COMSM1302
Overview of Computer Architecture

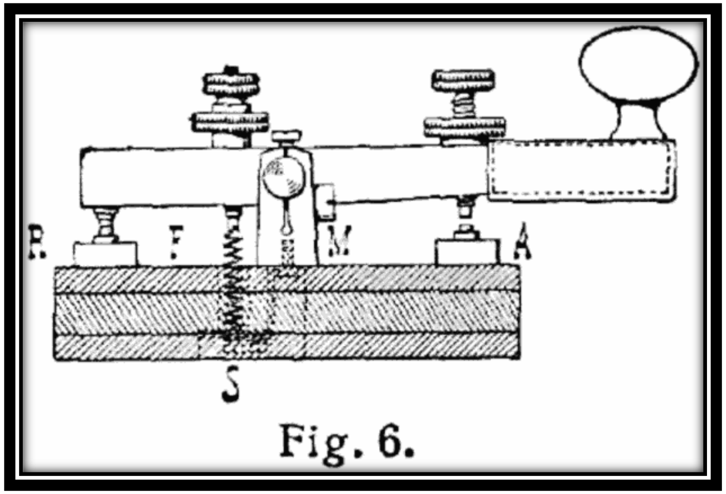
Lecture 3 – Transistor logic, CMOS

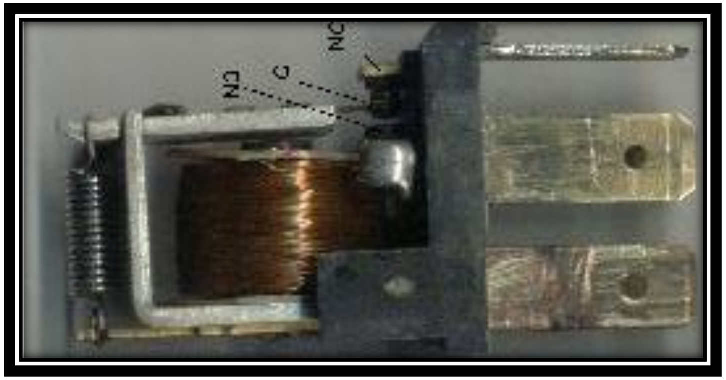
In this lecture

Foundations	Data representation, logic.
Building blocks	**Transistors, transistor based logic**, simple devices, storage.
Modules	Hex modules, memory, simple controller and processor.
Programming	Assembly, assembler, language, compilation phases, boot-strapping.
Bigger systems	ARM & Thumb, I/O, protecting shared systems, memory hierarchy, multi-processors, networks.
Wrap-up	More examples, historical computers, contemporary systems.

