Computer Systems from the Ground Up

Fall 2024

https://cs107e.github.io

Have you ever wondered ...

- how a computer represents data?
- what operations a computer understands?
- how a program executes?
- what happens when a user types on keyboard?
- how text and drawing appears on a display?

- how things *really* work inside this wondrous box?

These questions and more to be answered by studying **computer systems!**

Learning goals

1) Understand how computers represent data, execute programs, and control peripherals

2) Master your tools

Understanding is empowering!

RISC-V processor and memory architecture

Peripherals: GPIO, timers, UART, …

Assembly language and machine code

Low-level representation of information / bits

From assembly language to C

Function calls and stack frames

Serial communication and strings

Modules and libraries: Building and linking

Memory management: Memory map & heap

Processor control, interrupts

Master your tools

UNIX command line: bash, cd, ls, …

Text editor: vim, emacs, sublime, …

Programming languages: C, assembly

Compiler: gcc

Assembler: as

Linker/loader: ld

binutils: nm, objcopy, objdump, …

make

git and github.com

documentation: markdown


Non-Contact Voltage Detector
EXTECH
INSTRUMENTS
NCV
REL
MODE
HOLD
RANGE
OFF
VAC
VDC
μA
mA
10A
°F
°C
EX330
AutoRanging
MultiMeter
CAT III - 600V
CAT II -1000V
COM
V/mA/μA/CAP
Ω/Hz/Temp
10A
600V MAX
600V
400mA
MAX
FUSED


Consolidated
ELECTRONIC WIRE & CABLE
100
FT
RoHs
36920
36822
36792
36856


SOLID
IDEAL INDUSTRIES
SYCAMORE IL U.S.A.

Organized Development Environment

http://amhistory.si.edu/juliachild/

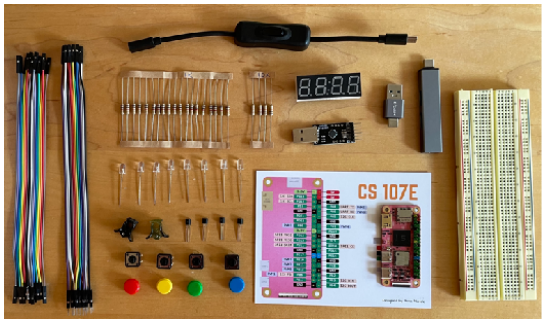

Practice, Practice, Practice
https://paulsellers.com/tag/gouge/

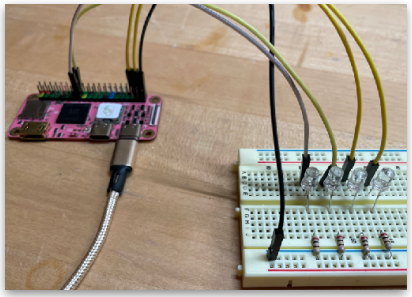
Debugging and Troubleshooting

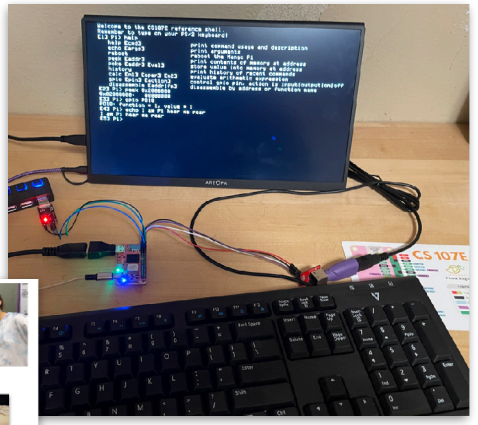
Syllabus

§1 Bare Metal Programming

- RISC-V architecture and assembly language
- C functions and pointers
- Serial communication
- Linking and loading
- Memory allocation

