Intro. to Computer Architecture

Daniel Page

Department of Computer Science,
University Of Bristol,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB. UK.
⟨csdsp@bristol.ac.uk⟩

January 9, 2018

Keep in mind there are *two* PDFs available (of which this is the latter):

1. a PDF of examinable material used as lecture slides, and
2. a PDF of non-examinable, extra material:
   - the associated notes page may be pre-populated with extra, written explaination of material covered in lecture(s), plus
   - anything with a "grey'ed out" header/footer represents extra material which is useful and/or interesting but out of scope (and hence not covered).

Notes:

Notes:

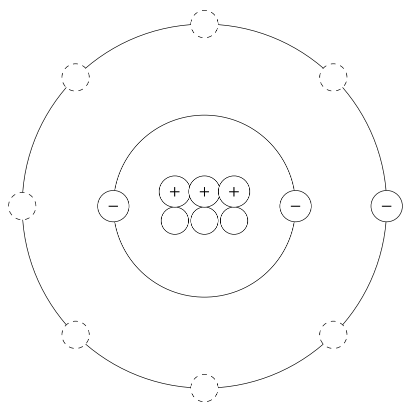
- The concrete implementation of most digital logic boils down use of **switches**: if the switch is
  - *on* then it conducts electricity,
  - *off* then it doesn't.
- Higher-level use-cases demand we make the switches as
  1. efficient (in space and time),
  2. robust, and
  3. manufacturable

  as possible, and thus turn electronics into *micro*-electronics.

Notes:

"A-level Physics in 10 minutes" (1)

Example (lithium atom)



- An **atom** is built from
  1. a group of nucleons, either **protons** or **neutrons**, called the nucleus, and
  2. a cloud of **electrons** arranged in shells.
- The protons dictate the atomic number; the number of neutrons dictates the isotope.
- The electron configuration is not arbitrary: the n-th shell can accommodate upto $2n^2$ electrons.
- An unfilled slot within a shell is called a **hole**.

Notes:

- Each sub-atomic particle has a (small) electrical **charge** associated with it
  1. electrons have a negative charge, i.e., $-$,
  2. protons have a positive charge, i.e., $+$, and
  3. neutrons have a neutral charge

  and an **ion** is an atom with a non-zero overall charge.

Notes:

"A-level Physics in 10 minutes" (2)

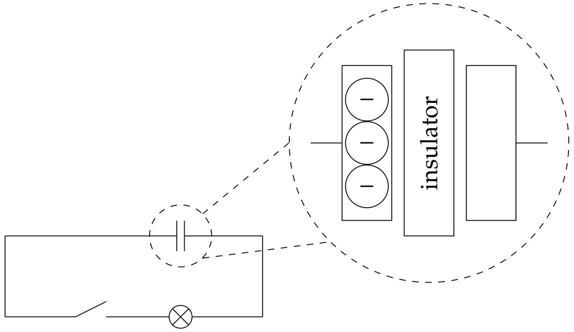
- The particles are bound together, but this binding can be disrupted:
  - As an electron absorbs more energy, it becomes excited; at some threshold, the electron is displaced and becomes free.
  - A free electron can move between shells *or* between atoms (from an outer shell); roughly speaking they are "attracted" by holes.

Notes:

© Daniel Page (csdsp@bristol.ac.uk)
Intro. to Computer Architecture git # 3b6f641 @ 2018-01-09 University of BRISTOL

"A-level Physics in 10 minutes" (3)

- An electrical **current** is a flow of electrons, i.e., a flow of charge; free electrons (bound to atoms or not) "move" from low to high potential ...
- ... this is effectively how a **capacitor** (i.e., a battery) works:



- A conductivity rating says how easily electrons can move:
  - a **conductor**, e.g., a metal, has high-conductivity (resp. low-resistivity) and allows electrons to move easily, and
  - an **insulator**, e.g., a vacuum, has low-conductivity (resp. high-resistivity) and does not allow electrons to move easily.

