Lecture 01: Introduction

CSCE 513 Computer Architecture
Fall 2018

Department of Computer Science and Engineering

Yonghong Yan

yanyh@cse.sc.edu

http://cse.sc.edu/~yanyh

Copyright and Acknowledgement

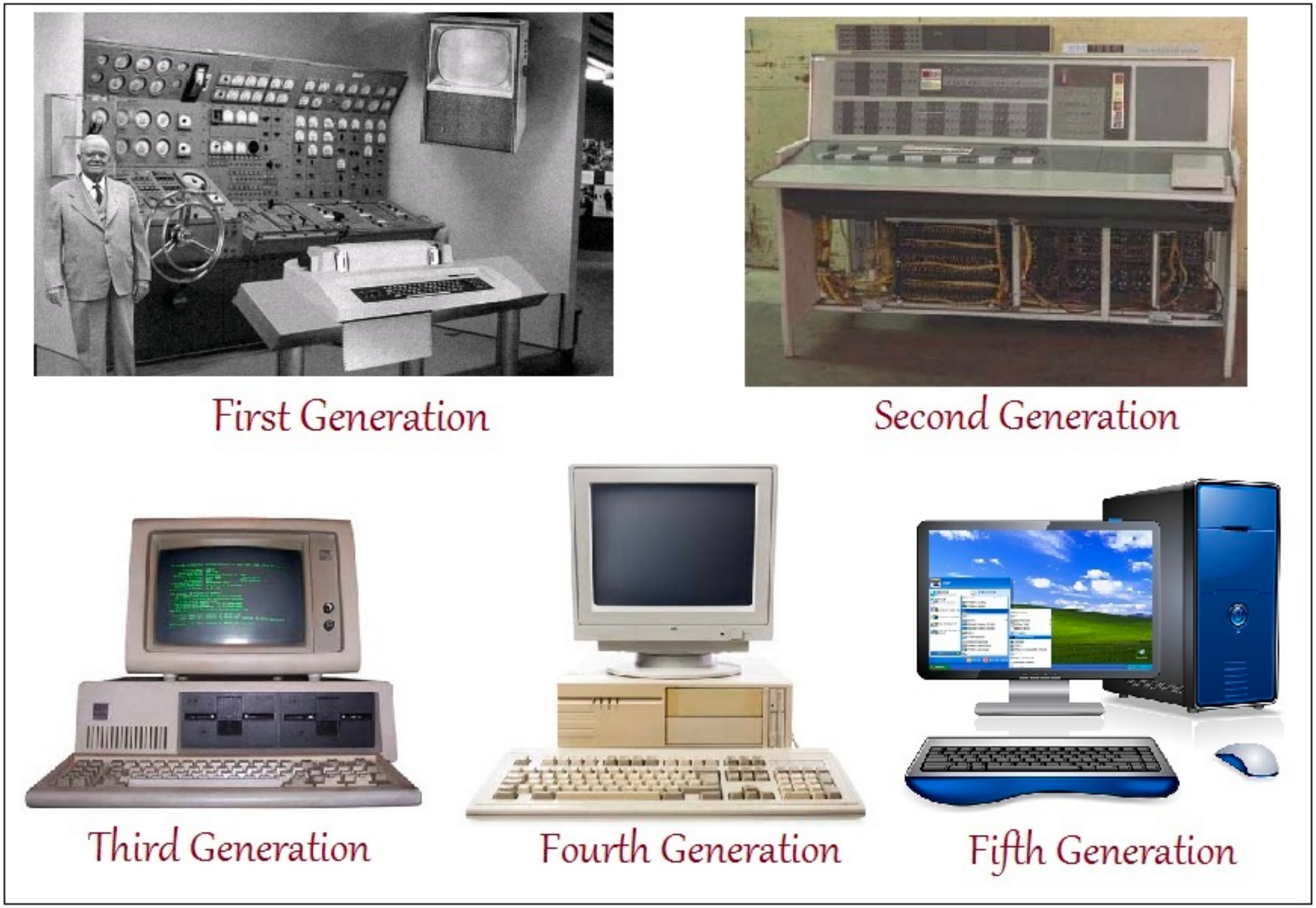
- Lots of the slides were adapted from lectures notes of the two textbooks with copyright of publisher or the original authors including Elsevier Inc, Morgan Kaufmann, David A. Patterson and John L. Hennessy.
- Some slides were adapted from the following courses:
  - UC Berkeley course “Computer Science 252: Graduate Computer Architecture” of David E. Culler Copyright 2005 UCB
    - http://people.eecs.berkeley.edu/~culler/courses/cs252-s05/
  - Great Ideas in Computer Architecture (Machine Structures) by Randy Katz and Bernhard Boser
    - http://inst.eecs.berkeley.edu/~cs61c/fa16/
- I also refer to the following courses and lecture notes when preparing materials for this course
  - Computer Science 152: Computer Architecture and Engineering, Spring 2016 by Dr. George Michelogiannakis from UC Berkeley
    - http://www-inst.eecs.berkeley.edu/~cs152/sp16/
  - Computer Science 252: Graduate Computer Architecture, Fall 2015 by Prof. Krste Asanović from UC Berkeley
    - http://www-inst.eecs.berkeley.edu/~cs252/fa15/
  - Computer Science S 250: VLSI Systems Design, Spring 2016 by Prof. John Wawrzynek from UC Berkeley
    - http://www-inst.eecs.berkeley.edu/~cs250/sp16/
  - Computer System Architecture, Fall 2005 by Dr. Joel Emer and Prof. Arvind from MIT
    - http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-823-computer-system-architecture-fall-2005/
  - Synthesis Lectures on Computer Architecture
    - http://www.morganclaypool.com/toc/cac/1/1

- **The uses of the materials (source code, slides, documents and videos) of this course are for educational purposes only and should be used only in conjunction with the textbook. Derivatives of the materials must acknowledge the copyright notices of this and the originals. Permission for commercial purposes should be obtained from the original copyright holder and the successive copyright holders including myself.**

Contents

- **Computer components**
- Computer architectures and great ideas in computer architectures
- Performance

Generation Of Computers



First Generation

Second Generation

Third Generation

Fourth Generation

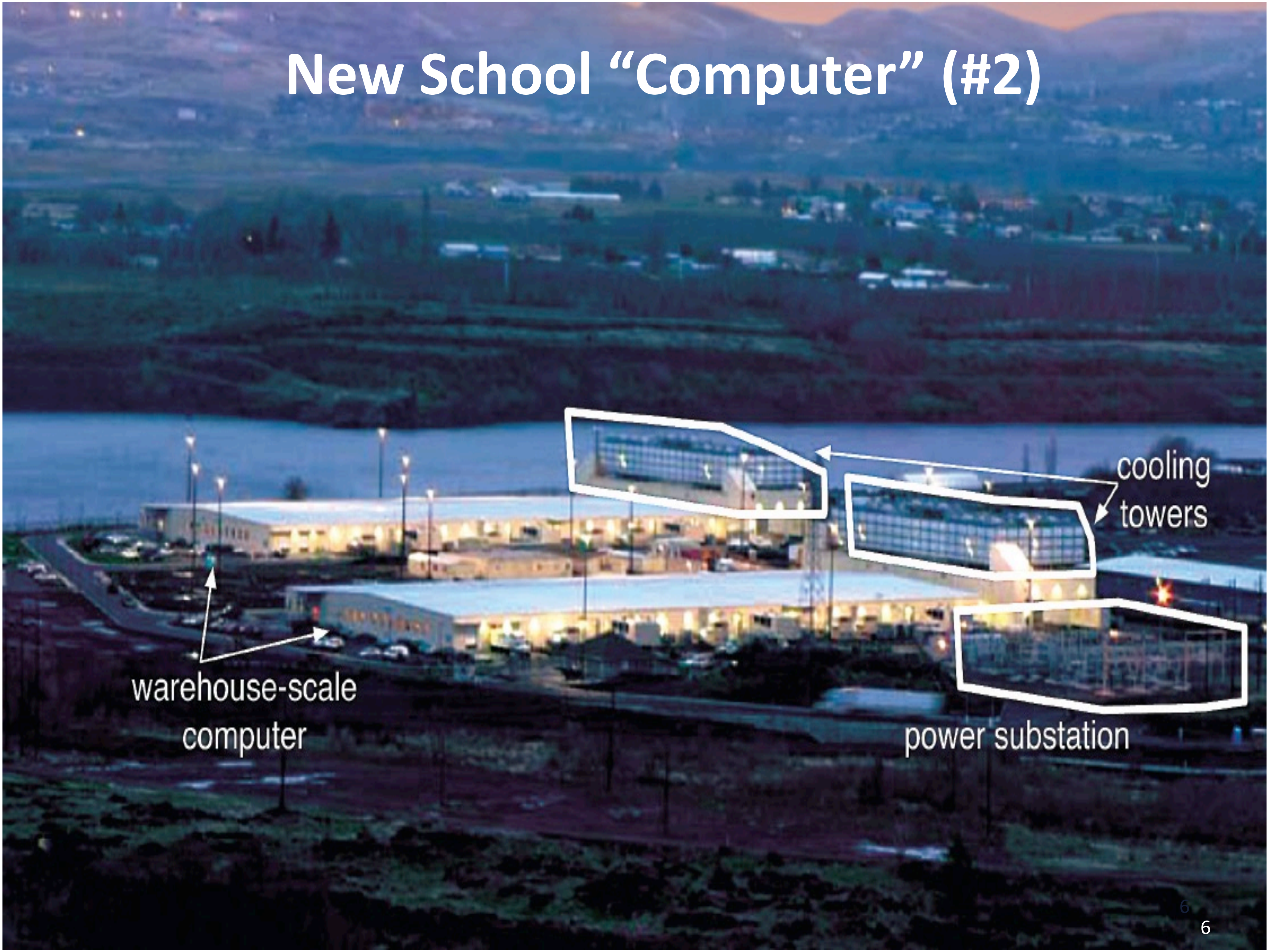
Fifth Generation

https://solarrenovate.com/the-evolution-of-computers/

New School Computer (#1)

Personal
Mobile
Devices

New School “Computer” (#2)



Classes of Computers

- Personal Mobile Device (PMD)
  - e.g. start phones, tablet computers
  - Emphasis on energy efficiency and real-time
- Desktop Computing
  - Emphasis on price-performance
- Servers
  - Emphasis on availability, scalability, throughput
- Clusters / Warehouse Scale Computers
  - Used for "Software as a Service (SaaS)"
  - Emphasis on availability and price-performance
  - Sub-class: Supercomputers, emphasis: floating-point performance and fast internal networks
- Internet of Things/Embedded Computers
  - Emphasis: price

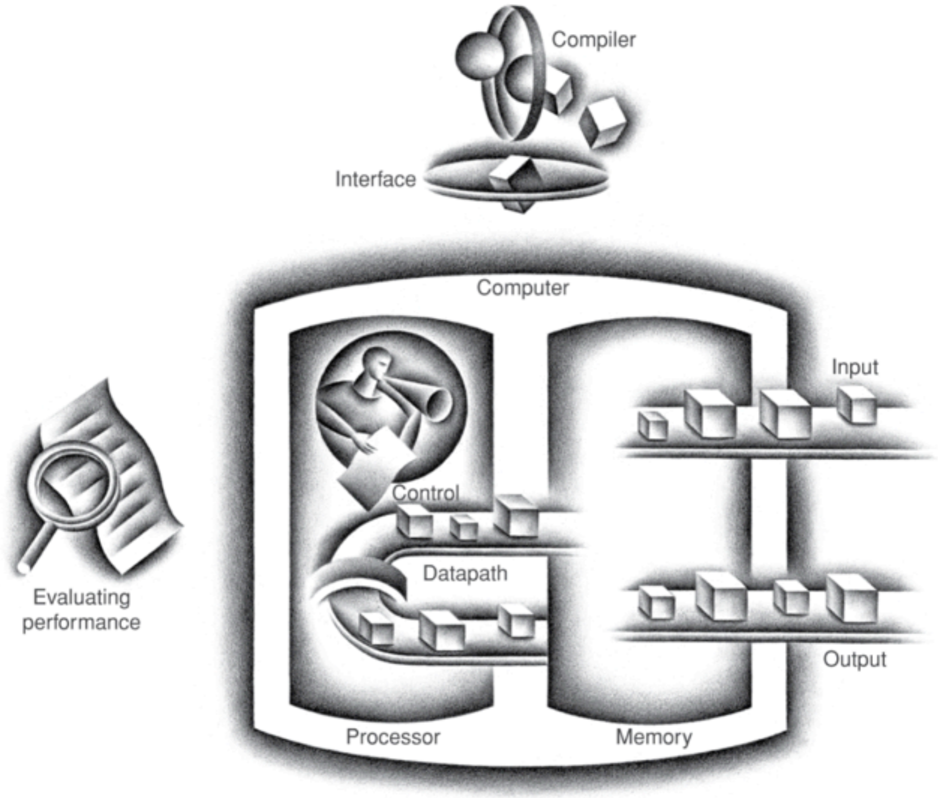
Notes by the Pioneers

- "I think there is a world market for maybe five computers."
  - Thomas Watson, chairman of IBM, 1943.

- "There is no reason for any individual to have a computer in their home"
  - Ken Olson, president and founder of Digital Equipment Corporation, 1977.

- "640K [of memory] ought to be enough for anybody."
  - Bill Gates, chairman of Microsoft,1981.