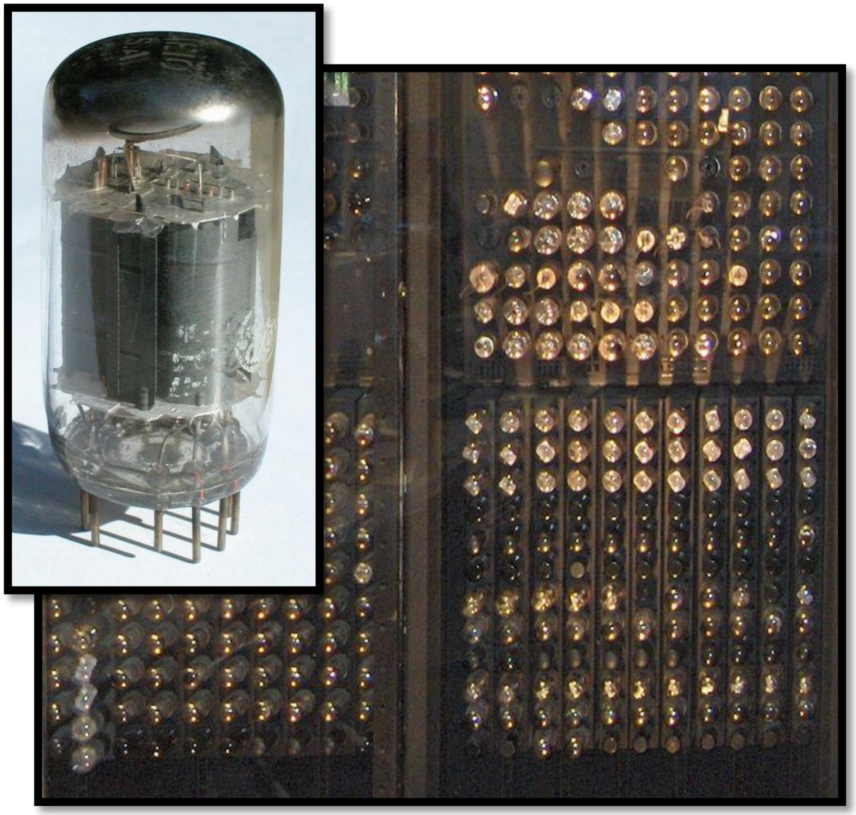
The switch - mechanical

- Mechanical switches are very useful
  - Turn things on/off
  - With several switches, we can **encode** more useful things.
- What's wrong with them?





The switch - valve



Bank of valves from the ENIAC computer.
Photo source: TexDex, Wikimedia Commons.

- Valves or vacuum tubes.
- Current controlled by thermionic emission.
  - They have a **heating element**.
- Quicker than mechanical switches.
- Fairly reliable
  - If they're **kept on**!

The switch - silicon

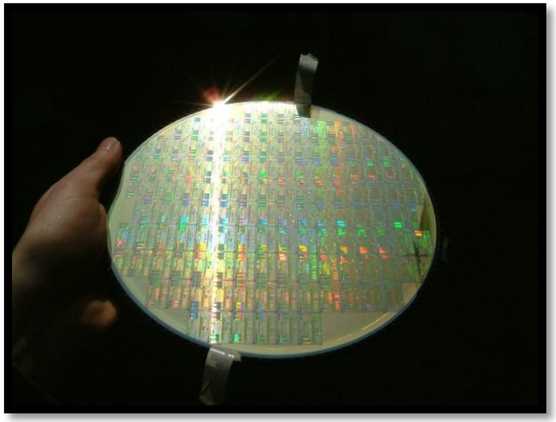
- Silicon, the element “Si”
  - The second most **abundant element** on Earth.
- A **semiconductor**.
  - Can be constructed to pass electrons through a channel, when a voltage is applied to a **gate**.

