CS 211: Computer Architecture
Digital Logic

Topics:

- Transistors
- Logic Gates
- Boolean algebra

Transistor: Building Block of Computers

Microprocessors contain millions (billions) of transistors

- Intel Pentium 4 (2000): 48 million
- IBM PowerPC 750FX (2002): 38 million
- IBM/Apple PowerPC G5 (2003): 58 million

Logically, each transistor acts as a switch

Combined to implement logic functions

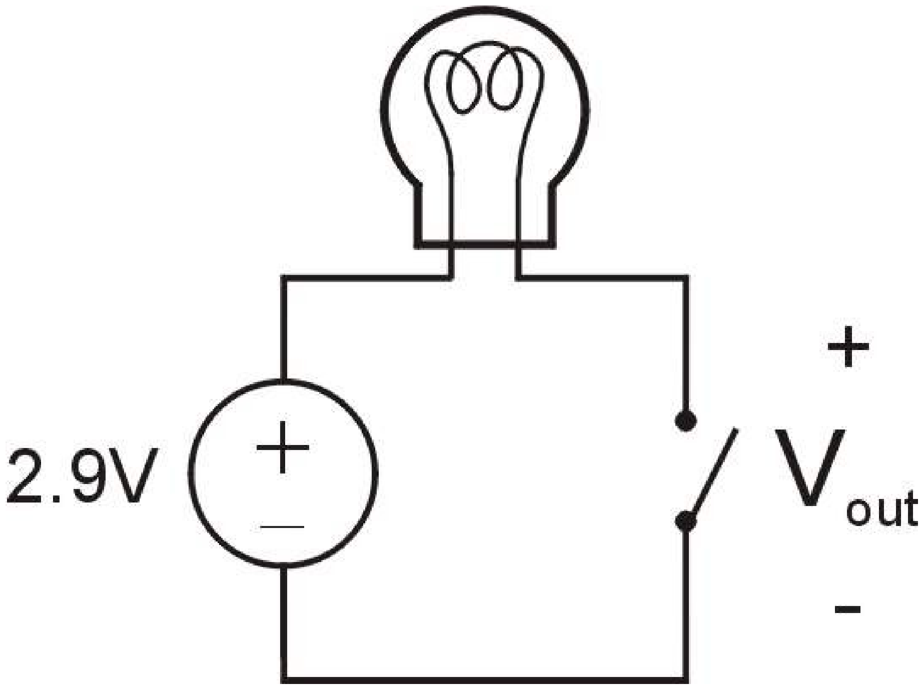
- AND, OR, NOT

Combined to build higher-level structures

- Adder, multiplexer, decoder, register, …

Combined to build processor

Simple Switch Circuit



Switch open:

- No current through circuit
- Light is off
- V_{out} is +2.9V

Switch closed:

- Current flows
- Light is on
- V_{out} is 0V

Switch-based circuits **can easily represent two states: on/off, open/closed, voltage/no voltage.**

