CS 2461
Computer Architecture 1
i.e., Introduction to Computer Systems

https://GW-CS2461-2022.github.io/

Fall 2022

Instructor: Dr. Bhagi Narahari

1

CSCI 2416 Fall 2022 Instruction team

- You will learn more from them than from the Instructor!
- Undergraduate TAs & LAs:
  - Jonathan Lee – BS CS Class of 2023 (Senior)
  - Lauren Hahn – BS CS Class of 2023 (Senior)
  - Sam Kusner – BS CS Class of 2024 (Junior)
  - Kate Halushka – BS CS Class of 2024 (Junior)
  - Karl Simon – BS CS Class of 2024 (junior)
- Grad TA:
  - Ruining Yang – MS CS Class of 2023
  - Grader – TBD

instruction team will have “office hours” and will be helping with in-class/in-lab activities

UTAs will be leading/teaching the lab sections

2

2

Course Structure

- Lecture materials – Read/View **before** class
  - Posted on course webpage: gw-cs2461-2022.github.io
  - In some cases, videos of lecture topics
  - Slides, Notes, and examples (Circuits, Code,..)
- lecture session activities
  - Work on solving problems in teams with instruction team
- Lab sessions
  - tutorials posted – watch them
  - labs will cover content/topics not covered in lecture (Testing, Debugging, Unix tools,) and
    - Reviews
    - Exercises/Demos/Experiments
    - May be asked to submit lab work for grading

3

3

Course Logistics….Where do I go to get course information & materials ?

- Blackboard:
  - Homeworks & Grades
  - Online class (lecture and lab) – recordings from these sessions
- Website: https://cs2461-2022.github.io/
  - Syllabus – schedule, grading criteria, contact info
    - Lecture notes(slides, exercises, code samples, circuits,…)
    - Tutorial Videos linked from website
- Github: projects, and code submission..
- Piazza:
  - General discussion– post questions to instructors or classmates, all students benefit from the discussion Q&A
  - Announcements – announcements from instruction team
  - Can also direct a question to instructors

4

4

Piazza discussions

- Online discussion forum…purpose:
  - to encourage students to ask well formed questions
  - To encourage students to answer each others questions
    - Most of the time, you do this better than we do!
  - *Be very careful not to border on plagiarism!*
  - *Don't post your HW solution to the world,*
- We will send you signup link
- Do not expect instant response or substitute slack for TA office hours!
  - not manned 24 hours/7 days a week
  - sometimes answer may take more than 24 hours!
  - Mainly a way for students to help one another with common questions/misunderstandings
    - Not a substitute for office hours
- NO TA can excuse you from anything/or give any extensions

5

5

Piazza

- Online discussion forum (with anonymous posts enabled)
- The purpose of this:
  - to encourage students to ask well formed questions
  - To encourage students to answer each others questions
    - Most of the time, you do this better than we do!
  - *Be very careful not to border on plagiarism!*
  - *Don't post your HW solution to the world,*
- Signup email will be sent…check, and sign up.
- Do not expect instant response or substitute Piazza for TA office hours!
  - Piazza is not manned 24 hours/7 days a week
  - sometimes answer may take more than 24 hours!
  - Mainly a way for students to help one another with common questions/misunderstandings
    - Not a substitute for office hours
- NO TA can excuse you from anything/or give any extensions
- Posting on piazza, not the same as telling instructor things
  - E.g. : I'm going to miss the exam! (cannot do this)

6

6

In-class exercises/activities

- This course is designed to help you learn through in-class exercises (lectures and labs)
  - For this to work, you must review the material and come to class
- We want you to complete the exercises while working as a group
  - Each group is assigned to a breakout room and will have a member of the instruction team to help.
  - We may ask a group to present solutions to class
- In-class questions/exercises counts towards your class participation grade

7

7

Course Schedule

- Part 1 (8-9 weeks) of the course spent on hardware stack and HW/SW Interface
  - From transistors to the design of a simple processor
    - Implementation of a simple processor ISA
  - Assembly programming
- Part 2 of the course (5-6 weeks) spent on C Prog Lang. and translation to Assembly
  - Quick review of C (you will cover some C in CS2113)
  - How are C constructs compiled into (LC3 assembly) machine language
  - Managing Memory
    - Stack
    - Heap
  - How to make your programs run faster

8

8

Requirements and Grading: Read website for details on how grade is computed

- 40% Exams: Two exams
  - Will be held approximately Weeks 7,12
  - Exam may also have an 'interview' (oral exam) component if necessary
    - Conducted by instructor and TA
- 18% Homework and Lab assignments
  - Some lab assignments may require completion within lab time
    - No late submissions….except a "one time pass" of 36 hours
- 10% Class participation and Quizzes
  - 9 quizzes, will drop lowest score
    - **Start of class** – if you join late, you miss the quiz
  - Class participation – includes inclass activities.
- 32% Projects: about 6 projects
  - 2-3 of them will have a teamwork component.
  - You may be asked to demo and explain some of your projects – be prepared to be examined on any (random) aspect of your solution (code, design, etc.)

