Durham University

COMP1071 - Digital Electronics

Transistors and Logic Gates

Ioannis Ivrissimtzis

email: ioannis.ivrissimtzis@durham.ac.uk

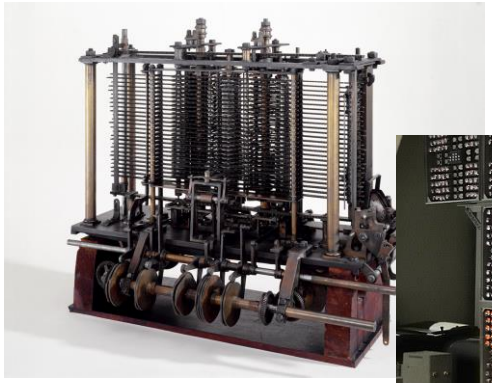
Room: MCS 2023

Slide acknowledgements: Eleni Akrida and Farshad Arvin

Overview of today's lecture

- Transistors
- Logic gates and truth tables
- Logic Levels
- Moore's Law

Transistors



Babbage's Analytical Engine – gears

Early electrical computers – relays or vacuum tubes

Modern computers – **transistors!**

Electrically controlled switches that turn ON or OFF when voltage or current is applied to a control terminal.

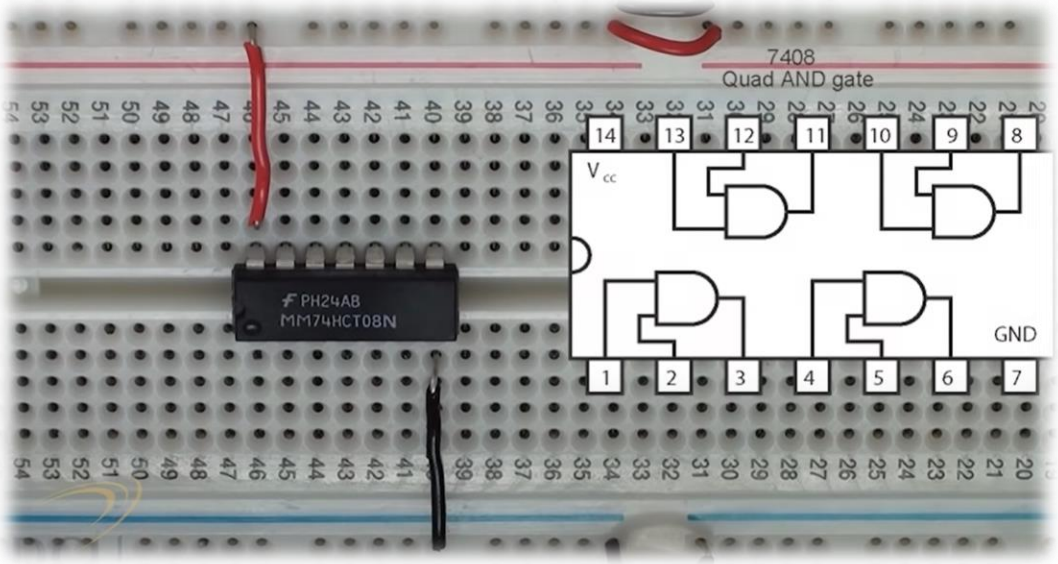
Most common transistor is the MOSFET:

Metal-Oxide-Semiconductor Field Effect Transistor.

- 57 billion MOSFETs onto a single chip of silicon!
- Building blocks of almost all digital systems.

