ELEC-H-473

Microprocessor architectures

Lecture 01

Dragomir Milojevic
dmilojev@ulb.ac.be

General information

1. Agenda

Lectures
2 ECTS = 12 sessions, 2h/session
Monday → from 10.00 to 12.00 (C3.122); CONFLICT 2B solved !

Friday → from 08.00 to 10.00 (H.2213)

TPs
3 ECTS
Monday → from 14.00 to 18.00 (Solbosch, building U UA5.217)

Friday → cancelled (moved to Monday)

2. ELEC-H-473 Internet resources

http://beams.ulb.ac.be/beams/

login: etudiants, password: SquareG! (it is case sensitive)

Attention: if you do not login you will not even see the notes.

General information

3. Conflict dates

- 3, 7 and 21 March I am travelling (we will organise this)

4. Practical work

- Presence is mandatory !
- Mini-projects to be implemented; each project to be presented (you have to show the working demo); Q&A are part of the evaluation
- Practical work account for 45% of the final mark

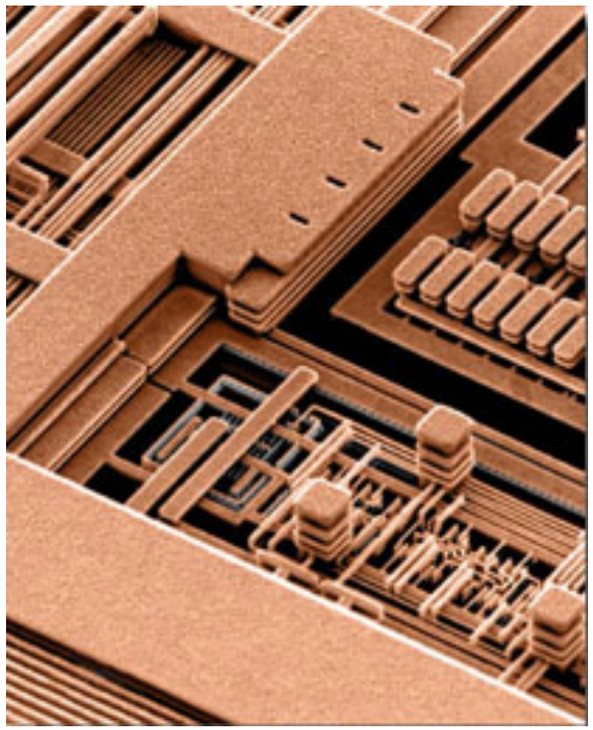
5. Examen

- Is oral
- Most of the questions are theoretical but some of the questions could be closely related to the practical work
- You are expected not only to show the lecture content (copy slides), but be able to reason on the matter

Today

1. Tale on computing machines
2. IC manufacturing technology perspective
3. Computing systems performance
4. Example of poor usage : data centers
5. What could happen in the future ?

The lecture starts with a tale on computing machines ...



… how they are made and

… and how to push their limits ...

Do u know from where does this comes from?

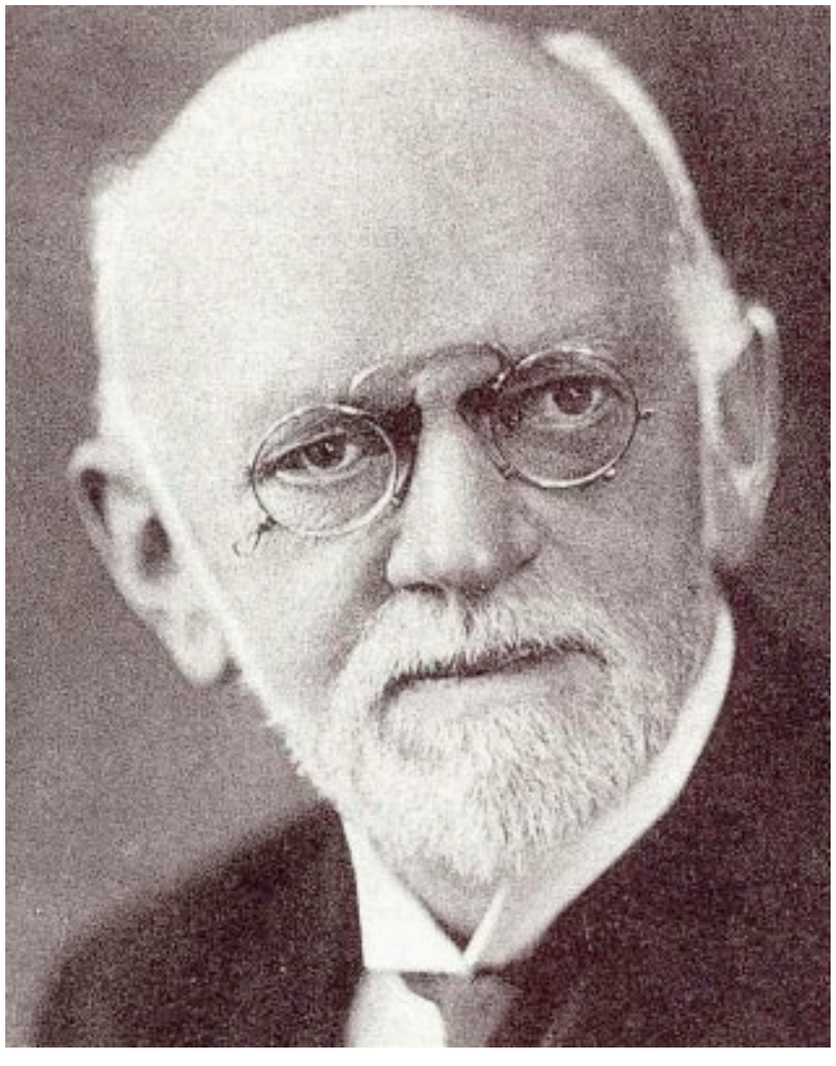
So once upon a time ...

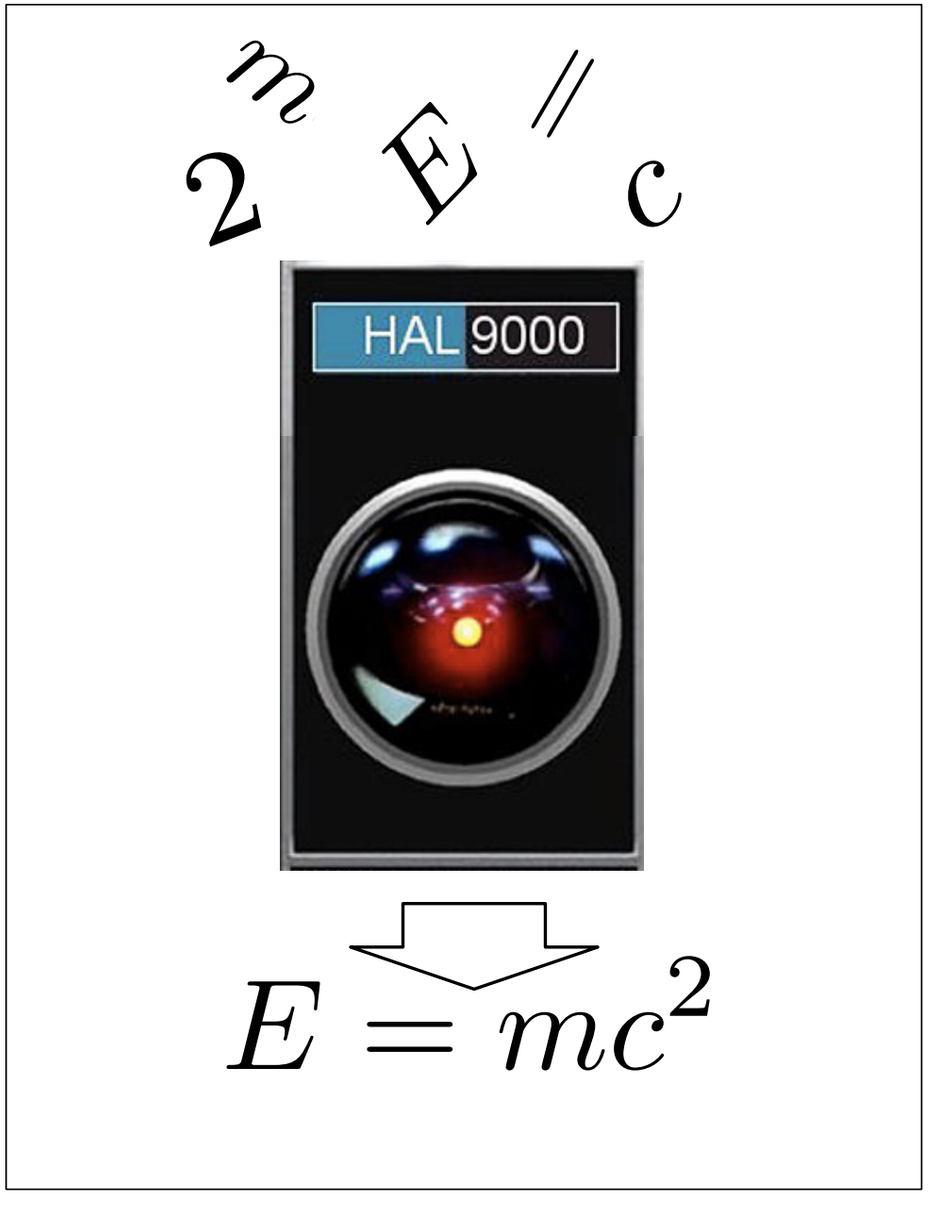
… there was a mathematician

that prepared a **BIG** question for XXth century:

"*Could maths be automatized?*"

David Hilbert, 1900

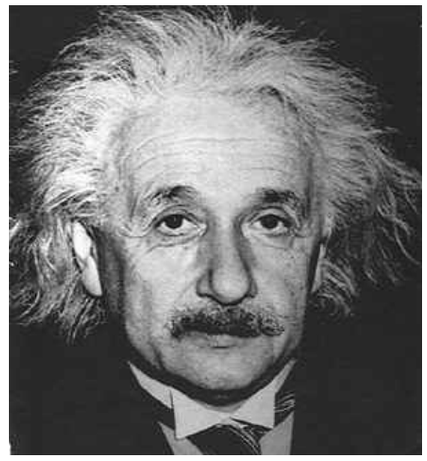




BIG QUESTION got an answer: **BIG NO**!

Kurt Gödel, 1931

$$2^{m} E // c$$

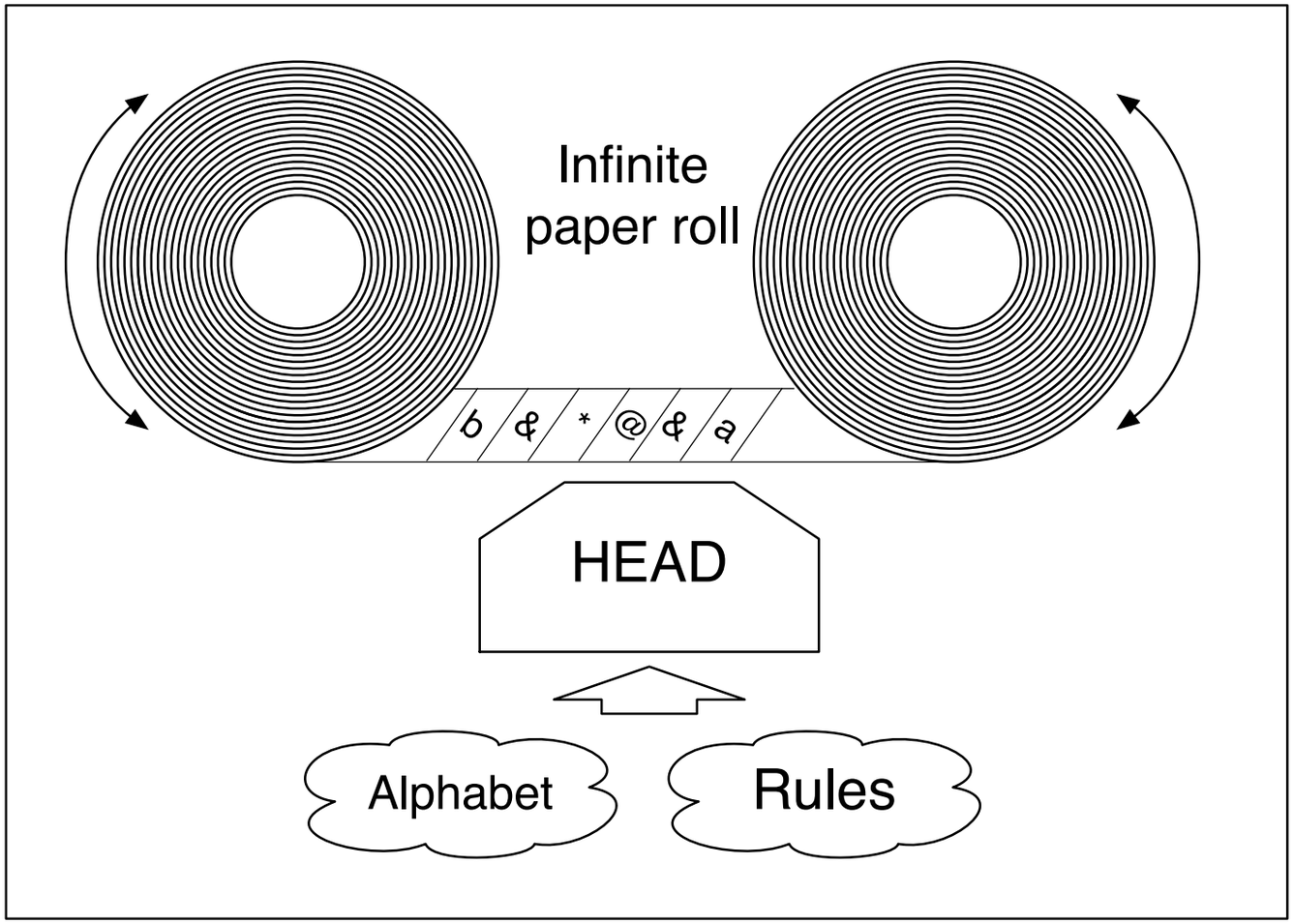


$$E = mc^2$$

What can machines compute then?