Ore

Boule

Source: Stahlkocher, Wikimedia Commons

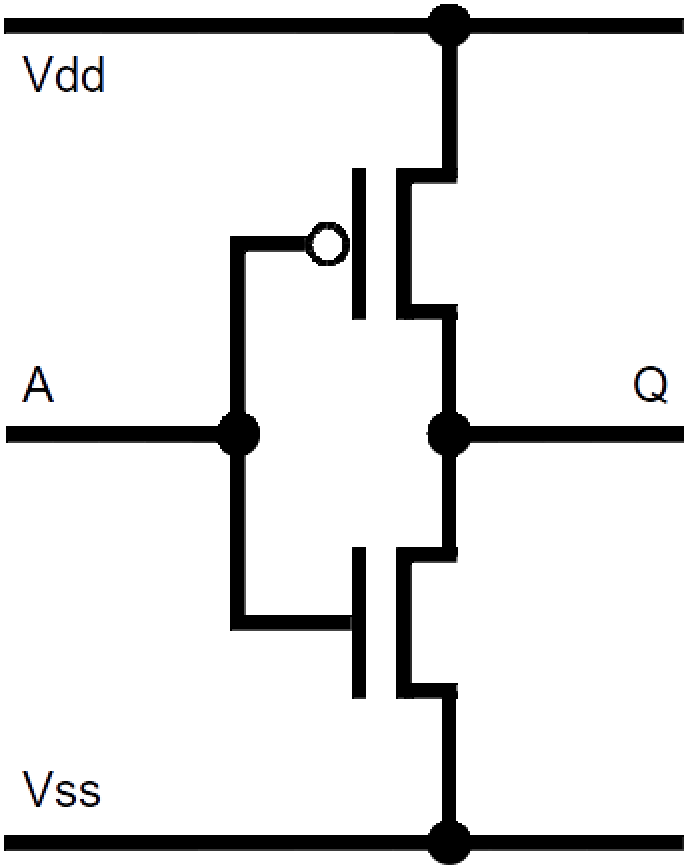
Wafer



Source: James Irwin, CC 2.0

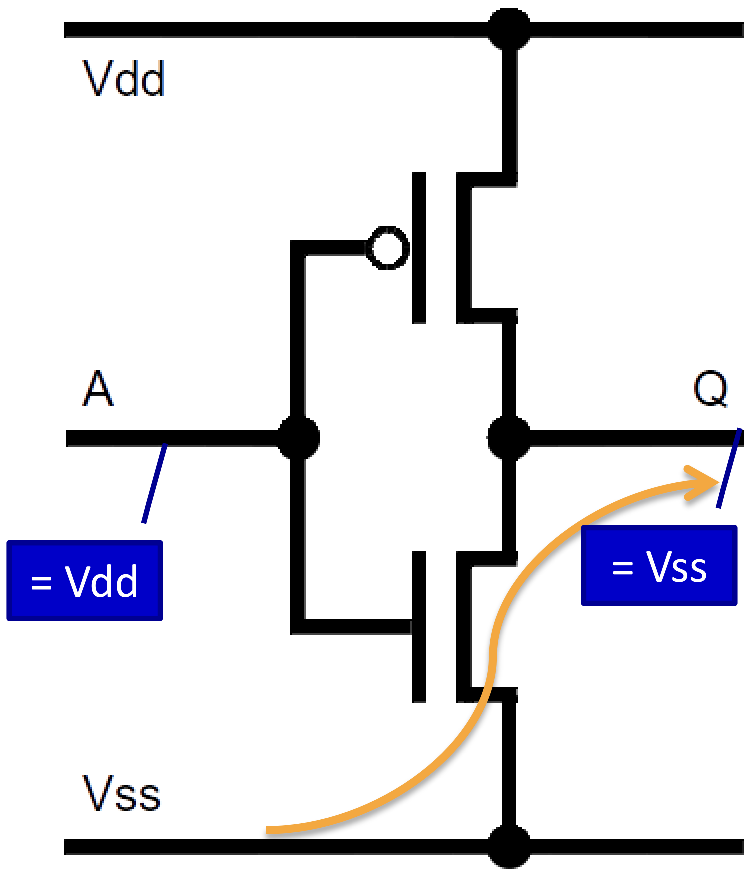
Making a switch out of silicon

- The **transistor**.
- Multiple uses
  - Amplification
  - **Switching**
- Multiple methods of construction
  - We're interested in **Integrated Circuits** (ICs), so we want **CMOS**

