

Fundamentals of Computers Design

Computer Architecture

J. Daniel Garcia

Computer Architecture Group.
Universidad Carlos III de Madrid

Last update: September 8, 2014

- 1 Introduction
- 2 Historic perspective
- 3 Classification
- 4 Parallelism
- 5 Computer Architecture
- 6 Conclusion

Computer Architecture



Source: http://commons.wikimedia.org/wiki/File:Amdahl_march_13_2008.jpg

*The term **architecture** is used here to describe the attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls, the logical design, and the physical implementation.*

Gene Amdahl et al.
Architecture of the IBM System.
IBM Journal of Research and Development
Vol 8 (2) pp. 87-101. 1964.

What is Computer Architecture?

- **Computer Architecture** is the science and art of *selecting and interconnecting* hardware components to create computers that meet **functional, performance and cost** goals.

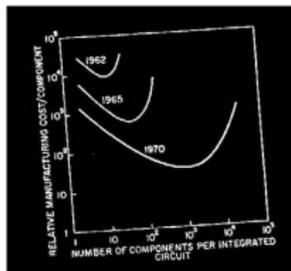
- Computer architecture is not about using computers to design buildings. 😊



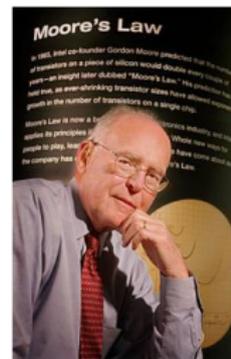
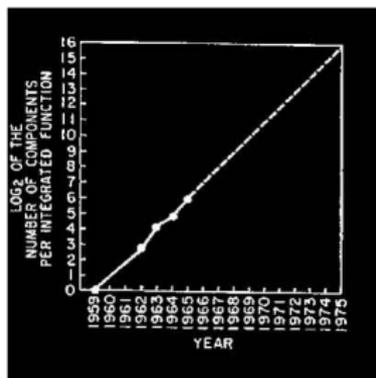
Why study Computer Architecture

- No computers \Rightarrow No Computer Engineers.
- To understand trends for next decade.
 - How will computers be in the future?
 - What will they be able to do?
- To understand computers limitations.
 - What can be done? What can't be done?
 - Where are performance limitations?
- To pass this course

