



CSC 252/452: Computer Organization

Fall 2020
Instructor: Sandhya Dwarkadas

Department of Computer Science
University of Rochester

1

1





Class Overview

- Introductions
 - Instructor, TAs, and classmates
 - Prerequisites: MATH 150 and CSC 172
 - Where to find information
 - Class, quizzes, assignments, exams, and grading
 - What is Computer Organization?




2

2

Meet the TAs

Loc Bui Elana Elman Nathan Reed Sudhanshu Gupta

Shuang Zhai Vladimir Maksimovski Yudi Yang

3

3

Class Overview

- Introductions
 - Instructor, TAs, and classmates
 - Prerequisites: MATH 150 and CSC 172
 - Where to find information
 - Class, quizzes, assignments, exams, and grading
 - What is Computer Organization?

4

4

Where to Find Information

- Class web page (cross-linked to blackboard page)
 - <http://www.cs.rochester.edu/u/sandhya/csc252>
 - Contains the schedule, readings, assignments, useful links, policies
- Class blackboard page (cross-linked to web page)
 - Lecture recordings
 - Quizzes, pre-assignments, exams
- CSUG machines
 - Assignment code, programming, programming assignment submission

5

5

Academic Honesty Policy

- Your objective in taking the course: to learn (I hope!)
 - The objective of activities in the course is to help you learn and to assess whether you have internalized the material
 - Collaboration encouraged at the level of ideas and tools
 - All work turned in, however, must be completed on your own
- ACTION ITEM:** Accept the academic honesty policy online

6

6

Textbook

- Required course textbook
 - Bryant and O'Hallaron's Computer Systems: A Programmer's Perspective (3rd edition)
- Other recommendations
 - Introduction to Computing Systems: From Bits and Gates to C and Beyond, 2/e.
 - Computer Organization and Design: The Hardware Software Interface, ARM Edition. More emphasis on hardware
 - *The C Programming Language*, Second Edition, Brian Kernighan and Dennis Ritchie, Prentice Hall, 1988

7

7

Evaluation: Assignments and Grading

- In-class problem solving and corresponding “quizzes”
- 6-8 programming and written assignments
- Midterm and final exam
- Tentative division of grades: 65% programming and written assignments, quizzes; 35% midterm and final exams
- Late assignments: 10% (of the total assignment score) penalty per day up to 3 days (assignments not accepted after this); up to 3 slip days per person across all assignments, be careful with this when working in teams

8

8

Class Overview

- Introductions
 - Instructor, TAs, and classmates
 - Prerequisites: MATH 150 and CSC 172
 - Where to find information
 - Class, quizzes, assignments, exams, and grading
 - What is Computer Organization?

9

9

Computer Organization

- Goal: In-depth understanding of the inner workings of modern computer systems
- Study the hierarchy of layers that comprise computer systems
 - Hardware
 - Systems software
 - Applications software

10

10

The Principle of Abstraction

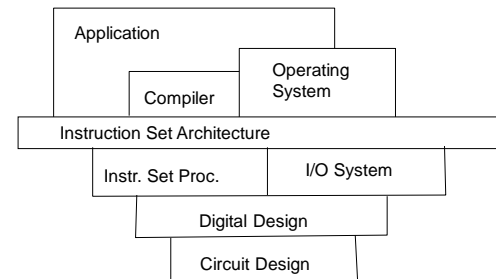
- Grouping principle
 - Levels/layers of abstraction by which each layer only needs to understand that immediately above and below it

11

11

Computer Architecture

- Coordination of levels of abstraction



12

12

Useful Outcomes

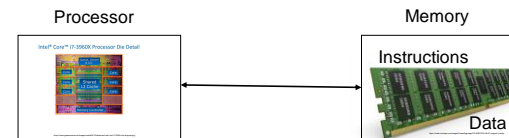
- Become more effective programmers
 - Able to find and eliminate bugs efficiently
 - Able to understand and tune for program performance and resource efficiency
- Prepare for later “systems” classes in CS & ECE
 - Operating Systems, Networks, Computer Architecture, Parallel and Distributed Systems, etc.