Notes:

"A-level Physics in 10 minutes" (4)

WE WERE GOING TO USE THE TIME MACHINE TO PREVENT THE ROBOT APOCALYPSE, BUT THE GUY WHO BUILT IT WAS AN ELECTRICAL ENGINEER.

http://xkcd.com/567/

Notes:

“A-level Physics in 10 minutes” (5)

- Silicon (Si) is neat because
  1. it’s **abundant**, i.e., there’s lots of it and so it doesn’t cost much,
  2. it’s fairly **inert**, i.e., it’s stable enough not to react in weird ways with other things, *and*
  3. it can be “**doped**” with a donor material, e.g.,
     - 3.1 a boron (B) or aluminium (Al) donor creates extra holes, or
     - 3.2 a phosphorus (P) or arsenic (As) donor creates extra electrons

     which, roughly speaking, allows us to construct materials with the exact sub-atomic properties we want.
- The result is a *semi*-conductor:
  - A P-type semi-conductor has extra holes, N-type has extra electrons.
  - If we sandwich P-type and N-type layers together, electrons can only move in *one* way, i.e., from an N-type layer to a P-type layer, but *not* in the other direction.

Notes:

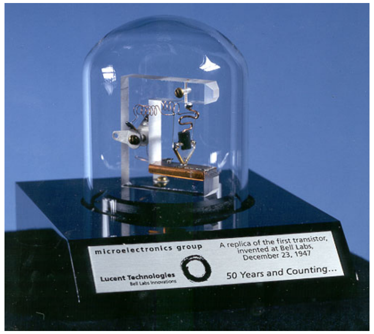
An Aside: History

- A historically dominant switch technology was the **vacuum tube**:
  - Looks like a light-bulb with a glass envelope holding a vacuum.
  - Inside vacuum is an electron producing filament (or cathode) and a metal plate (or anode).
  - When filament is heated, electrons are produced into vacuum which are attracted by plate.
  - Reliability of vacuum tubes was reasonably good, but most failed during power-on or power-off ...
  - ... the terms **bug** and **debug** both allegedly stem (in part) from failure of this sort.

http://en.wikipedia.org/wiki/File:6P1P.jpg

Notes:

An Aside: History



- The replacement for this generation of technology is the **transistor**.
- There are *many* types, but a potted overview of the one we'll focus on is:
  - 1925: **Field Effect Transistor (FET)**, initially designed and patented by Julius Lilienfeld ...
  - ... design not really used, due to lack of understanding and means of manufacture.
  - 1952: team led by William Shockley at Bell Labs invent junction gate FET (or JFET).
  - 1960: Dawon Kahng and Martin Atalla, also at Bell Labs invent the **Metal Oxide Semi-conductor FET (MOSFET)**.

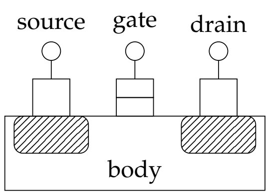
http://en.wikipedia.org/wiki/File:Replica-of-first-transistor.jpg

Notes:

Switches and Transistors (1)

- At a high level, the design of a MOSFET is as follows:

Circuit



- Idea [3]:
  - FET transistors allow charge to flow through a conductive channel between source and drain terminals.
  - The channel width, and hence conductivity, is controlled by the potential difference applied to gate terminal.
  - In a MOSFET transistor, the channel is *induced* (vs. a JFET, where an explicit semi-conductor layer is used).

noting that a more detailed description needs more Physics!

Notes:

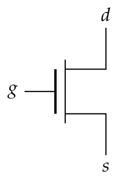
Switches and Transistors (2)

Definition

An **N-MOSFET** (or **N-type MOSFET**, or **N-channel MOSFET**, or **NPN MOSFET**) is constructed from N-type semiconductor terminals and a P-type body:

- applying a potential difference to the gate widens the conductive channel, meaning source and drain are connected (i.e., act like a conductor); the transistor is activated.
- removing the potential difference from the gate narrows the conductive channel, meaning source and drain are disconnected (i.e., act like an insulator); the transistor is deactivated.

Using d, s and g to denote the drain, source and gate terminals, an N-MOSFET is described symbolically as

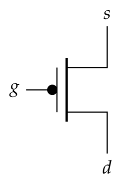


Definition

A **P-MOSFET** (or **P-type MOSFET**, or **P-channel MOSFET**, or **PNP MOSFET**) is constructed from P-type semiconductor terminals and an N-type body:

- applying a potential difference to the gate narrows the conductive channel, meaning source and drain are disconnected (i.e., act like an insulator); the transistor is deactivated.
- removing the potential difference from the gate widens the conductive channel, meaning source and drain are connected (i.e., act like a conductor); the transistor is activated.