Components of a Computer

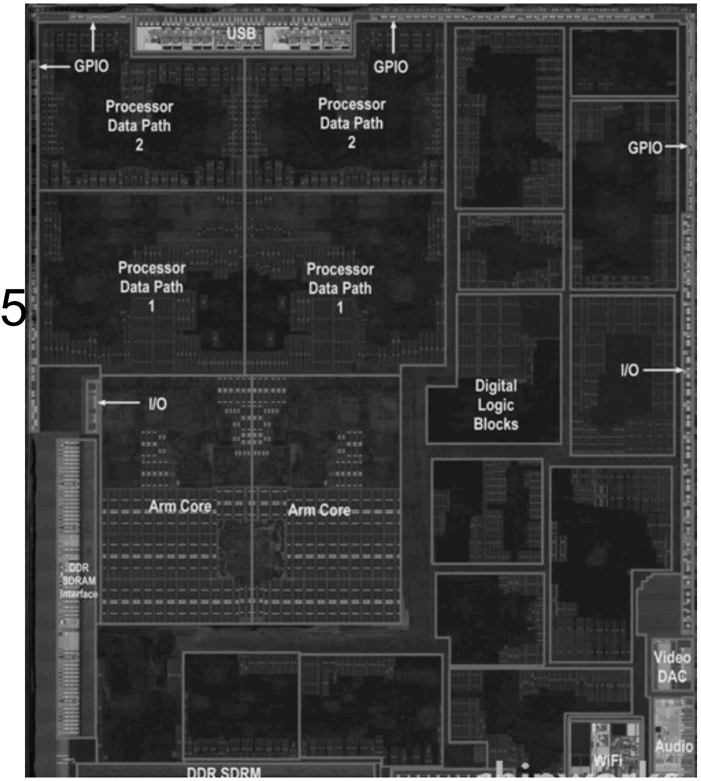


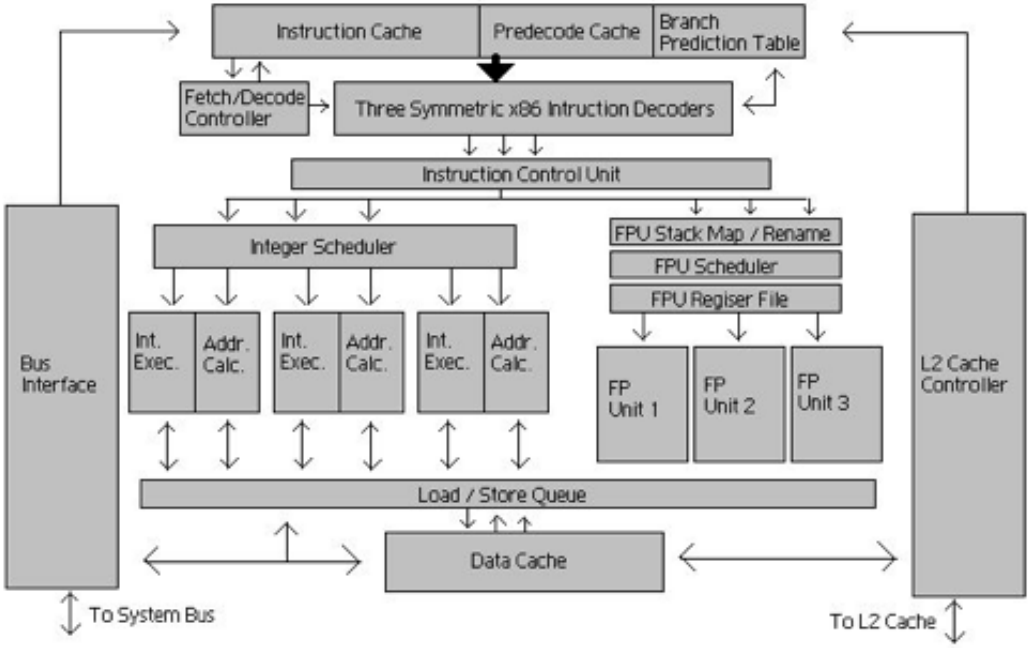
- Same components for all kinds of computer
  - Desktop, server, embedded
- Two core parts
  - Processor and memory
- Input/output includes
  - User-interface devices
    - Display, keyboard, mouse
  - Storage devices
    - Hard disk, CD/DVD, flash
  - Network adapters
    - For communicating with other computers

Inside the Processor (CPU)

- Functional units: performs computations
- Datapath: wires for moving data
- Control logic: sequences datapath, memory, and operations
- Cache memory
  - Small fast SRAM memory for immediate access to data

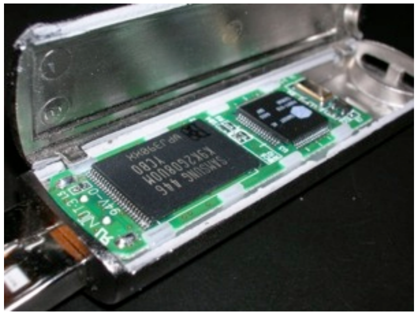
Apple A5





A Safe Place for Data

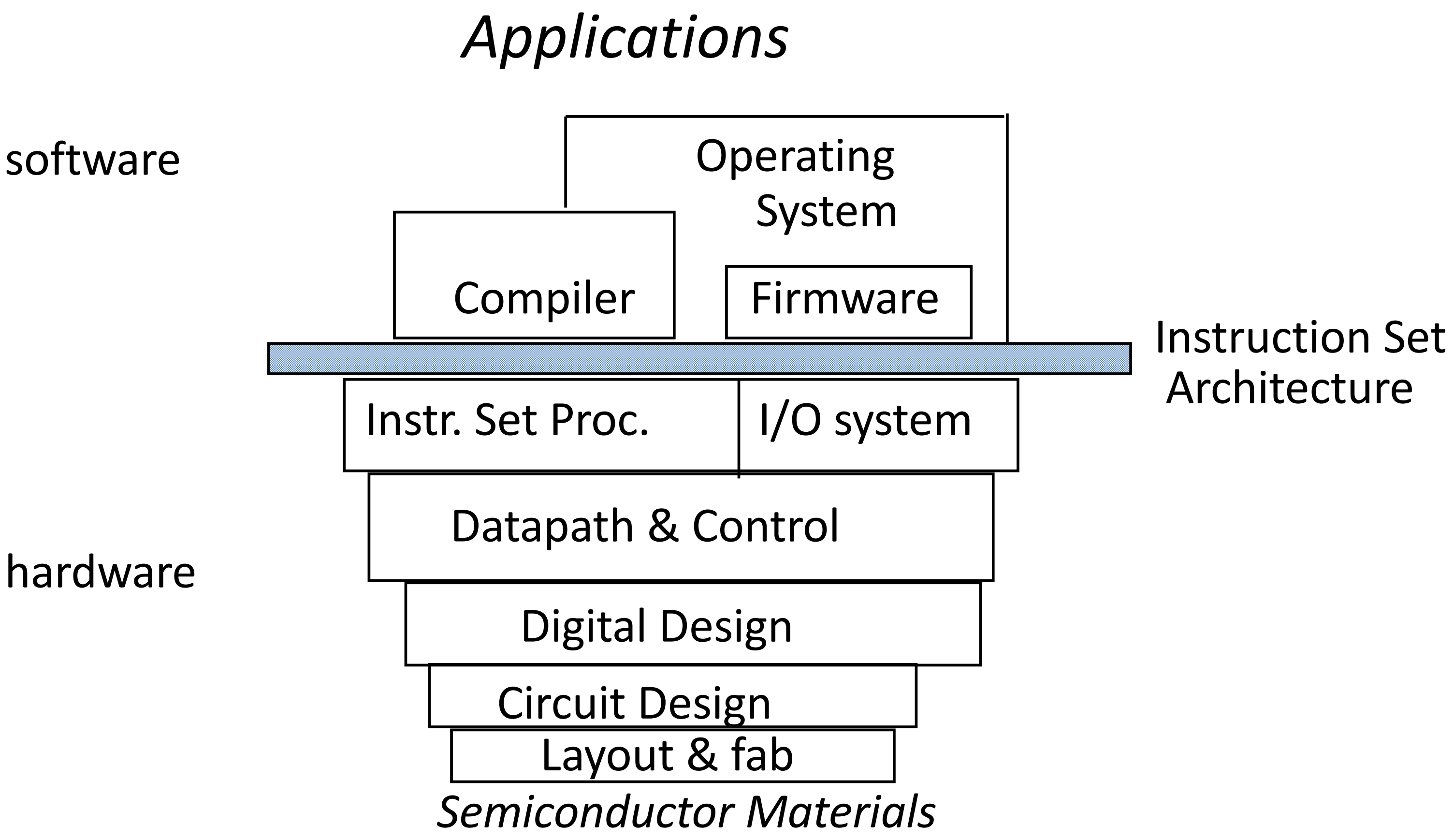
- Volatile main memory
  - Loses instructions and data when power off
- Non-volatile secondary memory
  - Magnetic disk
  - Flash memory
  - Optical disk (CDROM, DVD)



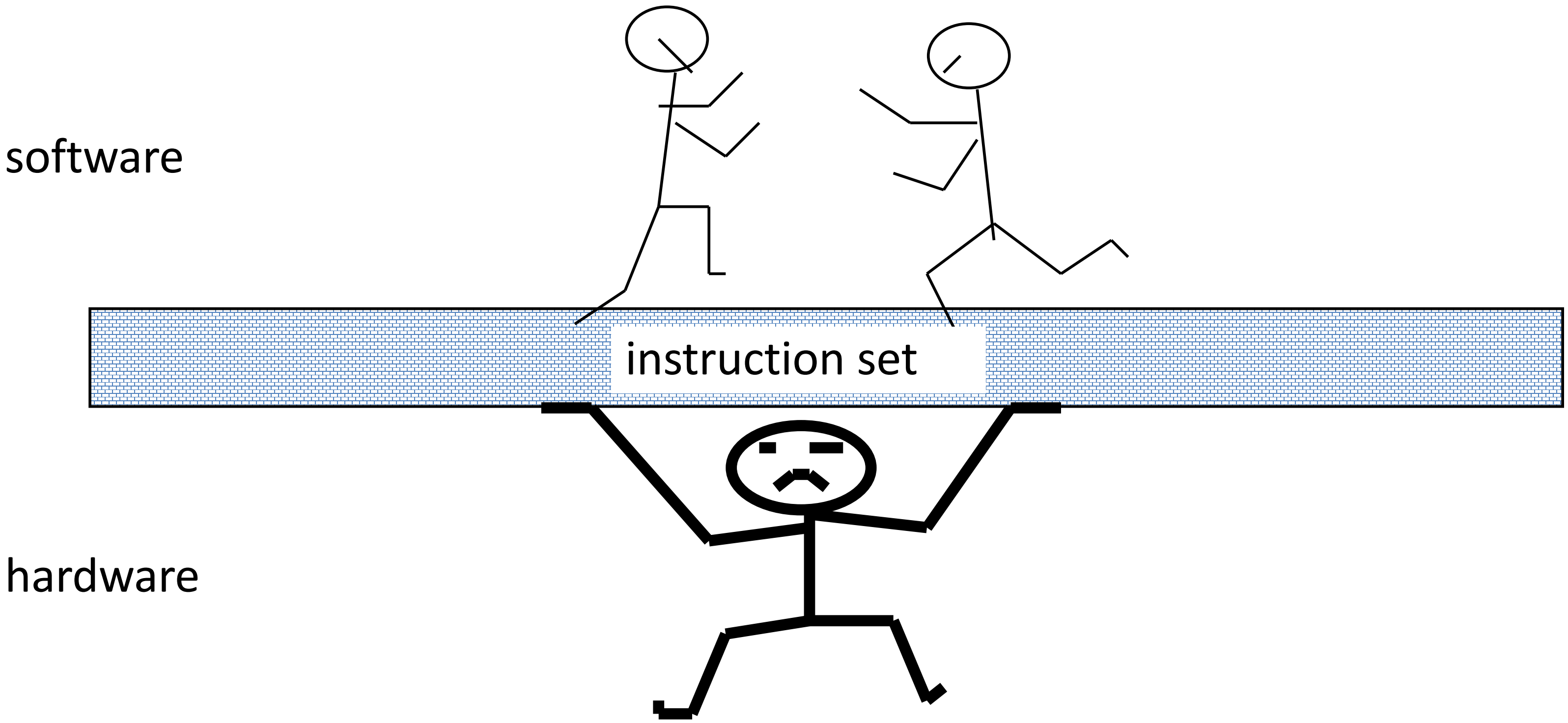
Contents

- Computer components
- **Computer architectures and great ideas in computer architectures**
- Performance

What is "Computer Architecture"?