13

13

Program as Data

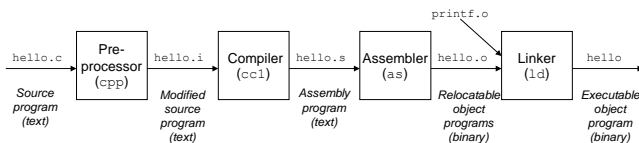
- Fundamental idea of general-purpose computing (referred to as the [Von Neumann architecture](#))
 - Program (instructions) stored in memory
 - Processors read instructions from memory and execute them one after the other



14

14

What happens to your program?: A Tool View



```
#include <stdio.h>

int main() {
    printf("hello, world\n");
}
```

15

15

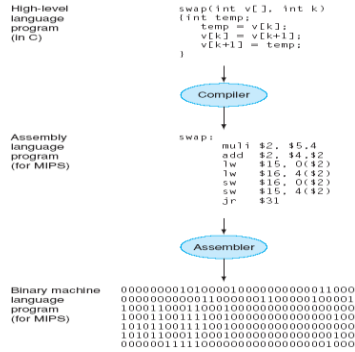
Instruction Set Architecture

- Key interface between levels of abstraction
- Interface between hardware and lowest level software
- Interface/functionality separated from implementation
 - Implementations of varying cost and performance can run identical software
- Implementation influences design