9

Academic Integrity

- You are here to learn – so keep that in mind
- Strictly enforced!
  - "no collaboration" means none of any kind
  - No asking friends
  - No searching on web for answers
- Violations will lead to at least a zero on the work and a grade lower than final grade..and formal report to the Integrity council.
  - A 2nd violation goes on your transcript
- Stay on top of your work – come ask me/TAs questions!
- PDT: Plagiarism detection software tool
  - I will be running code submissions through a software tool (MOSS)
  - Any pair of submissions with more than 25% similarity will be closely examined and reported to the acad. integrity panel

10

10

Textbooks/Software

- Intro. to Computing Systems, by Patt and Patel
  - Easy to read – fairly comprehensive.
  - The latest edition is the 3rd edition – but 2nd edition will work just as well
- Other useful books: Unix for Programmers, C Programming

- LC3 simulator: Links on the course webpage
- C compiler – gcc (use shell.seas.gwu.edu)
- Hardware simulators: CedarLogic
  - We will not be supporting Logisim
  - Use CedarLogic in the SEH4040 lab – and in Tompkins 4th floor labs…you have access
  - You can also remote login (remote desktop) to the lab computers.

11

11

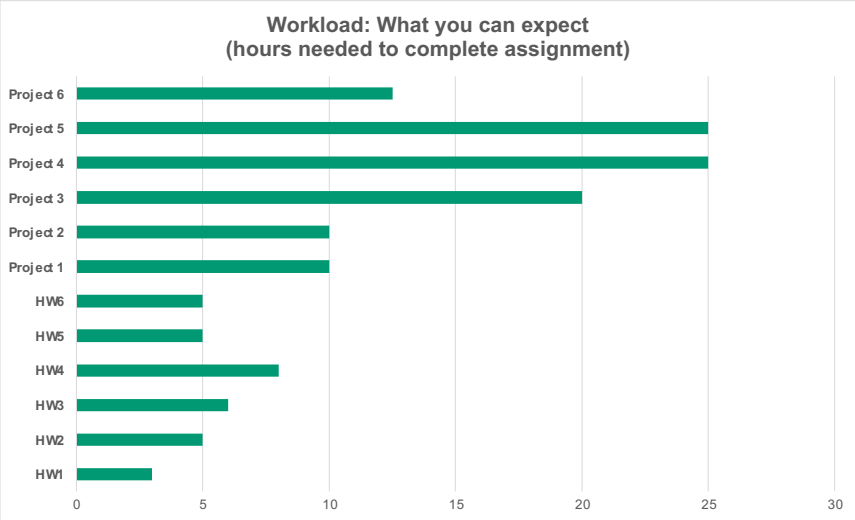
Expectations

- Come prepared to class
  - Read the notes/textbook and watch the videos (when available)
- work on problems assigned in class, ask questions
  - This is when you make sure you have learned the concept correctly
- need to spend *at least* 6 hours per week outside class
- This is considered a hard course because you will be seeing a lot of new concepts/topics
  - Practice, practice, practice…especially your programming skills
- You will be expected to learn some materials on your own…
  - This is only the beginning..things get more demanding when you get to your junior year….ask the TA team (they have been through this 'journey')
- Academic Integrity: No collaboration on assigned work

12

12

Course workload: What you've signed up for….



13

What is CS 2461 about?!

- Look 'under the hood' to see how a computer works
  - Explore the interface between hardware and software
  - Understand the components in a processor
  - Bottom up approach: from transistors on up to algorithm design
    - i.e., the *hardware stack*
- With this knowledge you can
  - Understand the link between hardware and software
  - Write better, more efficient software
  - Design better hardware
    - Link between hardware and software
  - Appreciate the abstractions that are built on top of these foundations

14

14

Pre-requisites

- Pre-requisites
  - CS 1112 – Data structures and algorithms
  - CS 1311 – Discrete Math 1
- Co-requisite: CS 2113
  - Knowledge of C programming language
  - I will be synchronizing with instructor
- Programming practice…system skills
  - Practice, practice,…and more practice

15

15

Course Objectives: What is CS2461 about?