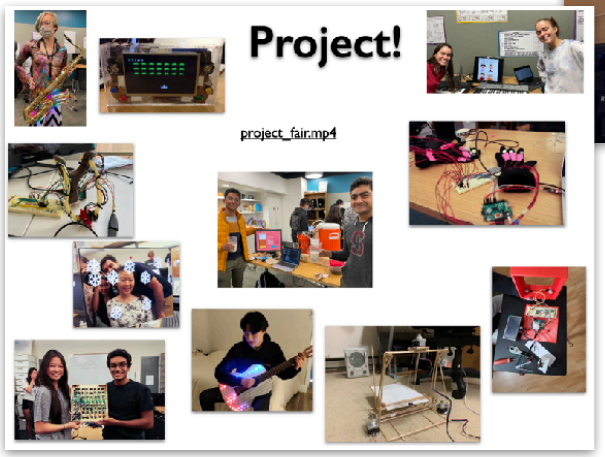






§2 Build a Personal Computer

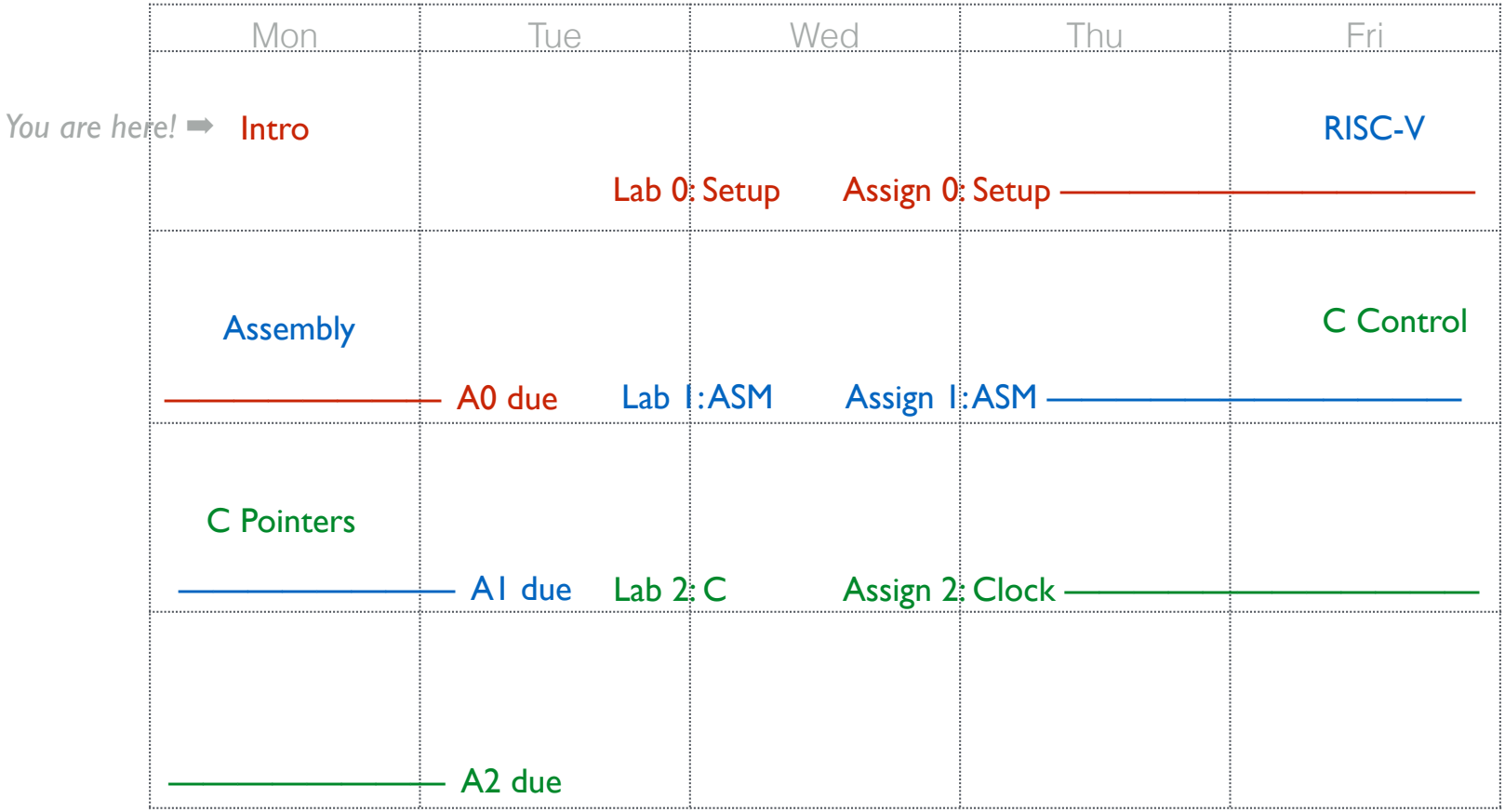
- Keyboard
- Graphics
- Interrupts



§3 Create Your Own Project

- Sensors
- Performance

Weekly Cadence



https://cs107e.github.io/schedule/

Each week has a focus **topic**

Pair of coordinated **lectures** on Fri and Mon

Lab on Tue/Wed evening

Assignment handed out Wed after lab, due following Tuesday 5pm

Staying on pace leads to best outcomes!

Lectures

Attendance is **necessary**

Content is unique to our course, no textbook
The readings/slides not a standalone resource
Lectures not recorded

In-person attendance allows you to participate, ask questions, keep on schedule, stay connected

Labs

Set of guided exercises, follow up on lecture content

- Work in groups
- Complete exercises and check in with staff
- Leave lab ready to start assignment!

Lab participation is mandatory

Philosophy: lab is hands-on, collaborative, supported, **fun**!

Assignments

Weekly assignments that build on each other

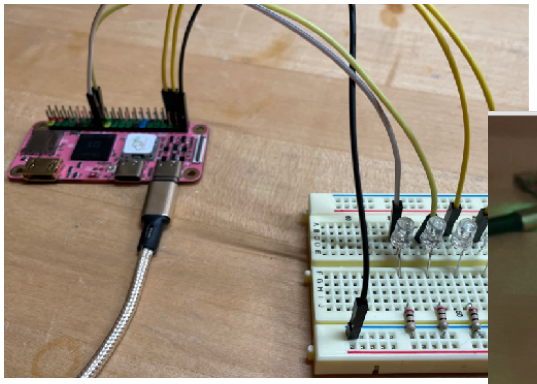
This is where the learning really happens!