16

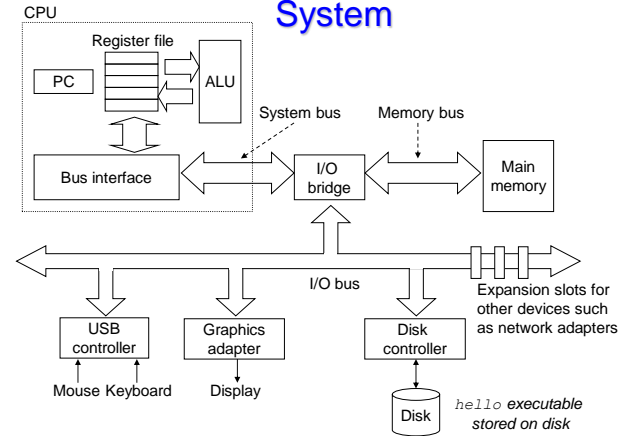
16

The Compilation Process: A Data View



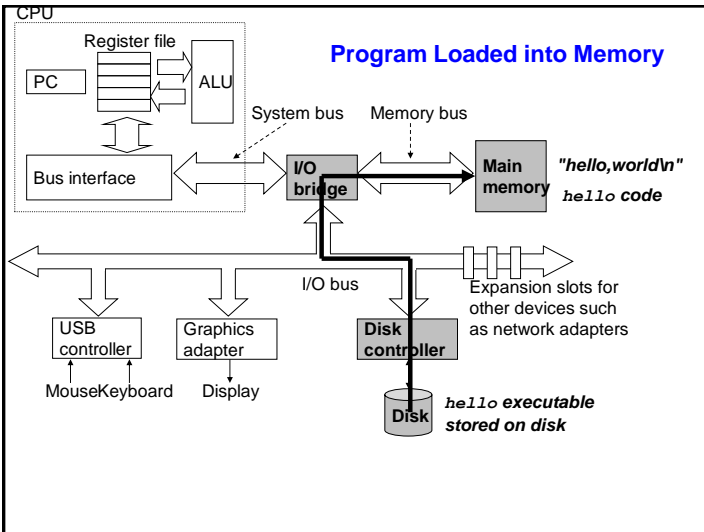
17

Hardware Organization of a Typical System



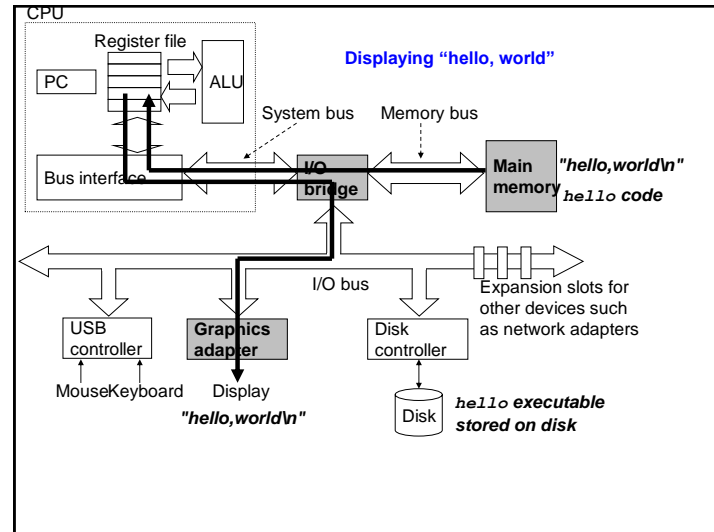
18

Program Loaded into Memory



19

Displaying "hello, world"



20

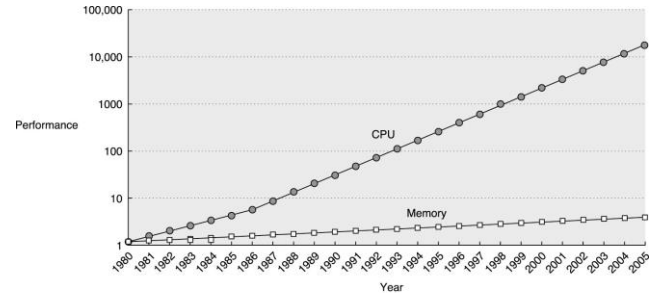
Where Abstractions Break

- Reality check
 - The finite nature of number representation
 - Knowing the ISA helps understand program correctness and performance bugs
 - Memory is bounded and access behavior is not uniform
 - I/O devices highly variable
 - Compatibility and reliability matters

21

21

Problem: Increasing cost of memory access



Reprinted from Hennessy and Patterson, "Computer Architecture: A Quantitative Approach," 3rd Edition, 2003, Morgan Kaufman Publishers.

© 2003 Elsevier Science (USA). All rights reserved.

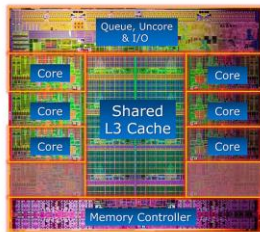
22

22

Caches: A solution to the memory latency problem

- Significant fraction of real-estate consumed up by caches

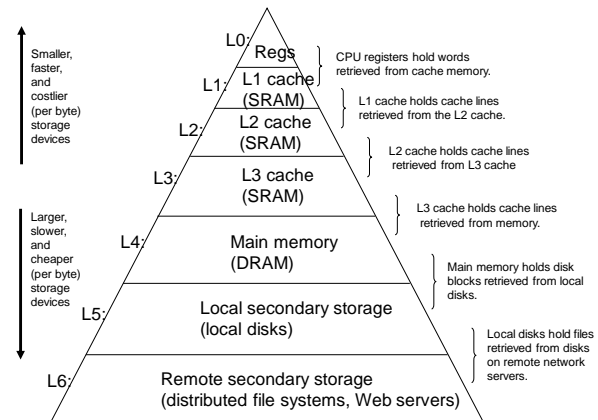
Intel® Core™ i7-3960X Processor Die Detail



23

23

Storage Devices form a Hierarchy

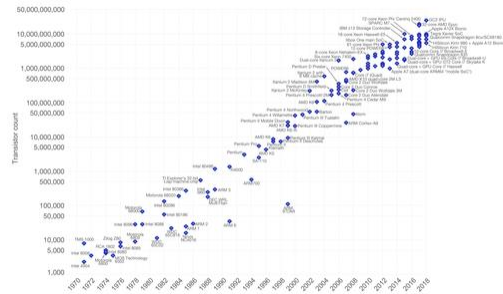


24

Technology Trends

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 achievement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



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.
https://upload.wikimedia.org/wikipedia/commons/thumb/8/8b/Moore%27s_Law_Transistor_Count_1971-2018.png/1000px-Moore%27s_Law_Transistor_Count_1971-2018.png

