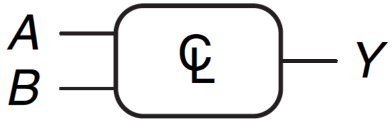


$$Y = F(A, B) = A + B$$



Combinational Circuit

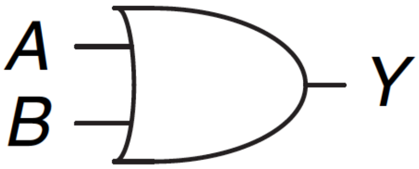
Outputs depend only on the current value of the inputs



$$Y = F(A, B) = A + B$$

How could this not be the case?

Answer: if the circuit has *memory* somewhere inside.



Combinational Circuit

We will see later what rules can be enforced to ensure a circuit is combinational

Why bother?

By restricting ourselves to combinational circuits, we make the behavior of circuits very easy to describe using **truth tables**

Since outputs depend only on inputs, simply list all input-output pairs. We are working with discrete systems, so there will be a finite number

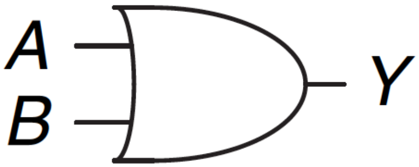
Truth Tables

Characterize the behavior of a group of inputs and outputs

- 0's and 1's
- 1 column per variable
- N variables 2^N rows

Lists all possible values of the variables

- x possible values of 0 or 1
- xy possible values of 0 0, 0 1, 1 0, 1 1
- xyz 0 0 0, 0 0 1, 0 1 0, 0 1 1, 1 0 0, 1 0 1, 1 1 0, 111



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