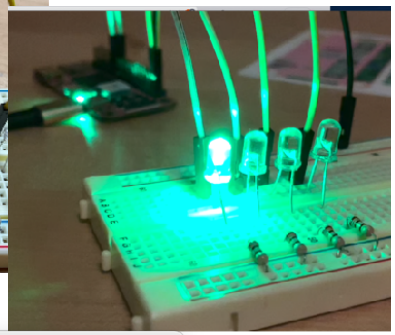
Each assignment has

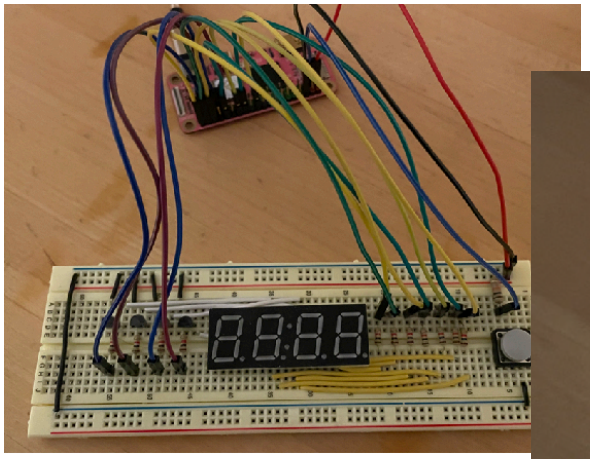
- **Basic** requirement (tight spec, guided steps)
- Optional **extension** (opportunity for exploration/ creativity)

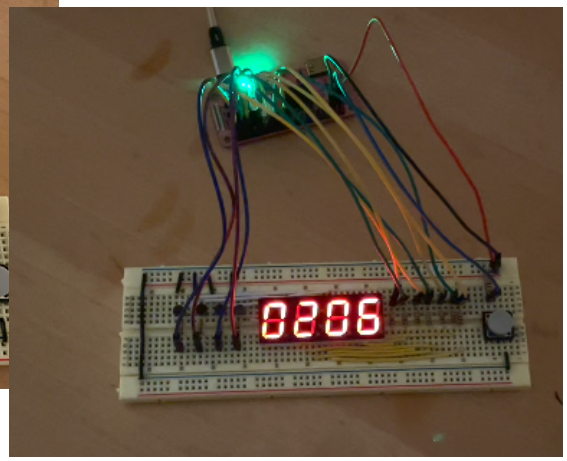
Opportunity to revise/resubmit to correct missed basic requirements

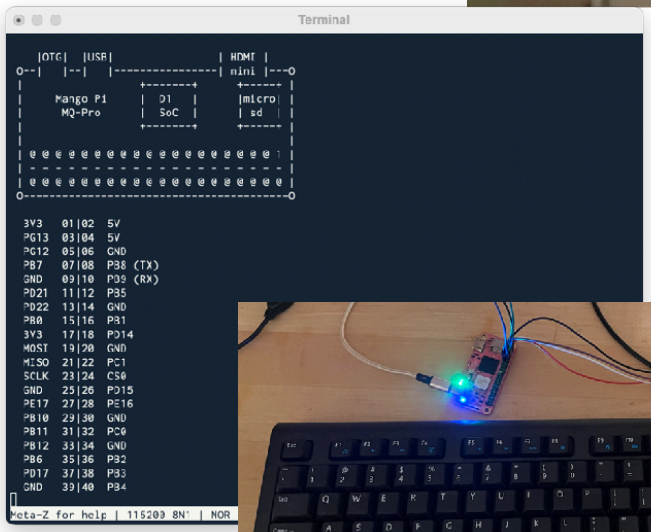
End goal is complete working system of your own