Alain Turing, 1936

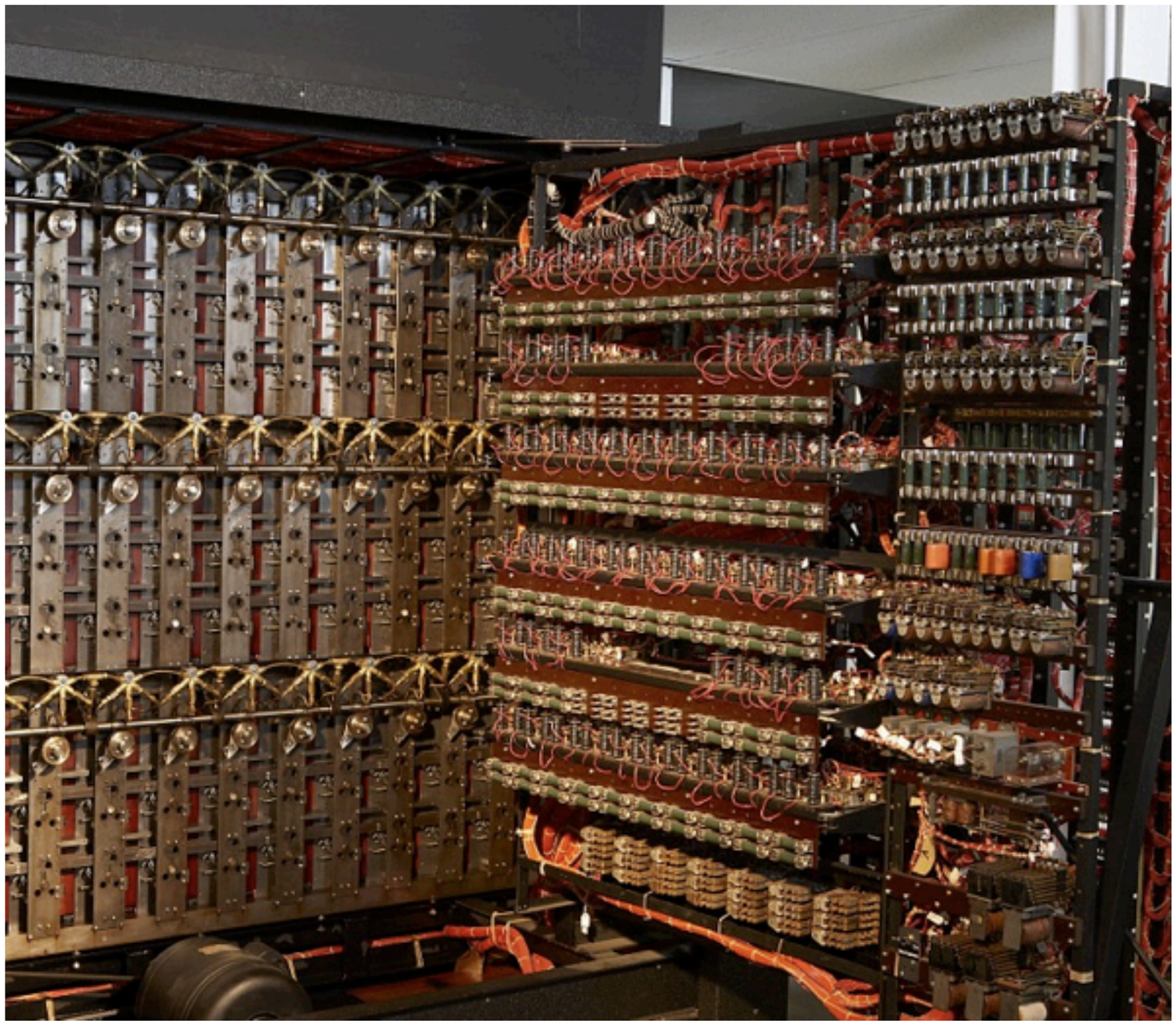
Anything that can be computed with a Turing machine !



Turing machine : conceptual but also real !

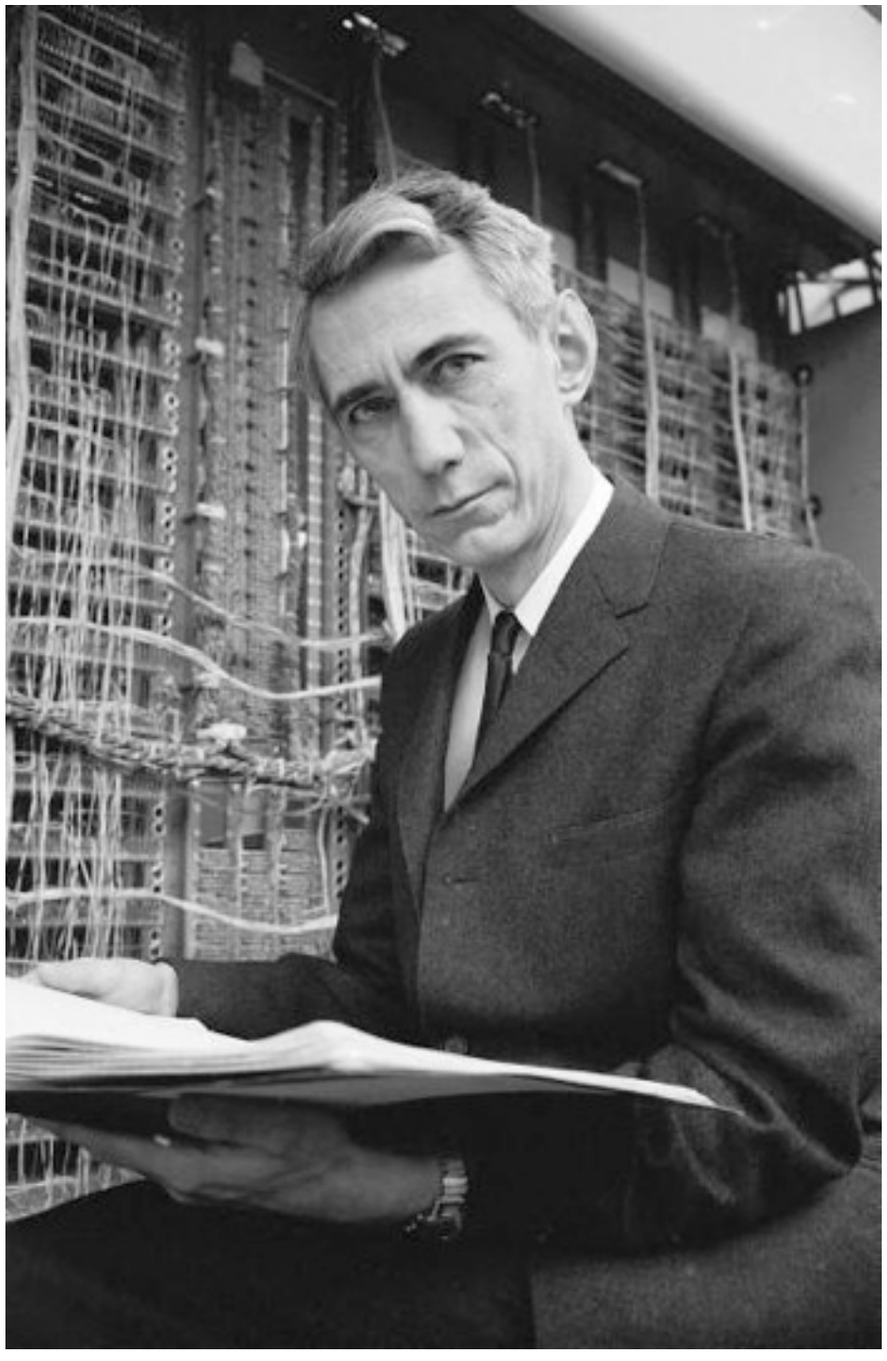
Enigma, 1936

The Bomb, 1940



Mechanics are not the best medium !

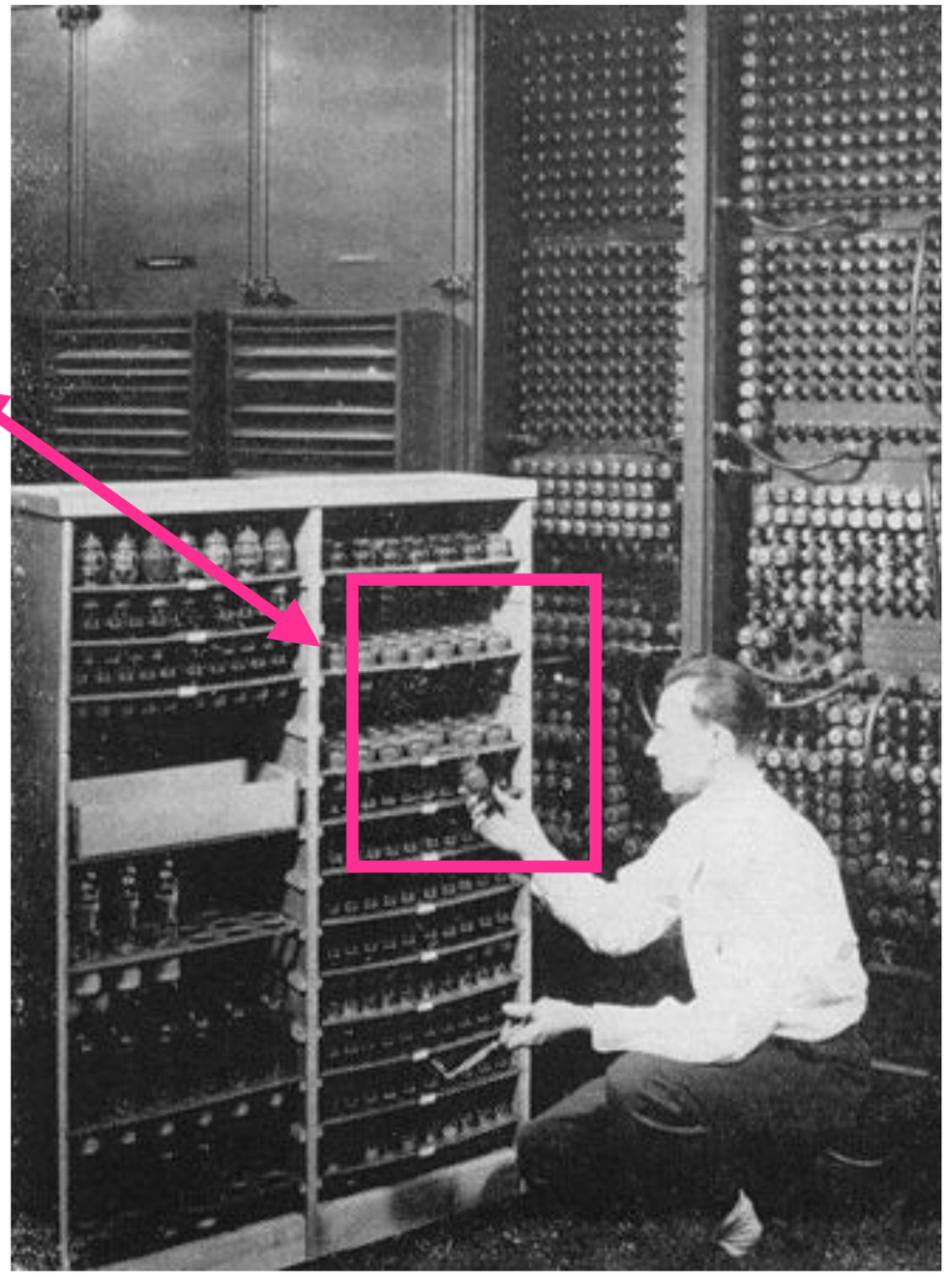
Claude E. Shannon, 1937



… but electric switches are, for sure !

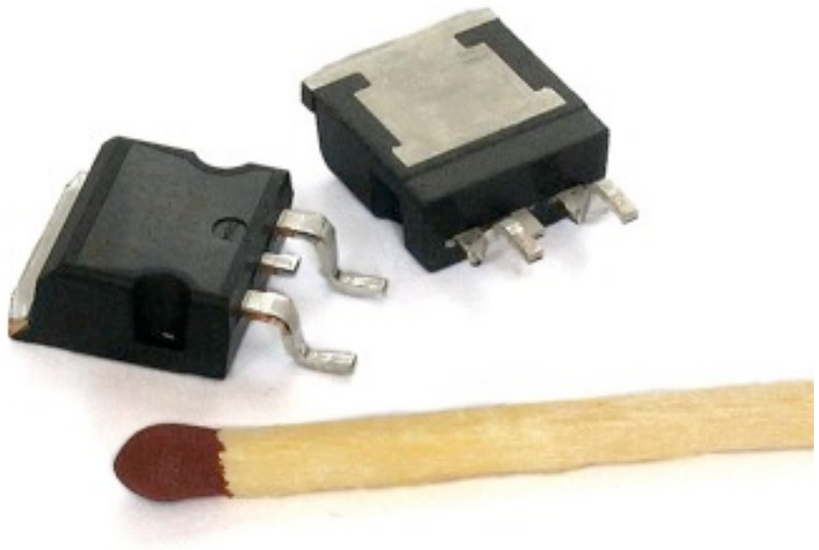
How to make a usable switch?

ENIAC, 1944

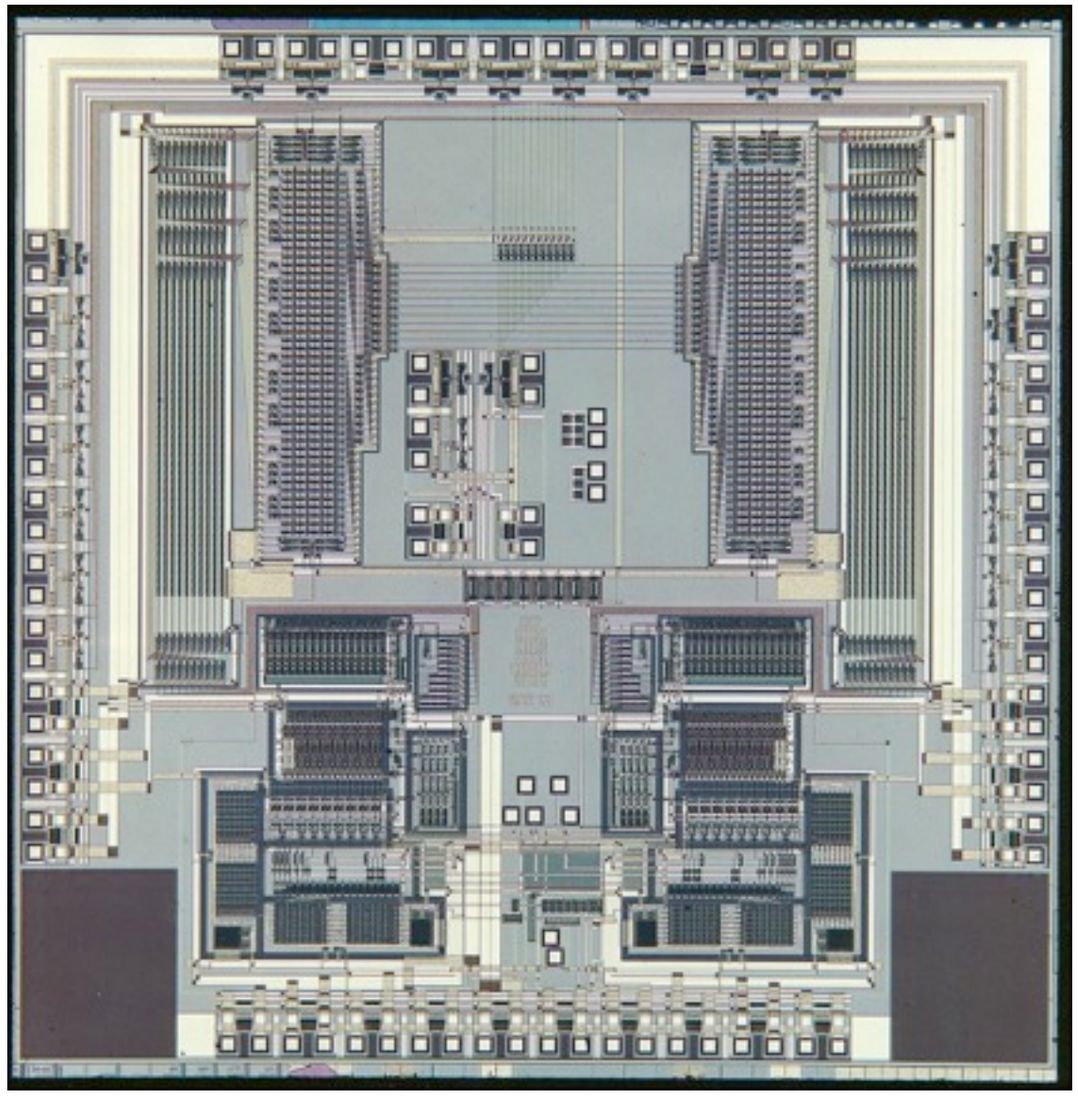


... but the size does meter !