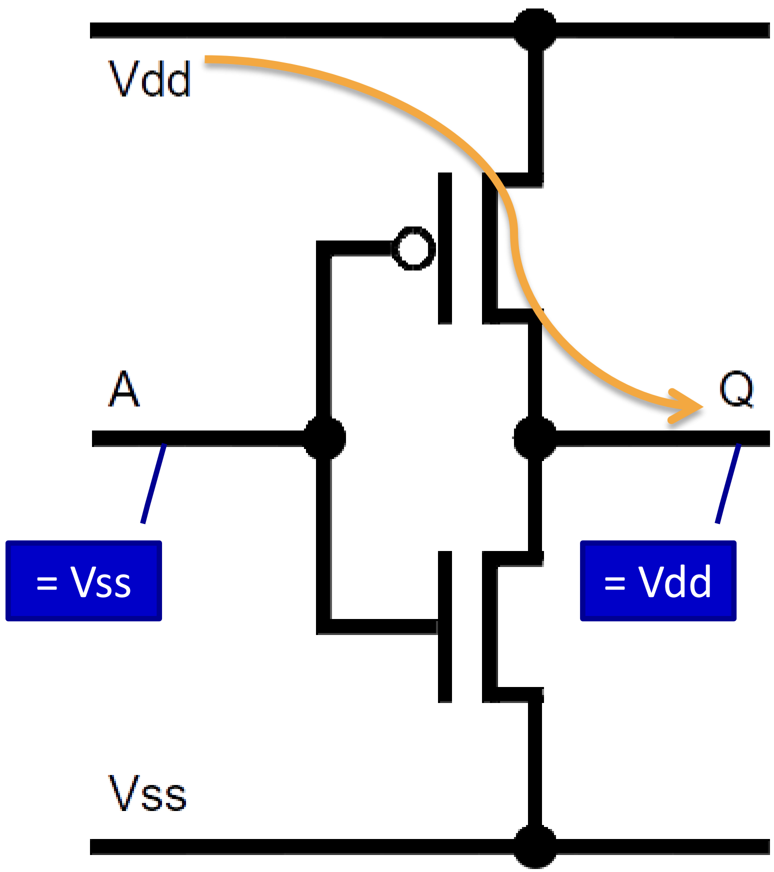


Modes of operation

- A = Vdd.
- The top “switch” is **off**.
  - PMOS transistor.
- The bottom “switch” is **on**.
  - NMOS transistor
- Q is “connected” to Vss.



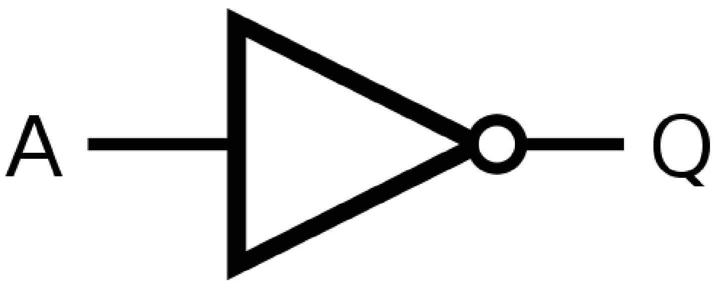
Modes of operation



- A = Vss.
- The top “switch” is **on**.
- The bottom “switch” is **off**.
- Q is “connected” to Vdd.

What did we just make?

- The circuit has **four connections**.
- Two are **power supply** related.
  - Providing Vdd and Vss.
- One is an **input**.
- One is an **output**.



Symbolic

A	Q
Vss	Vdd
Vdd	Vss

Voltage

A	Q
0 V	3.3 V
3.3 V	0 V

Binary

A	Q
0	1
1	0

PMOS + NMOS = CMOS

- PMOS is good for making connections to **Vdd**.
- NMOS is good for making connections to **Vss**.
- You can't make a reliable switch (or inverter) with just one type.
  - We either get 1/? or ?/0; we want 1/0.
- So we use a **pair**, one PMOS, one NMOS.
- They are **complementary**.