The Instruction Set: a 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

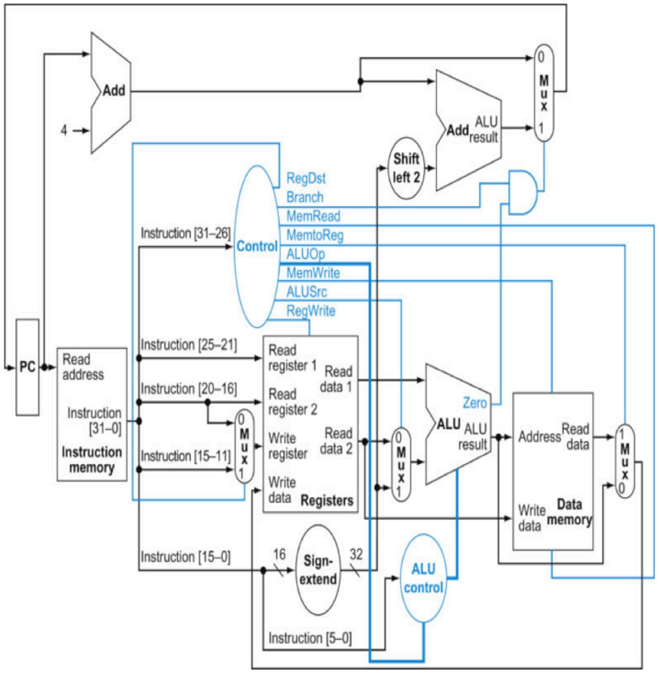
Elements of an ISA

- Set of machine-recognized data types
  - bytes, words, integers, floating point, strings, . . .
- Operations performed on those data types
  - Add, sub, mul, div, xor, move, ….
- Programmable storage
  - regs, PC, memory
- Methods of identifying and obtaining data referenced by instructions (addressing modes)
  - Literal, reg., absolute, relative, reg + offset, …
- Format (encoding) of the instructions
  - Op code, operand fields, …

Computer Architecture

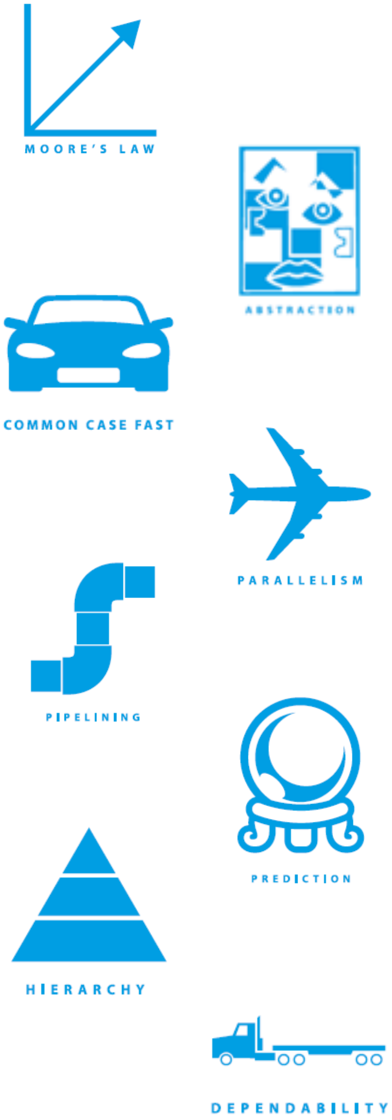
How things are put together in design and implementation

- Capabilities & Performance Characteristics of Principal Functional Units
  - (e.g., Registers, ALU, Shifters, Logic Units, ...)
- Ways in which these components are interconnected
- Information flows between components
- Logic and means by which such information flow is controlled.
- Choreography of FUs to realize the ISA



Great Ideas in Computer Architectures

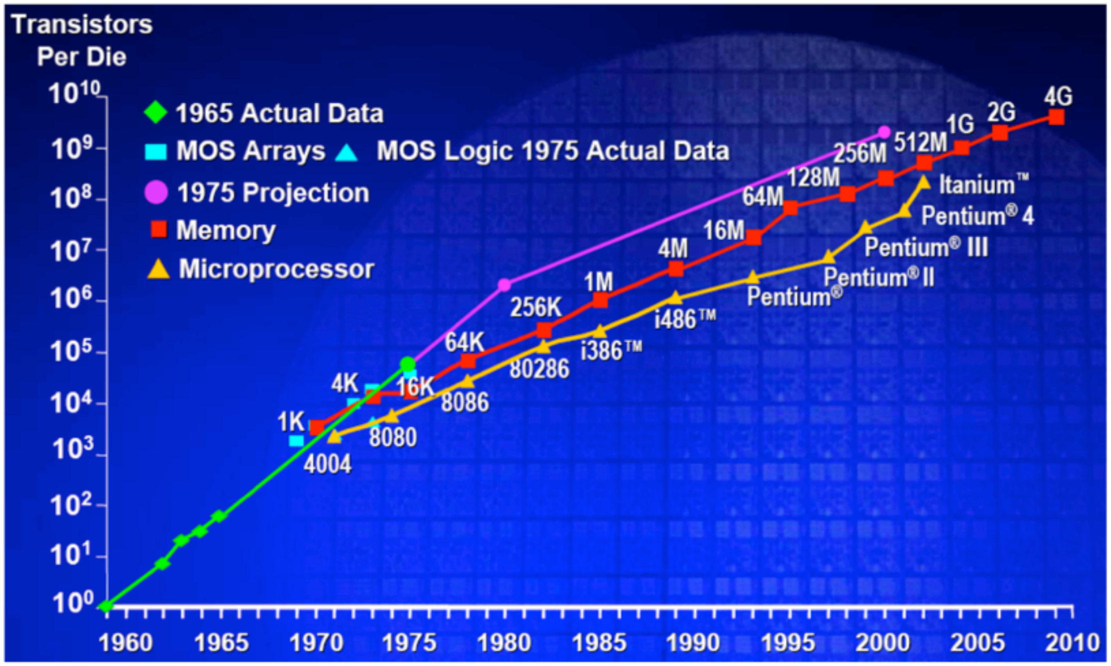
1. Design for ***Moore’s Law***
2. Use ***abstraction*** to simplify design
3. Make the ***common case fast***
4. Performance *via* ***parallelism***
5. Performance *via* ***pipelining***
6. Performance *via* ***prediction***
7. ***Hierarchy*** of memories
8. ***Dependability*** *via* redundancy



Great Idea: “Moore’s Law”

Gordon Moore, Founder of Intel

- 1965: since the integrated circuit was invented, the number of transistors/inch^2 in these circuits roughly doubled every year; this trend would continue for the foreseeable future
- 1975: revised - circuit complexity doubles every two years



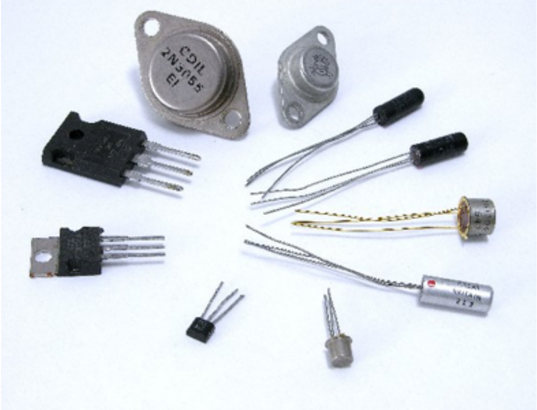


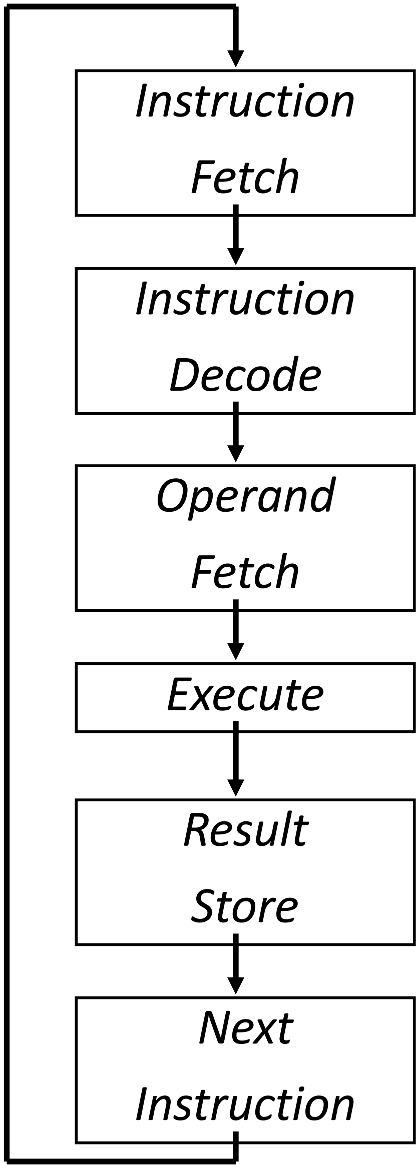
Image credit: Intel

Moore’s Law trends

- More transistors = ↑ opportunities for exploiting parallelism in the instruction level (ILP)
  - Pipeline, superscalar, VLIW (Very Long Instruction Word), SIMD (Single Instruction Multiple Data) or vector, speculation, branch prediction
- General path of scaling
  - Wider instruction issue, longer piepline
  - More speculation
  - More and larger registers and cache
- **Increasing circuit density ~= increasing frequency ~= increasing performance**
- Transparent to users
  - An easy job of getting better performance: buying faster processors (higher frequency)
- **We have enjoyed this free lunch for several decades, however (TBC) …**

Great Idea: Pipeline

Fundamental Execution Cycle



Instruction Fetch

Obtain instruction from program storage

Instruction Decode

Determine required actions and instruction size

Operand Fetch

Locate and obtain operand data

Execute

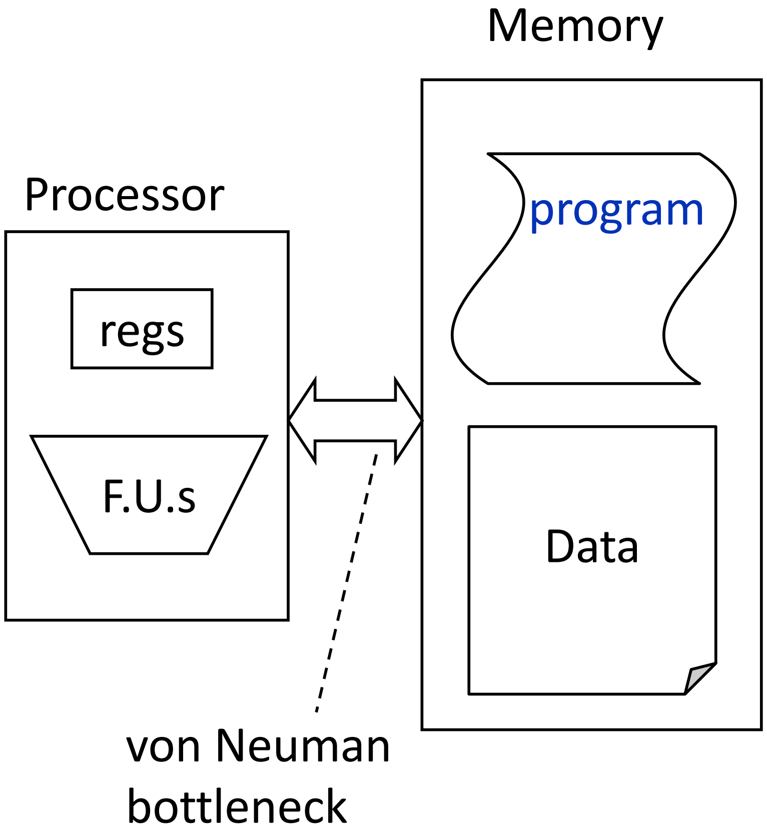
Compute result value or status

Result Store

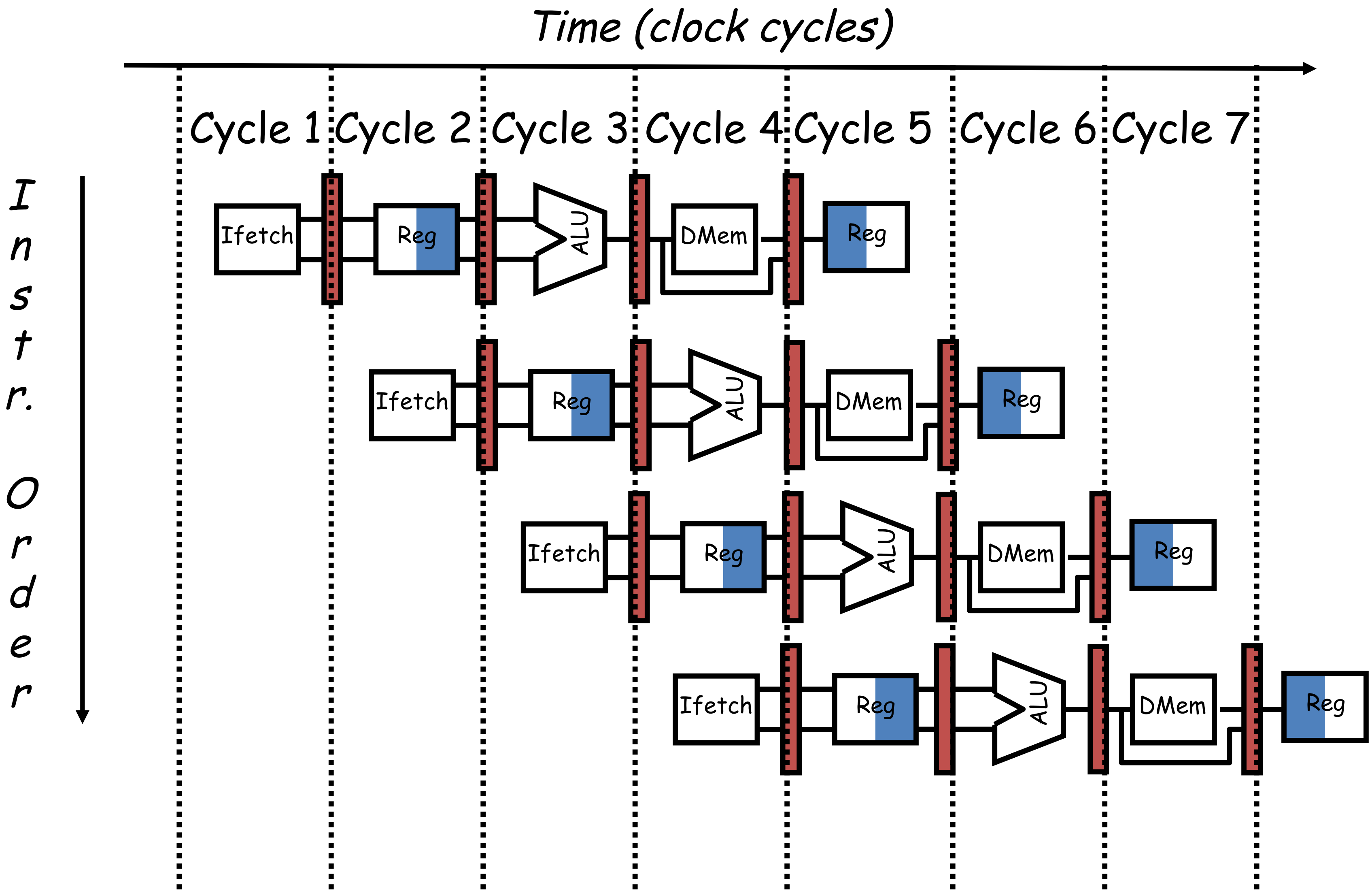
Deposit results in storage for later use