Transistor, 1947



Integrated Circuit, 1958



1cm

AND

THE SCALING

WAS BORN !!!

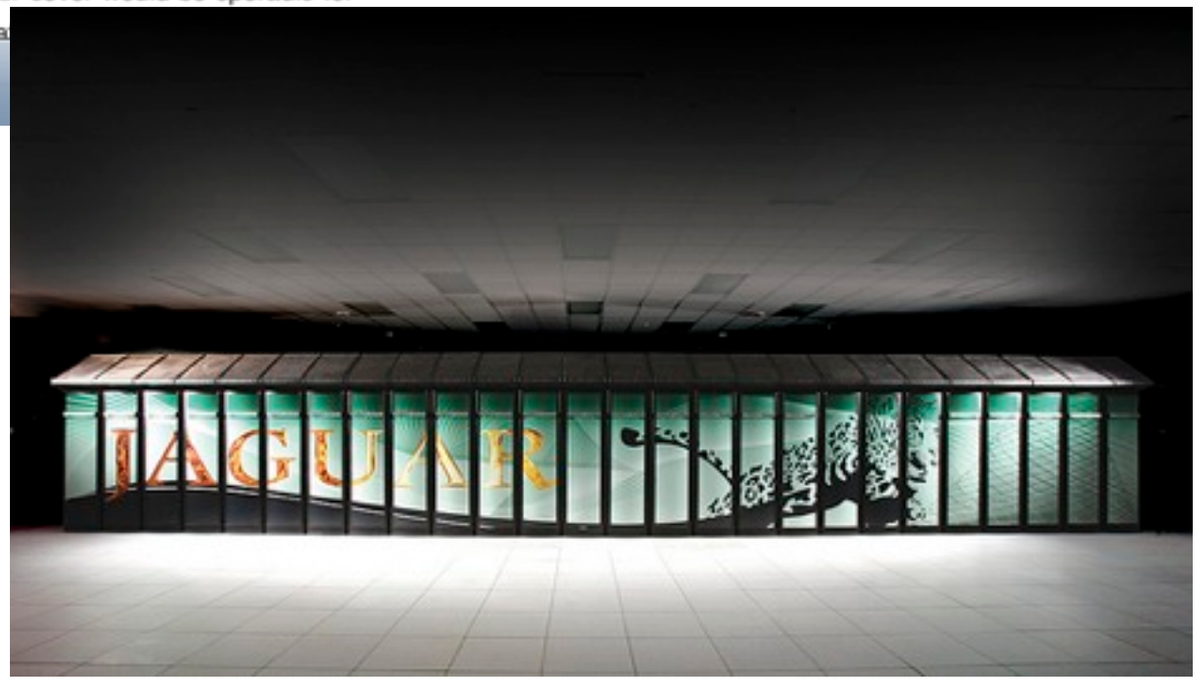
So, in the ’80 ...

ZX81

1kB RAM
machine

… and today:

Mobile
Encyclopedia
or
2,5 Penta FLOPS
in a big room

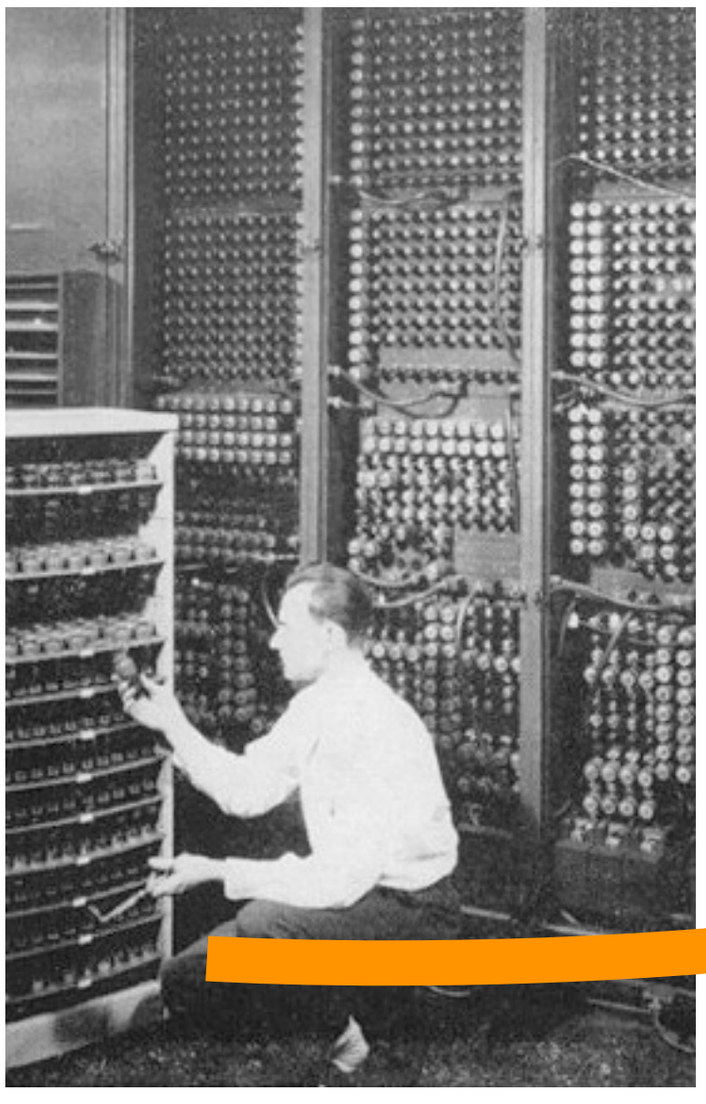


What is scaling?

MacBook pro, 2011

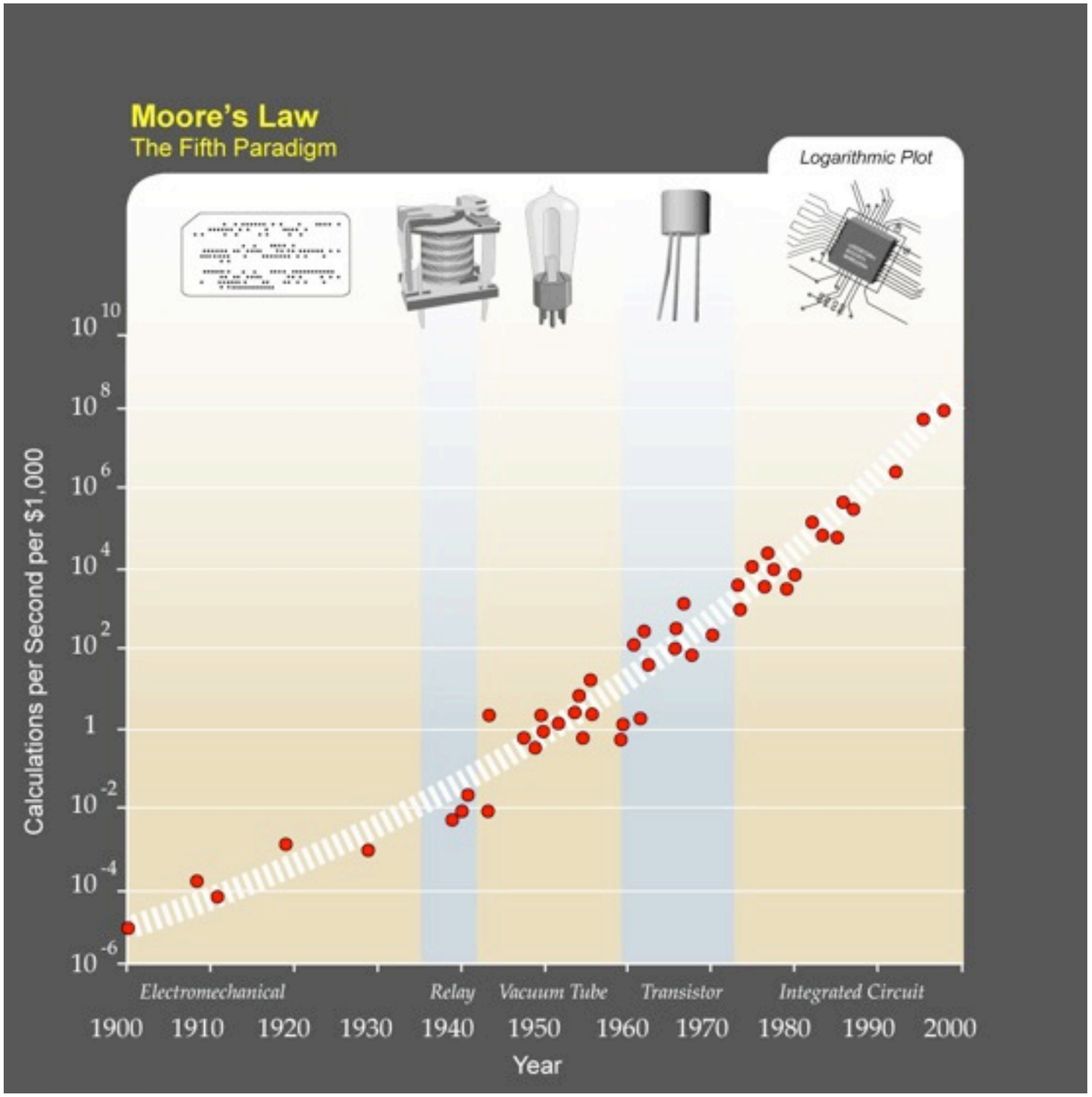
IBM XT, 1983

ENIAC, 1947

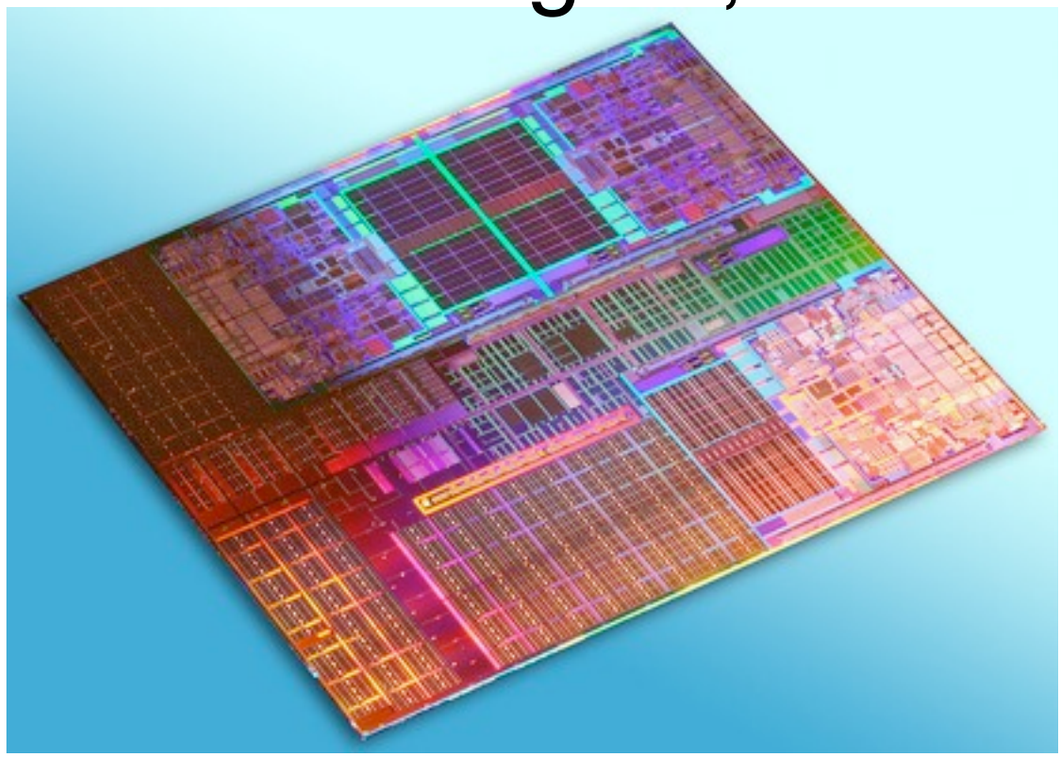


Tendency, transformed in a law...

Scaling : Moore's law and state of the art



Intel Dunnington, 2008



6 processors on the same die

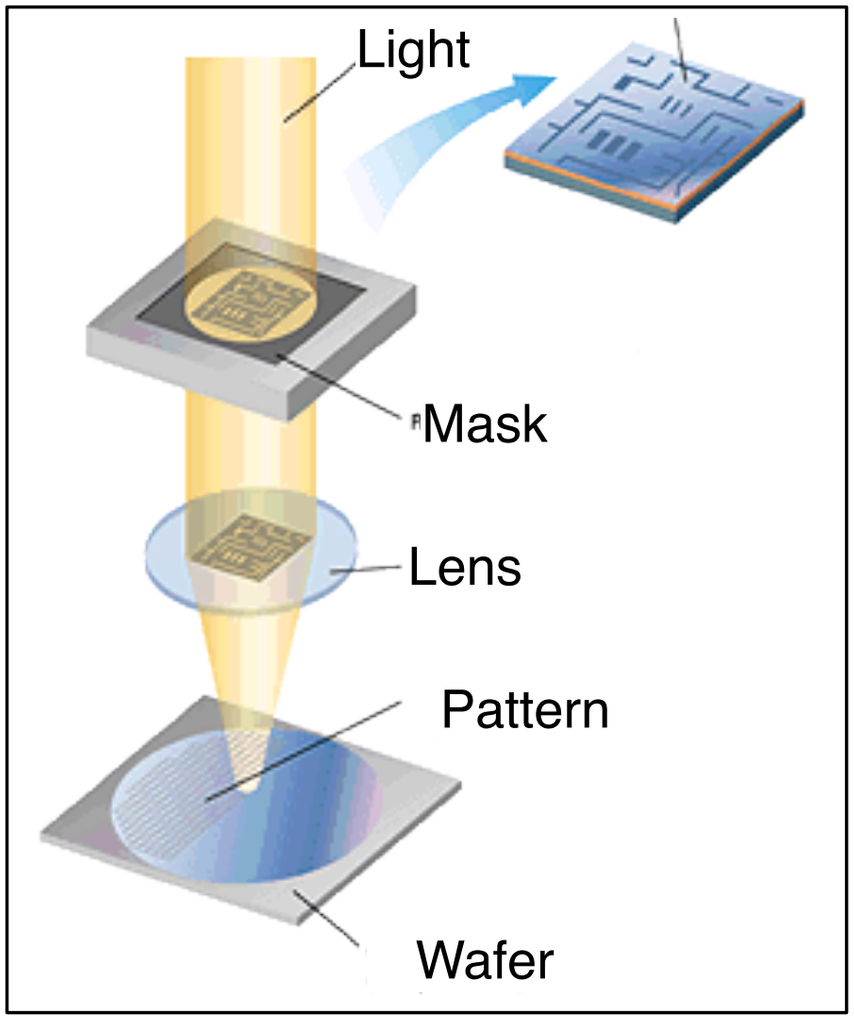
2 billion transistors

1 cm^2

... if only the car industry did the same …

Speed	180.000.000 km/h
Fuel	0,04 l/100km
Price	0,0003$

Technology scaling: the sky is the limit?



