Computer Architecture Lecture 2

Peter Mikulecky

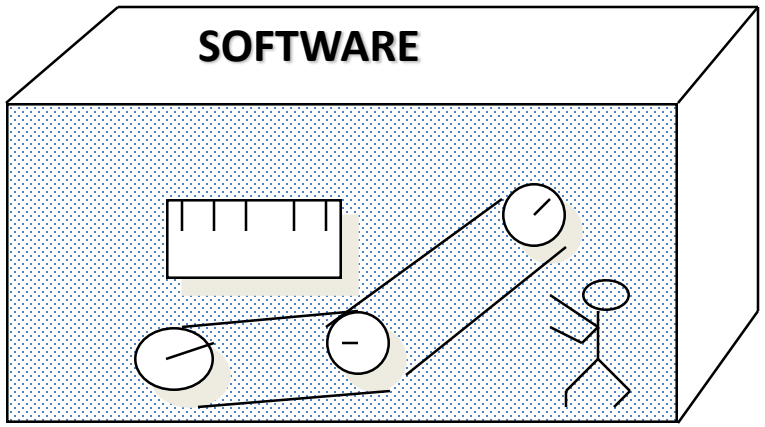
Integrated Approach

- Computer architecture should be viewed through an integrated approach, integrated view
- What really matters is the functioning of the complete system
  - hardware, runtime system, compiler, operating system, and application
  - In networking, this is called the "End to End argument"
- Computer architecture is not just about transistors, individual instructions, or particular implementations

What's Computer Architecture?

The attributes of a [computing] system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls the logic design, and the physical implementation.

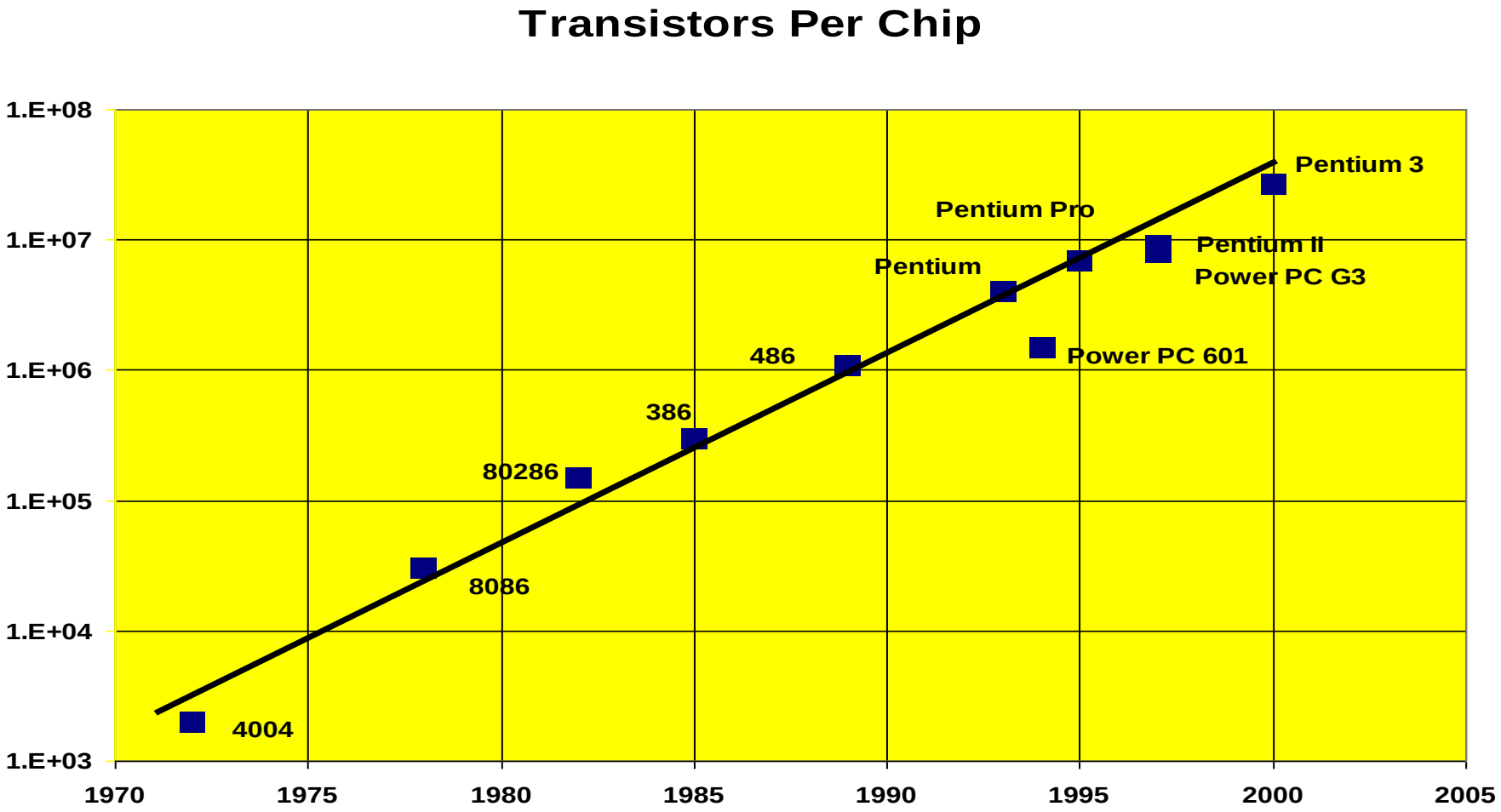
Amdahl, Blaaw, and Brooks, 1964



Trends

Gordon Moore (Founder of Intel) observed in 1965 that the number of transistors that could be crammed on a chip doubles every year.

This has CONTINUED to be true since then.



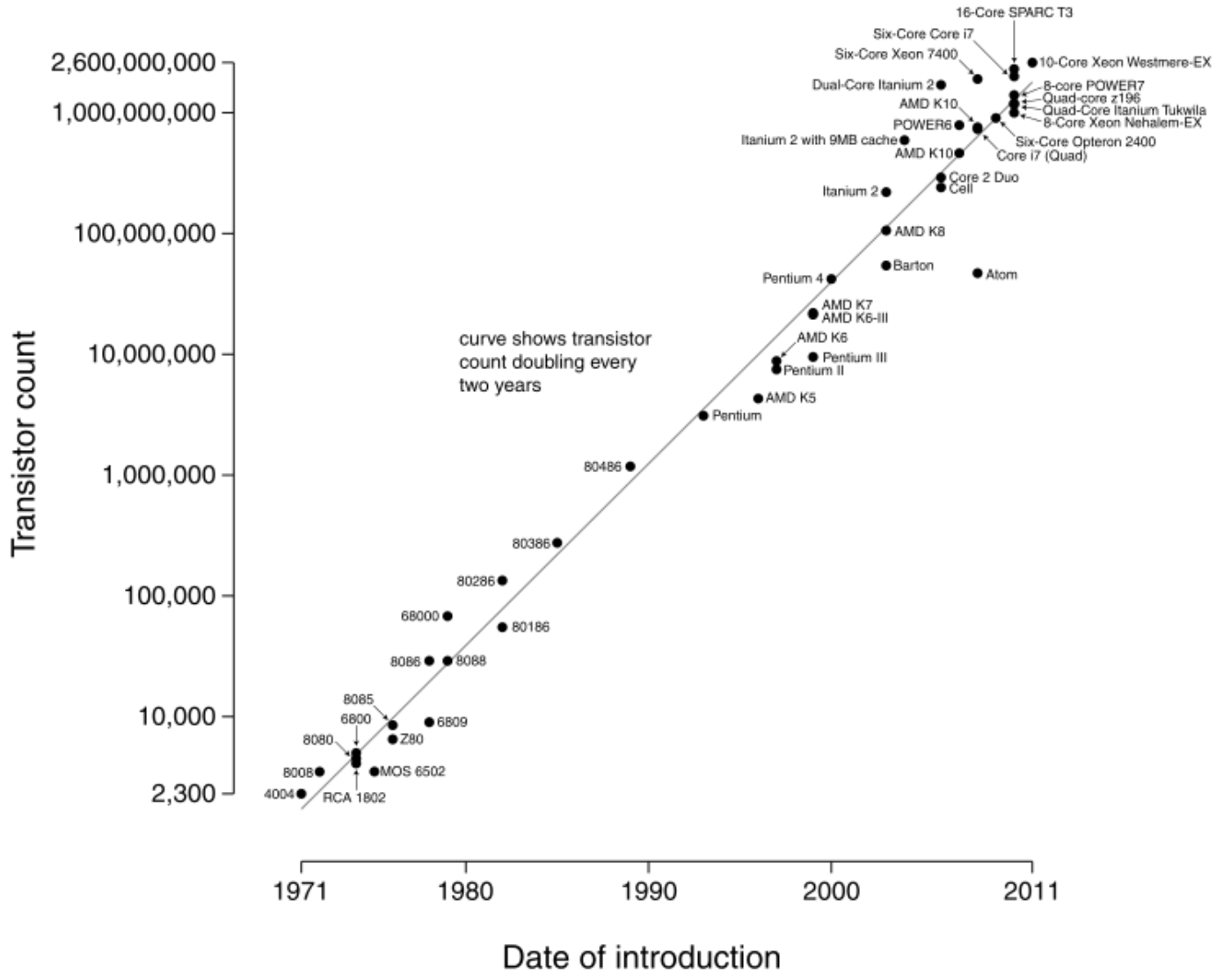
Moore‘s Law

- Moore’s Law, first hypothesised in 1965 by Intel founder Gordon Moore, states that the number of transistors in a dense integrated circuit will double approximately every two years. The shrinking of transistors enables a larger number to be held within the same area, which results in a faster processor that can operate at lower power requirements.
- Although the law was adhered to rigidly for half a century, in 2015 Intel admitted that the pace of advancement had started to slow down. Its eighth-generation Core CPUs, codenamed Coffee Lake, are set to launch in the second half of 2017 and will once again be built on the same 14-nanometre (nm) process used three generations prior for its Broadwell chips, originally released in 2014.

The end of the Moore‘s Law?