Next Instruction

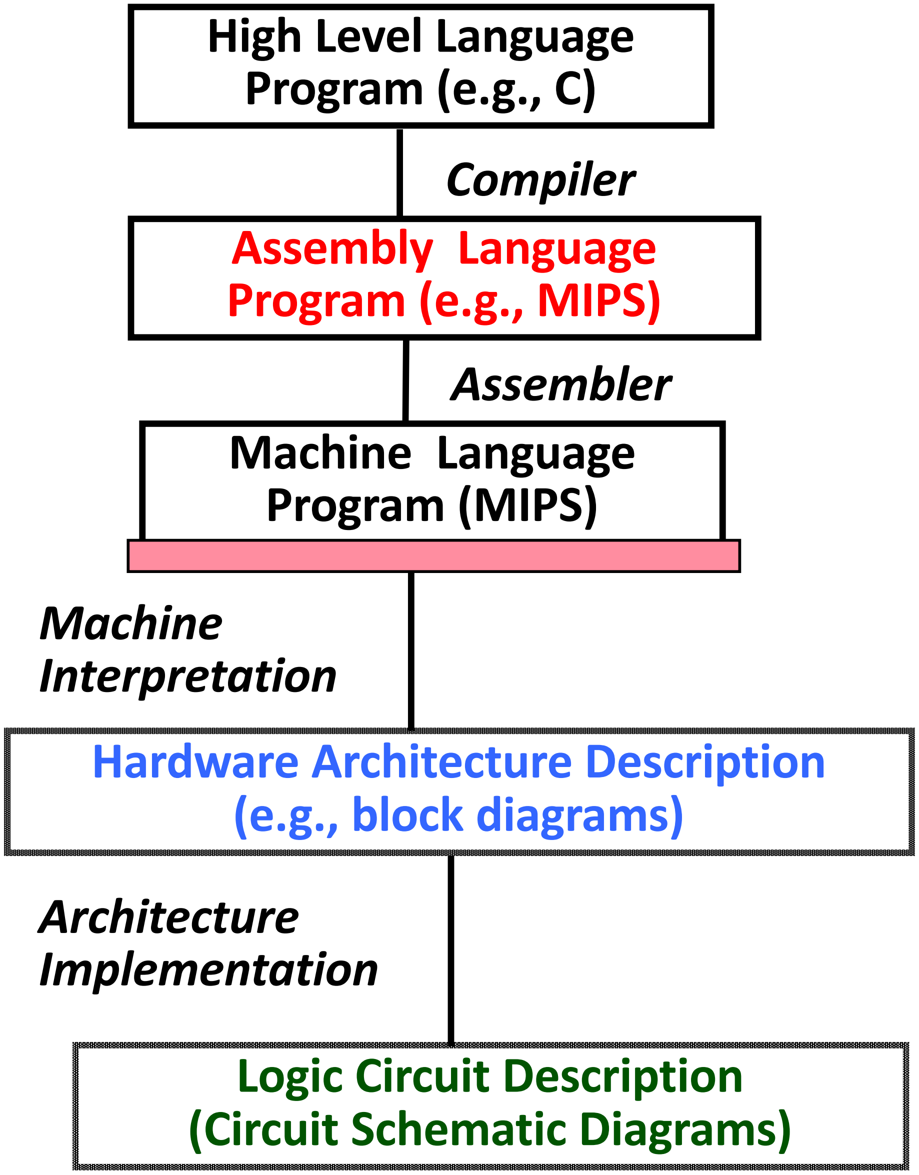
Determine successor instruction



Pipelined Instruction Execution



Great Idea: Abstraction
(Levels of Representation/Interpretation)



High Level Language Program (e.g., C)

Compiler

Assembly Language Program (e.g., MIPS)

Assembler

Machine Language Program (MIPS)

Machine Interpretation

Hardware Architecture Description (e.g., block diagrams)

Architecture Implementation

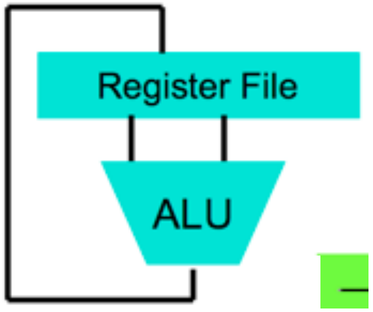
Logic Circuit Description (Circuit Schematic Diagrams)

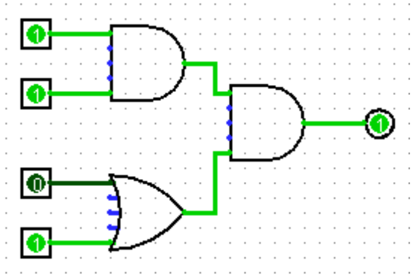
```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw   $t0, 0($2)
lw   $t1, 4($2)
sw   $t1, 0($2)
sw   $t0, 4($2)
```

Anything can be represented as a *number*, i.e., data or instructions

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

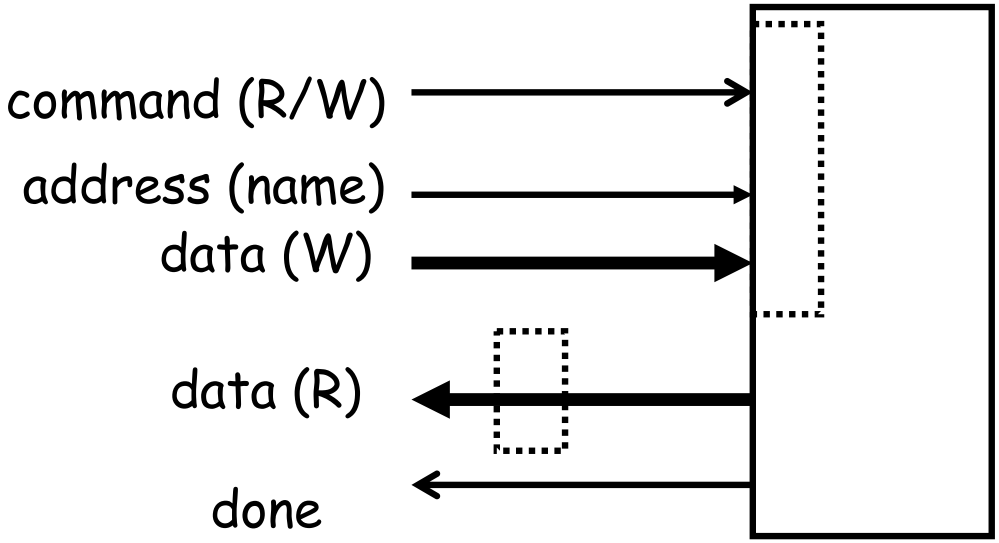




The Memory Abstraction

- Association of <name, value> pairs
  - typically named as byte addresses
  - often values aligned on multiples of size
- Sequence of Reads and Writes
- Write binds a value to an address
  - Left value
- Read of addr returns most recently written value bound to that address
  - Right value

int a = b;