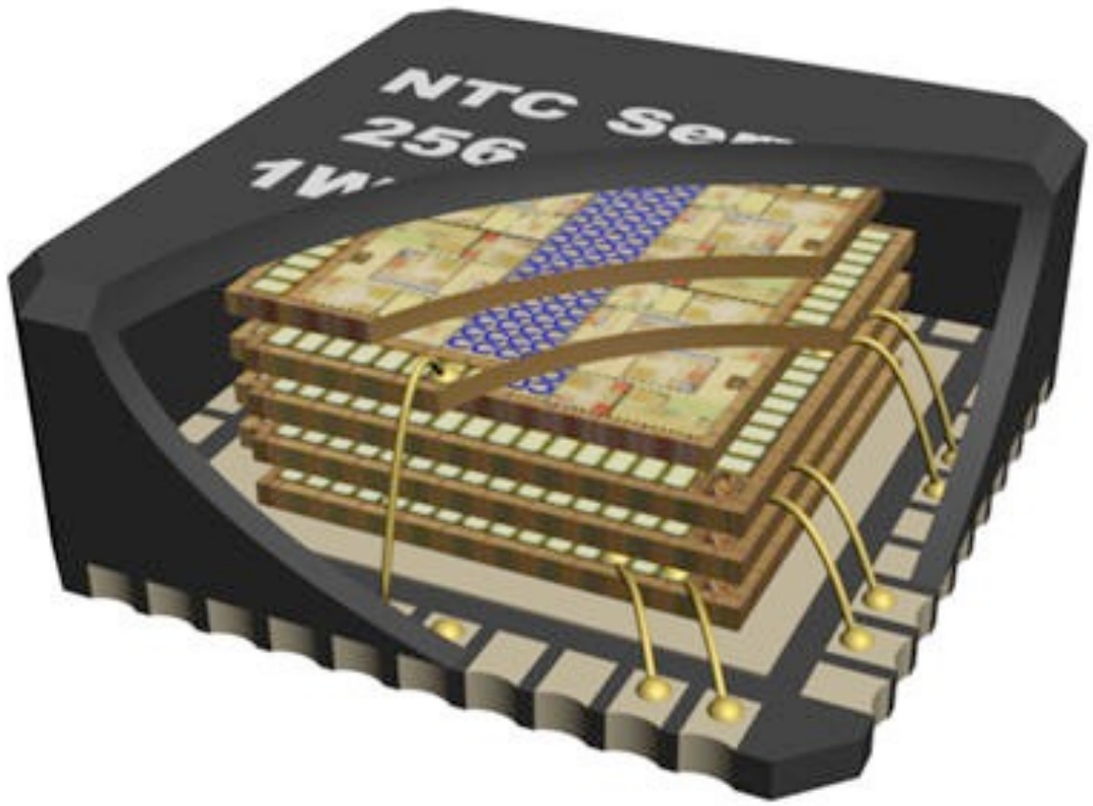
No, but the “size” of the light IS!

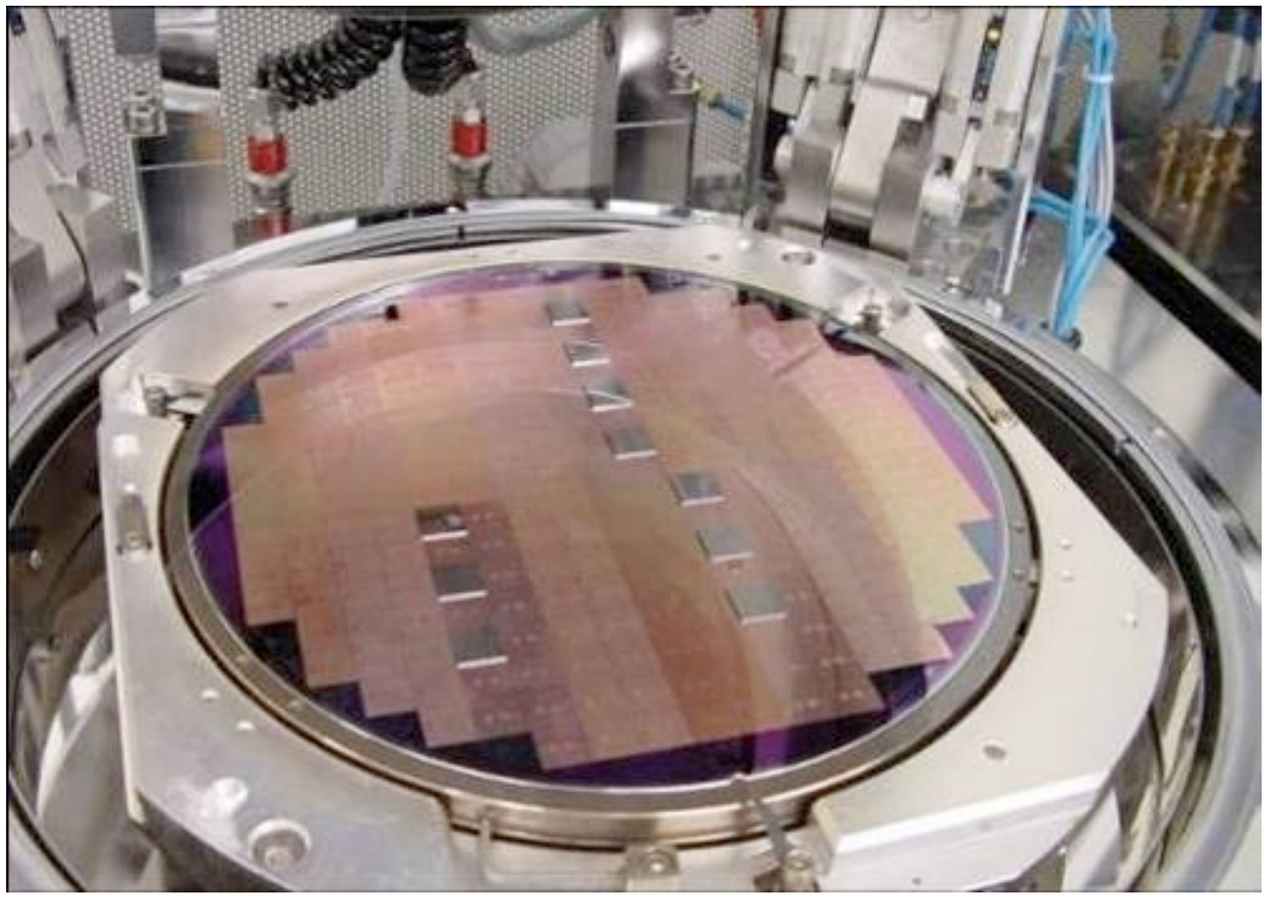
Potential end in 2020 ?

What to do next ?

Go for a non-exploited dimension

3D Circuits, 2010



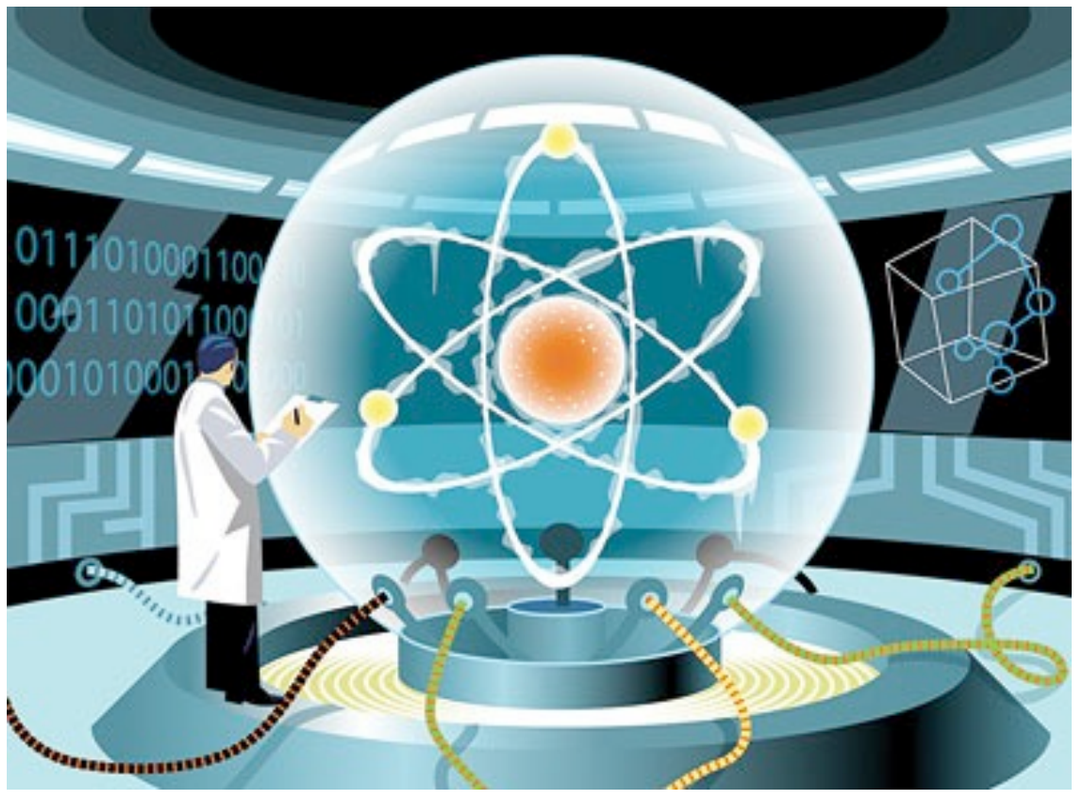


But, even if this solution sound fantastic, it is JUST to push the limits A BIT FURTHER AWAY, for next couple of years (u r concerned)

But what about the far future?

Optical computing, 2???

Quantum computing, 2???



Let’s (really) think of the future ...

a) New computational paradigm is **LESS** engineering problem and **MORE** fundamental one, as of today

b) Fundamental sciences are not that predictive and require some degree of fussiness ... Just think of what I’ve said in the beginning of this presentation ... and how mathematics led to all this

c) We can’t definitively **PLAN, PROJECT MANAGE and/or PREDICT** the arrival of a let’s say Quantum Computer on May 15th in 2035

and the business view of it ...

Fortune, 2007

	Company	$ millions
SW		
1	Hewlett-Packard	91,658
2	Intl. Business Machines	91,424
3	Dell	57,095
4	Apple	19,315
5	Xerox	15,895
6	Sun Microsystems	13,068
7	NCR	6,142
8	Pitney Bowes	5,811
9	Gateway	3,981
10	Palm	1,579
		305,968
HW		
1	Microsoft	44,282
2	Oracle	14,380
3	Symantec	4,143
4	CA	3,805
5	Electronic Arts	2,951
6	Adobe Systems	2,575
7	Intuit	2,362
		74,498
	Total SW + HW	**380,466**

Computer industry 390.000.000.000$

But profits are less…

even Apple moves to low-cost to get the volume

Conflicting visions

We need to make an important step forward, and the current state of the art says: **THAT WE ARE ABOUT TO HIT THE WALL IN BOTH WORLDS** !!!

Will everything stop because of the lack of gain/or because people would like to go "back to their sources" (i.e. life without computers)?

Questions for XXIth century ...

That we/you need to answer

- Do we want/need better technology for the future?
  → Personally, I would like this to happen ...
- How to motivate/enable fundamental research in this field?
- How to encourage capitalism to become more human friendly and really invest in fundamental research?

After all, didn't it all started as a very romantic and **COMPLETELY un-profitable story**?

Conclusion …

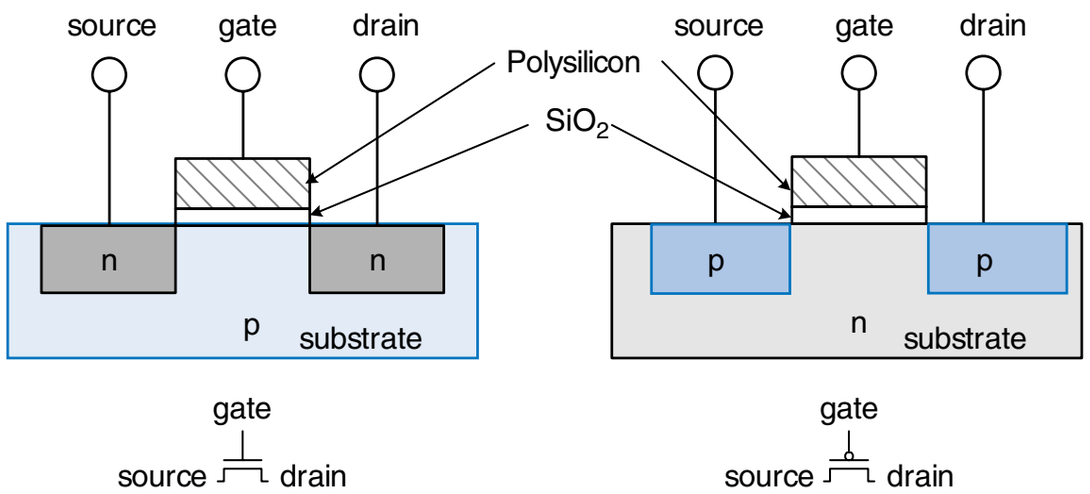
- Scaling (tech/business) model comes to an end ...
- Long term solution
  - ✓ Solid paradigm change (going beyond a short term solutions)
- Short term solutions
  - ✓ Better technology (still possible)
  - ✓ Better system understanding (today more then ever)
  - ✓ **Co-design of SW/HW/IC technology**

These lectures are about understanding HW better and how we can get the best out of it

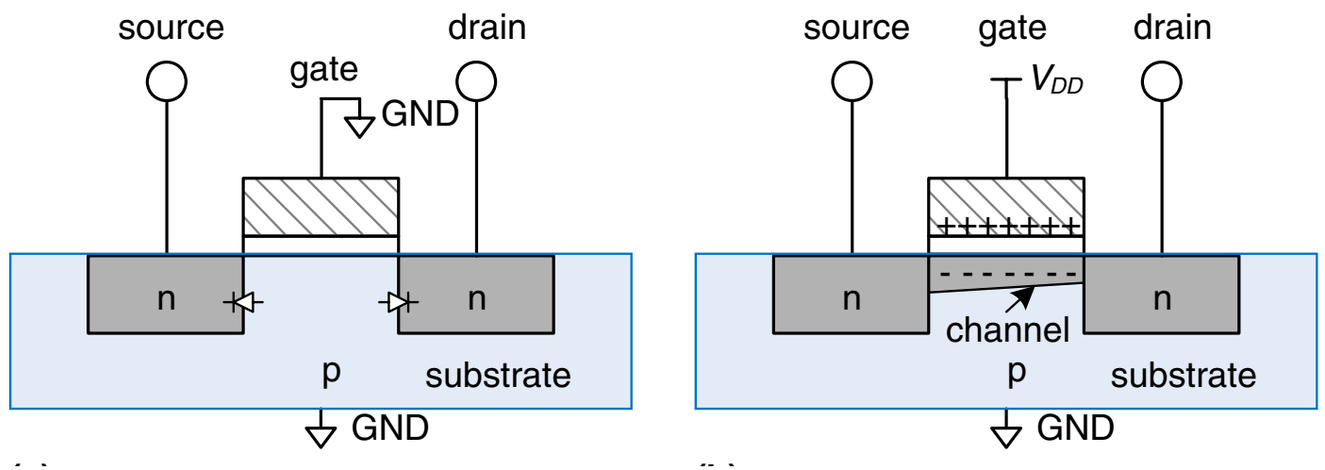
2. IC manufacturing technology perspective

From CMOS transistor ...

n-type and p-type transistors:

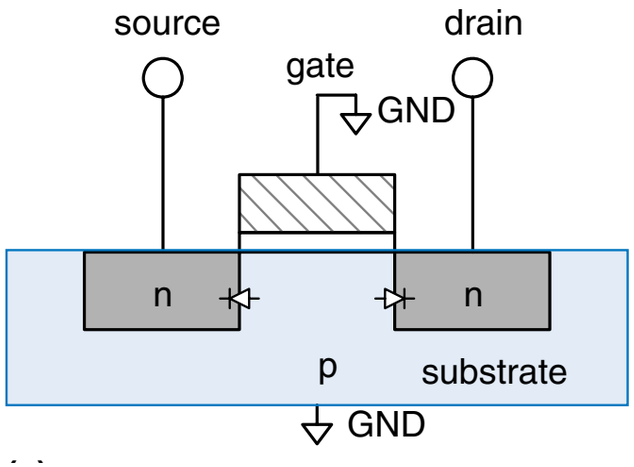


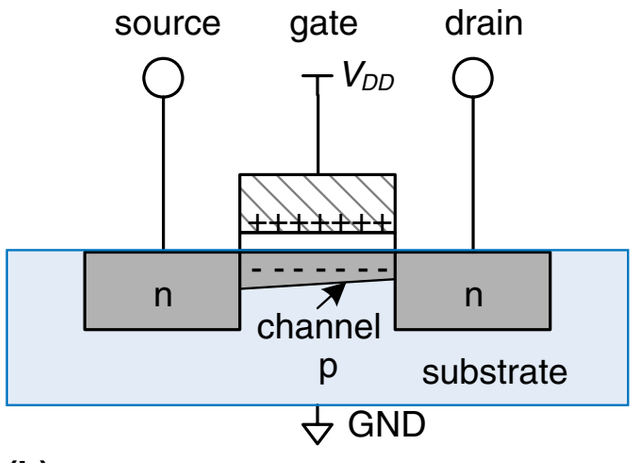
Current flow is controlled by the gate:



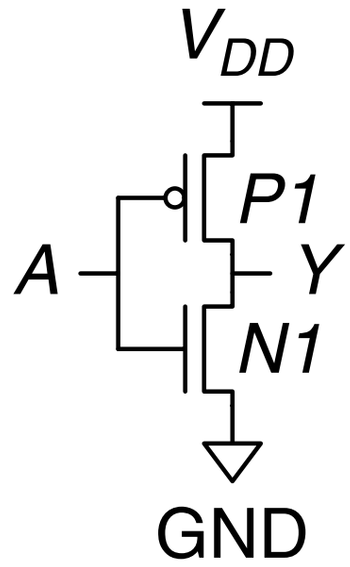
… to gate (switch)

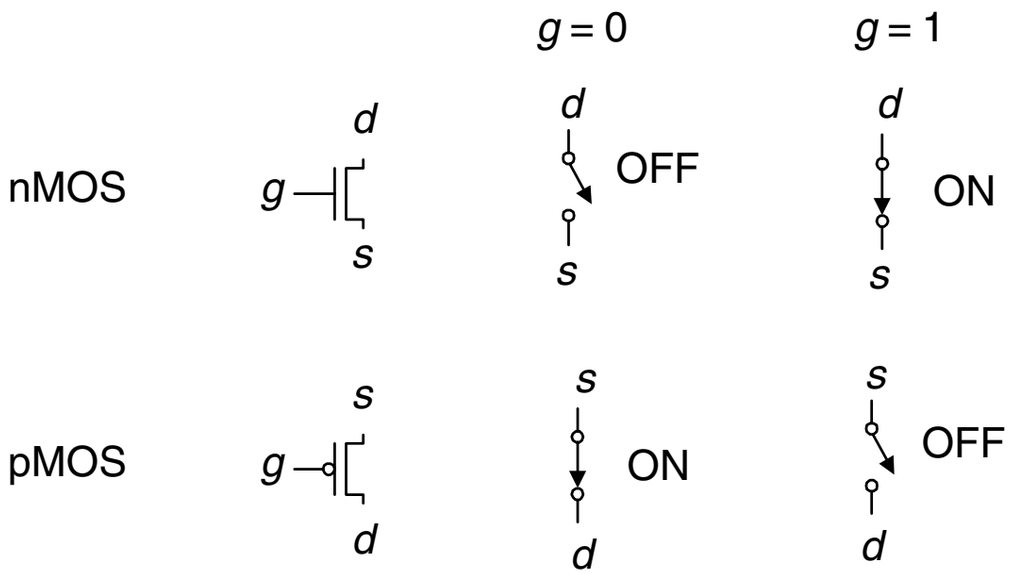
If the control is binary, the transistor acts like a switch:

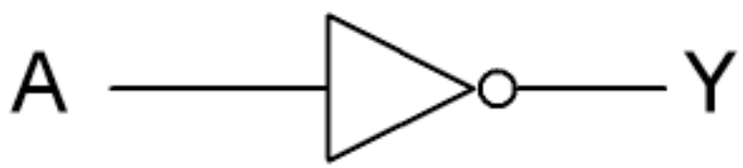




2 switches used to make an inverter:





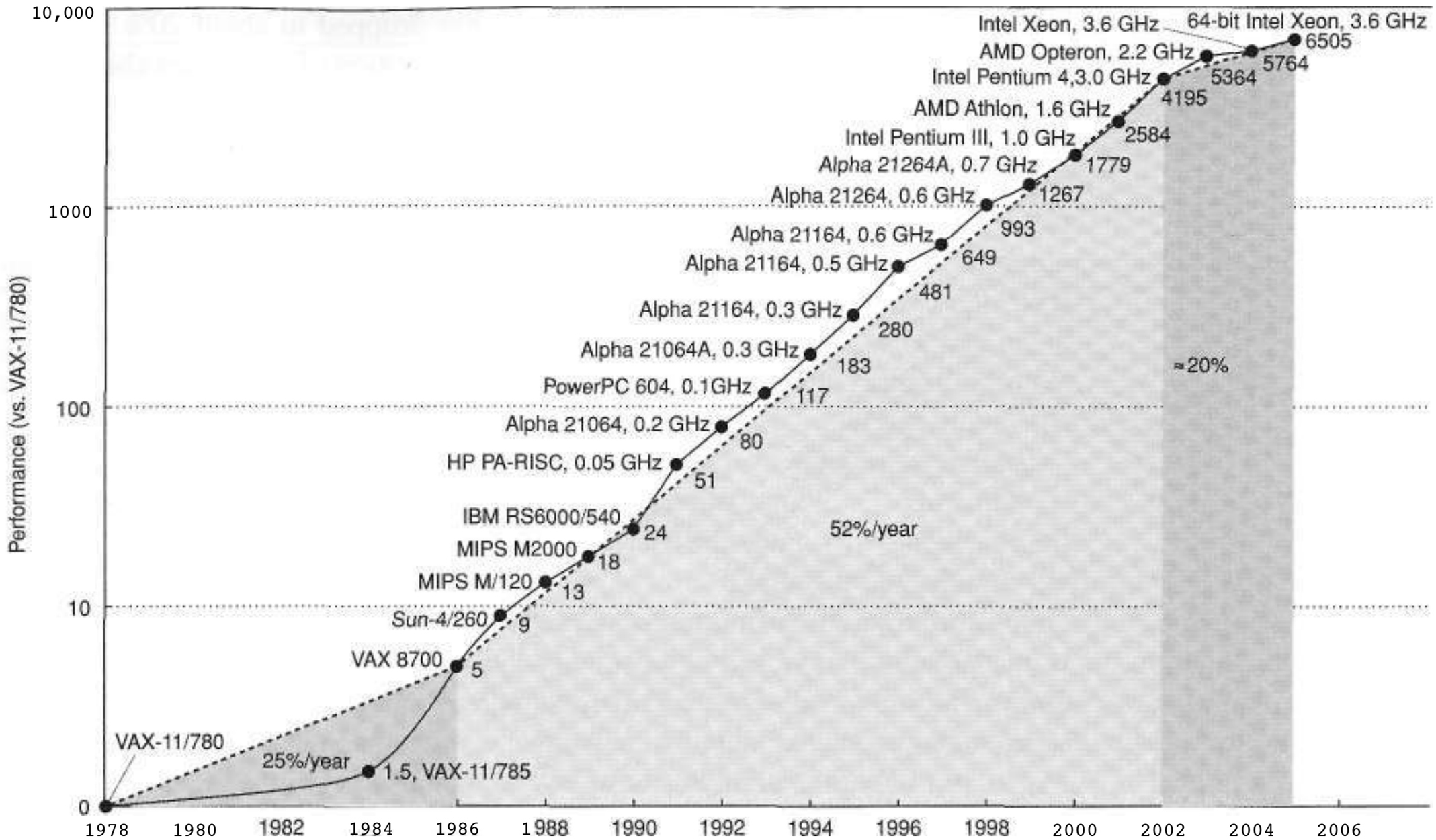


Evolution of CMOS

- We print features on silicon
- If we can print smaller features :
  - ✓ We can reduce transistors size
  - ✓ We can reduce width/length of the interconnect
  - ✓ **More functionality at higher performance for the same area (cost)**

  → This is SCALING
- Currently:
  - ✓ 28, 22nm but with lot's of issues
  - ✓ 14nm Intel – **DELAYED**
  - ✓ 11nm should arrive sometimes in the near future

Scaling enables better performance



Evolution of CMOS

- **This was the model that run smoothly for past 50 years**
- This is not the case any more …
- After 100nm (sub-micron, ultra deep sub-micron) technology nothing is going to be the same as before
  → *More then Moore* paradigm

✓ Inversion of scaling properties

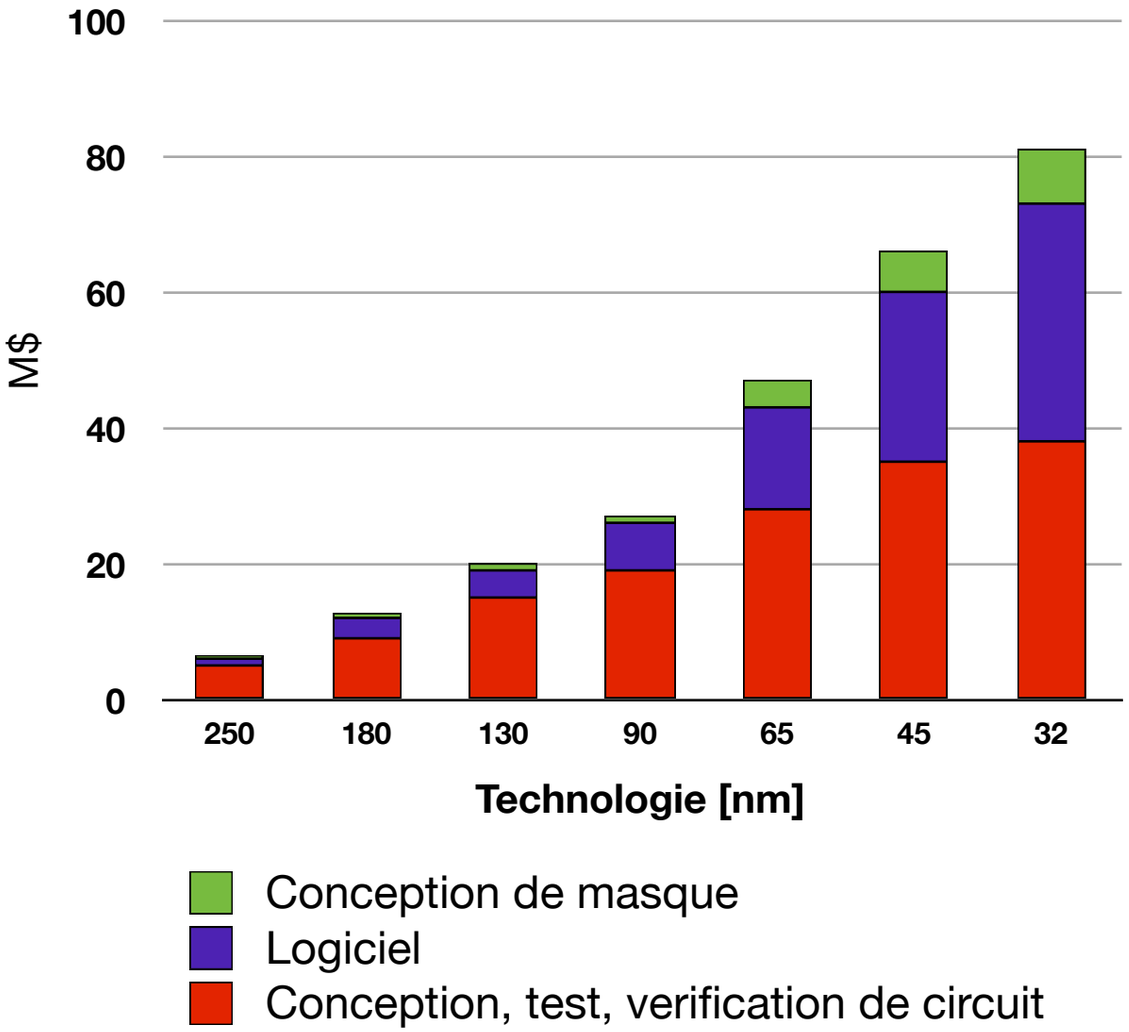
✓ Gains are not the same

✓ We start even loosing …

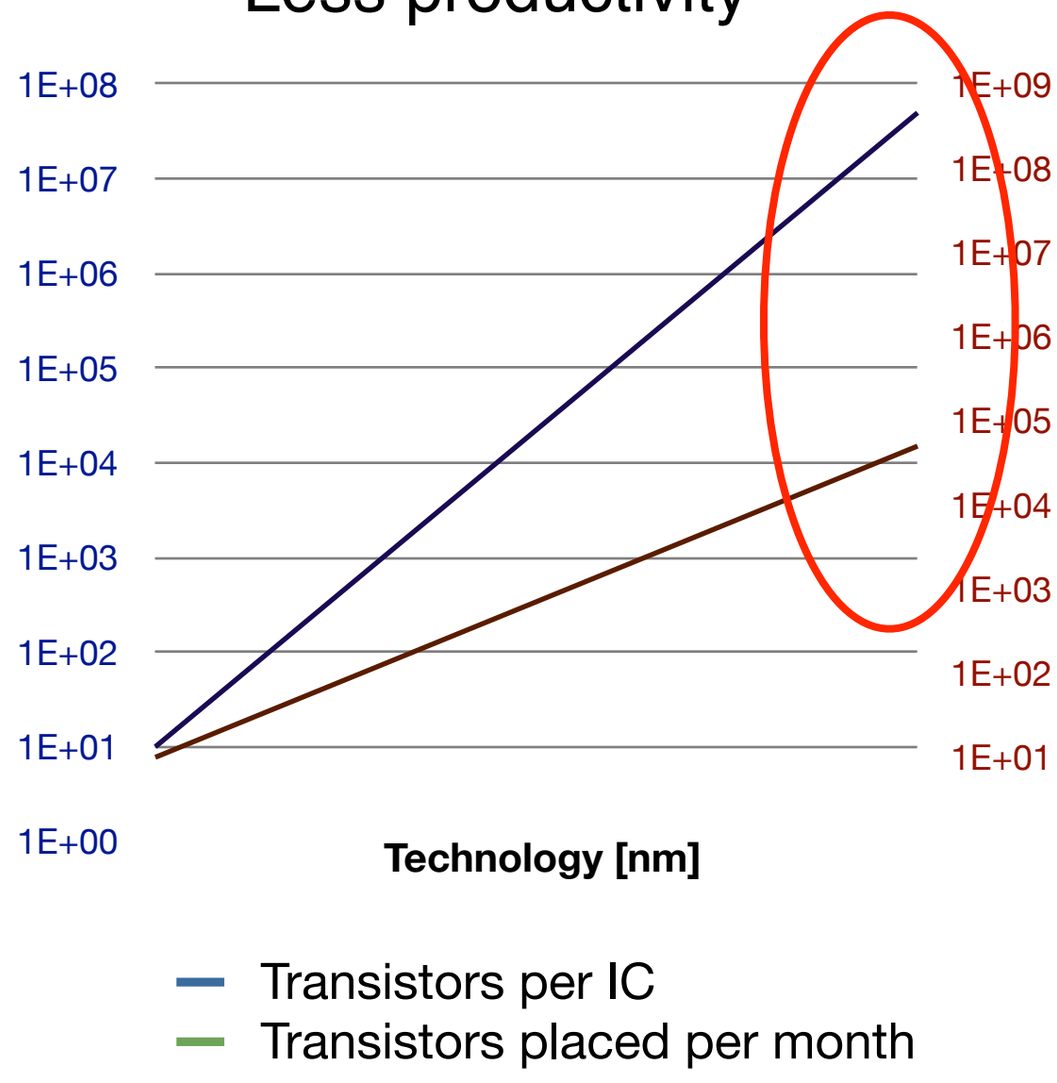
Scaling side effects !