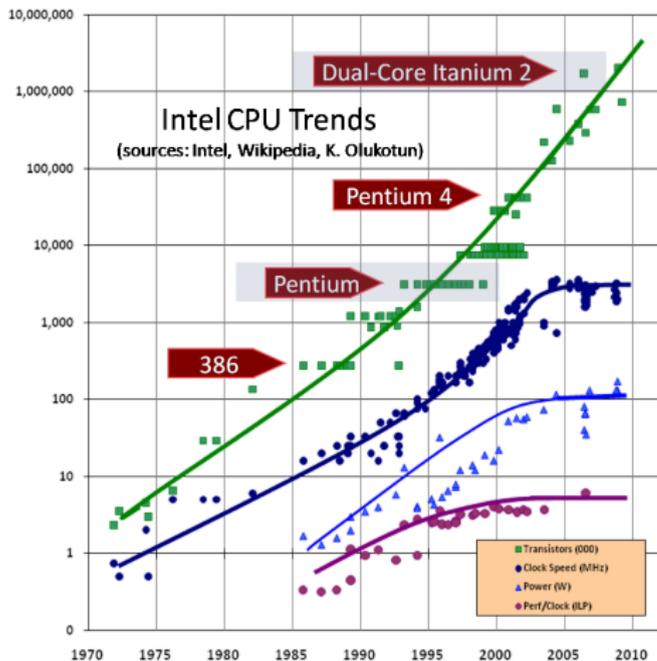
The Moore's Law



- Number of transistors per chip doubles every N months.
 - With $12 < N$



Transistors per chip

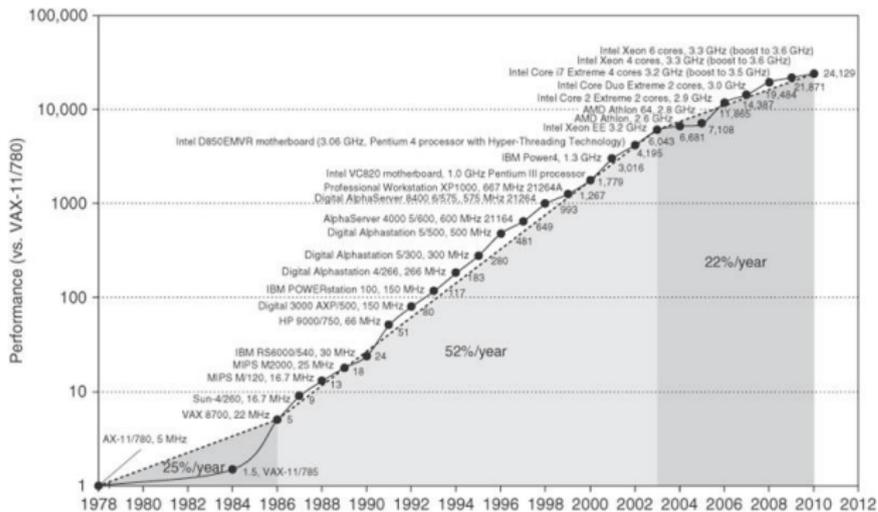


- Activity: Please, read the full article.

Source: **The free lunch is over.**
Herb Sutter.

<http://www.gotw.ca/publications/concurrency-ddj.htm>

Performance evolution

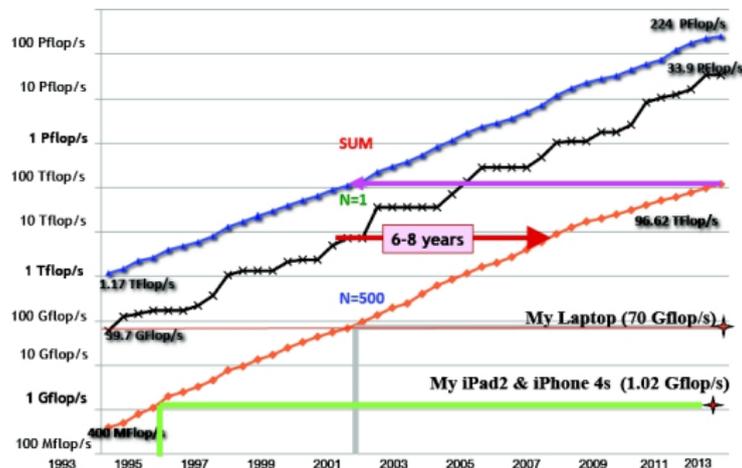


Source: Computer Architecture: A Quantitative Approach. 5 Ed
Hennessy and Patterson. Morgan Kaufmann. 2012.

- 1986:
RISC.
- 2005:
multi-core.

Effects of RISC emergence

- Available capacity improvement.
 - A high-end microprocessor more powerful than a supercomputer ten years before.



Source: *An overview of High Performance Computer and Future Requirements*. Jack Dongarra. SC13. 2013.

The RISC revolution

- Continuous semiconductor improvement led to microprocessor based computers to become dominant.
 - Minicomputers disappear.
 - Mainframes and supercomputers built as a collection of microprocessors.
- Sustained increase in performance from 1986 to 2003:
 - **52% per year.**

- **No longer true!**



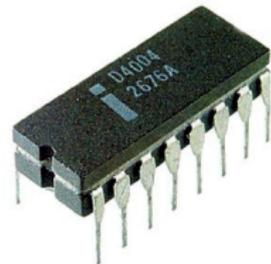
- 1 Introduction
- 2 Historic perspective
- 3 Classification
- 4 Parallelism
- 5 Computer Architecture
- 6 Conclusion

First revolution: Microprocessor

- The microprocessor revolution.
 - Generated from a single change.
 - Enough transistors (25,000) in a single chip for a 16-bit processor.
 - Advantages:
 - Faster: Less need to get out from the chip.
 - Cheaper: All in one chip.
- New market segments generated by the innovation.
 - Desktop computers, CD/DVD, laptops, video-game consoles, set-top-boxes, digital cameras, MP3, GPS, ...
- Impact on existing markets.
 - Supercomputers, mainframes, ...