Great idea: Memory Hierarchy

Levels of the Memory Hierarchy

Capacity
Access Time
Cost

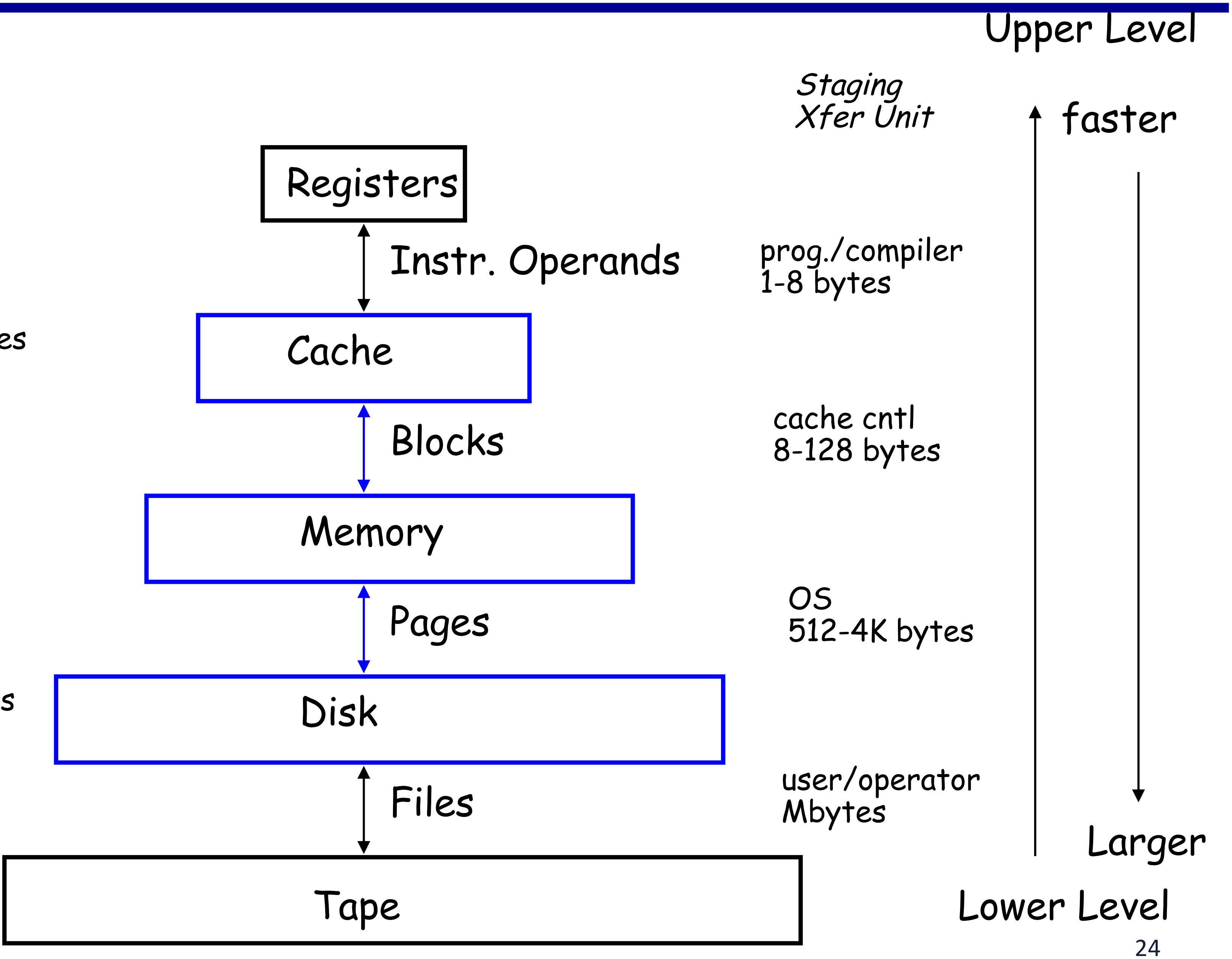
CPU Registers
100s Bytes
<< 1s ns

Cache
10s-100s K Bytes
~1 ns
$1s/ MByte

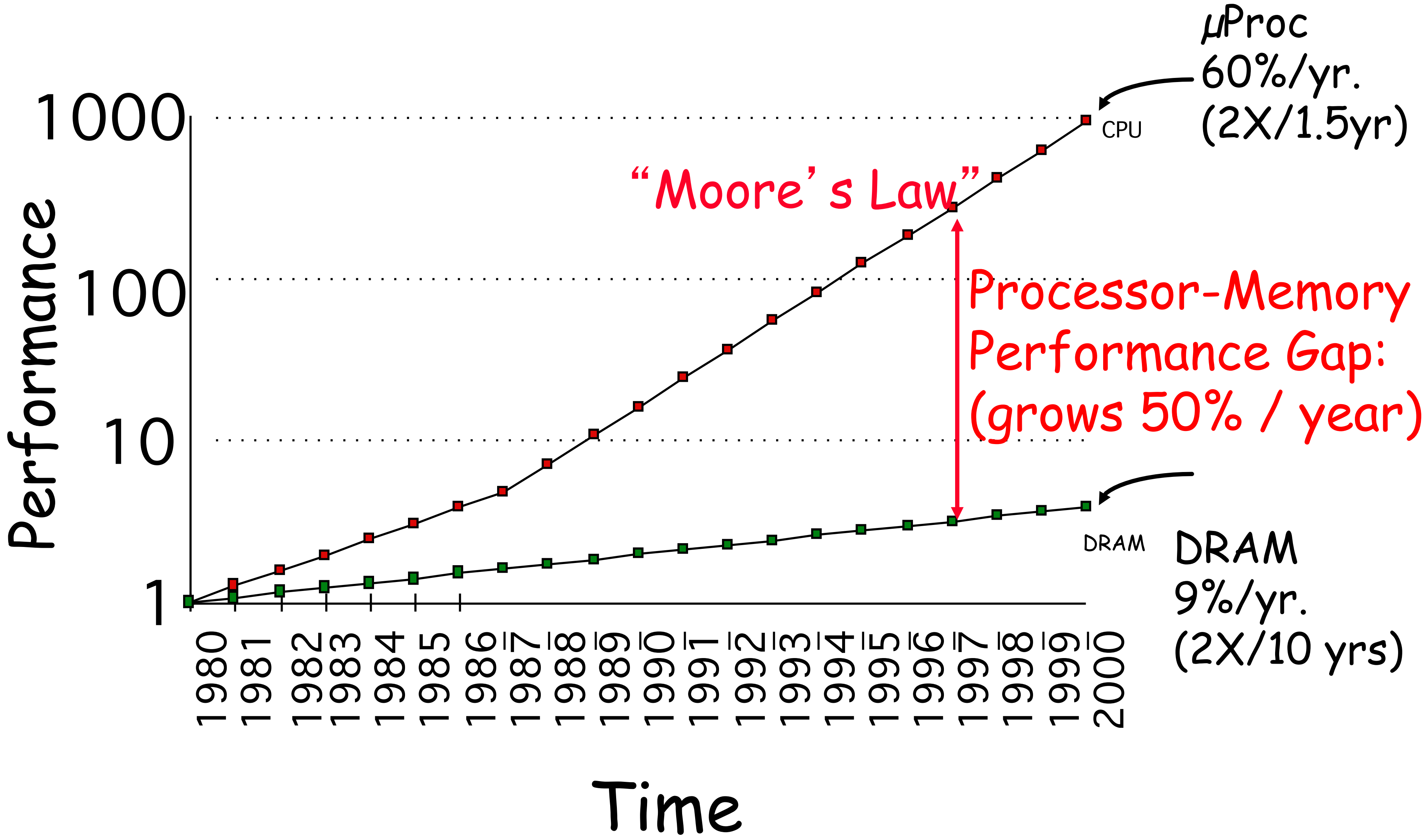
Main Memory
M Bytes
100ns- 300ns
$< 1/ MByte

Disk
10s G Bytes, 10 ms
(10,000,000 ns)
$0.001/ MByte

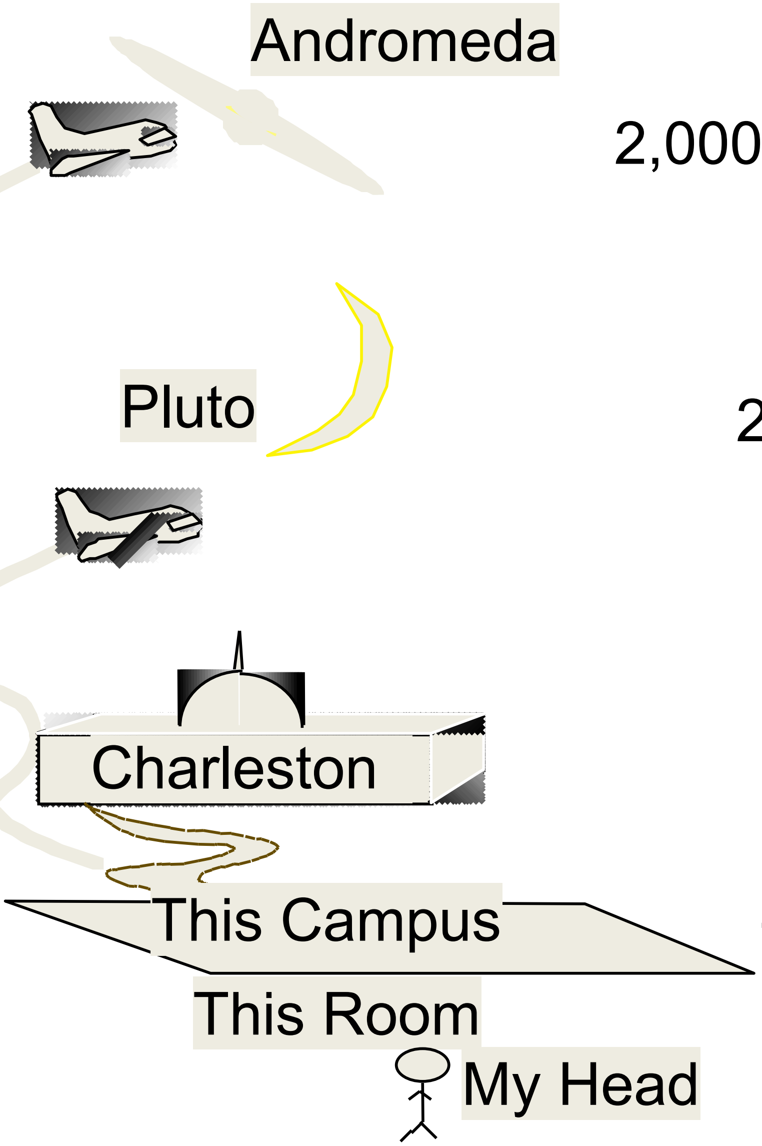
Tape
infinite
sec-min
$0.0014/ MByte



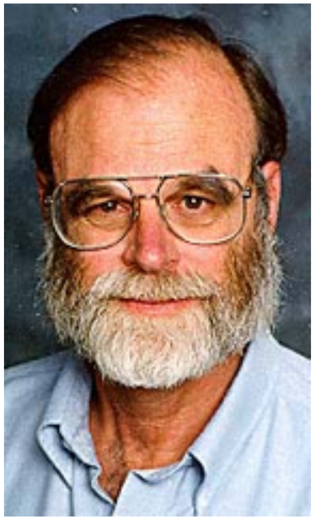
Processor-DRAM Memory Gap (latency)



Jim Gray's Storage Latency Analogy: How Far Away is the Data?



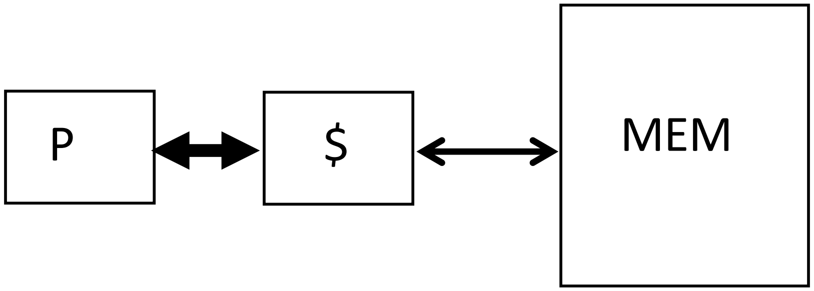
10^9	Tape /Optical Robot	Andromeda	2,000 Years
10^6	Disk	Pluto	2 Years
100	Main Memory	Charleston	2 hr
10	On Board Cache	This Campus	10 min
2	On Chip Cache	This Room	
1	Registers	My Head	1 min
(ns)			



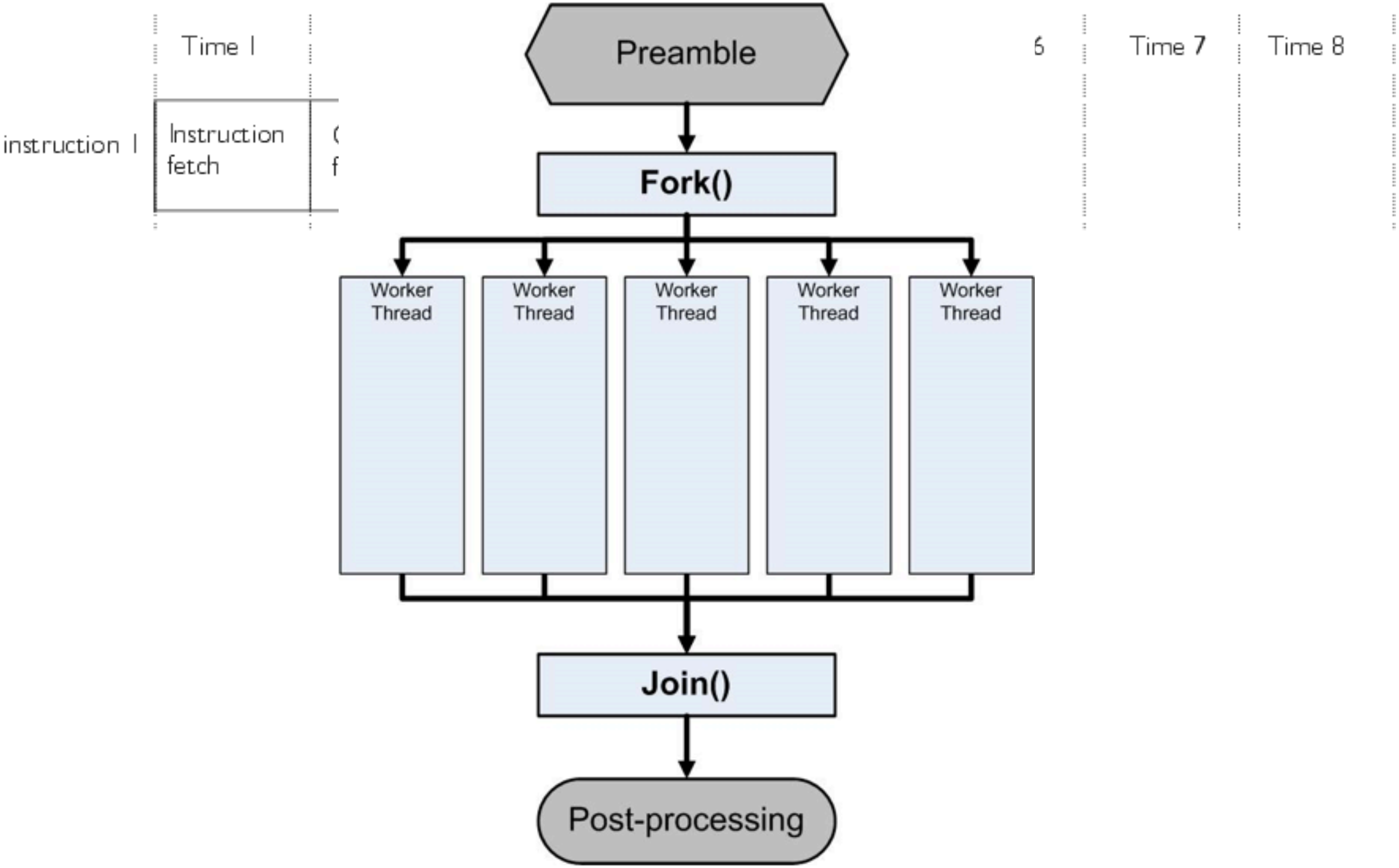
Jim Gray
Turing Award
B.S. Cal 1966
Ph.D. Cal 1969!