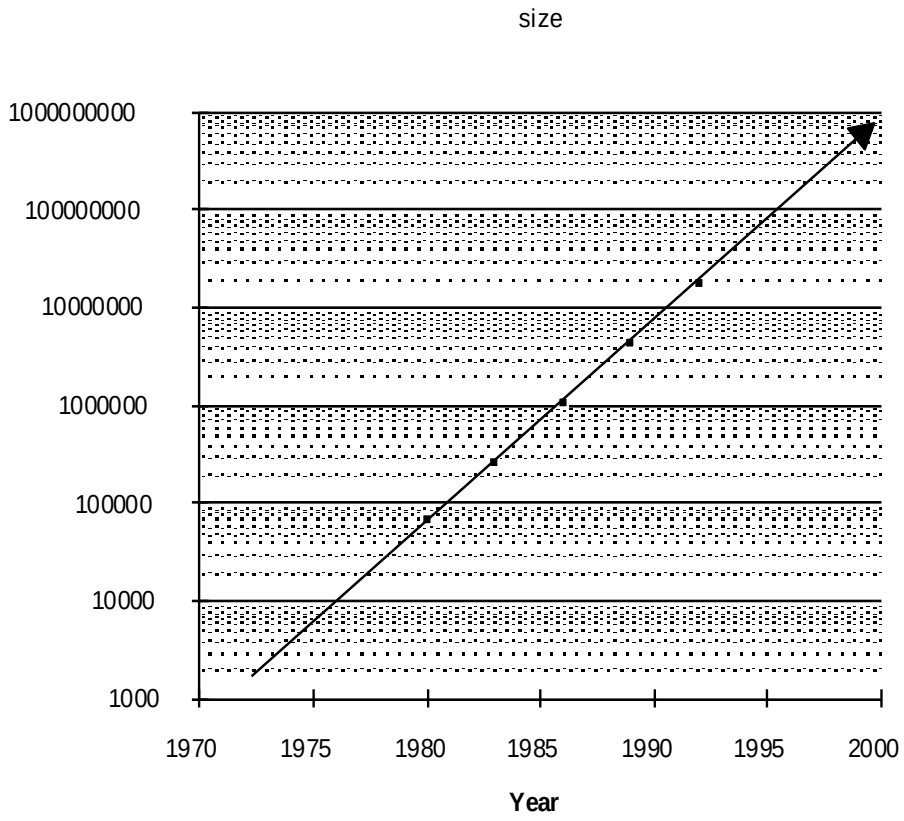
- “I don't think anyone could confidently tell you that they have a plan for 15 more years of Moore's law,” says Greg Yeric, director of future silicon technology for ARM Research.
- „Moore‘s Law 2017: An Uphill Battle“
- https://eandt.theiet.org/content/articles/2017/05/moore-s-law-2017-an-uphill-battle/
- „Is Moore’s Law still the law?“
- https://www.electronicsweekly.com/news/moores-law-still-law-2017-09/

Microprocessor Transistor Counts 1971-2011 & Moore's Law



Trends

Memory Capacity (and Cost) have changed dramatically in the last 20 years.



year	size(Mb)	cyc time
1980	0.0625	250 ns
1983	0.25	220 ns
1986	1	190 ns
1989	4	165 ns
1992	16	145 ns
1996	64	120 ns
2000	256	100 ns

Trends

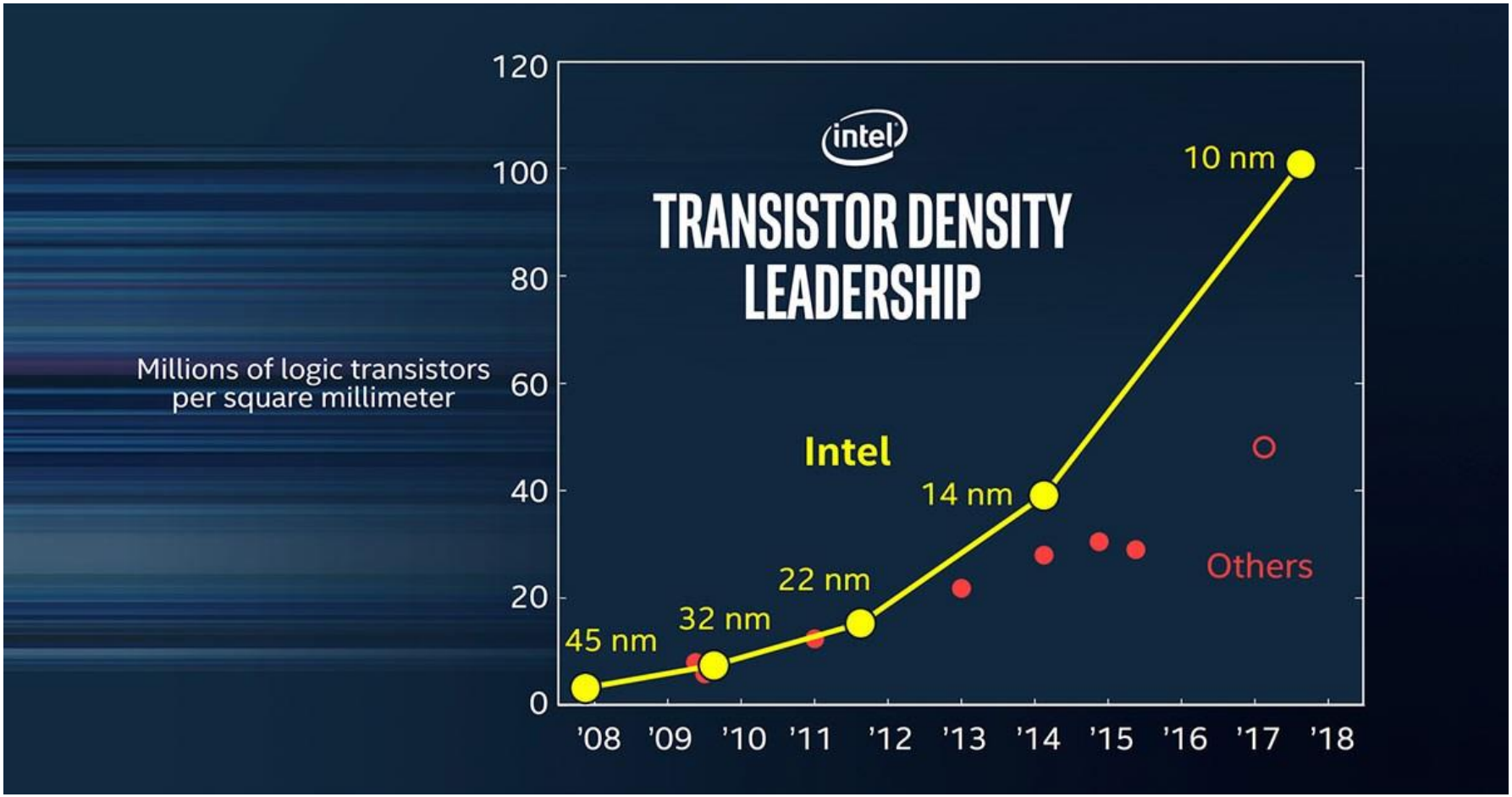
Based on SPEED, the CPU has increased dramatically, but memory and disk have increased only a little. This has led to dramatic changes in architecture, operating systems, and programming practices.

	Capacity	Speed (latency)
Logic	**2x in 3 years**	**2x in 3 years**
DRAM	**4x in 3 years**	**2x in 10 years**
Disk	**4x in 3 years**	**2x in 10 years**

Workstation performance (measured in Spec Marks) improves roughly 50% per year (2X every 18 months)

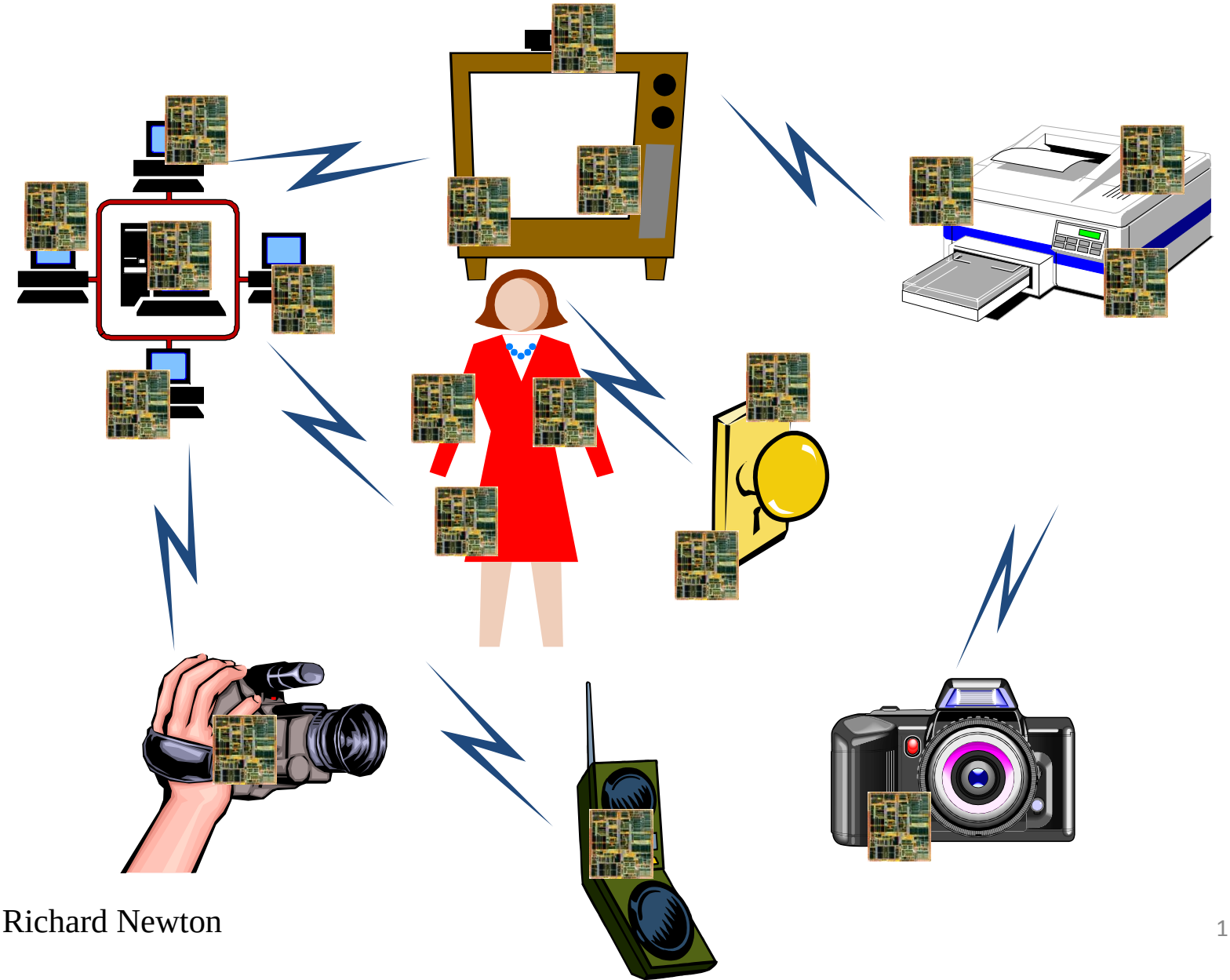
Improvement in cost performance estimated at 70% per year