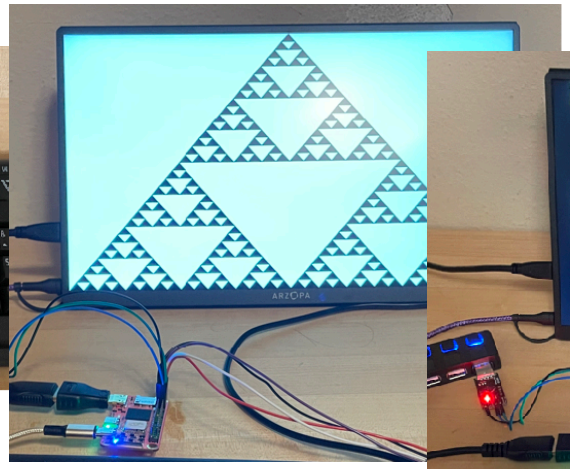


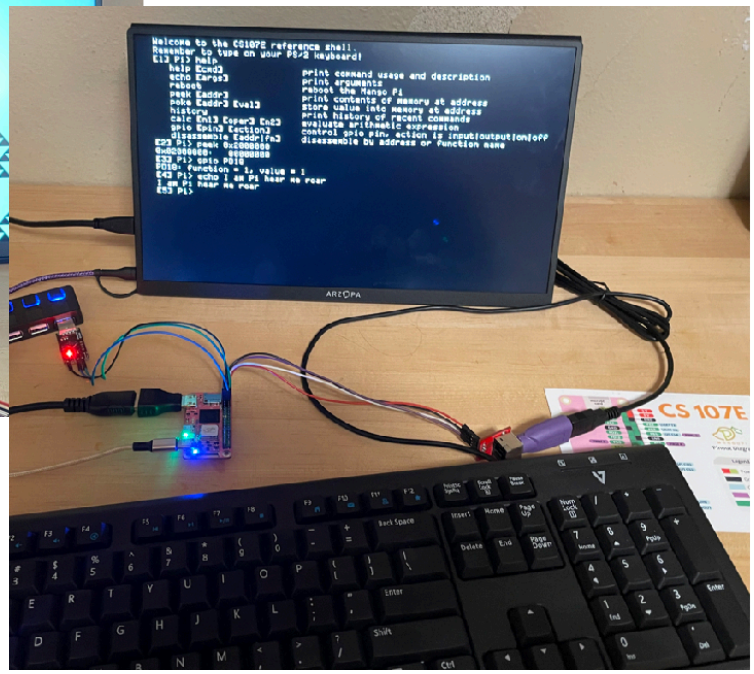






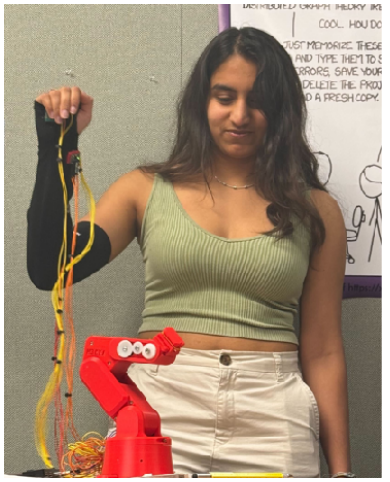


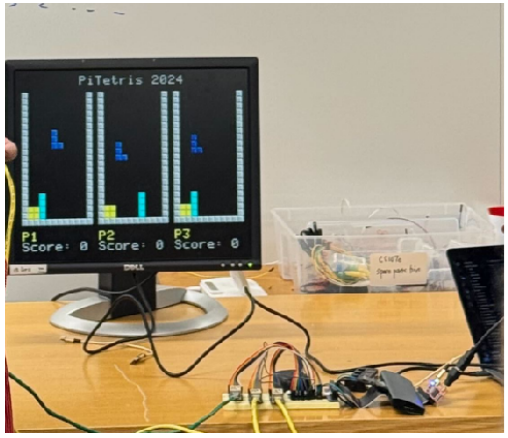


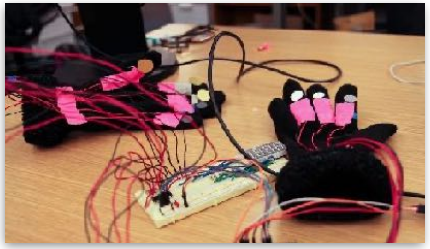


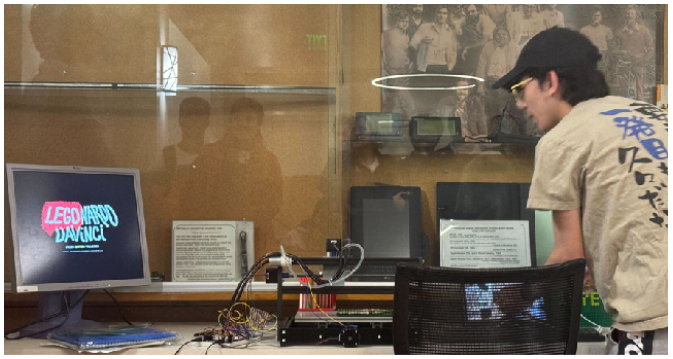
Nearly every instruction is code you wrote yourself!

Projects!




PiTetris 2024
P1
Score: 0
P2
Score: 0
P3
Score: 0




LEGONARDO DAVINCI

Markers for success

- Necessary prereqs: CS106B, C++, debugging

- Curiosity
- Perseverance
- Motivation

Value and appreciate small learning community!

How to thrive in this course

- Consistency, follow through
- Leverage our resources, support, feedback
- Be **curious**. Learn by **doing**. Ask for and offer **help**.