Scaling side effects : a) $$$$$

Fabrication cost



Less productivity



Consequence → Technology evolution and design capability do not follow the same path !

Cost examples 2/2

- IC cost : very complex equation that is in general carefully balanced (in a very simplified form)

$$\text{IC cost} = \frac{\text{Die cost} + \text{Testing cost} + \text{Packaging cost}}{\text{Final test yield}}$$

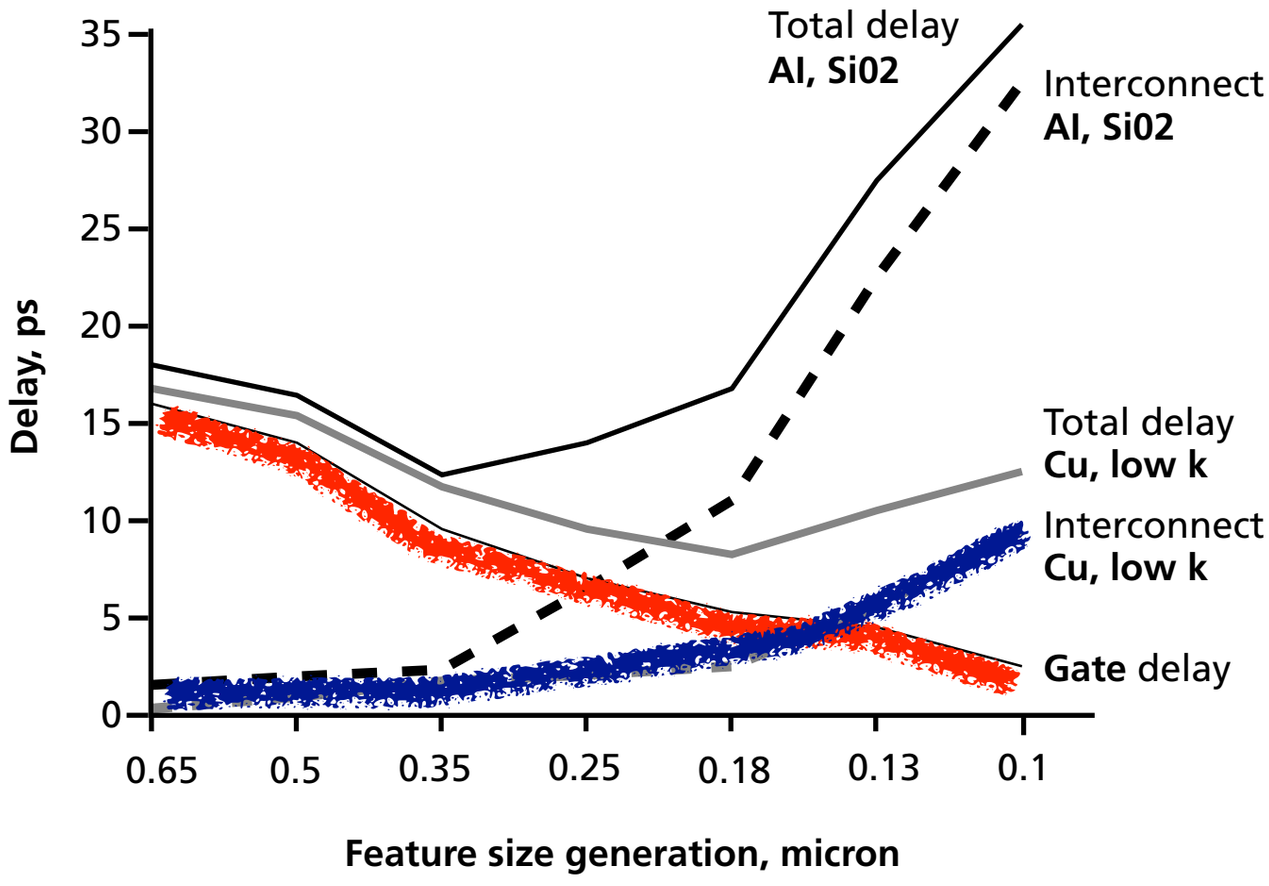
Packaging Cost: depends on pins, heat dissipation

- In practice: constantly increasing !

Chip	*Die cost*	*Package pins*	*Package type*	*Package cost*	*Test & Assembly*	*Total*
386DX	$4	132	QFP	$1	$4	$9
486DX2	$12	168	PGA	$11	$12	$35
PowerPC 601	$53	304	QFP	$3	$21	$77
HP PA 7100	$73	504	PGA	$35	$16	$124
DEC Alpha	$149	431	PGA	$30	$23	$202
SuperSPARC	$272	293	PGA	$20	$34	$326
Pentium	$417	273	PGA	$19	$37	$473

Scaling side effects : b) performance gains

We have gate delays that decrease,
but not those of the wires :
→ We can compute fast, but we communicate slowly !



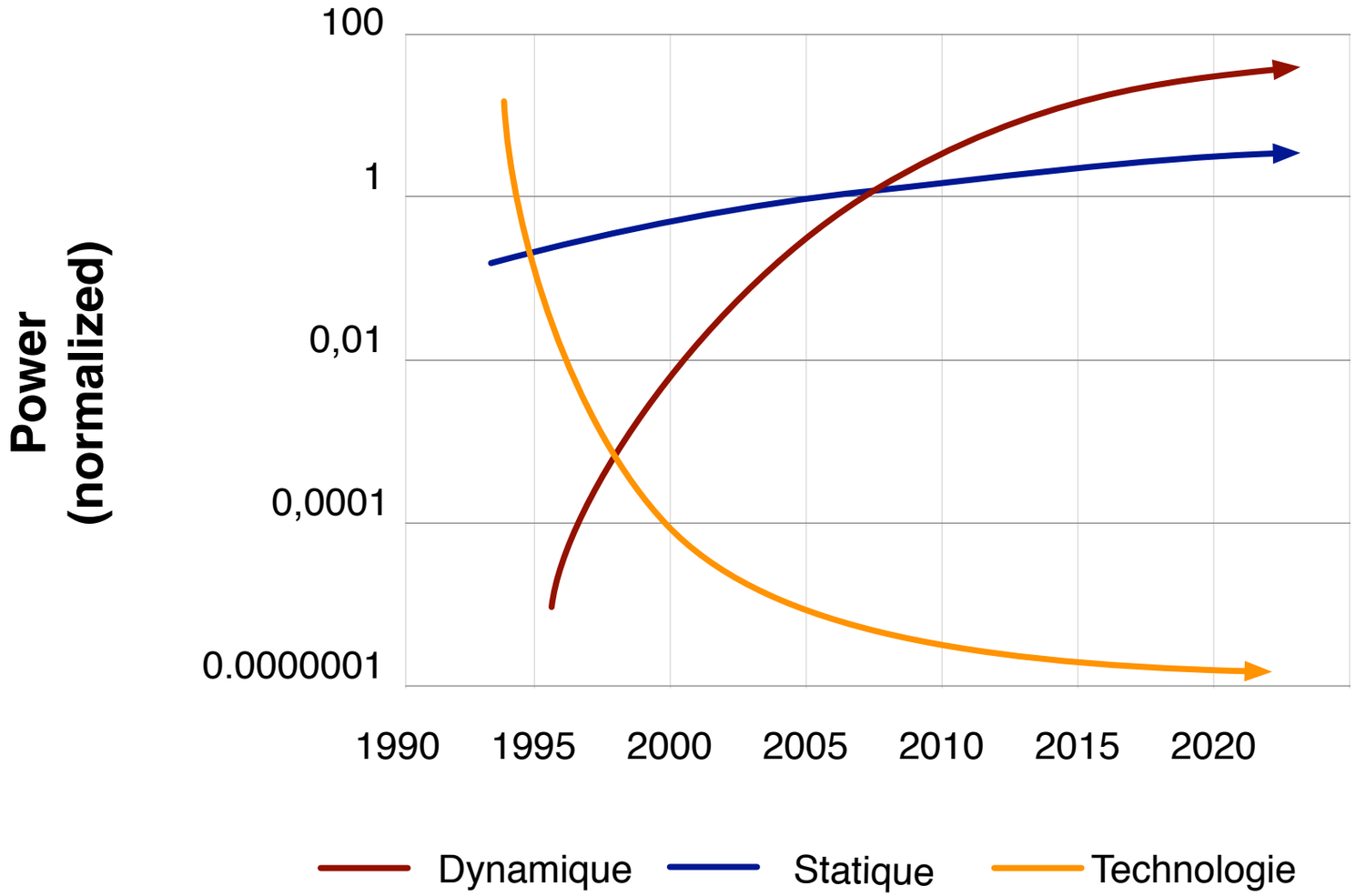
Gate delay

Wire delay

Consequence → Optimization should be done at communication level too !!! (NoCs)

Scaling side effects : c) power

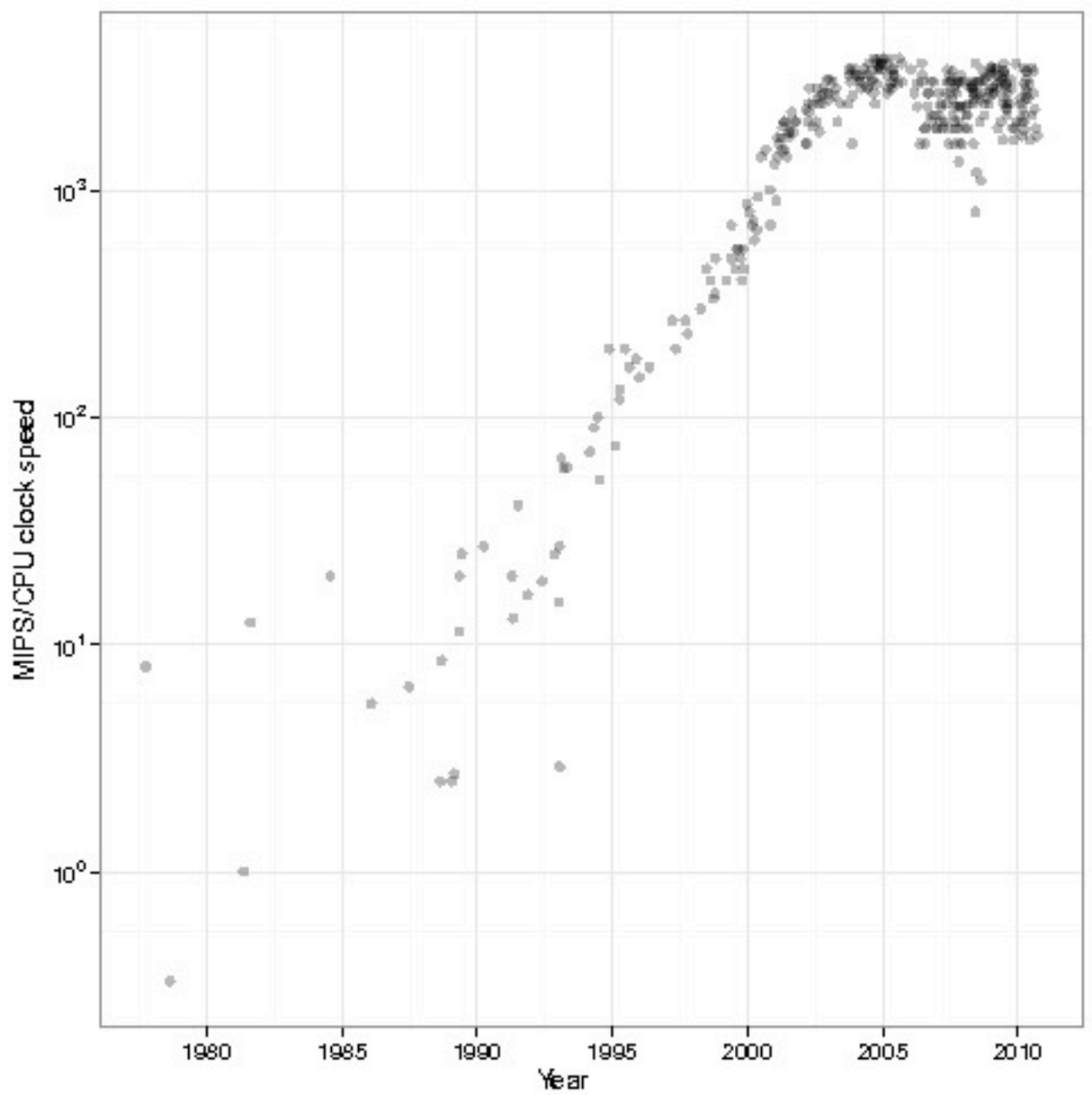
Tendency is changing
(curves are normalised to dynamic power dissipation)



Consequence → Get 10% savings in dynamic power dissipation is not significant any more !