Moore‘s Law 2017



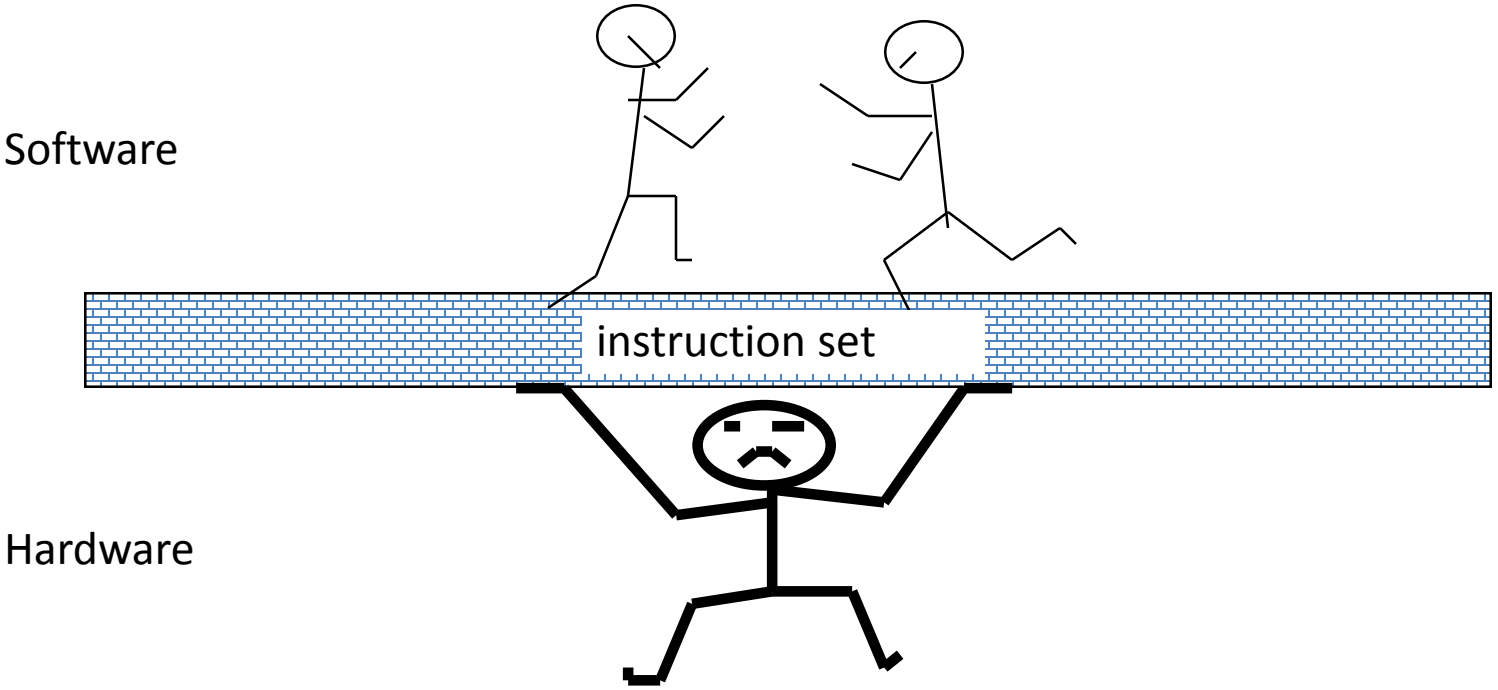
https://eandt.theiet.org/content/articles/2017/05/moore-s-law-2017-an-uphill-battle/

What is the next wave?



Source: Richard Newton

Instruction Set Architecture: Critical Interface



- Properties of a good abstraction
  - Lasts through many generations (portability)
  - Used in many different ways (generality)
  - Provides convenient functionality to higher levels
  - Permits an efficient implementation at lower levels

Software Abstraction

```
int sum(int x, int y)
{
  int t = x+y;
  return t;
}
```

C

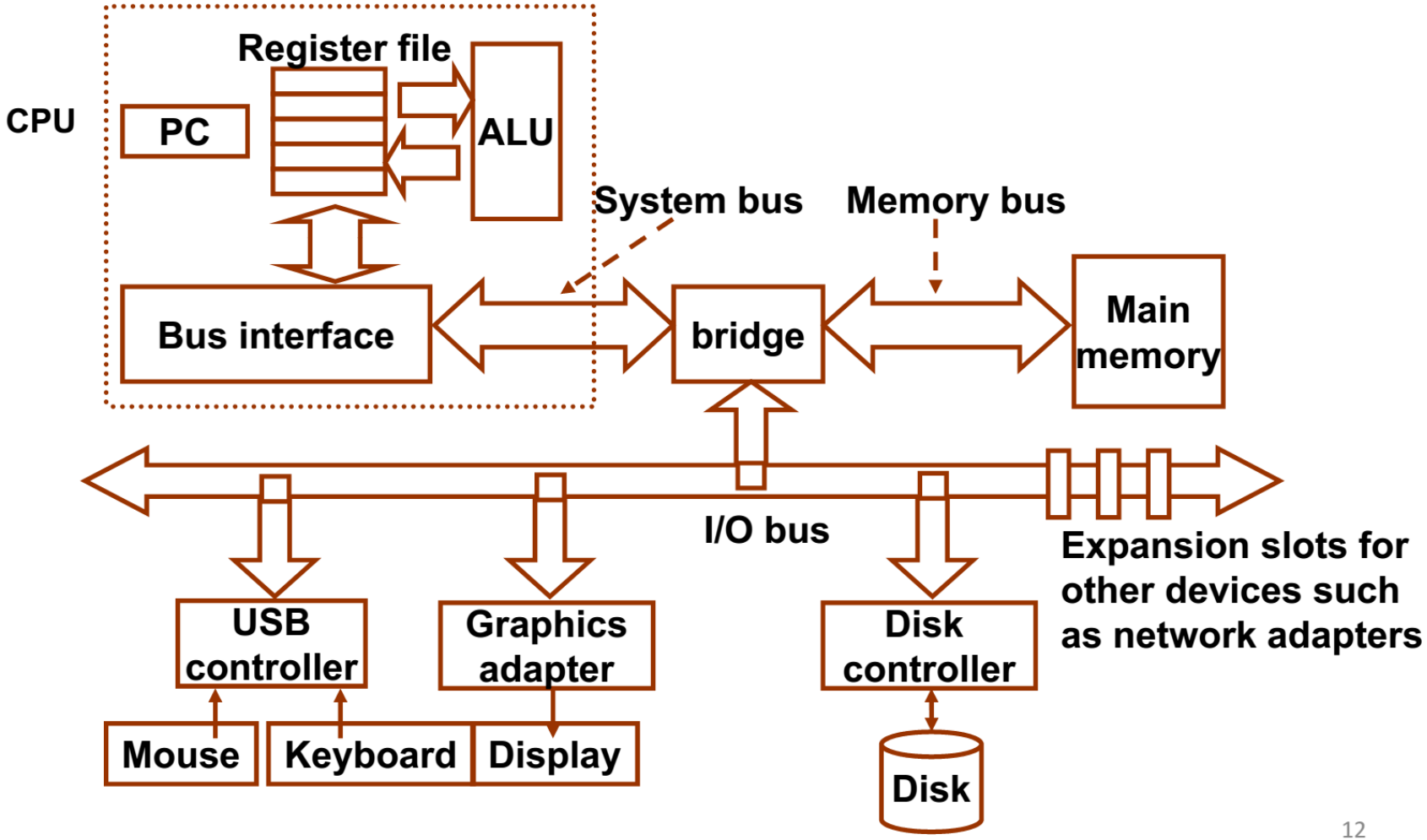
```
_sum:
      pushl %ebp
      movl %esp,%ebp
      movl 12(%ebp),%eax
      addl 8(%ebp),%eax
      movl %ebp,%esp
      popl %ebp
      ret
```