25

25

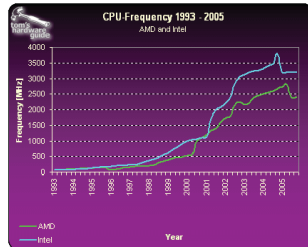
Leveraging Moore's Law

- More transistors – opportunities for exploiting parallelism
 - Implicit parallelism
 - Pipelining
 - Superscalar
 - Explicit parallelism
 - Streaming and multimedia processor extensions
 - E.g., MMX, AltiVec
 - Very long instruction words (VLIW)

26

Uniprocessor Limits

The power problem!



http://www.tomshardware.com/2005/11/21/the_mother_of_all_cpu_charts_2005

27

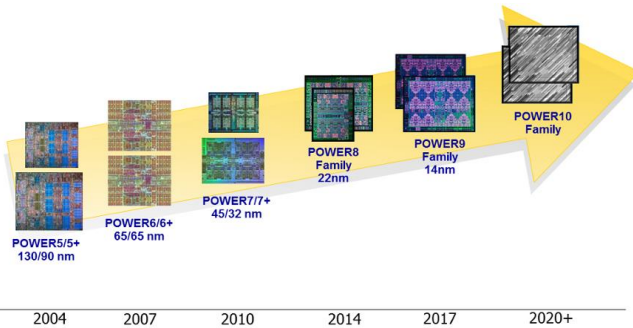
27

Current Trends

- Problems:
 - Fundamental circuit delay and heat limitations
 - Limited amount of instruction-level parallelism
- Solutions: proliferation of (from Sun, IBM, Intel, Nvidia, ...)
 - Accelerators
 - Multithreading
 - Multicore
 - Multiprocessors

28

Explicit Parallelism is here to stay!



<https://www.innovativeis.com/wp-content/uploads/2018/03/IS-Web-Page-slides-POWER-roamap-2022.png>

29

29

Qualcomm's Snapdragon, NVIDIA's Tegra



<http://www.gsmarena.com/news.php3?Tag=Qualcomm>

30

Top 500 List of Supercomputers (www.top500.org – Nov. 2019)

- Top 5 from the list
 - Summit, IBM Power and NVIDIA at Oakridge (ORNL), 2,414,592 cores, 148.6 (200.7 peak) Peflops/sec on Linpack, 10.1 MW
 - Sierra, IBM Power and NVIDIA at DOE/NNSA/LLNL, 1,572,480 cores, 94.6 (125.7 peak) Pflops/sec, 7.4 MW
 - Sunway TaihuLight, Sunway at NSC in Wuxi China, 10,649,600 cores, 93 (125.4 peak) Pflops/sec, 15.3 MW
 - Tianhe-2A, RIKEN Intel Xeon at NSC in Guangzhou China, 4,981,760 cores, 61.4 (100.6 peak) Pflops/sec, 18.5 MW
 - Frontera, Dell/Xeon at TACC UT Austin, 448,448 cores, 23.5 (38.7 peak) Pflops/sec



31

A Caveat: Amdahl's Law

- Speedup is a function of the fraction α of the overall execution improved (by a factor of k):
 - $T_{new} = (1 - \alpha)T_{old} + (\alpha T_{old})/k$
 - Overall Speedup = $T_{old}/T_{new} = \frac{1}{(1-\alpha) + \frac{\alpha}{k}}$

32

32

Topics to be covered:

- Data representation and computer arithmetic
- Assembly-level programs and instruction-set architectures
- Processor architectures
- Memory and storage hierarchies
- Performance optimization
- Exceptional control flow
- I/O devices
- Concurrency

33

33

Getting Help

- Class web page:
<http://www.cs.rochester.edu/u/sandhya/csc252>
 - Complete schedule of lectures, exams, and assignments
 - Lecture slides, assignments, exams, solutions
- Blackboard Discussion Forum
- TA Office Hours

34

Action Items

- Get a CSUG account
 - at <https://accounts.csug.rochester.edu/>
 - cycle1.csug.rochester.edu (or [cycle2](https://cycle2.csug.rochester.edu/), [cycle3](https://cycle3.csug.rochester.edu/))
 - Get familiar with using Linux and C
 - Attend an office hour this week!
- Accept the Academic Honesty Policy on blackboard
- Introduce yourself: “meet your classmates” forum
- Acquire the textbook for the course
 - Read Chapter 1, start reading Chapter 2
- Finish Quiz 0

35

35