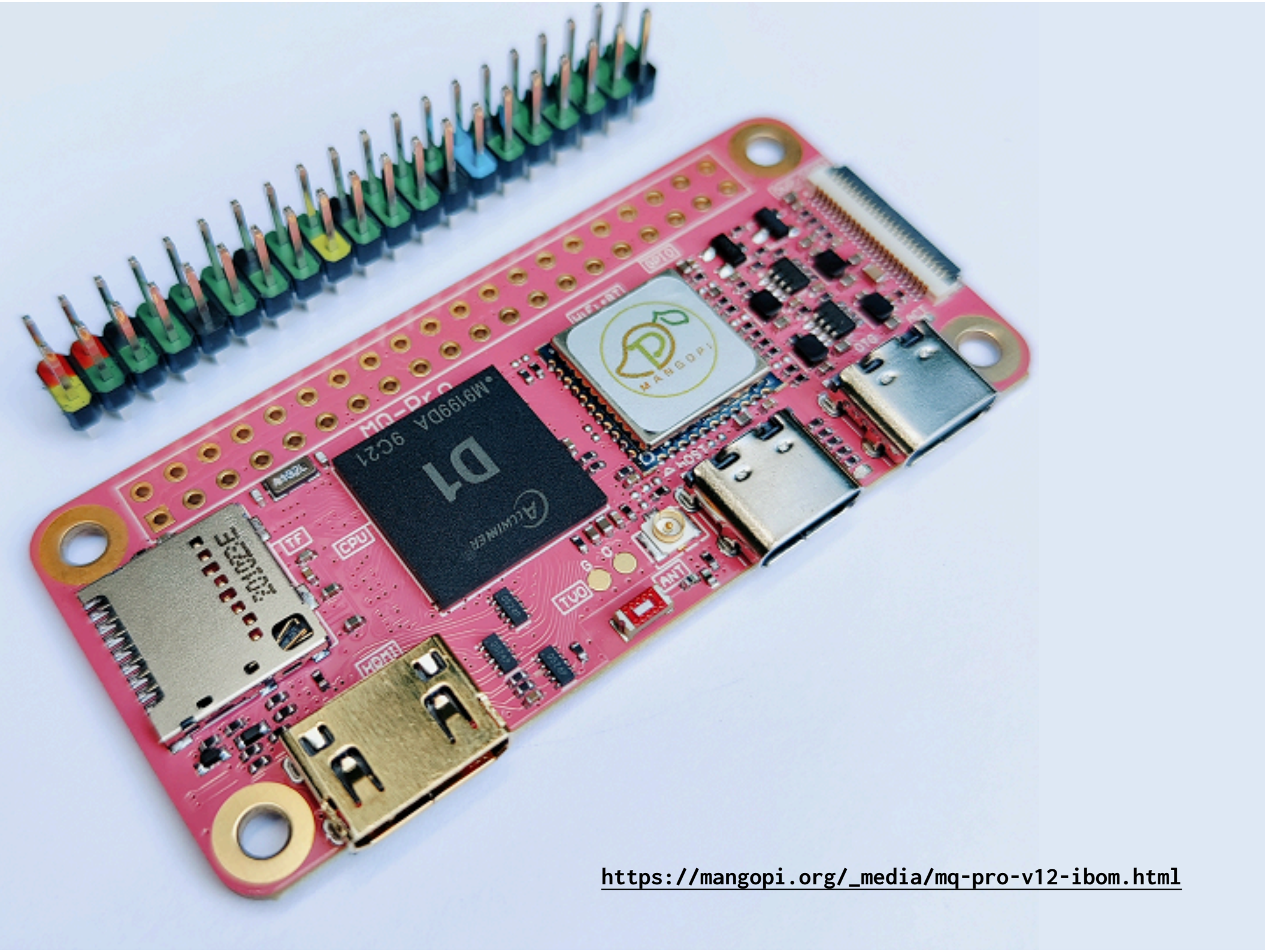
Enrollment process

We can enroll up to 40 students

Lightweight questionnaire for you to share your interest and fit (linked at https://cs107e.stanford.edu)

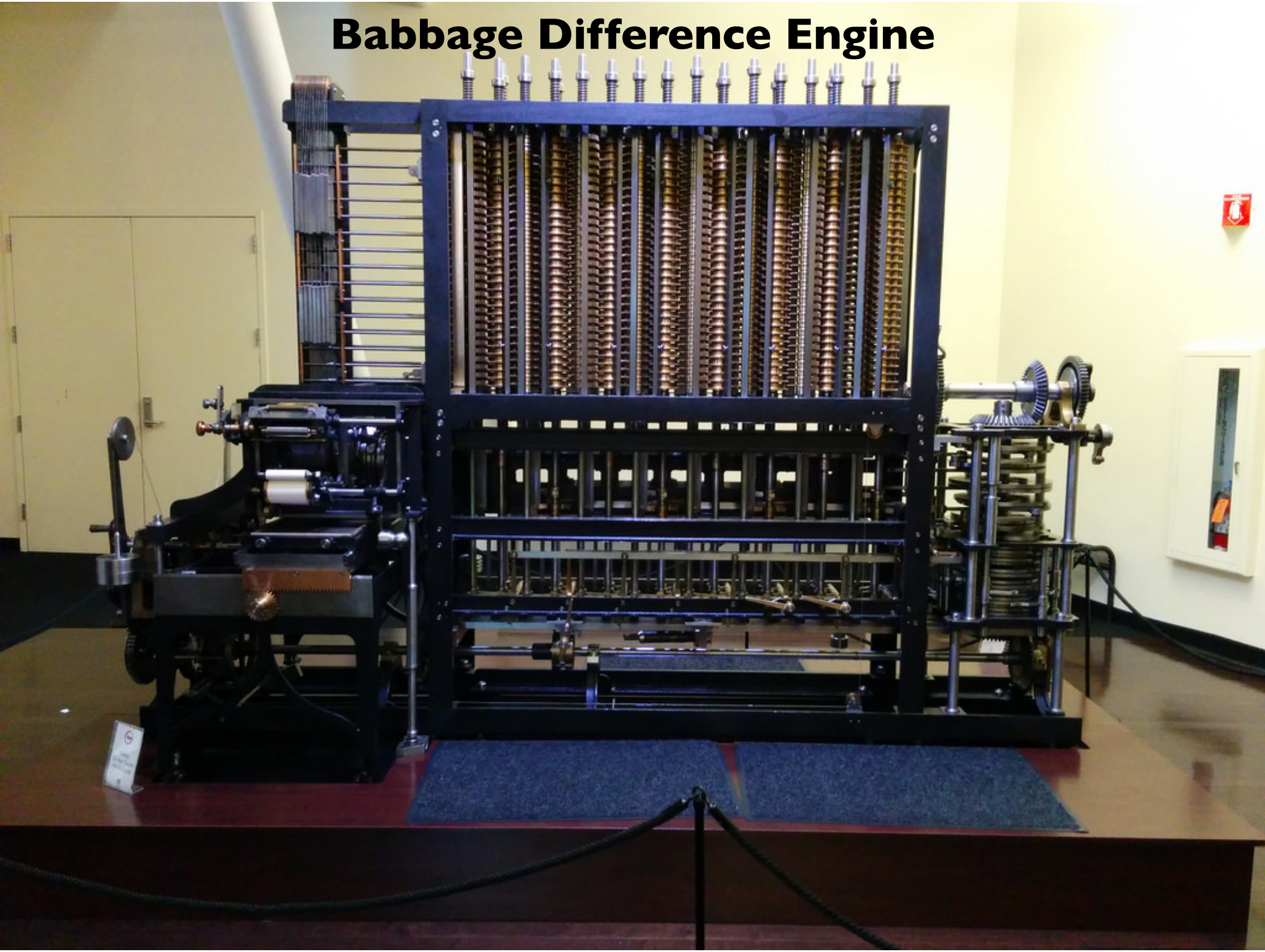
Questionnaire submissions due 5pm today (**firm**!)