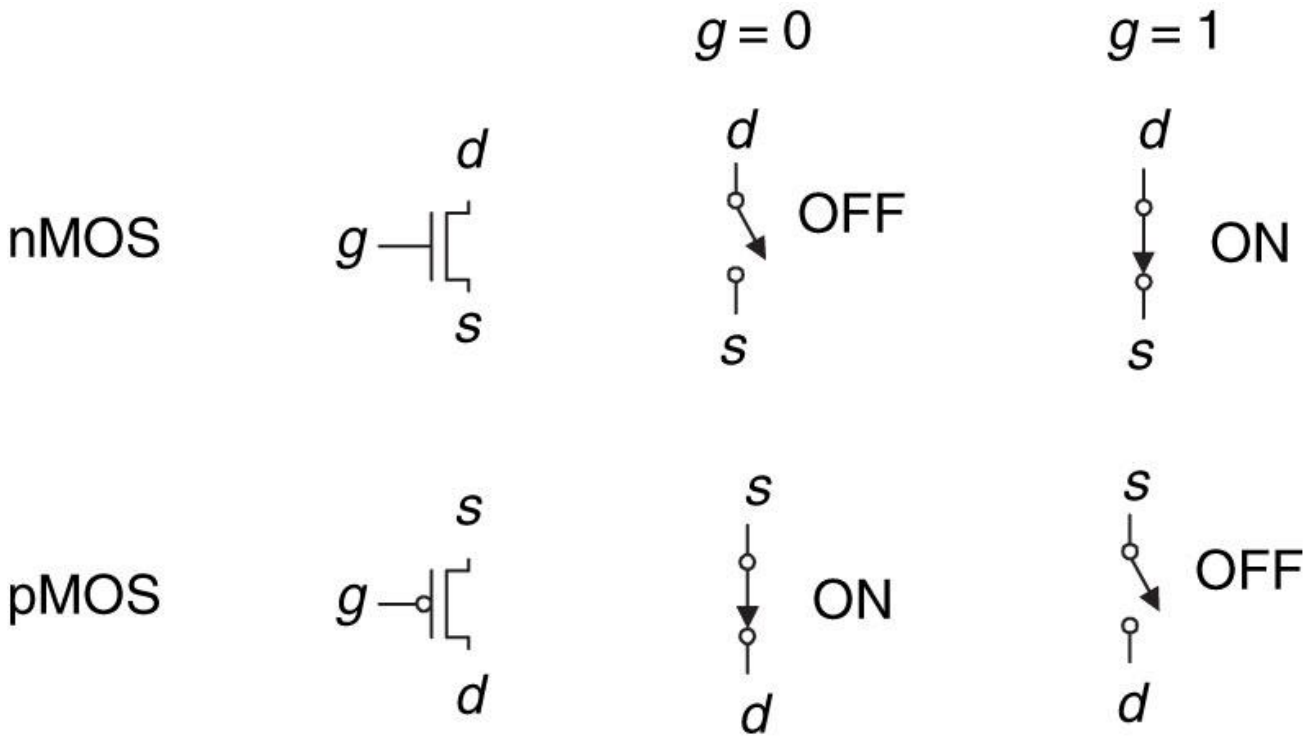
From Transistors to Digital circuits

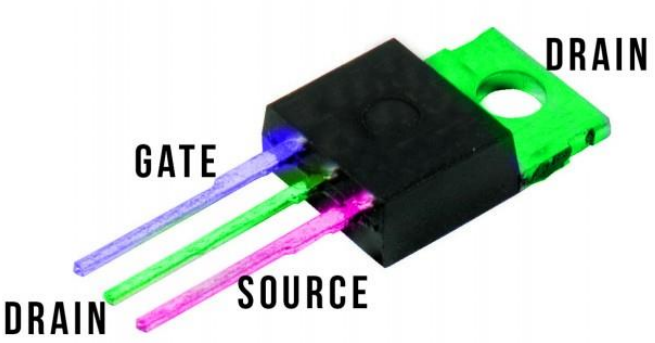
- Digital Circuits are constructed from small electronic circuits called **logic gates**
  - Each logic gate is designed to perform a function of **Boolean logic**
  - A logic gate is created from one or more **transistors**
    - The output of a logic gate can feed into more logic gates…

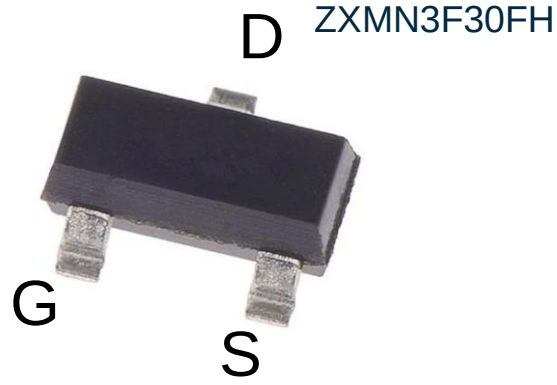


Transistors

- 2 ports (**d** and **s**) are connected depending on voltage of 3rd (**g**)



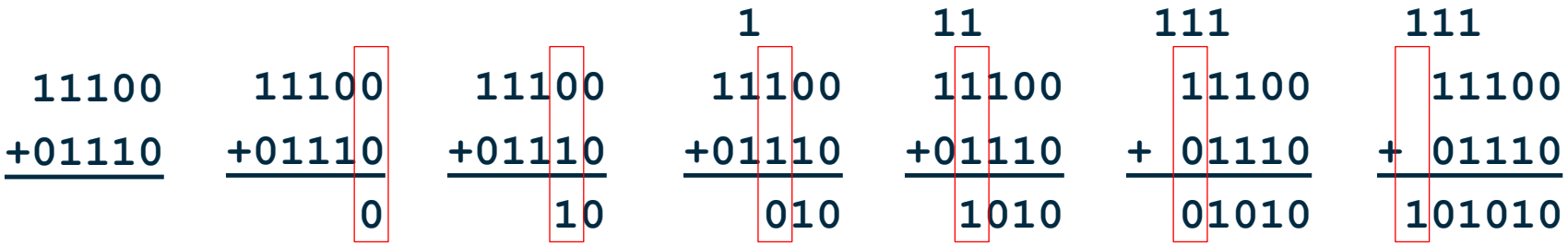




For more detail, check: *3.1 Transistors_Extra Material.ppt*

Adding in binary

Example



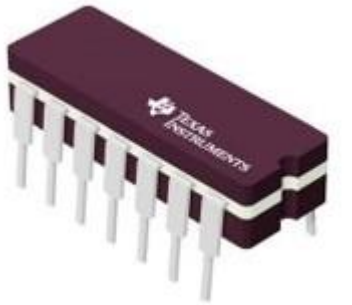
Based on 8 simple rules:

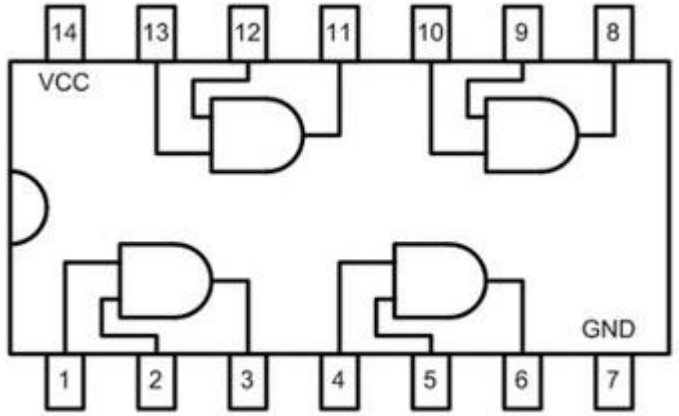
0 + 0 = 0	Carry + 0 + 0 = 1
0 + 1 = 1	Carry + 0 + 1 = 0 with Carry
1 + 0 = 1	Carry + 1 + 0 = 0 with Carry
1 + 1 = 0 with Carry	Carry + 1 + 1 = 1 with Carry

Durham University

How is this achieved?