The Principle of Locality

- The Principle of Locality:
  - Program access a relatively small portion of the address space at any instant of time.
- Two Different Types of Locality:
  - Temporal Locality (Locality in Time): If an item is referenced, it will tend to be referenced again soon (e.g., loops, reuse)
  - Spatial Locality (Locality in Space): If an item is referenced, closeby items tend to be referenced soon (e.g., straightline code, array access)
- Last 30 years, HW relied on locality for speed



Great Idea: Parallelism

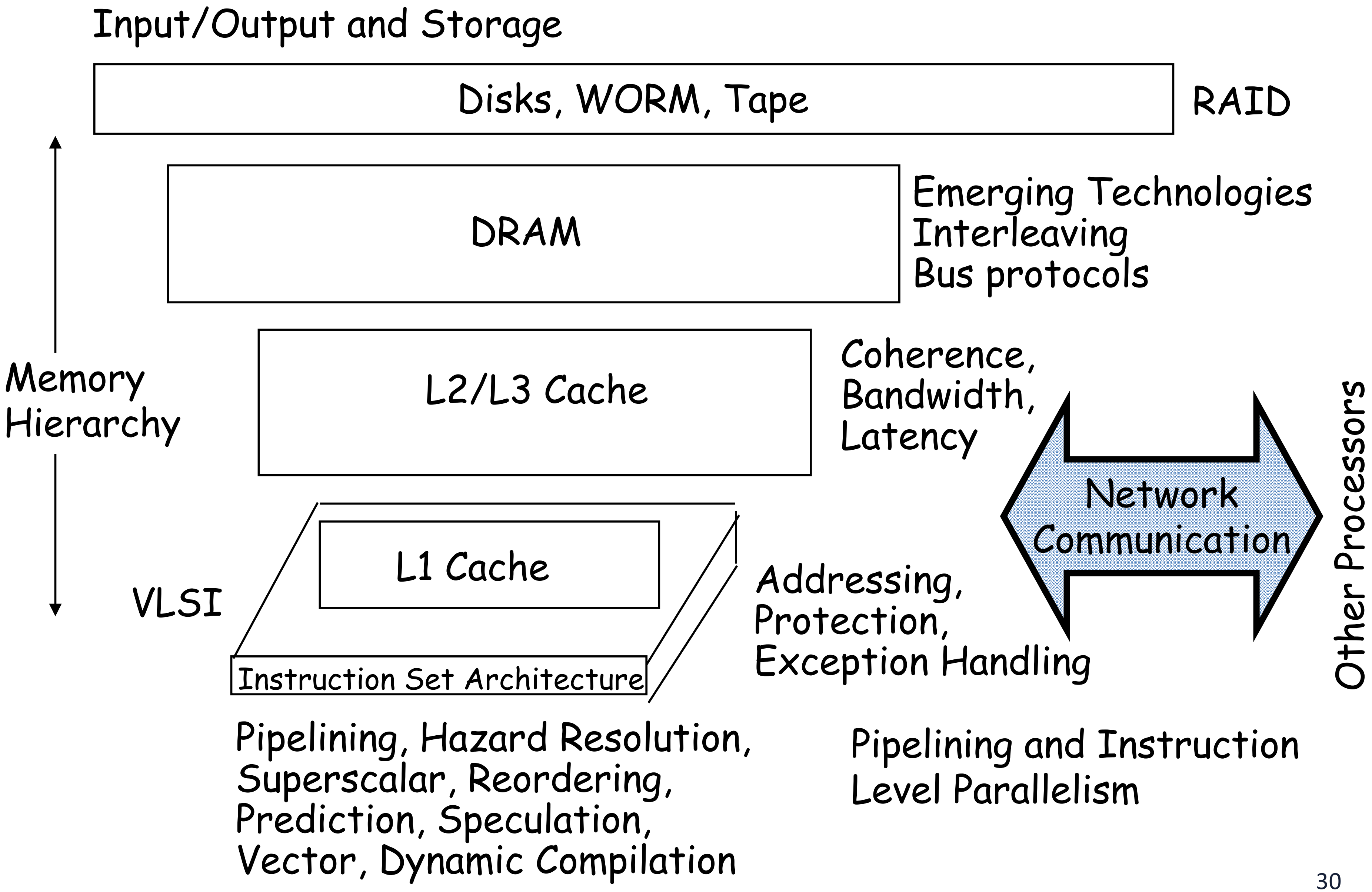


Parallelism

- Classes of parallelism in applications:
  - Data-Level Parallelism (DLP)
  - Task-Level Parallelism (TLP)

- Classes of architectural parallelism:
  - Instruction-Level Parallelism (ILP)
  - Vector architectures/Graphic Processor Units (GPUs)
  - Thread-Level Parallelism
  - Heterogeneity

Computer Architecture Topics



Input/Output and Storage

Disks, WORM, Tape

RAID

DRAM

Emerging Technologies
Interleaving
Bus protocols

Memory Hierarchy

L2/L3 Cache

Coherence,
Bandwidth,
Latency

Network Communication

Other Processors

L1 Cache

VLSI

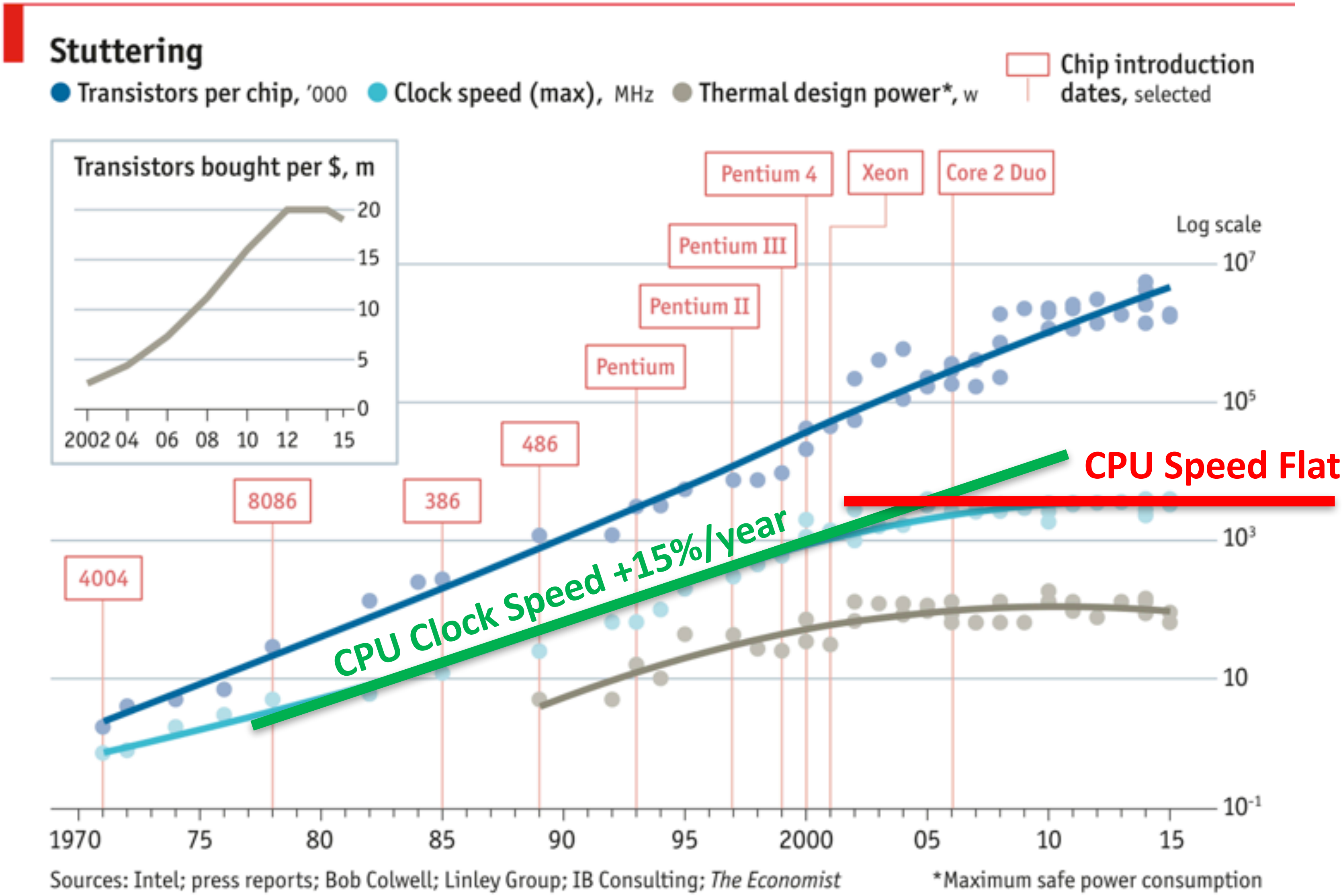
Instruction Set Architecture

Addressing,
Protection,
Exception Handling

Pipelining, Hazard Resolution,
Superscalar, Reordering,
Prediction, Speculation,
Vector, Dynamic Compilation

Pipelining and Instruction
Level Parallelism

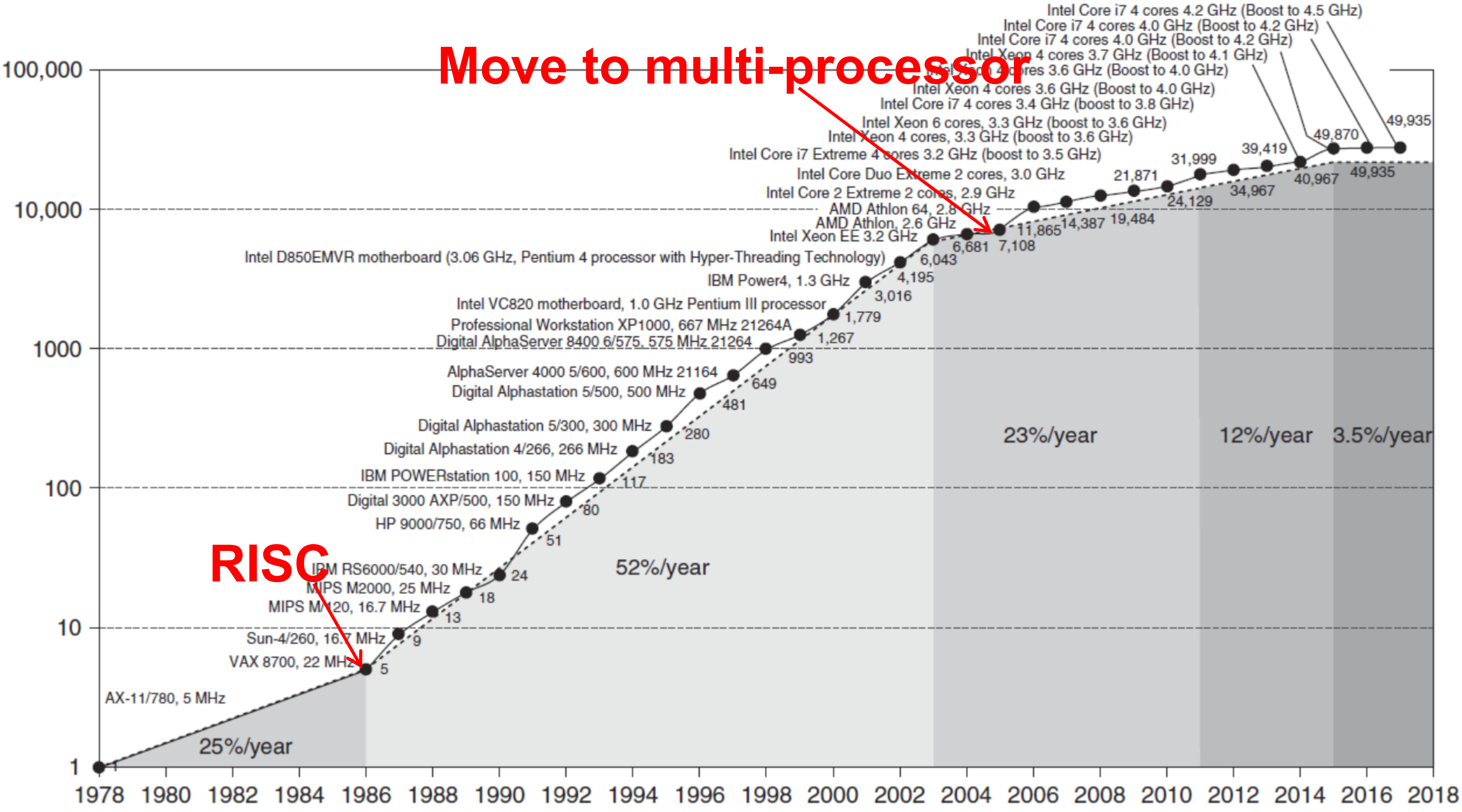
Why is Architecture Exciting Today?



Sources: Intel; press reports; Bob Colwell; Linley Group; IB Consulting; *The Economist*

*Maximum safe power consumption

Single Processor Performance



Move to multi-processor

RISC

Problems of Traditional ILP Scaling

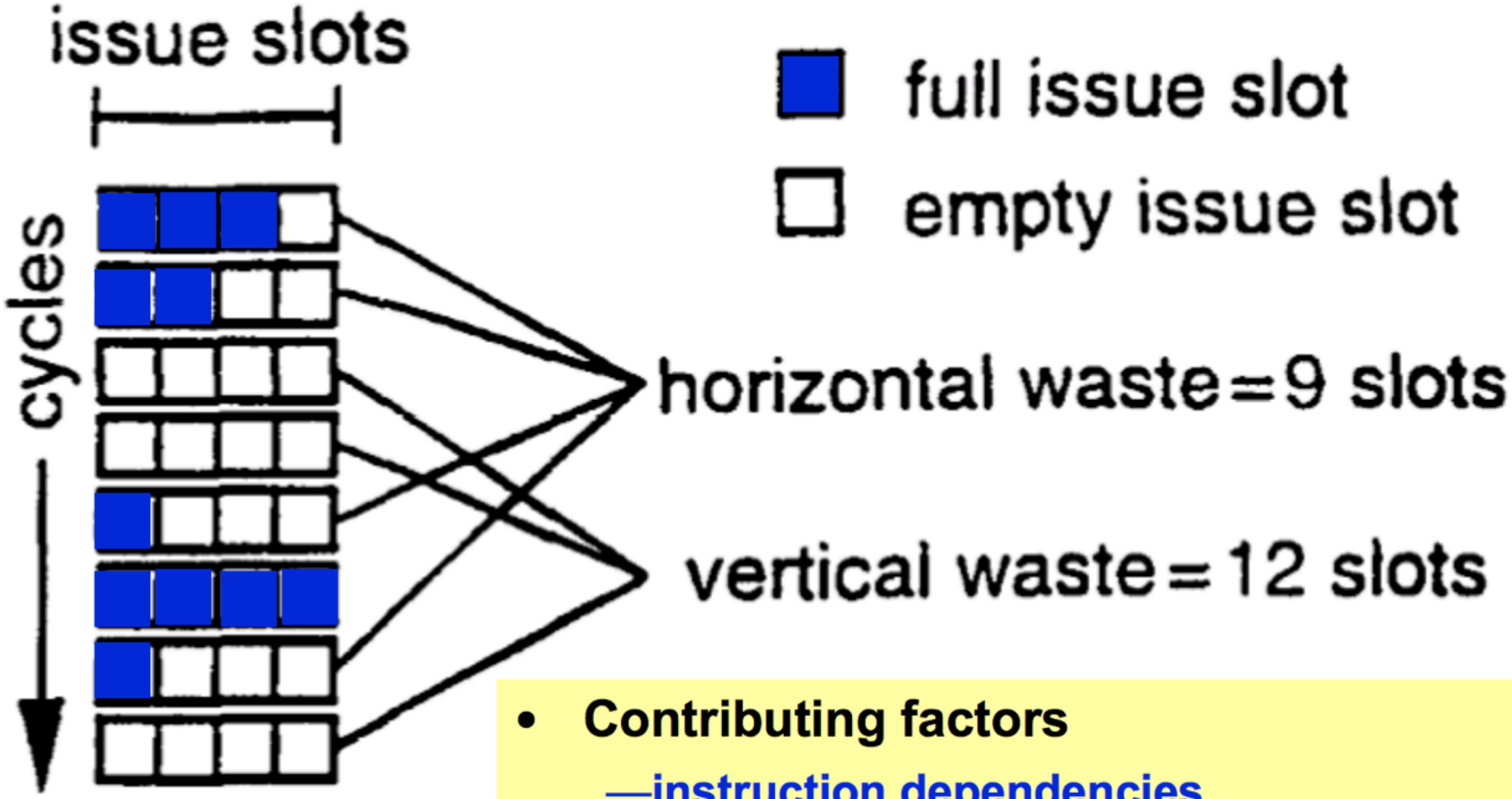
- Fundamental circuit limitations[1]
  - delays ⇑ as issue queues ⇑ and multi-port register files ⇑
  - increasing delays limit performance returns from wider issue
- Limited amount of instruction-level parallelism[1]
  - inefficient for codes with difficult-to-predict branches