Using d, s and g to denote the drain, source and gate terminals, an P-MOSFET is described symbolically as

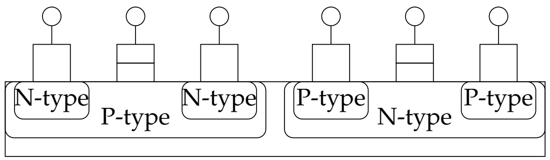


Notes:

Switches and Transistors (3)

- ... *but* they aren't typically used in isolation.
- A **CMOS** cell combines one N-MOSFET and one P-MOSFET; each transistor in a pair works in a **c**omplementary way:

Circuit



- We'll see later why this is functionally useful, but behaviourally
  1. only switching from one state to another consumes much power (or **dynamic consumption**) since one transistor does the opposite of the other, thus
  2. there isn't much "leakage" (or **static** consumption) at other times.

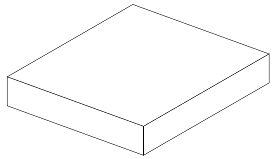
Notes:

Manufacture (1)

Algorithm

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semi-conductor.
3. Apply a layer of **photoresist** material.
4. Expose the photoresist to a precise negative or **mask** of design; this hardens the exposed photoresist.
5. Wash away unhardened photoresist.
6. Etch away uncovered substrate.
7. Strip hardened photoresist.

Example



- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90nm CMOS) relates to the resolution of this process, e.g., width of wires or density of transistors.

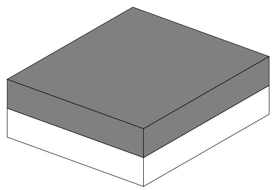
Notes:

Manufacture (1)

Algorithm

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semi-conductor.
3. Apply a layer of **photoresist** material.
4. Expose the photoresist to a precise negative or **mask** of design; this hardens the exposed photoresist.
5. Wash away unhardened photoresist.
6. Etch away uncovered substrate.
7. Strip hardened photoresist.

Example



- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90nm CMOS) relates to the resolution of this process, e.g., width of wires or density of transistors.

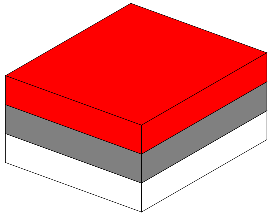
Notes:

Manufacture (1)

Algorithm

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semi-conductor.
3. Apply a layer of **photoresist** material.
4. Expose the photoresist to a precise negative or **mask** of design; this hardens the exposed photoresist.
5. Wash away unhardened photoresist.
6. Etch away uncovered substrate.
7. Strip hardened photoresist.

Example



- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90nm CMOS) relates to the resolution of this process, e.g., width of wires or density of transistors.

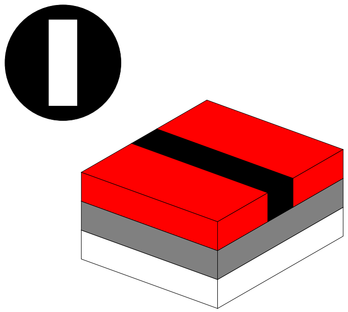
Notes:

Manufacture (1)

Algorithm

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semi-conductor.
3. Apply a layer of **photoresist** material.
4. Expose the photoresist to a precise negative or **mask** of design; this hardens the exposed photoresist.
5. Wash away unhardened photoresist.
6. Etch away uncovered substrate.
7. Strip hardened photoresist.

Example



- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90nm CMOS) relates to the resolution of this process, e.g., width of wires or density of transistors.

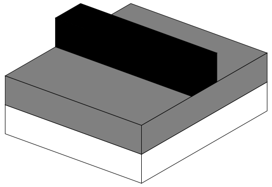
Notes:

Manufacture (1)

Algorithm

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semi-conductor.
3. Apply a layer of **photoresist** material.
4. Expose the photoresist to a precise negative or **mask** of design; this hardens the exposed photoresist.
5. Wash away unhardened photoresist.
6. Etch away uncovered substrate.
7. Strip hardened photoresist.

Example



- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90nm CMOS) relates to the resolution of this process, e.g., width of wires or density of transistors.