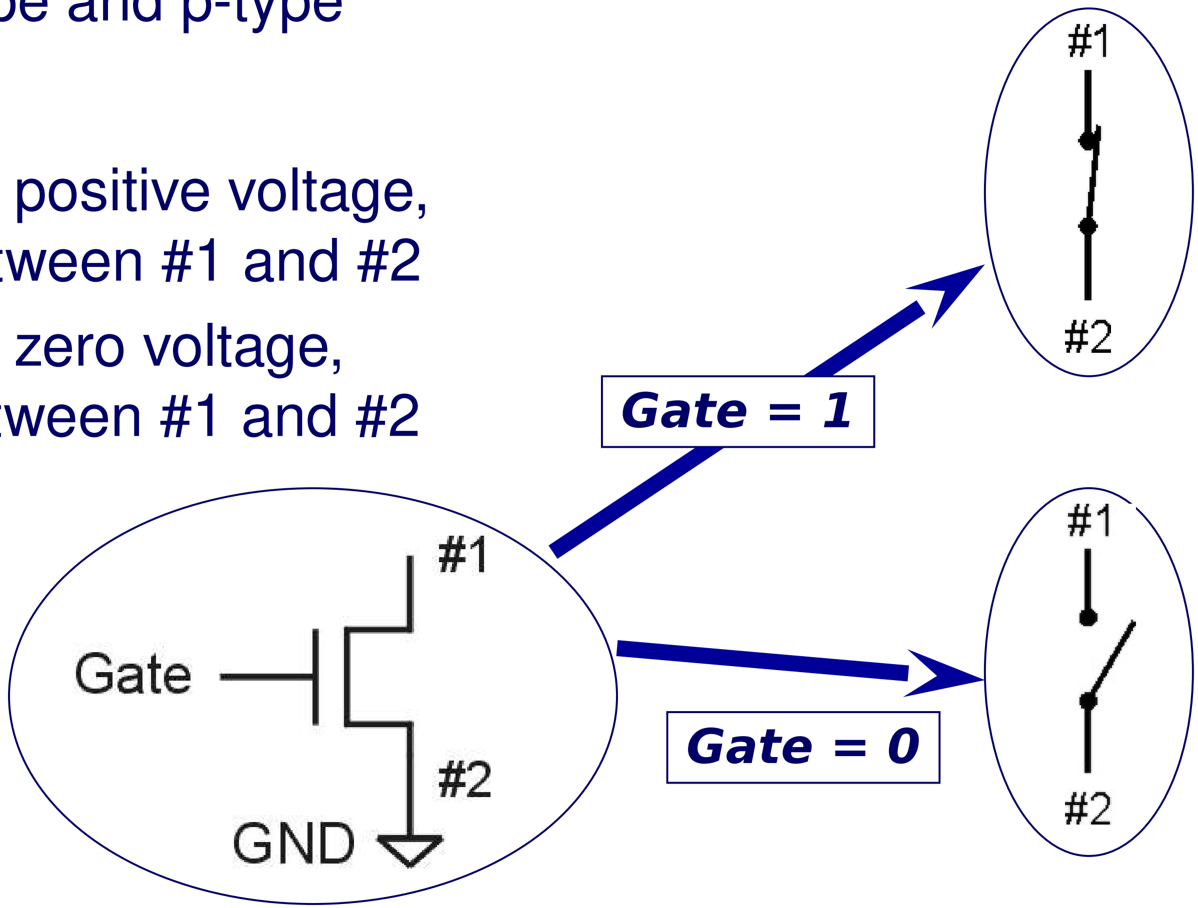
n-type MOS Transistor

MOS = Metal Oxide Semiconductor

- two types: n-type and p-type

n-type

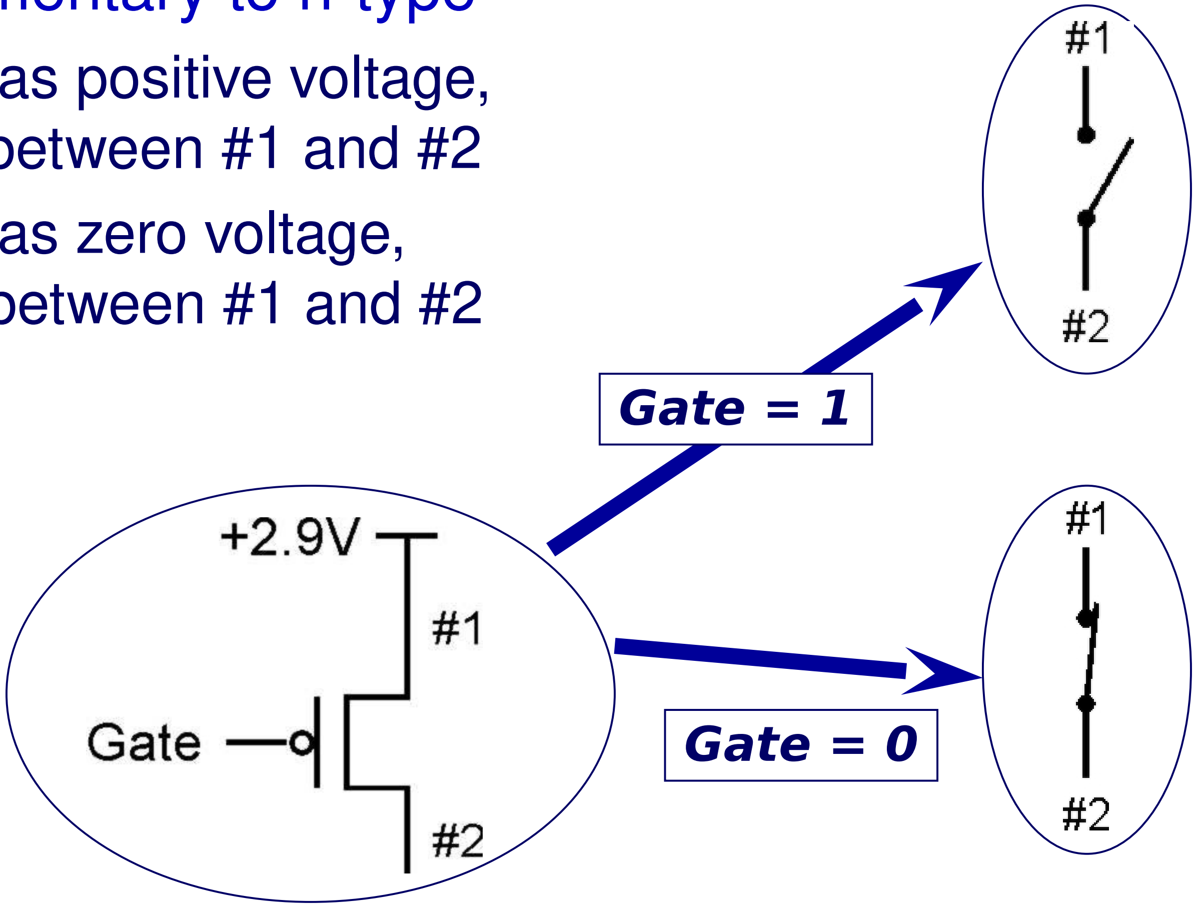
- when Gate has positive voltage, short circuit between #1 and #2
- when Gate has zero voltage, open circuit between #1 and #2



p-type MOS Transistor

p-type is complementary to n-type

- when Gate has positive voltage, open circuit between #1 and #2
- when Gate has zero voltage, short circuit between #1 and #2



Logic Gates

Use transistors to implement logical functions: AND, OR, NOT

Digital symbols:

- recall that we assign a range of analog voltages to each digital (logic) symbol

- assignment of voltage ranges depends on electrical properties of transistors being used
  - typical values for "1": +5V, +3.3V, +2.9V
  - from now on we'll use +2.9V