- Power and heat stall clock frequencies

[1] The case for a single-chip multiprocessor, K. Olukotun, B. Nayfeh, L. Hammond, K. Wilson, and K. Chang, ASPLOS-VII, 1996.

ILP impacts

Issue Waste



■ full issue slot

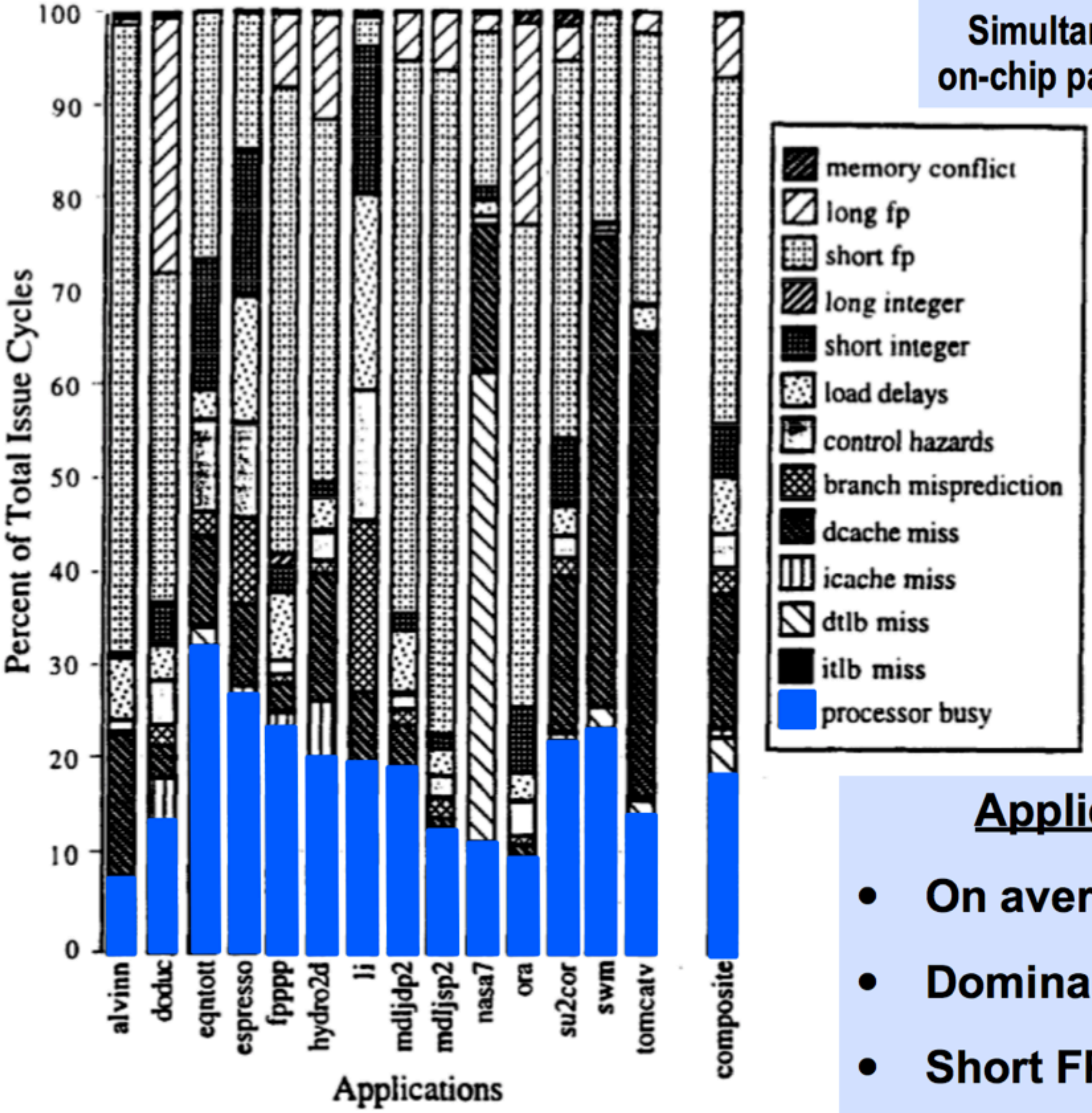
□ empty issue slot

horizontal waste=9 slots

vertical waste=12 slots

- **Contributing factors**
  - —**instruction dependencies**
  - —**long-latency operations within a thread**

Simulations of 8-issue Superscalar



Simultaneous multithreading: maximizing on-chip parallelism, Tullsen et. al. ISCA, 1995.

Summary: Highly underutilized

Applications: most of SPEC92

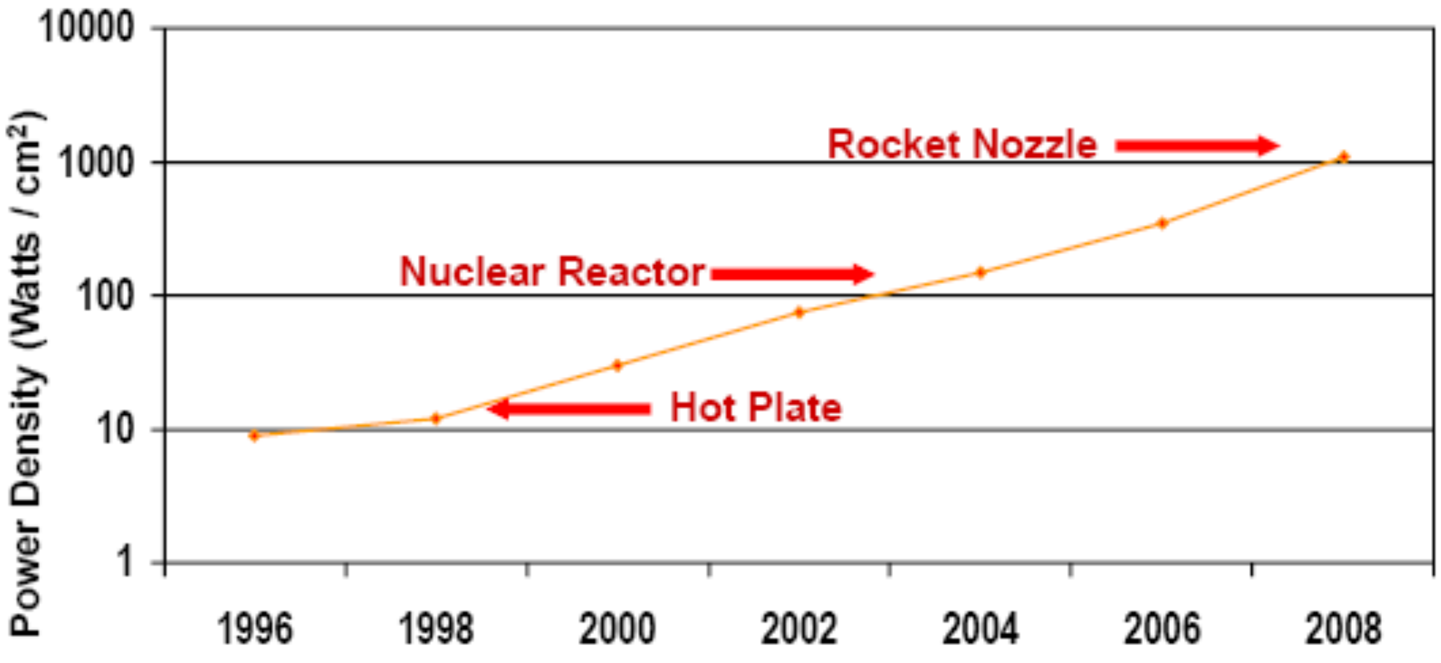
- **On average < 1.5 IPC (19%)**
- **Dominant waste differs by application**
- **Short FP dependences: 37%**

Power/Heat Density Limits Frequency

- Some fundamental physical limits are being reached

Moore’s Law Extrapolation:

Power Density for Leading Edge Microprocessors



Power Density Becomes Too High to Cool Chips Inexpensively

Source: Shekhar Borkar, Intel Corp

Recent Multicore Processors

- **Sept 13: Intel Ivy Bridge-EP Xeon E5-2695 v2**
  - — 12 cores; 2-way SMT; 30MB cache
- **March 13: SPARC T5**
  - — 16 cores; 8-way fine-grain MT per core
- **May 12: AMD Trinity**
  - — 4 CPU cores; 384 graphics cores
- **Nov 12: Intel Xeon Phi coprocessor**
  - — ~60 cores
- **Feb 12: Blue Gene/Q**
  - — 17 cores; 4-way SMT
- **Q4 11: Intel Ivy Bridge**
  - — 4 cores; 2 way SMT;
- **November 11: AMD Interlagos**
  - — 16 cores
- **Jan 10: IBM Power 7**
  - — 8 cores; 4-way SMT; 32MB shared cache
- **Tilera TilePro64**

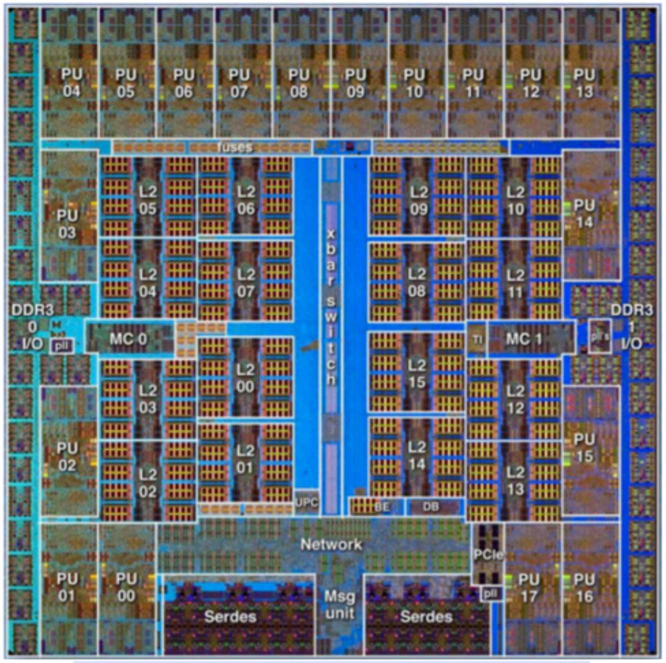
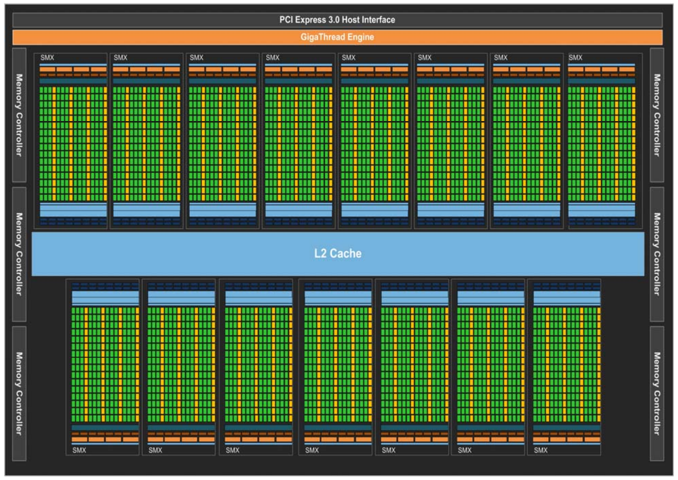
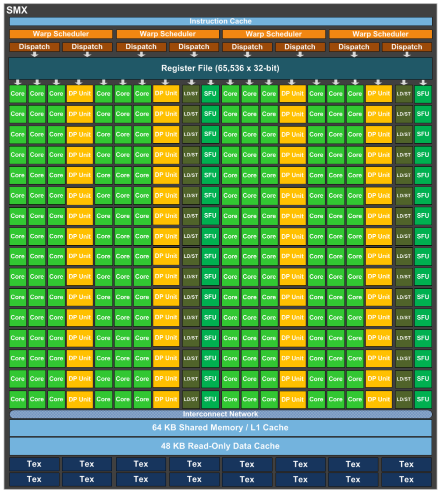


Figure credit: Ruud Haring, Blue Gene/Q compute chip, Hot Chips 23, August, 2011.