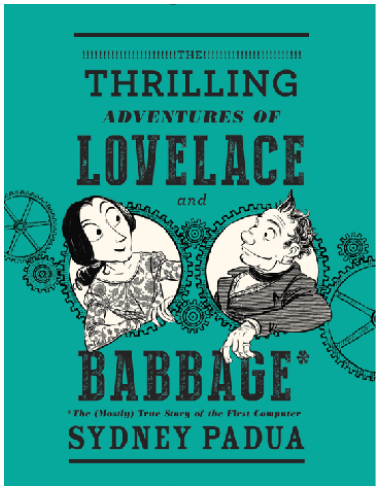
Email our decisions by tomorrow am



https://mangopi.org/_media/mq-pro-v12-ibom.html

Babbage Difference Engine





Analytical Engine

Universal Computer

http://www.theguardian.com/technology/2015/apr/12/thrilling-adventures-ada-lovelace-charles-babbage-sydney-padua

von Neumann architecture

CPU:
- ALU
- Registers
- Control Unit
  - instruction register, program counter

Memory:
- Stores data and instructions

Mechanisms for input/output

Critical innovation:
- both instructions and data kept in memory
- "stored program" computer

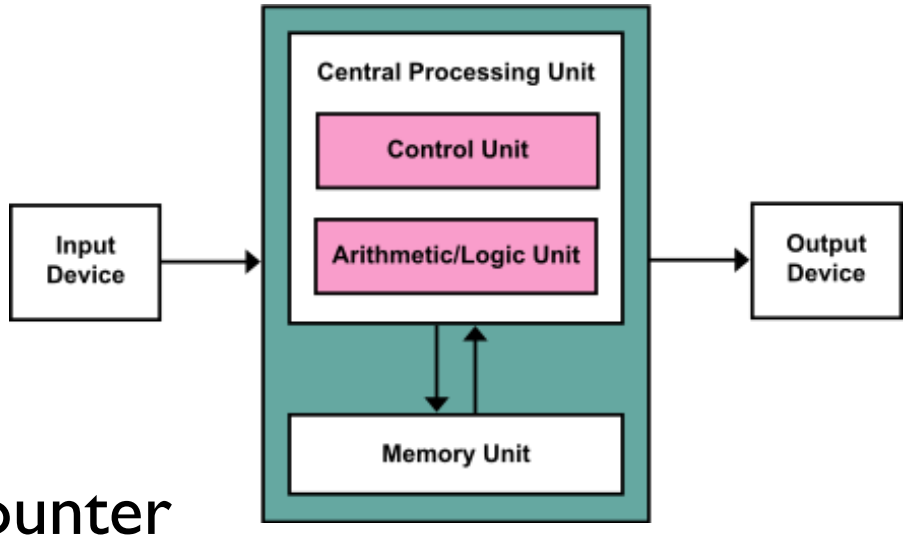


Image credit https://en.wikipedia.org/wiki/Von_Neumann_architecture#/media/File:Von_Neumann_Architecture.svg

