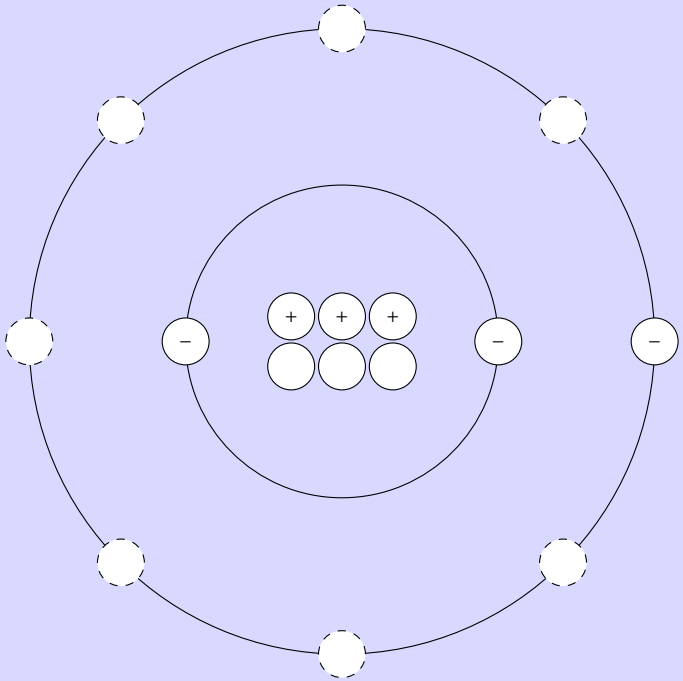
COMS12200 lecture: week #3

- The concrete implementation of most digital logic boils down use of **switches**: if the switch is "on" it conducts electricity, if it's "off" 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.

"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**.

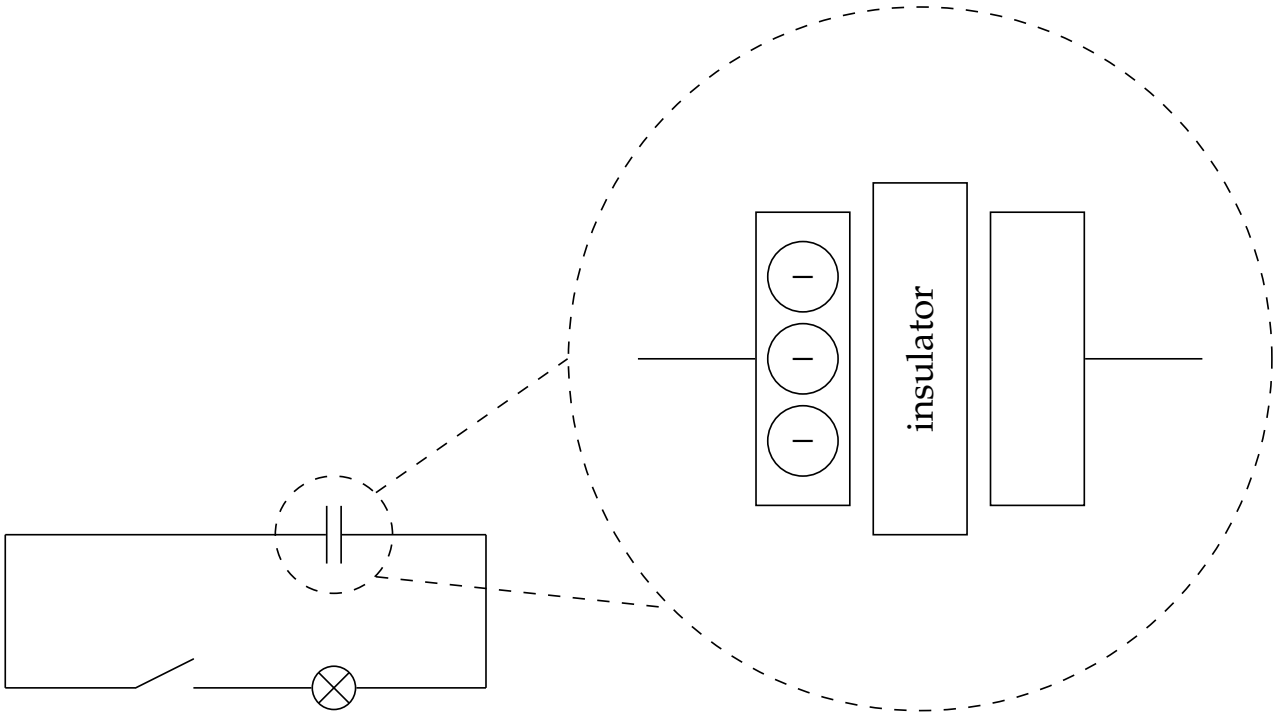
"A-level Physics in 10 minutes" (2)