CMOS Circuit

Complementary MOS

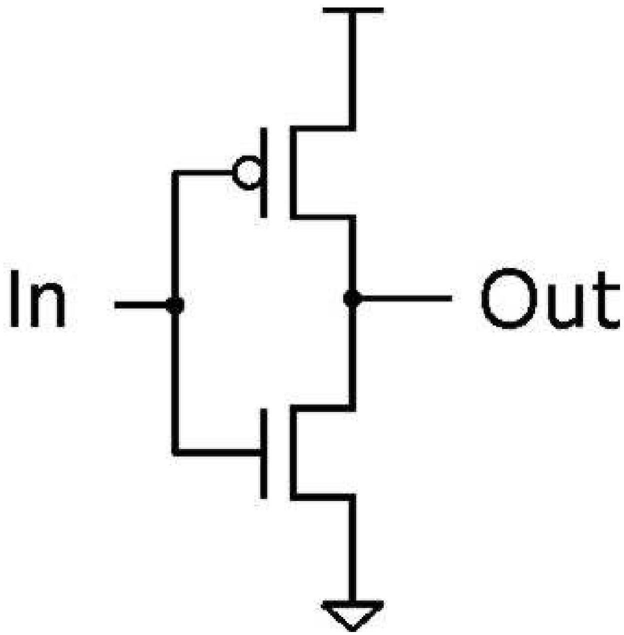
Uses both n-type and p-type MOS transistors

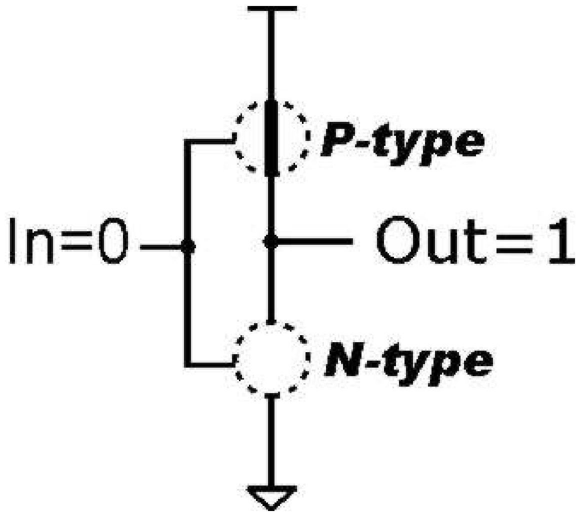
- p-type
  - Attached to + voltage
  - Pulls output voltage UP when input is zero
- n-type
  - Attached to GND
  - Pulls output voltage DOWN when input is one

MOS transistors are combined to form Logic Gates

For all inputs, make sure that output is either connected to GND or to +, but not both!

Inverter (NOT Gate)