Memory Map

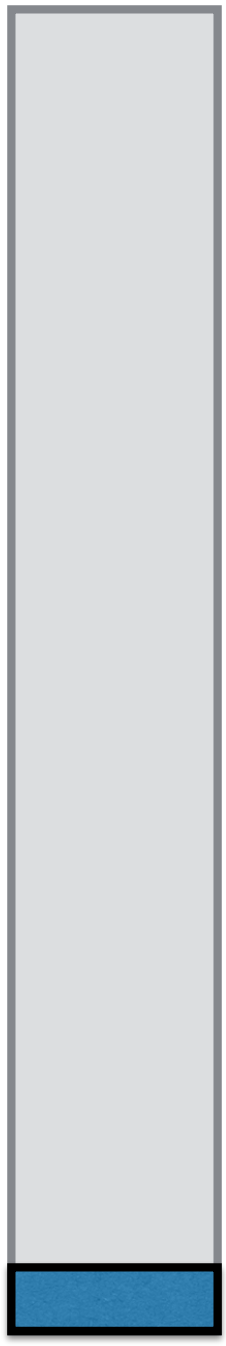
Memory is a large array

Storage locations are accessed using a 64-bit index, called the *address*

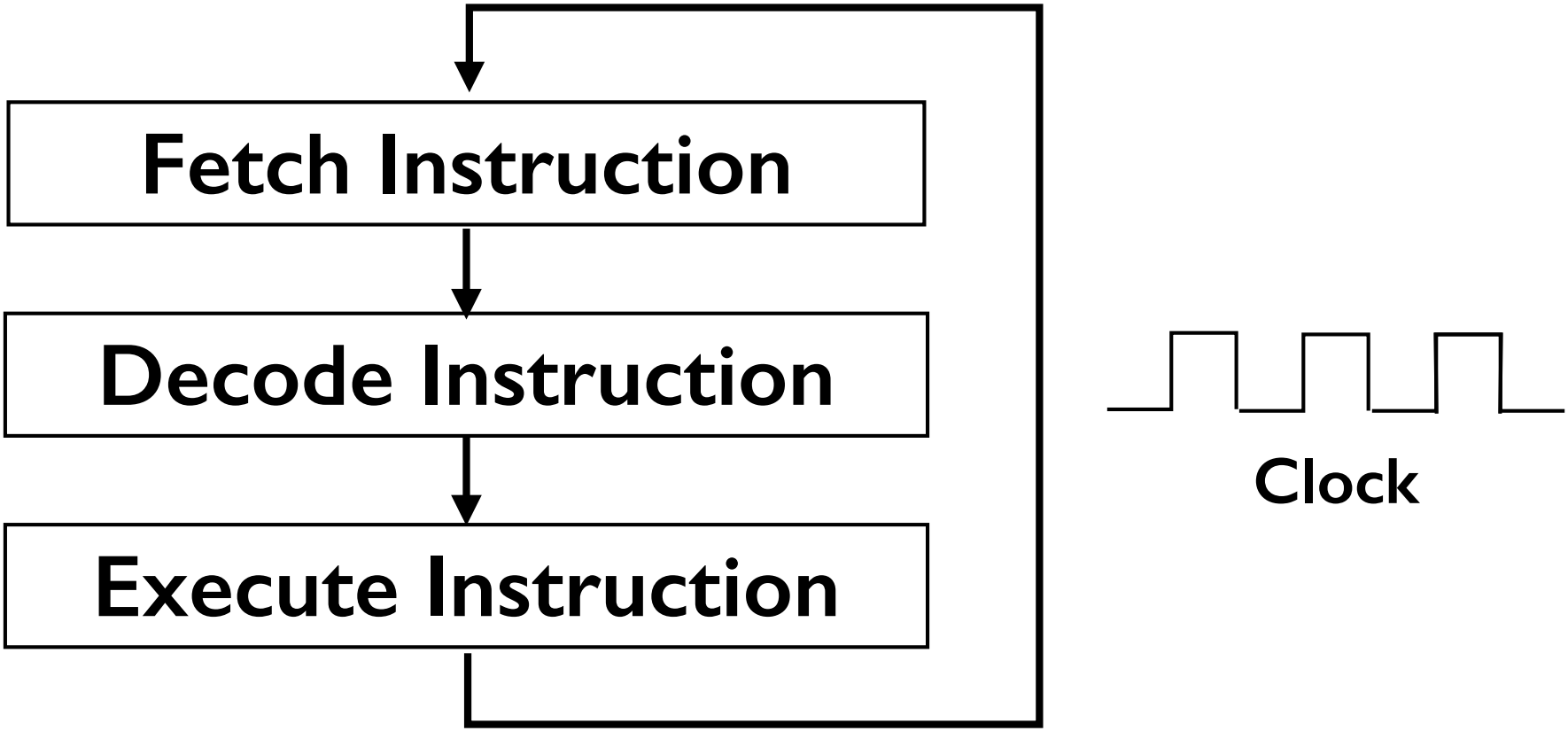
Address refers to a *byte* (8-bits)

4 consecutive bytes form a *word* (32-bits)

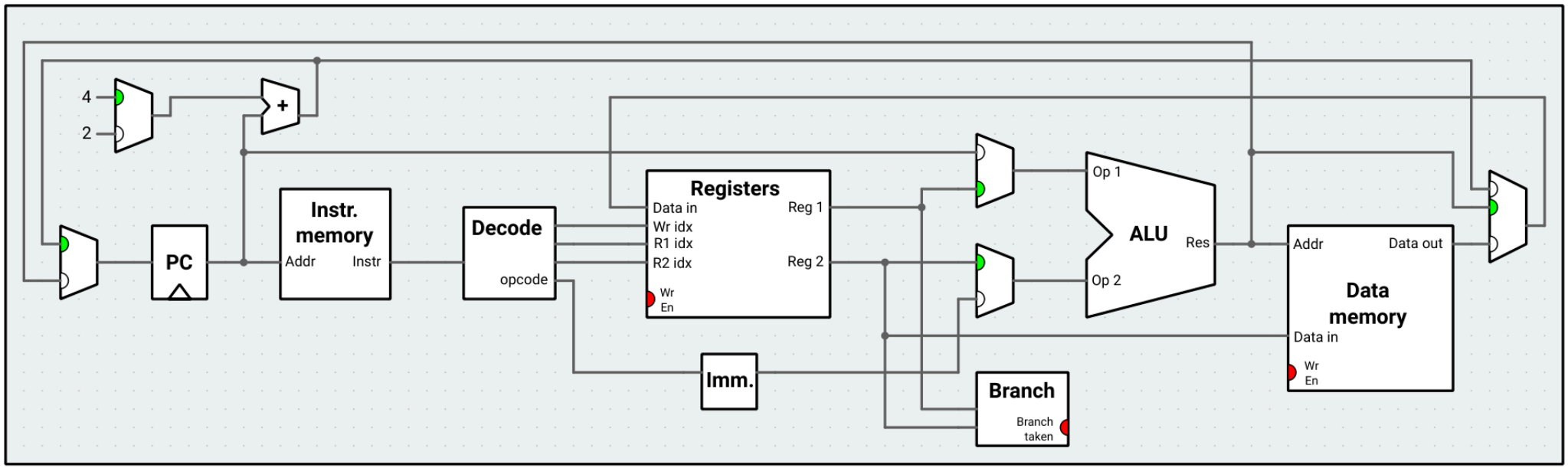
Maximum addressable memory is 16EB (But... 42-bit address lines)