- Using **gates** implementing **Boolean algebra**.
- **Boolean algebra:**
  an algebra of two values: **1,0**
  or **True/False** or **high/low voltage**.
- Basic operations: AND, OR, NOT
  0 AND 0 = 0
  0 AND 1 = 0
  1 AND 0 = 0
  1 AND 1 = 1





Truth tables: AND

- **AND** gate: 

Algebraic expression: $Y = A \cdot B$

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

Linear truth table

AND	0 (B)	1
0 (A)	0	0
1	0	1

Rectangular/Coordinate table

Truth tables: OR

- **OR** gate: 

Algebraic expression: $Y = A + B$

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

Linear truth table

OR	0	B 1
0 A	0	1
1	1	1

Rectangular/Coordinate table

Truth tables: Exclusive OR (XOR)

- **XOR** gate: 

Algebraic expression: $Y = A \oplus B$

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

Linear truth table

AND	0	1
0	0	1
1	1	0

(A: 0, 1; B: 0, 1)

Rectangular/Coordinate table

Truth tables: NOT

- **NOT** gate: 

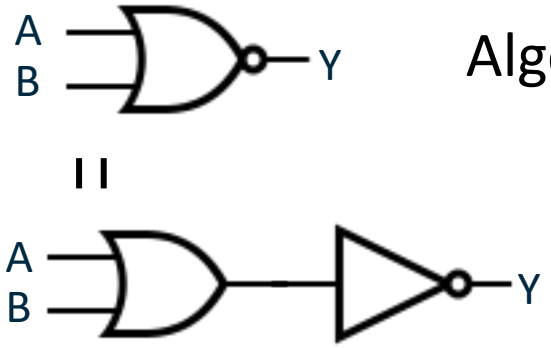
Algebraic expression: $Y = \overline{A}$

A	Y
0	**1**
1	**0**

Linear truth table

Truth tables: NOR

- **NOR** gate (inverse of OR)



Algebraic expression: $Y = \overline{A + B}$

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

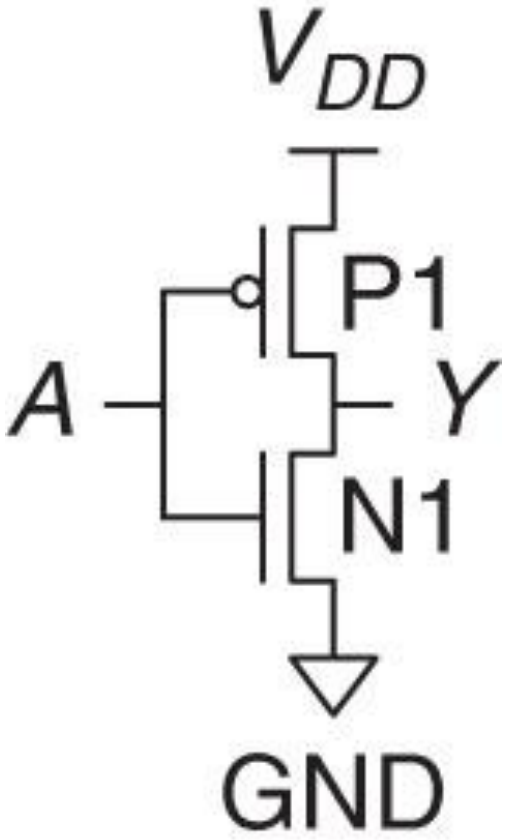
Linear truth table

From transistors to gates

- A **NOT** gate: 

- If **A** is **high** then the p-MOS P1 is off and the n-MOS N1 is on, so **Y** is connected to GND, i.e. **low**.

- If **A** is **low** then the p-MOS P1 is on and the n-MOS N1 is off, so **Y** is connected to V_{DD}, i.e. **high**.

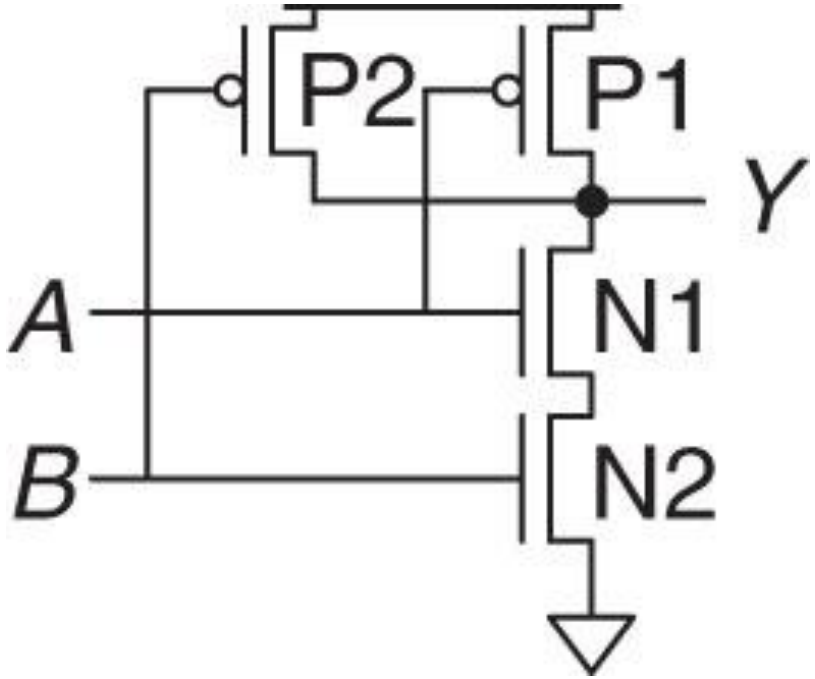


A	P1	N1	Y
0	**on**	**off**	**1**
1	**off**	**on**	**0**

From transistors to gates

- A **NAND** gate:

A	B	P1	P2	N1	N2	Y
0	0					
0	1					
1	0					
1	1					

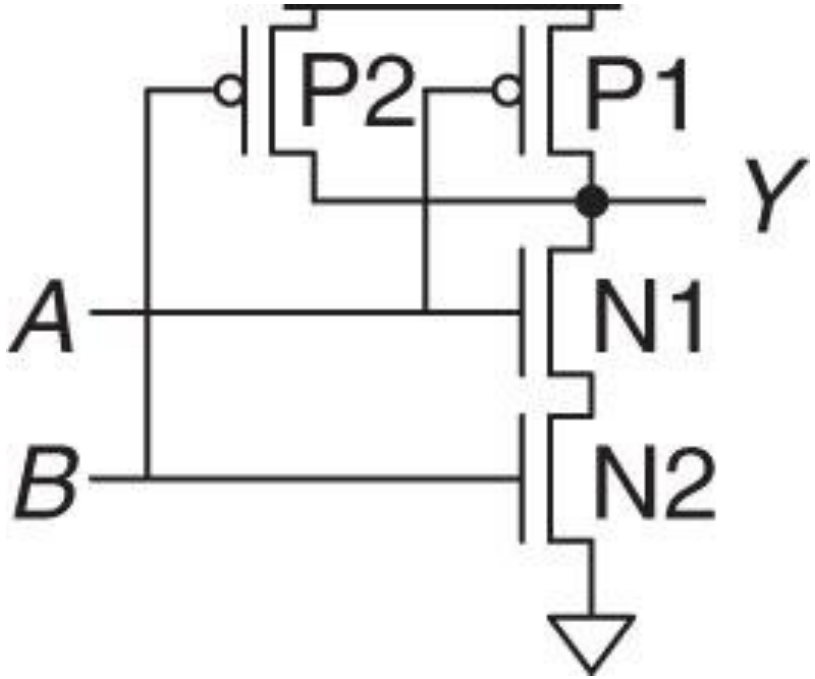


Note: 3-wire junctions are connected, but 4-wire junctions are only connected if there is a dot.

From transistors to gates

- A **NAND** gate:

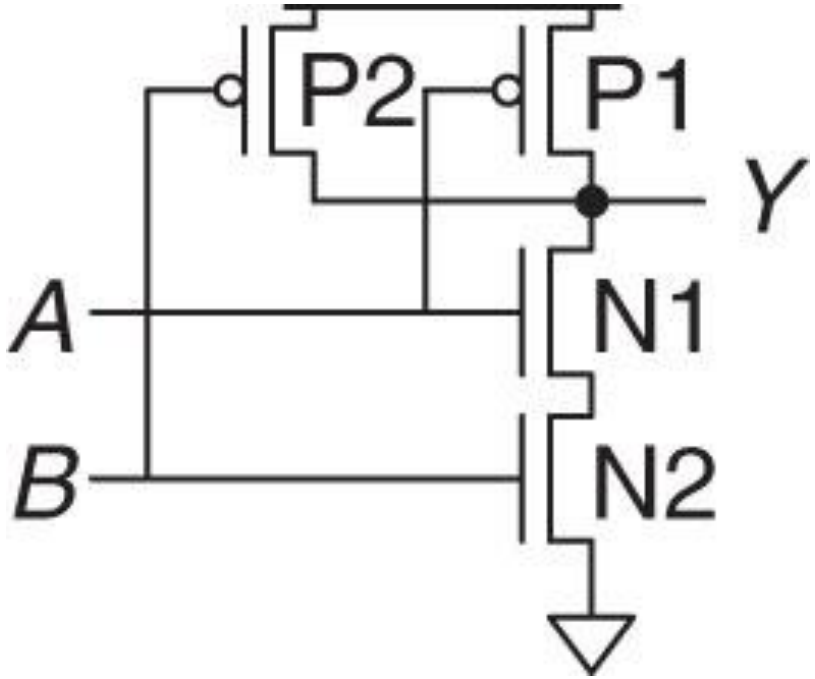
A	B	P1	P2	N1	N2	Y
0	0	**on**	**on**	**off**	**off**	**1**
0	1					
1	0					
1	1					



From transistors to gates

- A **NAND** gate:

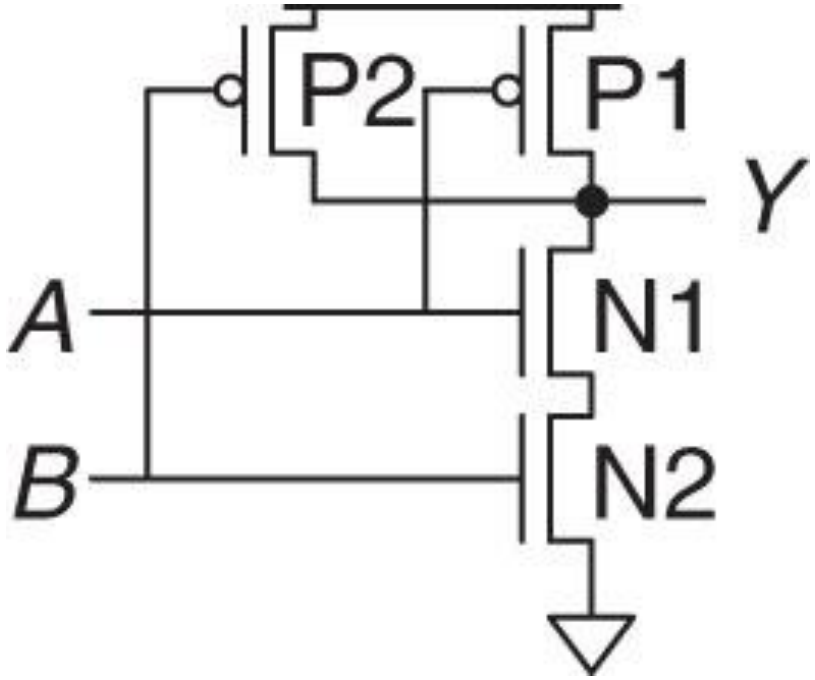
A	B	P1	P2	N1	N2	Y
0	0	**on**	**on**	**off**	**off**	**1**
0	1	**on**	**off**	**off**	**on**	**1**
1	0					
1	1					