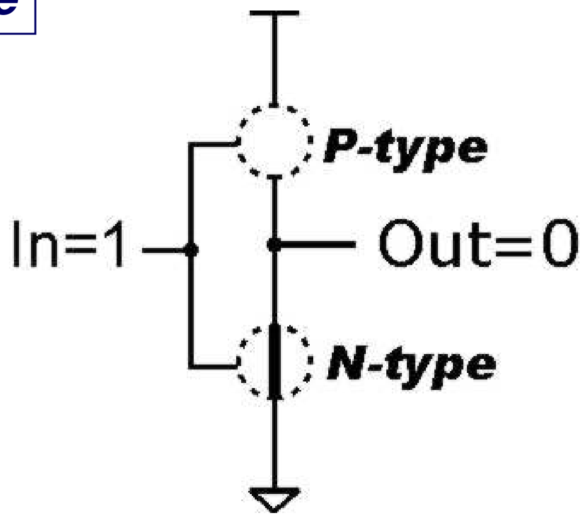




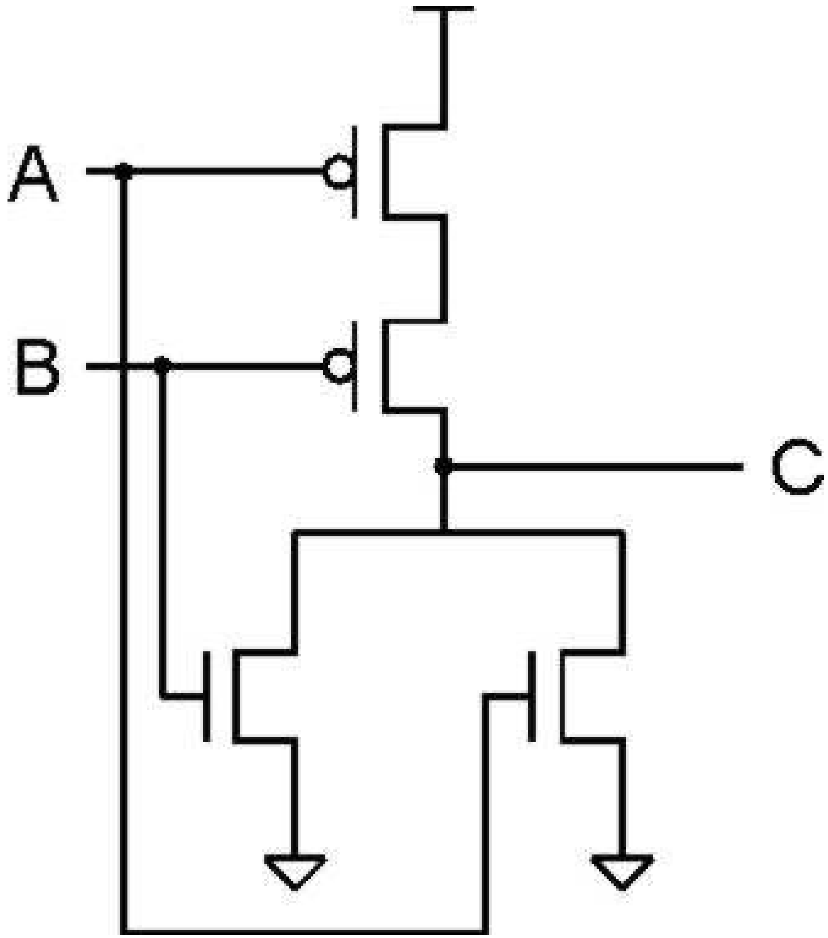
Truth table

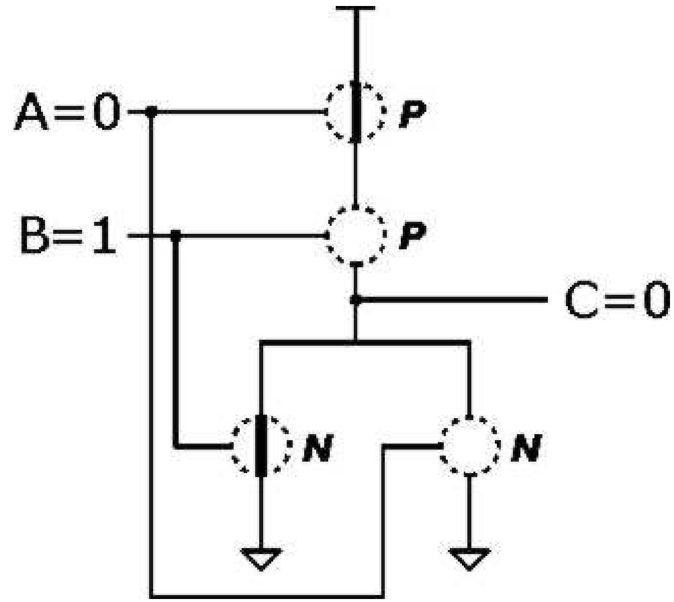
In	Out
0 V	2.9 V
2.9 V	0 V

In	Out
0	1
1	0



NOR Gate

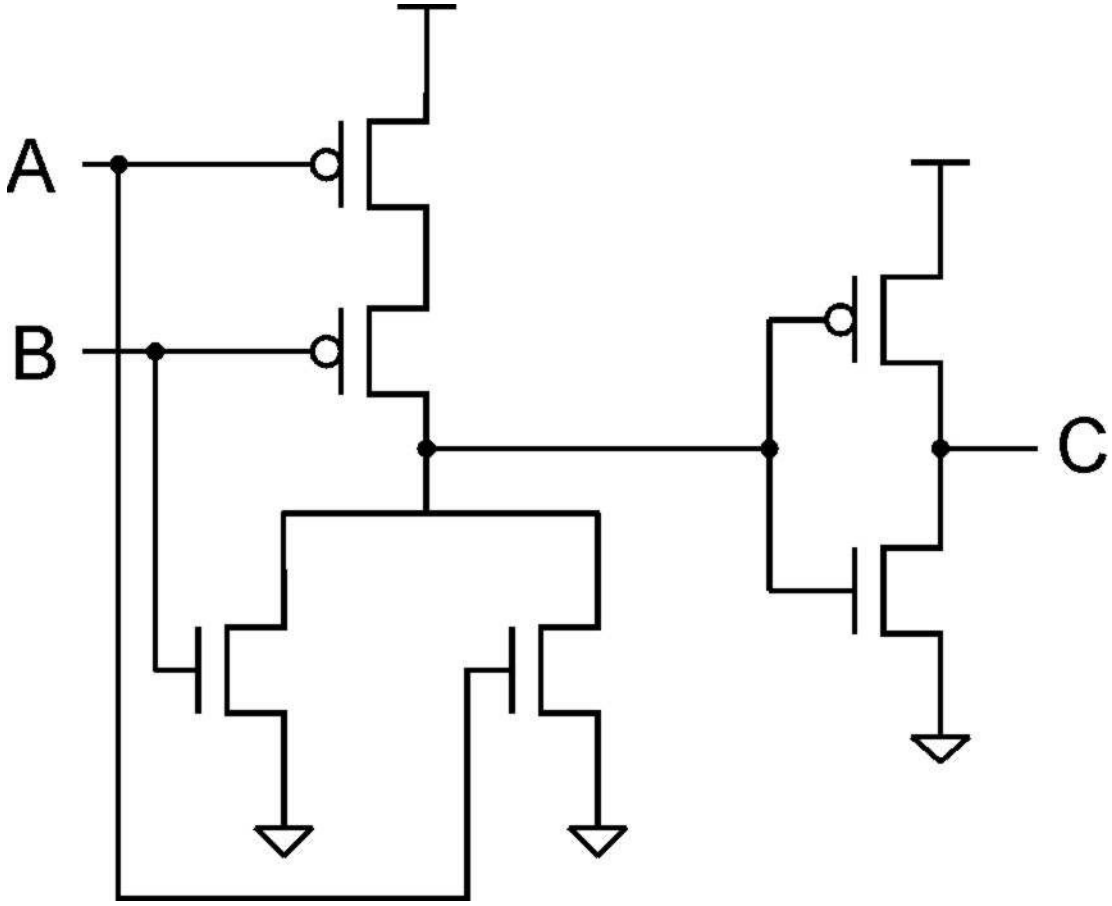




A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

Note: Serial structure on top, parallel on bottom.