assembly

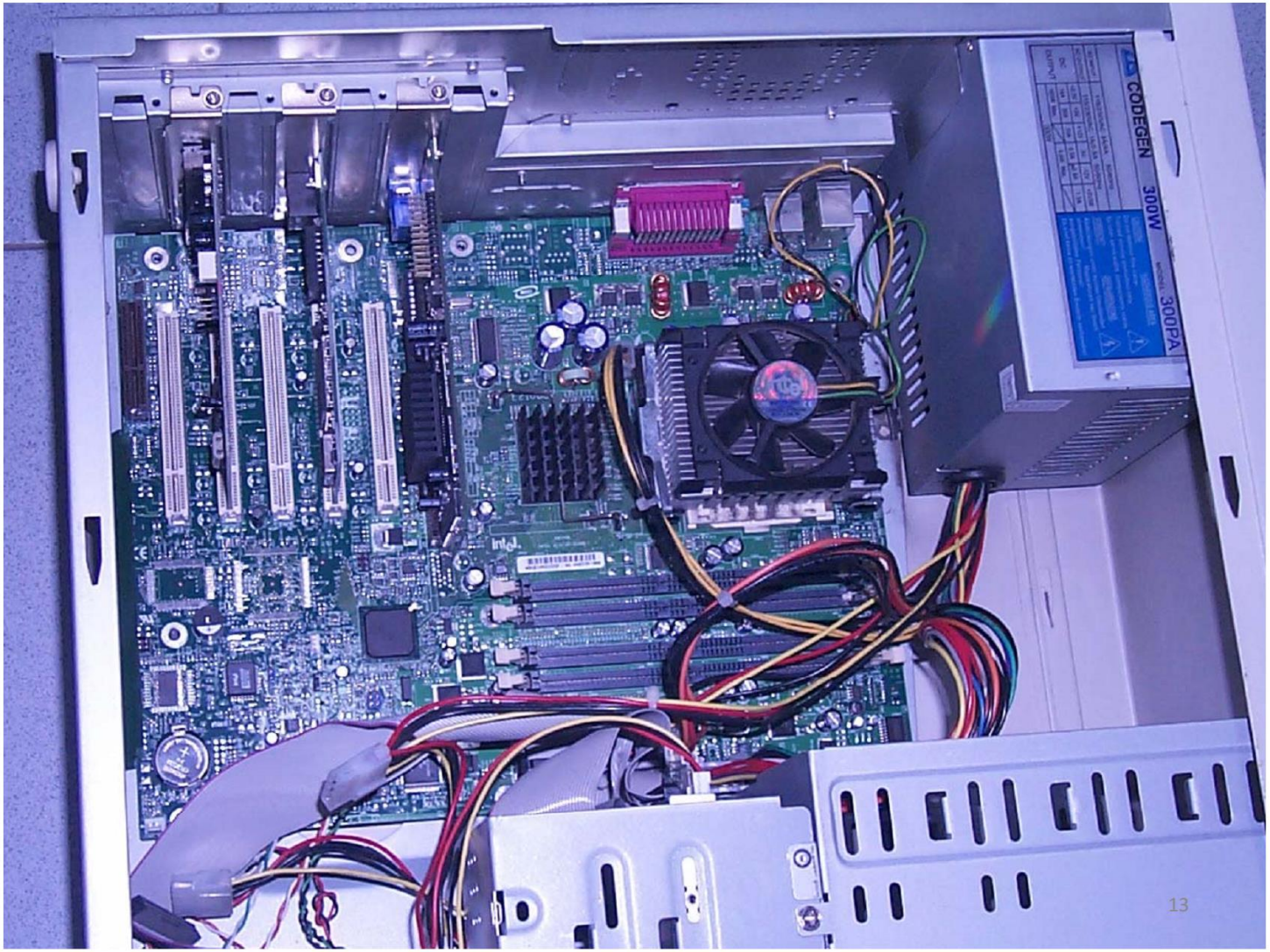
```
0x401040 <sum>:     0x55
                    0x89
                    0xe5
                    0x8b
                    0x45
                    0x0c
                    0x03
                    0x45
                    0x08
                    0x89
                    0xec
                    0x5d
                    0xc3
```