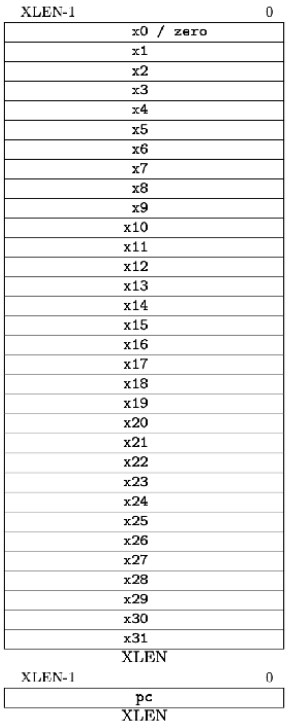
Running a Program



RISC-V Architecture / Floor Plan



https://ripes.me/

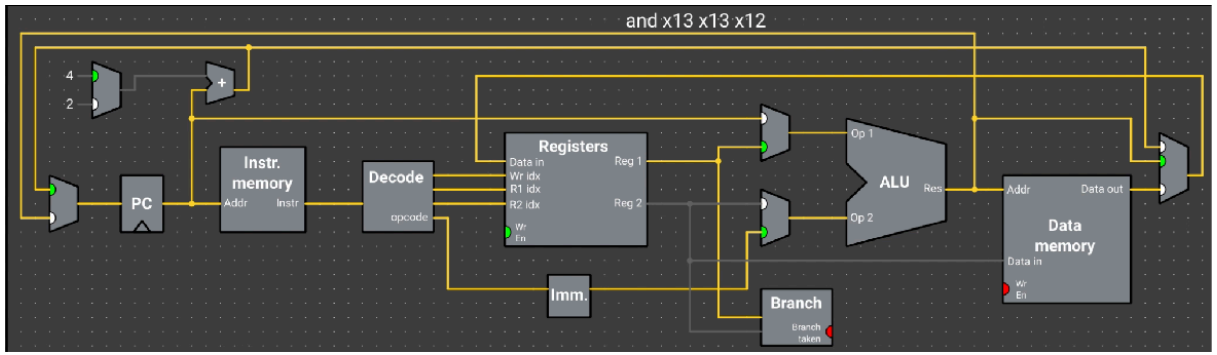


32 registers (x0-x31)

PC = program counter

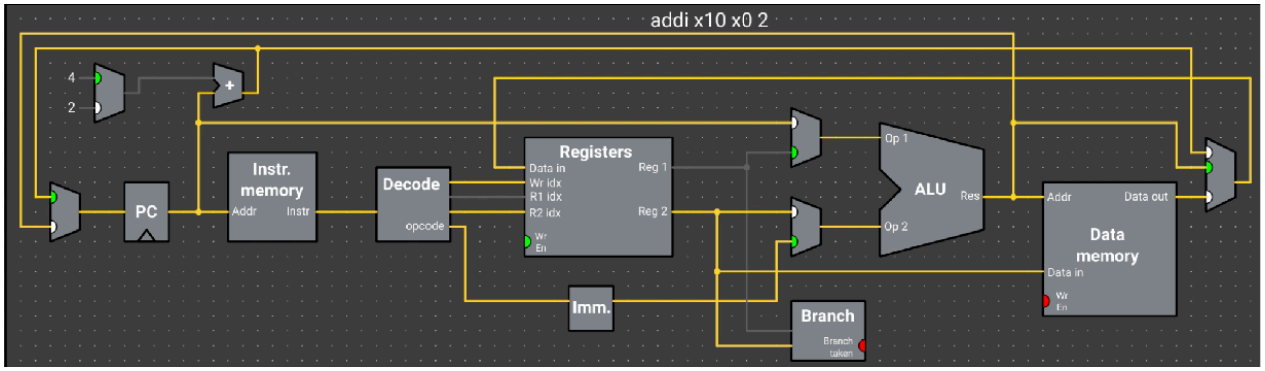
Data paths dictate constraints on operations

```
and a3,a3,a2
```



What are possible inputs to ALU?
Where does result go?

```
addi a2,zero,2
```



Where does an immediate value come from?
Where can it flow to?

Some ALU instructions

```
add rd,rs1,rs2
sub rd,rs1,rs2
and rd,rs1,rs2
or  rd,rs1,rs2
sll rd,rs1,rs2
srl rd,rs1,rs2
```

R-type (three registers: dest, source1, source2)

```
addi rd,rs,imm12
andi rd,rs,imm12
ori  rd,rs,imm12
slli rd,rs,imm12
```

I-type (source2 is constant not register)

First week: set up

Before lab:
- Review course guides:
    Unix tools (shell, editor, git)
    Electricity (simple circuits, Ohm's law)
    Number representation (binary, bit operations)
- Install development tools

During lab:
- Establish comfort with background topics
- Practice with environment/tools, build productive habits
- Get help resolving any installation snags
- Meet one another!