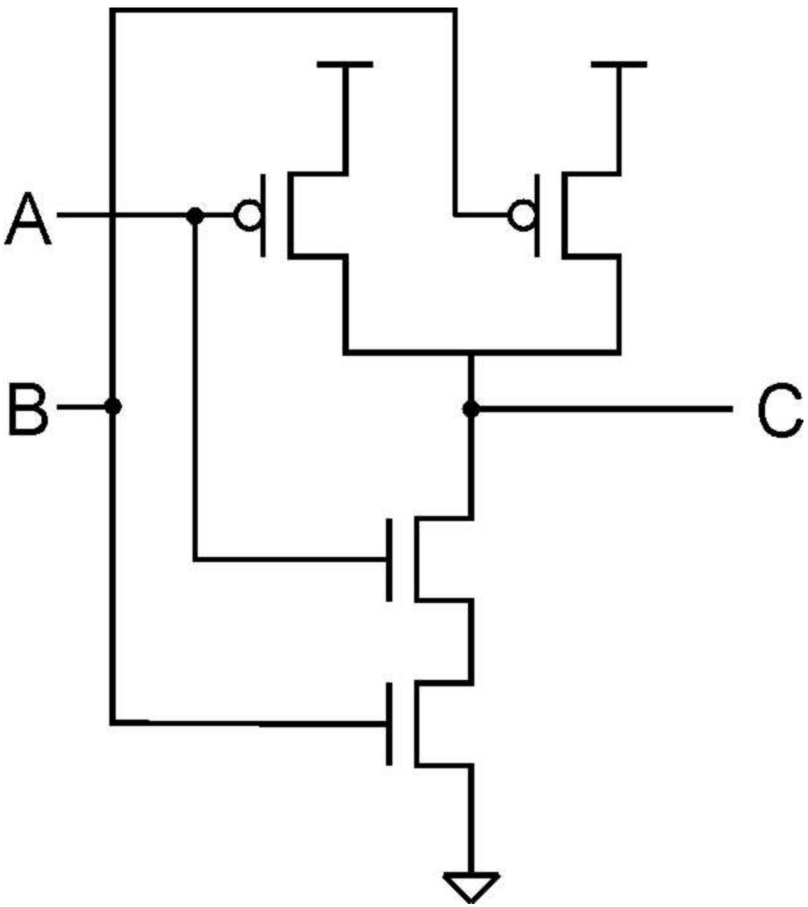
OR Gate

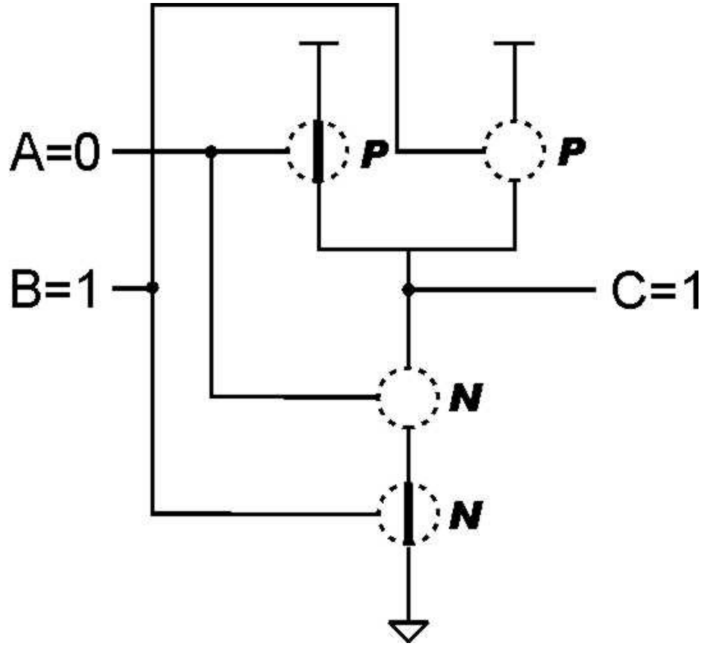


A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Add inverter to NOR.

NAND Gate (AND-NOT)

