

Introduction to: 18-740 "Computer Architecture" 18-640 "Foundations of Computer Architecture"

Prof. Onur Mutlu (18-740)
Prof. John Paul Shen (18-640)
August 31, 2015



18-640/18-740 Lecture 0

Carnegie Mellon University 1

18-740 vs. 18-640

RESEARCH PAPERS	INDUSTRY PREPARATION
ONE BIG PROJECT	SEVERAL PROJECTS
RECITATIONS	RECITATIONS
LECTURES	LECTURES
Prerequisite: 18-447 (CMU)	Prerequisite: Comp. Org. (BS)

← Shared



18-640/18-740 Lecture 0

Carnegie Mellon University 2

18-740 Instructor: Onur Mutlu

- Associate Professor @ Carnegie Mellon University ECE/CS
- PhD from UT-Austin 2006, BS from Michigan 2000
- Past experience @ Microsoft Research, Intel, AMD
- omutlu@gmail.com (Best way to reach me)
- <http://www.ece.cmu.edu/~omutlu>
- <http://users.ece.cmu.edu/~omutlu/projects.htm>
- **Research and Education** in
 - Computer architecture and systems, bioinformatics
 - Memory and storage systems, emerging technologies
 - Many-core systems, heterogeneous systems, core design
 - Interconnects
 - Hardware/software interaction and co-design (PL, OS, Architecture)
 - Predictable and QoS-aware systems
 - Hardware fault tolerance and security
 - Algorithms and architectures for genome analysis
 - ...



3

18-740 Teaching Assistant: Nandita

- Nandita Vijaykumar
 - PhD student with Onur Mutlu
 - BE from PES Inst. Of Technology 2011
 - Past experience @ AMD
 - nandita@cmu.edu
- Office hours and locations will be posted online
 - <http://www.ece.cmu.edu/~ece740/f15>
- Reach all of us at
 - 740-official@ece.cmu.edu



4

18-640 Cast of Characters

- **Instructor:** John Paul Shen (SV)
- **Academic Services Assistants:**
 - Zara Collier (PGH)
 - Stephanie Scott (SV)
- **Teaching Assistants:**
 - Priyank Sanghavi (PGH) } • Cover PGH & SV sections
 - Guanglin Xu (PGH) } • Cover GZ & PGH sections
 - Adeola Bannis (SV) } • Cover SV section
 - Revathy Shunmugam (PGH) } • Cover PGH & GZ sections
- **JIE Course Coordinator:** Ziyang He (GZ)



18-640/18-740 Lecture 0

Carnegie Mellon University 5

My Personal Background:

- **Academia**
 - Carnegie Mellon University
 - First Half → Computer Aided Design
 - Sabbatical at Stanford
 - Second Half → Computer Architecture
 - Sabbatical at Intel
- **Industry**
 - Intel, Research Lab
 - Microarchitecture Lab → Microprocessor Design
 - Nokia, Research Center
 - North America Lab → Mobile Computing System



18-640/18-740 Lecture 0

Carnegie Mellon University 6

18-640/18-740 Computer Architecture

Lecture 1: “Introduction To Computer Architecture”

John Paul Shen
August 31, 2015

➤ Required Reading Assignment:

- Chapters 1 and 2 of Shen and Lipasti (SnL).

➤ Recommended References:

- ❖ “Amdahl’s and Gustafson’s Laws Revisited” by Andrzej Karbowski. (2008)
- ❖ “High Performance Reduced Instruction Set Processors” by Tilak Agerwala and John Cocke. (1987)



Electrical & Computer
ENGINEERING

18-640/18-740 Computer Architecture

Lecture 1: “Introduction To Computer Architecture”

A. Instruction Set Architecture (ISA)

- Hardware / Software Interface
- Dynamic / Static Interface (DSI)

B. Historical Perspective on Computing

- Major Epochs
- Processor Performance Iron Law (#1)
- Course Coverage

C. “Economics” of Computer Architecture

- Amdahl’s Law and Gustafson’s Law
- Moore’s Law and Bell’s Law



Electrical & Computer
ENGINEERING

Anatomy of Engineering Design



Specification: Behavioral description of *“What does it do?”*

Synthesis: Search for possible solutions; pick best one. *Creative process*

Implementation: Structural description of *“How is it constructed?”*

Analysis: Validate if the design meets the specification.

“Does it do the right thing?” + “How well does it perform?”

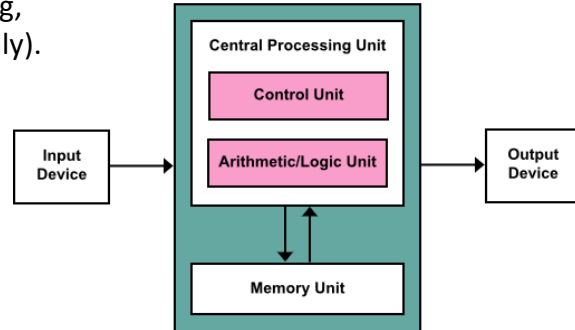
Lecture 1: “Introduction to Computer Architecture”

A. Instruction Set Architecture (ISA)

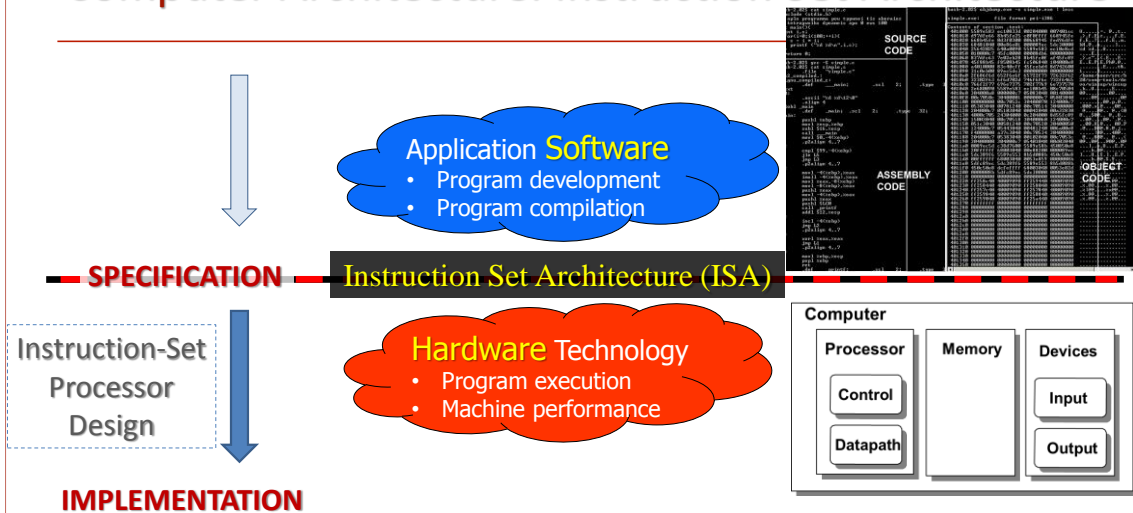
- a. Hardware / Software Interface
- b. Dynamic / Static Interface (DSI)

The Classic Von Neumann Computation Model

- Proposed in 1945 by John Von Neumann and others (Alan Turing, J. Presper Eckert and John Mauchly).
- A “Stored Program Computer”
 - One CPU**
 - One Control Unit
 - Program Counter
 - Instruction Register
 - One ALU
 - Monolithic Memory**
 - Data Store
 - Instruction Store
 - Sequential Execution Semantics**



Computer Architecture: Instruction Set Architecture



[Gerrit Blaauw & Fred Brooks, 1981]

Art and Science of Instruction Set Processor Design

ARCHITECTURE (ISA) programmer/compiler view

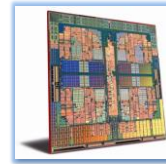
- Functional programming model to application/system programmers
- Opcodes, addressing modes, architected registers, IEEE floating point

IMPLEMENTATION (μ architecture) processor designer view

- Logical structure or organization that performs the ISA specification
- Pipelining, functional units, caches, physical registers, buses, branch predictors

REALIZATION (Chip) chip/system designer view

- Physical structure that embodies the implementation
- Gates, cells, transistors, wires, dies, packaging

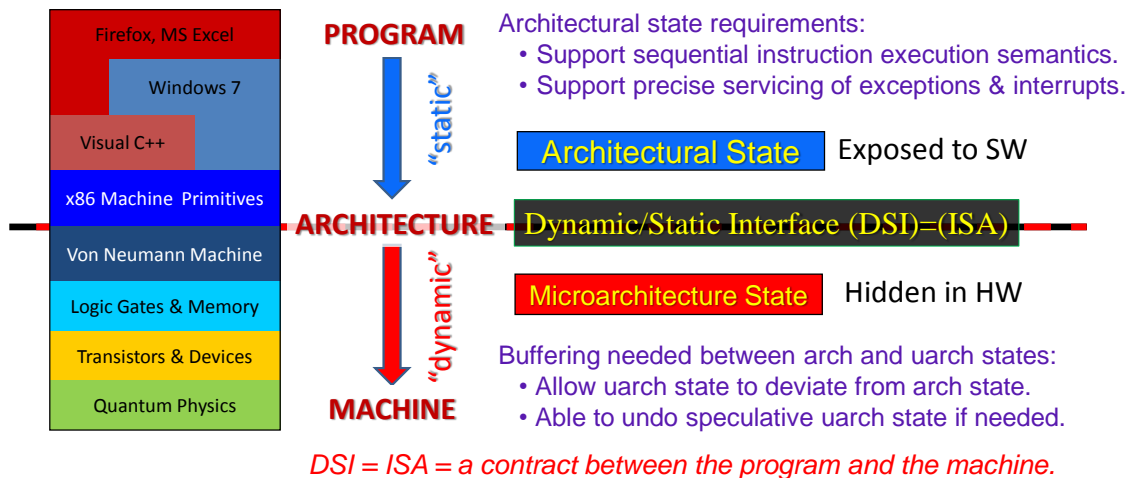


8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 13

Computer Architecture: Dynamic-Static Interface



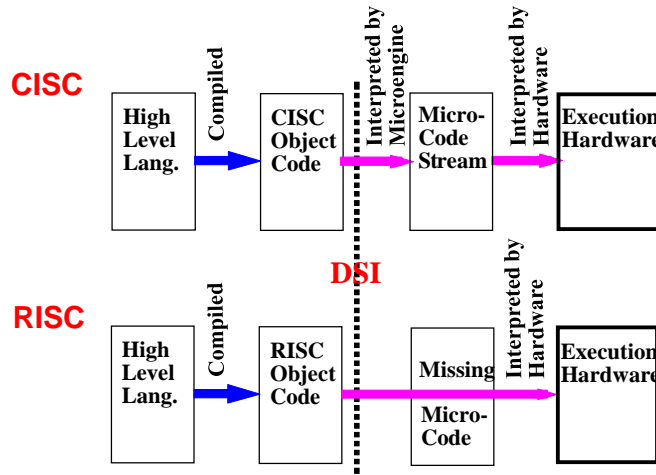
8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 14

[Josh Fisher, HP]

RISC vs. CISC *Transition from CISC to RISC:*

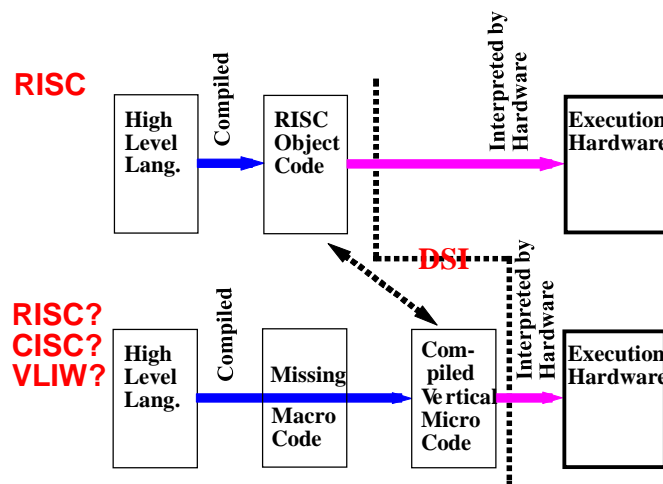


8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 15

Another way to view RISC



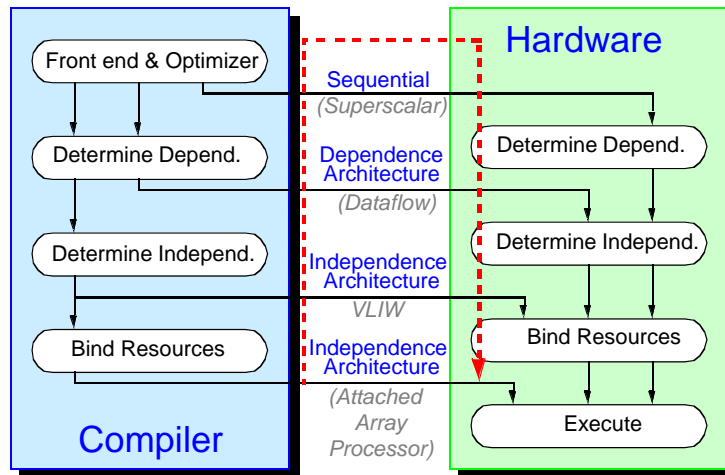
8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 16

[B. Rau & J. Fisher, 1993]

HW vs. SW and Dynamic vs. Static Design Space



8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 17

Lecture 1: "Introduction to Computer Architecture"

B. Historical Perspective on Computing

- a. Major Epochs
- b. Processor Performance Iron Law (#1)
- c. Course Coverage

8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 18

Almost Seven Decades of Modern Computing . . .



8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 19

Historical Perspective on the Last Five Decades

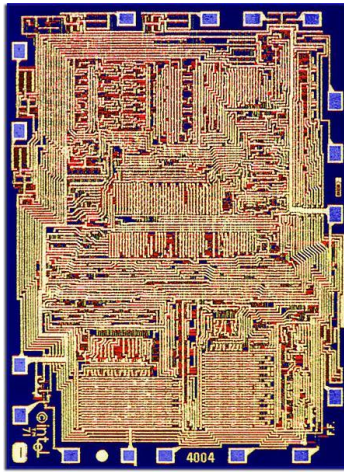
- **The Decade of the 1960's:** *"Computer Architecture Foundations"*
 - Von Neumann computation model, programming languages, compilers, OS's
 - Commercial Mainframe computers, Scientific numerical computers
- **The Decade of the 1970's:** *"Birth of Microprocessors"*
 - Programmable controllers, bit-sliced ALU's, single-chip processors
 - Emergence of Personal Computers (PC)
- **The Decade of the 1980's:** *"Quantitative Architecture"*
 - Instruction pipelining, fast cache memories, compiler considerations
 - Widely available Minicomputers, emergence of Personal Workstations
- **The Decade of the 1990's:** *"Instruction-Level Parallelism"*
 - Superscalar, speculative microarchitectures, aggressive compiler optimizations
 - Widely available low-cost desktop computers, emergence of Laptop computers
- **The Decade of the 2000's:** *"Mobile Computing Convergence"*
 - Multi-core architectures, system-on-chip integration, power constrained designs
 - Convergence of smartphones and laptops, emergence of Tablet computers

8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 20

Intel 4004, circa 1971

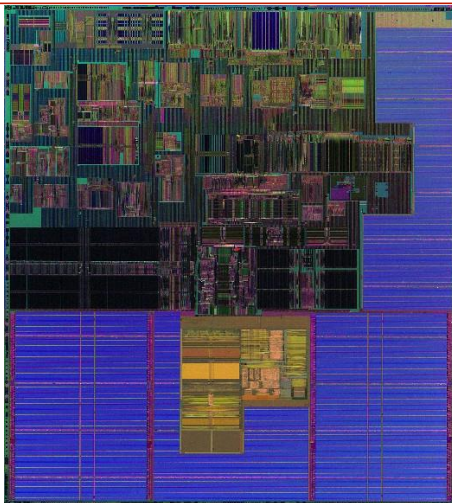


The first single chip CPU

- 4-bit processor for a calculator.
- 1K data memory
- 4K program memory
- 2,300 transistors
- 16-pin DIP package
- 740kHz (**eight clock cycles per CPU cycle** of 10.8 microseconds)
- ~100K OPs per second

Molecular Expressions: Chipshots

Intel Itanium 2, circa 2002



Performance leader in floating-point apps

- 64-bit processor
- 3 MByte in cache!!
- 221 million transistor
- 1 GHz, issue up to **8 instructions per cycle**

In ~30 years, about 100,000 fold growth in transistor count!

http://cpus.hp.com/images/die_photos/McKinley_die.jpg

[John Crawford, Intel, 1993]

Performance Growth in Perspective

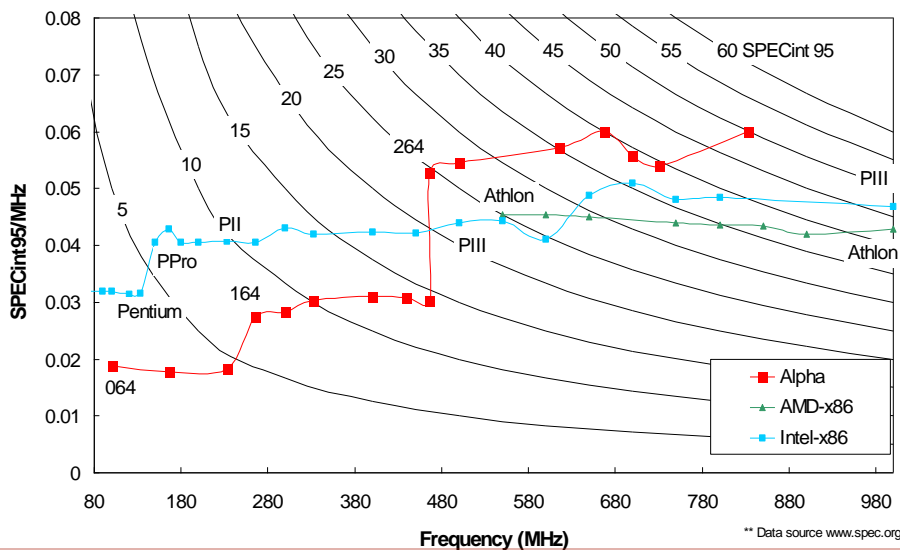
- **Doubling every 18 months (1982-2000):**
 - total of 3,200X
 - Cars travel at 176,000 MPH; get 64,000 miles/gal.
 - Air travel: L.A. to N.Y. in 5.5 seconds (MACH 3200)
 - Wheat yield: 320,000 bushels per acre
- **Doubling every 24 months (1971-2001):**
 - total of 36,000X
 - Cars travel at 2,400,000 MPH; get 600,000 miles/gal.
 - Air travel: L.A. to N.Y. in 0.5 seconds (MACH 36,000)
 - Wheat yield: 3,600,000 bushels per acre

Unmatched by any other industry!!

Convergence of Key Enabling Technologies

- **CMOS VLSI:**
 - Submicron feature sizes: 0.3 μ → 0.25 μ → 0.18 μ → 0.13 μ → 90nm → 65nm → 45nm → 32nm...
 - Metal layers: 3 → 4 → 5 → 6 → 7 (copper) → 12 ...
 - Power supply voltage: 5V → 3.3V → 2.4V → 1.8V → 1.3V → 1.1V ...
- **CAD Tools:**
 - Interconnect simulation and critical path analysis
 - Clock signal propagation analysis
 - Process simulation and yield analysis/learning
- **Microarchitecture:**
 - Superpipelined and superscalar machines
 - Speculative and dynamic microarchitectures
 - Simulation tools and emulation systems
- **Compilers:**
 - Extraction of instruction-level parallelism
 - Aggressive and speculative code scheduling
 - Object code translation and optimization

Landscape of Processor Families [SPECint95]

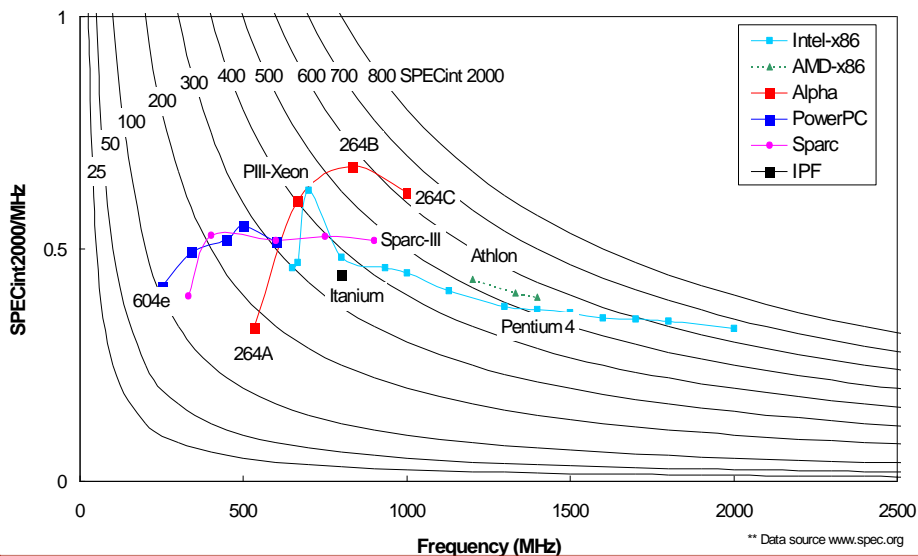


8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 27

Landscape of Processor Families [SPECint2000]



8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 28

Iron Law #1 – Processor (Latency) Performance

- ❖ Time to execute a program: T (latency)

$$T = \frac{\text{instructions}}{\text{program}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{time}}{\text{cycle}}$$

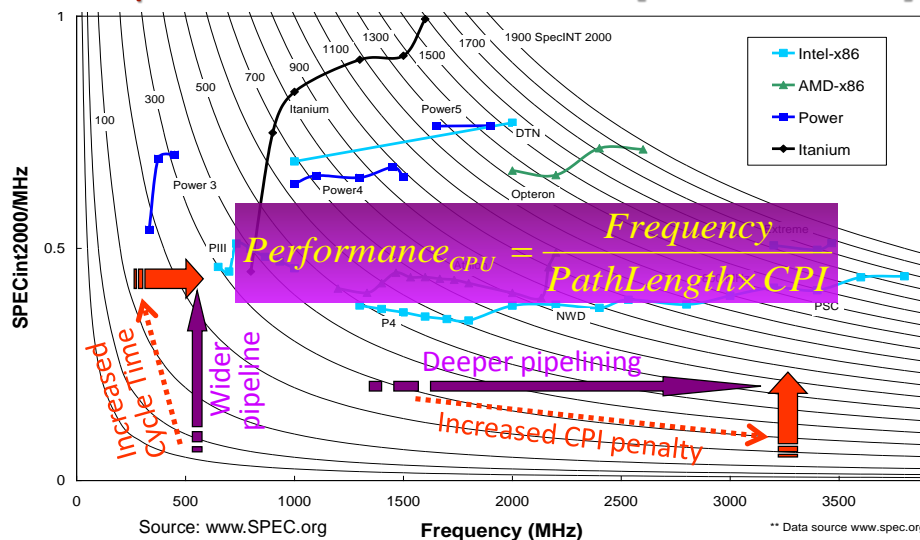
$$T = \text{PathLength} \times \text{CPI} \times \text{CycleTime}$$

- ❖ Processor performance: $\text{Perf} = 1/T$

$$\text{Perf}_{\text{CPU}} = \frac{1}{\text{PathLength} \times \text{CPI} \times \text{CycleTime}} = \frac{\text{Frequency}}{\text{PathLength} \times \text{CPI}}$$

Landscape of Processor Families [SPECint2000]

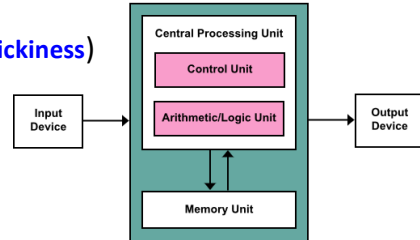
[John DeVale & Bryan Black, 2005]



18-640/18-740 Course Coverage: Processor Designs

Persistence of Von Neumann Model (**Legacy SW Stickiness**)

1. One CPU
2. Monolithic Memory
3. Sequential Execution Semantics



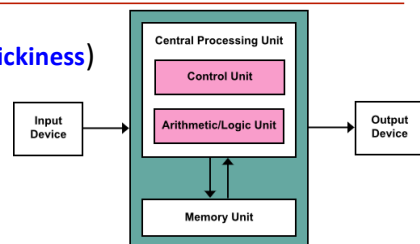
Evolution of Von Neumann Implementations:

- **SP**: Sequential Processors (direct implementation of sequential execution)
- **PP**: Pipelined Processors (overlapped execution of in-order instructions)
- **SSP**: Superscalar Processors (out-of-order execution of multiple instructions)
- **MCP**: Multi-core Processors = **CMP**: Chip Multiprocessors (concurrent multi-threads)
- **SMP**: Symmetric Multiprocessors (concurrent multi-threads and multi-programs)

18-640/18-740 Course Coverage: Parallelism Exploited

Persistence of Von Neumann Model (**Legacy SW Stickiness**)

1. One CPU
2. Monolithic Memory
3. Sequential Execution Semantics



Parallelisms for **Performance** (→ for **Power** Reduction → for **Energy** Efficiency)

- **ILP**: Basic Block (exploit **ILP** in PP, SSP)
- **ILP**: Loop Iteration (exploit **ILP** in SSP, VLIW)
- **DLP**: Data Set (exploit **DLP** in VP/SIMD, GPU)
- **TLP**: Task/Thread (exploit **TLP** in MCP)
- **PLP**: Process/Program (exploit **PLP** in MCP, SMP)

Lecture 1: "Introduction to Computer Architecture"

C. "Economics" of Computer Architecture

- a. Amdahl's Law and Gustafson's Law
- b. Moore's Law and Bell's Law



Carnegie Mellon University 33

8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

"Economics" of Computer Architecture

- Exercise in engineering tradeoff analysis
 - Find the fastest/cheapest/power-efficient/etc. solution
 - Optimization problem with 10s to 100s of variables
- All the variables are changing
 - At non-uniform rates
 - With inflection points
 - Only one guarantee: Today's right answer will be wrong tomorrow
- Two Persistent high-level "forcing functions":
 - **Application Demand (PROGRAM)**
 - **Technology Supply (MACHINE)**

8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 34

Four Foundational “Laws” of Computer Architecture

➤ Application Demand (PROGRAM)

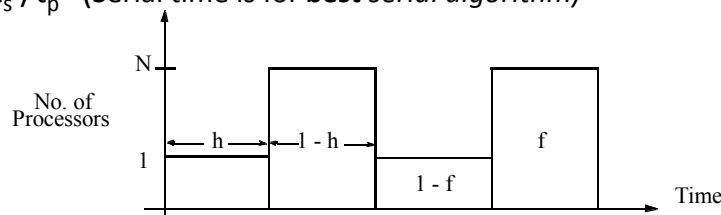
- **Amdahl’s Law** (1967)
 - Speedup through parallelism is limited by the sequential bottleneck
- **Gustafson’s Law** (1988)
 - With unlimited data set size, parallelism speedup can be unlimited

➤ Technology Supply (MACHINE)

- **Moore’s Law** (1965)
 - (Transistors/Die) increases by 2x every 18 months
- **Bell’s Law** (1971)
 - (Cost/Computer) decreases by 2x every 36 months

Amdahl’s Law

- **Speedup** = (Execution time on Single CPU)/(Execution on N parallel processors)
- t_s / t_p (Serial time is for **best serial algorithm**)

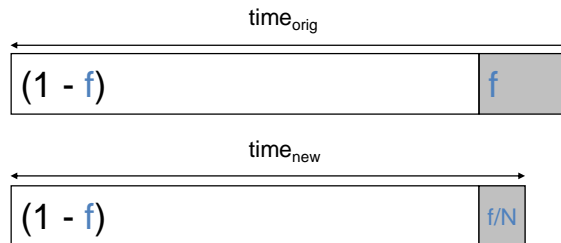


- h = fraction of time in serial code
- f = fraction that is vectorizable or parallelizable
- N = max speedup for f
- Overall speedup $\rightarrow \rightarrow$

$$Speedup = \frac{1}{(1-f) + \frac{f}{N}}$$

Amdahl's Law Illustrated

- Speedup = $\text{time}_{\text{without enhancement}} / \text{time}_{\text{with enhancement}}$
- If an enhancement speeds up a fraction f of a task by a factor of N
- $\text{time}_{\text{new}} = \text{time}_{\text{orig}} \cdot ((1-f) + f/N)$
- $S_{\text{overall}} = 1 / ((1-f) + f/N)$

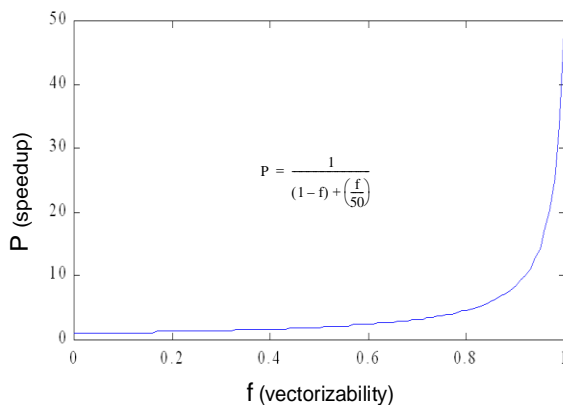


8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 37

“Tyranny of Amdahl’s Law” [Bob Colwell, CMU-Intel-DARPA]



- Suppose that a computation has a 4% serial portion, what is the limit of speedup on 16 processors?
 - $1 / ((0.04) + (0.96/16)) = 10$
 - What is the maximum speedup?
 - $1/0.04 = 25$ (with $N \rightarrow \infty$)

8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 38

From Amdahl's Law to Gustafson's Law

- **Amdahl's Law** works on a *fixed* problem size
 - This is reasonable if your only goal is to solve a problem faster.
 - What if you also want to solve a larger problem?
 - Gustafson's Law (Scaled Speedup)
- **Gustafson's Law** is derived by fixing the parallel execution time (Amdahl fixed the problem size -> fixed serial execution time)
 - For many practical situations, Gustafson's law makes more sense
 - Have a bigger computer, solve a bigger problem.
 - "Amdahl's Law turns out to be too pessimistic for high-performance computing."

Gustafson's Law

- Fix execution of the computation on a single processor as
 - $s + p = \text{serial part} + \text{parallelizable part} = 1$
- $\text{Speedup}(N) = (s + p)/(s + p/N)$

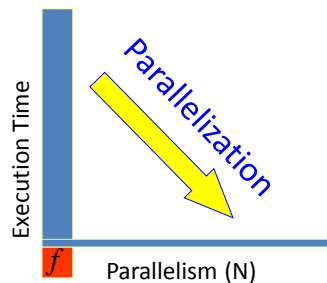
$$= 1/(s + (1 - s)/N) = 1/((1-p) + p/N) \leftarrow \text{Amdahl's law}$$
- Now let $1 = (a + b) = \text{execution time of computation on } N \text{ processors (fixed)}$ where $a = \text{sequential time}$ and $b = \text{parallel time on any of the } N \text{ processors}$
 - Time for sequential processing = $a + (b \times N)$ and $\text{Speedup} = (a + b \times N)/(a + b)$
 - Let $\alpha = a/(a+b)$ be the sequential fraction of the parallel execution time
 - $\text{Speedup}_{\text{scaled}}(N) = (a + b \times N)/(a + b) = (a/(a+b) + (b \times N)/(a+b)) = \alpha + (1 - \alpha)N$
 - If α is very small, the scaled speedup is approximately N , i.e. linear speedup.

The Two Laws on Algorithm and Performance

Amdahl's Law

$$Speedup(N)_{MC} = \frac{1}{\left(\frac{f}{1}\right) + \left(\frac{1-f}{N}\right)}$$

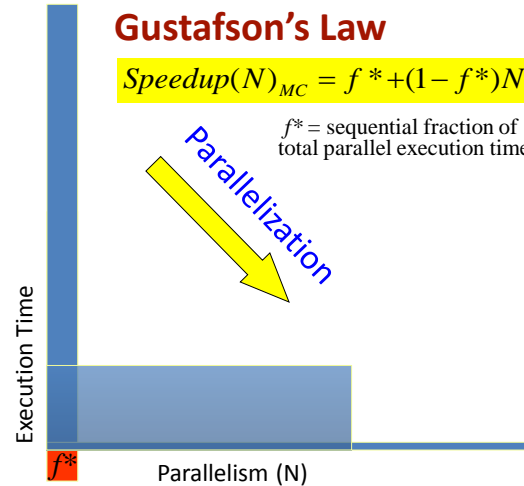
f = sequential %



Gustafson's Law

$$Speedup(N)_{MC} = f^* + (1 - f^*)N$$

f^* = sequential fraction of total parallel execution time



The Two "Gordon" Laws of Computer Architecture

➤ Gordon Moore's Law (1965)

- (Transistors/Die) increases by 2X every 18 months
- Constant price, increasing performance
- Has held for 40+ years, and will continue to hold

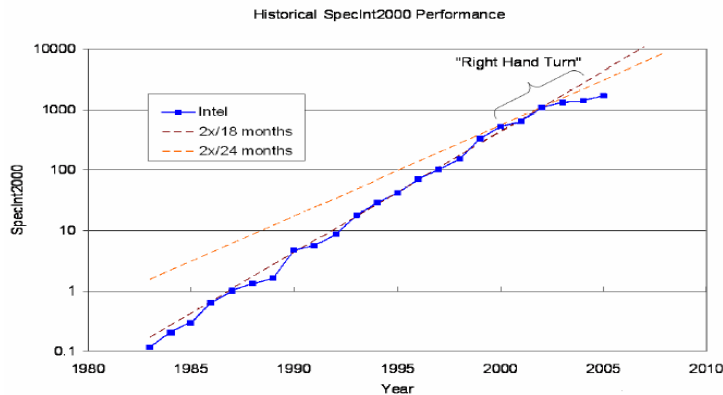
➤ Gordon Bell's Law (1971)

- (Cost/Computer) decreases by 2X every 36 months (~ 10X per decade)
- Constant performance, decreasing price
- Corollary of Moore's Law, creation of new computer categories

"In a decade you can buy a computer for less than its sales tax today." – Jim Gray

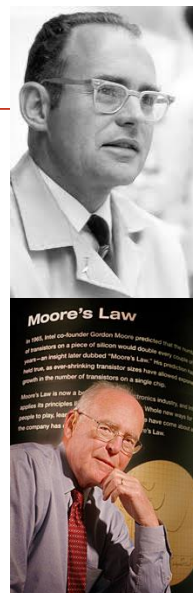
We have all been living on this exponential curve and assume it...

Moore's Law Trends

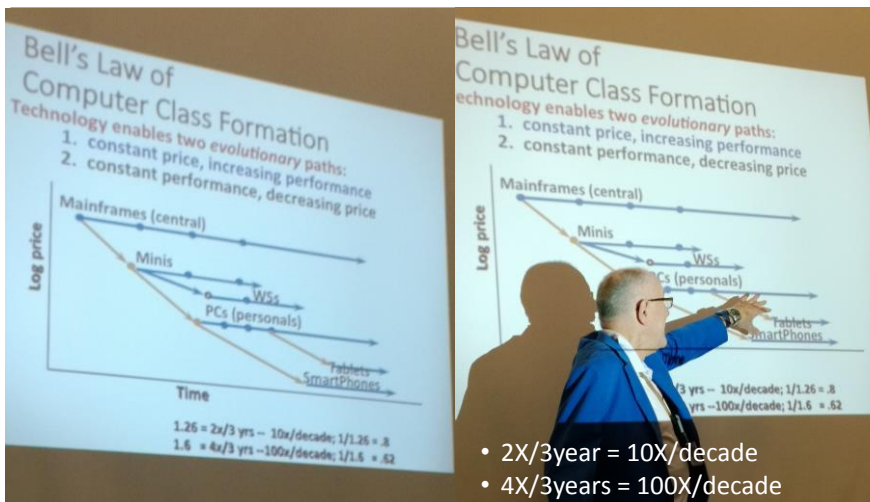


- Moore's Law for device integration
- Chip power consumption
- Single-thread performance trend

[source: Intel]



Bell's Law Trends

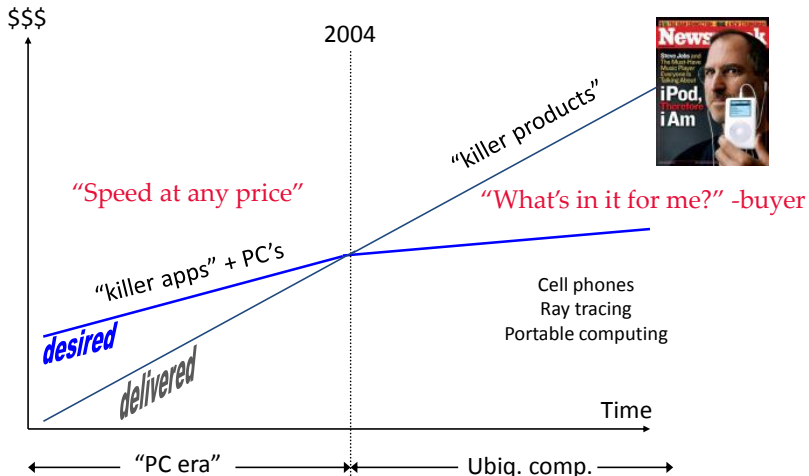


- 2X/3year = 10X/decade
- 4X/3years = 100X/decade

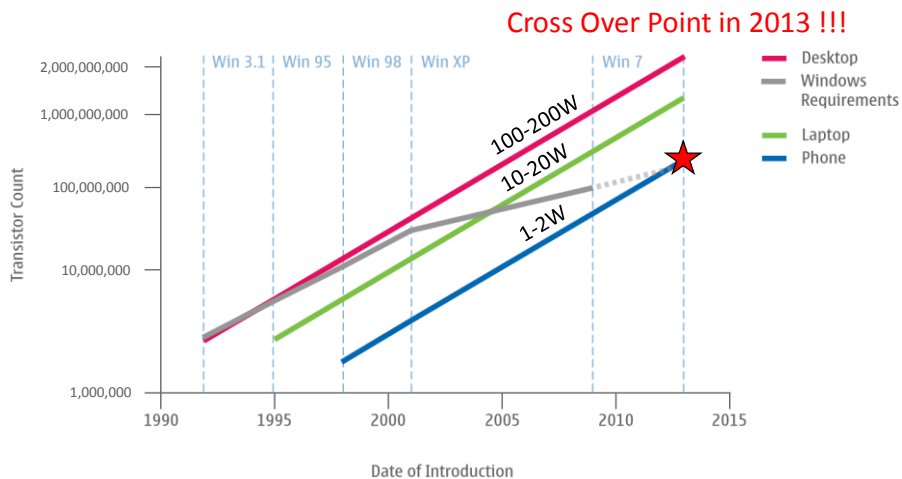


[Bob Colwell CRA Grand Challenges panel 2005]

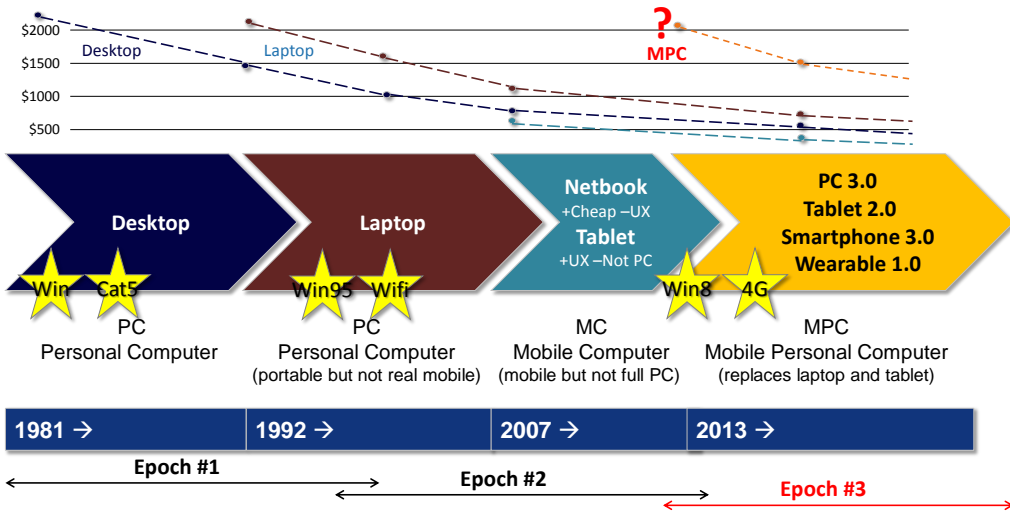
Know Your "Supply & Demand Curves"



Moore's Law and Bell's Law are Alive and Well



New Epoch of “Mobile Personal Computers” (MPC)



8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 47

18-640/18-740 Computer Architecture

Lecture 2: “Review of Pipelined Processor Design”

John Paul Shen
September 2, 2015

Next Time ...

➤ Required Reading Assignment:

- Chapter 2 of Shen and Lipasti (SnL).

➤ Recommended Reference:

- ❖ “Optimum Power/Performance Pipeline Depth” by A. Hartstein and Thomas R. Puzak, MICRO 2003.



Electrical & Computer
ENGINEERING

8/31/2015 (©J.P. Shen)

18-640/18-740 Lecture 1

Carnegie Mellon University 48