CMOS

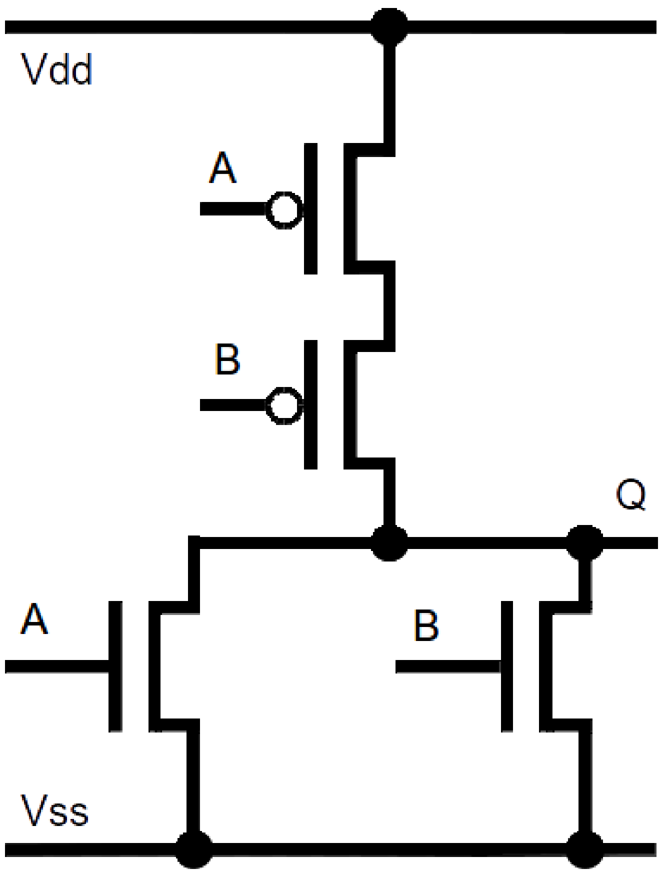
Complementary Metal Oxide Semiconductor

Boolean logic in CMOS

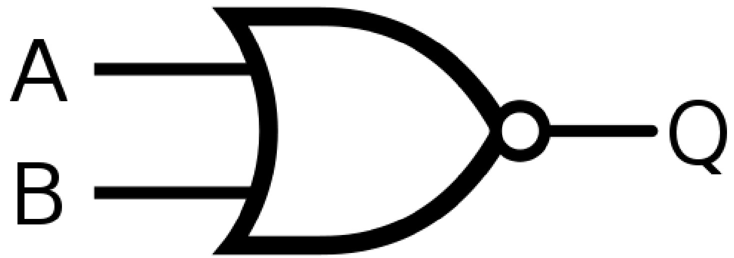
- We have a CMOS inverter.
  - Implements "not"
  - In a logic circuit, we call this a **NOT gate**.
- To build a more complex circuit, we need more than just a not gate.

So… what can we make?

NOR (NOT-OR)