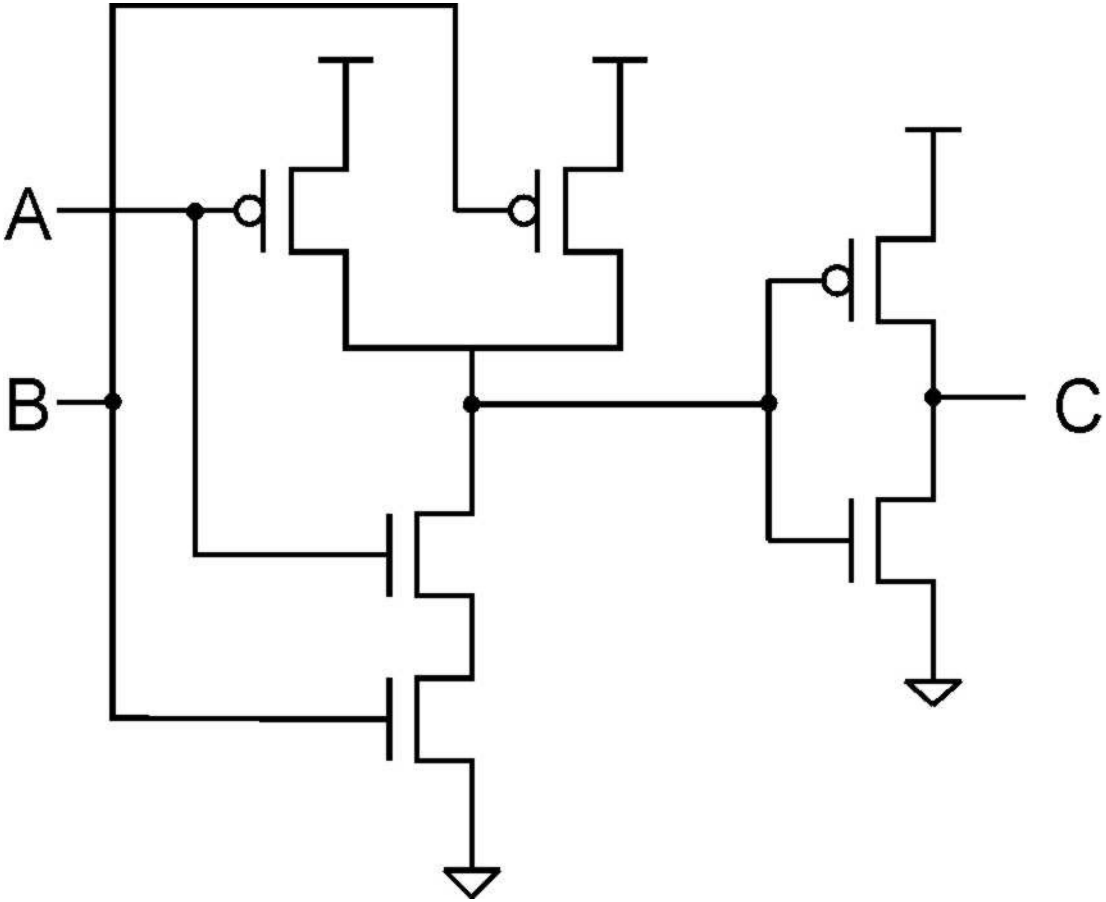




A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Note: Parallel structure on top, serial on bottom.

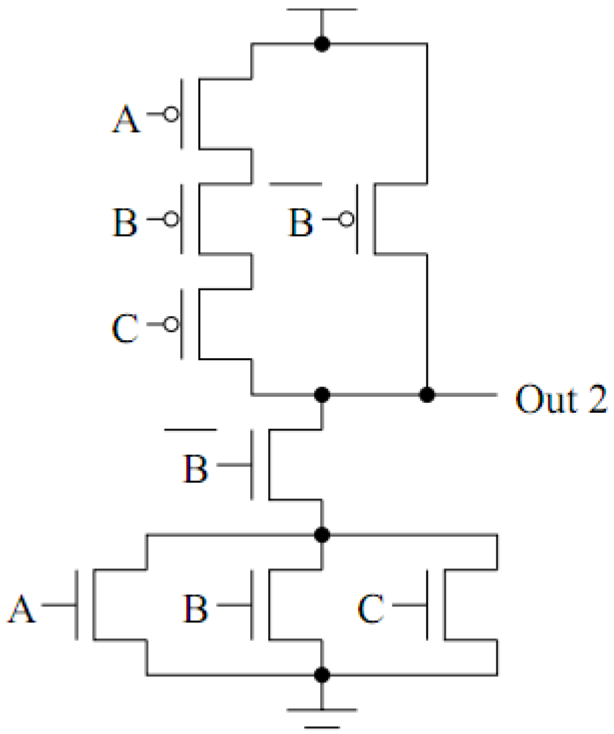
AND Gate



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

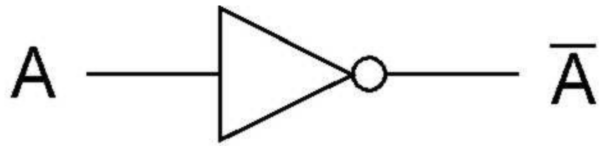
Add inverter to NAND.

What is this?

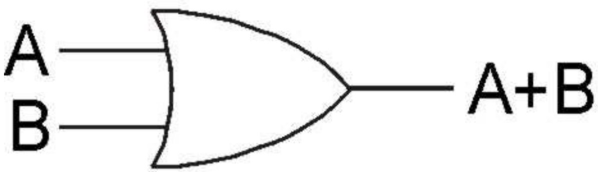


A	B	C	Out
0	0	0	**1**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**0**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