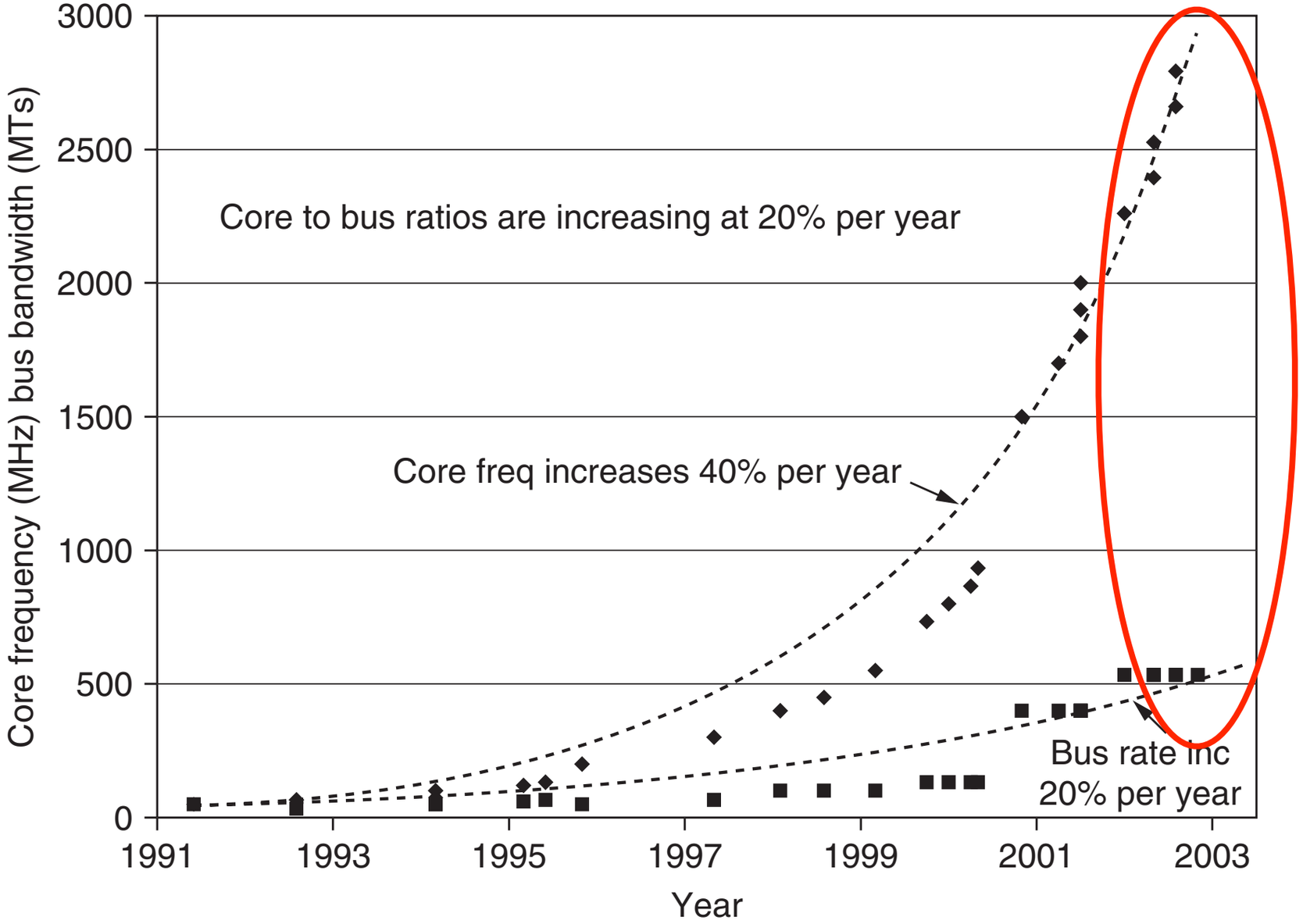
End result is :

- That the CPU F do not increase anymore, to get more functionality (performance) we increase the parallelism



System level impact of scaling

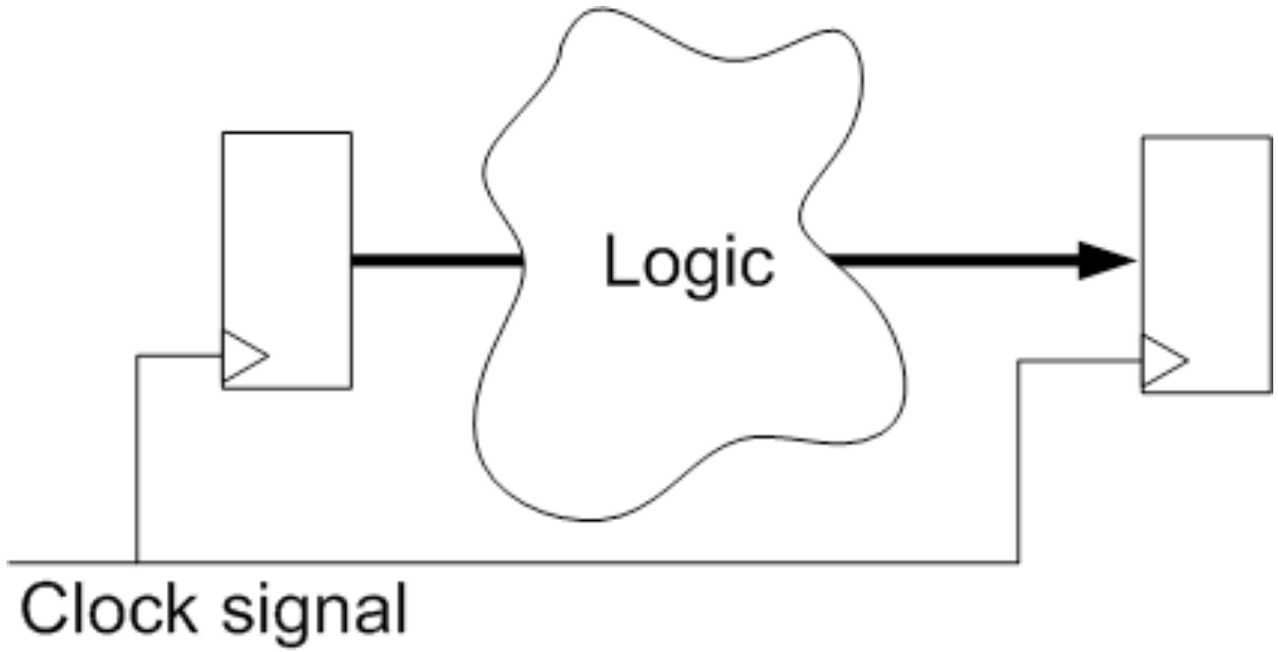
Memories and CPUs do not scale equally !



3. Computing systems performance

Performance

- **Clock cycle** (Clk)
  - ✓ Clk is there because CPU is a synchronous logic circuit (circuits with feedback) – system state is stored in flip-flops
  - ✓ Clk is used to drive all flip-flops in the design (data-flow from flops to flops, so for the combinatory circuits too)
  - ✓ Typically one master clock that supply different clock domains



Performance

- We can measure the number of cycles required to execute all instruction within a computer program
- We can count the number of executed instructions
- **Cycles per instruction** (CPI) — on average for a given program :

$$CPI = \frac{\text{Total number of cycles to execute}}{\text{Total number of instructions in the program}}$$

Performance

- CPI of each instruction (CPU data sheet)

  ✓ addition, logic operation (simple) — 1 cycle,

  ✓ multiplication (complex operation) — from 1 to few cycles, depending on hardware

- **Instruction(s) Per Cycle** (IPC) — for an application

  ✓ IPC = 1/CPI — but computed a posteriori (profiling)

  ✓ Measures the parallelism if it is > 1

  ✓ Most of the computers should have this TRUE !!!

Performance

- Execution of a computer program (IC — app instruction count):

  CPU_time = Clk x CPI x IC

- How to **minimize** CPU_time ?

  ✓ Increase Clk → Increase F (will not hold that long)
  → look at IC scaling predictions for the future from node to node:

	Feature size	Area	C	F	Vdd	Power	Power Density
45 ---> 32	0.755	0.57	0.665	1.1	0.925	0.626	1.096
32 ---> 22	0.755	0.57	0.665	1.08	0.95	0.648	1.135
22 ---> 14	0.755	0.57	0.665	1.05	0.975	0.664	1.162
14 ---> 10	0.755	0.57	0.665	1.04	0.985	0.671	1.175

Performance

- How to **minimize** CPU_time ?

  ✓ Increase Clk → Increase F (will not hold that long)

  ✓ Reduce CPI → Parallelism:
  inter et intra CPU (multi, scalar, super-pipeline etc.)

  ✓ Reduce IC → Algorithm, SIMD, implementation (SW), …

- Certain mechanisms are automatic, **others are not !**
  → Optimizations as function of the architecture

- **You need to know HW and the way that operate to be able to exploit at best all the possibilities that are there !**

Solutions?

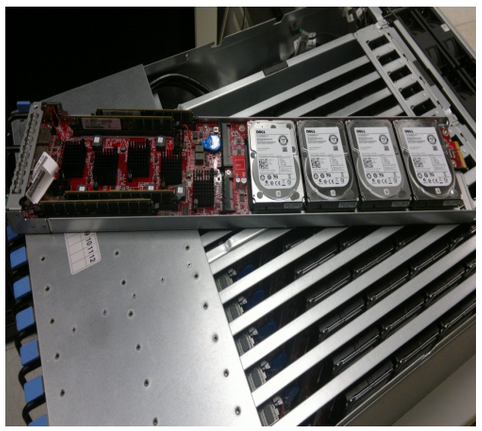
- Improve tech
- Increase parallelism … multi, many core → multi-processor
- Better usage at application level
- After all, all these systems are used badly …
- Let's see this on a **concrete example**

  DATA CENTERS!!! (cloud computing)

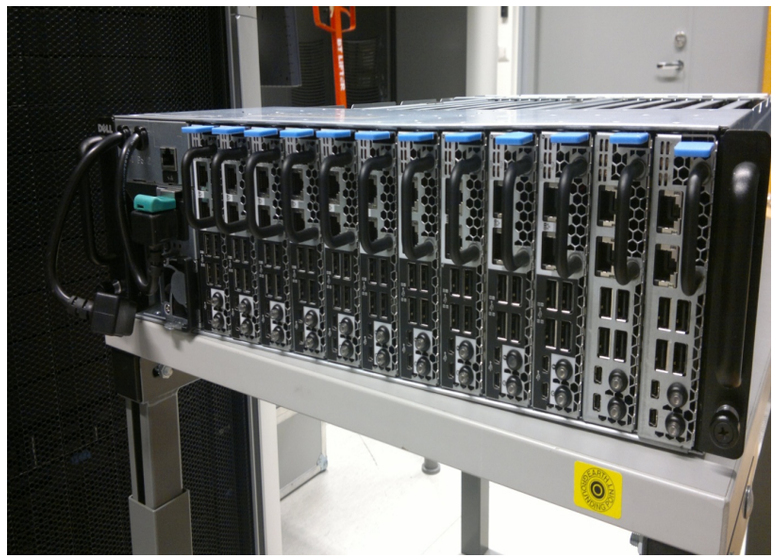
4. Example of poor usage: Data Centers

Data centers are power hungry !

Board



Rack



Building

… in all, thousands of CPUs using considerable power.

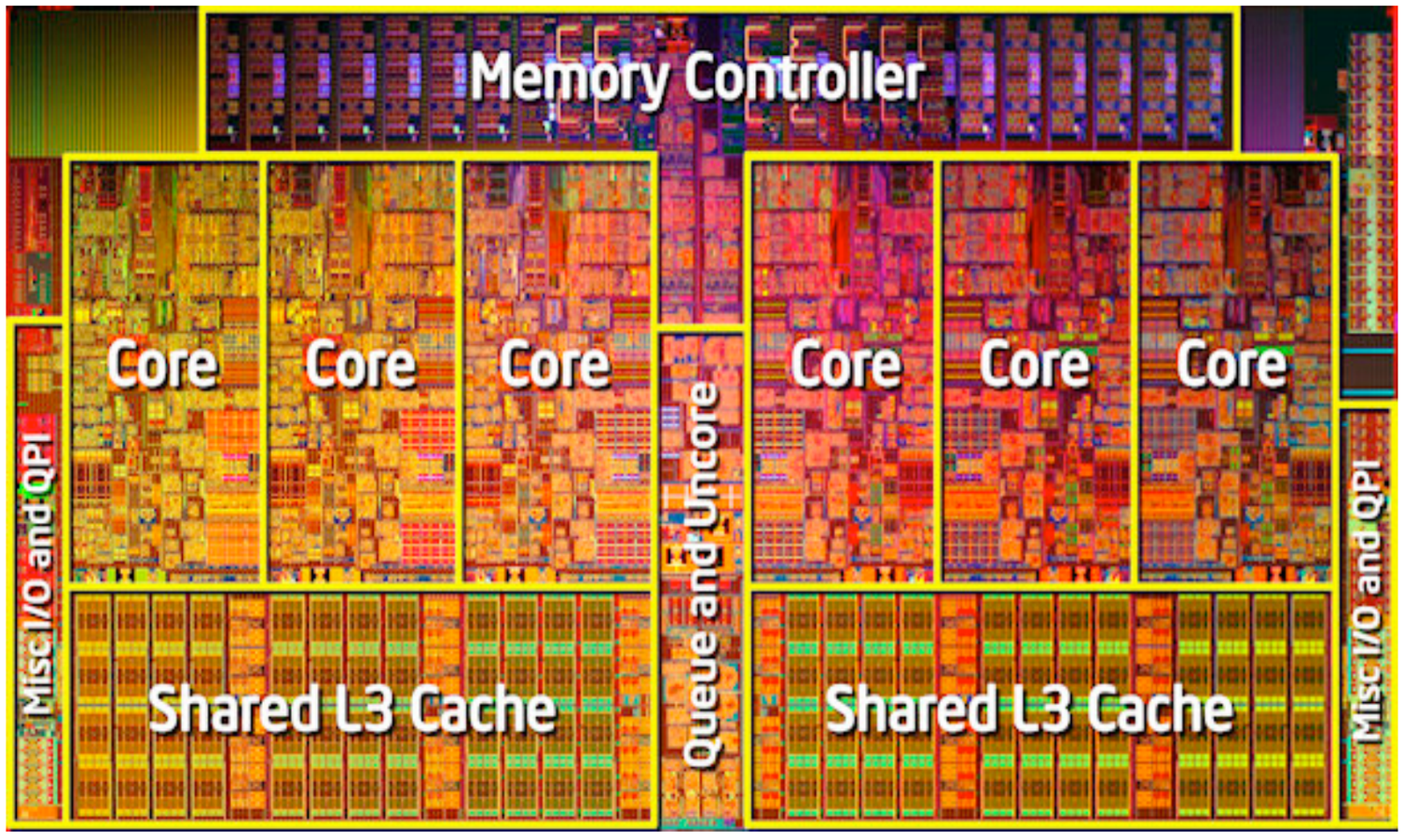
Did BIG ones (MS, Yahoo, etc.) became “GREEN” ?