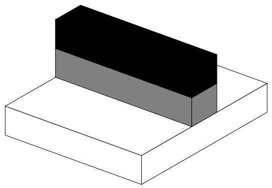
Notes:

Manufacture (1)

Algorithm

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semi-conductor.
3. Apply a layer of **photoresist** material.
4. Expose the photoresist to a precise negative or **mask** of design; this hardens the exposed photoresist.
5. Wash away unhardened photoresist.
6. Etch away uncovered substrate.
7. Strip hardened photoresist.

Example



- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90nm CMOS) relates to the resolution of this process, e.g., width of wires or density of transistors.

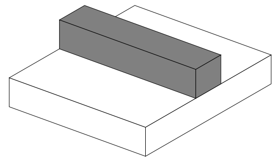
Notes:

Manufacture (1)

Algorithm

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semi-conductor.
3. Apply a layer of **photoresist** material.
4. Expose the photoresist to a precise negative or **mask** of design; this hardens the exposed photoresist.
5. Wash away unhardened photoresist.
6. Etch away uncovered substrate.
7. Strip hardened photoresist.

Example

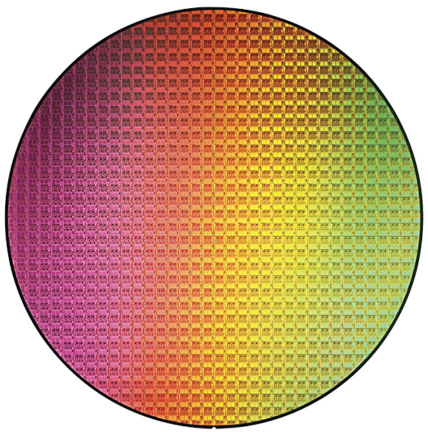


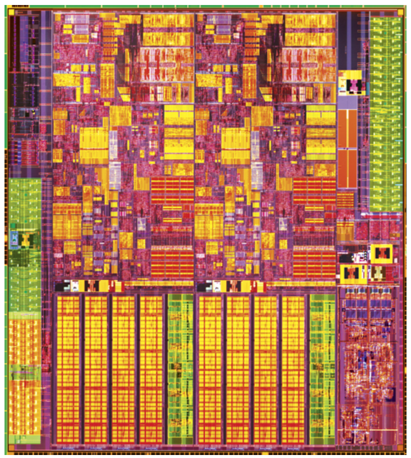
- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90nm CMOS) relates to the resolution of this process, e.g., width of wires or density of transistors.

Notes:

Manufacture (2)

- The end result (left-hand side) is a wafer where ...





- ... each "block" is an individual component, e.g., a Westmere model Intel processor (right-hand side).

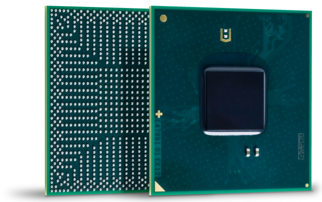
---

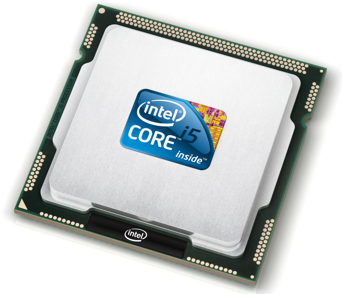
http://www.intel.com/pressroom/archive/releases/2010/20100107comp_sm.htm

Notes:

Manufacture (3)

- The components aren't much use in this form; they are **packaged** before use:





- Note that the packaging acts as
  - protection against damage, often including a heat sink as well, *and*
  - an interface between the component and the outside world, using external pins bonded to internal inputs and outputs.

http://www.intel.com/pressroom/archive/releases/2010/20100107comp_sm.htm

Notes:

Manufacture (4)

Moore's Law: originally an observation

> *The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000.*
>
> – *Moore* [7], 1965

and later updated: in short "number of transistors in unit area doubles roughly every two years".

http://download.intel.com/pressroom/images/events/moores_law_40th/Gordon_Moore/GordonMoore_young.jpg

Notes:

Example (Moore's Law [7], from 1970 to 2005)