machine code

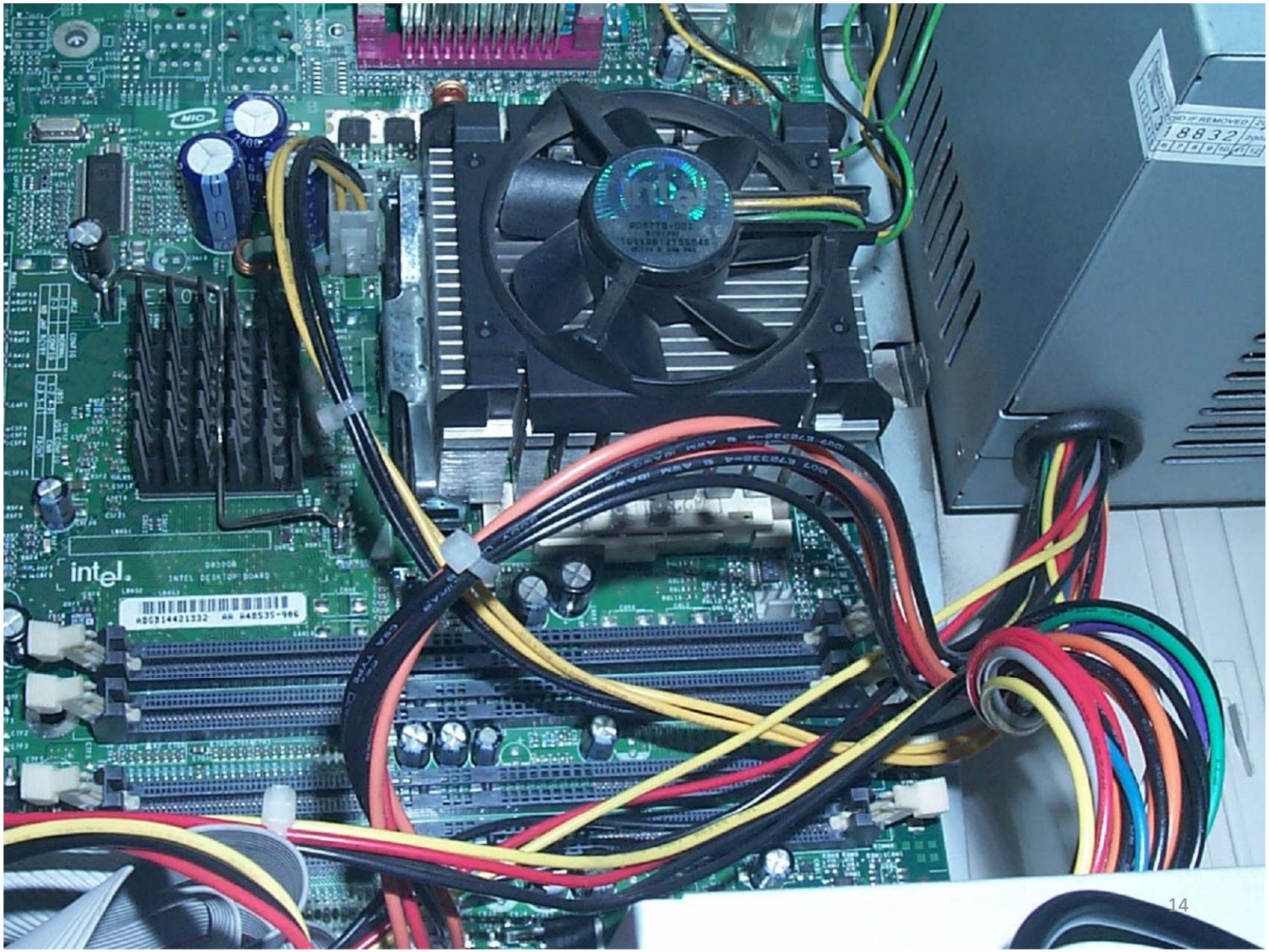
Hardware Abstraction



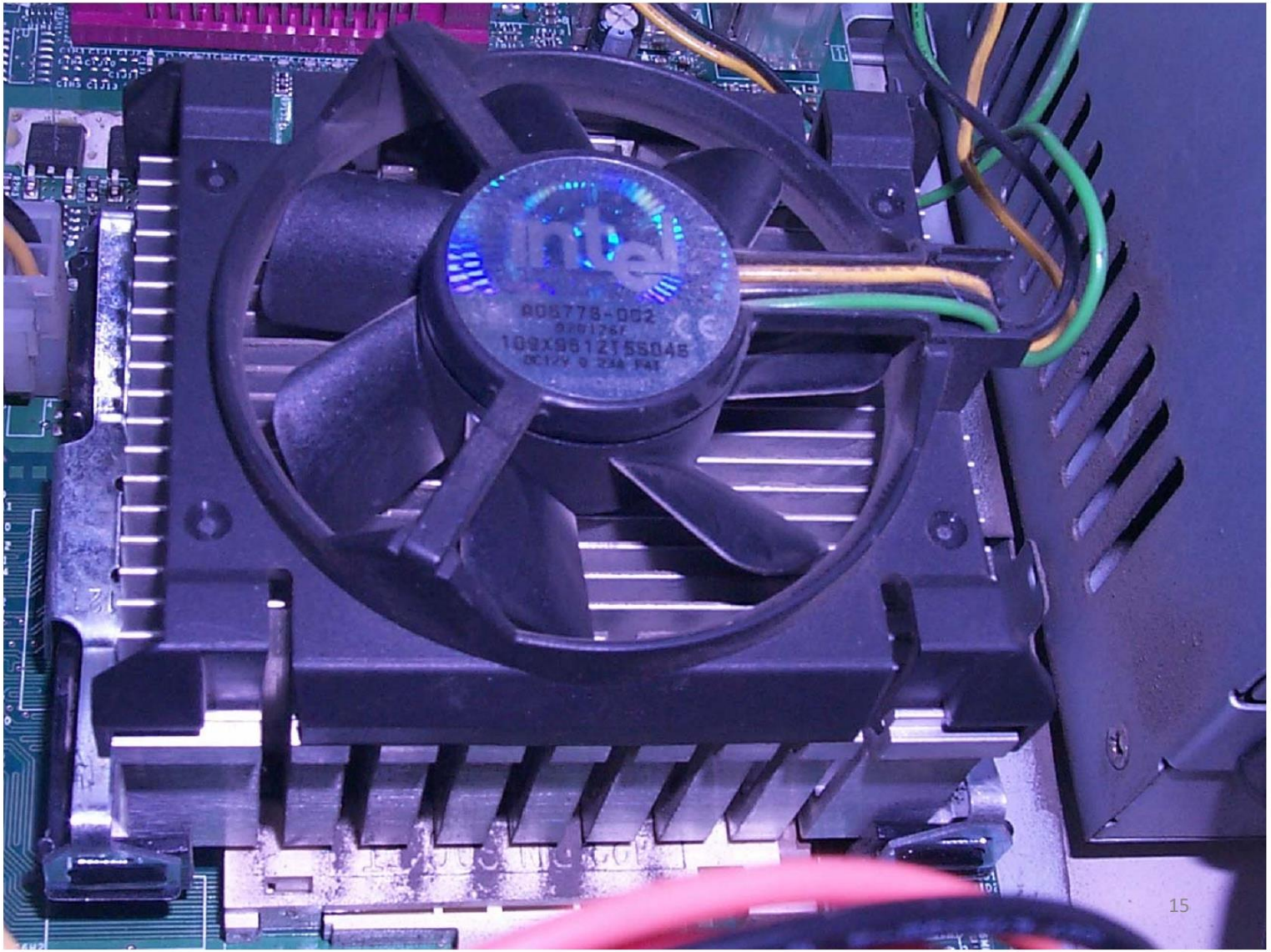
Real View



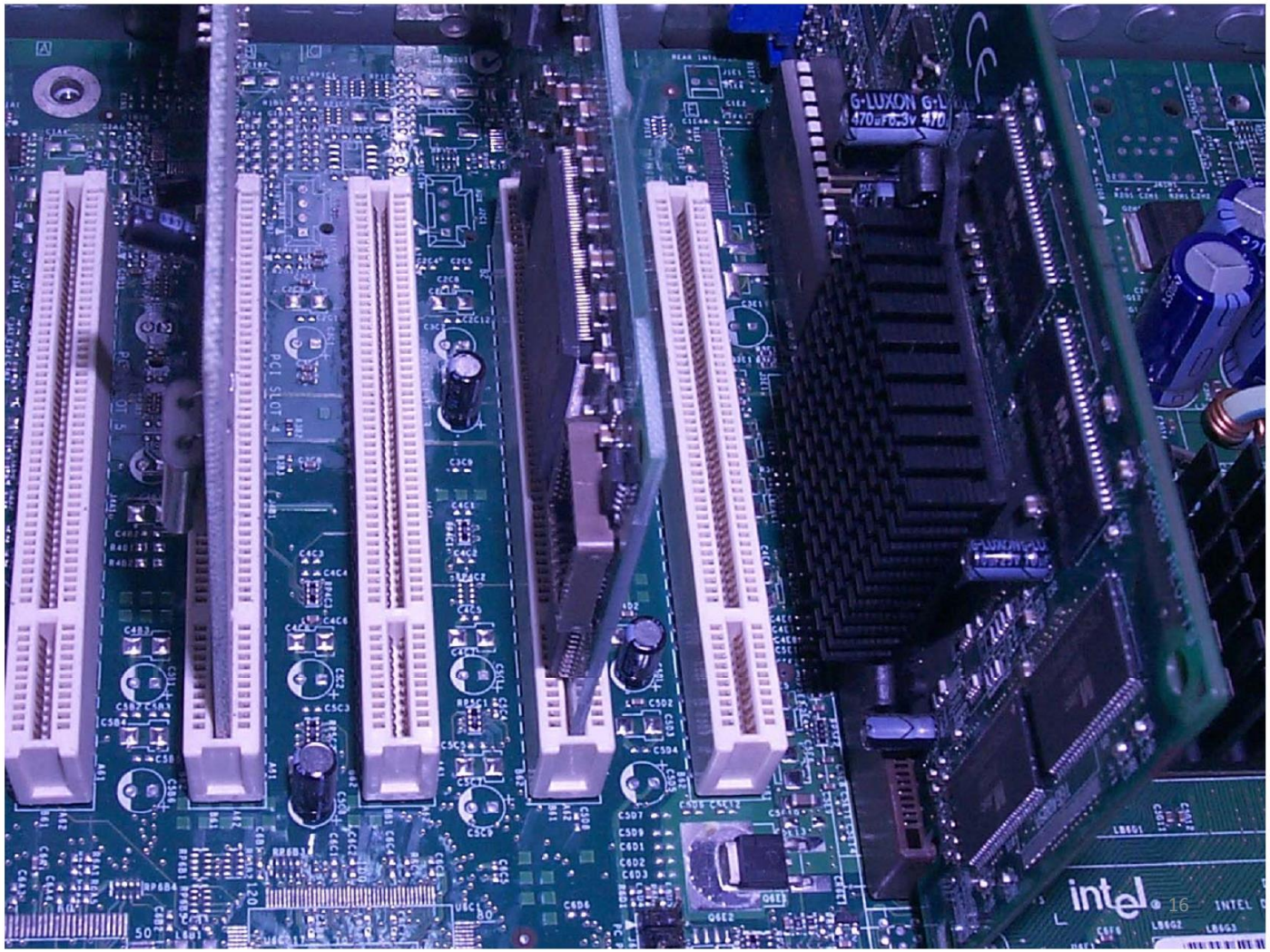
Another Real View



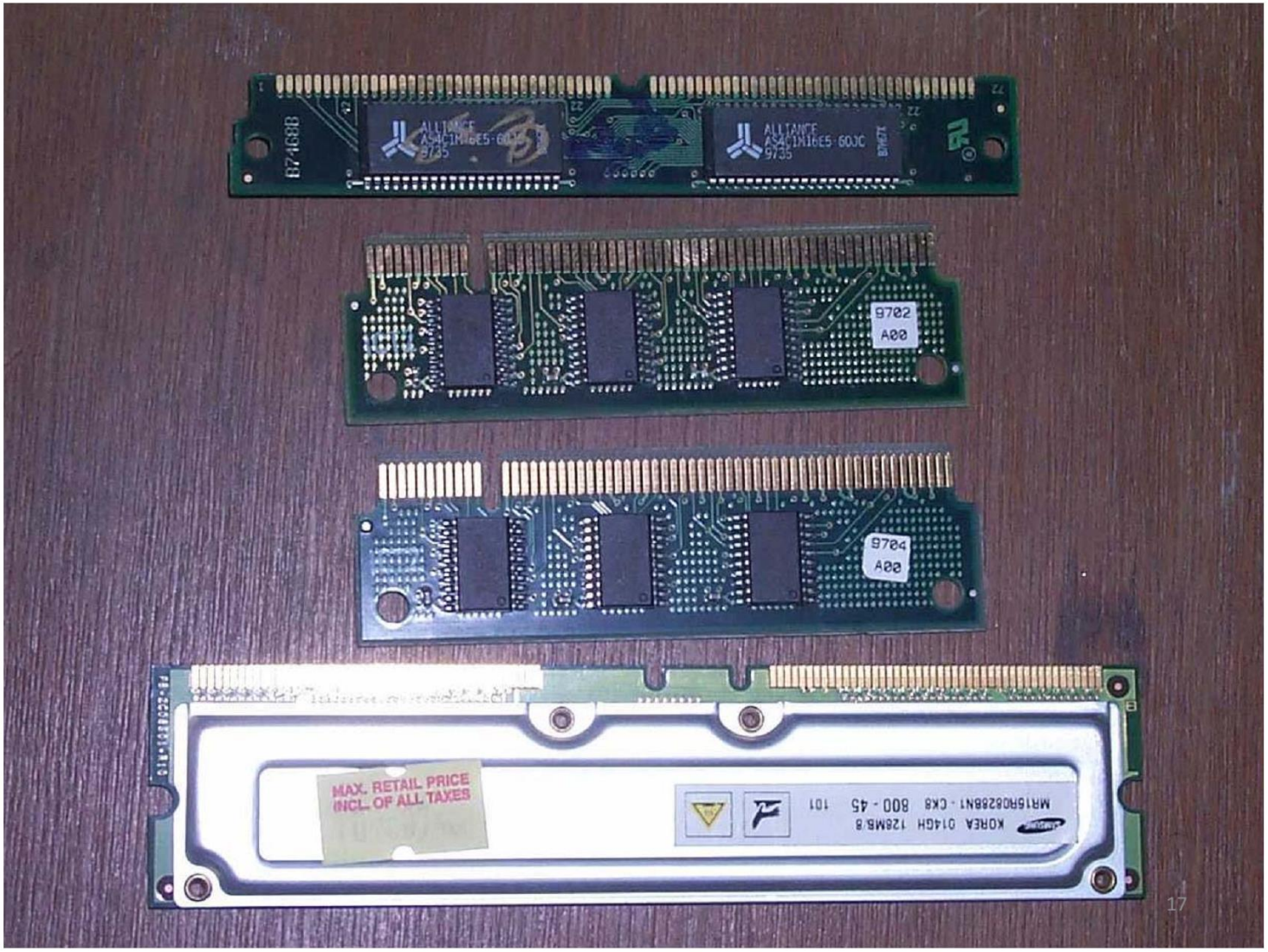
View 3 – Massive Cooler



Expanding Cards



Memories

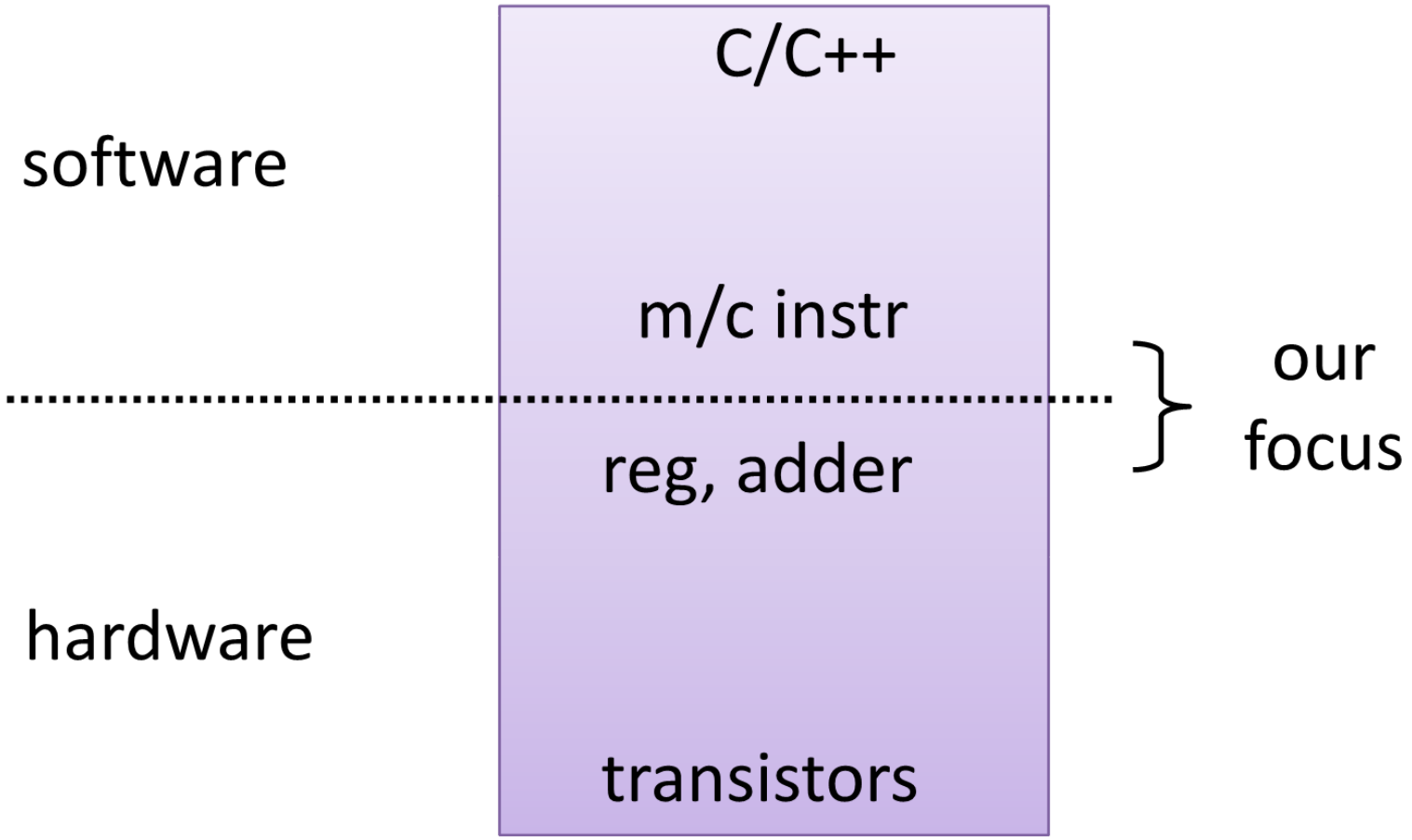

ALLIANCE
AS4C1M16E5-60JC
9735
9702
A00
9704
A00
MAX. RETAIL PRICE
INCL. OF ALL TAXES

Hard Discs


Seagate
WARNING
MODEL
ST3290A
915007-608
CBSHBLD16
07-5.04-A1
PRODUCT WARRANTY WILL BE VOID IF THIS LABEL IS REMOVED
Seagate
Barracuda ATA II
Model ST320420A
20.4 Gbytes
ULTRA ATA
FRAGILE
DRIVE PARAMETERS
MADE IN SINGAPORE

Two Sides of a HDD

Hardware/Software Interface

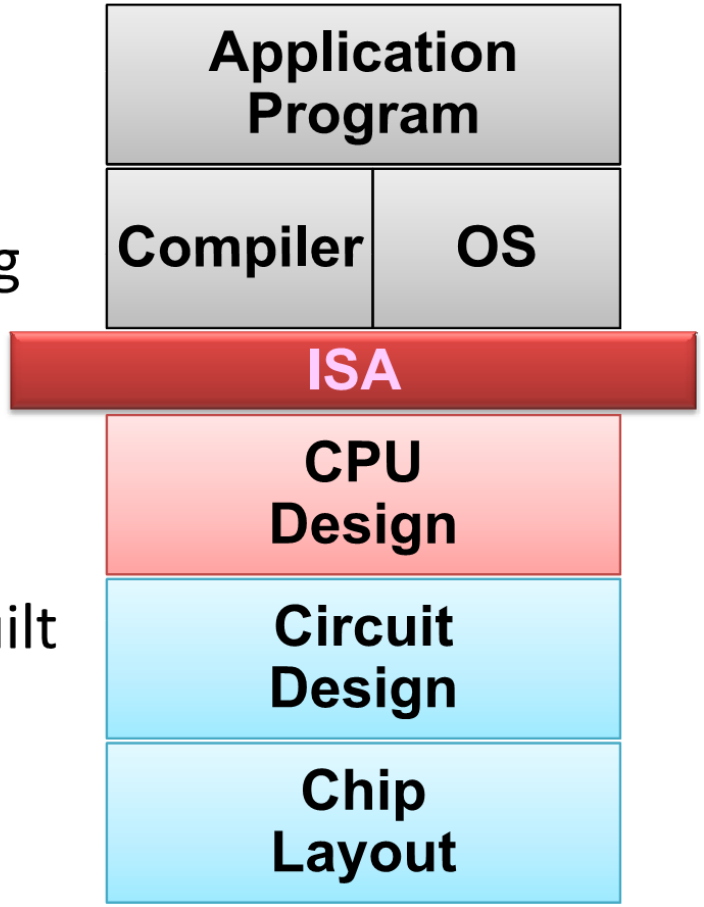


Levels of Architecture

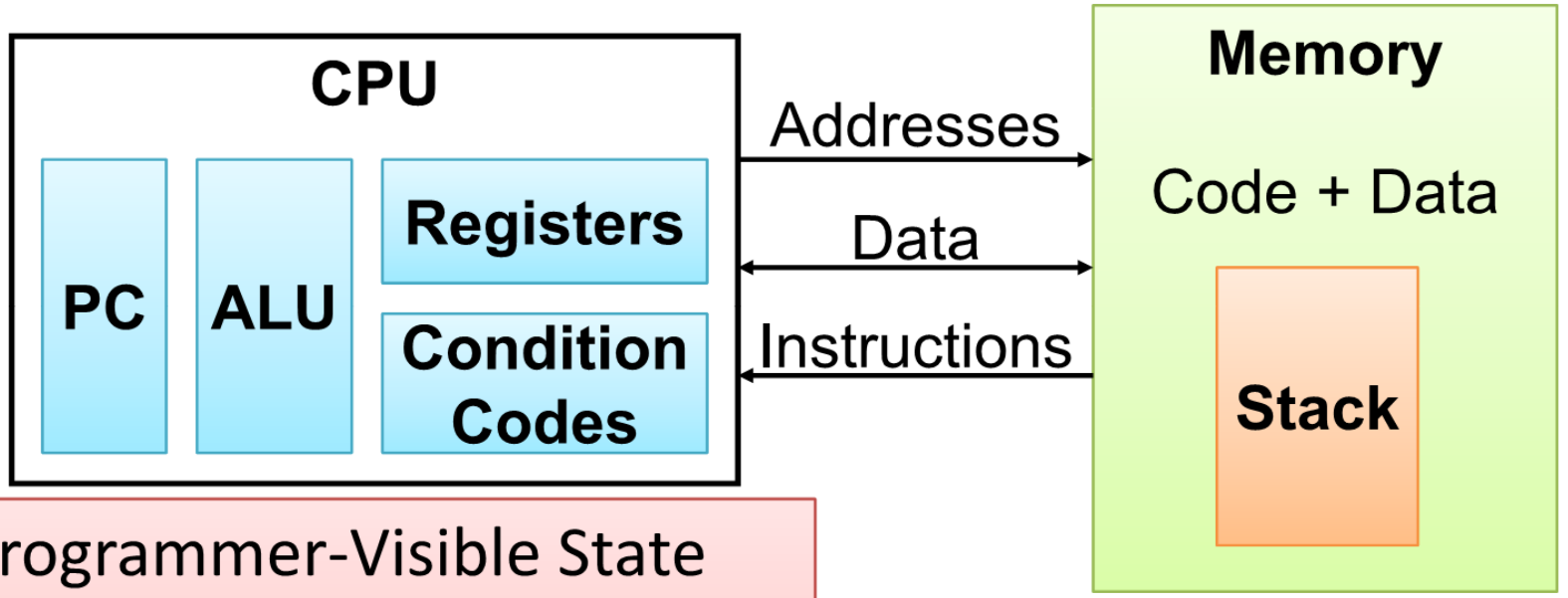
- Instruction set architecture (ISA)
  - Lowest level visible to a programmer
  - Operation (add/sub/mul/shift)
- Microarchitecture
  - Fills the gap between instructions and logic modules
  - Operation Vs Micro Operation

Instruction Set Architecture

- Assembly Language View
  - Processor state (RF, mem)