Recent Manycore GPU processors

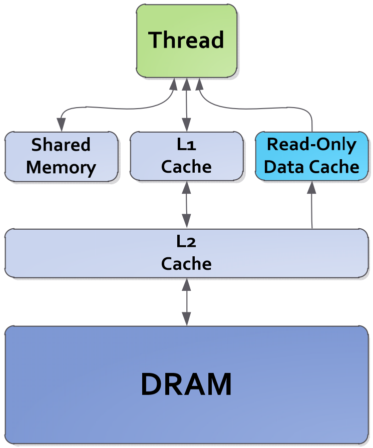
- ~5k cores

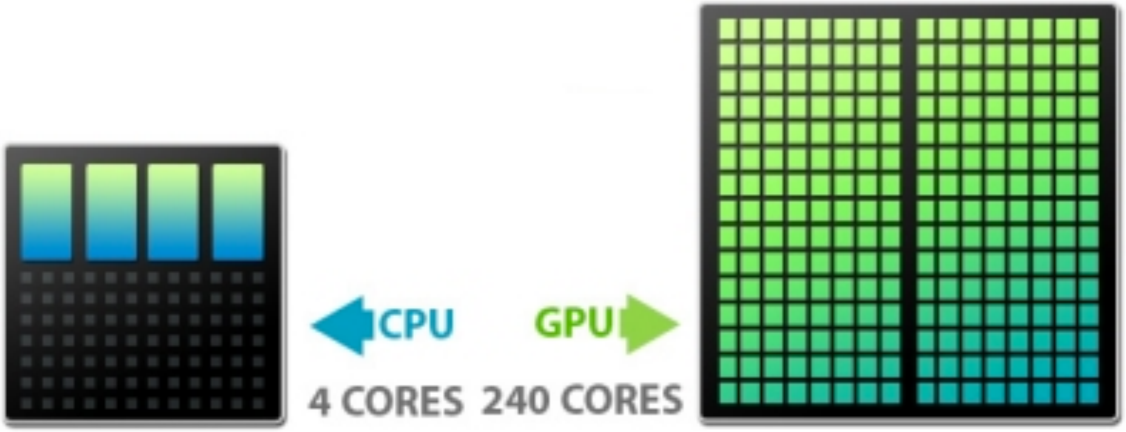



SMX
Instruction Cache
Warp Scheduler
Dispatch
Register File (65,536 x 32-bit)
Core
DP Unit
LD/ST
SFU
Interconnect Network
64 KB Shared Memory / L1 Cache
48 KB Read-Only Data Cache
Tex

SMX: 192 single-precision CUDA cores, 64 double-precision units, 32 special function units (SFU), and 32 load/store units (LD/ST).

Kepler Memory Hierarchy





Current Trends in Architecture

- Leveraging Instruction-Level parallelism (ILP) is near an end
  - Single processor performance improvement ended in 2003
- New models for performance:
  - Data-level parallelism (DLP)
  - Thread-level parallelism (TLP)

- Exciting topics and challenges
  - Heterogeneity
  - Domain specific architectures
  - Software and hardware co-design
  - Agile development

- DARPA Picks Its First Set of Winners in Electronics Resurgence Initiative, July 2018
  - https://spectrum.ieee.org/tech-talk/semiconductors/design/darpa-picks-its-first-set-of-winners-in-electronics-resurgence-initiative.amp.html

Hennessy & Patterson: A New Golden Age for Computer Architecture

By Staff

April 17, 2018

On Monday June 4, 2018, 2017 A.M. Turing Award Winners John L. Hennessy and David A. Patterson will deliver the Turing Lecture at the 45th International Symposium on Computer Architecture (ISCA) in Los Angeles.

- Video: https://www.acm.org/hennessy-patterson-turing-lecture
- Short summary
  - https://www.hpcwire.com/2018/04/17/hennessy-patterson-a-new-golden-age-for-computer-architecture/

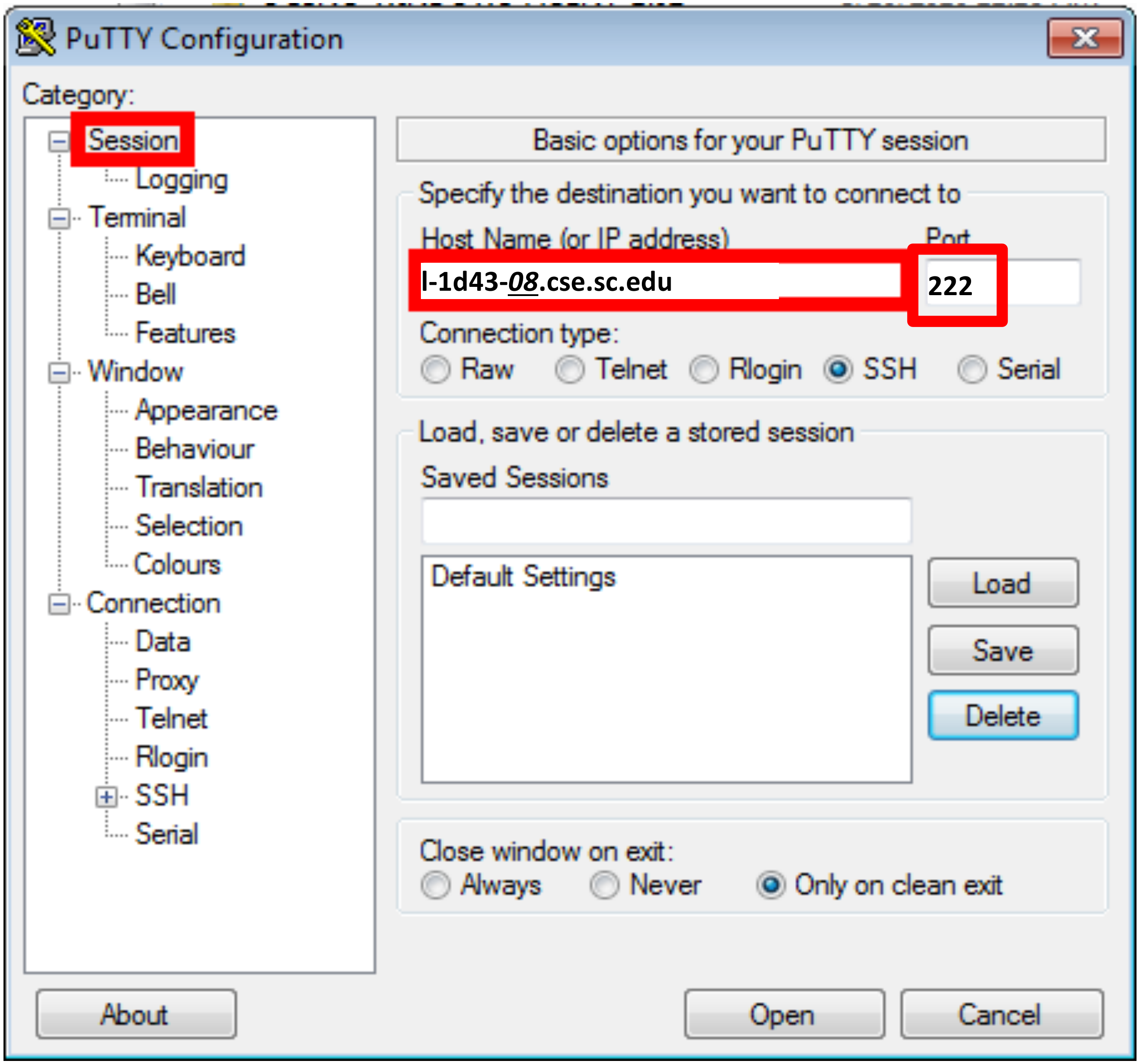
Exercise: Inspect ISA for sum

- Sum example
  - https://passlab.github.io/CSCE513/exercises/sum
- Check
  - sum_full.s,
  - sum_riscv.s
  - sum_x86.s
- Generate and execute
  - gcc -save-temps sum.c –o sum
  - ./sum 102400
- For how to compile and run Linux program
  - https://passlab.github.io/CSCE513/notes/lecture01_LinuxCProgramming.pdf
- Other system commands:
  - cat /proc/cpuinfo to show the CPU and #cores
  - top command to show system usage and memory

Machine for Development and Experiment

- **Linux machines in Swearingen 1D43 and 3D22**
  - **All CSCE students by default have access to these machine using their standard login credentials**
    - **Let me know if you, CSCE or not, cannot access**
  - **Remote access is also available via SSH over port 222. Naming schema is as follows:**
    - **l-1d43-*01*.cse.sc.edu through l-1d43-*26*.cse.sc.edu**
    - **l-3d22-*01*.cse.sc.edu through l-3d22-*20*.cse.sc.edu**
- **Restricted to 2GB of data in their home folder (~/).**
  - **For more space, create a directory in /scratch on the login machine, however that data is not shared and it will only be available on that specific machine.**

Putty SSH Connection on Windows



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): l-1d43-*08*.cse.sc.edu

Port: 222

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

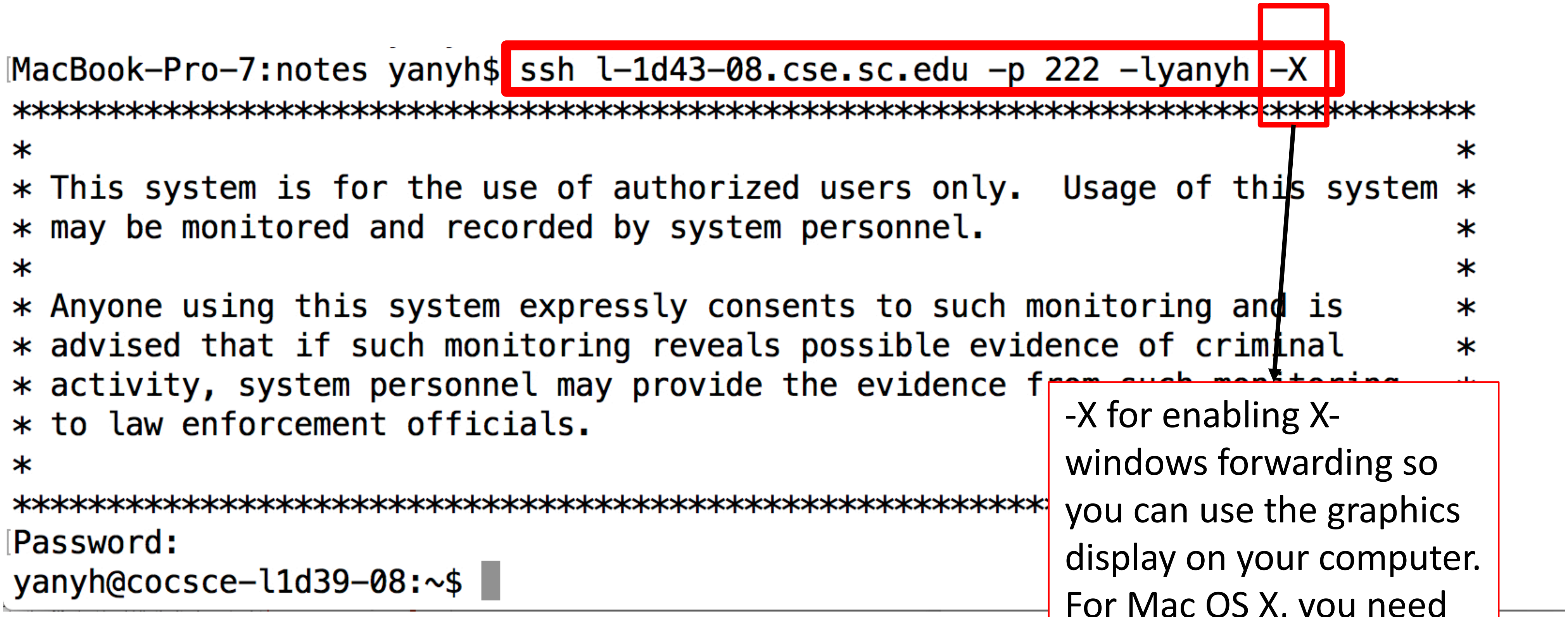
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

SSH Connection from Linux/Mac OS X Terminal



```
MacBook-Pro-7:notes yanyh$ ssh l-1d43-08.cse.sc.edu -p 222 -lyanyh -X
*****************************************************************************
*                                                                           *
* This system is for the use of authorized users only.  Usage of this system *
* may be monitored and recorded by system personnel.                         *
*                                                                           *
* Anyone using this system expressly consents to such monitoring and is     *
* advised that if such monitoring reveals possible evidence of criminal     *
* activity, system personnel may provide the evidence from such monitoring  *
* to law enforcement officials.
*
*********************************************************
Password:
yanyh@cocsce-l1d39-08:~$
```

-X for enabling X-windows forwarding so you can use the graphics display on your computer. For Mac OS X, you need have X server software installed, e.g. Xquartz(https://www.xquartz.org/) is the one I use.