First microprocessor

- Intel 4004 (1971).
 - Application domain: Calculators.
 - Technology: 10,000 nm.
 - 2300 transistors.
 - 13 mm²
 - 108 KHz
 - 12 Volts
 - 4-bits data.
 - Data-path in one cycle.



Intel 4004 by Rostislav Lisovy



Unicom 141P Calculator 3 by Michael Holley.



Second revolution

- Extract implicit instruction-level parallelism (ILP).
 - Hardware has resources that can be used in parallel.

- Elements:
 - **Pipelining**: Allowed increasing clock frequency.
 - **Caches**: Needed to increase those frequencies.
 - **Floating point**: Integrated into the chip.
 - Increasing pipeline **depth** and **speculative branching**.
 - **Multiple issue**: superscalar architectures.
 - **Dynamic scheduling**: Out of Order execution.

Top single core processors



Die of Intel Pentium 4 (Northwood)

Source:

<http://gecko54000.free.fr>



- Application domain: Desktop / Servers.
- Technology: 90 nm (1/100x).
- 55M transistors (20,000x).
- 101 mm² (10x).
- 3.4 GHz (10,000x).
- 1.2 Volts (1/10x).
- 32/64-bit data (16x).
- Data path with 22 pipeline stages (later 31).
- 3-4 instructions per cycle (superscalar).
- Two level cache on chip.
- Data parallel instructions (SIMD).
- Hyper-threading.



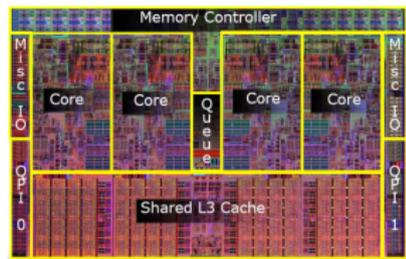
Third revolution

- Explicitly support thread-level and data-level parallelism.
 - Hardware offers parallel resources and software specifies its use.
 - Parallelism no longer hidden by hardware.
 - **Rationale**: Diminishing returns from ILP.

- Elements:
 - Vector instructions: Intel SSE.
 - General support for multi-threaded applications.

Multi-core processors

- Intel Core i7 (2009).
 - Application: Desktop / Server.
 - Technology: 45 nm (1/2x).
 - 774M transistors (12x).
 - 296 mm² (3x).
 - 3.2 GHz – 3.6 GHz ($\approx 1x$).
 - 0.7 – 1.4 Volts ($\approx 1x$).
 - 128-bit data (2x).
 - Datapath with 14-stages pipeline (0.5x).
 - 4 instructions per cycle ($\approx 1x$).
 - Three level cache on chip.
 - Data parallel instructions (SIMD).
 - Hyper-threading.
 - 4 cores (4x).



Die of Intel Core i7 (Nehalem)

Source: www.legitreviews.com

Architecture trends

- Instruction-Level Parallelism.
 - Parallel execution of instructions.
 - Impossible to improve Instruction-Level parallelism (since 2003-2005).
 - Hardware and compiler conspiracy to hide details to programmer.
 - Programmer with very simplified view of hardware.
- New models to improve performance.
 - Data-Level Parallelism (DLP).
 - Thread-Level Parallelism (TLP).
 - Request-Level Parallelism (RLP).
- **IMPORTANT**: All of them require to restructure applications to take advantage of promised performance increase.



- 1 Introduction
- 2 Historic perspective
- 3 Classification**
- 4 Parallelism
- 5 Computer Architecture
- 6 Conclusion

Personal Mobile Devices

- Wireless devices with multimedia UI.
 - Mobile devices, tablets, ...