  - Instruction set and encoding
- Layer of Abstraction
  - Above: how to program machine - HLL, OS
  - Below: what needs to be built - tricks to make it run fast



The Abstract Machine



- Programmer-Visible State
  - PC Program Counter
  - Register File
    - Heavily used data
  - Condition Codes

- Memory
  - Byte array
  - Code + data
  - stack

Why Different Processors are Used?

- What is the difference between processors used in desktops, laptops, mobile phones, washing machines etc. ?
  - Performance/speed
  - Power consumption
  - Cost
  - General purpose/special purpose
  - Domain Specific: Network, DSP, Image.Crypto

The Embedded Processor

What is it?

A programmable processor whose programming interface is not accessible to the end-user of the product.

The only user-interaction is through the actual application.

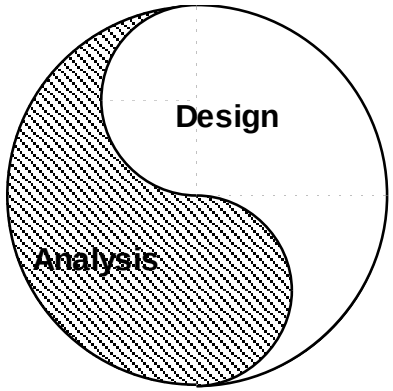
Examples:

- Sharp PDA's are encapsulated products with fixed functionality
- 3COM Palm pilots were originally intended as embedded systems. Opening up the programmers interface turned them into more generic computer systems.