- To understand the structure and operation of a modern computer system from the ground up.
  - Understand basic hardware concepts and design simple circuits
  - Understand the Von Neumann architecture/computing model
    - Assembly language, Processor design)
  - Introductory "system" concepts
    - runtime stack, simple I/O devices, Unix OS
- How high level languages are implemented on the machine (using the C language)
  - How are C programs translated to assembly and implemented on a machine
  - Proficiency in the C programming language
- Understand how software/program performance is linked to program and machine properties

16

16

Two recurring themes in Computer Sci.

- Abstraction: Productivity Enhancer
  - You don't need to worry about the details
    - You can drive a car without knowing about the internal combustion engine….until something goes wrong: where is that smoke coming from !!
  - The notion that we can concentrate on one "level" of the big picture at a time, with confidence that we can then connect with other levels.
- Hardware and Software
  - hardware and software are inseparably connected, especially at the level we will be studying
    - Even if you specialize in one, you must understand the capabilities of the other

17

17

What are Computers meant to do ?
A question to anchor our discussions

- Solve problems that are described in English (or Greek or French or Hindi or Chinese or ...) and use a box filled with electrons and magnetism to accomplish the task.*
  - This is accomplished using a system of well defined (sometimes) transformations that have been developed over the last 50+ years.
  - As a whole the process is complex, examined individually the steps are simple and straightforward
  - *Definition from the textbook*
- *So how do you get the electrons to run around and do our task ?*

18

18

Two Big Ideas in Computing

- Universal Computational Devices
  - Church-Turing Thesis: every computation can be performed by some "Turing Machine" - a theoretical universal computational device
    - – You will see this in the Foundations course CS 3313
- Problem Transformation (Abstraction!)
  - The ultimate objective is to transform a problem expressed in natural language into electrons running around a circuit (using a succession of transformations)
    - That's what Computer Science and Computer Engineering are all about: a continuum that embraces software & hardware.
    - Note the role of compilers/translators

19

19

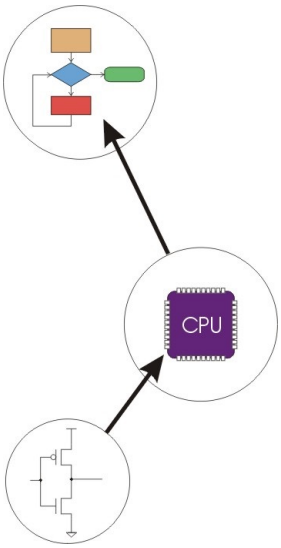
Big Idea #1: Universal Computing Device

- All computers, given enough time and memory, are capable of computing exactly the same things
  - Smartphone, laptop, supercomputer
    - Limited only by time and memory (and energy)
- Anything that can be computed, can be computed by a computer
  - If you can describe something in terms of computation, it can be done by a computer
- Formal (mathematical) model of computing = Turing Machine (1936)
- If all of them have same capabilities which one do you pick
  - Performance
  - Cost
  - Energy/power

20

20

Big Idea #2: Transformation between layers (Abstraction!): Putting the electrons to work!



- **Problems**
- **Algorithms**
- **Program**
- **Instruction Set Architecture**
- **Microarchitecture**
- **Circuits**
- **Devices**

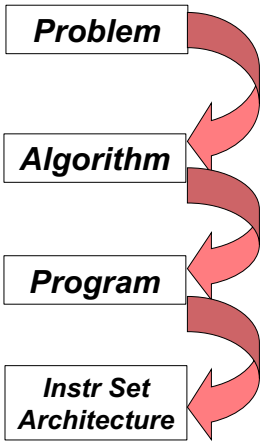
21

21

How do we solve a problem using a computer?

- A systematic sequence of transformations between abstraction layers.

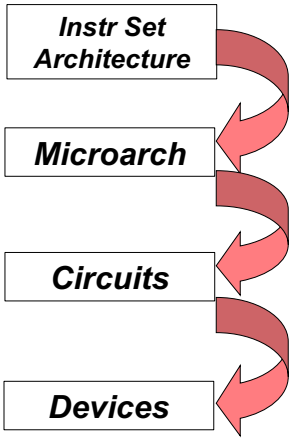
Example: Problem = Sort a set of numbers



- ***Software Design:*** choose algorithms and data structures. Bubble Sort
- ***Programming:*** use language to express design and implement algorithm. C lang.
- ***Compiling/Interpreting:*** convert language to machine instructions. Use gcc compiler (to compile to ARM ISA)