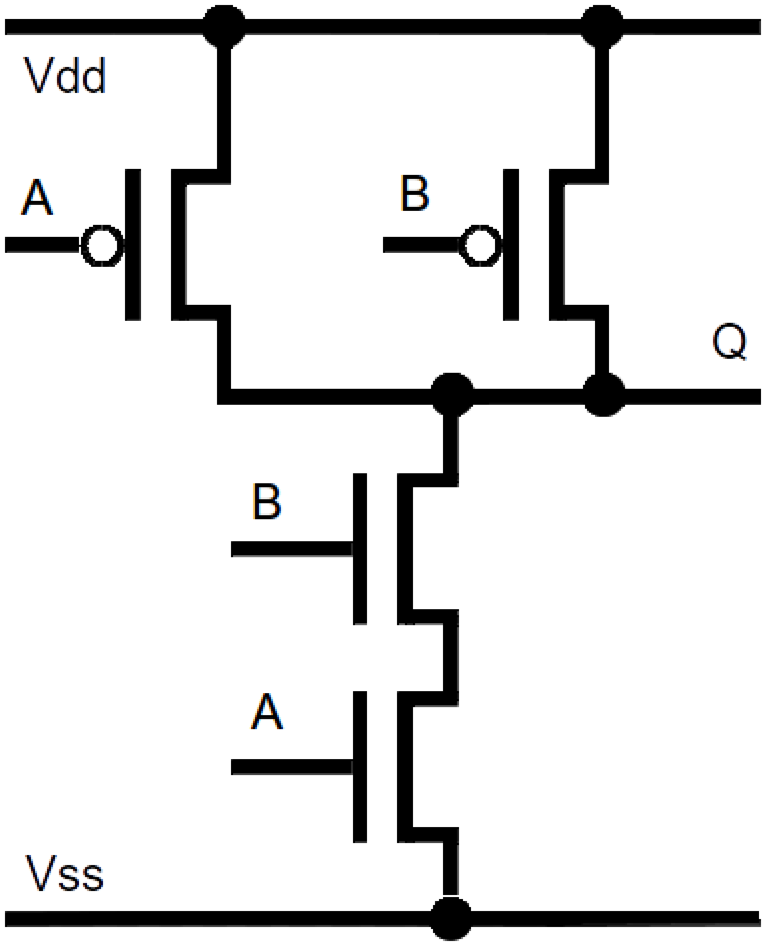


- We could build a NOR gate.

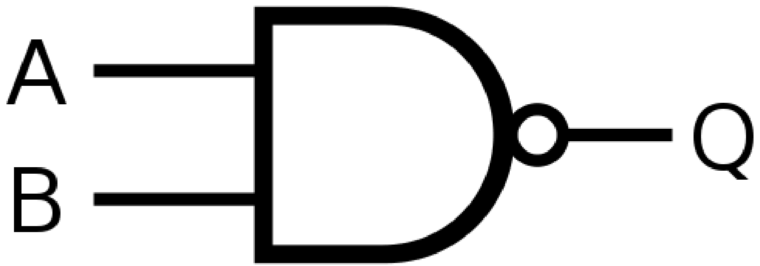


Explanation on board

NAND (NOT-AND)



- We could also build a NAND gate.

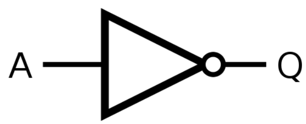


Explanation on board

Why NAND?

- NAND is an excellent **building block** for other Boolean logic.

- For example: **NOT**



Diagram

Algebra

$$\neg(A \land A) \equiv \neg A$$

- Idempotency axiom

Truth table

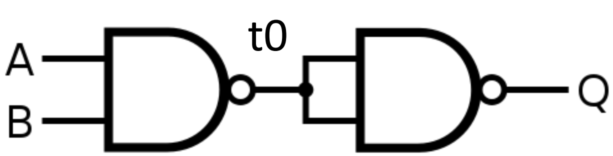
A	Q
0	1
1	0

Why NAND?

- NAND is an excellent **building block** for other Boolean logic.

- For example: **AND**

Diagram



Algebra

$$\neg(\neg(A \land B)) \equiv A \land B$$

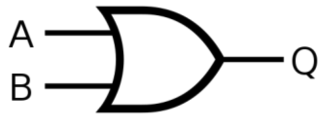
- Involution axiom

Truth table

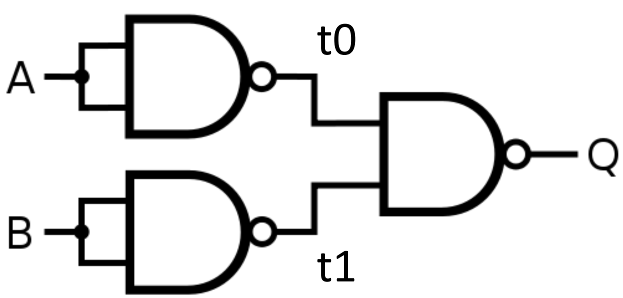
A	B	t0	Q
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

Why NAND?