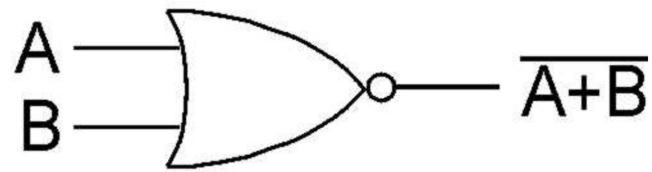
Basic Logic Gates Symbols



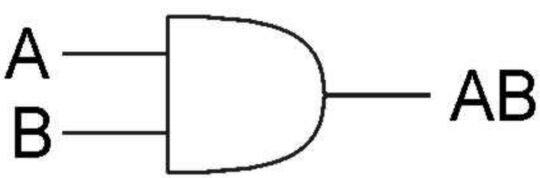
NOT



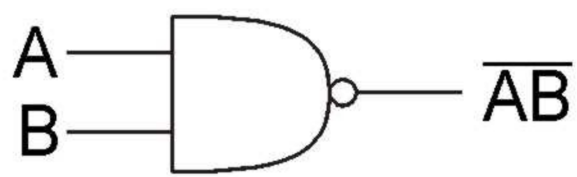
OR



NOR



AND



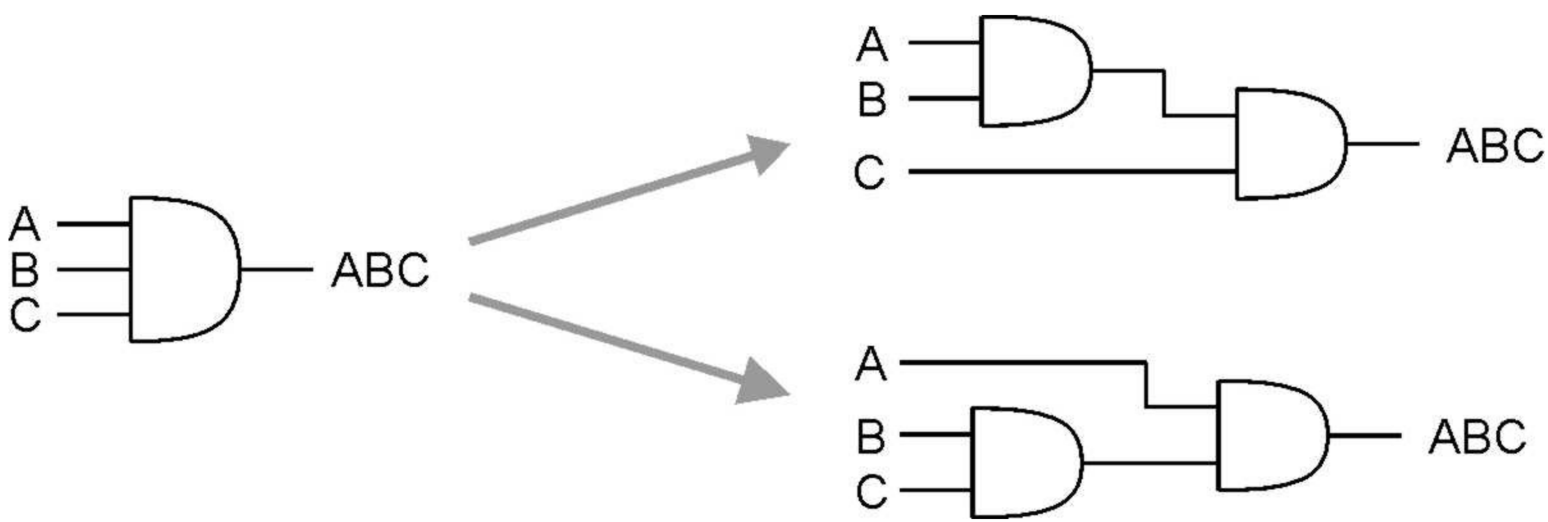
NAND

More than 2 Inputs?

AND/OR can take any number of inputs.

- AND = 1 if all inputs are 1.
- OR = 1 if any input is 1.
- Similar for NAND/NOR.

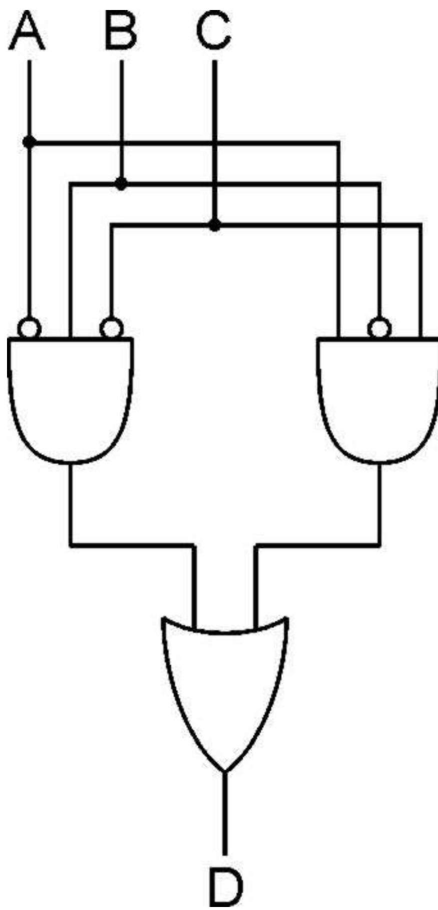
Can implement with multiple two-input gates.



Logical Completeness

Can implement ANY truth table with AND, OR, NOT.

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



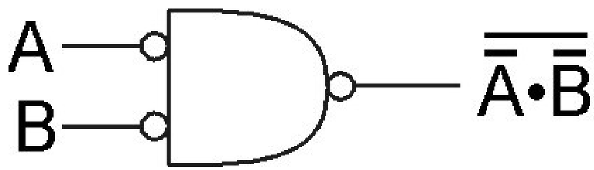
. AND combinations that yield a "1" in the truth table.

2. OR the results of the AND gates.

DeMorgan's Law

Converting AND to OR (with some help from NOT)

Consider the following gate:



To convert AND to OR (or vice versa), invert inputs and output.

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

Same as A+B!

Generally, DeMorgan's Laws:

1. $\overline{PQ} = \overline{P} + \overline{Q}$
2. $\overline{P + Q} = \overline{P}\ \overline{Q}$

NAND and NOR Functional Completeness

Any gate can be implemented using either NOR or NAND gates.

Why is this important?

- When building a chip, easier to build one with all of the same gates.

NAND, NOR universality

NAND, NOR universal because they can realize AND, OR, NOT

$\overline{A} = A \text{ NAND } A$	$\overline{A} = A \text{ NOR } A$
$AB = \overline{A \text{ NAND } B}$	$A+B = \overline{A \text{ NOR } B}$
$A+B = \overline{A} \text{ NAND } \overline{B}$	$AB = \overline{A} \text{ NOR } \overline{B}$

Terminology

Binary variable: a symbolic representation that might be 0 or 1 (eg. X, Y, A, B)

Complement: the opposite value of variable X

Literal: a boolean variable or its complement (eg, X, $\overline{X}$)

Expression: a set of literals combined with logical operations (eg. AB + C)

Boolean algebra

- Values: 0, 1
- Operations: and, or, not, xor, implies, etc.
  - X AND Y: like multiplication
  - X XOR Y: like addition (mod 2)

	0	1
0	0	0
1	0	1

	0	1
0	0	1
1	1	0

Boolean Identities

OR	AND	NOT	
$X+0 = X$	$X1 = X$		(identity)
$X+1 = 1$	$X0 = 0$		(null)
$X+X = X$	$XX = X$		(idempotent)
$X+\overline{X} = 1$	$X\overline{X} = 0$		(complementarity)
		$\overline{\overline{X}} = X$	(involution)
$X+Y = Y+X$	$XY = YX$		(commutativity)
$X+(Y+Z) = (X+Y)+Z$	$X(YZ) = (XY)Z$		(associativity)
$X(Y+Z) = XY + XZ$	$X+YZ = (X+Y)(X+Z)$		(distributive)
$\overline{X+Y} = \overline{X}\,\overline{Y}$	$\overline{XY} = \overline{X} + \overline{Y}$		(DeMorgan's theorem)

Boolean Algebra Example

$$F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$$

$\overline{X}Y(Z + \overline{Z}) + XZ$ (by reverse distribution)

$\overline{X}Y1 + XZ$ (by complementarity)

$\overline{X}Y + XZ$ (by identity)

Boolean Algebra Example 2

Find the complement of F

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

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

$$\left(\overline{A\overline{B}}\right)\left(\overline{\overline{A}B}\right) \qquad \text{(by DeMorgan's)}$$

$$\left(\overline{A} + \overline{\overline{B}}\right)\left(\overline{\overline{A}} + \overline{B}\right) \qquad \text{(by DeMorgan's)}$$

$$\left(\overline{A} + B\right)\left(A + \overline{B}\right) \qquad \text{(by involution)}$$

Using DeMorgan's Laws to Complement

1. Big bar over AND and OR of 2 or more functions
2. Replace AND with OR, OR with AND
3. 1 with 0, 0 with 1
4. F with not(F), not(F) with F

$$\overline{\overline{A\bar{B}C} + \overline{ACD} + B\bar{C}}$$

$F = \overline{A\bar{B}C},\ G = \overline{ACD},\ H = B\bar{C},\ \overline{F+G+H} = \bar{F}\,\bar{G}\,\bar{H}$

$$= (A\bar{B}C)(ACD)(\overline{B\bar{C}})$$

$(A\bar{B}C)(ACD) = A\bar{B}CD,\ F = B,\ G = \bar{C},\ \overline{FG} = \bar{F}+\bar{G}$

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

$$= A\bar{B}CD + A\bar{B}CD$$

$$= A\bar{B}CD$$

Duals

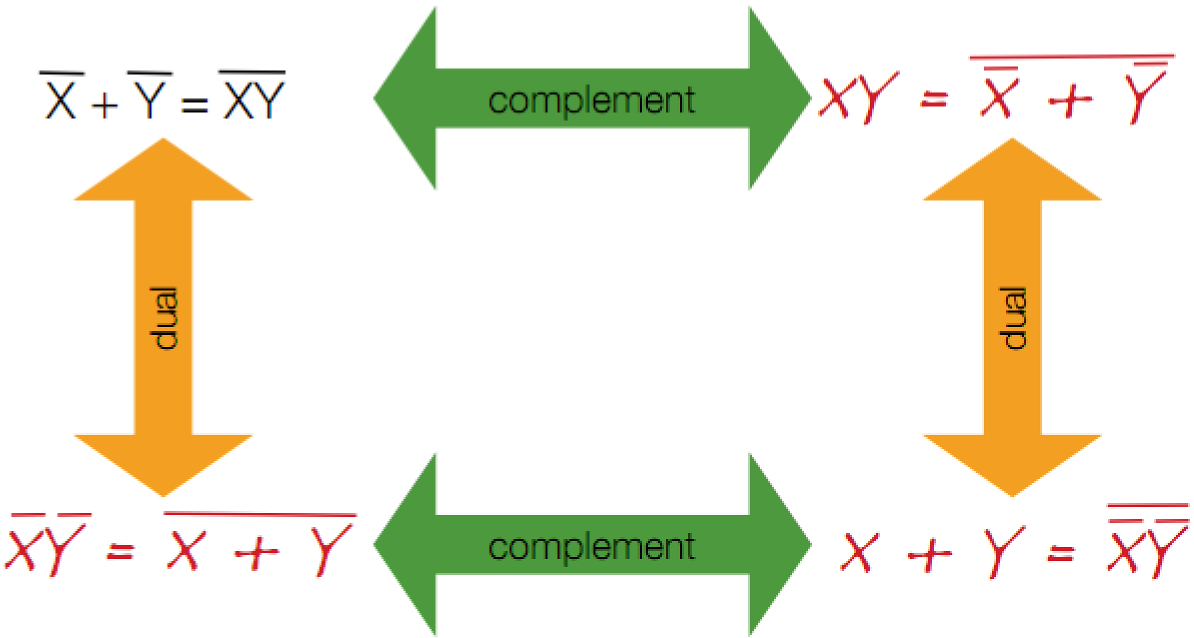
All boolean expressions have duals

Any theorem you prove, you can also prove for the dual

To form a dual

1. replace AND with OR, OR with AND
2. replace 1 with 0, 0 with 1

Complements and Duals



Complement Using Duals

Get dual and then complement each literal

$$F = X + A\,(Z + \overline{X}\,(Y + W) + \overline{Y}\,(Z + W))$$

Dual: $F_{dual} = X\,(A + Z\,(\overline{X} + YW)(\overline{Y} + ZW))$

$$\overline{F} = \overline{X}\,(\overline{A} + \overline{Z}\,(X + \overline{Y}\,\overline{W})(Y + \overline{Z}\,\overline{W}))$$

Simplifying Expressions

$$\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$\downarrow$$

$$BC(\bar{A} + A) + A\bar{B}C + AB\bar{C}$$

$$\downarrow$$

$$BC(1) + A\bar{B}C + AB\bar{C}$$

$$\downarrow$$

$$BC + A\bar{B}C + AB\bar{C}$$

$$\downarrow$$

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

$$\downarrow$$

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

$$\downarrow$$

$$BC + AB + A\bar{B}C$$

$$\downarrow$$

$$BC + A(B + \bar{B}C)$$

$$\downarrow$$

$$BC + A(B + C)$$

$$\downarrow$$

$$BC + AB + AC$$