Some interesting numbers

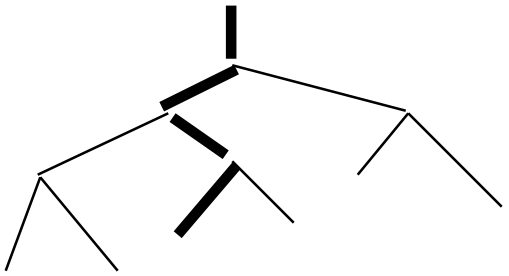
- The Intel 4004 was intended for an embedded application (a calculator)
- Of todays microprocessors
  - 95% go into embedded applications
    - SSH3/4 (Hitachi): best selling RISC microprocessor
  - 50% of microprocessor revenue stems from embedded systems
- Often focused on particular application area
  - Microcontrollers
  - DSPs
  - Media Processors
  - Graphics Processors
  - Network and Communication Processors

Architecture Design: Measurement and Evaluation

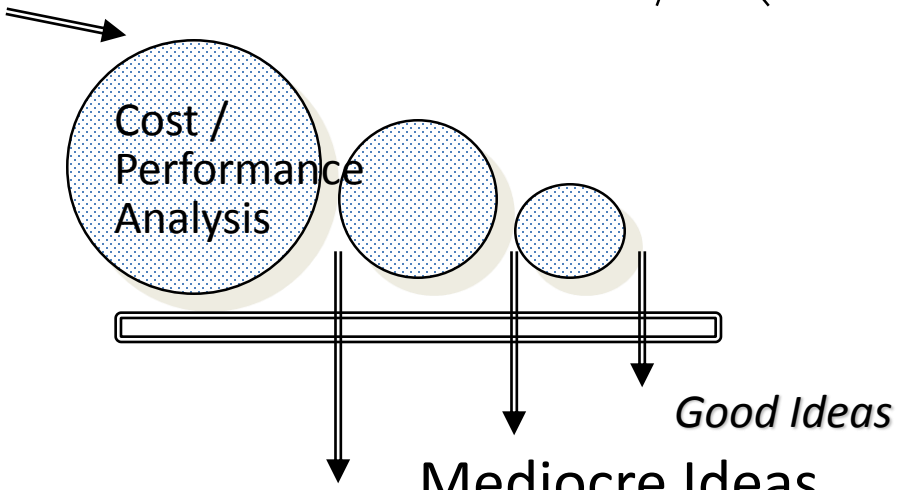


Architecture Design is an iterative process:

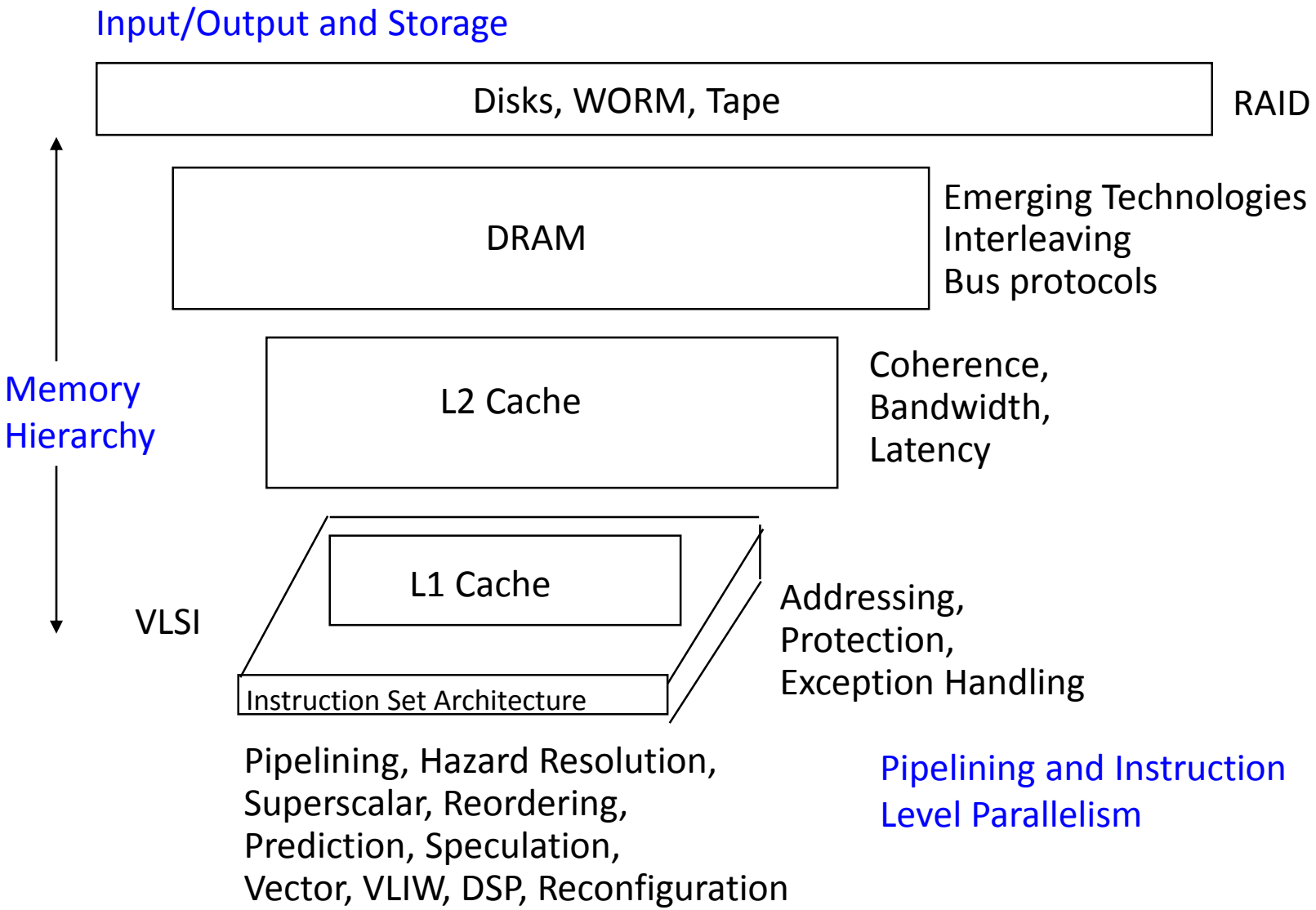
- Searching the space of possible designs
- At all levels of computer systems



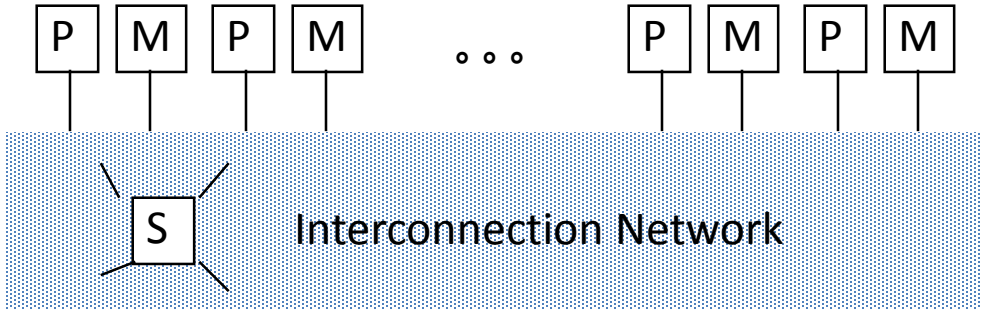
Creativity



Computer Architecture Topics



Computer Architecture Topics



Processor-Memory-Switch

Multiprocessors

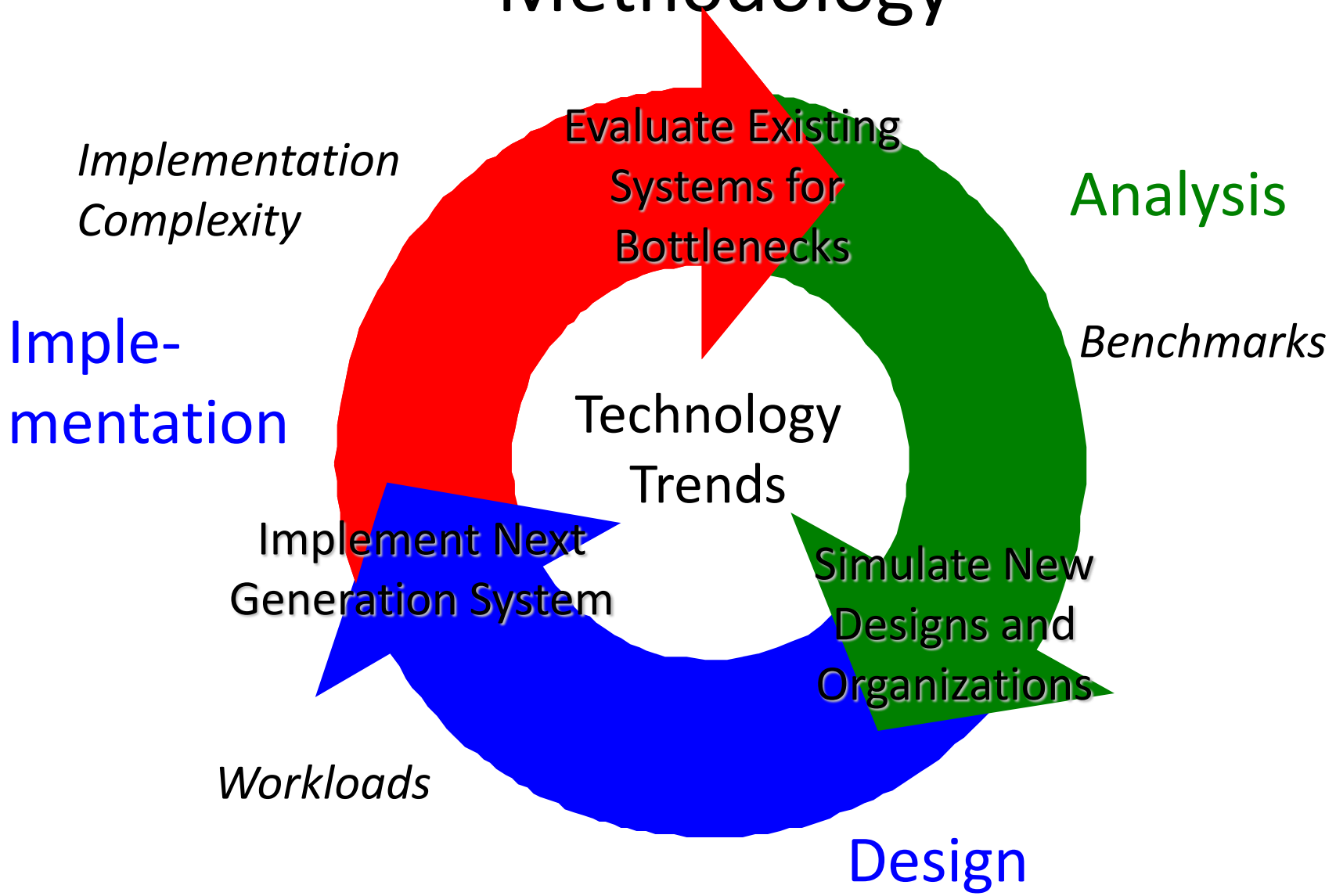
Networks and Interconnections

Shared Memory,
Message Passing,
Data Parallelism

Network Interfaces

Topologies,
Routing,
Bandwidth,
Latency,
Reliability

Computer Engineering Methodology



Measurement Tools

- Hardware: Cost, delay, area, power estimation
- Benchmarks, Traces, Mixes
- Simulation (many levels)
  - ISA, RT, Gate, Circuit
- Queuing Theory
- Rules of Thumb
- Fundamental “Laws”/Principles