- NAND is an excellent **building block** for other Boolean logic.
- For example: **OR**



Diagram



Algebra

$\neg(\neg(A \land A) \land \neg(B \land B))$

$\equiv \neg(\neg A \land \neg B)$

$\equiv \neg(\neg(A \lor B)$

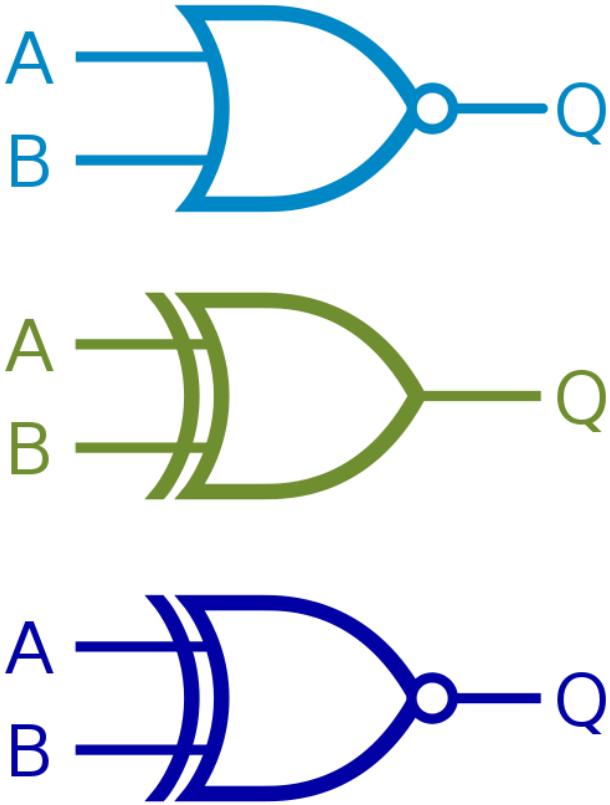
$\equiv A \lor B$

- Idempotency, involution & deMorgan

Truth table

A	B	t0	t1	Q
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

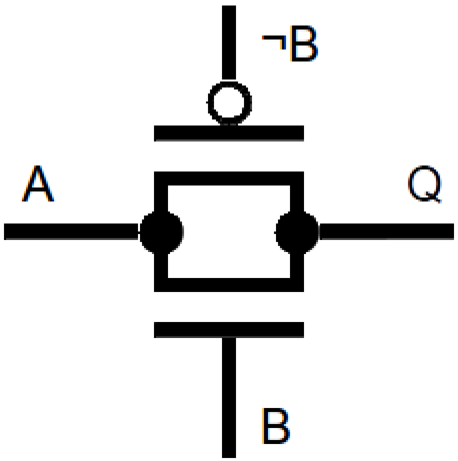
Why NAND?



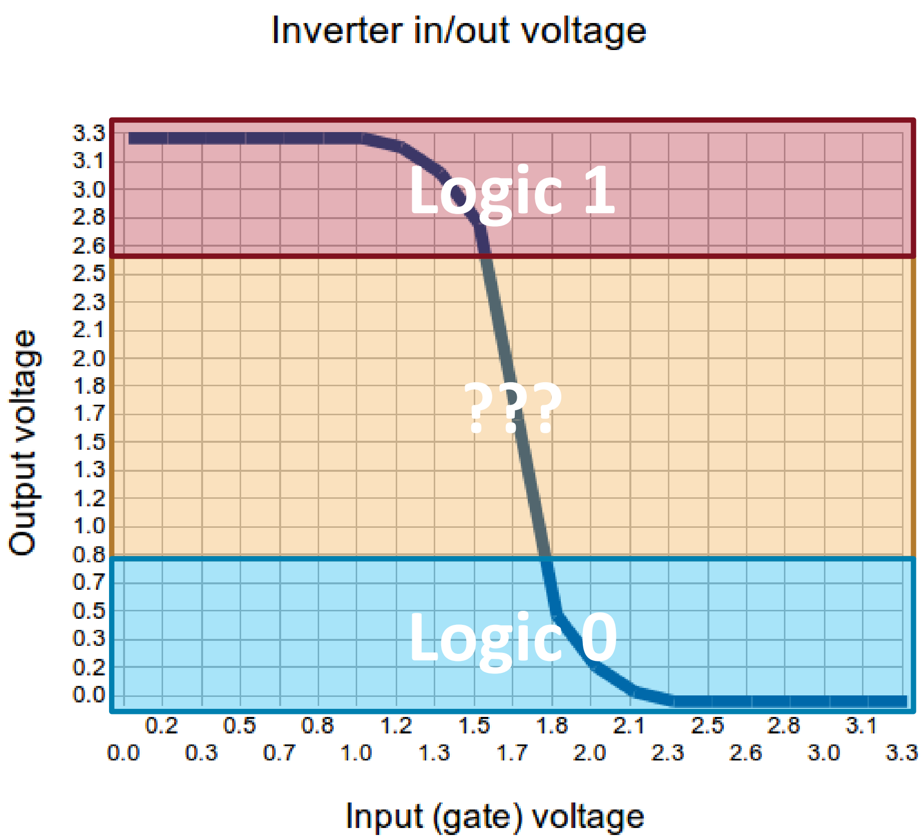
- NAND is an excellent **building block** for other boolean logic.
- NAND is **functionally complete**.
  - All gates can be expressed with NAND gates arranged in various ways.
- NOT, AND, OR as previously shown.
  - As well as **NOR**, **XOR**, **XNOR**.