From transistors to gates

- A **NAND** gate:

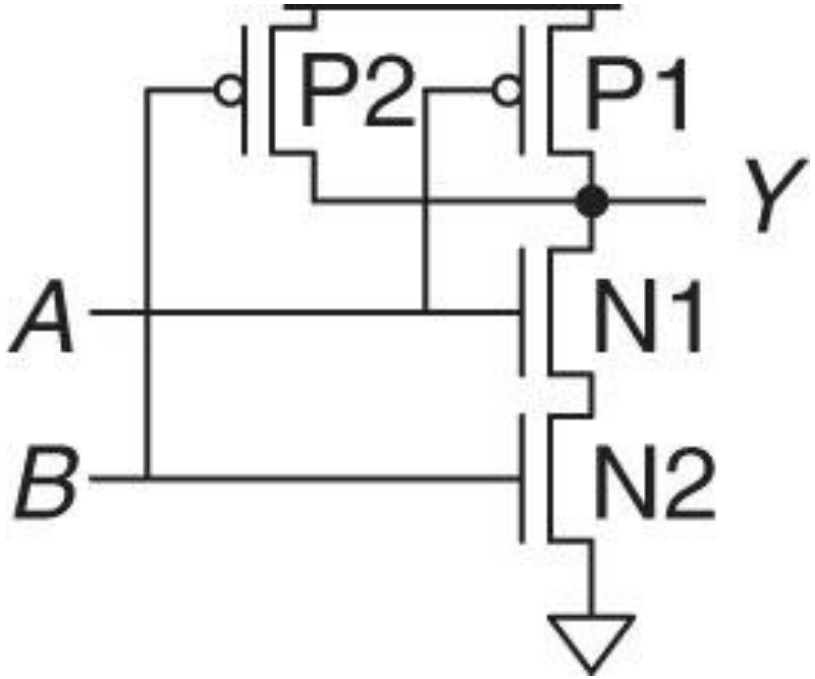
A	B	P1	P2	N1	N2	Y
0	0	**on**	**on**	**off**	**off**	**1**
0	1	**on**	**off**	**off**	**on**	**1**
1	0	**off**	**on**	**on**	**off**	**1**
1	1					



From transistors to gates

- A **NAND** gate:

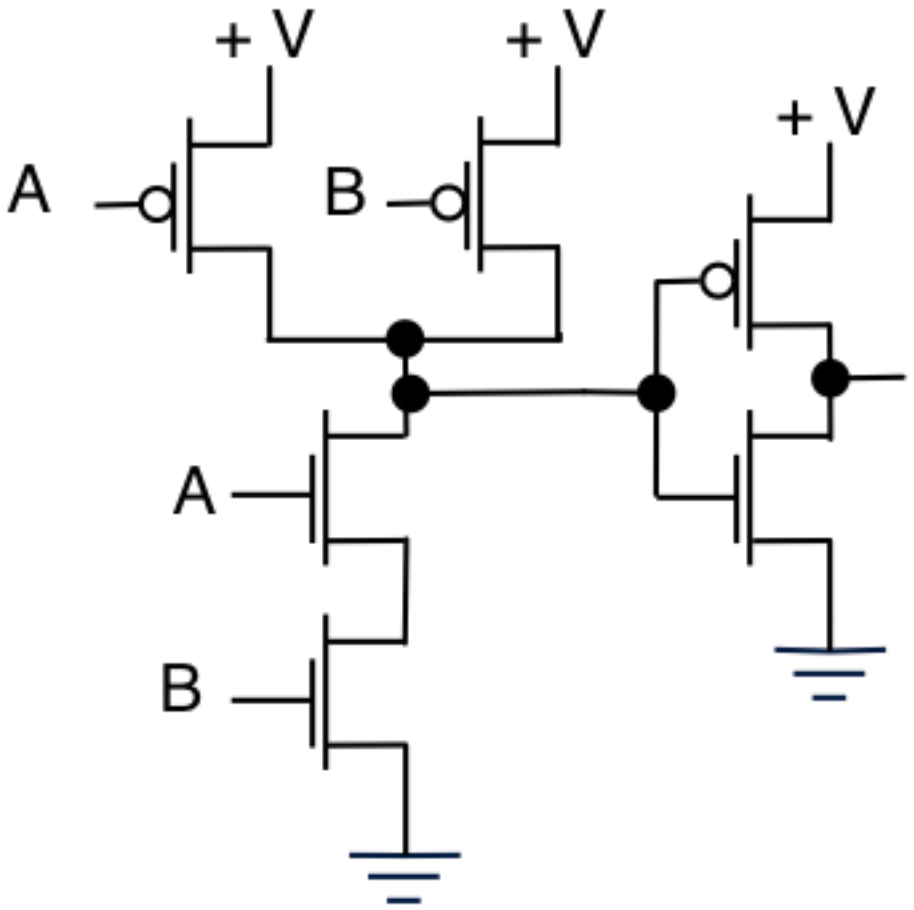
A	B	P1	P2	N1	N2	Y
0	0	on	on	off	off	1
0	1	on	off	off	on	1
1	0	off	on	on	off	1
1	1	off	off	on	on	0



From transistors to gates

- An **AND** gate:

Essentially made by combining a NAND and a NOT gate.