Metric 1: Performance

In passenger-mile/hour

Plane	DC to Paris	Speed	Passengers	Throughput
Boeing 747	6.5 hours	610 mph	470	286,700
Concorde	3 hours	1350 mph	132	178,200

- Time to run the task
  - Execution time, response time, latency
- Tasks per day, hour, week, sec, ns …
  - Throughput, bandwidth

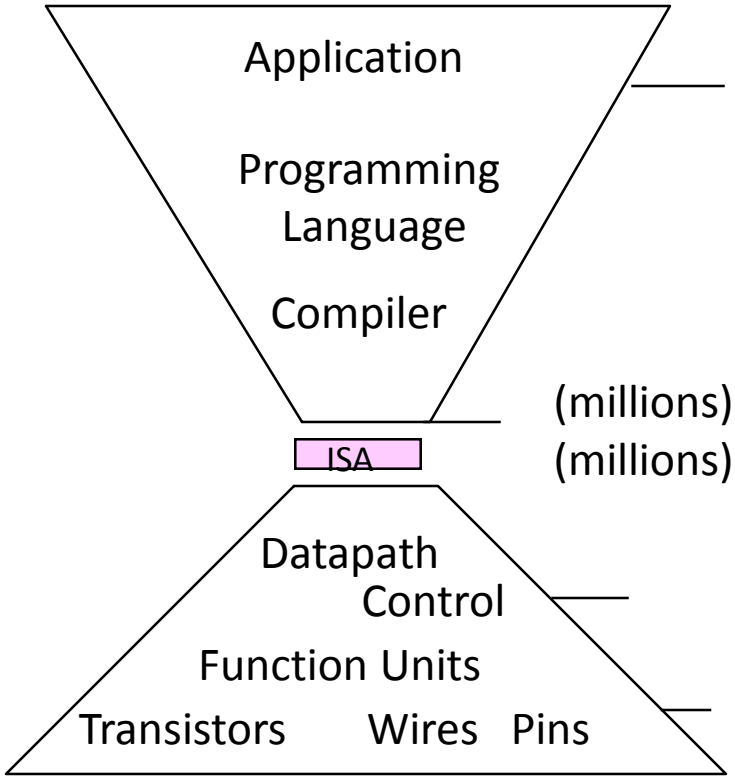
The Performance Metric

"X is n times faster than Y" means

$$\frac{\text{ExTime(Y)}}{\text{ExTime(X)}} = \frac{\text{Performance(X)}}{\text{Performance(Y)}}$$

- Speed of Concorde vs. Boeing 747
- Throughput of Boeing 747 vs. Concorde

Metrics of Performance



Answers per month
Operations per second

(millions) of Instructions per second: MIPS
(millions) of (FP) operations per second: MFLOP/s

Megabytes per second

Cycles per second (clock rate)

Creating Benchmark Sets

- Real programs
- Kernels
- Toy benchmarks
- Synthetic benchmarks
  - e.g. Whetstones and Dhrystones

SPEC: System Performance Evaluation Cooperative

- First Round 1989
  - 10 programs yielding a single number ("SPECmarks")
- Second Round 1992
  - SPECInt92 (6 integer programs) and SPECfp92 (14 floating point programs)
    - Compiler Flags unlimited. March 93 of DEC 4000 Model 610:

```
spice: unix.c:/def=(sysv,has_bcopy,"bcopy(a,b,c)=
       memcpy(b,a,c)"
wave5: /ali=(all,dcom=nat)/ag=a/ur=4/ur=200
nasa7: /norecu/ag=a/ur=4/ur2=200/lc=blas
```

- Third Round 1995
  - new set of programs: SPECint95 (8 integer programs) and SPECfp95 (10 floating point)
  - "benchmarks useful for 3 years"
  - Single flag setting for all programs: SPECint_base95, SPECfp_base95

Performance Evaluation

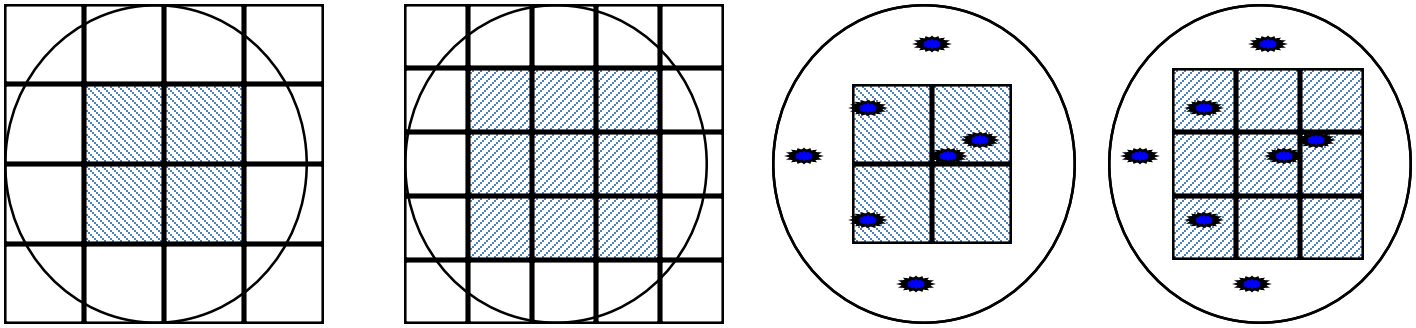
- “For better or worse, benchmarks shape a field”
- Good products created when have:
  - Good benchmarks
  - Good ways to summarize performance
- Given sales is a function in part of performance relative to competition, investment in improving product as reported by performance summary
- If benchmarks/summary inadequate, then choose between improving product for real programs vs. improving product to get more sales;
  Sales almost always wins!
- Execution time is the measure of computer performance!

Integrated Circuits Costs

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

$$\text{Die cost} = \frac{\text{Wafer cost}}{\text{Dies per Wafer} \times \text{Die yield}}$$

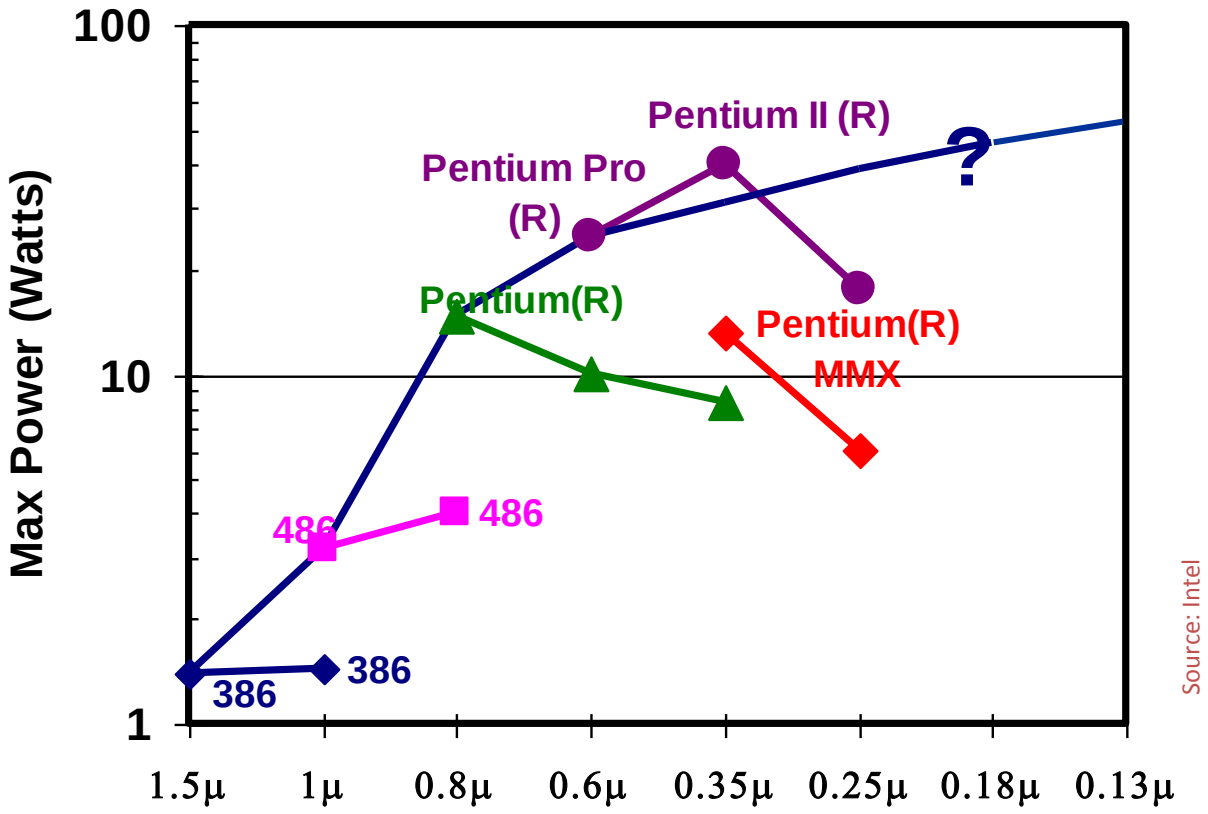
$$\text{Dies per wafer} = \frac{\pi\,(\text{Wafer\_diam}/2)^2}{\text{Die\_Area}} - \frac{\pi \times \text{Wafer\_diam}}{\sqrt{2 \cdot \text{Die\_Area}}} - \text{Test\_Die}$$



$$\text{Die Yield} = \text{Wafer\_yield} \times \left\{1 + \left(\frac{\text{Defect\_Density} \times \text{Die\_area}}{\alpha}\right)^{-\alpha}\right\}$$

Die Cost goes roughly with die area

Power/Energy



- Lead processor power increases every generation
- Compactions provide higher performance at lower power