22

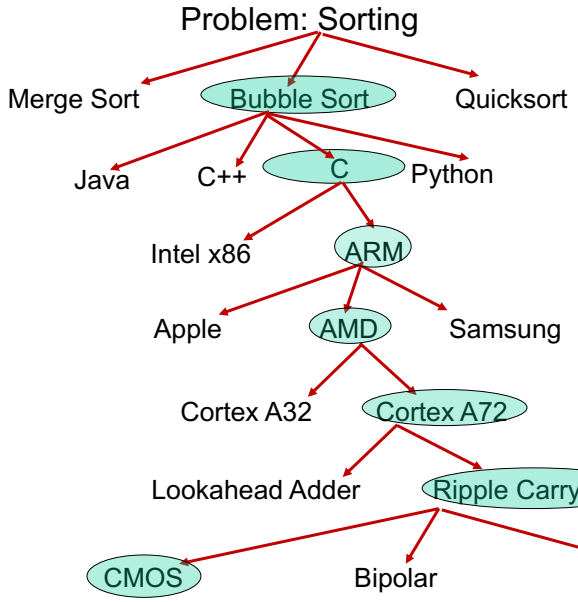
…and even more layers…



- ***Processor Design:*** choose structures to implement ISA. Choose ARM ISA
- ***Logic/Circuit Design:*** gates and circuits to implement components of AMD ARM processor
- ***Process Engineering & Fabrication:*** develop and manufacture **transistors**, and connect them using wires

23

Why do we need 'experts' at each layer – Choices at each layer

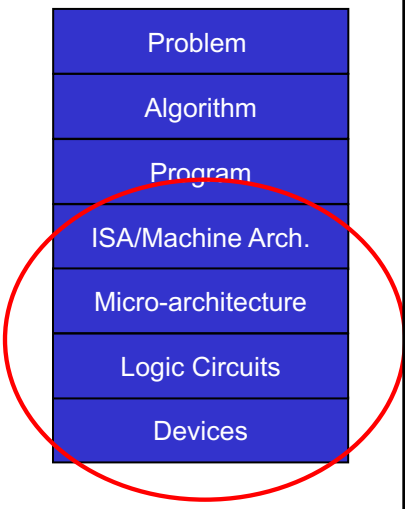


Tradeoffs:
Cost,
Performance,
Power, …

24

24

Our Computing Technology Stack & transformation between layers



25

25

Transistors – the lowest level in our technology stack

- What is the device that controls how electrons "run around" to solve your problem -- **the transistor**
- Question: How many transistors do you think are in a modern day Intel processor ?

26

26

Focus of this course: The Machine/Hardware Level

This is going to be 'all new' material for most of you…

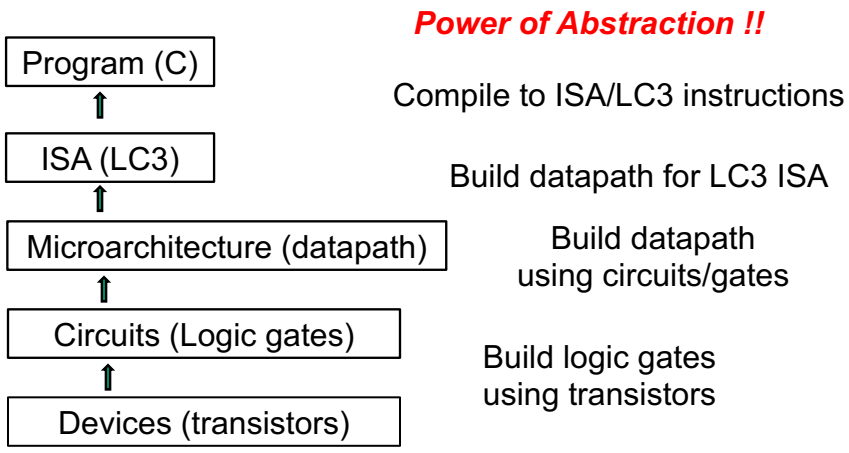
- Machine Architecture: formal specification of operations of processor – the *Instruction Set Architecture* (ISA).
  - We will study the ISA, and Assembly Language programming of a simple computer LC3 – why select a simple "unrealistic" computer?
- Microarchitecture: implementation of the ISA in a CPU.
  - We give an overview of the microarchitecture
- Logic circuits: build each component of Microarch using circuits
  - We study the basic building blocks of logic circuits and logic devices
- Devices: each logic circuit built from transistors
  - we take a quick look at CMOS transistors –device electronics not the focus!

27

27

Bottom up Approach….Power of Abstraction

Bottom-up approach: From bits to C programs



Our starting point is "how to represent data"

28

28

Starting point – how do we represent data in a computer ?

29

29