- Price: \$100 – \$1000.
- Processor price: \$10 – \$100.
- Critical factors:
 - Cost.
 - Energy.
 - Media performance.
 - Responsiveness.

Desktop

- Designed to offer good performance to end-user.
 - From ultra-books to workstations.
 - Since 2008 more than 50% are laptops.
- Price: \$300 – \$2500.
- Processor price: \$50 – \$500.
- Critical factors:
 - Price-Performance.
 - Energy.
 - Graphics performance.

Servers

- Used to run large scale applications and give service to multiple users simultaneously.
 - Growing since 80's replacing mainframes.
- Price \$5,000 – \$10,000,000.
- Processor price: \$200 – \$2,000.
- Critical factors:
 - Throughput.
 - Availability.
 - Scalability.
 - Energy.

Clusters / Warehouse Scale Computers (WSC)

- A collection of computers connected through a LAN and acting as a larger computer.
 - Made much more popular by SaaS (Software as a Service) growth.
 - Each node runs its own operating system.
 - WSC → 10,000+ nodes.
- Price: \$100,000 – \$200,000,000.
- Processor price: \$50 – \$250.
- Critical factors:
 - Price-Performance.
 - Throughput.
 - Energy proportionality.

Embedded

- Computer within another system to run pre-established applications.
 - Dish-washer, video-game console, MP3, ...

- Processor price: \$0.01 – \$100.
- Critical factors:
 - Price.
 - Energy.
 - Application-specific performance.

Sales (2010)

Class	Units Sold
Portable Mobile Devices	1,800,000,000
Desktop	350,000,000
Servers	20,000,000
Embedded	19,000,000,000



- 1 Introduction
- 2 Historic perspective
- 3 Classification
- 4 Parallelism
- 5 Computer Architecture
- 6 Conclusion

The speed of a serial computer

- A 1 TFLOP (10^{12} FLOPS) sequential machine:
 - Data must travel a certain distance (r) from memory to CPU.
 - 1 data element per cycle $\Rightarrow 10^{12}$ times per second $\Rightarrow 10^{-12}$ s. per cycle.
 - Data traveling at light speed: $c = 3 \cdot 10^8$ m/s.
 - $r = 3 \cdot 10^8 \cdot 10^{-12} = 0.3\text{mm}$
- 1 TB of data in 0.3 mm^2 surface:
 - Each data to be stored in 3 Angmstroms (approx.)
 - The size of a small atom!
- Sequential approach not feasible.



Parallelism Kinds

- All restrictions have cost and energy restrictions.
- Parallelism emerges as main mechanism to design computers.
- Application parallelism kinds:
 - **Data parallelism:** Same operation applied to many data.
 - **Task parallelism:** Tasks operate independently and in parallel.



Hardware parallelism

- **ILP**: Instruction-Level Parallelism.
 - Exploits data-parallelism with compiler help (pipeline, speculative execution, ...).
- **Vector** architectures and GPUs.
 - Exploits data parallelism applying the same instruction to several data in parallel.
- **TLP**: Thread-Level Parallelism.
 - Exploits data or task parallelism in highly coupled hardware.
 - Allows inter-thread interactions.
- **RLP**: Request-Level Parallelism.
 - Exploits parallelism among highly decoupled tasks.

Flynn Taxonomy (1966)

- A classification of possible parallel architectures.
- **SISD**: Single Instruction / Single Data Stream.
 - Mono-processor.
 - May use ILP techniques.
- **SIMD**: Single Instruction / Multiple Data Stream.
 - Same instructions executed by different processors on distinct data items.
 - *Alternatives*: Vector processors, multimedia extensions, and GPUs.
- **MISD**: Multiple Instructions / Single Data Stream.
 - No known commercial implementation.
- **MIMD**: Multiple Instructions / Multiple Data Stream.
 - Each processor operates on its own data items \Rightarrow Task Parallelism.



More on MIMD

- Variety in MIMD architectures:
 - Highly coupled architectures.
 - **Thread-Level Parallelism**: Multi/Many-core architectures.
 - Weakly coupled architectures:
 - **Request-Level Parallelism**: Clusters and WSCs.

- MIMD is:
 - More flexible and general than SIMD.
 - More expensive than SIMD.
 - Requires enough task granularity.

- 1 Introduction
- 2 Historic perspective
- 3 Classification
- 4 Parallelism
- 5 Computer Architecture**
- 6 Conclusion

Architecture Visions

- Computer design is complex.
 - Determine important attributes.
 - Maximize performance and energy efficiency with control over cost, power, and availability.

- Different visions on architecture design:
 - Instruction set design.
 - Functional organization.
 - Logic design.
 - Implementation: Circuit design, packaging, cooling, . . .
 - Integration with compilers, operating systems, . . .

ISA: Instruction Set Architecture

- ISA: Part of the architecture which is visible to the programmer.
 - Available instructions.
 - Registers number and type.
 - Instructions format.
 - Addressing modes.
 - Exception conditions.

ISA Kinds

- Almost every current ISA use general purpose registers.
- Popular versions:
 - **Register-Memory** ISA:
 - Many instructions may access main memory.
 - e.g.: Intel 80x86.
 - **Load-Store** ISA:
 - Only *load/store* instructions can access main memory.
 - Other instructions operate on registers.
 - e.g.: ARM, MIPS.
- Most recent ISAs are *load-store*.

Memory addressing

- Types:
 - *Byte addressing*: Address identifies a byte within memory.
 - *Word addressing*: Address identifies a word within memory.

- *Most popular addressing*: Byte addressing.
- Variants:
 - All accesses must be aligned (e.g. MIPS).
 - Aligned accesses are faster (e.g. 80x86).

Addressing modes

■ MIPS:

- Register.
- Immediate.
- Relative to register.

■ 80x86:

- Register, immediate, relative to register.
- Absolute.
- Base register with displacement (2 registers).
- Base register with scaled index and displacement. . . .

Operand types and sizes

■ Integers:

- 8 bits (character).
- 16 bits (Unicode character).
- 32 bits (integer / word).
- 64 bits (long integer / double word).

■ Floating point:

- 32 bits (single precision).
- 64 bits (double precision).
- 80 bits (80x86 extended double precision).

Operations

- **Categories:**
 - Data transfer, arithmetic, logic, control, and floating point.

- **MIPS:**
 - Simple instructions, easy to implement in a *pipeline* (RISC).

- **80x86:**
 - Many more instructions.
 - Variable complexity.

Flow control instructions

■ Categories:

- Conditional branching, jumps, procedure calls and returns.

■ MIPS:

- PC-relative addressing.
- Conditions on register values.
- Procedure calls place return value in register.

■ 80x86:

- PC-relative addressing.
- Conditions on bits with condition codes.
- Procedure calls through stack.

Instruction coding

■ Fixed length

- MIPS.
- All instructions are 32 bit.
- Simplified instruction decode.

■ Variable length

- 80x86.
- Length from 1 to 18 bytes.
- Reduces program size.

- 1 Introduction
- 2 Historic perspective
- 3 Classification
- 4 Parallelism
- 5 Computer Architecture
- 6 Conclusion

Summary

- Moore's law still alive.
 - But *"The free lunch is over"*.
- New models improve performance (DLP, TLP, RLP) but require application restructuring.
- Diversity in computer classes with varied properties and requirements.
 - PMD, Desktop, Servers, WSC, Embedded.
- Emerging architectures combining SIMD and MIMD.
- ISA is different from *architecture*.

References

- **Computer Architecture. A Quantitative Approach**

5th Ed.

Hennessy and Patterson.

Sections 1.1, 1.2, and 1.3.

- **The Free Lunch is over.**

Herb Sutter.

<http://www.gotw.ca/publications/concurrency-ddj.htm>

- **Welcome to the Jungle.**

Herb Sutter.

<http://herbsutter.com/welcome-to-the-jungle/>



Fundamentals of Computers Design

Computer Architecture

J. Daniel Garcia

Computer Architecture Group.
Universidad Carlos III de Madrid

Last update: September 8, 2014