Notes on NAND, NOR, inversion

- NOR is also functionally complete.
  - NOR structure is slower than NAND.
- Custom cells can be designed
  - Less silicon area, faster, …
  - Alternative devices such as pass transistors.
    - New ways of implementing logic.
    - Can create problems with signal integrity.



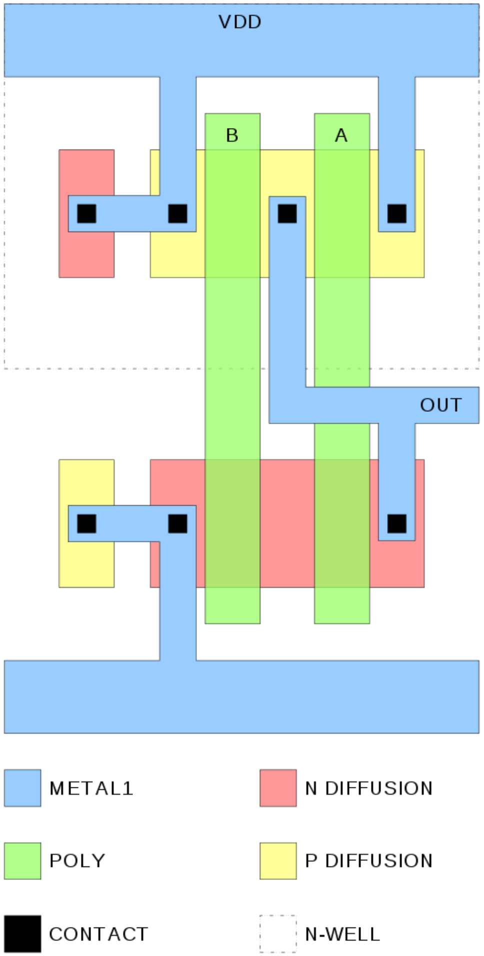
Voltages and logic levels



- Simple example.
- **Ideal** inverter would be **right-angled** response.
- Steepness of position of curve dependent on transistor **properties**.
  - Size ratio between p- and n-MOS
- There is also a **delay** between a change in input producing a change in output.

Summary

- Switches
  - Mechanical
  - Thermionic
  - Silicon
- Transistors
- CMOS
  - Inverter
  - NAND (functionally complete)
  - Making other gates from NAND



In this unit

Foundations	• Data representation, logic.
Building blocks	• **Transistors, transistor based logic**, simple devices, storage.
Modules	• Hex modules, memory, simple controller and processor.
Programming	• Assembly, assembler, language, compilation phases, boot-strapping.
Bigger systems	• ARM & Thumb, I/O, protecting shared systems, memory hierarchy, multi-processors, networks.
Wrap-up	• More examples, historical computers, contemporary systems.