$$$ Electricity bill $$$

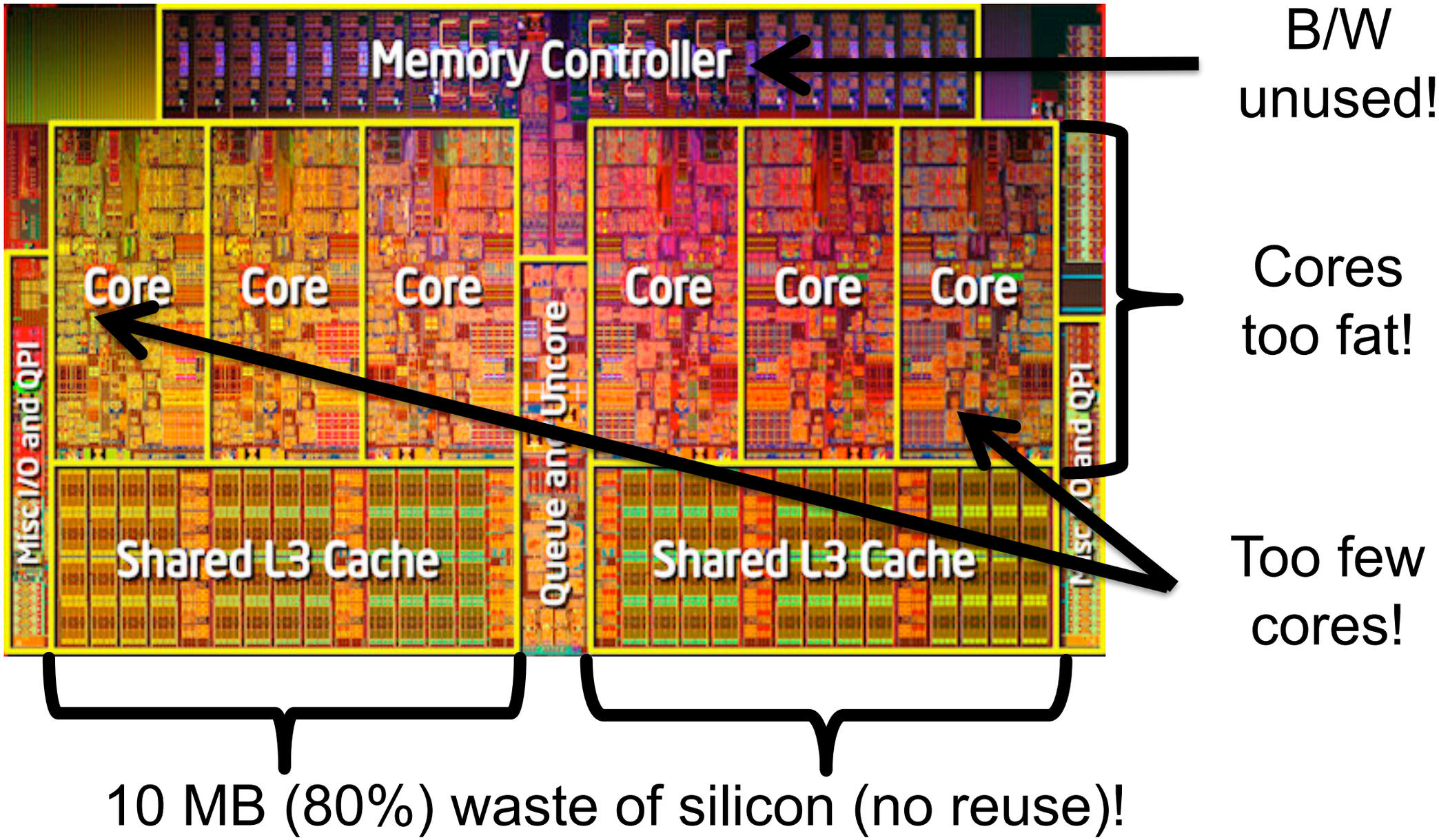
In 2007: 7.2 Billions US$

Data centers use traditional cores

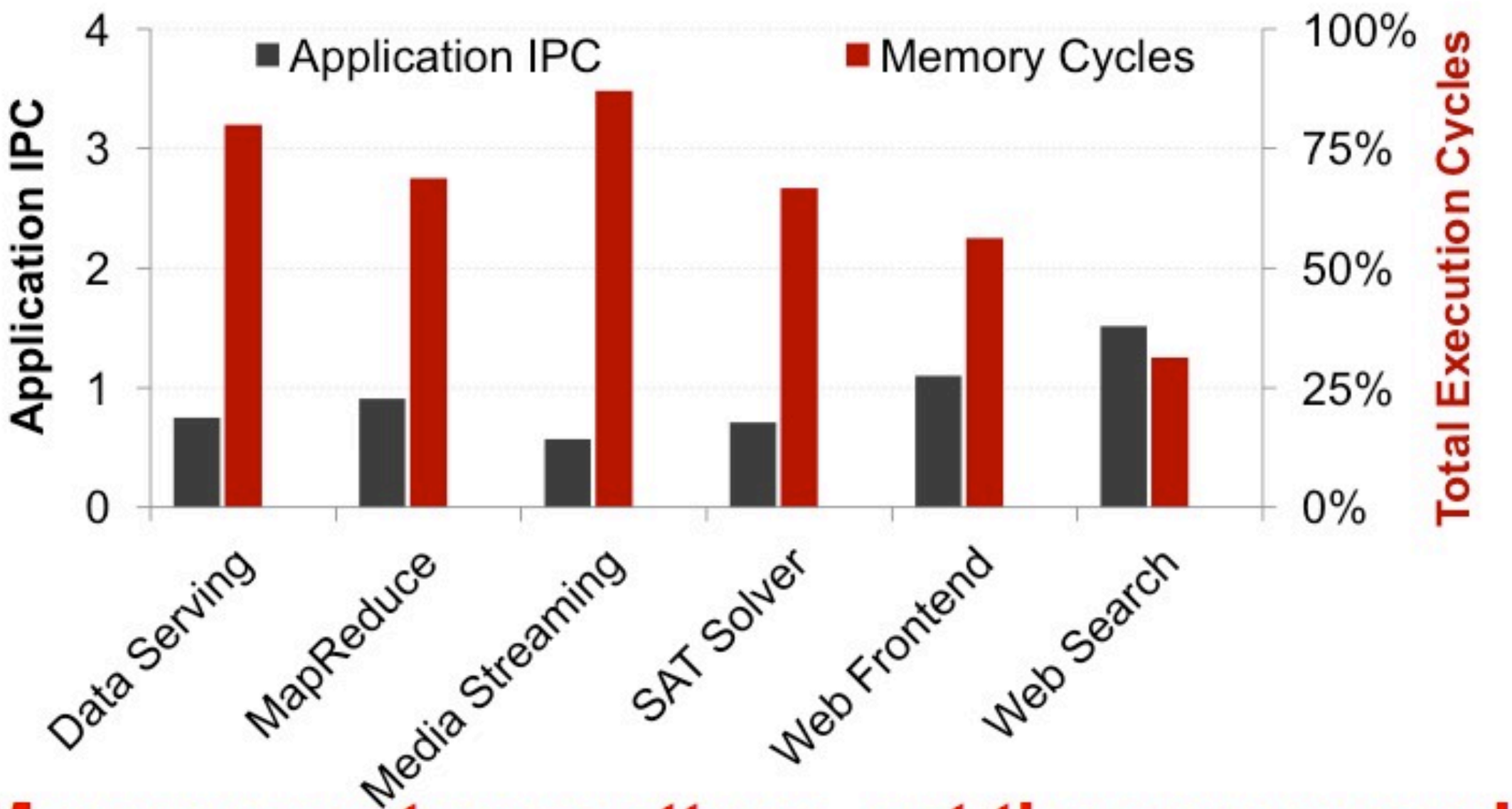
- Heavily pipelined
- Bunch of FPUs
- SIMD support
- Big, shared caches
- Complex circuits, built to suit any application …
  (as long as it is not embedded)



How good multi-core really is?



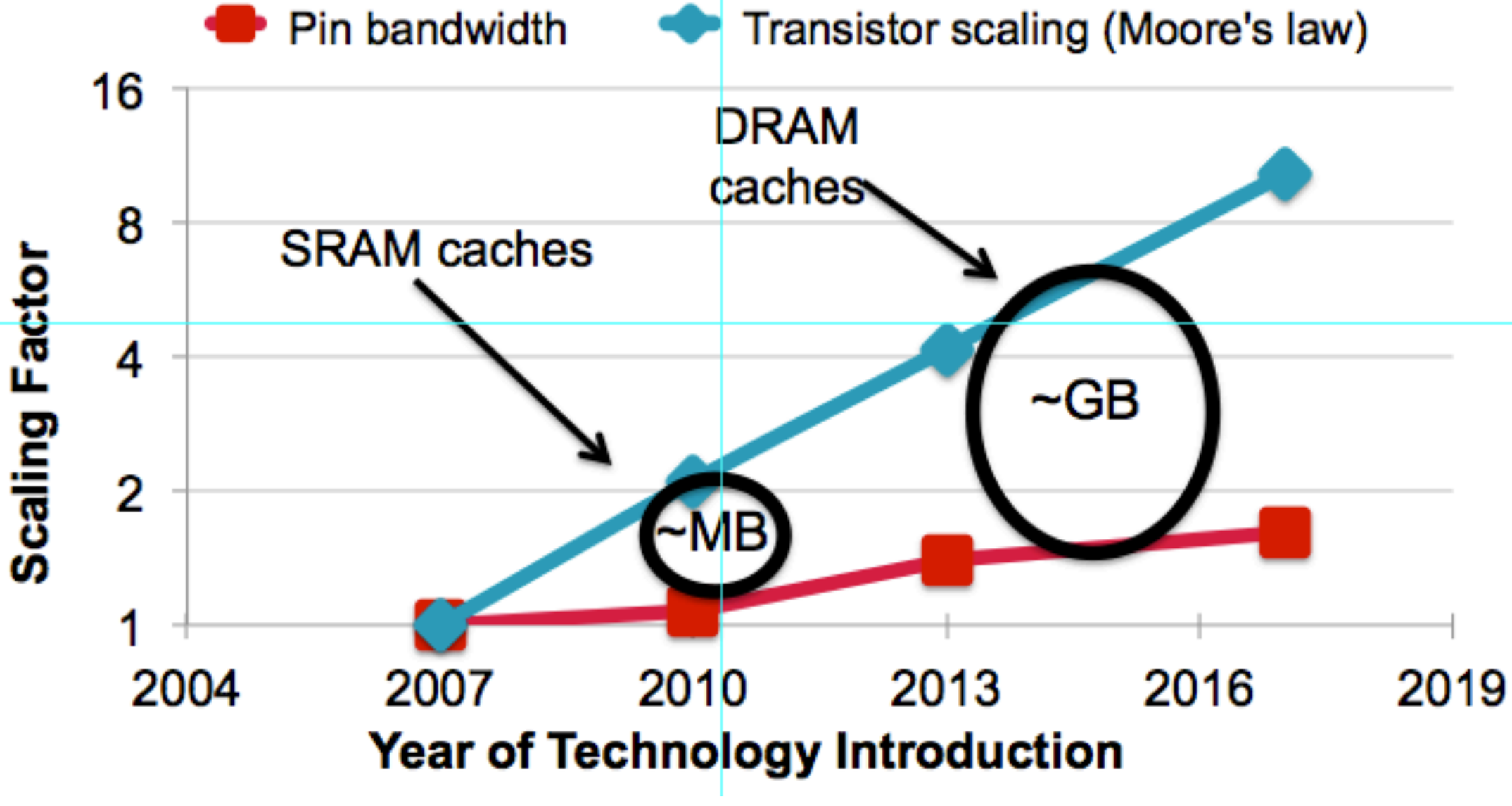
But how good parallelism really is?



Memory system matters, not the core speed

But how good parallelism really is?

- Tomorrow's chip: 100s of cores + little cache
  - off-chip bandwidth becomes bottleneck



Learnings

- “One fits all” solution was the only one economically viable
  - ✓ Same CPU: for gaming, scientific computing, grandma’s word-processing and data center
  - ✓ Worked very well in the past (Intel), but ...
  - ✓ Doesn’t work any more !
- Computing usage habit changed: we eventually went back to the terminal/main frame concept from the past (tablet/cloud)
  - ✓ Small/or not embedded computing power with IO capacity
- Demand on high-perf CPUs is slowing down, much more then even almighty Intel predicted: 14nm fab is delayed !

5. What could happen in the future ?

(this is not a tale)

Computer classes and important issues

- **Desktop Computing**

  ✓ Price-performance ratio and graphics capabilities (gaming!, look at NVIDIA)

- **Servers**

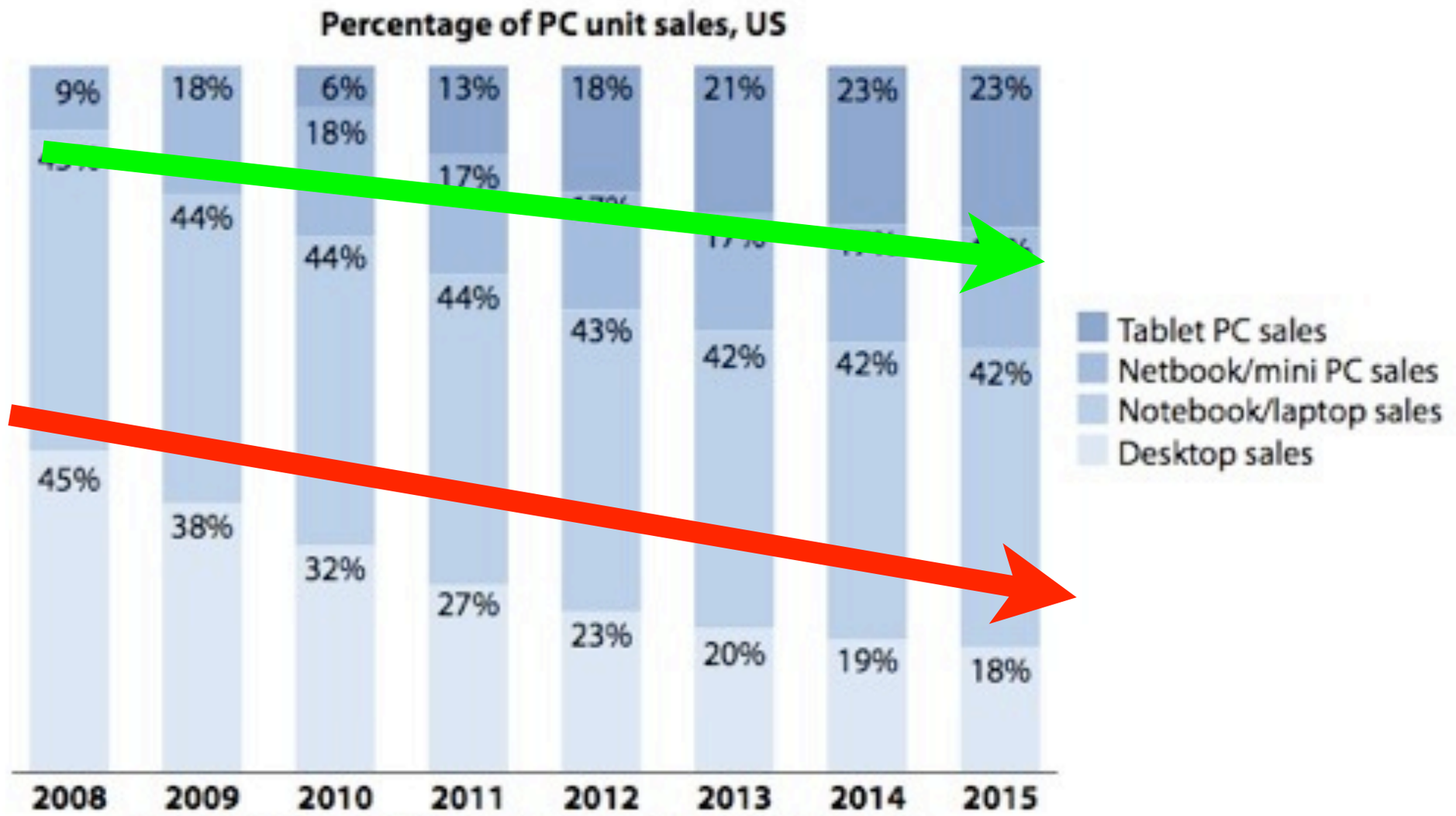
  ✓ Throughput, availability, scalability

- **Embedded Computers**

  ✓ Price, power consumption, application-specific performance

Computer classes — Winds of change

Figure 1 Forecast: Share Of US Consumer PC Sales By Form Factor, 2008 To 2015



(percentages may not total 100% because of rounding)

Source: Forrester Research eReader Forecast, 2010 To 2015 (US)

57210

Source: Forrester Research, Inc.

What could happen in the near future ?

- **Desktop Computing** — disappears, Intel opens their fab and stop working on CPUs
- **Servers** — made using low power cores like ARM
- **Embedded Computers** — made using the “same” low-power cores used for servers
  (just look at the Apple products: iPhone/iPad)

- What about CPUs?
  - ✓ CPU architectures are stable
  - ✓ Instructions set do not change much
    (although they can be adapted to a particular app)
  - ✓ We need to start really using them → plus system integration

So what's for us there?

- Whatever underlaying tech will be used (even in the far future) some **processing devices** will always be there

  ✓ Atomic adder — it is still an adder

- **Processing device = CPU**
- **Architectural concepts** of the CPU may vary depending on the technology offering, but lots of fundamental concepts will probably remain the same

  ✓ Even if low-power CPUs are killing desktop CPUs they still

  - ✦ Have pipelined structure
  - ✦ Use reg files and ALUs to compute things
  - ✦ Parallelize what ever could be done in parallel … & many others