Durham University

Beneath the digital abstraction

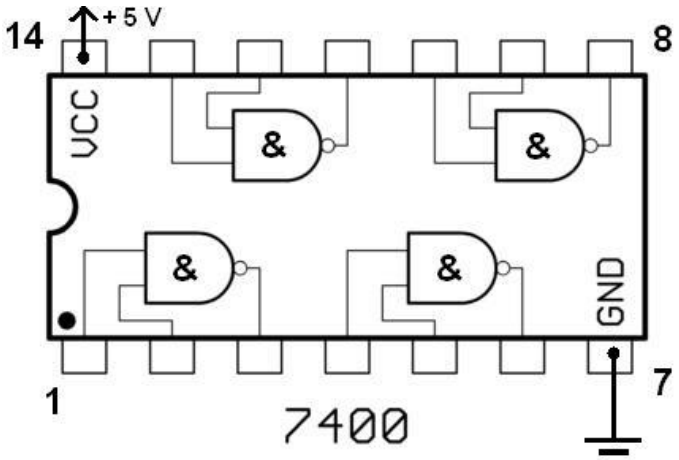
A chip does not really deal in 0s and 1s.

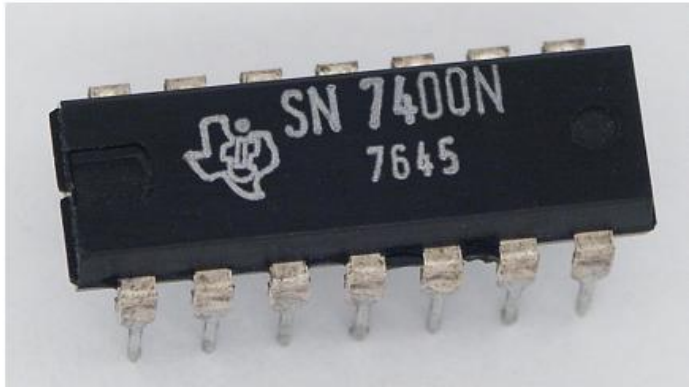
The **voltages** are real numbers between 0V and 5V (typically).

We can take 0V to indicate output 0 and 5V to indicate output 1, but we need to tolerate **noise**.

If the output is 4.9V that is probably meant to be a 1.

But what about 4.1V or 3V or 2.5V?





Supply voltage

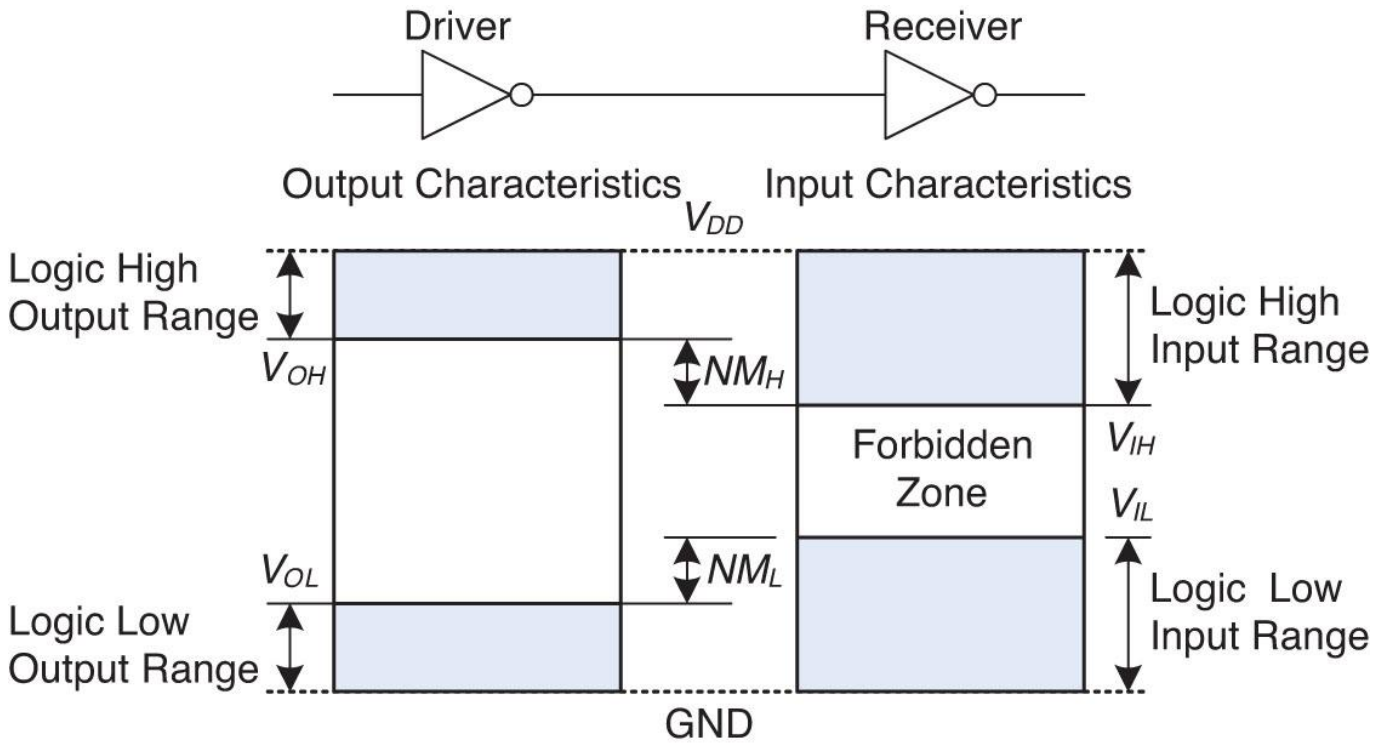
The **low** voltage in the system (connected to **ground** or GND) is 0V.

Historically the circuit was powered by connecting to a 5V supply, and the **high** voltage is therefore 5V, called V_{DD} if unspecified.

More modern chips, using much smaller transistors, have moved to V_{DD} = 3.3V, 2.5V, 1.8V, 1.5V and 1.2V, both to save power and avoid overloading transistors.

The mapping of the continuous voltage measured at any point in the circuit to the discrete 0 and 1 of the digital abstraction is governed by defining **logic levels**.

Logic levels



Permitted range for high output: V_{OH} to V_{DD}

Permitted range for low output: GND to V_{OL}

Acceptable range for high input: V_{IH} to V_{DD}

Acceptable range for low input: GND to V_{IL}

Noise margins:

$NM_H = V_{OH} - V_{IH}$

$NM_L = V_{IL} - V_{OL}$

Durham University

The Static Discipline

- A guarantee that "if inputs meet **valid input thresholds**, then the system guarantees outputs will meet **valid output thresholds**" and will not fall into the forbidden zone
- Gates are grouped into **logic families**: the gates of the same family obey the static discipline when used with other gates in the family.
- This means that (given noise limits) you can successfully apply the **digital abstraction** and **combine elements** without further concern about logic levels or analogue values.

Logic Family	V_{DD}	V_{IL}	V_{IH}	V_{OL}	V_{OH}
TTL	5 (4.75 - 5.25)	0.8	2.0	0.4	2.4
CMOS	5 (4.5 - 6)	1.35	3.15	0.33	3.84
LVTTL	3.3 (3 - 3.6)	0.8	2.0	0.4	2.4
LVCMOS	3.3 (3 - 3.6)	0.9	1.8	0.36	2.7

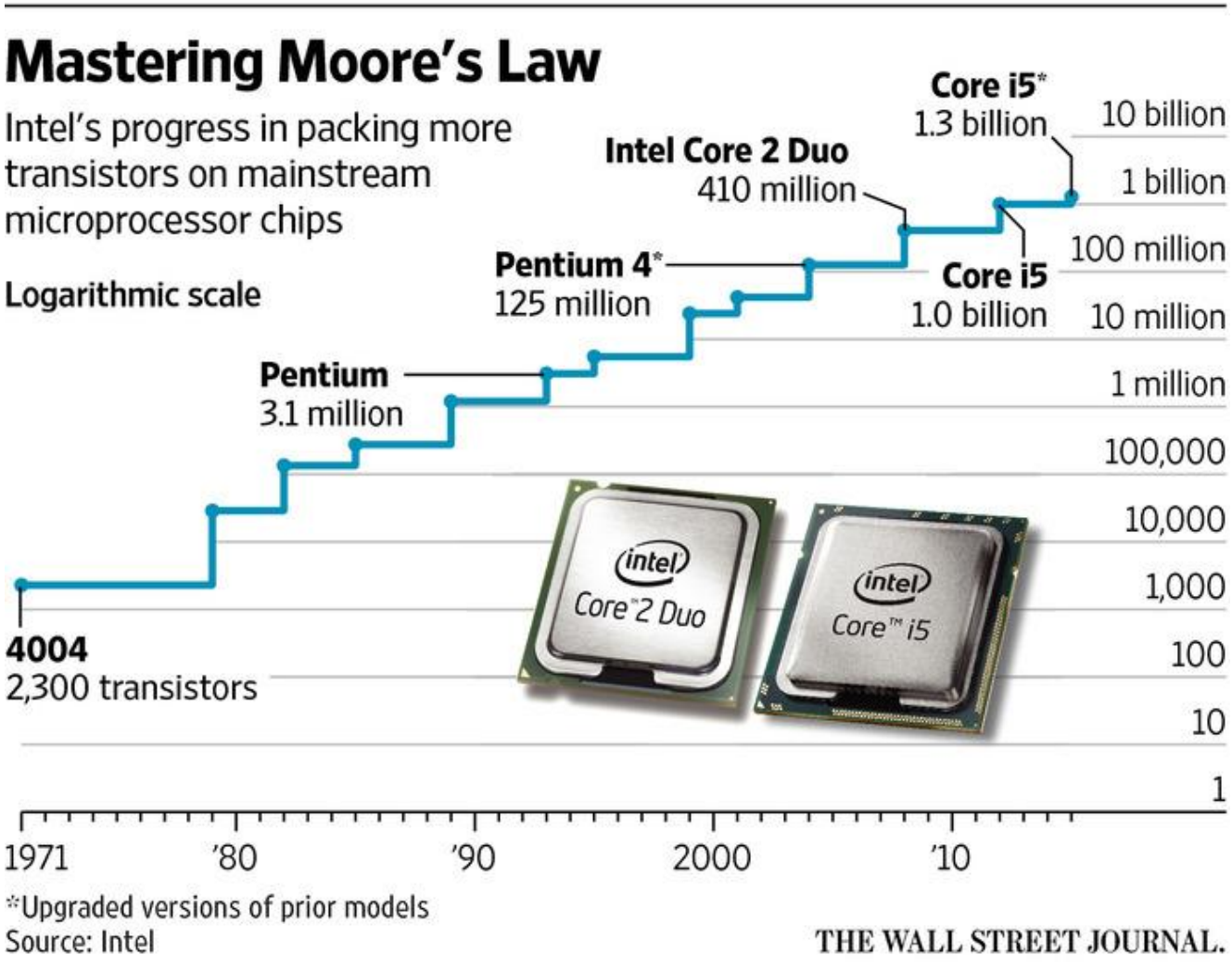
Moore’s law

Gordon Moore co-founded Intel in 1968.

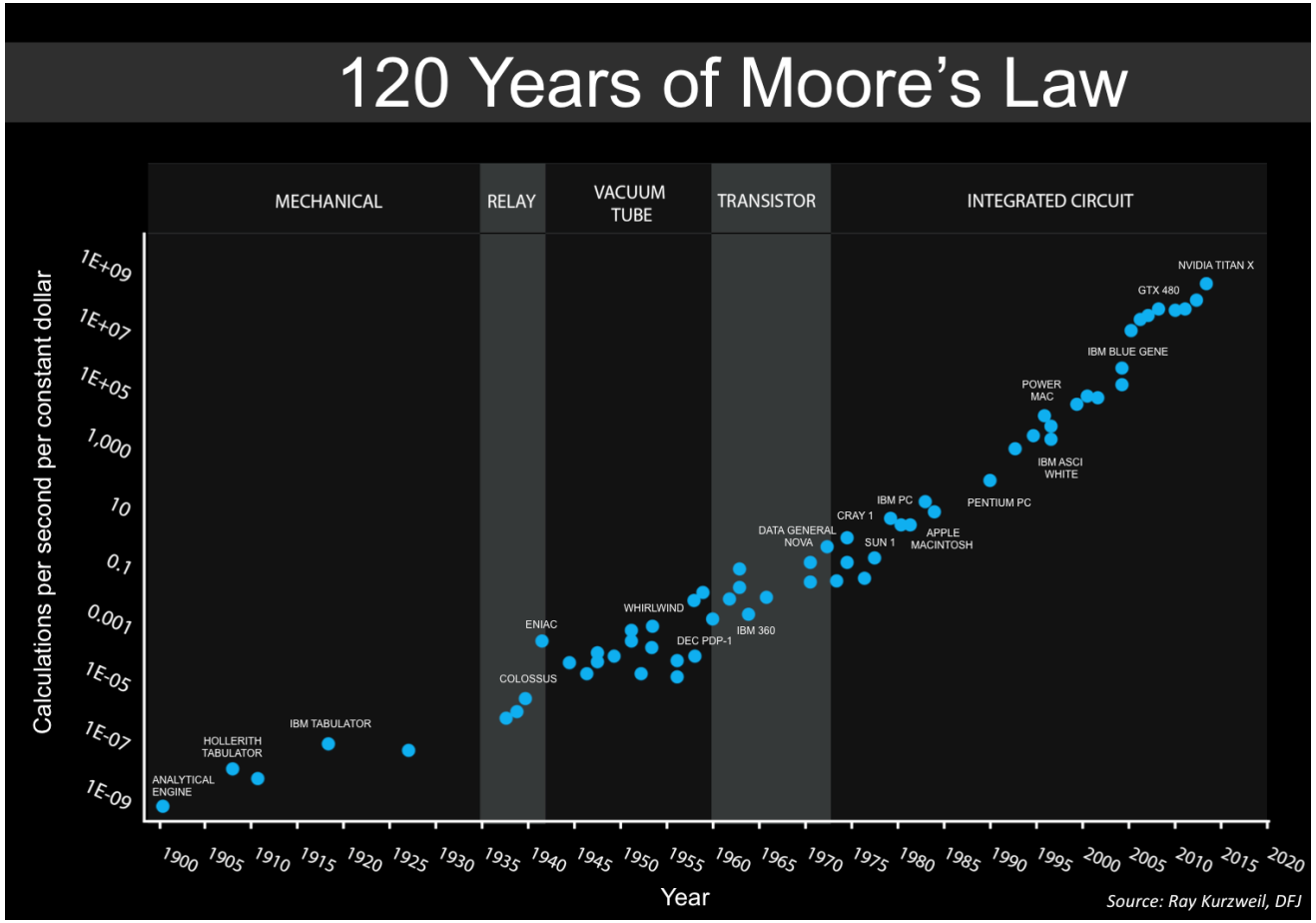
In 1965 he observed that the number of transistors on a computer chip doubles every year.

Moore’s law: normally quoted as “transistor density doubles in 2 years”, or “computer processing power doubles every 18 months”.

Moore's law


Mastering Moore's Law
Intel's progress in packing more transistors on mainstream microprocessor chips
Logarithmic scale
4004
2,300 transistors
Pentium
3.1 million
Pentium 4*
125 million
Intel Core 2 Duo
410 million
Core i5
1.0 billion
Core i5*
1.3 billion
10 billion
1 billion
100 million
10 million
1 million
100,000
10,000
1,000
100
10
1
1971
'80
'90
2000
'10
*Upgraded versions of prior models
Source: Intel
THE WALL STREET JOURNAL.

Moore's law



transistors available for use by a 'domestic' computer.
A lot of the growth recently has been in more powerful graphics cards available.

Moore’s law

“If the automobile had followed the same development cycle as the computer, a Rolls-Royce would today cost $100, get one million miles to the gallon, and explode once a year . . .”

Robert Cringely

Summary

- Transistors, pMOS & nMOS
- Logic gates and truth tables
- Logic Levels
- Moore’s Law