- 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.
- The particles are bound together in a variety of ways ...
- ... but the 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.

"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.

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

- 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

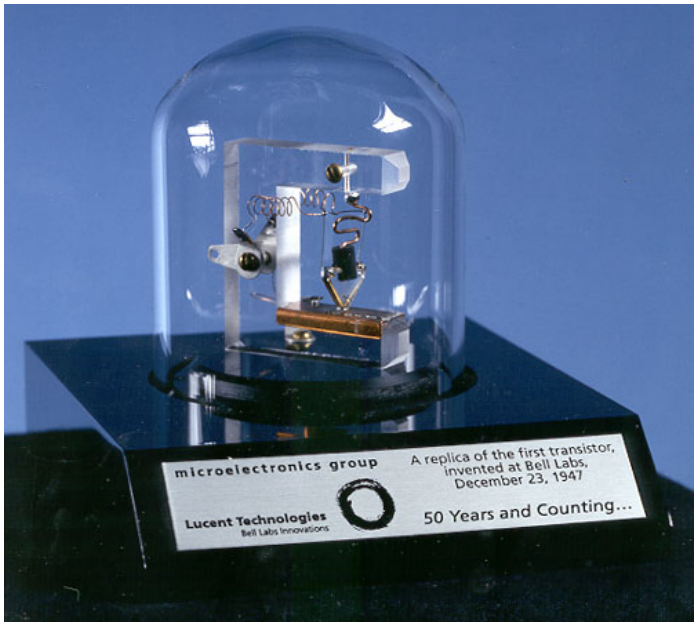
  which, roughly speaking, allows us to construct materials with the exact sub-atomic properties we want.
- The result is a *semi*conductor:
  - A P-type semiconductor has extra holes, N-type has extra electrons.
  - If we sandwich P-type and N-type layers together, electrons can only move in *one* way,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

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:5651RegulatorTubeInOperation.jpg

An Aside: History



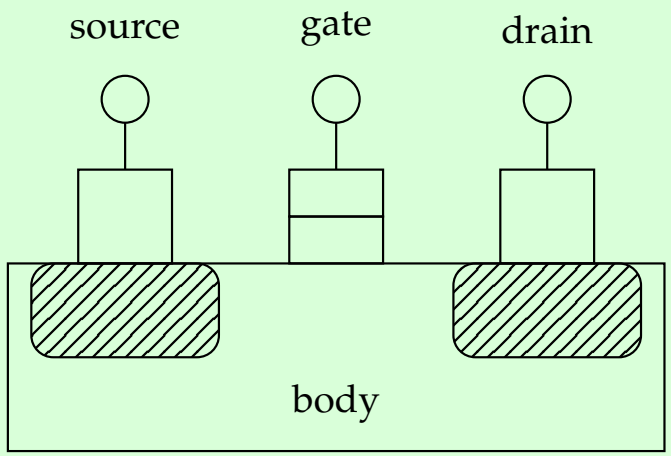
- The replacement for this generation of technology is the **transistor**.
- There are *many* types, but a potted overview of the one we'll focus on is:
  - 1925: **Field Effect Transistor (FET)**, initially designed and patented by Julius Lilienfeld ...
  - ... design not really used, due to lack of 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 Semiconductor FET (MOSFET)**.

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

Switches and Transistors (1)

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

Circuit (MOSFET)

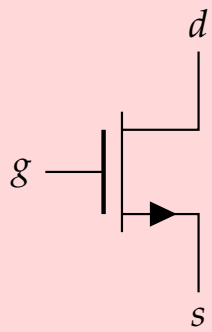


- The (very) rough idea is ...
  - Apply a potential difference between gate and source; this controls the size of a **depletion region** between the source and drain.
  - A large potential difference means a small depletion region, allowing electrons, i.e., current, to flow in a **channel** that links the source and drain.
  - A small potential difference (smaller than some **threshold**) means a large depletion region which closes the channel.
- ... a more detailed description needs more Physics!

Switches and Transistors (2)

Definition (**N-MOSFET**)

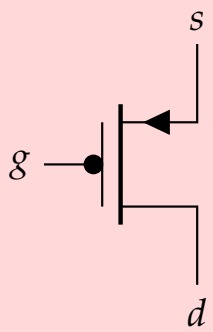
- Written symbolically as



  where d, s and g denote the drain, source and gate terminals.
- Built using **N-type** semiconductor terminals and **P-type** body.
- A positive potential difference between source and gate widens the depletion channel, meaning source and drain are connected.
- A negative potential difference between source and gate narrows the depletion channel, meaning source and drain are disconnected.

Definition (**P-MOSFET**)

- Written symbolically as

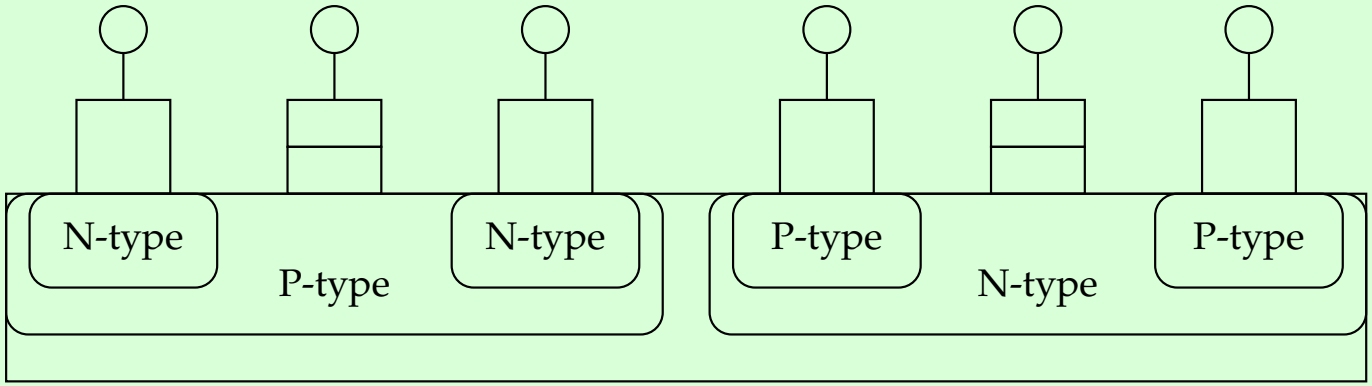


  where s, d and g denote the source, drain and gate terminals.
- Built using **P-type** semiconductor terminals and **N-type** body.
- A positive potential difference between source and gate narrows the depletion channel, meaning source and drain are disconnected.
- A negative potential difference between source and gate widens the depletion channel, meaning source and drain are connected.

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 complementary way:

Circuit (CMOS cell)



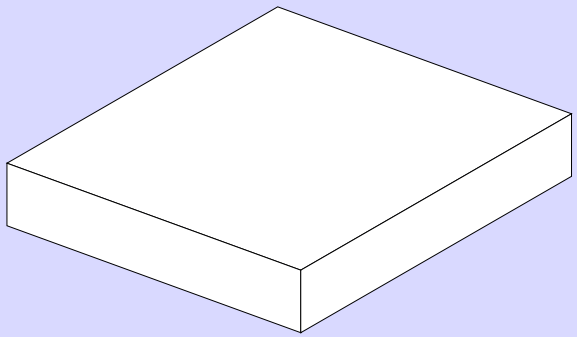
- We'll see later why this is functionally useful, but behaviourally
  1. only switching from one state to another consumes much power (or **dynamic consumption**) since one transistor does the opposite of the other, thus
  2. there isn't much "leakage" (or **static** consumption) at other times.

Manufacture (1)

Algorithm (photolithography)

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semiconductor.
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 (photolithography)



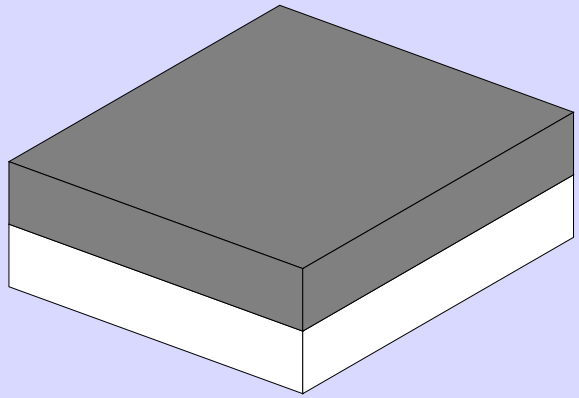
- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90 nm CMOS) relates to the resolution of this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

Manufacture (1)

Algorithm (photolithography)

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semiconductor.
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 (photolithography)



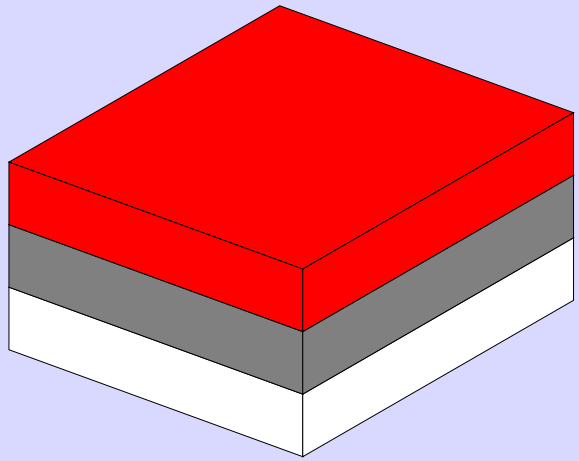
- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90 nm CMOS) relates to the resolution of this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

Manufacture (1)

Algorithm (photolithography)

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semiconductor.
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 (photolithography)



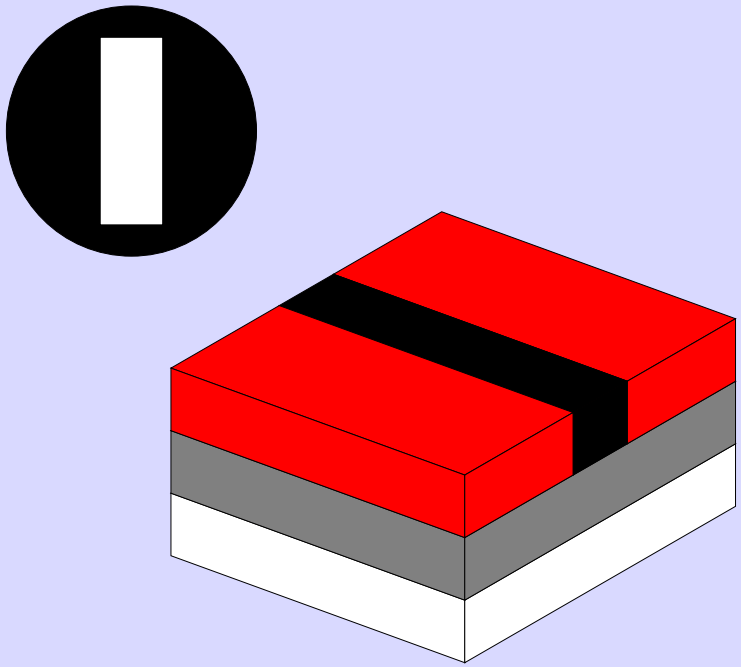
- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90 nm CMOS) relates to the resolution of this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

Manufacture (1)

Algorithm (photolithography)

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semiconductor.
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 (photolithography)



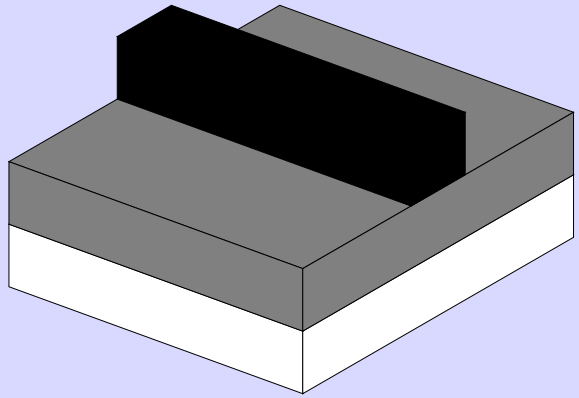
- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90 nm CMOS) relates to the resolution of this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

Manufacture (1)

Algorithm (photolithography)

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semiconductor.
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 (photolithography)



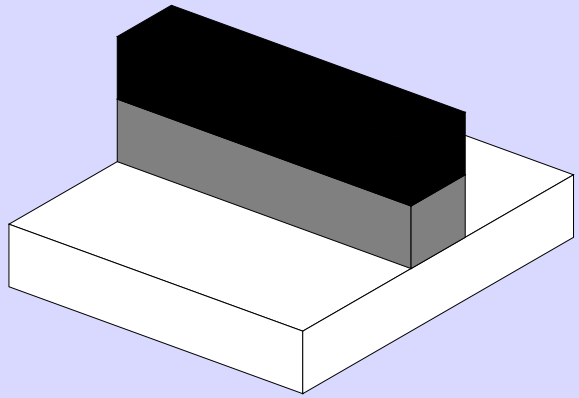
- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90 nm CMOS) relates to the resolution of this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

Manufacture (1)

Algorithm (photolithography)

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semiconductor.
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 (photolithography)



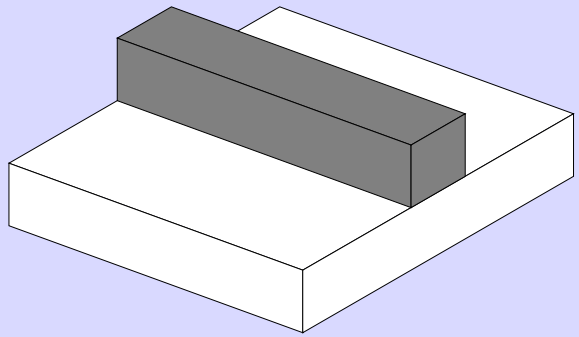
- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90 nm CMOS) relates to the resolution of this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

Manufacture (1)

Algorithm (photolithography)

1. Start with a clean, prepared **wafer**.
2. Apply a layer of **substrate** material, e.g., metal or semiconductor.
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.

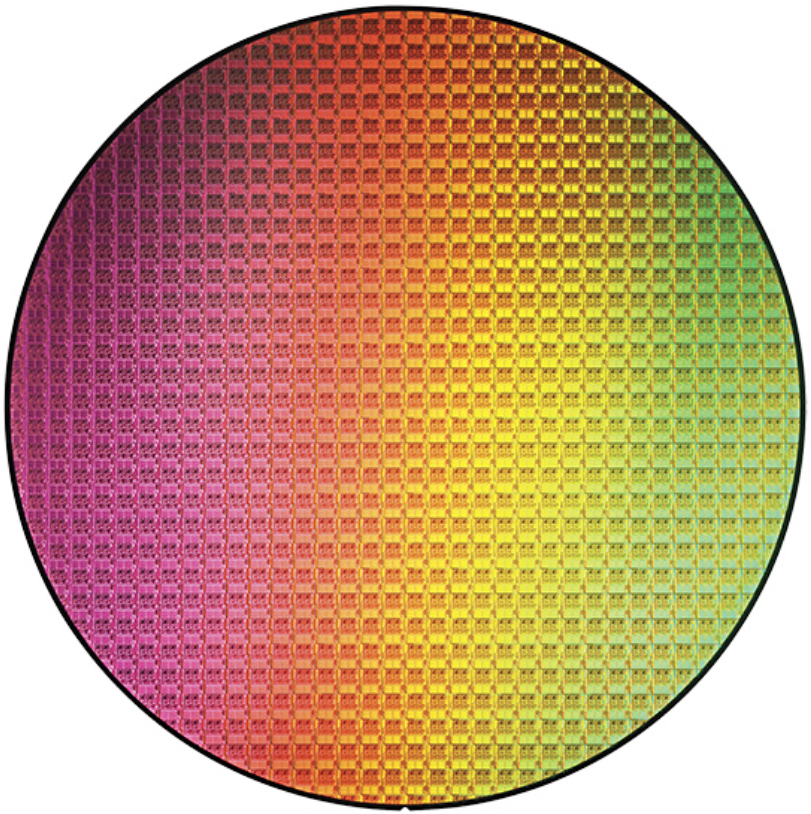
Example (photolithography)

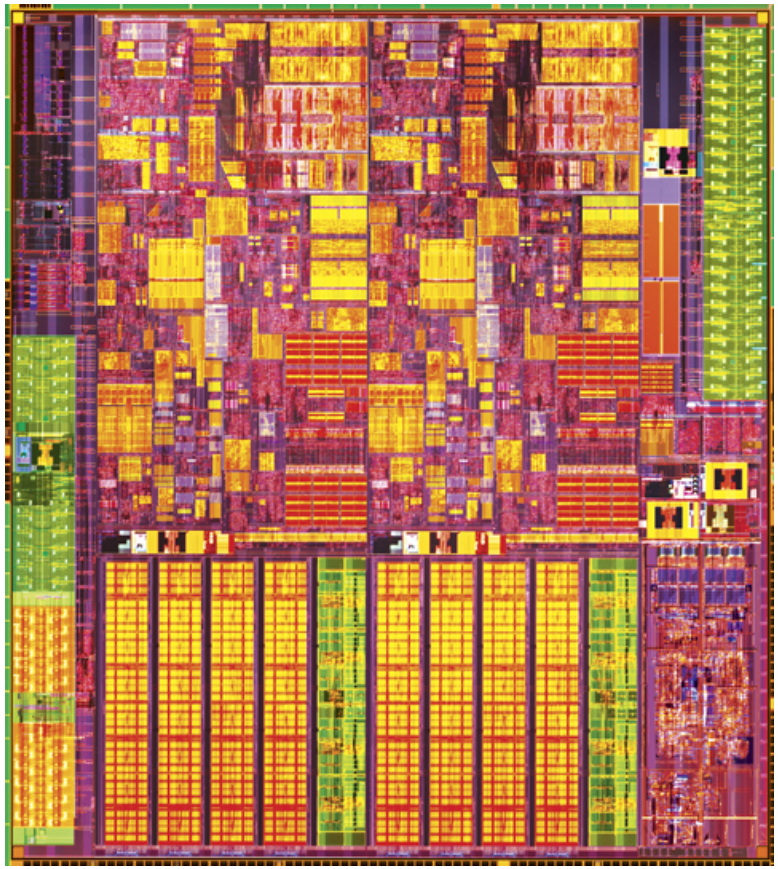


- Note that:
  - The algorithm iterates to produce many layers, i.e., the result is 3D not 2D.
  - Regularity is a major advantage: we can manufacture *many* similar components in a layer using *one* photolithographic process.
  - The feature size (e.g., 90 nm CMOS) relates to the resolution of this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

Manufacture (2)

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





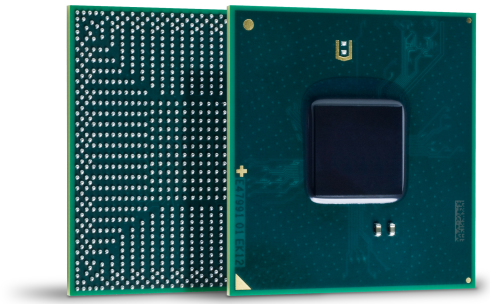
- ... each "block" is an individual component, e.g., a Westmere model Intel processor (right-hand side).

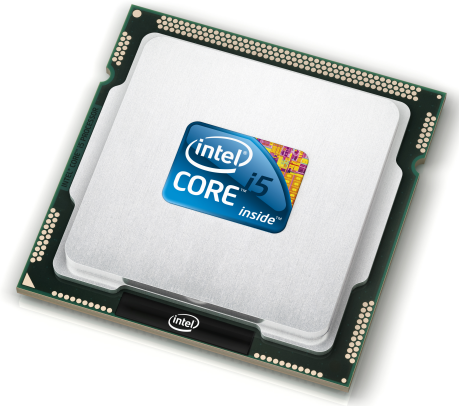
---

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

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

Manufacture (4)

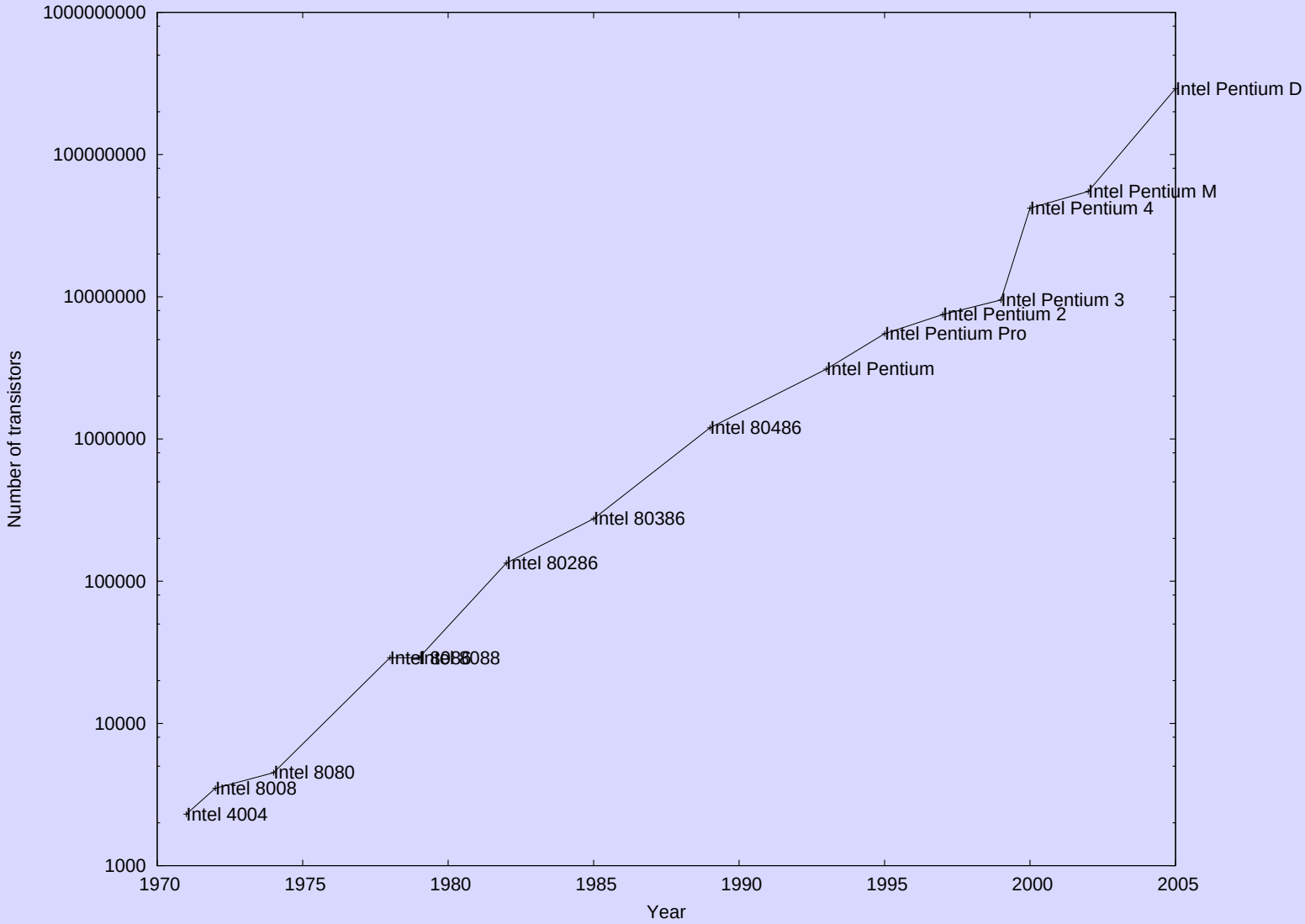
Moore's Law: originally an observation

> *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 [5]*, 1965

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

Manufacture (5)

Example (Moore's law, from 1970 to 2005)



Manufacture (6)

- Great: with smaller transistors we can
  1. produce more complicated designs that use more transistors, or
  2. shrink existing designs, making them smaller and faster

  without really thinking about anything!
- But, there's no "free lunch" because (as Moore notes)
  1. **power consumption** and **heat dissipation** become more and more important, and
  2. process variation, which may imply **defects** and reduce yield, starts to increase

  so *smarter* design techniques are crucially important.

Conclusions

- Take away points:
  1. Semiconductor and transistor design and manufacturing technology lurks in the background of almost *everything* we do ...
  2. ... transistors *are* fairly complicated wrt. Physics, but from a conceptual perspective they're *just* switches.
  3. 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.
- Don't panic!
  - A reasonable comment is "WTF?! I thought this was CS, not EE!".
  - The point is to now abstract away the detail:
    - A micro-processor, at a low-level, is just a complicated arrangement of transistors (i.e., switches).
    - Now we know how transistors work, we can start to focus more on their function and less on their form.
    - That is, we'll view them as "black boxes" and build high-level components on top of the functionality they afford.

References and Further Reading

[1] Wikipedia: Hydraulic analogy.
http://en.wikipedia.org/wiki/Hydraulic_analogy.

[2] Wikipedia: Moore's law.
http://en.wikipedia.org/wiki/Moore's_law.

[3] Wikipedia: Semiconductor device fabrication.
http://en.wikipedia.org/wiki/Semiconductor_fabrication.

[4] Wikipedia: Transistor.
http://en.wikipedia.org/wiki/Transistor.

[5] G.E. Moore.
Cramming more components onto integrated circuits.
Electronics Magazine, 38(8):114–117, 1965.

[6] D. Page.
Chapter 2: Basics of digital logic.
In *A Practical Introduction to Computer Architecture*. Springer-Verlag, 1st edition, 2009.

[7] W. Stallings.
Chapter 11: Digital logic.
In *Computer Organisation and Architecture*. Prentice-Hall, 9th edition, 2013.

References and Further Reading

[8] A.S. Tanenbaum.
Section 3.1: Gates and Boolean algebra.
In *Structured Computer Organisation* [10].

[9] A.S. Tanenbaum.
Section 3.2: Basic digital logic circuits.
In *Structured Computer Organisation* [10].

[10] A.S. Tanenbaum.
Structured Computer Organisation.
Prentice-Hall, 6th edition, 2012.