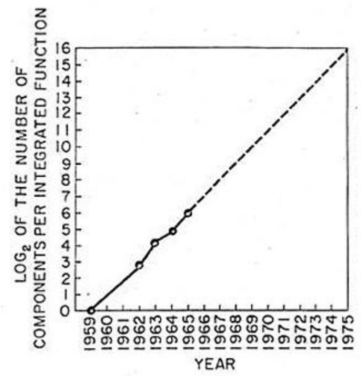
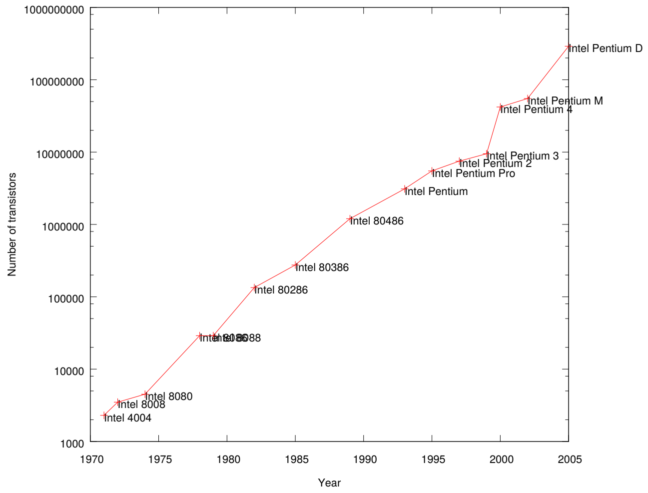


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

Notes:

Manufacture (5)

Example (Moore's Law [7], from 1970 to 2005)



Notes:

Conclusions

Notes:

- Take away points:
  1. Semi-conductor and transistor design and manufacturing technology lurks in the background of almost *everything* we do.
  2. Various implications of modern transistor design place tangible constraints on us: at least a high-level understanding of these (e.g., Moore's Law etc.) is valuable.
  3. The underlying Physics *is* fairly complicated ...
  4. ... but, at the right level of abstraction/detail, the concept *isn't*: fundamentally, transistors are *just* switches.

Additional Reading

Notes:

- *Wikipedia: Transistor*. URL: http://en.wikipedia.org/wiki/Transistor.
- D. Page. "Chapter 2: Basics of digital logic". In: *A Practical Introduction to Computer Architecture*. 1st ed. Springer-Verlag, 2009.
- R.J. Smith and R.C. Dorf. "Chapter 12: Transistors and Integrated Circuits". In: *Circuits, Devices and Systems*. 5th ed. John Wiley, 1992.
- W. Stallings. "Chapter 11: Digital logic". In: *Computer Organisation and Architecture*. 9th ed. Prentice-Hall, 2013.
- A.S. Tanenbaum and T. Austin. "Section 3.1: Gates and Boolean algebra". In: *Structured Computer Organisation*. 6th ed. Prentice-Hall, 2012.
- A.S. Tanenbaum and T. Austin. "Section 3.2.1: Integrated circuits". In: *Structured Computer Organisation*. 6th ed. Prentice-Hall, 2012.

References

[1] *Wikipedia: Transistor*. URL: http://en.wikipedia.org/wiki/Transistor (see p. 55).

[2] D. Page. "Chapter 2: Basics of digital logic". In: *A Practical Introduction to Computer Architecture*. 1st ed. Springer-Verlag, 2009 (see p. 55).

[3] R.J. Smith and R.C. Dorf. "Chapter 12: Transistors and Integrated Circuits". In: *Circuits, Devices and Systems*. 5th ed. John Wiley, 1992 (see pp. 23, 55).

[4] W. Stallings. "Chapter 11: Digital logic". In: *Computer Organisation and Architecture*. 9th ed. Prentice-Hall, 2013 (see p. 55).

[5] A.S. Tanenbaum and T. Austin. "Section 3.1: Gates and Boolean algebra". In: *Structured Computer Organisation*. 6th ed. Prentice-Hall, 2012 (see p. 55).

[6] A.S. Tanenbaum and T. Austin. "Section 3.2.1: Integrated circuits". In: *Structured Computer Organisation*. 6th ed. Prentice-Hall, 2012 (see p. 55).

[7] G.E. Moore. "Cramming more components onto integrated circuits". In: *Electronics Magazine* 38.8 (1965), pp. 114–117 (see pp. 47, 49, 51).

Notes: