Course Outline

I. Introduction

II. Digital Logic Design

- Boolean algebra (review)
- logic gates and fundamental components
- combinational and sequential logic
- integer addition
- counters

III. Datapath Design and Analysis

- single-cycle MIPS32 datapath
- defining and measuring performance
- basic pipelined MIPS32 datapath
- exceptions

IV. Memory Hierarchy

- SRAM vs DRAM
- cache memory
- virtual memory
- secondary storage

The Computer Revolution

Progress in computer technology

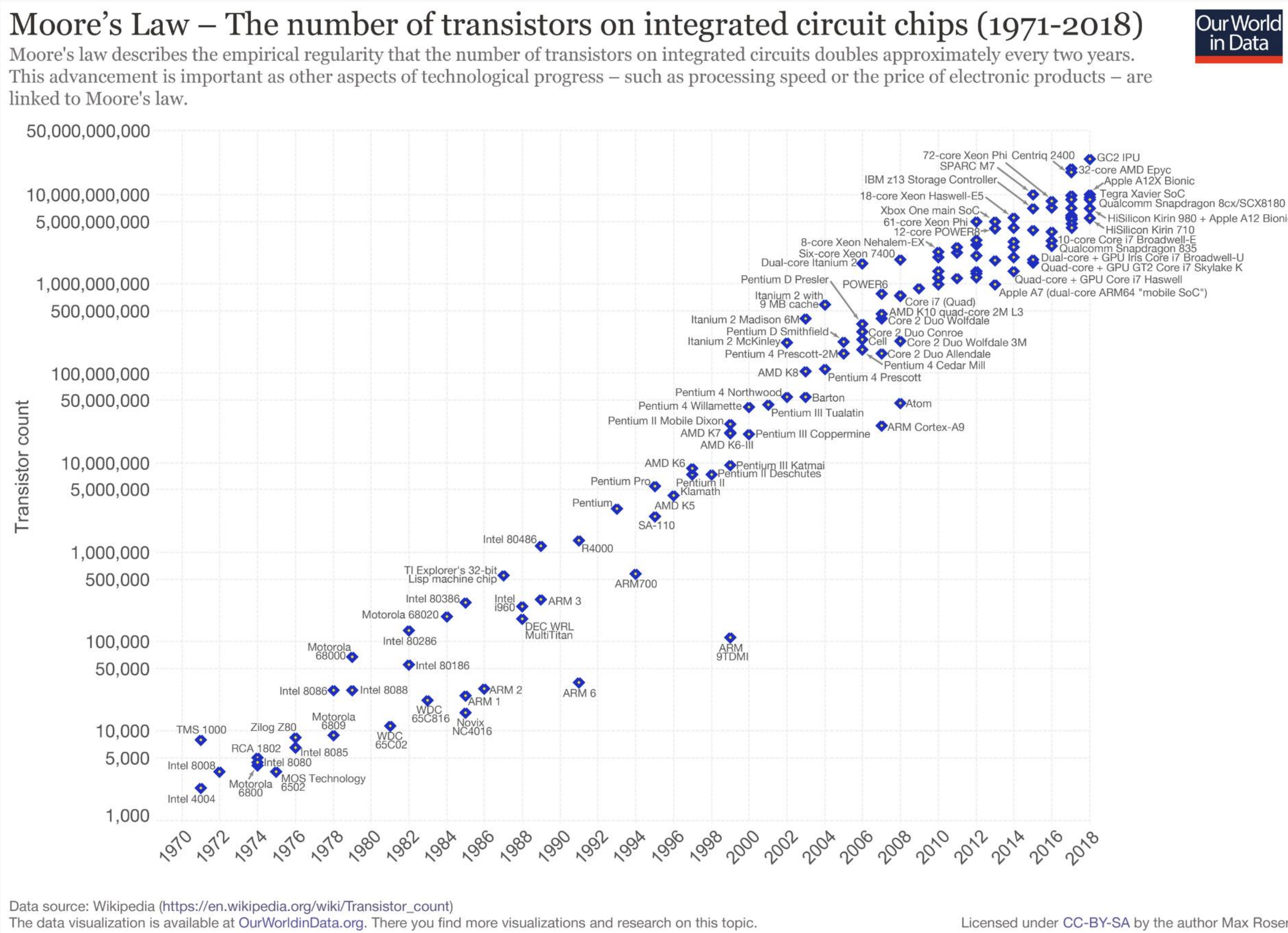
- Underpinned by Moore's Law

> The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years.
>
> Gordon Moore (1965)

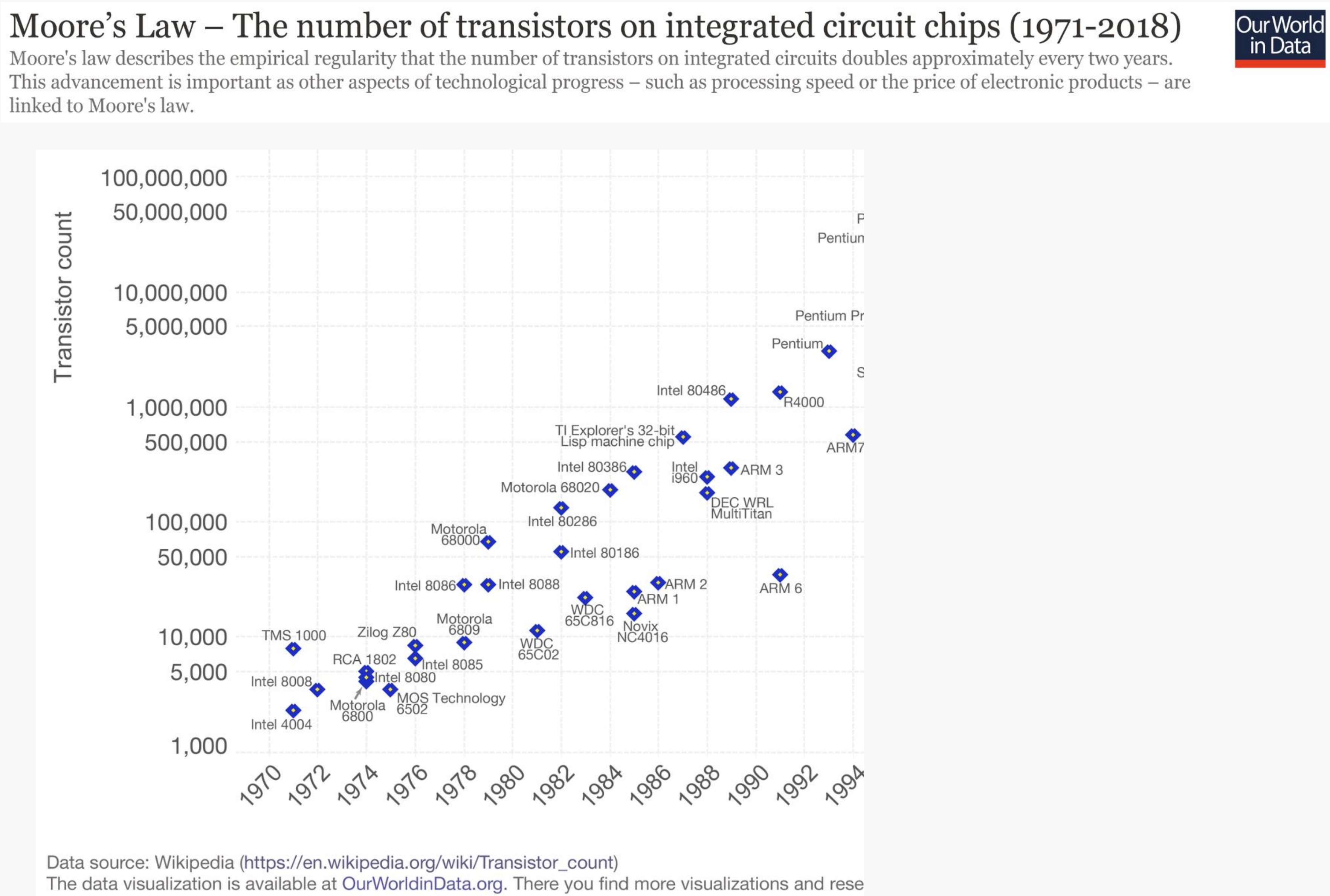
Makes novel applications feasible

- Computers in automobiles
- Cell phones
- Human genome project
- World Wide Web
- Search Engines

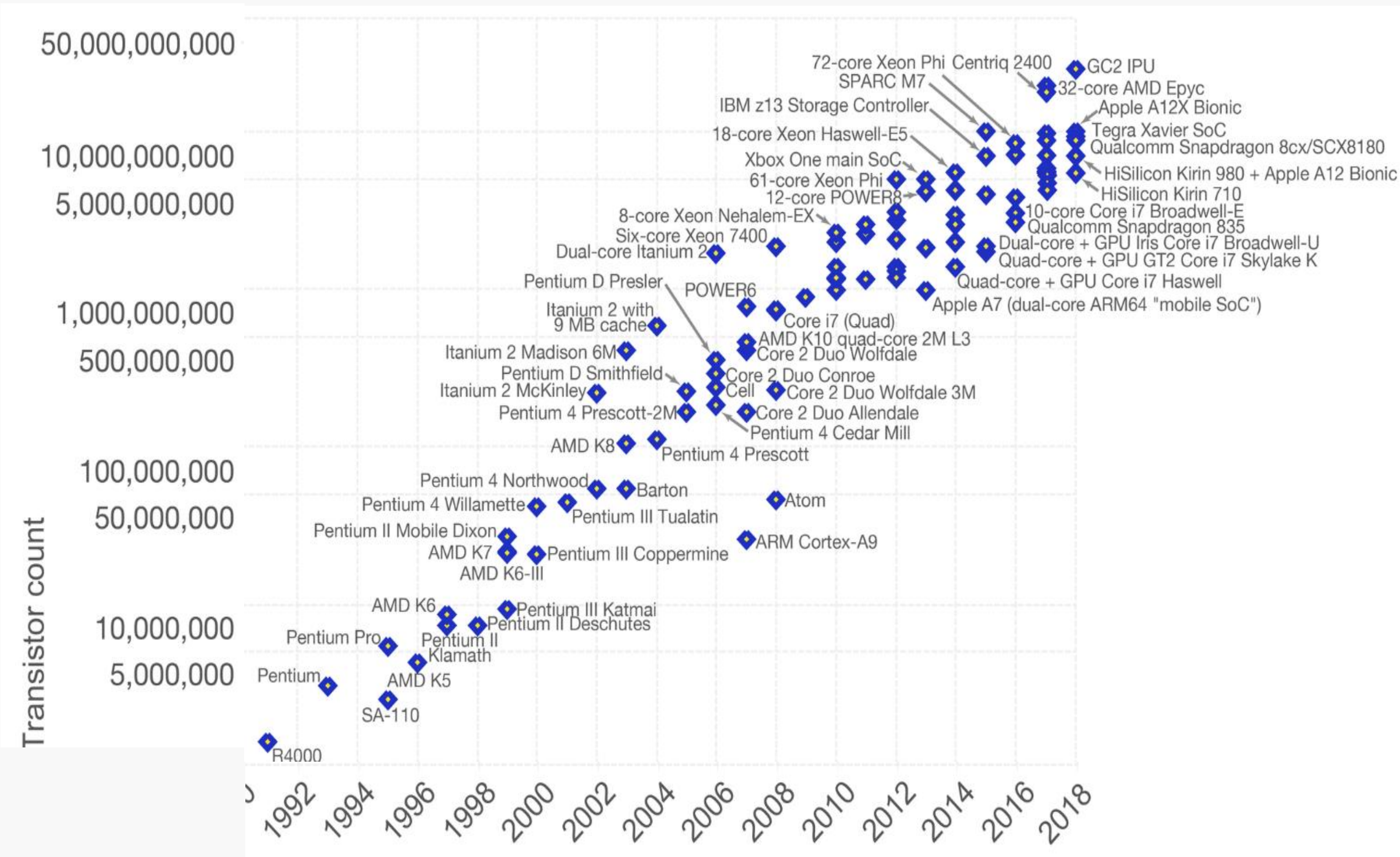
Computers are pervasive


Moore's Law – The number of transistors on integrated circuit chips (1971-2018)
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.
Our World in Data
Transistor count
50,000,000,000
10,000,000,000
5,000,000,000
1,000,000,000
500,000,000
100,000,000
50,000,000
10,000,000
5,000,000
1,000,000
500,000
100,000
50,000
10,000
5,000
1,000
1970 1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018
72-core Xeon Phi
Centriq 2400
GC2 IPU
SPARC M7
32-core AMD Epyc
IBM z13 Storage Controller
Apple A12X Bionic
18-core Xeon Haswell-E5
Tegra Xavier SoC
Qualcomm Snapdragon 8cx/SCX8180
Xbox One main SoC
61-core Xeon Phi
HiSilicon Kirin 980 + Apple A12 Bionic
12-core POWER8
HiSilicon Kirin 710
8-core Xeon Nehalem-EX
10-core Core i7 Broadwell-E
Six-core Xeon 7400
Qualcomm Snapdragon 835
Dual-core Itanium 2
Dual-core + GPU Iris Core i7 Broadwell-U
Quad-core + GPU GT2 Core i7 Skylake K
Pentium D Presler
POWER6
Quad-core + GPU Core i7 Haswell
Itanium 2 with 9 MB cache
Apple A7 (dual-core ARM64 "mobile SoC")
Core i7 (Quad)
AMD K10 quad-core 2M L3
Itanium 2 Madison 6M
Core 2 Duo Wolfdale
Pentium D Smithfield
Core 2 Duo Conroe
Itanium 2 McKinley
Cell
Core 2 Duo Wolfdale 3M
Pentium 4 Prescott-2M
Core 2 Duo Allendale
Pentium 4 Cedar Mill
AMD K8
Pentium 4 Prescott
Pentium 4 Northwood
Barton
Pentium 4 Willamette
Atom
Pentium III Tualatin
Pentium II Mobile Dixon
AMD K7
Pentium III Coppermine
ARM Cortex-A9
AMD K6-III
AMD K6
Pentium III Katmai
Pentium II Deschutes
Pentium Pro
Pentium II Klamath
AMD K5
Pentium
SA-110
Intel 80486
R4000
TI Explorer's 32-bit Lisp machine chip
ARM700
Intel 80386
Intel i960
ARM 3
Motorola 68020
DEC WRL MultiTitan
Intel 80286
ARM 9TDMI
Motorola 68000
Intel 80186
Intel 8086
Intel 8088
ARM 2
ARM 6
WDC 65C816
ARM 1
Novix NC4016
TMS 1000
Zilog Z80
Motorola 6809
WDC 65C02
RCA 1802
Intel 8085
Intel 8008
Intel 8080
MOS Technology 6502
Motorola 6800
Intel 4004
Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.
Licensed under CC-BY-SA by the author Max Roser.

The Computer Revolution



The Computer Revolution



8 Great Ideas in Computing

Design for ***Moore's Law***

Use ***abstraction*** to simplify design

Make the ***common case fast***

Performance *via* ***parallelism***

Performance *via* ***pipelining***

Performance *via* ***prediction***

Hierarchy of memories

Dependability *via* redundancy

This course is all about how computers work

But what do we mean by a computer?

- Different types: desktop, servers, embedded devices
- Different uses: automobiles, graphics, structural analysis, finance, genomics…
- Different manufacturers: Intel, Apple, IBM, Microsoft, Sun…
- Different underlying technologies and different costs!

Classes of Computers

Personal computers

- General purpose, variety of software
- Subject to cost/performance tradeoff

Server computers

- Network based
- High capacity, performance, reliability
- Range from small servers to building sized

Supercomputers

- High-end scientific and engineering calculations
- Highest capability but represent a small fraction of the overall computer market

Embedded computers

- Hidden as components of systems
- Stringent power/performance/cost constraints

Pedagogy

Analogy: Consider a course on “automotive vehicles”

- Many similarities from vehicle to vehicle (e.g., wheels)
- Huge differences from vehicle to vehicle (e.g., gas vs. electric)

Best way to learn:

- Focus on a specific instance and learn how it works
- While learning general principles and historical perspectives

Why learn this stuff?

You want to call yourself a “computer scientist”

You want to build software people use (need performance)

You need to make a purchasing decision or offer “expert” advice

Debugging skills often benefit from understanding architecture

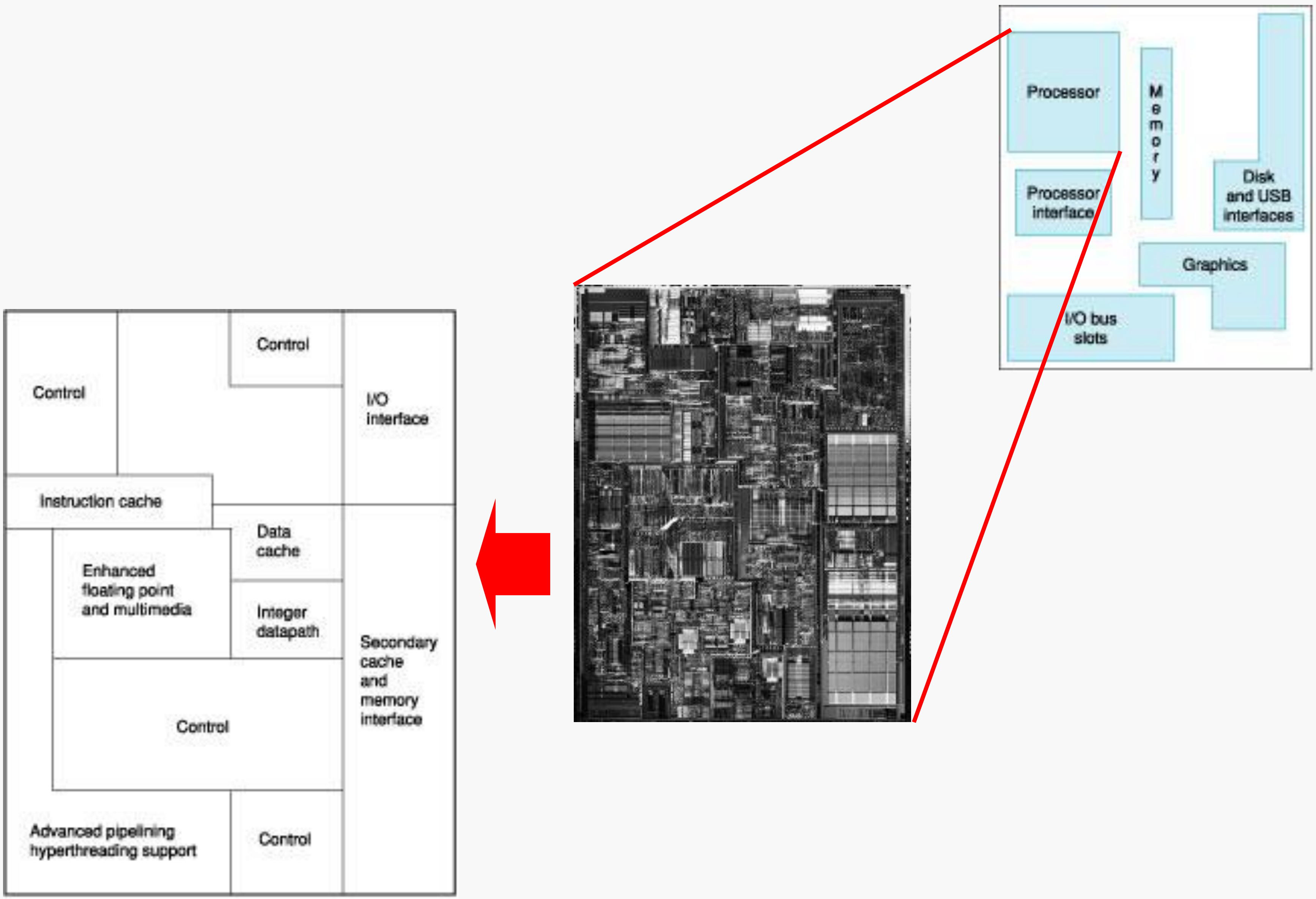
- better understand system error messages
- better understand translators (compilers and interpreters)

Both hardware and software affect performance:

- Algorithm determines number of source-level statements
- Language/Compiler/Architecture determine machine instructions
- Processor/Memory determine how fast instructions are executed

Assessing and Understanding Performance in Chapter 4

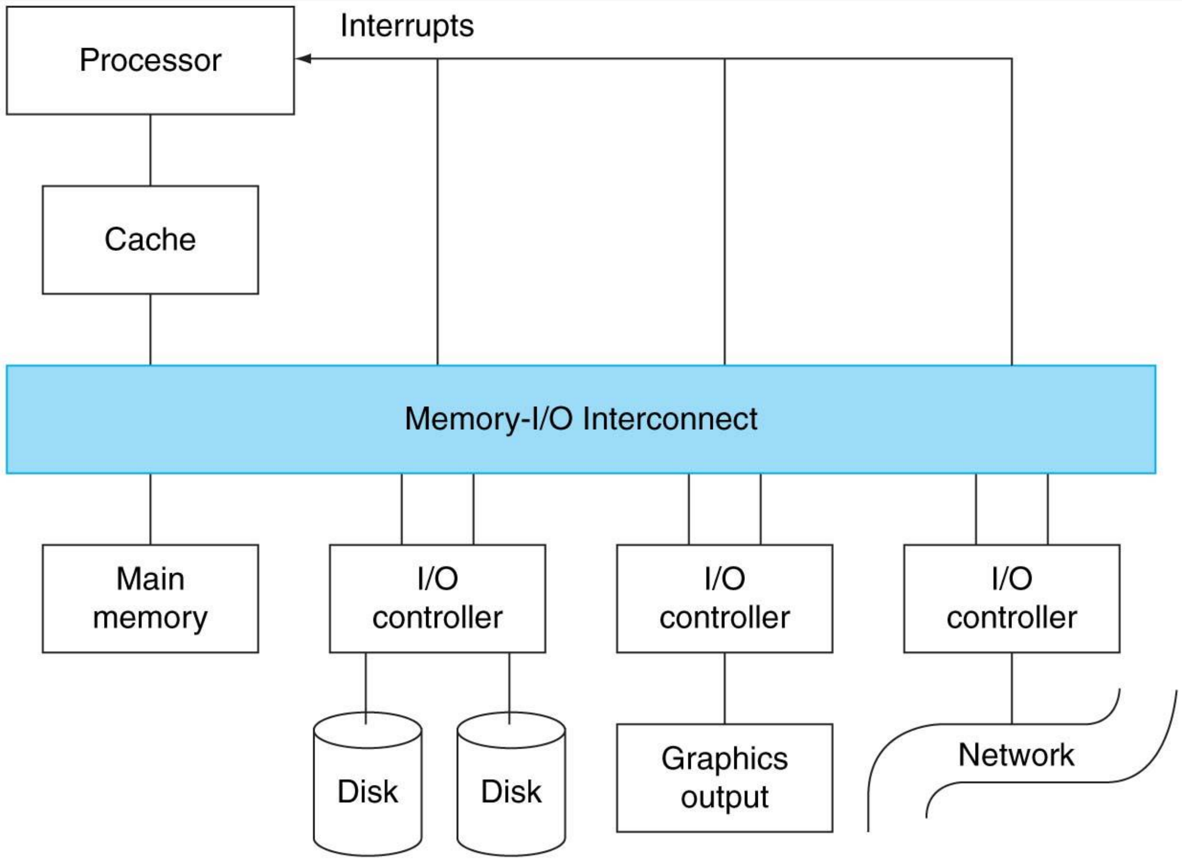
Focus: the Processor



Focus: the Memory Hierarchy

	Processor			
Fastest	Cache	Smallest	$$$$$	SRAM
	Primary memory			DRAM
Slowest	Secondary storage	Largest	$	SSD Magnetic disk

Focus: I/O


Processor
Interrupts
Cache
Memory-I/O Interconnect
Main memory
I/O controller
I/O controller
I/O controller
Disk
Disk
Graphics output
Network

Below Your Program

Application software

Written in high-level language

System software

Compiler/Assembler/Linker:

translates HLL code to machine code

Operating System: service code

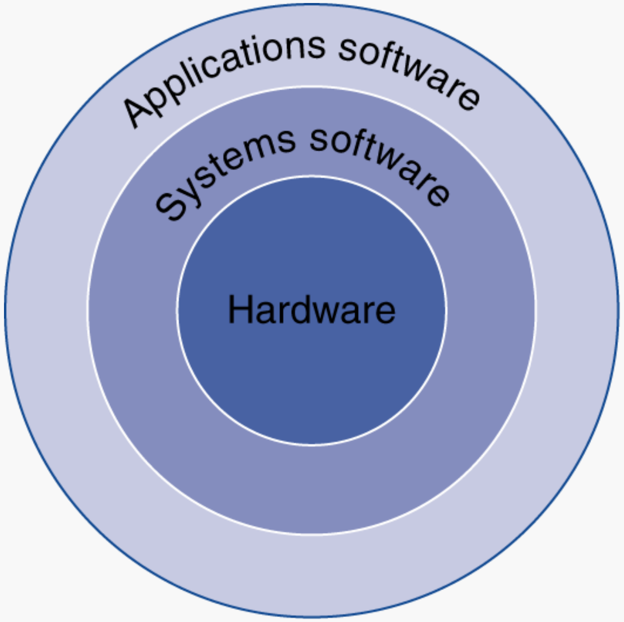
Handling input/output

Managing memory and storage

Scheduling tasks & sharing resources

Hardware

Processor, memory, I/O controllers



Language Abstractions

Delving into the depths reveals more information

An abstraction omits unneeded detail, helps us cope with complexity

What are some of the details that appear in these familiar abstractions?

High-level language (C)

```
swap(int v[], int k) {
   int temp;
   temp   = v[k];
   v[k]   = v[k+1];
   v[k+1] = temp;
}
```

compiler

Assembly language (MIPS)

```
swap:
   muli  $2, $5, 4
   add   $2, $4, $2
   lw    $15, 0($2)
   lw    $16, 4($2)
   sw    $16, 0($2)
   sw    $16, 4($2)
   jr    $31
```

assembler

```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Binary machine language (for MIPS)

How do computers work?

Need to understand abstractions such as:

- Applications software
- Systems software
- Assembly Language
- Machine Language
- Architectural Issues: i.e., Caches, Virtual Memory, Pipelining
- Sequential logic, finite state machines
- Combinational logic, arithmetic circuits
- Boolean logic, 1s and 0s

- Transistors used to build logic gates (CMOS)
- Semiconductors/Silicon used to build transistors
- Properties of atoms, electrons, and quantum dynamics

So much to learn!

50 Years of Progress

Year	Name	Size (cu. ft.)	Power (watts)	Performance (adds/sec)	Memory (KB)	Price	Price-performance vs. UNIVAC	Adjusted price (2003 $)	Adjusted price-performance vs. UNIVAC
1951	UNIVAC I	1,000	125,000	2,000	48	$1,000,000	1	$6,107,600	1
1964	IBM S/360 model 50	60	10,000	500,000	64	$1,000,000	263	$4,792,300	318
1965	PDP-8	8	500	330,000	4	$16,000	10,855	$75,390	13,135
1976	Cray-1	58	60,000	166,000,000	32,000	$4,000,000	21,842	$10,756,800	51,604
1981	IBM PC	1	150	240,000	256	$3,000	42,105	$5,461	154,673
1991	HP 9000/ model 750	2	500	50,000,000	16,384	$7,400	3,556,188	$9,401	16,122,356
1996	Intel PPro PC (200 MHz)	2	500	400,000,000	16,384	$4,400	47,846,890	$4,945	239,078,908
2003	Intel Pentium 4 PC (3.0 GHz)	2	500	6,000,000,000	262,144	$1,600	1,875,000,000	$1,600	11,452,000,000

Units of Measurement

Spatial units:

Decimal term	Abbreviation	Value	Binary term	Abbreviation	Value	% Larger
kilobyte	KB	10^3	kibibyte	KiB	2^{10}	2%
megabyte	MB	10^6	mebibyte	MiB	2^{20}	5%
gigabyte	GB	10^9	gibibyte	GiB	2^{30}	7%
terabyte	TB	10^{12}	tebibyte	TiB	2^{40}	10%
petabyte	PB	10^{15}	pebibyte	PiB	2^{50}	13%
exabyte	EB	10^{18}	exbibyte	EiB	2^{60}	15%
zettabyte	ZB	10^{21}	zebibyte	ZiB	2^{70}	18%
yottabyte	YB	10^{24}	yobibyte	YiB	2^{80}	21%

Time units:

picosecond (ps) one-trillionth (10^{-12}) of a second

nanosecond (ns) one-billionth (10^{-9}) of a second

microsecond (ms) one-millionth (10^{-6}) of a second

millisecond (ms) one-thousandth (10^{-3}) of a second

Light in vacuum travels 1 foot in 1.016703362164 nanoseconds…